ZheroTag

Yiğit Kılıçoğlu[*], Lucas Huang[†]

Last updated: September 2022

Abstract: We describe a game, currently called ZheroTag, that uses advanced cryptographic concepts like zero-knowledge proofs, homomorphic encryption and multi-party computation. The game is relatively simple, but the challenge is to 1) come up with a P2P protocol that can be used to drive the game, 2) find libraries that can be used to implement the game. Our current progress and algorithms are listed at the end.

[*] `yigit.kilicoglu@yale.edu`.

[†] `lucas.huang@yale.edu`.

Game Description

The game is played on a finite square grid with two players. Each player has a single piece on the grid. The initial coordinates of the pieces are predetermined and are relatively far from each other. The pieces move like how kings move in chess. The players can only see the squares that are the Moore neighbors of the positions of their pieces (i.e. the 8 surrounding squares if the piece is not at a border of the grid). The game is played in alternating turns. At each turn, one player moves their piece to one of the Moore neighbors. The goal is to capture the opponent's piece, which can be done only when the opponent is on a Moore neighbor.

Note 1: The game is somewhat similar to Dark Chess, a chess variant. It can be played on chess.com. A tutorial can be found here.
Note 2: The game described above is the simplest version of this game that still requires advanced cryptography. Once we solve the cryptography part, the game can be made more complex.

Implementation

With a trusted third party

The implementation is very simple if there is a trusted party. The players share each of their moves with the trusted party, which in turn updates the board and tells each player what they can see on the board.

Without a trusted third party

This is what we are trying to do.

The challenge with implementing without a trusted third party is to keep the locations of the pieces private while updating the views of the players. Zero-knowledge proofs can be used to prove that a move was valid without actually revealing the move. However, this information is not enough for the next player to know which moves they can make. After each move, each player needs to update their board in such a way that the opponent does not learn anything about the location of their piece. If the last move brought a piece to a square around the other piece, then both players should learn about this fact.

P2P Protocol Solutions

PSI based on Diffie-Hellman

Let $U \subseteq \mathbb{R}^2$ be the set of positions on the board. Let $\mathcal{P} = \{u, \mathcal{N}, \mathcal{S}\}$ be a ZheroTag player where $u \in U$ is the current position on the board, $\mathcal{N}$ is the set of positions that $\mathcal{P}$ can move to (*neighbors*), and $\mathcal{S}$ is the set of positions that $\mathcal{P}$ can currently see.

Let Alice ($\mathcal{P}_A$) and Bob ($\mathcal{P}_B$) be the players of a ZheroTag game. Alice moves to u'. Call her new sets $\mathcal{N}'$ and $\mathcal{S}'$. Now noth parties need to update their views of the board. To update the boards, we define the following protocol:

One-sided Board Update Protocol:

1. Player $\mathcal{P}_1$ picks a uniform $\alpha \in \mathbb{Z}_p$ and sends $\mathcal{X}_1 = (H(\mathcal{N}_1'))^\alpha$ where H is a random oracle. to player $\mathcal{P}_2$ with a zero-knowledge proof that proves that (1) the move from u to u' is valid and that (2) the elements in $\mathcal{N}_1'$ contain the neighbors of u'.
2. $\mathcal{P}_2$ picks a uniform $\beta \in \mathbb{Z}_p$ and sends back $\mathcal{X}_1' = (\mathcal{X}_1)^\beta$ and $\mathcal{X}_2 = \{(H(u_2))^\beta\}$. $\mathcal{P}_2$ sends along a zero-knowledge proof that proves that $\mathcal{X}_2$ was calculated correctly.
3. $\mathcal{P}_1$ calculates $\mathcal{X}_2' = (\mathcal{X}_2)^\alpha$ and checks whether $\mathcal{X}_1'$ and $\mathcal{X}_2'$ intersect. If they intersect, then $\mathcal{P}_1$ is able to see $\mathcal{P}_2$. Otherwise, $\mathcal{P}_2$ is in the dark.

After Alice's move, Alice and Bob execute the One-sided Board Update Protocol two times. For the first one, Alice assumes the role of $\mathcal{P}_1$ and Bob assumes the role of $\mathcal{P}_2$. The roles switch in the second round. If a player can see the opponent when it is their turn, they win.

Problems/Concerns:

- PSI based on Diffie-Hellman works for sets that contain single numbers, not tuples. However, a random oracle that maps tuples to single numbers can be found.