Cryptography

Priv.-Doz. Dr. Adrian Spalka, Konstantin Ziegler

Assignment 10: more RSA, ElGamal sigantures, and the index calculus

Due: Monday, 31 January 2011, 10^{00}

Exercise 10.1 (more RSA)**.** (6 points) (*The lecture notes may provide valuable hints.*)

(i) Small public keys are dangerous – even for single users. Let $e = 3$ and assume knowledge of two ciphertexts $c_1 = m_1^3 \mod N$ and $c_2 = m_2^3 \mod N$ with the property $m_2 = m_1 + 1$. Derive a computable formula for m_1 involving only c_1 and c_2.

(ii) You have already seen existential forgery for RSA signatures. But it gets better. RSA is *homomorphic*, meaning

$$\operatorname{enc}(m_1) \cdot \operatorname{enc}(m_2) = \operatorname{enc}(m_1 \cdot m_2).$$

You want to forge the signature for a specific message m and have access to an oracle which will provide you with valid signatures for two messages of your choice (of course, different from m). Can you do it?

(iii) An RSA infrastructure should be established for a large group of people. In order to speed up the key generation and unify further processes, the management has decided that the same N should be used by everybody and the keygeneration-algorithm is adjusted accordingly. Still p, q, and $\varphi(N)$ are properly destroyed after the keygeneration and everybody's secret is chosen individually and randomly. Comment on the security of this system.

Exercise 10.2 (ElGamal signatures)**.** Let us get some hands-on experience with the ElGamal Signature Scheme. (*Use the notation that was fixed in the tutorial.*)

Let $p = 2^{28} + 3$ and $g = 3$ a generator of $G = \mathbb{Z}_p^\times = \{1, \ldots, p-1\}$. The injective encoding function $G \to \mathbb{Z}_{p-1} = \{0, \ldots, p-2\}, x \mapsto x^\star$ is given by

$$x^\star = \begin{cases} 0 & \text{for } x = p-1 \\ x & \text{else.} \end{cases}$$

Our message m will be the first four letters of your given name. Add an exclamation mark, if your given name has less than four letters and mind the capitalization. Look up the 7-bit ASCII encodings for each letter and concatenate them for the 28-bit number m.

Let us take the role of Alice and let $a = 100$ be our secret key.

(i) (2 points) Choose a random session key k (of at least three digits) and generate a signature for your message m.

(ii) (2 points) What is your public key? Use it to verify the signature you just produced.

Even small errors in the process can compromise the whole system.

(iii) (2 points) Alice sends the signed message

$$(m, K, \sigma) = (500, 10\,296\,631, 248\,708\,422).$$

By accident the secret session key $k = 787$ is revealed. Compute Alice's secret key a.

(iv) (2 points) After this experience, Alice changes her secret key and the public version is now $A = 138\,309\,740$. Unfortunately a bug/feature in the random number generator revealed that the same value for k was generated twice in a row. This is known for the signed messages

$$(501, 32\,067\,479, 51\,030\,675)$$

and

$$(502, 32\,067\,479, 60\,076\,072)$$

Compute Alice's secret key.

Exercise 10.3 (index calculus for the discrete logarithm problem)**.** We want to see the index calculus for the discrete logarithm problem in action. We are interested in the multiplicative group $G = \mathbb{Z}_p^\times$ with $p = 227$ and generator $g = 2$. We choose as factor base $\mathcal{B} = \{2, 3, 5, 7, 11\}$ with all primes up to the bound $B = 11$.

In the preprocessing step we compute the discrete logarithms of all elements in the factor base $\mathcal{B}$.

(i) (2 points) Instead of randomly choosing exponents e and testing, whether $g^e \operatorname{rem} p$ factors over $\mathcal{B}$, we have already prepared a list with suitable exponents for you. Let e take values from $\{40, 59, 66\}$, give the factorization of $g^e \operatorname{rem} p$ over $\mathcal{B}$ and the corresponding linear congruence modulo $(p-1)$ involving the discrete logarithms of the elements in $\mathcal{B}$.

(ii) (2 points) The discrete logarithm of the generator $g = 2$ is obviously 1, but even with this information, the three linear relations from (ii) are not enough to determine the remaining four unknown discrete logarithms. Find one additional linear congruence from an exponent $e > 10$ yourself.

(iii) (2 points) Assuming that your additional congruence is linearly independent from the three previous ones, solve the system of congruences for the discrete logarithms of the base elements. (If you do this by hand, note that division by 2 is impossible modulo $(p-1)$. If you use a computer algebra system, note that those are aware of this problem and have special commands to solve systems of congruences with a given modul, e.g. `msolve` in MAPLE and `solve_mod` in SAGE.)

Once we have found the discrete logarithms for the elements in the factor base, we can finally compute the discrete logarithm of any element x in the group with the following method:

- Choose random exponents e until $xg^e \operatorname{rem} p$ factors over $\mathcal{B}$, say $xg^e \equiv p_1^{\beta_1} p_2^{\beta_2} \cdots p_h^{\beta_h} \mod p$.
- The corresponding linear relation reads

  $$\operatorname{dlog}_g x + e = \beta_1 \operatorname{dlog}_g p_1 + \beta_2 \operatorname{dlog}_g p_2 + \cdots + \beta_h \operatorname{dlog}_g p_h \mod (p-1)$$

- Since all the $\operatorname{dlog}_g p_i$ have already been determined in the preprocessing step, you can solve this equation modulo $(p-1)$ for $\operatorname{dlog}_g x$.

(iv) (2 points) Apply this procedure to compute $\operatorname{dlog}_2 224$ in $\mathbb{Z}_{227}^\times$.