# 2-Faktor-Authentifikation mit Secret Sharing

Es wurde eine schweres Passwort erzeugt (eine große Zahl die man entsprechend einer Kodierung als Zeichenfolge interpretiert). Für Shamir's Secret Sharing genügt uns jedoch die Interpretation als Dezimalzahl welche wir als Geheimnis $s$ für unser Polynom wählen um Shares zu generieren.
Der Algorithmus ist so weit fortgeschritten, dass bereits $k-2$ Monome des Polynoms *zufällig* erstellt wurden, i.a.W. $a_2, \ldots, a_{k-1}$ wurden zufällig erzeugt. Jetzt muss noch $a_1$ aus dem Share deterministisch erzeugt werden, denn der Share soll vom User gewählt werden. Wir haben also folgendes Polynom:

$$P[X] = \underbrace{s}_{\substack{\text{secret}}} \quad \underbrace{+a_1 x}_{\substack{\text{abhängig vom} \\ \text{Share des Users}}} \quad \underbrace{+a_2 x^2 + \ldots + a_{k-1} x^{k-1}}_{\substack{a_i \text{ zufällig,} \\ f(x):=}}$$

Der User hat sich für das Passwort **Dolphin** entschieden (Faktor „Wissen"). In UTF-8 Codierung ist das[1]:

$$44\ 6F\ 6C\ 70\ 68\ 69\ 6E_{16}$$

Wir interpretieren das Wort jetzt aber als Dezimalzahl und teilen sie willkürlich (z.B. in der Mitte):

$$\underbrace{6811110811}_{x:=}\underbrace{2104105110}_{y:=}{}_{10}$$

Sei $y$ die Auswertung des Polynoms für die Eingabe $x$. Dann können wir $a_1$ berechnen wie folgt:

$$a_i = \frac{y - s - f(x)}{x}$$

Damit ist das Polynom fertig bestimmt und nun können die Shares für den Faktor „Besitz" erzeugt werden.

[1] `https://r12a.github.io/apps/conversion/`