Solutions for week 5, Cryptography Course - TDA 352/DIT 250

In this weekly exercise sheet: you will construct some secret sharing schemes, study hash functions and identification protocols.
Completing the ex. sheet: you will have a good understanding of hash function's theory, know how to build some secret sharing scheme and have some knowledge on identification protocols.

Easy

1. To prove that $g = 6$ is a generator of $\mathbb{Z}_{41}^*$ we start by observing that 41 is prime and so Z_{41}^* is a cyclic group of order $\phi(41) = 41 - 1 = 40$.
   By definition g is a generator if and only if $g^i \neq 1 \pmod{41}$ for every $i \in \{1, \ldots, \phi(41) - 1\}$.
   The negation of this statements tells us that if there exists an exponent $i \in \{1, \cdots, \phi(41) - 1\}$ such that $g^i = 1 \pmod{41}$ then g is not a generator of $\mathbb{Z}_{41}^*$. In this case, g will generate a subgroup $\langle g \rangle$ of $\mathbb{Z}_{41}^*$ and $\text{ord}(\langle g \rangle) | \phi(41) = 40$ (by Lagrange theorem).
   Therefore, we only need to check if there exists a divisor d of $\phi(41)$ such that $g^d = 1 \pmod{41}$.
   The divisors of 40 are $\{1, 2, 4, 5, 8, 10, 20\}$ and so we compute

   $$6^1 = 6 \neq 1 \pmod{41} \qquad 6^2 = 36 \neq 1 \pmod{41} \qquad 6^4 = 25 \neq 1 \pmod{41}$$

   $$6^5 = 27 \neq 1 \pmod{41} \qquad 6^8 = 10 \neq 1 \pmod{41} \qquad 6^{10} = 32 \neq 1 \pmod{41} \qquad 6^{20} = 40 \neq 1 \pmod{41}$$

   The above computations show that $g = 6$ is a generator of $\mathbb{Z}_{41}^*$.

2. **Definition:** A **secret-sharing scheme** usually involves

   - a *dealer* D who has a secret s
   - n *parties* $P_1, \ldots, P_n$

   A secret-sharing scheme is a method by which the dealer distributes shares of s to the n parties such a way that:

   - any subset of $t + 1$ parties can reconstruct the secret from its shares **and**
   - any subset of t parties cannot retrieve any partial information on the secret s

3. - $(t + 1)$**-correctness:** any $t + 1$ parties together can compute the secret s.
   - **privacy:** no single party alone learns knows anything about the secret s.
   - t**-unconditional security:** any subset of t parties cannot recover the secret s, no matter how much computational power the parties have.

4. - **Completeness:** An (interactive) identification protocol is **complete** if an honest prover P succeeds in convincing a *honest* verifier V that a *true* statement is *true*.
   - **Soundness:** An (interactive) identification protocol is **sound** if no *dishonest* prover P succeeds in convincing a *honest* verifier V that a *false* statement is *true*.

5. A Σ-protocol is a protocol that has the following three-move structure:

   (a) the prover P generates a *random looking* value called **commitment** (a witness of P's statement) and sends it to the verifier V
   (b) V replies with a random **challenge** to P
   (c) P performs some computations based on the challenge, the chosen (committed) witness and the secret (connected to the statement). The result is the *response* to V.

Medium

6. Let us consider the Mignotte's SSS with $n = 4$ and $t = 1$. Let $m_1 = 3$, $m_2 = 4$, $m_3 = 5$, $m_4 = 7$. Let the secret be $s = 9$.

   (a) We have that $\gcd(m_i, m_j) = 1$ for every choice of i, j with $i \neq j$. We have that $m_1 < m_2 < m_3 < m_4$.
   We check that $m_1 \cdot m_2 = 3 \cdot 4 = 12 > 7 = m_4$. So the given number are a valid Mignotte's series.

   (b) Let the secret be $s = 9$. The share s_i is computed as $s_i = s \pmod{m_i}$:

   $$s_1 = 9 = 0 \pmod 3 \qquad s_2 = 9 = 1 \pmod 4$$

   $$s_3 = 9 = 4 \pmod 5 \qquad s_4 = 9 = 2 \pmod 7$$

   (c) We start from $s_1 = 0 \pmod 3$ and $s_4 = 2 \pmod 7$.
   In order to reconstruct s, we need the Bezout's identity $7 \cdot (1) + 3 \cdot (-2) = 1$. From the CRT we get

   $$s = 0 \cdot (7 \cdot 1) + 2 \cdot (3 \cdot (-2)) = -12 \pmod{21} = 9 \pmod{21}$$

   (d) We start from $s_1 = 0 \pmod 3, s_3 = 4 \pmod 5$ and $s_4 = 2 \pmod 7$.
   We have, by the previous question, that $s_{1,4} = 9 \pmod{21}$. We now consider the linear system of congruences:

   $$\begin{cases} s_{1,4} = 9 \pmod{21} \\ s_3 = 4 \pmod 5 \end{cases} \qquad \begin{array}{c} \text{Bezout identity} \\ 21 \cdot (1) + 5 \cdot (-4) = 1 \end{array}$$

   and so we obtain

   $$s = 4 \cdot 21 + 9 \cdot (-20) \pmod{105} = -96 \pmod{105} = 9 \pmod{105}$$

7. *You may need a calculator to facilitate the computations.* Let us consider the Mignotte's SSS with $n = 4$ and $t = 2$. Let $m_1 = 6$, $m_2 = 11$, $m_3 = 13$, $m_4 = 19$. Let the secret be $s = 666$.

   (a) Let the secret be $s = 666$. To compute the share s_i, we compute $s_i = s \pmod{m_i}$:

   $$s_1 = 0 \pmod 6 \qquad s_2 = 6 \pmod{11}$$

   $$s_3 = 3 \pmod{13} \qquad s_4 = 1 \pmod{19}$$

   (b) We start from $s_1 = 0 \pmod 6, s_3 = 3 \pmod{13}$ and $s_4 = 1 \pmod{19}$.
   From the Bezout's identity $13 \cdot 3 + 19 \cdot (-2) = 1$, we have

   $$s_{3,4} = 3 \cdot (19 \cdot (-2)) + 1 \cdot (13 \cdot 3) = -75 \pmod{247}$$

   Now, from the Bezout's identity $247 + 6 \cdot (-41) = 1$, we have

   $$s = 0 \cdot 247 - 75 \cdot 6 \cdot (-41) = 18450 = 666 \pmod{1482}$$

8. Let us consider the Shamir SSS with $n = 2$ and $t = 1$. The dealer choose to work in $\mathbb{Z}_3$. The secret is $s = 1$ and the polynomial that he randomly generate is $f(x) = 1 + 2x \in \mathbb{Z}_3[x]$

   (a) To compute the shares, the dealer computes $s_i = f(i)$ and so obtains

   $$s_1 = f(1) = 1 + 2 = 0 \pmod 3 \qquad s_2 = f(2) = 1 + 4 = 2 \pmod 3$$

   (b) We have $s_1 = f(1) = 1 + 2 = 0 \pmod 3$ and $s_2 = f(2) = 1 + 4 = 2 \pmod 3$.
   The Lagrange interpolation coefficients are

   $$\delta_1^{1,2} = 2 \cdot (2-1)^{-1} = 2 \cdot 1^{-1} = 2 \qquad \delta_2^{1,2} = 1 \cdot (1-2)^{-1} = 1 \cdot (-1)^{-1} = -1 = 2 \pmod 3$$

   since $(-1)^2 = 1 \pmod 3$.
   So we can compute

   $$s = s_1 \delta_1^{1,2} + s_2 \delta_2^{1,2} = 0 \cdot 2 + 2 \cdot 2 = 1 \pmod 3$$

9. Let us consider the Shamir SSS with $n = 4$ and $t = 2$. The dealer chooses to work in $\mathbb{Z}_7$. The secret is $s = 1$ and the polynomial that he randomly generates is $f(x) = 1 + 3x + 6x^2$

   (a) To compute the shares, the dealer computes $s_i = f(i)$ and so obtains

$$s_1 = f(1) = 1 + 3 + 6 = 3 \pmod 7 \qquad s_2 = f(2) = 1 + 6 + 24 = 3 \pmod 7$$

$$s_3 = f(3) = 1 + 9 + 54 = 1 \pmod 7 \qquad s_4 = f(4) = 1 + 12 + 6 \cdot 2 = 4 \pmod 7$$

   (b) We have $s_1 = 3 \pmod 7$, $s_2 = 3 \pmod 7$ and $s_3 = 1 \pmod 7$.
   The Lagrange interpolation coefficients are

$$\delta_1^{1,2,3} = \left(2 \cdot (2-1)^{-1}\right)\left(3 \cdot (3-1)^{-1}\right) = 2 \cdot 1^{-1} \cdot 3 \cdot 2^{-1}$$

   to compute 2^{-1}, we use the extended Euclidean algorithm and obtain that $2^{-1} = 4 \pmod 7$

$$\delta_1^{1,2,3} = \left(2 \cdot (2-1)^{-1}\right)\left(3 \cdot (3-1)^{-1}\right) = 2 \cdot 1^{-1} \cdot 3 \cdot 2^{-1} = 2 \cdot 3 \cdot 4 = 3 \pmod 7$$

$$\delta_2^{1,2,3} = \left(1 \cdot (1-2)^{-1}\right)\left(3 \cdot (3-2)^{-1}\right) = 1 \cdot (-1)^{-1} \cdot 3 \cdot 1^{-1} = 1 \cdot (-1) \cdot 3 = -3 = 4 \pmod 7$$

$$\delta_3^{1,2,3} = \left(1 \cdot (1-3)^{-1}\right)\left(2 \cdot (2-3)^{-1}\right) = 1 \cdot (-2)^{-1} \cdot 2 \cdot (-1)^{-1}$$

   since $2^{-1} = 4$, the inverse of -2 is $(-2)^{-1} = (-1)^{-1}(2)^{-1} = (-1) \cdot 4 = -4 = 3 \pmod 7$

$$\delta_3^{1,2,3} = \left(1 \cdot (1-3)^{-1}\right)\left(2 \cdot (2-3)^{-1}\right) = 1 \cdot (-2)^{-1} \cdot 2 \cdot (-1)^{-1} = 3 \cdot 2 \cdot (-1) = -6 = 1 \pmod 7$$

   Finally, we have:

$$s = s_1\delta_1^{1,2,3} + s_2\delta_2^{1,2,3} + s_3\delta_3^{1,2,3} = 3 \cdot 3 + 3 \cdot 4 + 1 \cdot 1 = 2 + 5 + 1 = 1 \pmod 7$$

10. Victor's transcript will consist of a sequence of three-message rounds of the form

$$\begin{array}{lcl} P \to V. & : & R_1 \\ V \to P. & : & b_1 \\ P \to V. & : & z_1 \\ P \to V. & : & R_2 \\ V \to P. & : & b_2 \\ P \to V. & : & z_2 \\ & \dots & \end{array}$$

   When $b_k = 0$, Victor checked $z_k^2 = R_k$ and when $b_k = 1$, he checked $z_k^2 = R_k \cdot X$. Since the check succeeded a number of times with random choices of b_k, Victor became convinced that Peggy knows x.

   But the transcript does not convince you, since Victor could have produced this transcript without interacting with Peggy at all. He just chooses in each round both b_k and z_k at random and then sets $R_k = z_k^2$ if $b_k = 0$ and $R_k = z_k^2 \cdot X^{-1}$ if $b_k = 1$.

Hard

11. Let us consider a Secure Multi Party Computation (SMPC) protocol for addition between 2 parties. Every party will use a Shamir SSS with $n = 2$ and $t = 1$. The parties decide to work in $\mathbb{Z}_5$. The secrets are $s_1 = 1$ and $s_2 = 2$ and they want to compute the sum of the two values. The polynomials that they randomly generate are $f_1(x) = 1 + 3x$ for P_1 and $f_2(x) = 2 + x$ for P_2.

(a) The shares are computed with $s_{i,j} = f_i(j) \pmod 5$ and so we obtain

$$s_{1,1} = f_1(1) = 4 \qquad s_{1,2} = f_1(2) = 2$$

$$s_{2,1} = f_2(1) = 3 \qquad s_{2,2} = f_2(2) = 4$$

(b) The partial results are

$$a_1 = s_{1,1} + s_{2,1} = 4 + 3 = 2 \qquad a_2 = s_{1,2} + s_{2,2} = 2 + 4 = 1$$

(c) The Lagrange interpolation coefficients are

$$\delta_1^{1,2} = 2 \cdot (2-1)^{-1} = 2 \cdot (1)^{-1} = 2 \qquad \delta_2^{1,2} = 1 \cdot (1-2)^{-1} = 1 \cdot (-1)^{-1} = 4 \pmod 5$$

where the inverse of -1 modulus 5 is -1 since $(-1)^2 = 1$. The final result is

$$a_1\delta_1^{1,2} + a_2\delta_2^{1,2} = 2 \cdot 2 + 1 \cdot 4 = 3 \pmod 5 = s_1 + s_2$$

12. (a) If her received response is c, she computes $r \oplus c$ and checks that she gets k. If the receiver does know k and follows the protocol, $c = r \oplus k$ and Alice's computation will be $r \oplus (r \oplus k) = k$.

    (b) No. An eavesdropping adversary that hears a protocol run can do the same computation as Alice and recover k.

Think

13. (a) B has received $M \oplus N_A$ in message 1 and $M \oplus N_A \oplus N_B \oplus N_A$ in message 3. The latter can be simplified to $M \oplus N_B$. Thus B can recover M by xor-ing the content of message 3 with his own nonce N_B.

    (b) No. An evesdropper can compute $M_1 \oplus M_2 = N_B$; he then has the same knowledge as B and can recover M in the same way.