Computer Security
Homework 2

Sebastián Romero 00216765

Exercise 1:

1. **Suppose a password is chosen as a concatenation of seven lower-case dictionary words. Each word is selected uniformly at random from a dictionary of size 50,000. An example of such a password is "mothercathousefivenextcrossroom". How many bits of entropy does this have?**

   L = Password Length; Number of symbols in the password
   S = Size of the pool of unique possible symbols (character set).
   Number of Possible Combinations = S^L
   Entropy = log2(Number of Possible Combinations)

   L=7
   S=50000
   Entropy=log2(50000^7)=109.267…

2. **Consider an alternative scheme where a password is chosen as a sequence of 10 random alphanumeric characters (including both lower-case and upper-case letters). An example is "dA3mG67Rrs". How many bits of entropy does this have?**

   L=10
   S=26*2+10
   Entropy=log2(62^10)=59.541…

3. **Which password is better, the one from 1. or 2.?**

   The first password with 109.2 bits of entropy is significantly stronger than the second password with 59.5 bits of entropy. It would be much more difficult for an attacker to guess the first password through a brute force attack, as it has a larger number of possible combinations.

Exercise 2:

1. **Design a data verification system using hash functions. Explain the steps involved in the process.**

   For this system, the sender hashes the message using a secure hash algorithm like SHA-256 and sends the message+hash to the receiver.
   The receiver hashes the received message using the same algorithm and compares the output with the received hash, if it is the same then the

message is verified. If the hashes do not match, then the receiver can conclude that the data has been altered during transmission.

2. **Discuss the advantages and disadvantages of using hash functions for data verification.**

   Pros:

   - It is not expensive to compute.
   - It is deterministic, they consistently produce the same hash value for the same input.
   - Produces fixed-size hash values, regardless of the size of the input.

   Cons:

   - It is susceptible to collisions.
   - Using weak or outdated algorithms will reduce its secureness.
   - Properly storing keys is a challenge.
   - It does not provide non-repudiation.

3. **Provide an example of a real-world application where a data verification system using hash functions is used.**
   - Git uses SHA-1 hashes to track changes and ensure integrity of code revisions.
   - It is used on cryptocurrency blockchain transactions.
   - Anti-virus use hashes of known malware to detect malicious code.

Exercise 3:

1. **Define what a Message Authentication Code (MAC) is and how it is used in cryptography.**

   Is a piece of cryptographic code that authenticates a message and ensures integrity and authenticity.

   It is generated from a message using a symmetric key and a hash function or cipher algorithm. The same key is used by the sender to generate the MAC and by the receiver to verify it.

   MACs are used in secure communication protocols and cryptographic systems, providing a way to verify the authenticity and integrity of data in applications like secure messaging, digital signatures, and network security.

2. **Explain the process of generating and verifying a MAC.**
    - The sender has a message and a symmetric key.
    - Sender computes the MAC using inputs like the message and key.
    - The message + MAC is sent to the receiver.
    - Receiver gets the message + MAC.
    - Receiver computes the MAC using the same inputs and algorithm.
    - Receiver compares the calculated MAC' to the received MAC. If they are equal, the message's integrity and authenticity are verified.

3. **Discuss the importance of using MACs in secure communication systems.**

    MACs provide important cryptographic protections for message integrity and authenticity in communication systems, complementing encryption mechanisms in a lightweight and efficient manner. It provides authenticity by using symmetric keys only known to sender and receiver, as well as preventing replay attacks through use of timestamps, sequence numbers, or one-time keys. Overall it offers additional confidentiality assurances beyond just encryption

Exercise 4:

1. **Given the values of p = 17 and q = 23, generate a pair of keys for RSA.**

```
import math

p = 17

q = 23

# Compute n

n = p * q

# Compute phi(n)

phi = (p-1) * (q-1)

# Choose e coprime of phi, 1<e<phi

e = 35

# Compute modular inverse of e

d = pow(e, -1, phi)
```

```
print("Public key (n,e): ", n,e)

print("Private key (n,d): ", n,d)
```

Output:

Public key (n,e): 391 35

Private key (n,d): 391 171

Exercise 5:

1. **Design a public key infrastructure (PKI) system. Explain the components and their roles in the system.**

   A Public Key Infrastructure will need the following components:

- *Certificate Authority (CA)* - This is the central trusted entity who issues and manages digital certificates.
- *Registration Authority (RA)* - Checks and verifies each user before they get a certificate from the CA.
- *Users* - We generate public/private key pairs and get our identity certified by the CA as digital certificates.
- *Certificate Database* - Public store of all issued certificates and revoked certificates lists for anyone to check if a certificate is still valid.
- *Digital Certificates* - Identity documents issued by CA binding our keys and info. Used for authentication.
- *Certificate Policies* - The rules that define how certificates are issued, validated, trusted and used in the system.

2. **Discuss the advantages and challenges of implementing a PKI system.**

   Advantages

   - Allows secure communication through public key encryption and digital signatures
   - A centralized Certificate Authority provides an efficient certificate management
   - Hierarchical trust model provides transitive trust
   - Revocation mechanisms improve security
   - Scalable to support large organizations and internet-level deployments

   Challenges

   - It is a complex infrastructure
   - Key generation, distribution and storage has to be robust and secure
   - Certificate Authority is a single point of trust but also failure if compromised
   - Computationally intensive

3. **Provide an example of a real-world application where a PKI system is used**.

    - PKI with trusted certificate authorities is used to secure HTTPS websites. Browsers validate website certificates to TLS encrypt traffic.
    - Developers digitally sign software and app code using their private keys and certificates to assure integrity.
    - S/MIME email encryption uses PKI to encrypt and digitally sign messages between users.

Exercise 6:

1. **Design a system for digital signatures based on public-key cryptography. Explain the steps involved in the process and the role of each component.**

    Components:
    - A public/private key pairs for each user
    - The cryptographic hash function SHA-256
    - A public Key Infrastructure (PKI) with Certificate Authority

    Signature generation:
    - The user has a private key and a certificate containing their public key signed by a Certificate Authority.
    - Document/data to be signed is hashed.
    - The hash value is then encrypted with the user's private key to generate the digital signature.
    - The digital signature is bundled with the original data.
    - Data and signature are sent to the verifier.

    Signature verification:
    - Verifier receives the data and signature.
    - Received data is hashed with the same algorithm.
    - The digital signature is decrypted using the user's public key from their certificate.
    - If the decrypted value matches the calculated hash, the signature is valid.