# Question 7

Recall the Beaver triple protocol for computing $xy$ given $[x], [y]$:

1. A trusted third party generates random $a, b \xleftarrow{\$} \mathbb{Z}_n^*$ and compute $c = ab$
2. This third party distributes $[a], [b], [c]$ to the computation parties. We claim that since the circuit of the computation is known ahead of time, the trusted parties can generate and distributes $[a], [b], [c]$ before the MPC actually starts. Therefore, distributing $[a], [b], [c]$ does not count toward online communication cost during the MPC
3. Each party computes $[x - a], [y - b]$, broadcasts their shares, which publicizes $\epsilon = x - a, \delta = y - b$. **This invokes 2 units of online communication**
4. Each party computes $[z] = \delta[x] + \epsilon[y] - \epsilon\delta + [c]$ which evaluates to $[xy]$
5. Each party opens $[z]$, which assembles back into $xy$

When computing the square of a number, we don't need to obfuscate two distinct operands $x, y$; instead, we only need to obfuscate one operand $x$, so we only need to generate a single one-time pad $a$ and its square $c = a^2$. This means that individual parties only need to broadcast $[x - a]$ during their computation, bring the units of online communication during MPC from 2 to 1.

Here is the full protocol

1. A trusted third party generates random $a \xleftarrow{\$} \mathbb{Z}_n^*$ and computes $c = a^2$
2. The shares $[a], [c]$ are pre-distributed to the computation parties.
3. Each party computes $[x - a]$ and boardcasts the value of their share, which publicizes the value of $x - a$. **This invokes 1 unit of online communication**.
4. Each party computes $[z] = 2 \cdot (x - a)[x] - (x - a)^2 + [c]$, which evaluates to $[x^2]$
5. Each party opens $[z]$, which assembles back into $x^2$

Proof of step 4:

$$
\begin{aligned}
[z] &= 2 \cdot (x - a)[x] - (x - a)^2 + [c] \\
&= [2x^2 - 2ax] - (x^2 - 2ax + a^2) + [a^2] \\
&= [2x^2 - 2ax - (x^2 - 2ax + a^2)] + [a^2] \\
&= [x^2 - a^2] + [a^2] \\
&= [x^2]
\end{aligned}
$$