Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2021

Prof. R. Wattenhofer
Ard Kastrati

Computational Thinking
Solutions to Exercise 6 (Cryptography)

1 Zero Knowledge Proofs in Geometry

a) The constructions are simple and we show here for example how to bisect an angle. First we open the compass in an arbitrary angle and draw a circle around the endpoint of the angle. We label the intersection points of the circle and the angle as A and B. We draw a circle around the point of A and B. And we construct a line between the endpoint of the angle and the intersection of the (newly constructed) circles [1].

b) The following example is one of the possible protocols:

ZKP in Geometry

Peggy		**Vic**
knows $\alpha, \beta = 3\alpha$		knows β
create random angle γ		
construct $\tau = 3\gamma$	send over τ $\longrightarrow$	
	$\longleftarrow$ send over c	choose randomly $c \in \{0, 1\}$
create $\rho = \gamma + c\alpha$	send over ρ $\longrightarrow$	check $3\rho \stackrel{?}{=} \tau + c\beta$

- **Completeness.** One can easily see that if Peggy is honest and knows α, Vic always accepts. More concretely, in the last step $3\rho = 3(\gamma + c\alpha) = 3\gamma + 3c\alpha = \tau + c\beta$.
- **Soundness.** We show that if Peggy can answer both challenges then she really knows α. Assume Peggy can answer for both challenges $c = 0$ and $c' = 1$ correctly with $\rho = \gamma + 0 * \alpha = \gamma$ and $\rho' = \gamma + \alpha$. Then it follows that Peggy can compute $\alpha = \rho' - \rho$. In other words, if she doesn't know α she can at most answer one of the challenges, and fail at the other challenge. That is, Peggy can correctly answer in one round only with probability $1/2$, and therefore n rounds only with probability $1/2^n$.
- **Zero Knowledge.** The main idea is to show that the same[2] transcript that Victor has after the protocol could be generated by himself (without knowing α). During the protocol the transcript contains the triples (τ, c, ρ) and can be produced as follows. For each challenge c, generate a random ρ and construct $\tau = 3\rho - c\beta$. To show zero-knowledge in general we need to show that the transcript can be generated for any strategy V'. We can use the same trick as in the lecture notes. Each round V guesses what the challenge will be and if he does not guess correctly he can simply remove that part of the transcript and try again.

[1] You may have a look at the following video about these constructions and the impossibility of trisecting an angle, if desired: `https://youtu.be/O1sPvUr0YC0`

[2] With the same distribution, in case of random values.

Note. The soundness and zero knowledge property may sound contradictory to each other but they are not. Even though the transcript has "no information about α" after the protocol, Victor is convinced that Peggy knows α because she was able to answer to the challenges *during* the protocol. In particular, first τ is constructed and *only then* based on the challenge c, ρ is constructed.

2 MPC with Secret Sharing

a) One can easily see that each party after summing locally holds a share of the polynomial $f(x) = f_1(x) + f_2(x)$ (since the degree of the polynomial by summing doesn't change and t points uniquely define a polynomial of degree $t-1$). It follows, $s = f(0) = f_1(0) + f_2(0) = s_1 + s_2$. Hence, if the polynomial is reconstructed and evaluated at point 0, it will result to the sum of s_1 and s_2.

b) Dave can just continue the same way as other participants:

Alice	**Bob**	**Carol**	**Dave**
Choose r			

$m_1 = a + r$ (Alice $\longrightarrow$ Bob)

$\mathrm{m}_2 = b + m_1$ (Bob $\longrightarrow$ Carol)

$\mathrm{m}_3 = c + m_2$ (Carol $\longrightarrow$ Dave)

$m_4 = d + m_3$ (Dave $\longrightarrow$ Alice)

Send $s = m_4 - r$ to everyone (Alice $\longrightarrow$ everyone)

c) Alice and Carol can easily compute the salary of Bob as follows. Since Carol has $m_2 = b + m_1$ and Alice has m_1 they can compute $b = m_2 - m_1$. The same technique can be used for Dave as well, namely $d = m_4 - m_3$.

d) The main idea is to use secret sharing and the fact that they are linear. Each party shares their salary by using (n, n) Shamir secret sharing, compute locally the sum of each share (namely the share of each salary), and in the end reconstruct the sum only.

More concretely, initially each party has a (public) number, for example: Alice has 1, Bob has 2, Carol has 3 and Dave has 4.

(a) Sharing Phase:
Alice has her salary s_1. She wants to share this value. She chooses a random polynomial $s_A(x) = s_1 + a_1x + a_2x^2 + a_3x^3$. She sends to herself $s_A(1)$, to Bob $s_A(2)$, to Carol $s_A(3)$, and to Dave $s_A(4)$. Bob has his salary s_2. He wants to share this value. He chooses a random polynomial $s_B(x) = s_2 + b_1x + b_2x^2 + b_3x^3$. He sends to Alice $s_B(1)$, to himself $s_B(2)$, to Carol $s_B(3)$, and to Dave $s_B(4)$. The same way works for Carol and Dave. This process is sharing. That is, when we reach here, every participant has 4 shares (one for each salary).

(b) Local Summation Phase:
Now each participant sums locally those shares (For example Alice adds $sum_A = s_A(1) + s_B(1) + s_C(1) + s_D(1)$).

(c) Reconstruction Phase:
Now every one broadcasts to everyone what the summation of the shares is. For example, Alice sends to everyone sum_A publicly. So, does Bob, Carol and Dave. With those values $(sum_A, sum_B, sum_C, sum_D)$ everyone can reconstruct the polynomial (by Lagrange) and evaluate the polynomial at $x = 0$. That value, is the sum of the salaries.