ALGORITHMICS, CRYPTOGRAPHY & COMPUTER SECURITY, graduate programs 2023
CRYPTOGRAPHY, SPRING 2023, assignments list # 10, 31.5.2023

1. Estimate the probability that two RSA numbers of bitlength 2012 generated according to the specification (the starting point for the search is really chosen at random) are not coprime. Of course, you have to refer to the Prime Number Theorem on density of primes.

2. Take $n = 77$. Estimate the probability that a single iteration of the Miller-Rabin test provides a witness that 77 is not a prime number.

3. Consider RSA-OAEP and the situation where r is always a string of zeros. Is it possible to manipulate such RSA-OAEP ciphertexts? What is the advantage of having unpredictable r?

4. Consider McEliece asymmetric encryption for the parameter $t = 0$ (that is, no errors are injected). Could you attack the scheme in this case?

5. A commitment scheme is that a user presents a value $C = F(x)$, where x is the committed value, and C is a commitment for x. The user can *open* the commitment, that is, can present x and a witness w to prove that $C = F(x)$. The point is that the verifier cannot verify that C is a commitment for x, unless C is *opened*.

   With a bilinear mapping one can create commitments of a constant size so that one can commit to q elements at once. Moreover, each value x_i can be opened separately.

   setup: a biliner mapping $e : G \times G \to G_T$, where G, G_T are multiplicative groups of prime order p, a trapdoor $(g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^q})$ for a secret α.

   creating a commitment: for $x_0, \ldots, x_q < p$: create a polynomial $f(X)$ such that $f(i) = x_i \bmod p$, compute $C := g^{f(\alpha)}$.

   opening x_i: compute $f_i(X) = \frac{f(X)-f(i)}{X-i}$. Return $i, x_i, w_i = g^{f_i(\alpha)}$

   verification: check that $e(w_i, g^\alpha/g^i) = e(C/g^{x_i}, g)$

   Questions:

   - Why α should be secret?
   - If you do not trust the setup, you modify α to $\alpha \cdot \delta \bmod p$ for random $\delta < p$. How to recompute the trapdoor? We assume that you know only the original trapdoor and δ.
   - How to create f?
   - How to compute C? You do not know α, so direct application of the definition is impossible!
   - Why f_i is a polynomial?
   - Why does the verification test succeed, if the opening is correct?
   - Why the author of the commitment cannot cheat and present $x'_i \neq x_i$? An informal argument suffices.

6. What would happen if you modify ANSI X9.31 padding for RSA signatures so that the bytes in front of Hash(M) (more significant positions) are replaced by zeros? Try to find the sources of problems.

/-/ Mirosław Kutyłowski