MTAT.07.003 Cryptology II
Spring 2008 / Homework 1

1. Let $\mathbb{G}$ be a finte group such that all elements $y \in \mathbb{G}$ can be expressed as powers of $g \in \mathbb{G}$. Then the discrete logarithm problem is following. Given $y \in \mathbb{G}$, find a smallest integer x such that $g^x = y$ in finite group $\mathbb{G}$. Discrete logarithm problem is known to be hard in general, i.e., all universal algorithms for computing logarithm run in time $\Omega(\sqrt{|\mathbb{G}|})$.

   (a) Show that for a fixed group $\mathbb{G}$, there exists a Turing machine that finds the discrete logarithm for every $y \in \mathbb{G}$ in $\mathrm{O}(\log_2 |\mathbb{G}|)$ steps.

   (b) Show that for a fixed group $\mathbb{G}$, there exists an analogous Random Access Machine that achieves the same efficiency.

   (c) Generalise the previous construction and show that for every fixed function $f : \{0,1\}^n \to \{0,1\}^m$ there exists a Turing machine and a Random Access Machine such that they compute $f(x)$ for every input $x \in \{0,1\}^n$ in $\mathrm{O}(n+m)$ steps.

   (d) Are these constructions also valid in practise? Explain why these inconsistencies disappear when we formalise algorithms through universal computing devices.

   Hint: What is the time-complexity of binary search algorithms?

2. Consider a classical Turing machine without internal working tapes, i.e., the Turning machine has a single one-sided (input) tape that initially contains inputs and must contain the desired output after the execution.

   (a) Show that all sorting algorithms take at least $\Omega(n^2)$ steps where n is the total length of inputs $x_1, \ldots, x_k$. What is the time-complexity of best sorting algorithms? Explain this contradiction.

   (b) Does the minimal time-complexity change if the Turing machine has internal working tapes?

   (c) Sketch how one can simulate execution of Random Access Machines on a Turing machine. What is the corresponding overhead?

   ($\star$) Construct a set of tasks that can be implemented significantly more efficiently on Turing machines with $\ell+1$ working tapes than on Turing machines with ℓ tapes.

   Hint: It is well-known fact that reversing n-bit string takes $\Omega(n^2)$ steps on a Turing machine without working tapes.

3. Bob has a biased coin such that in each throw the probability of getting a tail is α. Additionally, assume that all coin tosses are independent.

   (a) How many throws are needed on average to see the first tail?

   (b) How many throws are needed on average to see k tails?

Now consider a scenario, where Bob must see at least two tails to succeed.

(c) How many throws are needed to succeed with probability at least $\frac{1}{2}$? Give a simple and safe upper bound on the number of throws.

(d) Show that Bob must make at least $\Omega(\frac{1}{\alpha})$ throws to achieve constant success probability in the process $\alpha \to 0$.

(e) How many throws are needed to achieve exponentially small failure probability ε?

Hints: Use Markov's and Chebyshev's inequalities. Answers of the questions (c) and (e) are tightly connected.

4. A cryptosystem is a triple of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ such that the equality $\mathcal{D}(\mathcal{E}(m,k),k) = m$ holds for all messages $m \in \mathcal{M}$ and keys $k \leftarrow \mathcal{K}$. Cryptosystem is perfectly secure if a ciphertext c reveals nothing about the corresponding message m, i.e., $\Pr[m|c] = \Pr[m]$.

   (a) Prove that cryptosystem is perfectly secure only if $H(m|c) = H(m)$. What about the implication to the other direction?

   (b) Show that $H(k,m,c) \geq H(m|c) + H(c)$. For which enciphering algorithms does the equality $H(k,m,c) = H(m|c) + H(c)$ hold?

   (c) Show that $H(k,m,c) = H(k) + H(c|k)$. Conclude that cryptosystem is perfectly secure only if $H(k) \geq H(m)$.

   (d) Show that $H(k|c) = H(m) + H(k) + H(c|m,k) - H(c)$. What does the result mean in practise?

5. Estimate how much time is needed to break the following three file encryption methods without using cipher-specific attacks.

   (a) The file is encrypted with 128-bit AES cipher and the key is stored in a special file that is protected with a password. Namely, the key is encrypted with another key that is derived form the password.

   (b) The file is encrypted with old 56-bit DES cipher and the key is stored in the special file that is encrypted with a public key. The corresponding secret key is stored in the ID card.

   (c) The file is encrypted with 80-bit IDEA cipher and the key is stored in the special file that is encrypted with a public key. The corresponding secret key is stored in the TPM chip.

6. Let $\mathcal{X}_0$ be a uniform distribution over $\mathbb{Z}_{16}$ and let $\mathcal{X}_1$ be a uniform distribution over $\{0, 2, 4, 6, 8, 10, 12, 14\}$.

   (a) What is the statistical difference between $\mathcal{X}_0$ and $\mathcal{X}_1$?

   (b) What is the best distinguishing strategy if we can only compare the sample x with other values up to t times? Consider the same dependency for the AND and GT predicates.