JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

Fachbereich Wirtschaftswissenschaften
Institut für Wirtschaftsinformatik
Lehrstuhl für M-Business & Multilateral Security

Fachbereich

Institut für Wirtschaftsinformatik
Chair of Mobile Business & Multilateral Security
www.m-chair.de

Sascha Löbner, M.Sc.

E-Mail security@m-chair.de

Information and Communications Security WS 2020/21 Assignment 2 – Bonus Material *Cryptography*

In the following you will find some bonus material to the second exercise including Modular Exponentiation and Fermat's factorization.

Modular Exponentiation

The fast exponentiation algorithm is based on the assumption that for

$$x^a mod\ n = (((x^2 mod\ n)^2\ mod\ n)^2 \dots)^2\ mod\ n, \qquad with\ \ a = 2^k\ (a\ power\ of\ 2)$$

squaring is iterated k times. For example, $x^{(2^{2048})}$ can be computed in 2048 modular squaring operations (Knospe 2019, p. 67).

For exponents that are not a power of 2, we can write it as a sum of powers of 2. (Remember that $x^b \cdot x^c = x^{b+c}$)

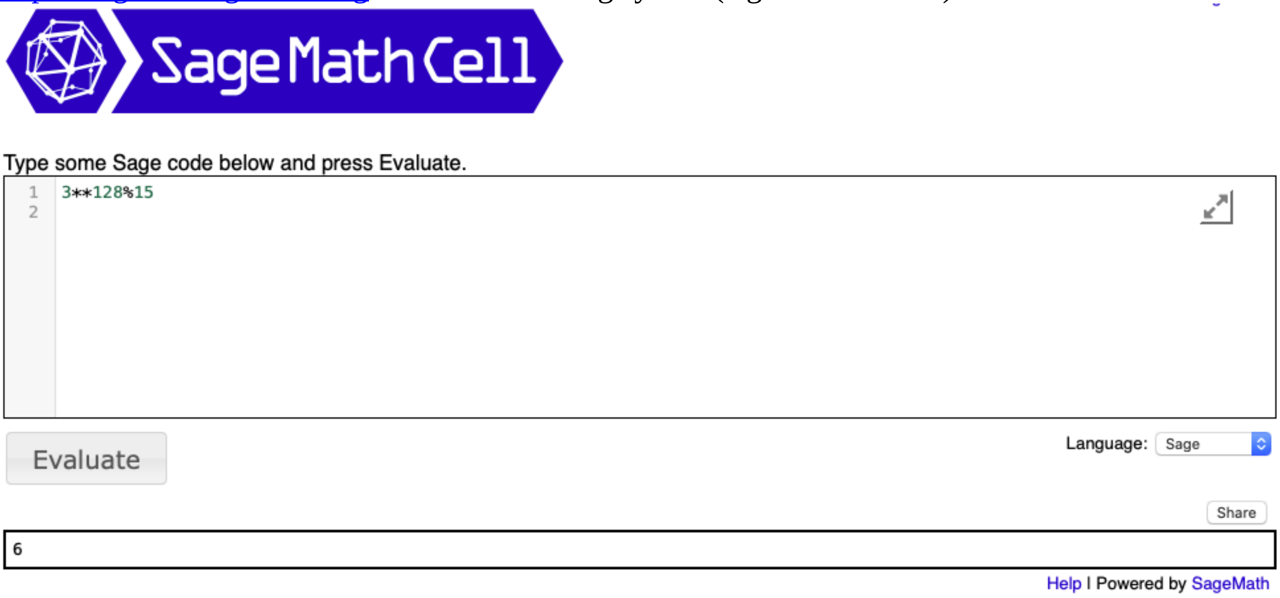
Example (Lecture Slide 21):
To calculate $7^{17} mod\ 77$ we can write 17 = $2^4 + 2^0$.
We also have $2^4 = 2^2 \cdot 2^2$ and $2^2 = 2^1 \cdot 2^1$.

Powers of 2				
16	8	4	2	1
2^4	2^3	2^2	2^1	2^0

$7^1 \equiv 7 \text{ mod } 77 \Leftrightarrow 7 \text{ mod } 77\ = 7$
$7^2 \equiv 49 \text{ mod } 77 \Leftrightarrow 49 \text{ mod } 77\ = 49$
$7^4 \equiv 7^2 \cdot 7^2 \equiv 49 \cdot 49 \text{ mod } 77 \Leftrightarrow 2401 \text{ mod } 77\ = 14$
$7^8 \equiv 7^4 \cdot 7^4 \equiv 14 \cdot 14 \text{ mod } 77 \Leftrightarrow 196 \text{ mod } 77\ = 42$
$7^{16} \equiv 7^8 \cdot 7^8 \equiv 42 \cdot 42 \text{ mod } 77 \Leftrightarrow 1764 \text{ mod } 77\ = 70$
$7^{17} \equiv 7^{16} \cdot 7^1 \equiv 70 \cdot 7 \text{ mod } 77 \Leftrightarrow 490 \text{ mod } 77\ = 28$

Bonus Exercise:

Use modular exponentiation to solve $4^{64} mod\ 7$ and $3^{210} mod\ 15$. You can check your results on https://sagecell.sagemath.org with the following syntax (e.g. $3^{128} mod\ 15$):



Fermat's Factorization

Let p, q be prime and $N = pq$. Fermat's factoring represents N as a difference of 2 squares:

$$N = \boldsymbol{x^2 - y^2 = (x + y)(x - y)).}$$

First, we start with x = $\lceil\sqrt{N}\rceil$ and then increase x by 1 until $x^2 - N$ is square (so that we can derive y) so that $\boldsymbol{N} = x^2 - y^2$ holds.

This method works because we can represent N as a difference of 2 squares:

$$pq = (\tfrac{1}{2}(p + q)^2 - (\tfrac{1}{2}(p - q)^2)^2 = x^2 - y^2.$$

You will find this explanation with more details in Knospe 2019, p. 178 f.

Bonus Exercise:

Let $N = 247$; then we first set $x \approx \sqrt{N}$. We obtain $x = 16$ and derive $x^2 - N = 9$. Because 9 is square we know that $y = 3$. From above we know that $pq = (x + y)(x - y)$ so we receive $247 = 19 \cdot 13$.

If you need more practice, just choose some primes p, q and derive N. Then use Fermat's factoring to solve N.

Literature

(Knospe 2019) Knospe, Heiko. A Course in Cryptography. Vol. 40. American Mathematical Soc., 2019. https://ebookcentral.proquest.com/lib/senc/reader.action?docID=5962876