Number theory in cryptography

– Exercise set 5 –

The exercises T5.2, P5.1 have to be handed in on Tuesday, 26th March 2024, 8:30 at latest.

Theoretical questions

T 5.1 In this exercise, we will study the following probabilistic algorithm which outputs a generator of $\mathbb{F}_p^*$, when p is a given odd prime number such that the factorization into prime powers of

$$p - 1 = \prod_{i=1}^{r} q_i^{e_i}$$

is known.

Algorithm 1 Probabilistic generator

Require: A prime p and factorization $p - 1 = \prod_{i=1}^{r} q_i^{e_i}$.
Ensure: A generator γ of $\mathbb{F}_p^*$.

for $i = 1, \dots, r$ **do**
    $\beta_i := 1$
    while $\beta_i = 1$ **do**
        pick $\alpha \in \mathbb{F}_p^\times$ randomly
        $\beta_i := \alpha^{(p-1)/q_i}$
        $\gamma_i := \alpha^{(p-1)/q_i^{e_i}}$
    end while
end for
return $\gamma := \prod_{i=1}^{r} \gamma_i$

---

a) Prove that the algorithm indeed returns a generator of $\mathbb{F}_p^*$ when it terminates.

b) At a given step i, what is the probability that $\beta_i = 1$ in the `while` loop? Deduce the expected number of trials before getting $\beta_i \neq 1$. (Hint : you can use the fact that the expected value of a geometric distribution with success probability p_0 is $1/p_0$).

c) Using the previous part, show that the expected running time of the algorithm is $O(\log(p)^d)$ for some integer $d > 0$.

d) (optional) Implement the algorithm in SAGE. You are allowed to use a factorization function from SAGE to pre-compute the factors $q_i^{e_i}$ of $p - 1$.

T 5.2 Here is a polynomial analogue to RSA. Alice chooses a finite field k with q elements and two distinct irreducible polynomials $P, Q \in k[X]$, of degree m and n respectively. She also picks some public key $e \in \mathbb{Z}_{>0}$ coprime to $f := (q^m - 1)(q^n - 1)$, and publishes the pair $(e, N := PQ)$.

Bob chooses a message $M \pmod{PQ}$ (where $M \in k[X]$ is coprime to N) and sends the ciphertext $C := M^e \pmod{PQ}$ to Alice.

a) Explain how Alice can decrypt the ciphertext. Prove in detail the correctness of your computations.

b) Explain briefly why this protocol is insecure.

T 5.3 Here you are going to see a concrete example of the Chinese Reminder Theorem for polynomials. Compute (mostly by hand) a polynomial $a(X) \in \mathbb{F}_{13}[X]$ such that

$$\begin{aligned} a(X) &\equiv X \pmod{X^2+4} \\ a(X) &\equiv 1 \pmod{2X+3} \end{aligned}$$

Programming exercises

P 5.1 Implement a function `remove_repeated_factors(f, q)` on SAGE, which takes a polynomial $f \in \mathbb{F}_q[X]$ as input, and outputs a *square-free* polynomial $\widetilde{f}$ that divides f and which has the same irreducible factors as f.

Hint: don't forget to consider polynomials of the form $f(X) = g(X^p) = g(X)^p$ where $q = p^r$ for some $r > 0$. Also you can use the pre-implemented functions of SAGE for euclidean division, gcd, derivative, coefficients, but *not* the `factor` function.