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

Exercise (Random self-reducibility and time-success profile). *Many security assumptions can be viewed as inversion tasks. More precisely, there exists an efficiently computable function $f : \mathcal{X} \to \mathcal{Y}$ such that a security assumption $\mathfrak{P}$ is satisfied when for any t-time algorithm $\mathcal{A}$:*

$$\mathsf{Adv}^{\mathfrak{P}}_{\mathcal{X}}(\mathcal{A}) = \Pr\left[x \xleftarrow{u} \mathcal{X} : \mathcal{A}(f(x)) = x\right] \leq \varepsilon \ .$$

Study what implications does random self-reducibility add to the time success profile $\varepsilon(t)$.

Solution. Recall that time-success profile $\varepsilon(t)$ corresponds to the maximal success for each time bound, which can be further compressed into a single number by considering the best time-success ratio over the entire time-success profile:

$$\alpha = \min_t \frac{t}{\varepsilon(t)} \ .$$

Intuitively, the time-success ratio shows the minimal time that is needed to reverse the function with complete certainty, as a single run will succeed with probability $\varepsilon(t)$. Consequently, we need to repeat the procedure at least $\frac{1}{\varepsilon(t)}$-times to cover the entire probability space. The latter is only an estimate, since the second run of an algorithm does not have to succeed when the first run fails. If the problem is randomly self-reducible then the outcomes of individual runs are uncorrelated and we can formally justify the intuition.

Near-linear behaviour. Random self-reducibility in our context means that inverting any instance of $f(x)$ can be reduced to inversion of a random instance in $\mathcal{Y}$. More formally, there exist efficiently computable re-randomisation function

$$h : \mathcal{Y} \times \mathcal{R} \to \mathcal{Y}$$

and its inverse for the solution recovery

$$g : \mathcal{X} \times \mathcal{R} \to \mathcal{X}$$

such that following conditions are met. First, for any $f(x) \in \mathcal{Y}$ the outcome $h(f(x), r)$ has uniform distribution over $\mathcal{Y}$ whenever r is chosen uniformly from $\mathcal{R}$. Second, a solution x_* for the randomised instance $h(f(x), z)$ can be converted back to a solution for the original problem:

$$\forall r \in \mathcal{R} : \forall x_* \in \mathcal{X} : \quad f(x_*) = h(f(x), r) \Longrightarrow f(g(x_*, r)) = f(x) \ . \tag{1}$$

If these functions exist, we can convert t-time adversary $\mathcal{A}$ to an adversary with time complexity $\Theta(\ell t)$

$\mathcal{B}(y)$
- For $i \in \{1, \ldots, \ell\}$ do
  - $r \leftarrow \mathcal{R}$
  - $x_* \leftarrow \mathcal{A}(h(y, r))$
  - $x \leftarrow g(x_*, r)$
  - if $f(x) = y$ then **return** x
- **return** $\perp$.

More precisely, the running time of $\mathcal{B}$ is $\ell(t_h + t + t_g + t_f)$ where t_f, t_g, t_h are running times for evaluating functions f, g, h. Equation (1) guarantees that $\mathcal{B}$ succeeds whenever $\mathcal{A}$ succeeds. The first property guarantees that each call to $\mathcal{A}$ is an independent and random problem instance. Thus, the probability of a failure is

$$\Pr\left[\mathcal{B}(y) = \perp\right] = (1 - \varepsilon)^{\ell} \ .$$

The inequalities stemming from exclusion-inclusion principle assure that

$$1 - \ell\varepsilon \leq (1-\varepsilon)^\ell \leq 1 - \ell\varepsilon + \frac{\ell(\ell-1)}{2}\varepsilon^2$$

and thus we have bounds for the success probability:

$$\ell\varepsilon\left(1 - \frac{\varepsilon(\ell-1)}{2}\right) \leq \Pr\left[x \leftarrow \mathcal{B}(y) : f(x) = y\right] \leq \ell\varepsilon \ .$$

Note that as long as $\ell \ll \frac{1}{\varepsilon}$ the lower bound is negligibly close to the upper bound and the success probability grows indeed linearly for all practical purposes.

Regime shift to exponential saturation. Let us now observe the behaviour when ℓ is around $\frac{1}{\varepsilon}$ and thus the approximation outlined above is imprecise. Bernoulli's inequality implies that for any $\varepsilon \in (0, 1]$

$$(1-\varepsilon)^{1/\varepsilon} \leq \frac{1}{e}$$

and thus the success probability converges exponentially fast to one:

$$\Pr\left[x \leftarrow \mathcal{B}(y) : f(x) = y\right] = 1 - (1-\varepsilon)^\ell \geq 1 - e^{-\varepsilon\ell} \ .$$

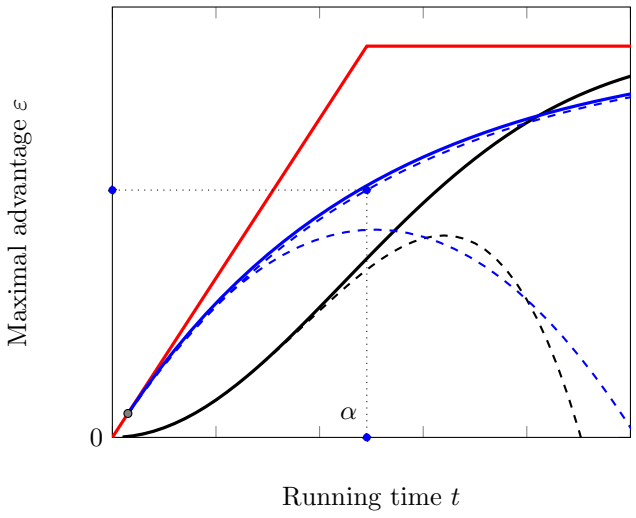
Connection to maximal time-success ratio. Let $\mathcal{A}$ be t-time algorithm that achieves the best time-success ratio α. Then we can construct $(\alpha + t)$-time algorithm by doing $\ell = \lceil 1/\varepsilon \rceil$ repetitions. Consequently, the success of the algorithm is at least $1 - 1/\varepsilon \geq 0.63$ and thus the naive hope that there exist α-time algorithm that completely solves the problem is not so far off.

Generic square root algorithm. Note that by tabulating ℓ values of $f(x)$ we can succeed with probability $\frac{\ell}{|\mathcal{X}|}$ for a random inversion instance. By doing ℓ trials we can amplify this probability to

$$\Pr\left[x \leftarrow \mathcal{B}(y) : f(x) = y\right] \geq \frac{\ell^2}{|\mathcal{X}|} \cdot \left(1 - \frac{\ell(\ell-1)}{2\,|\mathcal{X}|}\right) \ ,$$

which is a near-quadratic function for $\ell \leq \sqrt{|\mathcal{X}|}$. A naive implementation of this algorithm, which tabulates f each time, also runs in quadratic time. However, if we cache the tabulation results, the running-time becomes linear in ℓ and thus we get a quadratic lower bound to time-success profile $\varepsilon(t)$.

Restrictions to the time-success profile. The following figure shows sharpness and relations between bounds and fixes a region where the time-success profile must be located.



The black line corresponds to the generic square root algorithm. Blue line corresponds to the amplification through random-self reduction of the algorithm with best time-success ratio α. Red line corresponds to the upper bound placed on the α value. Dashed lines correspond to lower bounds that are analytically more tractable. Note that all bounds are quite tight. The profile itself can be divided into three regions. In the first region preceding the best time-success tradeoff, the success can drop faster than linearly but not faster than quadratically, as otherwise we hit the generic square root algorithm. In the second region, the time-success profile grows near linearly, as it is confined by two near-linear bounds. In the third region where $t \geq \alpha$ the time-success profile approaches one with exponential speed. Note that again the genetic square root algorithm might be better than the amplification. However, this does not effect the nature of exponential convergence. Obviously, we can find the inverse by looking through all possible values of $\mathcal{X}$ but this takes at least $\Theta(\alpha^2)$ steps and thus cuts of the graph very far from interesting values of t.