# Security

**Assignment 13, Friday, December 22, 2017**

**Handing in your answers:**

- Include your name and student number **in** the document (they will be printed!), as well as the name of your teaching assistant (Bart or Joost). When working together, include **both** your names and student numbers.
- Submit one single **pdf** file – when working together, only hand in **once**.
- Hand in via Blackboard, before the deadline.

**Deadline:** Monday, January 15, 09:00 sharp!

**Goals:** After completing these exercises successfully you should be able to

- be able to perform all operations in the context of RSA signatures.
- perform computations for ElGamal signatures;
- understand the dangers involved in reusing randomness;

**Marks:** You can score a total of 100 points.

1. **(30 points)** Consider RSA as a signature scheme. For each question, give intermediate steps to show how you got your results. Where it is not requested otherwise, you are allowed to use a calculator—provided that you specify clearly what you calculate.
   (a) Alice has chosen primes $p = 11$ and $q = 19$, compute $n$ and $\varphi(n)$.
   (b) Take $e = 7$ and, applying the *extended Euclidean algorithm*, compute $d$ such that $d \cdot e = 1 \pmod{\varphi(n)}$. Now $(n, e)$ and $(n, d)$ are Alice's public and private keys, respectively.
   (c) Compute Alice's signature on $m = 16$. Use $h(m) = m$ as the hash function.
   (d) Verify the signature using the corresponding public key. Use *square-and-multiply*.
2. **(40 points)** DSA.
   Suppose $G = \mathbb{Z}_p^*$ for $p = 29$, with generator $g = 3$. For the order of $G$, we write $\#g = \varphi(p)$ (since $g$ is its generator). In this exercise we will use the (otherwise completely insecure) hash function $h(m) = m$. Let's assume that Alice's private key is $a = 21$. Please make sure to use the correct modulus for each step. Show intermediate steps.
   (a) Determine Alice's corresponding public key $A$.
   (b) Sign the message $m = 15$ using DSA with random value $r = 5$.
      i. Verify that $r$ and $\#g$ are relatively prime.
      ii. Compute $s_1 = R = g^r \bmod p$.
      iii. Compute $r^{-1} \bmod \#g$.
      iv. Compute $s_2 = (h(m) - a \cdot R) \cdot r^{-1} \bmod \#g$
   (c) Verify that the signature $(s_1, s_2)$ is correct on message $m$ using Alice's public key $A$.
      i. Check that $1 \leq s_1 < p$.
      ii. Compute $v := s_1^{s_2} \cdot A^{s_1} \bmod p$.
      iii. Verify $g^{h(m)} \stackrel{?}{=} v$.

3. **(30 points)** Predictable 'randomness'.

   When using the ElGamal scheme, it is crucial that one uses a fresh random number $r$ for each use. However, true random numbers are not that easy to obtain - in practice, they are typically generated *pseudo*-randomly, and sometimes this is done poorly. When this is done in an insecure fashion, an attacker could influence the randomness, cause a system to use the same 'random' value twice or even predict the randomness completely.

   **Note:** for the following questions, carefully consider which parameters are publicly available (and thus also available to an attacker).

   (a) Consider ElGamal encryption (let $G = \mathbb{Z}_p^*$ for some prime $p$). What can an attacker learn if the randomness $r$ is known, and he intercepts an ElGamal ciphertext? Show how!

   (b) Now consider DSA from the lecture. Show what an attacker can learn when the randomness $r$ is known, and he obtains an signature $(s_1, s_2)$ (with the corresponding message $m$). Again, show how!

   (c) Which of these scenarios has more devastating consequences? For example, consider the security of other ciphertexts and signatures for which the used randomness $r'$ is still unknown.