Wrocław University of Technology, WPPT 25.10.11

CRYPTOGRAPHY AND SECURITY, 2011 Assignments, list # 4

1. Construct a field consisting of 4 elements. Write down tables defining multiplication and addition in this field. Hint: use the construction via polynomials over $\mathbb{Z}_2$.
2. Write down a pseudo-code for computing $a^{-1} \bmod p$ for a prime p and $a < p$. Use Binary GCD.
3. One of the ideas to bypass security of ElGamal signatures (and similar ones) is to compute random parameters in so called "kleptographic way". The solution is to
   - store $U = g^u$ inside a infected device (but not u, element u must be kept secret by the attacker)
   - instead of choosing parameter k at random during signature creation, execute the following procedure
     (a) restore k' from the previous signature generated by the device,
     (b) $k := H(U^{k'})$

   Show how the attacker may derive k and consequently signing key x using the previous signature (r', s'). Hint: compute $(r')^u$... Is DSA secure against such kind of attacks?
4. Consider a signing device D such that after receiving the message m to be signed, D performs the following steps:
   (a) choose k and $r := g^k$
   (b) compute $H(M)$
   (c) compute signature component s according to ElGamal scheme (or DSA, Schnorr, ...).

   We assume that we can "rewind" D to exactly the same state as occurs after step (a) and replace the module for computing H by another one. How to derive the signing key in this scenario? Use such D to compute discrete logarithms of public keys. Formulate the attack in the language of *random oracle model*.
5. Assume that p is a prime number, $a < p$, $i < p - 1$. What is the number of roots of $a \bmod p$ of degree i? Describe all possible cases.
6. Let n be an RSA number. Let $k > 2$, and $a < n$ with $\gcd(a, n) = 1$. What is the number of roots of a of degree k?
7. Estimate the expected runtime of factorization on an RSA number $n = pq$ with the rho-Pollard algorithm.
8. One can factorize RSA number n based on knowledge of a pair of RSA keys e, d:
   - compute $ed - 1 = 2^s t$, where t is an odd integer,
   - choose $a < n$ at random,
   - find maximal i such that $a^{t \cdot 2^i} \neq 1$,
   - if $a^{t \cdot 2^i} \neq -1$, then compute $GCD(n, a^{t \cdot 2^i})$ and get a nontrivial factor of n.

   Why this method works? What is the probability of success in a single iteration with an a chosen at random?

/-/ Mirosław Kutyłowski