ASCrypto 2025: MPC and ZKP

September 29-30 of 2025

Author Emanuel Nicolás Herrador
Speaker Sophia Yakoubov

1 Example: Prove sudoku solvability

We can see it with an example if Alice wants to prove Dani that she knows the solution of a Sudoku problem.

Here, we're focused in completeness (if the statement is true, Dani accept), soundness (if the statement is false, Dani rejects even if Alice cheats) and zero knowledge (Dani learns nothing other than the fact that the statement is true). The way to formalize ZK is via the existence of a simulator: $\forall$ PPT $D^*, \exists$ PPT S : VIEW$(D^*) \equiv S(x)$ where PPT is probabilistic polynomial time and D^* is malicious.

First try The way to solve it is doing a permutation of the numbers in the sudoku and then send it to Dani. He should check the constraints and if the permutation is okay. Therefore, we get completeness. For soundness, it holds because if Alice could fool Dani then she could solve the sudoku and then the existence of a solution for this sudoku instance holds. However, we can't have ZK because Dani can very much reverse engineer Alice solution.

Second try Alice can send only an specific row with permutated numbers. Here, we've completeness and ZK, but not soundness obviously.

Third try Now, Alice starts sending the sudoku empty and Dani can ask to open a constrain i. Then, Alice shows the sudoku with this constrain (and the other numbers blocked). Therefore, here we've completeness and ZK. We can create a simulator and it works with probability $\frac{1}{28}$ where 28 is the total of constraints. But it doesn't have soundness because exists a chance where Alice cheats and answer correctly to Dani. If Alice cheated, she might get away with it with probability less or equal to $\frac{27}{28}$.

The solution for that is repeating it k times such that $\left(\frac{27}{28}\right)^k$ is small enough. With that in mind, we can solve soundness problem without getting away ZK.

Remark. The constraints are revealing a column, row, square or the initial sudoku conditions. Each one is made with different permutations, so Dani doesn't learns anything.

How to do this online? Here we can use commitments. The properties here are hiding (commit reveals nothing about what's inside without the key) and binding (commit can only be opened to one thing).

Each property can be perfect (unbreakable even with unlimited resources) or computational (reliant on the hardness of some problem).

To construct it, Alice sends the full sudoku with commitments and when Dani ask for constraint i, Alice answer with the key for this constraint letting Dani open it. Here, we get completeness, soundness (from binding) and ZK (from hiding).

In reality In practice we want to prove thinks like identity or possession of credentials, correct computation, or more generally knowledge/existence of w such that $R(x, w) = 1$. A way to solve it is transform a problem to a sudoku that can be solved like that. However, the sudoku will be bigger and then it will be inefficient.

2 Multi-Party Computation

First example Suppose Alice and Dani picks each one a random number between 1 and 10. We want privacy, i.e., if $x_A \neq x_D$ that is all they learn. A way to do that is closing eyes and open it in the corresponding x-th second. If they opens it at the same time, then they've the same number. Otherwise, not.

If we've more people, we want correctness and t-privacy (the combined views of t or fewer participants reveal nothing other than y).

3 ZKP from MPC

Attempt 1 We want to use a MPC protocol that garantice privacy between the two parties. So the communication complexity here is $\mathrm{poly}(k\,|R|)$ but using lightweight tools (commitments) reducing to sudoku. However, if we run 2PC, the complexity is better $O(k\,|R|)$ but with heavyweight tools (like public key operations).

We want to construct MPC from lightweight tools. With more participants, we can get t-privacy for $t < \frac{n}{2}$ using only lightweight tools. However, there are another way using randomness.

Attempt 2 In this attempt Alice runs the MPC protocol in her mind (for 3 parties). We're supposing MPC with 1-privacy and perfect correctness.

The protocol can be seen as the sudoku protocol (with commitments) and where the constraint i is that party i did not cheat and output is 1. Here, the communication between MPC nodes is commitment and also the party choices.

In this case if Alice runs MPC honestly then all works and we've completeness (that follows form MPC correctness). For soundness, we've it because we've perfect correctness (assumption about the MPC protocol) and, therefore, if all of them hold, the statement is true. Therefore, to convince Dani, Alice must cheat on behalf of at least one party. The probability is $\frac{2}{3}$ and therefore repeating it k times the probability will be small enough: $\left(\frac{2}{3}\right)^k$. However, here we don't have ZK. The ZK can be solved using secret sharing.

Results of this way to solve the problem Here, if we run MPC in the head, the communication complexity will be $O(k\,|\mathrm{VIEW}|) = O(k\,|R|)$ and the tools used is lightweight ones (commitments). Therefore, it's better than the another two ways (reducing to sudoku problem or running 2MPC in the two parties Alice and Dani).

3.1 A concrete lightweight MPC scheme

If we want to have MPC from correlated randomness, it's important to express f as a circuit, and the idea is that for Bob and Charly, we've the invariant that says: for wire value x, we've $x = x_B + x_C \pmod{m}$. The input x is the secret share x and to open x, they have to share their parts between them.

Now, if we want to add two values x, y, we just have to add individual parts: $z = z_B + z_C \pmod{m}$ with $z_B = x_B + y_B$ and $z_C = x_C + y_C$.

For multiplication, we've that $z = x_B y_B + x_C y_C + x_B y_C + x_C y_B$. We've two terms with a value from Charly and the other from Bob. So, the idea is to add Eve in the MPC calculation to help us with it (we won't go in more detail). For that, we've some openings and additions to do.

And finally, to reveal and show the output, they only have to open y and send it to Eve. It has a ZK property and we can create our simulators for Eve and Bob/Charly.

3.2 Round parallelization

If we parallelize the rounds, completeness and soundness holds, but we've to analyze deeper the design to adapt it for ZK. If Dani is honest, it works but if Dani is not honest then we don't know if it has ZK (because we can't make our simulator). However, there're workarounds for this problem. One way to do that is letting Dani prove knowledge of their choices first and then running an knouledge-extractor to get the indices that Dani chooses and then run the simulator.

Another way to solve this problem is don't letting Dani pick the random indices. Then, Alice will pick her own random challenges. To make it fair for the proof to Dani, we'll use Fiat-Shamir Heuristic (as H random oracle).

3.3 Better communication efficiency

The question, now, is if we can avoid repetition. For one round we saw that Alice only needs to cheat on behalf of one party, so bad probability in soundness is $\frac{2}{3}$. The solution is to have t challenges and run, instead of 3MPC, a larger MPC set (more nodes). So, we want this MPC with t-privacy and the probability to get unlucky now (for Dani) is $\frac{n-t}{n}$.

Also, we'll want this MPC with perfect correctness even if up to t parties cheat. So, Alice needs to cheat on behalf of $t+1$ parties. The probability, now, for Dani to get unlucky is $\frac{\binom{n-(t+1)}{t}}{\binom{n}{t}} = \mathrm{negl}$.

With that, we've completeness, soundness and ZK. Also, the communication complexity is the same as before $O(t\,|\mathrm{VIEW}|) = O(k\,|R|)$. But if we use a very special MPC (we won't talk about that), we can do better: $O(|R|) + \mathrm{poly}(k, \log(|R|))$.

3.4 In reality

Repetition performs better than parallelism and also this schemes asymptotically loses to zk-STARKs/zk-SNARKs, but wins for small computations. As an application, this protocol gives us efficient post-quantum digital signatures (because it does small and efficient computation).

4 MPC from ZKP

The idea is that every time each party sends a message, it proves that it runs honestly. It's a general way to transform a protocol with pasive security to active security. However, the messages are more complex because each one is attached with a proof of correct behaviour.