Homework 11 in Advanced Methods of Cryptography  
- Proposal for Solution -  
Prof. Dr. Rudolf Mathar, Michael Reyer, Henning Maier  
31.01.2014

Solution to Exercise 32.

(a) \( \gcd(a, p - 1) \in \{1, 2, q, 2q\} \) for all \( a \in \mathbb{N} \) since \( p - 1 = 2 \cdot q \) holds.

(b) Consider the following congruence:

\[
k(x_1 - x'_1) \equiv x'_0 - x_0 \pmod{p - 1}. \tag{1}
\]

It follows directly that \( k = \log_p(b) \neq 0 \) since \( b \) is a PE, and hence, \( b \neq 1 \) holds. To determine \( k \), assume both \( 0 < k, k' < p - 1 \) fulfill (1). Then

\[
k(x_1 - x'_1) \equiv x'_0 - x_0 \pmod{p - 1} \land \;
k'(x_1 - x'_1) \equiv x'_0 - x_0 \pmod{p - 1}
\]

\[\Rightarrow (k - k')(x_1 - x'_1) \equiv 0 \pmod{p - 1}. \tag{2}\]

It holds:

\[-(p - 2) < k - k' < p - 2 \land \]
\[-(q - 1) \leq x_1 - x'_1 \leq q - 1 \land \]
\[x_1 \neq x'_1\]

Let \( d = \gcd(x_1 - x'_1, p - 1) \), then it follows from (1) that \( d \mid (x'_0 - x_0) \):

1) \( d = 1 \Rightarrow k - k' \equiv 0 \pmod{p - 1} \Rightarrow k \equiv k' \pmod{p - 1} \), i.e., there is the solution:

\[k = (x_1 - x'_1)^{-1}(x'_0 - x_0) \pmod{(p - 1)}. \]

2) \( d > 1 \):

\[k \left(\frac{x_1 - x'_1}{d}\right) = \left(\frac{x'_0 - x_0}{d}\right) \pmod{\frac{p - 1}{d}}. \tag{3}\]

It holds \( \gcd\left(\frac{x_1 - x'_1}{d}, \frac{p - 1}{d}\right) = 1 \Rightarrow (3) \) has exactly one solution \( k_0 < \frac{p - 1}{d} \) which can be determined by using the Extended Euclidean Algorithm:

\[k_0 = \left(\frac{x_1 - x'_1}{d}\right)^{-1} \left(\frac{x'_0 - x_0}{d}\right) \pmod{\frac{p - 1}{d}}.\]

For the solution of (1), there are multiple candidates

\[k_l = k_0 + l \left(\frac{p - 1}{d}\right), \; l = 0, \ldots, d - 1.\]

Recall from (a) that \( p - 1 = 2q \Rightarrow d \in \{1, 2, q, 2q\} \Rightarrow d \in \{1, 2\} \) as \( (x_1 - x'_1) \leq q - 1 \Rightarrow d = 2 \) as \( d > 1 \).

Check: for \( l = 0 \) if \( a^{k_0} \equiv b \pmod{p} \) or for \( l = 1 \) if \( a^{k_0 + \frac{p - 1}{d}} \equiv b \pmod{p} \) holds.
(c) $p, q$ are prime with $p = 2q + 1$ ($\Rightarrow$ Sophie-Germain primes), $a, b$ are primitive elements modulo $p$. The hash function is defined by:

$$h(m) = a^{x_0}b^{x_1} \mod p$$

with $0 \leq x_0, x_1 \leq q - 1 \wedge m = x_0 + x_1q$.

The given function is slow but collision-free as it will be shown in the following.

Assume a collision exists, i.e., at least one pair of messages satisfies:

$$m \neq m' \wedge h(m) = h(m')$$

$$\iff m \neq m' \wedge a^{x_0}b^{x_1} \equiv a^{x_0'}b^{x_1'} \pmod{p}. \quad (4)$$

for two different messages $m, m'$ with

$$m = x_0 + x_1q,$$
$$m' = x_0' + x_1'q.$$

Furthermore, $x_1 - x_1' \neq 0 \pmod{p - 1}$ must hold, otherwise it would follow from (4) that $m = m'$.

Let $k = \log_a(b)$ modulo $p$, so that:

$$a^{x_0}a^{kx_1} \equiv a^{x_0'}a^{kx_1'} \pmod{p}$$

$$\iff a^{k(x_0 - x_0') + (x_1 - x_1')} \equiv 1 \pmod{p}.$$  

Since $a$ is a primitive element modulo $p$, we may consider the exponent-term as:

$$k(x_1 - x_1') - (x_0' - x_0) \equiv 0 \pmod{p - 1}$$

$$\iff k(x_1 - x_1') \equiv x_0' - x_0 \pmod{p - 1}.$$  

As shown in (b), finding collisions is equivalent to computing the discrete logarithm. This is a hard problem because the determination of a discrete logarithm is computationally extensive.