KTH Computer Science
and Communication

Homework I, Foundations of Cryptography 2008

Due on Wednesday Feb 13 at 10.15. The general rules on homework solutions available on the course home-page apply. In particular, discussions of ideas in groups of up to at most three people are allowed but implementation should be done individually. Note that there are two types of problems (T(heory) and P(ractice)). Bonus points are counted as usual points but the corresponding problems might be more challenging. How points translate into grades is described in the course memo.

1 (20P) Solve the cryptogram "unknown1" available on the course home-page. The clear text is in English with character "space". Note that space is handled exactly as any other character and for instance two adjacent spaces in the cipher-text is very much different from one space.

2 (14T) Many cryptosystems like AES use a finite field of the type $GF(2^k)$ for some k. The situation is described in the book of Stinson for $k = 3$ and the irreducible polynomial $x^3 + x + 1$. Call this field F. Your task is to investigate what happens if we instead use the irreducible polynomial $t^3 + t^2 + 1$ getting a field K. The following calculations should be done by hand.

2a (4T) Compute the multiplication table of K.

2b (4T) Solve the system of equations

$$\begin{array}{rclclcl} 1+t^2 & = & (1+t)y_1 & + & t^2 y_2 & + & y_3 \\ t & = & (1+t^2)y_1 & + & t y_2 & + & (1+t)y_3 \\ 0 & = & y_1 & + & (1+t)y_2 & + & (t+t^2)y_3 \end{array}$$

2c (6T) One usually speak of *the* field with 8 elements and the reason is that F and K are isomorphic, i.e. that they only differ in the way they name the elements. Find an isomorphism! In other words for each element $\alpha \in F$ find $f(\alpha) \in K$ such that it is always true that $f(\alpha_1 + \alpha_2) = f(\alpha_1) + f(\alpha_2)$ and $f(\alpha_1 \cdot \alpha_2) = f(\alpha_1) \cdot f(\alpha_2)$.

2d (10T, bonus) Find all such isomorphisms! Can you explain the number of such isomorphisms?

3 (5T+15P) Construct your own RSA system and analyze the performance of the algorithm.

3a (15P) Implement a key generator, encryption and decryption. Your algorithm should support a variable key length and make sure that both e and d are at least $N/100$.

Encrypt the message "Send more money" with the public key and show also decryption.

Run your algorithm for key lengths 512, 1024 and 2048 recording running times.

3b (5T) Make a detailed complexity analysis of, finding N, calculating d and e and encryption/decryption. You should even calculate a rough guess of the constant for the highest order term. Compare the theoretical result with the practical times observed using the speed of your computer. Comment on the result.

In this problem you are allowed to use pre-made routines for arithmetic (for instance for addition, multiplication and division with remainder) for large integers, but not more powerful operations such as modular exponentiation and primality tests). If in doubt whether a pre-made routine is allowed please ask Johan.

4 (20P) In the file "gskriv" on the course home page there is the input and output of the "Geheim-schreiber". For details of the machine we refer to the one-page description available on the course home page. Reconstruct the wheels! We also have the following two possible bonus problems each worth 20P individually but only add up to 30P jointly.

4a The file "gskriv2" contains an encryption observed later the same day on the same communication line. Reconstruct the clear-text of this messages.

4b The file "gskrivmod" contains a clear text and crypto text of a slightly modified "Geheim-schreiber". Reconstruct the workings of this machine and the keys used for encryption.

5 (12T) Suppose you have a sequence of n bits which are independent and each is 0 with probability .99.

5a (4T) Use the formula for entropy to compute the entropy of this n-bit string.

5b (4T) Find an efficient way to encode this source using as few bits as possible. Analyze your scheme and compare to the answer to the first problem. If there is a big difference comment on whether this is needed.

5c (4T) Suppose instead the bits are chosen as follows. Let x_1 be 0 with probability .99 and for $i > 1$ we set $x_i = x_{i-1}$ with probability .99 and otherwise we set it to the complement. What is the entropy of this source and how do you compress it?

6 (20T) Your task is to theoretically describe and analyze attacks on AES when the number of rounds is very small. Consider three different attack models. In the first you get pairs of (random) clear text and cipher text blocks. In the second you are allowed to choose the clear text blocks and get the corresponding cipher text blocks. In the third you can also ask for the decryption of cipher text blocks of your choice. This is a variant of the so called "lunch-time attack" where an evil person temporarily gets control of a decryption device and wants to make permanent damage.

To be more precise: You should describe (but not implement) and analyze attacks against 1, 2 and 3 round AES. You should estimate the number of blocks needed in each case and try to make this number as low as possible. Each of the three cases is worth 10T but the case of 3-rounds is more difficult and is seen as a bonus problem.

This problem should be solved without the aid of published sources. If you by mistake find detailed information from somewhere, please include the source. Note that no implementation is needed.