MTAT.07.003 Cryptology II
Spring 2012 / Exercise session ?? / Example Solution

Exercise (Randomised self-reducibility of the most significant bit of DL). *It is not known whether the computation of the most significant bit of discrete logarithm (MSB-DL) can be efficiently reduced to a random instance of MSB-DL. Study the consequences of such random self-reducibility by finding a lowest success probability that can be still amplified up through the majority voting.*

Solution. Let $\mathcal{A}$ be an adversary that outputs the guessed significant bit x for an input $y \in \mathbb{G}$. Under the assumption that MSB-DL is efficiently self-reducible, we can assume that the success probability of $\mathcal{A}(y)$ is $\frac{1}{2} + \varepsilon$ for each input, while the running time t is still feasible. Therefore, we can run the algorithm multiple times and use majority voting to calculate the most correct answer. Let $\mathcal{B}$ be the corresponding algorithm:

$\mathcal{B}(y)$
$\lceil$ For $i \in \{1, \ldots, n\}$ do
$\lceil\, x_i \leftarrow \mathcal{A}(y)$
$c \leftarrow x_1 + \cdots + x_n$
return $[2x > n]$

where n is a fixed constant for repetitions. Although the higher number of repetition increases the success probability, it also increases the running time. Hence, we must estimate, which are the consequences of various trade-offs and which of them are in the realm of feasibility. Now let us recall the Hoeffding's inequality allows to estimate the size of the lower tail of a binomial distribution:

$$\Pr[x_1 + \cdots + x_n \leq (p - \delta)n] \leq \exp\left(-2\delta^2 n\right),$$

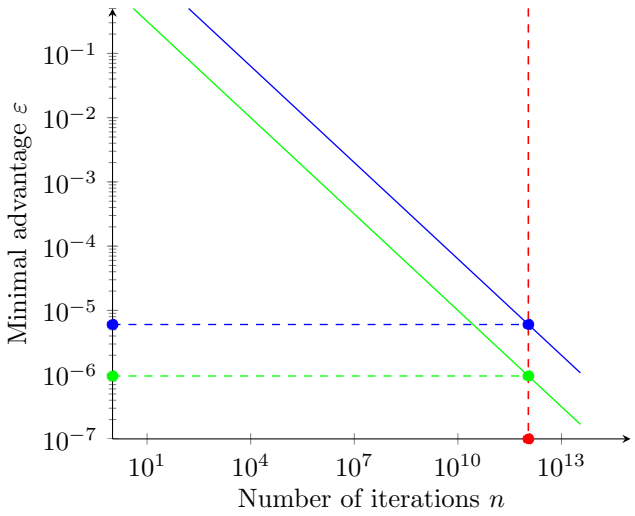
where p is the probability of obtaining one, i.e., $p = \Pr[x_i = 1]$. For the majority voting algorithm $p = \frac{1}{2} + \varepsilon$ and we must take $p - \delta = \frac{1}{2}$ to estimate failure probability. This yields

$$\Pr[\mathsf{Failure}] = \Pr[2(x_1 + \cdots + x_n) \leq n] \leq \exp\left(-2\varepsilon^2 n\right) \quad .$$

The bound shows clearly that the probability of failure ρ decreases exponentially in the number of trials. However, in the context of this exercise, we are more interested how what is the lowest success probability we can amplify to the level $1 - \rho$ and how it depends on the number of repetitions. Hence, we must solve the inequality $\exp\left(-2\varepsilon^2 n\right) \leq \rho$ with respect to ε. By taking logarithm form both sides, we obtain a lower bound for ε:

$$-2\varepsilon^2 n \leq \ln \rho \quad \Leftrightarrow \quad \varepsilon \geq \sqrt{\frac{\ln(1/\rho)}{2n}}$$

To understand how this bound works in real life, consider a setting where the running time of $\mathcal{A}$ is 1 second on a modern computer and we are willing to use 50,000 computers throughout the year to compute the most significant bit y. Then the number of repetitions at our disposal is roughly 2^{40}. The tradeoff graphs for two values of failure probability $\rho = 2^{-2}$ and $\rho = 2^{-80}$ are given below



where the lower line corresponds to $\rho = 2^{-2}$ and the higher line corresponds to $\rho = 2^{-80}$. Note that even if we are willing to use significant amount of computational power, the minimal advantage is only slightly below 10^{-6}. In another words the amplification is not very effective. Indeed, note that the lower bound of the advantage $\Theta(1/\sqrt{n})$ and thus slowdown by a factor 100 decreases the lower bound by a factor of 10.

Note that for any q element group $\mathbb{G}$ achieving advantage of order $\frac{1}{q}$ is trivial. However, we cannot use majority voting to amplify it up to a reasonable level, since the slowdown factor would be of order q^2. The latter is clearly unpractical, as we can find the correct answer in $\Theta(q)$ by brute force. Thus, random self-reducibility does of MSB-DL does not imply that MSB-DL is computationally simple. Consequently, there might exists groups where MSB-DL is random self-reducible while the problem itself is hard.