Exercises on Secret Sharing and SMC

Patrick Ah-Fat

1 Secret Sharing

Exercise 1 *Let $\mathcal{P} = \{A, B, C, D, E\}$ be a set of parties.*

1. *Compute the following monotone access structures Γ given their minimal elements $m(\Gamma)$:*
   (a) $m(\Gamma) = \{\{A, B, C\}\}$
   (b) $m(\Gamma) = \{\{A, B, C\}, \{A, B, D\}\}$
   (c) $m(\Gamma) = \{\{A, B, C\}, \{B, C, D\}, \{C, D, E\}\}$
   (d) $m(\Gamma) = \{\{A, B, C\}, \{A, D\}\}$
   (e) $m(\Gamma) = \{\{A, B, C, D, E\}\}$
2. *Compute the minimal elements of the following monotone access structures:*
   (a) $\Gamma = \{\{A, B, C, E\}, \{B, C, D, E\}, \{A, B, C, D, E\}\}$
   (b) $\Gamma = \{\{A, B, D, E\}, \{A, B, C, D, E\}, \{B, D, E\}, \{B, C, D, E\}\}$
   (c) $\Gamma = \{\{A, C, E\}, \{A, B, C, D\}, \{A, C, D, E\}, \{A, B, C, D\}, \{A, B, C, E\}, \{A, B, C, D, E\}, \{B, C, D\}, \{B, C, E\}, \{B, C, D, E\}, \{C, E\}, \{C, D, E\}\}$
   (d) $\Gamma = \{\{A, B, C, D\}, \{A, B, C, E\}, \{A, C, D, E\}, \{B, C, D, E\}, \{A, B, C, D, E\}, \{B, C, D\}, \{B, C, E\}, \{C, D, E\}\}$

Exercise 2 *Let s be in $\mathbb{Z}_{11}$. The value s has been honestly secretly shared amongst 5 participants with Shamir secret sharing scheme in order to allow up to $t = 2$ passive adversaries. The second and fourth shares have been lost, and the first, third and fifth shares are respectively equal to 6, 2 and 2.*

1. *Reconstruct the secret.*
2. *Same question with the first, third and fifth shares being respectively equal to 10, 8 and 9.*

Exercise 3 *In order to share a secret s in the presence of up to t passive adversaries, Shamir secret sharing scheme requires the dealer to:*

1. *pick a polynomial* f ***of degree at most*** t *such that* $f(0) = s$

2. *send* $f(k)$ *to each party* P_k

Explain what security concern would arise if step 1 *of the protocol was replaced by:*

1. *pick a polynomial* f ***of degree*** t *such that* $f(0) = s$

Hint*: In order to show that a scheme is not information-theoretically secure, it suffices to show one case where it fails to guarantee perfect secrecy.*

Exercise 4 *Let us consider four parties* A, B, C *and* D. *Let* s *be a secret. Shamir secret sharing scheme enables us to distribute secret* s *via* $[s, f]_t$ *while allowing up to* t *adversaries. However, sending more than one point of polynomial* f *to some parties may help us to achieve more general monotone access structures.*

1. *Based on this idea, propose a scheme that allows* s *to be shared under the following monotone access structures defined by their minimal elements:*

   (a) $m(\Gamma_1) = \{\{A, D\}, \{B, D\}, \{C, D\}\}$

   (b) $m(\Gamma_2) = \{\{A, D\}, \{B, D\}, \{C, D\}, \{A, B, C\}\}$

   (c) $m(\Gamma_3) = \{\{B, D\}, \{C, D\}, \{A, B, C\}\}$

2. *Consider:*

$$m(\Gamma_4) = \{\{A, D\}, \{B, C\}, \{C, D\}\}$$

   (a) *Prove that a secret shared in the sense of Shamir using a polynomial* f*, where each point of the polynomial is held by at most one party cannot satisfy this monotone access structure* Γ_4.

   (b) *Based on that observation, propose a simple solution that allows a secret to be shared with respect to* Γ_4.

2 Secure Multi-Party Computation

Exercise 5 *Let us consider* 3 *parties* P_1, P_2 *and* P_3*. Let us place ourselves in* $\mathbb{Z}_{11}$*. Let us assume that they secretly share two secrets* $[a = 4, f_a = 4 + 3X]_1$ *and* $[b = 9, f_b = 9 + 2X]_1$*.*

1. *Compute the shares that each parties hold.*
2. *Perform the local computations that enable them to secretly share* $a + b$ *and compute the corresponding polynomial* $f_a + f_b$*.*
3. *Now, perform the computation that parties* P_2 *and* P_3 *should follow to recover* $a + b$*.*
4. *Show that recombining* $a + b$ *using shares of the three parties would yield the same result.*
5. *Now assume that the parties wish to secretly share* $a \cdot b$*. Show how they can achieve that using only local computations, and explicitly compute the underlying polynomial* P*.*
6. *Show the computation that all the parties together should follow to recover* $a \cdot b$*.*
7. *Show parties* P_2 *and* P_3 *would fail in recovering* $a \cdot b$ *if they tried to use the recombination vector from Question 3.*
8. *Parties* P_1, P_2 *and* P_3 *now respectively decide to generate the following polynomials* g_1, g_2 *and* g_3 *and distribute* $[0, g_1 = 6X]_1, [9, g_2 = 9 + X]_1$ *and* $[8, g_3 = 8 + 3X]_1$*. In other words, they distribute* $[0, 6, 1, 7]_1, [9, 10, 0, 1]_1$ *and* $[8, 0, 3, 6]_1$

   Show that they can now perform local computations in order to share the product $a \cdot b$ *via a polynomial of degree at most* 1*, and explicitly compute this polynomial* Q*.*

Exercise 6 (harder) *Let us study the influence that an active attacker may have on the SMC multiplication protocol robust against passive adversaries. We recall that the protocol assumes that* $[a, f_a]_t$ *and* $[b, f_b]_t$ *are shared and that the parties can easily compute a recombination vector* r *which ensures that* $\sum_k r_k g(k) = g(0)$ *for any polynomial of degree at most* $2t$*. Then:*

1. *The parties locally multiply their shares to get* $[ab, f_a f_b]_{2t}$*.*
2. *Each party* P_k *generates* g_k *and distributes* $[(f_a f_b)(k), g_k]_t$*.*
3. *The parties locally compute* $\sum_k r_k[(f_a f_b)(k), g_k]$ *to get* $[ab, \sum_k r_k g_k]_t$*.*

Importantly, we note that the scheme holds since $\sum_k r_k (f_a f_b)(k) = (f_a f_b)(0) = ab$ *by definition of* r*.*

1. *Assume that P_1 is an active attacker. Explain what she can do in step 2 so that performing step 3 would lead the parties to secretly share $ab + 1$ instead of ab.*

2. *Assume that the parties are computing $(ab)c$ by starting with the elementary operation (ab). Assume that a is held by P_1 and that b and c are private inputs held by honest parties. Based on the previous question, explain what P_1 can do so as to learn the value of private input c when the parties reconstruct the intended output $(ab)c$.*

Solutions

Solution of Exercise 1:

1. (a) $\Gamma = \{\{A, B, C\}, \{A, B, C, D\}, \{A, B, C, E\}, \{A, B, C, D, E\}\}$
   (b) $\Gamma = \{\{A, B, C\}, \{A, B, C, D\}, \{A, B, C, E\}, \{A, B, C, D, E\}, \{A, B, D\}, \{A, B, D, E\}\}$
   (c) $\Gamma = \{\{A, B, C\}, \{A, B, C, D\}, \{A, B, C, E\}, \{A, B, C, D, E\}, \{A, C, D, E\}, \{B, C, D\}, \{B, C, D, E\}, \{C, D, E\}\}$
   (d) $\Gamma = \{\{A, B, C\}, \{A, B, C, D\}, \{A, B, C, E\}, \{A, B, C, D, E\}, \{A, D\} \{A, B, D\}, \{A, C, D\}, \{A, D, E\}, \{A, B, D, E\}, \{A, C, D, E\}\}$
   (e) $\Gamma = \{\{A, B, C, D, E\}\}$

2. (a) $m(\Gamma) = \{\{A, B, C, E\}, \{B, C, D, E\}\}$
   (b) $m(\Gamma) = \{\{B, D, E\}\}$
   (c) $m(\Gamma) = \{\{B, C, D\}, \{C, E\}\}$
   (d) $m(\Gamma) = \{\{B, C, D\}, \{B, C, E\}, \{C, D, E\}\}$

Solution of Exercise 2:

1. Let $Z = \{1, 3, 5\}$ be the set of parties who wish to reconstruct the secret. Let r be its associated recombination vector. We recall that for all k in Z, we have:

$$r_k = \prod_{\substack{j \in Z \\ j \neq k}} \frac{-j}{k - j}$$

Thus, we have:

$$r_1 = \prod_{\substack{j \in Z \\ j \neq 1}} \frac{-j}{1 - j} = \frac{-3}{1 - 3} \cdot \frac{-5}{1 - 5} = \frac{3 \cdot 5}{2 \cdot 4} = \frac{4}{2 \cdot 4} = 2^{-1} = 6$$

$$r_3 = \prod_{\substack{j \in Z \\ j \neq 3}} \frac{-j}{3 - j} = \frac{-1}{3 - 1} \cdot \frac{-5}{3 - 5} = \frac{5}{-4} = -5 \cdot 4^{-1} = -5 \cdot 3 = -4 = 7$$

$$r_5 = \prod_{\substack{j \in Z \\ j \neq 5}} \frac{-j}{5-j} = \frac{-1}{5-1} \cdot \frac{-3}{5-3} = \frac{3}{4 \cdot 2} = 3 \cdot 8^{-1} = 3 \cdot 7 = 10$$

We know that the polynomial P that has been used to distribute the secret is of degree at most $t = 2$. But we have $|Z| = 3$, and $3 > 2$ so by definition of the recombination vector r, we have:

$$P(0) = \sum_{k \in Z} r_k \cdot P(k) = 6 * 6 + 7 * 2 + 10 * 2 = 4$$

and thus $s = 4$.

2. Similarly, we have:

$$P(0) = \sum_{k \in Z} r_k \cdot P(k) = 6 * 10 + 7 * 8 + 10 * 9 = 8$$

and thus $s = 8$.

Solution of Exercise 3: Let us take an example. Let us place ourselves in $\mathbb{Z}_{11}$ again. Let $t = 2$ and let us assume that parties 1 and 2 are passive adversaries. Let us imagine that the secret is $s = 3$ and that the dealer chose polynomial $P = 3 + 7X + 6X^2$ to share his secret.

The shares of parties 1 and 2 would respectively be 5 and 8. Now, if the degree of P is known to be equal to $t = 2$, we claim that the attackers will know that the secret cannot be equal to 2, which is an obvious security concern.

Indeed, we observe that polynomial $Q = 2 + 3X$ passes by the three points $(0, 2)$, $(1, 5)$ and $(2, 8)$. Moreover, the Fundamental Theorem of Algebra ensures that only 1 polynomial of degree at most 2 passes by those three points. So if the secret s was 2, P would equal Q, which is not possible since their degrees are not equal.

Solution of Exercise 4: In the first case, we can notice that party D is "more important" than the other parties. The idea is to allow the dealer to send a different number of shares to different parties, and intuitively, to send more shares to more important parties. The degree t of the polynomial used also has to be controlled accordingly, such that any $t + 1$ different shares enable the secret to be recovered.

1. (a) Instead of sending one share to each of the parties, the dealer will send more shares to party D. Let f be a polynomial of degree at most 3 such that $f(0) = s$. The dealer sends:
   - $f(1)$ to party A
   - $f(2)$ to party B
   - $f(3)$ to party C
   - $f(4)$, $f(5)$ and $f(6)$ to party D

That satisfies the expected Γ_1.

An optimisation requiring less shares to be created would be to let the dealer pick a polynomial g of degree at most 1 such that $g(0) = s$. He would then send $g(1)$ to A, B and C, and he would send $g(2)$ to party D, which would achieve a similar result.

(b) This result can be achieved by letting the dealer picking a polynomial f of degree at most 2 and to send:

- $f(1)$ to party A
- $f(2)$ to party B
- $f(3)$ to party C
- $f(4)$ and $f(5)$ to party D

(c) Finally, the dealer can choose a polynomial of degree at most 4 and send:

- $f(1)$ to party A
- $f(2)$ and $f(3)$ to party B
- $f(4)$ and $f(5)$ to party C
- $f(6)$, $f(7)$ and $f(8)$ to party D

2. (a) The idea here is to notice that parties A and B play a symmetrical role, and that so do C and D. Thus, if we decide to share a secret by selecting a polynomial f and by sending a single point of this polynomial to at most one party, then the only qualitative way of differentiating the parties is to count the number of shares that they hold. Moreover, we know that A and D should be able to recover the secret, and that B play the "same" role as A, so B and D would intuitively also be able to recover the secret, which is a contradiction since $\{B, D\}$ is not part of the access structure. We formalise this result next.

Let a, b, c and d the number of shares that parties A, B, C and D receive respectively. Let t be the degree of the polynomial created to share secret s. The expression of $m(\Gamma_4)$ informs us in particular that $\{A, D\} \in \Gamma_4$ and that $\{B, D\} \notin \Gamma_4$, which we express as:

$$\begin{aligned} a + d &> t \\ b + d &\leq t \end{aligned}$$

which implies that $a > b$. Similarly, we have $\{B, C\} \in \Gamma_4$ and $\{A, C\} \notin \Gamma_4$, which means that:

$$\begin{aligned} b + c &> t \\ a + c &\leq t \end{aligned}$$

which implies that $b > a$, which is a contradiction. We thus cannot respect the access structure Γ_4 by sharing a secret in this way.

(b) However, a simple trick that solves the problem is to let the dealer send the same point of the polynomial f to different parties in order to accommodate Γ_4. The dealer can pick a polynomial of degree at most 2, and send:

- $f(1)$ to party A
- $f(2)$ to party B
- $f(1)$ and $f(3)$ to party C
- $f(2)$ and $f(4)$ to party D

which should do the job.

Solution of Exercise 5:

1. The parties hold $[a = 4, f_a = 4 + 3X]_1 = [4, 7, 10, 2]_1$ and $[b = 9, f_b = 9 + 2X = [9, 0, 2, 4]_1$

2. The parties add their shares to get $[a + b, f_a + f_b]_1 = [2, 7, 1, 6]_1 = [2, 2 + 5X]_1$.

3. Let $S = \{2, 3\}$. Parties P_2 and P_3 are enough to recover $a+b$ since $|S| > 1$. Let them compute the corresponding recombination vector (r_2, r_3):

$$r_2 = \frac{-3}{2-3} = 3$$
$$r_3 = \frac{-2}{3-2} = -2 = 9$$

   The parties now compute $a + b = 3 \cdot 1 + 9 \cdot 6 = -1 + 3 = 2$.

4. Let $S = \{1, 2, 3\}$. We now have:

$$r_1 = \frac{-2}{1-2} \cdot \frac{-3}{1-3} = 3$$
$$r_2 = \frac{-1}{2-1} \cdot \frac{-3}{2-3} = 8$$
$$r_3 = \frac{-1}{3-1} \cdot \frac{-2}{3-2} = 1$$

   The parties now compute $a + b = 3 \cdot 7 + 8 \cdot 1 + 1 \cdot 6 = -1 + 8 + 6 = 2$.

5. The parties can locally multiply their shares to get $[a \cdot b = 3, P = f_a \cdot f_b = 3 + 2X + 6X^2]_2 = [3, 0, 9, 8]_2$.

6. Using the recombination vector from Question 4, the parties would recover $a \cdot b = 3 \cdot 0 + 8 \cdot 9 + 1 \cdot 8 = 3$.

7. Using the recombination vector from Question 3, parties P_2 and P_3 would get $3 \cdot 9 + 9 \cdot 8 = 0$, which does not equal $a \cdot b$.

8. The parties should use the recombination vector from Question 4. Each party P_k should locally compute $\sum_{j=1}^{3} r_j g_j(k)$ so as to secretly share a polynomial of degree at most 1 that equals $a \cdot b$ in 0. They locally compute:

$$3\dot{[}0, 6, 1, 7]_1, +8 \cdot [9, 10, 0, 1]_1 + 1 \cdot [8, 0, 3, 6]_1 = [3, 10, 6, 2]_1$$

The underlying polynomial is:

$$\sum_{j=1}^{3} r_j g_j = 3 \cdot (6X) + 8 \cdot (9 + X) + 1 \cdot (8 + 3X) = 3 + 7X$$

In conclusion, the parties now share $a \cdot b$ with a polynomial of degree at most 1. We have performed the SMC multiplication protocol.

Solution of Exercise 6:

1. In step 2, instead of distributing $(f_a f_b)(1)$, P_1 can distribute $(f_a f_b)(1) + r_1^{-1}$, i.e. she generates a polynomial g_1 such that $g_1(0) = (f_a f_b)(1) + r_1^{-1}$. Step 3 would lead the parties to share $\sum_k r_k (f_a f_b)(k) + r_1 \cdot r_1^{-1} = ab + 1$.

2. As an active attacker, P_1 could be deceitful[1] and select $a = 0$ intentionally. She would then follow the strategy described in Question 1 so as to let the parties share the value $(ab + 1)$, which would equal 1 as $a = 0$. Behaving honestly during the rest of the protocol, i.e. during the multiplication of $(ab + 1)$ by c, the parties would end up in sharing and opening the value $(ab + 1)c = 1 \cdot c = c$ which would leak the value of private input c.

[1] Ah-Fat, P., & Huth, M. (2017, April). Secure multi-party computation: Information flow of outputs and game theory. In International Conference on Principles of Security and Trust (pp. 71-92). Springer, Berlin, Heidelberg.