T8: Multiple Anagramming and Unicity Distance

Crypto 2006

2 November 2006

Work this problem by yourself. Give yourself maybe an hour, don't let it ruin your week. If you use ANY kind of outside material (outside your head, that is), please reference it explicitly. There may well be more than one "right" answer consistent with the assumptions. Concentrate on constructing a tight, correct argument rather than on getting "the answer". Hand in the hardcopy as usual to Bin. A pretty, word-processed document would be appropriate for this work.

1 Problem and Questions

What good are confusion, diffusion, and key-scheduling anyway? If large chunks of DES and AES boil down to look-up tables and hard-wired shifts, etc., why have them? The only secret part is the key, after all; who cares if it's manipulated if everyone knows just how?

This problem is to investigate the complexity of finding the key to a block cipher that simply XORs the key with English text.

Assumptions: 100-bit key, message is N 100-bit blocks, characters encoded in 5 bits (32 values, let's assume they are all valid plaintext characters for simplicity – maybe we add 6 punctuation marks to the 26 letters). Each message block is XORed with the key. Assume that 1/2 the possible tetragrams (strings of four characters) actually appear in English text (e.g. "T, P" is OK, as is "GLOP", but "XA;Q" is not.) [1] Assume that there is a negligible probability that more than one key will decipher the cryptotext (even for a tetragram) to some intelligible plaintext (see below).

Below, a "bad result" is a non-English 'decryption' of a tetragram.

Question 1: Appealing to all these assumptions, what is the minimum number N of independent blocks you need to find the key? (clearly, with 1 block we have a true one-time pad but with more than one we are committing the no-no of re-using the pad.)

Question 2: How many XOR operations do you need to find the key?

Question 3: How do your answers change if the key and blocks are 1000 bits long?

Question 4: How do your answers change in the light of Michalak's result (*cf.* footnote 1)?

Question 5: There is a possibility of more than one key giving a meaningful decryption for tetragrams. The concept here is called *unicity distance*, and you can read about it in Trappe and Washington Section 15.5 or Schneier's book (Section 11.1).

The *rate* of information transmission for uniformly-distributed letters (A – Z) is $R = \log_2(26) = 4.7$. Due to *redundancy* (leading to predictability of, say, the next letter), the actual rate of English

[1] Probably very conservative. Phil Michalak computed the number of 4-grams that appeared in a text of 51115 characters and found 4772 of 614656 possible 4-grams in an alphabet of size 28 (letters, space, apostrophe). Thus a fraction .00776 of the possible tetragrams actually appeared [Personal Communication].

is about 1.0 - 1.5 bits/letter (b/l). For 8-letter chunks of English, the rate is in between at 2.3 b/l. [2]. What would you guess the entropy of English tetragrams is?

The *entropy* $H(K)$ of a cryptosystem increases with the size of its keyspace. It is the logarithm (base 2 of course) of the number of keys, and is the number of bits of information we need to represent the key, or how many bits we need to learn to cryptanalyze the system. What is the entropy of our system here, which involves XOR-ing with a 20-bit random key?

Now the *redundancy* of a message is $D = R - r$. That is, it is the bits in a letter that are not transmitting information *about the message contents*. So English is very r*d*ndt. But for us those bits are carrying information about something else, which is the key. In this sense the more redundant the message is, the more material the codebreaker has to work with. It is his job to use redundancy (say statistical dependence) to break the code (consider the IOC, for example). Thus those D bits are information about the key — our challenge is to use it. But speaking purely theoretically, if cryptanalysis were not a problem, we would expect to need at least $U = H(K)/D$ letters to accumulate enough bits of information to determine the key. That is why U is called the *unicity distance* and why we say that if you have deciphered U letters and the result makes sense then it is unlikely that you have the wrong key, or that another key would yield acceptable English output.

Compute U for our system and relate your findings to your previous answers.

[2] Schneier, *Applied Cryptography*, p. 232 ff.