Dr. Michael Reyer

Tutorial 9
- Proposed Solution -

Friday, January 11, 2019

Solution of Problem 1

a)
- $q \mid p-1 : 7 \mid 71-1 = 70\checkmark$
- $\beta \in \mathbb{Z}_p^*$ shall have order $q=7$.
  $\beta^2 \mod p = 20^2 \mod p = 45$
  $\beta^4 \mod p = 45^2 \mod p = 37$
  $\beta^6 \mod p = \beta^2 \cdot \beta^4 \mod p = 45 \cdot 37 \mod p = 32$
  $\beta^7 \mod p = \beta \cdot \beta^6 \mod p = 20 \cdot 32 \mod p = 1\checkmark$
- $2^t < q : 2^2 = 4 < 7\checkmark$

b) $v \equiv \beta^{-a} \equiv \beta^{-5} \equiv \beta^2 \equiv 45 \pmod{71}$

c)
1. A chooses $r = 3 \in \{1,\dots,q-1\}$ and computes the witness $x = \beta^r \mod p = 20^3 \mod 71 = 48$ and sends it to B.
2. B chooses $e = 4 \in \{1,\dots,2^t\}$ and sends it to A as challenge.
3. A checks $1 \leq e = 4 \leq 2^t = 4$, computes $y = a \cdot e + r = 5 \cdot 4 + 3 = 23 \equiv 2 \pmod 7$ and sends it to B.
4. B computes $z = \beta^y \cdot v^r \equiv 20^2 \cdot 45^4 \equiv 45 \cdot 37^2 \equiv 45 \cdot 20 \equiv 48 \pmod{71}$ and sees that $48 = x = z = 48\checkmark$

Solution of Problem 2

We have the polynomial over $\mathbb{F}_7$

$$q(X) = X^3 + 5.$$

a) The secret is 5.

b) Four pairs $(i, q(i))$, $i \in \mathbb{F}_7$ need to be issued. The candidates are $(1,6)$, $(2,6)$, $(3,4)$, $(4,6)$, $(5,4)$, $(6,4)$.

c) Now we have the four pairs $(1,6)$, $(2,2)$, $(3,5)$, and $(4,0)$, named as (x_i, y_i), $i = 1,\dots,4$. The polynomial over $\mathbb{F}_7$ has the form

$$r(X) = aX^3 + bX^2 + cX + d = (X^3, X^2, X, 1) \cdot (a,b,c,d)^T$$

with a, b, c, $d \in \mathbb{F}_7$ and d is the secret. Moreover, those points must fulfill $y_i = r(x_i)$ which is equivalent to $A \cdot (a,b,c,d)^T = y$, where $y = (y_1, y_2, y_3, y_4) = (6, 2, 5, 0)$ and

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 2 & 1 \\ 6 & 2 & 3 & 1 \\ 1 & 2 & 4 & 1 \end{pmatrix}$$

$$\left(\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 6 \\ 1 & 4 & 2 & 1 & 2 \\ 6 & 2 & 3 & 1 & 5 \\ 1 & 2 & 4 & 1 & 0 \end{array}\right) \longrightarrow \left(\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 6 \\ 0 & 3 & 1 & 0 & 3 \\ 0 & 3 & 4 & 2 & 4 \\ 0 & 1 & 3 & 0 & 1 \end{array}\right) \longrightarrow \left(\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 6 \\ 0 & 1 & 5 & 0 & 1 \\ 0 & 0 & 3 & 2 & 1 \\ 0 & 0 & 4 & 4 & 2 \end{array}\right) \longrightarrow$$

$$\left(\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 6 \\ 0 & 1 & 5 & 0 & 1 \\ 0 & 0 & 1 & 3 & 5 \\ 0 & 0 & 0 & 6 & 3 \end{array}\right) \longrightarrow \left(\begin{array}{cccc|c} 1 & 1 & 1 & 0 & 2 \\ 0 & 1 & 5 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 \end{array}\right) \longrightarrow \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 \end{array}\right)$$

Hence, $r(X) = X^3 + X^2 + 4$ and the secret is $d = 4$.

Solution of Problem 3

a) The binary representation of 45 is 101101.

$$\begin{aligned} 45P &= P + 4P + 8P + 32P \\ &= P + 2^2P + 2^3P + 2^5P \\ &= P + 2 \cdot 2P + 2 \cdot 2 \cdot 2P + 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2P \\ &= P + 2(2(P + 2P) + 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2P \\ &= P + 2(2(P + 2(P + 2 \cdot 2P)))) \end{aligned}$$

The last line corresponds to the representation of Horner's scheme. It also holds.

$$45P = 2(2(2(2(2P + O \cdot P) + 1 \cdot P) + 1 \cdot P) + O \cdot P) + 1 \cdot P$$

b) The iterative algorithm starts with the point $Q = P$. Then it iterates i from $m - 1$ downto 0. It doubles in all iterations Q and adds P if the current bit k_i is one. At the end of the loop it returns the computed point $Q = kP$.

When the iterative algorithm is applied to the given example with $k = 45$, we obtain the following sequence from the for-loop.

$$P, 2P+O\cdot P, 2(2P+O\cdot P)+P, 2(2(2P+O\cdot P)+P)+P, 2(2(2(2P+O\cdot P)+P)+P)+O\cdot P,$$

$$2(2(2(2(2P + O \cdot P) + P) + P) + O \cdot P) + P$$

c) In the recursive algorithm, it calls itself recursively without the last bit.

When the recursive algorithm is applied to the given example with $k = 45$, we obtain $45P = P + 2(2(P + 2(P + 2(2P))))$ which corresponds to the Horner's scheme of $45P$.

Algorithm 1 $f_{\text{it}}(P, k = (k_m, \ldots, k_0)_2)$

```
Q ← P
for i ← m − 1 downto 0 do
    Q ← 2Q              // Double
    if k_i = 1 then     // if i-th bit is 1
        Q ← Q + P       // Add
    end if
end for
return Q
```

Algorithm 2 $f_{\text{rec}}(P, k = (k_m, \ldots, k_0)_2)$

```
if m = 0 then                                       // This implies k = 1
    return P
else
    if k_0 = 0 then
        return 2 · f_rec(P, (k_m, ..., k_1)_2)         // Double
    else
        return P + 2 · f_rec(P, (k_m, ..., k_1)_2))    // Double and Add
    end if
end if
```