Cryptography

PROF. DR. JOACHIM VON ZUR GATHEN, KONSTANTIN ZIEGLER

3. Assignment: CRT for RSA, Orders, and Repeated Squaring

(Due: **Thursday**, 19 November, 23^{59})

Exercise 3.1 (Small Public Exponent RSA Cryptosystem). (10 points)

This exercise will show that, when using the RSA cryptosystem as a public key encryption scheme, small public exponents may be a real danger.

In a public domain the exponent $e = 3$ is used as public exponent, thus every user chooses a public modulus N such that $\gcd(\varphi(N), 3) = 1$ and computes his respective secret exponent d such that $(3 \cdot d) \mod \varphi(N) = 1$. Suppose that the users A, B, C have the following public moduli:

$$N_1 = 5000746010773,\ N_2 = 5000692010527,\ N_3 = 5000296004107.$$

(i) ALICE sends a message m to A, B, C by encrypting: $m_i = m^3 \mod N_i$. EVE drops in and captures the following values: $\boxed{6}$

$$m_1 = 1549725913504,\ m_2 = 2886199297672,\ m_3 = 2972130153144.$$

Show that EVE can recover the value of m without factoring N_i and compute this value with a Computer Algebra System of your choice (Maple, MuPAD, Mathematica, SAGE, etc.). (Hint: Use the Chinese Remainder Theorem.)

(ii) Generalize the method used by EVE above for a general public exponent e. How many messages should EVE intercept in order to recover the clear text message? $\boxed{4}$

Exercise 3.2 (Orders). (15 points)

Let G be a (multiplicative) commutative group, a an element of order u and b an element of order v. We want to investigate two questions:

- What is the order of $a^2, a^3, \ldots$?
- What are possible orders of ab?

First, let us look at an example: Take $G = \mathbb{Z}_{1321}^{\times}$, $a = 53$ and $b = 17$. We have $a^{33} = 1$ and $b^{24} = 1$ in G and for all respective smaller positive exponents the result is different from 1.

computer security

(i) Compute the order of a^2, a^3, a^9, a^{10}, a^{11}. $\boxed{2}$

$\boxed{2}$ (ii) Compute the order of ab, a^2b, a^3b.

Now, we want to investigate the general case:

$\boxed{3}$ (iii) Show:

$$a^k = 1 \text{ if and only if } \operatorname{ord}(a) | k.$$

$\boxed{3}$ (iv) Show: The order of the power x^n of a group element $x \in G$ is the order of x divided by the greatest common divisor of n and that order.

In short:

$$\operatorname{ord}(x^n) = \operatorname{ord}(x) / \gcd(n, \operatorname{ord}(x)).$$

(Hint: Look at the special cases $\gcd(n, \operatorname{ord}(x)) = 1$ and $n | \operatorname{ord}(x)$ and derive the general solution from there.)

$\boxed{3}$ (v) Show: If the orders of two group elements $x, y \in G$ are coprime, then the order of xy is actually equal to the product of those orders.

In short:

$$\text{If } \gcd(\operatorname{ord}(x), \operatorname{ord}(y)) = 1, \text{ then } \operatorname{ord}(xy) = \operatorname{ord}(x) \cdot \operatorname{ord}(y).$$

$\boxed{2}$ (vi) Provide an example for x, y and N, such that $\operatorname{ord}(xy) = \operatorname{ord}(x) \cdot \operatorname{ord}(y)$ is wrong.

Exercise 3.3 (`RepeatedSquaring`). (13 points)

The purpose of this exercise is to examine the `RepeatedSquaring`-algorithm more closely. Given the multiplicative group $\mathbb{Z}_N^\times$, a base $x \in \mathbb{Z}_N^\times$ and an exponent $e \in \mathbb{Z}$ with $1 \le e < \#\mathbb{Z}_N^\times$, this algorithm determines $x^e \in \mathbb{Z}_N^\times$.

$\boxed{4}$ (i) Implement `RepeatedSquaring` *as it was presented in the lecture* with a Computer Algebra System of your choice (Maple, MuPAD, Mathematica, SAGE, etc.). Furthermore, add a counter, that returns the number of group operations (squarings and multiplications) that were performed.

Hand in a properly documented print-out of your source code.

$\boxed{2}$ (ii) We consider the following modification of the algorithm: Let $\sum_{0 \le i < \lambda} e_i 4^i$ be the 4-ary representation of e with $e_0, \ldots, e_{\lambda-1} \in \{0, 1, 2, 3\}$. To obtain the correct result now, we modify the squaring step by using $y \leftarrow y^4$ instead of $y \leftarrow y^2$ and the multiplication step by using $y \leftarrow y \cdot x^{e_i}$ instead of $y \leftarrow y \cdot x$.

The powers x^{e_i} occur over and over again, so it makes sense to precompute them. Insert an appropriate step. You also have to modify the initial $y \leftarrow x$. How?

cosec computer security

$\boxed{7}$ (iii) Finally we compare the performance of the two algorithms. Let the modulus $N = 13699155487$, the base $x = 561$ and the exponents e be all numbers of the form $1234567890 + (ID \mod 100 + i) \cdot 54321$, where $i = 0, 1, \ldots, 99$ and $ID \mod 100$ are the last two digits of your student ID.

Draw a coordinate system, where the horizontal axis is labelled with the exponents e and the vertical axis with the number of group operations.

Plot the number of group operations for both algorithms and give in each case the result of the algorithm (the power x^e and the number of operations) for both, the smallest and the largest exponent.

Finally, compute the average number of group operations for each algorithm.

computer CoSec security