**IIS P1a – Cryptochallenge**

**–**

**Group 29**

Thomas Hödl, Mario Theuermann, Stephan Valentan, Camilla Reis

Specifications:

Just start programs „factoringOfN.py“ (Task 1) and „collision\_search.py“ (Task 2) with Python 3.5; no parameters needed.

Task1:

Pollard‘s Rho algorithm

<https://www.cs.colorado.edu/~srirams/courses/csci2824-spr14/pollardsRho.html>

Ext. Euclid algorithm:

<https://en.wikibooks.org/wiki/Algorithm_Implementation/Mathematics/Extended_Euclidean_algorithm>

RSA:

<http://doctrina.org/How-RSA-Works-With-Examples.html>

Given:

e: 867512337310254731119

n: 1180592782282757817137

c: "MMCTNLAZNXEFUEN"

Solutions:

q: 34359771223

p: 34359739319

phi: 1180592782214038306596

d: 42006330222808060219

Message: KAPFENBERG

Runtime in seconds: ~ 0.853201150894165

Memory requirements: ~ 3,2 MB

Task2:

Brents algorithm for circle detection:

<https://en.wikipedia.org/wiki/Cycle_detection>

Detecting a Loop:

<http://codingfreak.blogspot.com/2012/09/detecting-loop-in-singly-linked-list_22.html>

Given:

Prefix: 1430320143050014305291430751

Started with number: 2ac5 (randomly chosen)

Solutions:

Collision with hash value df62fa19cc3ac958

Colliding values (prefix + hash value):

14303201430500143052914307512f1a142ddfa25079

1430320143050014305291430751d71c3c584b5bf4d3

Runtime in xx: ~

Memory requirements: ~ 3,3 MB