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

Exercise (Tradeoffs between false positives and false negatives). *Let $\mathcal{A}$ be a t-time distinguisher and let $\alpha(\mathcal{A}) = \Pr[\mathcal{A}(x) = 0|\mathcal{H}_1]$ be the ratios of false negatives and $\beta(\mathcal{A}) = \Pr[\mathcal{A}(x) = 1|\mathcal{H}_0]$ be the ratio of false positives. Show that for any c there exists a $t + O(1)$-time adversary $\mathcal{B}$ such that*

$$\alpha(\mathcal{B}) = c \cdot \alpha(\mathcal{A}) \quad \textit{and} \quad \beta(\mathcal{B}) = (1 - c) + c \cdot \beta(\mathcal{A}).$$

Give some real world examples where such trade-offs are economically justified.

Solution. Let $\mathsf{CoinToss}(c)$ denote the binary distribution created by tossing a c-biased coin, i.e.,

$$\Pr[b \leftarrow \mathsf{CoinToss}(c) : b = 0] = 1 - c \qquad \text{and} \qquad \Pr[b \leftarrow \mathsf{CoinToss}(c) : b = 1] = c \ .$$

It is relatively easy to implement such a procedure given access to a fair coin. For instance, we can create $\frac{3}{8}$-biased coin by first making three fair coin tosses and outputting one if the outcome is either 000, 001 or 010. In general, we need $n - 1$ tosses of a fair coin to approximate the output of a biased coin with an error 2^{-n}. Moreover, the procedure is really efficient, as the decision procedure corresponds to a single comparison.

Armed with this knowledge, we can define a new distinguisher $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine:

$\mathcal{B}(x)$
- $b \leftarrow \mathsf{CoinToss}(c)$
- if $c = 1$ then **return** $\mathcal{A}(x)$
- else **return** 1

For fixed bias c, the overhead in the running time of $\mathcal{B}$ is constant. However, it is also interesting to consider c as a varying parameter. Let n be the number of fair coin-tosses inside $\mathsf{CoinToss}$ procedure. Then we can coose the parameter c from the set $\left\{\frac{0}{2^n}, \frac{1}{2^n}, \ldots, \frac{2^n}{2^n}\right\}$ without changing the computational complexity.

By the construction it is straightforward to express:

$$\Pr[\mathcal{B}(x) = 0|\mathcal{H}_1] = \Pr[b \leftarrow \mathsf{CoinToss}(c) : b = 1] \cdot \Pr[\mathcal{A}(x) = 0|\mathcal{H}_1]$$
$$\Pr[\mathcal{B}(x) = 1|\mathcal{H}_0] = \Pr[b \leftarrow \mathsf{CoinToss}(c) : b = 0] + \Pr[b \leftarrow \mathsf{CoinToss}(c) : b = 1] \cdot \Pr[\mathcal{A}(x) = 1|\mathcal{H}_0]$$

and thus the ratio of false negatives and false positives for new algorithm are

$$\alpha(\mathcal{B}) = \Pr[\mathcal{B}(x) = 0|\mathcal{H}_1] = c \cdot \alpha(\mathcal{A})$$
$$\beta(\mathcal{B}) = \Pr[\mathcal{B}(x) = 1|\mathcal{H}_0] = (1 - c) + c \cdot \beta(\mathcal{A}) \ .$$

Figure 1 shows how the choice of c influences these ratios. The blue line shows the tradeoffs achievable with the construction given above. The read line shows the tradeoffs achievable by the symmetric construction.

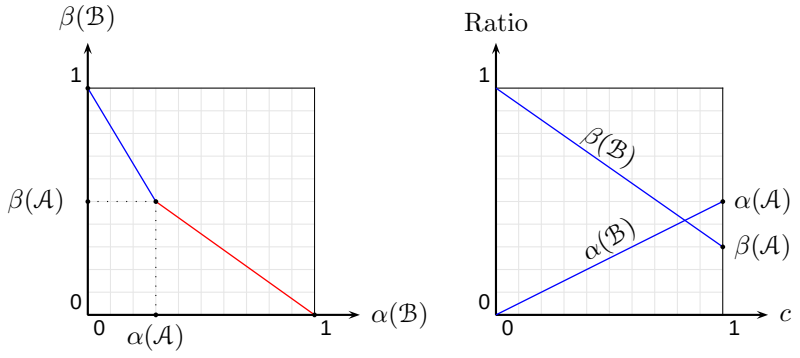

Figure 1: Various achievable tradeoffs between false positives and false negatives

Note that the tradeoff is not without cost. Recall that we can always keep the aggregate error $\gamma(\mathcal{A}) \leq 1$ by inverting the output if necessary. Hence, the aggregate error

$$\gamma(\mathcal{B}) = (1-c) + c\gamma(\mathcal{A}) = \gamma(\mathcal{A}) + (1-c) \cdot (1-\gamma(\mathcal{A}))$$

always increases as long as c decreases. The effect is larger for good distinguishers that have smaller the aggregate error.

Practical applications. In practice, the cost of making false positive and making false negative errors is not the same. For instance, if a bank gives out loans, then the false positives are companies or persons who should not get the loan because they cannot afford it. In this case, making a false positive decision can cost millions dollars, while not giving a loan as much smaller penalty. In healthcare, the situation can be reversed. Usually, a false positive diagnosis is not a problem, as further studies correct the diagnosis while undetected cancer can cause severe complications or even death.