A Protocol to Play Battleship Without a Trusted Party

Gavin Pai

Problem

The problem this project aimed to solve was making a game of Battleship where each player can ensure that the other is not cheating in real time. This may sound easy; however, an additional limitation present is that there can be no trusted agent to check each board. Therefore, no one but the person who created the arrangement of ships can know where each ship is until they choose to reveal that information. The mathematical subproblems to this problem are committing to an array of values without revealing anything about those values, ensuring and proving that those values are only ones and zeroes, and proving that all values add to an arbitrary amount without revealing anything else about those values.

Solution

A cryptographic primitive that was used to solve this problem was the Pedersen commitment. A commitment scheme is a scheme wherein a value is selected without revealing anything about that value. The value can not thereafter be changed. The Pedersen is perfectly hiding and computationally binding, which means that it satisfies the first requirement of a commitment scheme and is good enough for the second one. This was used to commit to the Battlefield array of ships. The other two mathematical sub problems are solved with the proofs below. The bit proof is used to prove that a specific value is a zero or a one, and the sum proof is used to prove that a list of commitments add up to an arbitrary number.

Bit Proof

Bit proof generation protocol:
C_c and C_r are properties of a commitment output. $H(x)$ is a cryptographic hash function. p, g, and h are properties of a commitment generator. x is the message.

$$
\begin{aligned}
x &= 0 \\
r, e_1, y_1 &\in \{x | x \in \mathbb{N}, x < p\} \\
x_0 &= h^r \mod p \\
x_1 &= h^{y_1}(g/C_c)^{e_1} \mod p \\
e &= H(x_0, x_1) \mod p \\
e_0 &= (e - e_1) \mod p \\
y_0 &= (r + C_r e_0) \mod (p-1)
\end{aligned}
\qquad
\begin{aligned}
x &= 1 \\
r, e_0, y_0 &\in \{x | x \in \mathbb{N}, x < p\} \\
x_1 &= h^r \mod p \\
x_0 &= h^{y_0}(C_c)^{-e_0} \mod p \\
e &= H(x_0, x_1) \mod p \\
e_1 &= (e - e_0) \mod p \\
y_1 &= (r + C_r e_1) \mod (p-1)
\end{aligned}
$$

All variables except r and x are sent. The receiver does not know which procedure has been performed.
Bit proof verification protocol:
c is the public commitment. $H(x)$ is a cryptographic hash function. p, g, and h are properties of a commitment generator.

$$
\begin{aligned}
e &= H(x_0, x_1) \mod p \\
x_1(c/g)^{e_1} \mod p &= h^{y_1} \mod p \\
x_0 c^{e_0} \mod p &= h^{y_0} \mod p \\
(e_0 + e_1) \mod p &= e \mod p
\end{aligned}
$$

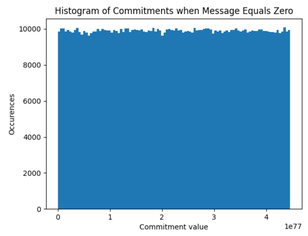
Sum Proof

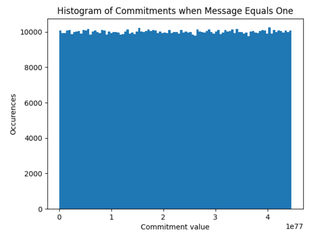
X is a list of messages. C is a list of public commitments that correspond to those messages. R is a list of blinding factors that correspond to those commitments.

$$X_f = \sum_{n=1}^{|X|} X_n \qquad C_f = \sum_{n=1}^{|C|} C_n \mod p \qquad R_f = \sum_{n=1}^{|R|} R_n$$

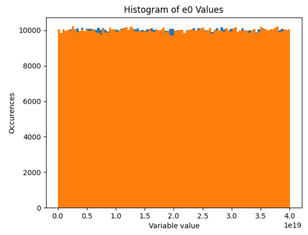
These can be verified using the generic commitment verification function.

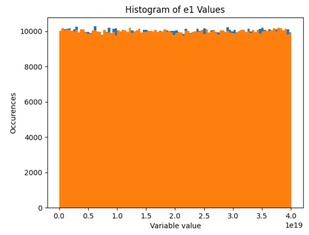
Results

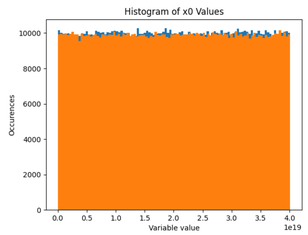


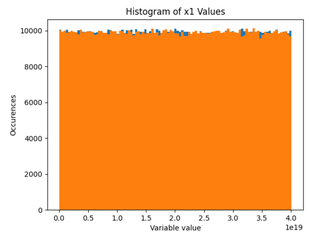


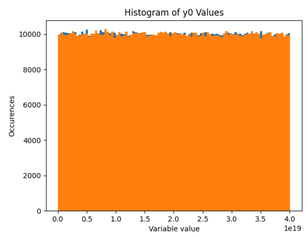
The top two histograms show the distribution of the one million Pedersen commitments when the message is zero and when the message is one. They look uniform and from the same distribution.

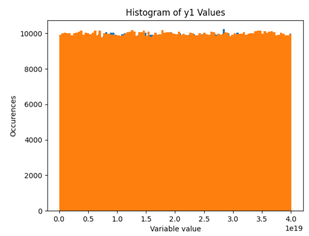












The next six graphs histograms are for the output variables of one million bit proofs. The histogram for when the message is one is overlaid on the histogram over when the message is zero. Each histogram for the variables in the bit proof show a uniform distribution with no significant differences.

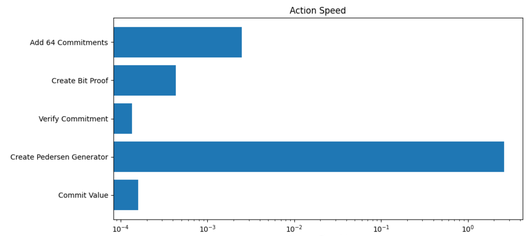
Statistical Analysis

For statistical analysis, a Kolmogorov-Smirnov test was performed on each variable of the bit proof and on the public commitment to see if each output was from the same distribution regardless of the value put into the bit proof or commitment. The Kolmogorov-Smirnov test tests to see if two samples come from different distributions. The results from this test are below.

Kolmogorov-Smirnov Test P-Values						
x_0	x_1	y_0	y_1	e_0	e_1	c
0.8198	0.5837	0.4030	0.7190	0.5396	0.7549	0.8362

The results were all over 0.1 which is the critical value, so the null hypothesis that the sample for when the value is zero and for when the value is one come from the same distribution cannot be rejected. This means that the other player in battleship cannot tell which bit is being committed when they receive a bit proof and a Pedersen commitment.

Practical Concerns



In the case of Battleship, time concerns are negligible. Even so, the operations performed at incredibly high speeds. The only operation that took a significant amount of time was creating the Pedersen generator, which only happens once per game. Otherwise, the operations happened in the blink of an eye. This is useful if this protocol is to be applied to other use cases.

Conclusion and Further Research

The project fulfilled its goal in making a functional battleship where no player can cheat through the concept of zero-knowledge. Not only did it work, but it worked in a fast manner such that there is a possibility that it can be used in other cases. One aspect that was ignored was ships occupying multiple spaces on a board, which could be considered in further research, however it is not really necessary to do so because other use cases would not have the same problem.

References

[1] Amos Fiat and Adi Shamir. “How To Prove Yourself: Practical Solutions to Identification and Signature Problems”. In: *Advances in Cryptology — CRYPTO’ 86*. Ed. by Andrew M. Odlyzko. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 186–194. ISBN: 978-3-540-47721-1.

[2] Ninghui Li. *Zero-Knowledge Proof and Cryptographic Commitment*. Apr. 2012. URL: https://www.cs.purdue.edu/homes/ninghui/courses/555_Spring12/handouts/555_Spring12_topic23.pdf.

[3] Frank J Massey Jr. “The Kolmogorov-Smirnov test for goodness of fit”. In: *Journal of the American statistical Association* 46.253 (1951), pp. 68–78.

[4] Torben Pryds Pedersen. “Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing”. In: *Advances in Cryptology — CRYPTO ’91*. Ed. by Joan Feigenbaum. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 129–140. ISBN: 978-3-540-46766-3.

[5] C. P. Schnorr. “Efficient signature generation by smart cards”. In: *Journal of Cryptology* 4.3 (Jan. 1991), pp. 161–174. ISSN: 1432-1378. DOI: 10.1007/BF00196725. URL: https://doi.org/10.1007/BF00196725.

[6] Vitaly Shmatikov. *Introduction to Zero-Knowledge*. 2009. URL: https://www.cs.utexas.edu/~shmat/courses/cs380s_fall09/16zk.pdf.