Homework2 - EstebanAlvarado

Exercise 1

Suppose a password is chosen as a concatenation of seven lowercase dictionary words. Each word is selected uniformly at random from a dictionary of 50,000 words. An example password is `"mothercathousefivenextcrossroom"`. How many bits of entropy does it have?

A common way to calculate entropy is using Shannon's formula, which is based on the probability that each symbol, character, or *word* appears in the data. The formula is the following:

$H = -\sum_{i=1}^{n} p_i \times log_2(p_i)$

The formula used to calculate the entropy of a password is similar to Shannon's, but with some differences:

$H = L \times log_2(N)$

Where H is the entropy, L is the length of the password, N is the number of possible characters that can be used to form the password, and log_2 is the base 2 logarithm. The difference between The two formulas is that Shannon's assumes that each symbol or character has a known and fixed probability of appearing, while the password formula assumes that each symbol or character has a uniform and equal probability of appearing. In addition, Shannon's adds the individual entropies of each symbol or character, while password multiplies the length of the password by the logarithm of the number of possible characters.

In this case we will not rely on the characters, but on the words. Therefore, in terms of words, the length of the password is: $L = 7$. Since we have a dictionary of 50,000 words, then $N = 50000$. Plugging this into the last formula, we get:

$H = 7 \times log_2(50000) = 109.26$

Solution: 109.27 bits of entropy

Consider an alternative scheme in which the password is chosen as a sequence of 10 random alphanumeric characters (including upper and lower case letters). An example is `"dA3mG67Rrs"`. How many bits of entropy does it have?

Alphanumeric characters include numbers `0-9` (10) and letters `A-Z` (26). The letters of the alphabet can also be lowercase, so the letters `a-z` (26) in lowercase would be added to this list. That is, the number of possible characters to choose from is $N = 10 + 26 + 26 = 62$. If we know that the length of the password is $L = 10$, then:

$H = L \times log_2(N) = 10 \times log_2(62) = 59.54$

Solution: 59.54 bits of entropy

Which password is better, 1 or 2?

Cryptographic keys with high levels of entropy are more difficult to predict or crack, making the associated cryptographic algorithm more secure. Therefore, password number 1 is the best, as it has the highest entropy.

Exercise 2

Design a data verification system using hash functions. Explain the steps of the process.

- Before data transmission, data will be passed through a secure hash function such as SHA-256; thus obtaining a hash or *digest* value of the data.
- The hash value will be shared with the receiver over a secure communication channel.
- The data is transmitted to the receiver.
- The receiver receives the data, which will pass through the same hash function that the sender used, obtaining another hash value for the received data.
- If the new hash value matches the hash value shared by the sender, then the data has not been modified or altered. If they do not match, then it implies that the integrity of the data was violated during transmission.

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

Advantages	Disadvantages
It is very difficult to discover the original message through hash value	If an attacker manages to access the hash value of the original message, he could try to generate the same hash value but with different data.
It is computationally economical to calculate the hash value of the data	If the hash function is vulnerable to length extension attacks, an attacker could add malicious commands to a valid message and generate a valid hash for the modified message

Provide an example of a real application using a data verification system using hash functions.

Hash functions are widely used in password management. Instead of storing passwords in plain text, systems use hash functions to convert passwords into irreversible values. This means that even if someone accesses the password database, they will not be able to obtain the real passwords. Additionally, when a user attempts to log in, the hash function is used to check whether the entered password matches the stored hash value, without revealing the actual password.

Exercise 3

Define what a message authentication code (MAC) is and how it is used in cryptography.

The *message authentication code (MAC)* is a block of data that is sent along with an encrypted message so that the recipient can confirm the authenticity of the origin of the information. MAC codes are very similar to hash functions, but, unlike them, they have a secret key that guarantees the integrity of the message.

If the MAC value sent matches the value the recipient calculates, the recipient can ensure that:

- The message was not altered
- The message comes from the sender indicated in the message
- If the message includes a sequence number, that the message follows the correct sequence

MACs are typically used for authentication (hence the name). The person to be authenticated and the verifier share the MAC function key and keep it secret. In this way, when the verifier receives the MAC value, it can verify if that MAC value corresponds to the one that must be generated from a given message.

Explain the process of generating and verifying a MAC

MAC values are calculated by applying a cryptographic hash function with secret key *K*, which is known only to the sender and recipient, but not to the attackers. Mathematically, the cryptographic hash function takes two arguments: a key *K* of fixed size and a message *M* of arbitrary length. The result is a fixed length MAC code:

$MAC = C_k(M)$

Where:

- M is a message of arbitrary length
- C_K is the function that transforms the message into a *MAC* value and uses a secret key K as a parameter
- MAC is the calculated MAC value.

To verify if the MAC code is correct, you need to know the secret key that was used to generate the MAC code. The same cryptographic hash function is then applied to the received message and the result is compared to the MAC code sent. If they are the same, it means that the message was not altered and that it came from the authentic sender. If they are different, it means that the message was manipulated or that the sender is not who they say they are.

Exercise 4

Given the values of $p = 17$ and $q = 23$, generate a key pair for RSA.

1. We calculate n using the following equation:

   $n = p \times q = 17 \times 23 = 391$ (9 bits)
2. RSA uses the Euler function ϕ of n to calculate the secret key. This is defined as $\phi(n) = (p-1){\times}(q-1)$. The prerequisite is that p and q are different. Otherwise, the function ϕ would be calculated differently.

   $\phi(n) = (p-1){\times}(q-1) = (17-1) \times (23-1) = 16 \times 22 = 352$
3. The public key is made up of the module n and an exponent e. We select the integer e such that: $1 < e < \phi(n)$ and $gcd(e, \phi(n)) = 1$ (co-prime).

   $e = 3$

   To check that e is co-prime with $\phi(n)$ we will apply Euclid's algorithm to calculate the GCD between e and $\phi(n)$:

   $a = 352$, $b = e = 3$
   $q = a/b = 352/3 = 117 \rightarrow$ quotient
   $r = a - b \times (a/b) = 352 - 3 \times (117) = 1 \rightarrow$ remainder

   Since $r \neq 0$, we have to continue the algorithm until $r = 0$:

   $a = b = 3$, $b = r = 1$
   $q = a/b = 3/1 = 3 \rightarrow$ quotient
   $r = a - b \times (a/b) = 3 - 1 \times (3) = 0 \rightarrow$ remainder

   We obtain $r = 0$, therefore the current b value is the searched GCD value:

   $gcd(3, 352) = b = 1$

   We conclude then that e and $\phi(n)$ are co-prime.
4. The secret key also consists of a d (the multiplicative inverse of $e\ mod(\phi(n))$) with the property that $d \times e = 1\ mod(\phi(n))$. To find d we can use the extended Euclid algorithm, which allows us to find the greatest common divisor of two numbers and also their *Bezout* coefficients, which are the numbers that satisfy the equation: $a \times x + b \times y = gcd(a, b)$. To

find the *Bezout* coefficients, we must substitute the previous remainders into the equations of Euclidean divisions, starting with the last one:

$352 - 3 \times (117) = 1 = gcd(3, 352)$

The Bezout coefficients that satisfy this equation $a \times x + b \times y = 1 = gcd(a, b)$ are:

$352 \times 1 + 3 \times (-117) = 1$

Therefore, $x = 1$ and $y = -117$. The value of d is the coefficient that multiplies e in the equation, which is -117. However, to simplify the calculation and security of the private key, it is usually chosen a value of d that is less than $\phi(n)$ and positive, which is achieved by applying:

$d = d\, mod(\phi(n)) = -117\, mod(352) = 235$

Finally, we have obtained the private and public key:

> Public Key: (n, e) = (391, 3)
> Private Key: (n, d) = (391, 235)

Exercise 5

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

- The main component of a PKI is the *certification authority (CA)*. CA will be in charge of issuing certificates, renewing them, revoking them and managing them. When issuing a certificate, the CA will sign the digital certificates with its private key to ensure its validity and trust. Additionally, you will publish your public key and certificate in a place accessible to users.
- That accessible place is the *certificate repository*. This service allows users to view and download certificates.
- *Registration Authority (RA)* is also needed. RA acts as an intermediary between an entity and CA. When an entity requests a certificate, RA will review the request, and if approved, it is sent to CA. RA will then deliver the certificates issued by the CA to the requesting entity.
- If, for example, it is suspected that the private key associated with a certificate has been compromised or if the entity no longer meets the requirements to have the certificate, the certificate must be revoked. This is where the CA uses the so-called *certificate revocation list (CRL)*, which contains the serial numbers of revoked certificates, in a repository

accessible to users. CRL allows users to check the validity status of digital certificates they receive or use.

Discuss the benefits and challenges of implementing a PKI system.

Advantages	Challenges
It allows you to distribute information safely and reliably.	Implementing this infrastructure can be very expensive
Protects a wide range of communications and transactions	This infrastructure is vulnerable to phishing and "man in the middle" attacks. if not configured correctly

Provide an example of a real application using a PKI system.

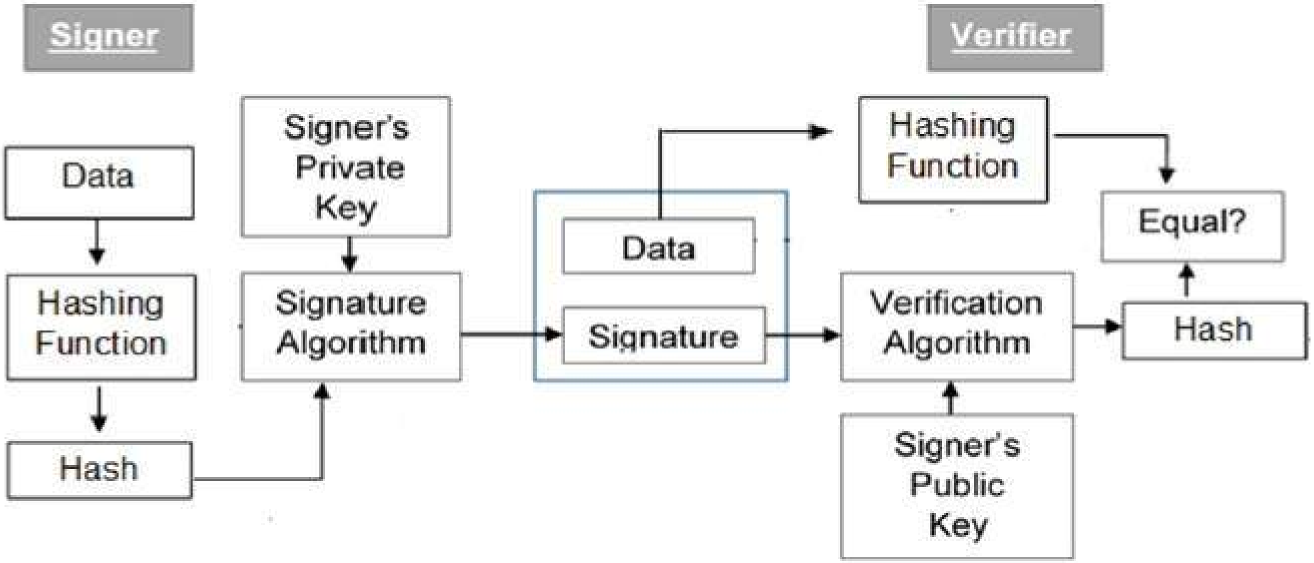
In many companies, email is the Achilles heel of security. As with websites and banking transactions, email data must also be protected while it is transmitted from sender to recipient. PKI certificates are a proven encryption and authentication solution that not only protects the content of the email message, but also confirms the identity of the sender.

The DigiCert® company created a solution for this called DigiCert® Trust Lifecycle Manager, which allows us to control and manage all email certificates from one place. With a single login, you can integrate business email services, authenticate users, remove access to those who no longer work for the company, and recover secure emails. DigiCert® PKI protects against phishing, spearhead phishing, spoofing, and other types of email fraud. Using S/MIME certificates, the PKI encrypts transmitted messages while linking the verified identity of the sender to the message itself.

Exercise 6

Design a digital signature system based on public key cryptography. Explain the stages of the process and the function of each component.

The digital signature scheme is based on public key cryptography. The digital signature scheme model is shown in the following illustration:



- Each person has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verification are different. The private key used for signing is known as the signing key and the public key is known as the verification key.
- The signer feeds data to the hash function and generates data hashes. The hash function is a mathematical algorithm that transforms any data into a fixed and unique value. This value is used to verify the integrity of the message when it reaches the receiver.
- The hash value and signing key are then fed to the signing algorithm which produces the digital signature on a given hash. A signature algorithm is a cryptographic algorithm that uses the sender's private key to generate a digital signature from a given hash. The signature is added to the data and then both are sent to the verifier.
- The verifier enters the digital signature and verification key into the verification algorithm. The verification algorithm is a cryptographic algorithm that uses the sender's public key to check whether a digital signature is valid for a given hash.
- The verifier also runs the same hash function on the received data to generate hash value.
- For verification, this hash value and the output of the verification algorithm are compared. Based on the comparison result, the verifier decides whether the digital signature is valid.
- Since the digital signature is created with the "private" key of the signer and no one else can have this key; the signer cannot refuse to sign the data in the future.