# Digital signatures

**Exercise 1.** *Complexity analysis of the extended Euclidean Algorithm*

The goal of the exercise is to analyze the complexity of the extended Euclidean Algorithm, reminded below.

*Input:* $a$, $b \in \mathbb{Z}_{\geq 0}$, $a > b$
*Output:* $g$, $u$, $v$ such that $g = \gcd(a, b) = au + bv$

```
1 (r_0, u_0, v_0) ← (a, 1, 0)
2 (r_1, u_1, v_1) ← (b, 0, 1)
3 i ← 2
4 While r_{i−1} ≠ 0:
5     (q_i, r_i) ← QuoRem(r_{i−2}, r_{i−1})
6     (u_i, v_i) ← (u_{i−2} − q_i u_{i−1}, v_{i−2} − q_i v_{i−1})
7     i ← i + 1
8 Return (r_{i−2}, u_{i−2}, v_{i−2})
```

**1.** The first goal is to bound the number of iterations of the while loop. For two integers $a$ and $b$, we define $s(a, b) = a + \frac{1}{\varphi} b$ where $\varphi = \frac{1}{2}(1 + \sqrt{5})$, so that $\varphi^2 = \varphi + 1$.

- **i.** Let $a \geq b \in \mathbb{Z}$ and $(q, r) = \textsc{QuoRem}(a, b)$. Prove that $s(b, r) \leq \frac{1}{\varphi} s(a, b)$. *Prove and use that* $\varphi - 1 = \frac{1}{\varphi}$.
- **ii.** Deduce that the number of iterations of the while loop is $O(\log a)$.

**2.** We now bound the growth of the $u_i$'s and $v_i$'s.

- **i.** Prove that for all $i \geq 0$, $r_i v_{i+1} - r_{i+1} v_i = (-1)^i a$ and $r_i u_{i+1} - r_{i+1} u_i = (-1)^{i+1} b$.
- **ii.** Prove that for all $i \geq 0$, $u_{2i} \geq 0 \geq u_{2i+1}$ and $v_{2i} \leq 0 \leq v_{2i+1}$.
- **iii.** Deduce that for $i \geq 1$, $|u_i| \leq b/r_{i-1}$ and $|v_i| \leq a/r_{i-1}$.

**3.** Finally we bound the bit complexity of the algorithm. For, we remind that the product and Euclidean division of two integers $a$ and $b$ can be computed in time $O(\ell_a \ell_b)$ and $O((\ell_a - \ell_b + 1)\ell_b)$ respectively where $\ell_a = \log a$ and $\ell_b = \log b$.[1] For $i \geq 0$, let $\ell_i = \log(r_i)$.

- **i.** Prove that line 5 has cost $O((\ell_{i-2} - \ell_{i-1} + 1)\ell_1)$.
- **ii.** Prove that line 6 has cost $O((\ell_{i-2} - \ell_{i-1})(\ell_0 - \ell_{i-2}))$.
- **iii.** Conclude that the bit complexity of the algorithm is $O(\log(a)\log(b))$.

[1] The fastest algorithms have running time approximately $O(\ell_a \log \ell_b)$ for both problems.

**Exercise 2.** *DSA*

The *Digital Signature Algorithm* (DSA) is a standardized signature scheme based on the discrete logarithm problem. It uses an indentification protocol, which is transformed into a signature scheme (though not through Fiat-Shamir transform). In the exercise, $p$ is a prime number and $G$ is a (cyclic) subgroup of $(\mathbb{Z}/p\mathbb{Z})^\times$ of prime order $q$ with generator $g$. We define a pair keys $sk = x \in \{0, \dots, q-1\}$ and $pk = h = g^x$.

1. The identification protocol works as follows: The prover chooses $k \twoheadleftarrow \{1, \dots, q-1\}$ and sends $\ell \leftarrow g^k$; The verifier chooses $\alpha, r \twoheadleftarrow \{0, \dots, q-1\}$ and sends them; The prover computes $s = k^{-1} \cdot (\alpha + xr) \bmod q$; The verifier accepts iff $s \neq 0$ and $g^{\alpha \cdot s^{-1}} \cdot h^{r \cdot s^{-1}} = \ell$ (where $s^{-1}$ is the inverse of $s$ modulo $q$).
   - **i.** Prove that if $s \neq 0$, the protocol is correct.
   - **ii.** Compute the probability that $s = 0$.
2. To define the DSA signature scheme, we consider a hash function $H : \{0, 1\}^* \to \{0, \dots, q-1\}$. To sign with the private key $x$, the signer simulates the identification protocol, replacing the random choices of $\alpha$ and $r$ by $\alpha \leftarrow H(m)$ and $r \leftarrow \ell \bmod q$. If $s = 0$, the signer restarts with a new value $k$.
   - **i.** Write the algorithm $\mathsf{Sign}$ formally. *What should be the output?*
   - **ii.** Describe the verification algorithm $\mathsf{Vrfy}$ and prove that it is correct.
   - **iii.** We define a variant of DSA where the message space is $\{0, \dots, q-1\}$, and where $H$ is simply omitted. Show that this variant is insecure, that is one can forge a signature without knowing the private key. Is this an existential or a universal forgery?