Compression, Encryption and Hashing (1.3.1)

The purpose of Compression is to reduce file size and download times.

When a compressed file arrives at its destination, it needs to be decompressed so it can be read.

With images, audio and video, a small reduction in quality is almost unnoticeable. Therefore, compression is an acceptable compromise between quality and file size.

With text documents and executable programs, we must not lose any of the data during compression. Therefore, for these files, we must use lossless compression.

Lossy compression

- Suitable for images, audio, and video (JPEG, MP3)
- Advantages:
  - Takes out a large amount of data, Greatly increases download/upload speeds
- Disadvantages:
  - Irreversible, Always involves loss of quality

Lossless Compression

- Suitable for executable files and documents (PNG, WAV, FLAC)
- ZIP is very popular (can store any kind of file)
- Advantage: Reversible
- Disadvantage: Less effective at reducing file size than lossy compression

Dictionary Compression (Form of Lossless Compression)

- The algorithm looks for repeated long chains of bytes in the file
- Each byte sequence is placed in a data table within the file and a short codeword allocated to it
- For every occurrence of each sequence (called a symbol), it is replaced by its short codeword
- Advantages
  - Can be used on any file type, but best for text
  - Removes a large amount of data
  - No loss of quality, Reversible
- Disadvantage: Less effective than lossy compression

Run Length Encoding (Form of Lossless Compression)

- Looks for 'runs' of repeated binary patterns
- Replaces them with a single instance of the pattern and the number of times it is repeated
- Can be used on any file type, but best for image files (JPEG, GIF, etc)
- Advantages
  - No loss of quality, Reversible
- Disadvantages
  - Less effective than lossy compression
  - Only works well with long runs of data. If there are no repeated patterns, the file size could double

Encryption: Converting plaintext into ciphertext so that only authorised users can read messages transferred across the Internet. The original message should be impossible to crack without the key.

Symmetric Encryption:

- A single key is used to both encrypt and decrypt a message
- Advantages: Symmetric keys are much stronger than asymmetric keys of the same length, since the asymmetric key is in two smaller pieces. With a larger key, the encryption is harder to break.

- Disadvantages: The key must be kept secret. Thus, key exchange is a problem. With a larger key, the message will also take longer to encrypt and decrypt.

- Asymmetric Encryption
  - Two different keys are used. Much more secure than Symmetric.
  - Anything encrypted with one key can be decrypted with the other. Together, they form a key pair.
  - One is made public (the public key), the other is kept private (the private key)
  - The plaintext is encrypted with the public key. It is decrypted with the private key.
  - Private Communication: Encrypt with the recipient's public key. They will decrypt with their private key.
  - Public Communication: Encrypt with the sender's private key. Anyone can decrypt with their public key.

If I want to send a message, I will use my private key and their public key (a combined encryption key) to encrypt it. To decrypt the message, they will use their private key and my public key.

- Asymmetric Encryption Advantages:
  - Better than Symmetric (Key distribution is much easier)
  - No one else can read my message, you can be sure my message is authentic
  - With a larger key, the encryption is harder to break

- Asymmetric Encryption Disadvantages:
  - With a larger key, the message will also take longer to encrypt and decrypt

Encoding: makes data understandable to everyone and so that it can be properly inputted into another system, by converting data into another format.

Hashing: the process of 'mapping' data of variable length into a fixed-length data value. Uses complicated mathematics. Made to be a one-way process.

Hashing allows searching to be faster and more efficient.

Before transferring data, create a hash of it. Send both the message and the hash. At the destination, create a hash of the received message. Compare the received and calculated hashes. If any data was modified or corrupted during transit, the hashes will be different.

Login systems, instead of storing the password itself, may store the hash of the password. When logging in, a hash is calculated and compared with the stored hash. If they match, then the password is correct. Therefore, only the user knows their password, not the website they are logging into.

Applying a hash function to data items converts a data set into a hash table.

Digital Signatures: Prove to the receiver that the document or data comes from the authorised sender, and that the received information is unaltered

They are used to encrypt data and create a checksum. This checksum is sent with the ciphertext. At the receiving end, a checksum of the data is calculated and compared with the received checksum. If they do not match, then the information has been tampered with.