Introduction to Homomorphic Cryptosystems
Exercise Sheet 7: Newton-Raphson

Task 1 – Newton Loop

To show how important the choice of a good starting value x_0 is for the Newton-Raphson method, we consider the following function $f(x) = x^3 - 2x + 2$ for which we want to calculate x so that $f(x) = 0$.

a) Calculate the derivative of f(x).

b) Write down the Newton-Raphson iteration formula for the function $f(x)$, substituting $f(x)$ and $f'(x)$ accordingly.

c) Using the Newton-Raphson iteration formula, calculate the first five iteration results x_1, x_2, x_3, x_4, x_5 if the value 0 is used for x_0.

d) Using the Newton-Raphson iteration formula, calculate the first five iteration results x_1, x_2, x_3, x_4, x_5 if the value -1 is used for x_0.

e) Using the Newton-Raphson iteration formula, calculate the first five iteration results x_1, x_2, x_3, x_4, x_5 if the value -2 is used for x_0.

Task 2 – Inverse Calculation via Brute Force Newton

In the lecture, we learned that the inverse of b can be calculated using the following iteration formula: $x_{n+1} = x_n(2 - x_n b)$. In this task, we implement the brute force approach, which calculates the initial guess x_0 for the inverse determination using the Newton-Raphson approach. To do this, we first define the following nomenclatures and assumptions: (1) we denote the value whose inverse is to be determined as b, (2) we assume that $b \in [u, o]$, (3) we denote the number of calculated iterations of the Newton-Raphson method as iN, (4) we calculate x_0 using the linear function $x_0 = m * t + y$, and (5) we denote the amount of sampling points as s.

Note: The tasks a - e must not be implemented homomorphically.

a) Implement the auxiliary function $h(t, m, y)$, which returns x_0

b) Implement the function $\textbf{Newton}(t, x_0, iN)$, which returns x_{iN}

c) Implement the function $\textbf{Error}(t, x_0, iN)$, which returns the difference of t^{-1} and $\textbf{Newton}(t, x_0, iN)$

d) Implement the function $\textbf{ErrorTotal}(m, y, o, u, s, iN)$, which returns $\sum_{i=0}^{s} \textbf{Error}(t_i, h(t_i, m, y), iN)^2$ where $t_i = u + \frac{o-u}{s} * i$

e) Use the above implementations to find the good values for $m \in \{-1, -0.9, ..., 0.9, 1\}$ and $y \in \{-1, -0.9, ..., 0.9, 1\}$ for the calculation of the division function on the interval $[10, 20]$ with $s = 20, iN = 3$.

f) Use the found values for m and y to homomorphically compute the inverse of 15.

Task 3 – Square Root Calculation via Brute Force Newton

In the lecture, we learned that the square root of a can be calculated by first using the Newton-Raphson approach to approximate $\frac{1}{\sqrt{a}}$ and then multiply this result with a. Implement the calculation of the square root analogous to Task 2.