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 $(ab)c$.*