Cryptography and Secure Development

Matthew Henderson 2031944H

Part 1 – Known Plain Text Attack

The first part of the exercise of was a known plain text attack. The attack involved a brute force search of the entire keyspace. The first block of cipher text was decrypted with every possible key and then compared to the block of known text. If the decrypted text matched the plain text, the corresponding key was taken to be the original encryption key. Since the encryption algorithm used a 16 bit key, there were 65536 possible keys.

Once the correct decryption key was found, it was used to decrypt the remaining cipher text and reveal the plain text message. The decryption key and the deciphered text can be seen below.

```
Decryption key as hex block - 0xfa76 Decryption key as an integer – 64118

Decrypted message - A day can really slip by when you're deliberately avoiding what
you're supposed to do
```

Part 2 – Cipher Text Only Attack

The second part of the exercise was a cipher text only attack. A brute force attack was performed upon the entire keyspace, then the redundancy of the English language was used to calculate which key provided the correct message. The cipher text provided can be seen in appendix 1.

There were two methods used to narrow down the results to those that contained plausible English language messages. Firstly, all results were eliminated that contained ASCII characters that are not commonly used in the English language.

The second technique involved searching the decrypted message for the digrams and trigrams that are most commonly used in the English language [1]. A count was made of each digram and trigram that the string contained. All keys but those that provided the highest count were eliminated. When enough blocks of cipher text were decrypted, the results were narrowed down to one possible message. This message was the original plain text document.

The theoretical number of cipher text blocks needed for the message to be unambiguously decoded can be calculated with information theory. Firstly, the redundancy of the keyspace must be calculated as shown below in equation 1, where D is the redundancy, R is the absolute rate of the language and r is the actual rate of the language. Since ASCII characters use one byte per letter, R = 8. Futhermore the actual rate of the English language is known to be 1.5. This gives a redundancy of 6.5.

$$D = R - r \tag{1}$$

The unicity distance can then be calculated with equation 2 below where H(K) is the entropy. Since the encryption algorithm uses a key length of 16 bits, and the key is presumed to be randomly chosen, the entropy is 16.

$$N_u = \frac{H(K)}{D} \quad (2)$$

This gives a unicity distance of 2.5. We can assume that the unicity distance is the minimum number of characters to unambiguously break the code. A text block converts to 2 characters, hence the number of blocks that theoretically needs to be decoded is 2.

Upon testing, the number of cipher blocks required to unambiguously decode the message was found to be 10. This testing was done through an iterative process. The programme was run multiple times, each time the number of blocks deciphered was increased. The point when there was only one message with the largest count of common digrams/trigrams was determined to be that which the message was unambiguously decoded.

There are a few possible reasons for this number being higher than the theoretical number of blocks needed. Firstly, although the exact encryption algorithm is unknown, it is presumed to have low confusion. This assumption was made by comparing the real message with the message that recorded the second highest trigram/digram count. Both messages can be seen below. As can be seen they are very similar. This low confusion causes several messages to be generated that are very similar to English language messages, hence skewing results.

```
For perfect safety... sit on a fence and watch the birds.
foR PeRfEcT SaFeTy. SiT OnafEnCeaNdwAtChtHebIrDs
```

Another possible reason for the difference between the theoretical value and the actual value may be the method used to identify the English language message. If a more refined algorithm was used that perhaps assigned weighting to particular digrams/trigrams based on their statistical prevalence, or evaluated language structure such as full stops and capitalisation, then the correct message could be found using a smaller number of blocks.

The correctly decoded message along with the decryption key used to find it can be seen below.

```
Decryption key as hex block - 0x42bb Decryption key as an integer - 17083

Decrypted message - For perfect safety... sit on a fence and watch the birds.
```

Part 3 – Time Memory Trade Off Attack

The final part of the exercise was a time memory trade off attack. A time memory trade off attack is similar to a table look up attack but involves building numerous long chains of encrypted text that can be traversed to find the key. This reduces the size of the table, but it also increases the lookup time. Since this attack is based on a table it is a known plain text attack.

As part of the exercise, the block of plain text was received in advance of the cipher text. A time memory trade of table was then constructed from this text. For this table a chain length of 256 was used and a chain number of 256, this gives the right number of potential keys. However, since this is a probabilistic attack there may be some duplicates keys, and some incomplete coverage. This meant there was a possibility the table would have to be remade once the cipher text arrived.

The java Random class was used to generate a random number for the start of each chain. The first member of the chain and the last member were then written to a text file for storage. Once the cipher text arrived, these values were read from the file and put in a HashMap. The HashMap was then

searched, and the chains were traversed until a match with the known text was found. From this match the key was derived, and the entire message decoded. Both the key and the message can be seen below.

```
Decryption key as hex block - 0x8aff Decryption key as an integer – 35583

Message - Selective Gravity Law: An object will fall so as to do the most damage
```

References

[1] http:/academic.regis.edu/jseibert/Crypto/Frequency.pdf