KTH Computer Science and Communication

Homework I, Foundations of Cryptography 2009

Due on Monday Feb 16 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 implementations should be done individually. 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 using only capital letters and the 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 As discussed in class the finite field $GF(2^k)$ is useful in many cryptographic situations. Let us first study the situation for $k = 3$ by hand. Let the field be defined by the irreducible polynomial $t^3 + t^2 + 1$.

2a (4p) Compute the multiplication table of this field.

2b (4p) Solve the system of equations

$$\begin{aligned} 1 + t^2 &= (1+t)y_1 + t^2 y_2 + y_3 \\ t^2 &= (1+t^2)y_1 + t y_2 + (1+t)y_3 \\ 1 &= y_1 + (1+t)y_2 + (t+t^2)y_3. \end{aligned}$$

2c (10p) Your boss asks you for a short description, to be handed to a colleague of yours, on how to find a representation of $GF(2^{1000})$. She wants it to be defined by a polynomial which no human has previously seen and hence to be defined by a (pseudo)random polynomial of degree 1000. She trusts your favorite compiler but not any algebra system. She asks you to make a theoretical description on how to find such a polynomial and to estimate how much it will cost in the form of CPU time.

2d (10p) Hopefully satisfied with your solution to the previous problem the boss comes back and asks you to do the implementation. Please do this! If the outcome does not agree with the theoretical discussion please explain why.

Your code should be handed in as an attachment to an email to dog@csc.kth.se with subject line "krypto09-irr". The attachment should consist of a single file named ''`firstname_lastname.tar.gz`'', such that the command ''`tar xvfz firstname_lastname.tar.gz`'' produces a single directory named ''`firstname_lastname`'' containing your files. If you write your solution in C/C++, the directory must contain a Makefile such that `make` builds your program. Regardless of the programming language used, your directory must contain a file `README` containing a brief description of how your program can be invoked, including a few concrete examples that execute in a reasonable amount of time. Your code should obviously be structured and documented.

Remember that you are only allowed to use standard functions available in a standard programming language such as Java or C/C++. Of course you are allowed to look for a mathematical description of an efficient algorithm to test irreducibility anywhere.

3 (20p) Make an efficient implementation of AES. Your implementation should be submitted to the tool "Kattis" for automatic evaluation. Details about the input and output format that should be used is available at `http://kattis.csc.kth.se/problem?id=aes`.

If you have not used Kattis before, you need to have registered for the course using the `res checkin krypto09` command in order to get an account (which might take a while and in this case you probably also want to have a look at the documentation about how to use Kattis, available from the URL above).

To get any points on this problem your implementation has to be accepted by Kattis. Provided this is true and your implementation runs in T seconds your score is $(150/\max(T, 6)) - 5$ rounded up to the closest integer. Note that $T \leq 6$ gives a full score while $T = 30$ gives score 0. Also, in order to get your points, you need to state the Kattis Submission ID of the code that you want to get points for in your handed in solutions.

You may use implementation ideas from the book or from other sources, but if you do then you should include references when you hand in your solution. You are not allowed to simply download, copy by hand, or otherwise use somebody else's implementation.

Bonus of 20 and 10 points will be awarded to the two fastest implementations. Only those programs that were turned in on time will be considered for this bonus.

4 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.

4a (20p) Reconstruct the wheels!

4b (20p, bonus) The file "gskriv2" contains an encryption observed later the same day on the same communication line. Reconstruct the clear-text of this messages.

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

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

5b (4p) 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.

6 A common wisdom in encryption is that the longer the key the safer the cryptosystem. This is true sometimes but not always. Consider a cryptosystem that encrypts n message bytes $(M_i)_{i=1}^n$ using $2n$ key bytes $(X_i)_{i=1}^n$ $(Y_i)_{i=1}^n$ and defining the i'th crypto byte by first computing

$$F_i \equiv (X_i + 1)(Y_i + 1) - 1 \bmod 257$$

and then setting $C_i = M_i + F_i \bmod 256$.

6a (1p) Show how to decrypt this cryptosystem.

6b (5p) Is this a secure cryptosystem? We suppose that for each new message we use fresh crypto bytes X_i and Y_i which are picked uniformly at random. You should supply a formal proof of security or describe some nontrivial attack.

6c (5p) Does the answer to the previous question change if we are lazy and reuse the key bytes Y_i? In other words, suppose for each new message encrypted a fresh set of uniformly independent bytes X_i are picked but we use the same value of Y_i as for any other message. Is the system secure?