Component 1 | 1.3.1| Compression, encryption and hashing

Specification & learning objectives

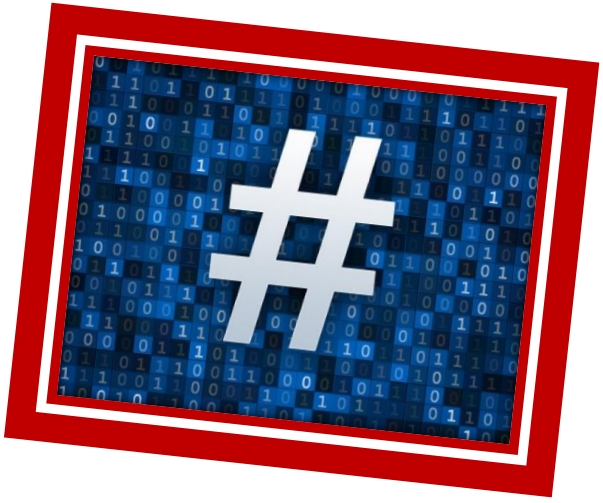
A Level	Specification point description
1.3.1a	Lossy vs Lossless compression
1.3.1b	Run length encoding and dictionary coding for lossless compression
1.3.1c	Symmetric and asymmetric encryption
1.3.1d	Different uses of hashing

Resources

PG Online textbook page ref: 75-81

Hodder textbook page ref: 230-234

CraignDave videos for SLR 9



Key question: What is the difference between lossy and lossless compression?

Compression is any means of repackaging a file in order to reduce the file size. Compression is used all the time in information processing to reduce storage requirements and, more often these days, reduce bandwidth usage & download sizes. There are two main types of compression: lossy & lossless compression.

Lossy compression

- Real data is actually removed from the file in order to reduce the file size.
- An algorithm is used to strip out the least important data.
- Typically this is used for multimedia files where content is still usable if it is missing data e.g. using 480p on YouTube when your internet is slow in order to reduce the bandwidth required.
- Due to the nature of data being removed, the original data cannot be restored.

Lossless compression

- Instead of being removed, data is encoded in such a way that file size is reduced.
- Typically lossless compression is worse at reducing file sizes than lossy compression.
- However, lossless compression must be used rather than lossy in cases such as downloading programs or documents where a single missing bit could cause the whole program to crash.
- Because the data has been encoded rather than removed, the original data can be restored.
- There are two main methods of lossless. compression: dictionary and run length encoding.
- An example of lossless compression is .zip files or .png image files.

Key question: What is the difference between lossy and lossless compression?

Lossy compression	Lossless compression
Removes data to reduce file size	No data loss
Achieves more compression than lossless	Results in less compression than lossy
Results in reduced quality	No loss of quality
Irreversible process, discarded data cannot be recovered	Reversible process, the exact uncompressed data can be recovered

Key question: How does run length encoding work?

RLE identifies repeating patterns and stores one copy of the pattern and how many times it repeats in succession. This can drastically reduce the size of the stored file, especially in text/source with spaces and other formatting parts. For example, if we take the following text document:

We could represent a lot of spaces and certain characters by their runs:

20 Title Run Length Encoding
20 25-
2 <carriage return>

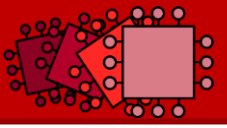
We can see that this reduces the amount of characters needed, and therefore the storage space required.

```
                    Title Run Length Encoding
                    -------------------------


We can see that this reduces the amount of characters needed. So
we can compress the file by a certain amount

Things like tables and formatting shown below can benefit from
RLE.

     Column1     Column2
        A           B
        C           D
```

Things like tables and formatting shown below can benefit from RLE.

5 Column1 4 Column2
8 A 10 B
8 C 10 D

As can be seen we change the run of characters by giving a number and then the character to repeat. The first document consisted of 332 characters, whereas the RLE version contained 264.



Key question: How does dictionary encoding work?

Dictionary-based method uses a type of substring search to match strings in the file to those stored in a data structure called a *library*. If a match is found then the *substitution encoder* substitutes the string with a reference pointer to the item in the dictionary. If there is no match then the string is added to the dictionary and the reference is substituted.

This gives 70 characters including the dictionary; although saving only seven characters may not seem much, in a large document quite significant savings can be made using this technique.

Example:

'Ask not what your country can do for you but what you can do for your country' (77 characters) could be dictionary-compressed to:

```
1 Ask
2 not
3 what you
4 r
5 country
6 can do for you
7 but
8 your
```

Is encoded to:

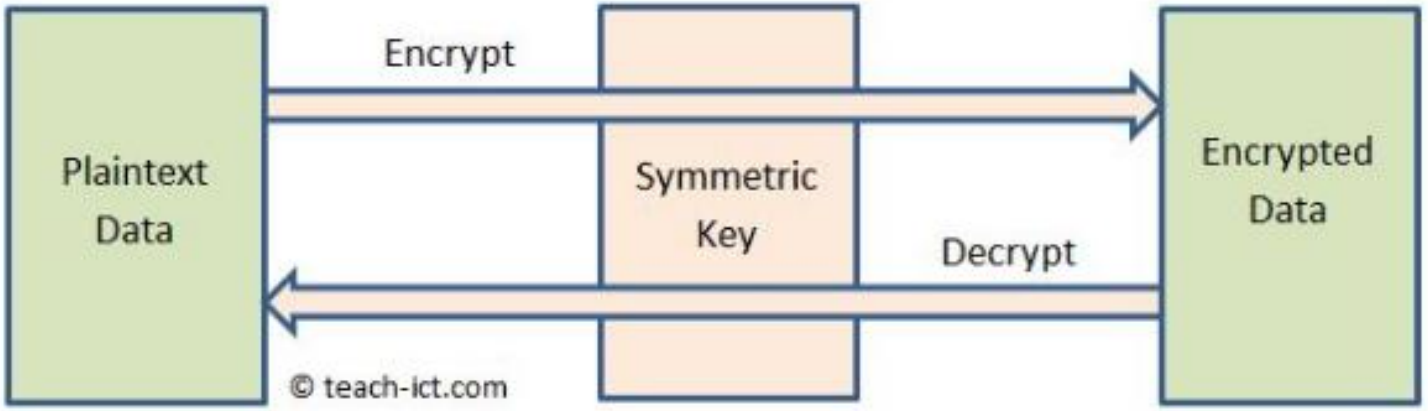
1 2 34 5 6 7 3 64 5

Key question: How does encryption work?

Encryption is the process of scrambling data in such a way that only legitimate users can read it.

Symmetric Encryption

- Symmetric encryption means the same key is used for both encryption and decryption.
- 128 to 256 bit symmetric keys are considered to be strong enough for ordinary use.
- If a symmetric key is made known, then any message using it can be decrypted.

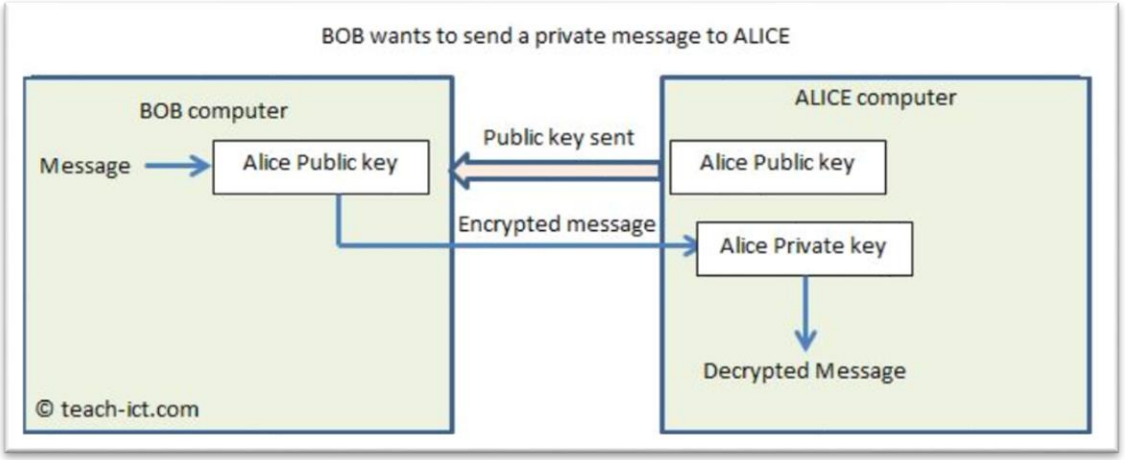


Key question: How does encryption work?

Encryption is the process of scrambling data in such a way that only legitimate users can read it.

Asymmetric Encryption

- Asymmetric encryption means one key (the public one) is used to encrypt a message and another one (the private one) is used to decrypt it.
- AES-256 is an example of a symmetric key algorithm.
- RSA-2048 is an example of an asymmetric key algorithm.



Key question: What is hashing?

Hashing is a technique that allows you to 'map' data of variable length into a fixed-length data value (which is effectively just a large number).

Original Data	Calculated hash
This is the original data which could be a readable string like this or perhaps a data block such as 26262772782882769439474777afaf666aff5s5sfcc2c3c44bb	4b7019286c923e077eb1c78a746b22b0

This number remains the same length, no matter how long the original string was. The number is quite long, so it is usually shown in hexadecimal.

Hashing algorithms use complicated mathematics to convert strings into hashes. You won't need to know the details of how hashing algorithms work, but you will need to know what hashing is useful for.

Typical exam questions

1. Two methods of compression are lossy and lossless. State which method of compression you would use to reduce the file size of a spreadsheet and justify your choice. **[3]**

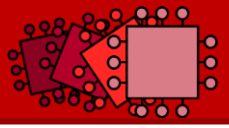
Compression method:

Justification:

2. Explain the similarities and differences between lossy and lossless compression. **[4]**

3. Discuss the use of hashing providing an example of program that would sensibly use hashing and justify your answer. **[6]**

Component 1 | 1.3.1| Compression, encryption and hashing



Target: Overall grade:

Minimum expectations & learning outcomes

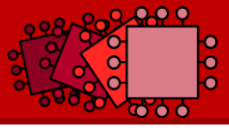
☐	Terms 107-113 from your A Level Key Terminology should be included and formatted.
☐	You must include a clear comparison of lossy vs. lossless compression, explaining situations in which each is not suitable.
☐	You must include illustrations of how dictionary and run-length encoding works.
☐	You must include a clear comparison of symmetric vs. asymmetric encryption.
☐	You must include a section which clearly outlines the different uses of hashing.
☐	Answer the exam questions.

Feedback

Breadth	Depth	Presentation	Understanding
☐ All	☐ Analysed	☐ Excellent	☐ Excellent
☐ Most	☐ Explained	☐ Good	☐ Good
☐ Some	☐ Described	☐ Fair	☐ Fair
☐ Few	☐ Identified	☐ Poor	☐ Poor

Comment & action required

Component 1 | 1.3.1| Compression, encryption and hashing



Reflection & Revision checklist

Confidence	Clarification
☹ 😐 ☺	Candidates need to understand the need for compression, especially when transferring data via the Internet.
☹ 😐 ☺	Candidates need to understand the difference between lossy and lossless compression, and the benefits and drawbacks of each type.
☹ 😐 ☺	Candidates need to be able to recommend a type of compression for a given scenario.
☹ 😐 ☺	Candidates need to understand how run-length encoding can reduce the size of a file for example with a text file or image.
☹ 😐 ☺	Candidates should understand how dictionary coding works by substituting entries with a unique code.
☹ 😐 ☺	Candidates should have practical experience of using these algorithms with small example files.
☹ 😐 ☺	Candidates should understand the need for encryption.
☹ 😐 ☺	Candidates should understand how symmetric and asymmetric encryption work to encrypt and decrypt data.
☹ 😐 ☺	Candidates should understand the need for and purpose of using hashing algorithms to store data.
☹ 😐 ☺	Candidates should be aware of different uses for hashing, such as the storing of passwords.