Modern Cryptography December 27, 2019

Solutions to Homework 9

Tutoren: Karen Klein, Guillermo Pascual Perez *Due: 23.59 CET, Dec 11, 2019*

To get credit for this homework it must be submitted no later than Wednesday, December 11th via TUWEL. If you have not registered for the tutorial (192.063 Tutorial on Introduction to Modern Cryptography 2019W) on TUWEL, please do so. If you are unable to register for the course on TUWEL for some reason, submit your homework via email to via email to `michael.walter@ist.ac.at`, please use "MC19 Homework 9" as subject.

1. Groups

   - Let $N \in \mathbb{Z}_{>0}$ and let $G = \mathbb{Z}_N^*$. Prove that G is a group under the operation $a \cdot b := (a \cdot b) \bmod N$.

     Solution: We prove that $G = \mathbb{Z}_N^* = \{a \in \mathbb{Z}_N \mid \mathsf{gcd}(a, N) = 1\}$ satisfies the properties of a group. Let $a, b, c \in G$ be arbitrary elements.

     - **Closure**: Since both a and b are coprime to N, by unique prime factorization also $ab \in \mathbb{Z}$ is coprime to N. To prove closure, we have to show that $\mathsf{gcd}(ab \bmod N, N) = 1$. To this aim, recall that for all $k \in \mathbb{Z}$ there exist $X, Y \in \mathbb{Z}$ such that $\mathsf{gcd}(k, N) = X \cdot k + Y \cdot N$ and the greatest commen divisor is the smallest positive integer that can be written in this form. Now, let $X, Y \in \mathbb{Z}$ be such that $\mathsf{gcd}(ab, N) = X \cdot ab + Y \cdot N$. Since $ab = [ab \bmod N] + qN$ for some $q \in \mathbb{Z}$, we get

       $$1 = \mathsf{gcd}(ab, N) = Xab + YN = X([ab \bmod N] + qN) + YN = X \cdot [ab \bmod N] + (q + Y) \cdot N.$$

       This implies $\mathsf{gcd}(ab \bmod N, N) = 1$, i.e., G is closed.
     - **Unit element**: Since $\mathsf{gcd}(1, N) = 1$, we have $1 \in G$ and $1 \cdot a = a = a \cdot 1$ for all $a \in G$.
     - **Inverses**: By definition, for all $a \in G$ we have $\mathsf{gcd}(a, N) = 1$. Thus, there exist $X, Y \in \mathbb{Z}$ such that $Xa + YN = 1$. Writing $X = qN + [X \bmod N]$ for some integer q, we get

       $$\begin{aligned} 1 = Xa + YN &= (qN + [X \bmod N]) \cdot a + YN \\ &= [X \bmod N] \cdot a + (aq + Y)N = [X \bmod N] \cdot a \bmod N. \end{aligned}$$

       Thus, $[X \bmod N]$ is the inverse of a modulo N and obviously $[X \bmod N] \in G$.
     - **Associativity**: For $ab = [ab \bmod N] + qN$ with $q \in \mathbb{Z}$, it holds

       $$[[a \cdot b \bmod N] \cdot c \bmod N] = [(ab - qN) \cdot c \bmod N] = [abc \bmod N]$$

       and similarly for $bc = [bc \bmod N] + pN$ with $p \in \mathbb{Z}$

       $$[a \cdot [b \cdot c \bmod N] \bmod N] = [a \cdot (bc - pN) \bmod N] = [abc \bmod N].$$

     □

- List the elements of $\mathbb{Z}_{17}^*$; What is its order?; What are the orders of 2 and 5?; Is $\mathbb{Z}_{17}^*$ cyclic?

  Solution: Since 17 is a prime, all integers $1 \leq x \leq 16$ are coprime to 17, thus

  $$\mathbb{Z}_{17}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}, \quad \Rightarrow \quad |\mathbb{Z}_{17}^*| = 16.$$

  By Proposition 8.54, the order of any element in a group divides the group order. Hence, all elements in $\mathbb{Z}_{17}^*$ have order 1,2,4, 8, or 16, and one can compute the orders of 2 and 5 as

  $$2^1 = 2,\ 2^2 = 4,\ 2^4 = 16,\ 2^8 = 256 = 1 \bmod 17 \quad \Rightarrow \quad \mathsf{ord}(2) = |\langle 2 \rangle| = 8.$$

  $$5^1 = 5,\ 5^2 = 25 = 8 \bmod 17,\ 5^4 = (5^2)^2 = 64 = 13 \bmod 17,$$
  $$5^8 = (5^4)^2 = (-4)^2 = 16 = -1 \bmod 17 \quad \Rightarrow \quad \mathsf{ord}(5) = |\langle 5 \rangle| = 16.$$

  Since the order of 5 is equal to the order of the group, 5 is a generator mod 17 and, thus, $\mathbb{Z}_{17}^*$ is cyclic. Alternatively, one can apply Theorem 8.56, which states that for any prime p the multiplicative group $\mathbb{Z}_p^*$ is cyclic. □

- Let $\mathbb{G}$ and $\mathbb{H}$ be groups. Show that $\mathbb{G} \times \mathbb{H}$ is a group (with $\times$ being the direct product).

  Solution: Let the groups operations of $\mathbb{G}$ and $\mathbb{H}$ be denoted by $+$ and $\cdot$, respectively. We prove that the direct product $\mathbb{G} \times \mathbb{H}$ with the operation $(g_1, h_1) \circ (g_2, h_2) = (g_1 + g_2, h_1 \cdot h_2)$ satisfies the properties of a group. Let $g_1, g_2, g_3 \in \mathbb{G}$ and $h_1, h_2, h_3 \in \mathbb{H}$ be arbitrary group elements.

  - **Closure**: By closure of $\mathbb{G}$ and $\mathbb{H}$, we have $g_1 + g_2 \in \mathbb{G}$ and $h_1 \cdot h_2 \in \mathbb{H}$. Thus, $(g_1, h_1) \circ (g_2, h_2) = (g_1 + g_2, h_1 \cdot h_2) \in \mathbb{G} \times \mathbb{H}$.
  - **Unit element**: Let 0_G, 1_H be the unit elements in $\mathbb{G}, \mathbb{H}$, respectively. Then $(0_G, 1_H) \in \mathbb{G} \times \mathbb{H}$ is the unit element in $\mathbb{G} \times \mathbb{H}$:

    $$(0_G, 1_H) \circ (g_1, h_1) = (0_G + g_1, 1_H \cdot h_1) = (g_1, h_1) = (g_1 + 0_G, h_1 \cdot 1_H) = (g_1, h_1) \circ (0_G, 1_H).$$

  - **Inverses**: Let $-g_1$, h_1^{-1} be the respective inverses of of g_1 and h_1 in $\mathbb{G}$ and $\mathbb{H}$. Then $(-g_1, h_1^{-1}) \in \mathbb{G} \times \mathbb{H}$ is the inverse of (g_1, h_1):

    $$(-g_1, h_1^{-1}) \circ (g_1, h_1) = (-g_1 + g_1, h_1^{-1} \cdot h_1) = (1_G, 1_H),$$

    $$(g_1, h_1) \circ (-g_1, h_1^{-1}) = (g_1 - g_1, h_1 \cdot h_1^{-1}) = (1_G, 1_H).$$

  - **Associativity**: By associativity of $\mathbb{G}$ and $\mathbb{H}$, the following sequence of equations holds:

    $$\big((g_1, h_1) \circ (g_2, h_2)\big) \circ (g_3, h_3) = (g_1 + g_2, h_1 \cdot h_2) \circ (g_3, h_3)$$
    $$= \big((g_1 + g_2) + g_3, (h_1 \cdot h_2) \cdot h_3\big) = (g_1 + g_2 + g_3, h_1 \cdot h_2 \cdot h_3).$$

    $$(g_1, h_1) \circ \big((g_2, h_2) \circ (g_3, h_3)\big) = (g_1, h_1) \circ (g_2 + g_3, h_2 \cdot h_3)$$
    $$= \big(g_1 + (g_2 + g_3), h_1 \cdot (h_2 \cdot h_3)\big) = (g_1 + g_2 + g_3, h_1 \cdot h_2 \cdot h_3).$$

□

2. Algorithmic aspects

- Compute $101^{4800000002} \mod 35$ (by hand).

  Solution: We have $|\mathbb{Z}_{35}^*| = \varphi(35) = \varphi(5)\varphi(7) = 4 \cdot 6 = 24$. Thus

$$\begin{aligned} 101^{4800000002} \mod 35 &\equiv 101^{4800000002 \bmod 24} \mod 35 \equiv 101^2 \mod 35 \\ &\equiv (-4)^2 \mod 35 \equiv 16 \mod 35 \ . \end{aligned}$$

□

- Use the Extended Euclidean Algorithm (extGCD) to compute X, Y for $a = 2493$ and $b = 8709$. Illustrate all steps.

  Solution: We have

$$\begin{aligned} 8709 &= 3 \cdot 2493 + 1230 \\ 2493 &= 2 \cdot 1230 + 33 \\ 1230 &= 37 \cdot 33 + 9 \\ 33 &= 3 \cdot 9 + 6 \\ 9 &= 1 \cdot 6 + 3 \\ 6 &= 2 \cdot 3 + 0 \end{aligned}$$

  Thus $\gcd(2493, 8709) = \gcd(8709, 2493) = 3$. Further, we obtain

$$\begin{aligned} 3 &= 9 - 1 \cdot 6 \\ &= 9 - 1 \cdot (33 - 3 \cdot 9) = 4 \cdot 9 - 33 \\ &= 4 \cdot (1230 - 37 \cdot 33) - 33 = 4 \cdot 1230 - 149 \cdot 33 \\ &= 4 \cdot 1230 - 149 \cdot (2493 - 2 \cdot 1230) = 302 \cdot 1230 - 149 \cdot 2493 \\ &= 302 \cdot (8709 - 3 \cdot 2493) - 149 \cdot 2493 = 302 \cdot 8709 - 1055 \cdot 2493. \end{aligned}$$

  Thus $X = -1055$ and $Y = 302$. □

- **[B.2 in book, 2nd edition]** Prove that the extGCD runs in time polynomial in the length of its inputs.

  Solution: Let a and b denote the inputs to extGCD and assume w.l.o.g. that $a \geq b$. We show that extGCD runs in time polynomial in $\lceil \log(a) \rceil$ by first bounding the number of recursive calls to extGCD and then analyzing the running time of the computations at each level of the recursion.

  At every level of the recursion the algorithm computes an expression of the form $a_i = q_i b_i + r_i$ and sets $a_{i+1} := b_i$, $b_{i+1} := r_i$. Thus, we always have $b_{i+1} < b_i$. We show that for every i we have $b_{i+2} \leq b_i/2$. Indeed, if $b_{i+1} \leq b_i/2$ then $b_{i+2} < b_{i+1} \leq b_i/2$. If, on the other hand, $b_{i+1} > b_i/2$ then we have

$$b_{i+2} = r_{i+1} = a_{i+1} - q_{i+1} b_{i+1} = b_i - q_{i+1} b_{i+1} = b_i - b_{i+1} < b_i/2,$$

where $q_{i+1} = 1$ follows since $2b_{i+1} > b_i = q_{i+1}b_{i+1} + r_{i+1} > b_{i+1}$, and the last equality again holds since $b_{i+1} > b_i/2$. Hence $b_{i+2} < b_i/2$ for all i. As $b_i > 0$ for all i this implies that there are at most $2\lceil \log(b) \rceil$ recursive calls to extGCD.

At every level of the recursion the algorithm performs one division with remainder and, when updating the values for X and Y, computes one multiplication and one subtraction. Note that a_i, b_i, q_i, and r_i are bounded from above by a. We proceed by computing bounds for X and Y as well. To this end, denote by X_i and Y_i the values assigned to X and Y at the end of the ith recursion call. We show inductively that for every i we have $|X_i| \leq b_i$ and $|Y_i| \leq a_i$. Let k be the index of the last time the recursion is called, i.e., let k such that $a_k = q_k b_k + 0$. Our induction hypothesis holds for $i = k$ since $|X_k| = 0 \leq b_k$ and $|Y_k| = 1 \leq a_k$. Assuming that the induction hypothesis holds for $i+1$ we show that it also holds for i. Indeed;

$$
\begin{aligned}
|X_i| &= |Y_{i+1}| \leq a_{i+1} = b_i \\
|Y_i| &= |X_{i+1} - q_i Y_{i+1}| \leq |X_{i+1}| + q_i \, |Y_{i+1}| \leq b_{i+1} + q_i a_{i+1} = r_i + q_i b_i = a_i,
\end{aligned}
$$

where in the last inequality of each line we used the induction hypothesis.

Summing up, the algorithm performs at most $2\lceil \log(b) \rceil$ divisions with remainder, multiplications, and subtractions of integers of size at most a. Since each of those operations can be computed in time polynomial in $\lceil \log(a) \rceil$ we conclude that extGCD runs time polynomial in the length of its inputs. □