Introduction to Information Security
P1.a Cryptochallenge

Winter term 2016/2017

1 Public-Key Cryptography - RSA

You need to solve a challenge involving the RSA cryptosystem. A string $a_0, a_1, \ldots a_k$, with a_i a letter from $\{A, B, C, \ldots Z\}$, is letter-wise encoded into integers e_i:

$$e_i = f(a_i), \text{ with } f(A) = 0, f(B) = 1, \ldots f(Z) = 25$$

The message m is then encoded as:

$$m = f(a_0) + f(a_1) \cdot 26 + f(a_2) \cdot 26^2 + \ldots + f(a_k) \cdot 26^k$$

- In the file `ciphertext` you can find the cipertext.
- The file `pubkey` contains the public key (e, n) with the public exponent e and the public modulus n.
- Compute the plaintext. You can assume that the plaintext does not end with an A.

2 Hash-Collision Search

Your task is to find a collision for a modified version of the SHA-2 hash function. We use the SHA-256, which generates a 256-bit output. For the exercises, the output of the hash function is truncated to 64 bits using the following rule: let $h = A\|B\|C\|D\|E\|F\|G\|H$ the 256-bit output of SHA-256, with A, B, C, D, E, F, G and H being 32-bit chunks. The modified hash is computed using a bitwise XOR and a concatenation: $h' = E \oplus F \oplus G \oplus H \| A \oplus B \oplus C \oplus D$.

Your colliding messages need to start with the same prefix, which is a concatenation of your matriculation numbers in ascending order (in standard ASCII encoding). There are no restrictions for the message after this prefix. For instance, a group with members 1030123 and 1030456 needs to find colliding messages of form $m_1 = 10301231030456X$ and $m_1 = 10301231030456Y$, with $X \neq Y$ and a freely chosen length of X and Y.

```
SHA-256("10301231030456") =
         = 0xc899398cd246818ffab4ba8842f3251ac2fc27d690979d1891322345818ff381
SHA-256-mod("10301231030456") = 0x42d66a0aa2982791
```

You should use an existing SHA-2 implementation, do not implement it yourself.

The memory requirements of your program must not exceed 2GB (RAM and HDD). Look for solutions that do not require such a large amount of memory.

2.1 Tips

Start off with collision search for smaller hashes, e.g., only the first 32 bits of the hash. Expect that the collision search takes some time. Depending on your code and machine, the search can take more than an hour.

3 Results and Discussion

Write a short summary for each of the solved challenges. This summary should include:

- Your used algorithms
- Solution (plaintext, keys, colliding messages) or a reason why you could not solve an example
- Runtime and memory requirements of your algorithm (especially for hash collision)

4 Further Information

- It is sufficient to provide distinct programs for each challenge.
- You are free in the choice of programming language, but you must include instructions on how to build and run your code.
- A GUI is not required.
- Document your code! Use comments to explain the purpose of your different algorithm parts.