Zero Knowledge Proofs Final Report

Rohit Das (@rdas49); Richard Cho (@rcho9); Dheeraj Eidnani (@deidnani3)

November 19, 2018

1. Basic Definition:

Zero-knowledge proofs allow one party (a prover) to prove to another party (a verifier) that they know a certain value of "x" without revealing anything about the value "x". Zero-knowledge proofs have three properties: completeness (an honest verifier can prove a true statement made by an honest prover), soundness (an honest verifier cannot be convinced of a false statement by a cheating prover), and zero-knowledge (verifier doesn't need a secret to understand whether the statement is true).

Zero knowledge proofs were first invented by Goldwasser, Micali, and Rackoff in 1982. In general, zero knowledge proofs fall into two categories: protocol design and identification scheme. A protocol design is an algorithm for parties to achieve some goal; an example of this is the Diffie-Hellman protocol that assumes that both parties will follow the algorithm instructions. However, zero knowledge proofs take this one step further by achieving security even when one of the parties is "cheating" and not following instructions. This involves designing a cryptographic protocol that first assumes everyone will follow the algorithm, and then "forces" them to follow the instructions using a zero knowledge system. Identification schemes are equivalent to a secret PIN number or key card used on a box outside a door for entrance. However, with a traditional PIN or a keycard, the box can be examined and extracted for the secret keys. Instead, using a zero knowledge proof, we can have the box contain a composite number n= p*q, give authorized people the solution p, q, and have them prove to the box they know the factorization in zero knowledge.

2. Easy-to Understand Examples (These will have pictures/diagrams accompanying them):

To understand these complex protocols, let us first consider the "color blind friend" proof. Imagine your friend is colour-blind and you have two balls: one that is red and one that is green, but are both otherwise identical. To your friend, they seem completely identical and he is skeptical that they are actually distinguishable. You want to prove to him they are in fact differently-coloured, but nothing else - thus you do not reveal which one is the red and which is the green.

Here is the proof system. You give the two balls to your friend and he puts them behind his back. Next, he takes one of the balls and brings it out from behind his back and displays it. This ball is then placed behind his back again and then he chooses to reveal just one of the two balls, switching to the other ball with probability 50 percent. He will ask you, "Did I switch the ball?" This whole procedure is then repeated as often as necessary.

By looking at their colours, you can of course say with certainty whether or not he switched them. On the other hand, if the balls were the same colour and hence indistinguishable, there is an infinitely small change you could guess correctly with probability higher than 50 percent.

If you and your friend repeat this process multiple times, your friend should become convinced ("completeness") that the balls are indeed differently coloured; otherwise, the probability that you would have randomly succeeded at identifying all the switch/non-switches is close to zero ("soundness"). The above proof is "zero-knowledge" because your friend never learns which ball is green and which is red; indeed, he gains no knowledge about how to distinguish the balls.

3. Explanation of the Code:

We programmed the Feige-Fiat Shamir Authentication Protocol and the Discrete Logarithm Protocol, two common forms of zero knowledge proofs. Typically, in cryptography and the programs created, Peggy represents the prover and Victor represents the verifier. Peggy is trying to prove that she knows her password: s (where s is the smallest square root of the multiplicative inverse of v (modulo n)), where v is a quadratic residue modulo n.

To prove this password with the Feige-Fiat Shamir Authentication Protocol, she chooses a random number $r < n$, and computes $x = r^2 \pmod{n}$. She sends x to Victor, who then chooses b as 0 or 1 (and sends it to Peggy). Peggy computes $y = rs^b \pmod{n}$ and sends y to Victor. To check whether Peggy is honest with what she has picked, Victor checks $x = y^2v^b \pmod{n}$. This works because $y^2v^b = (rs^b)^2v^b = r(s^2v)^b = x(v^{-1}v)^b = x \pmod{n}$.

Let us now prove that Mallory can cheat with success probability $\frac{1}{2}$ in each round. If Mallory satisfies a condition if $b = 0$ or 1, when $x = y_0 \pmod{n}$ and $x = y_1^2 \pmod{n}$ for some y_0, y_1. We can solve these solutions to get $v - 1 = (y_1)^2 * (y_0)$. However, $y_1y^{-1} \pmod{n}$) is a square root of v^{-1} which contradicts the assumption that Mallory is cheating as she doesn't know a square root of $v - 1$. Now, let us suppose that Mallory guesses that $b = 0$, he chooses an $x = r^2 \pmod{n}$) and $y = r \pmod{n}$ for some random number r, which satisfies $b = 0$ but not $b = 1$. Similarly, let us suppose that Mallory guesses that $b = 1$, then he chooses an y and x such that $x = y^2v \pmod{n}$), which would fail the condition for $n = 0$. Thus, the probability of Mallory successfully cheating is the probability of Mallory correctly guessing b for each round, which is $\frac{1}{2}$.

However, Mallory's overall probability of correctly guessing b for every round approaches 0 as more checking iterations are done. This can be modeled by the function $y = (1/2)^r$, where r is the number of rounds and y is the probability that Mallory successfully cheats.

The second program also uses the power of zero knowledge proofs, but confirms whether Peggy knows the discrete logarithm (knows x if $A^x = B \pmod{n}$). First, Peggy picks a random number between 0 and $p - 1$ and sends Victor $h = A^r \pmod{p}$. Victor randomly returns either 0 or 1 to Peggy. Peggy sends $s = (r + bx) \pmod{p - 1}$ to Victor, who then computes $A^s \pmod{p}$ and checks whether it is equal to $hB^b \pmod{p}$. This works because $A^s = A^{r+bx} = (A^x)^bA^r = hB^b \pmod{p}$. Note: this only works if x is the discrete logarithm of A modulo B.

Now, let us now prove the success probability in this program to be at least $\frac{1}{2}$. If $b = 0$, Mallory can choose a random r and set $h = A^r \pmod{p}$, which would satisfy $b = 0$ as Mallory can send $s = r$, and the verifier will be able to check that $h = A^s \pmod{p}$. But, if $b = 1$, Mallory fails to cheat as he can't compute an s that would satisfy $A^s = hB \pmod{p}$ because this would require Mallory to find the discrete log of hB. Similarly, in the case if $b = 1$, Mallory can choose an h such that $h = A^sB^{-1}$ for random integer s, which would satisfy $b = 1$, as this would satisfy $A^s = hB^b \pmod{p}$. But, if $b = 0$, this fails to work as Mallory cannot make an r such that $A^r = h \pmod{p}$. Thus, similarly, the probability of Mallory cheating successfully in a round is the probability of Mallory correctly guessing b, which is $\frac{1}{2}$.

Once again, Mallory's overall probability of correctly guessing b for every round approaches 0 as more checking iterations are done. This can be modeled by the function $y = (1/2)^r$, where r is the number of rounds and y is the probability that Mallory successfully cheats.

4. Real World Applications:

One real world application of Zero Knowledge Proofs (ZKPs) are to confirm the authenticity of nuclear weapons without sharing any secret design information. Although ZKPs are mainly used in cryptography, due to recent arms control agreements, a trusted mechanism to verify the authenticity of items presented as nuclear warheads has been developed using ZKPs.

Zero knowledge proofs are also used in Blockchains via the zero knowledge protocol in zkSnarks. SNARK stands for succinct non-interactive arguments of knowledge.

Succinct: the messages sent are small in comparison to the calculations.

Non-interactive: there is little to no interaction, meaning anyone can verify without acting anew.

ARguments: The verifier is protected against computationally limited provers (Soundness).

Knowledge: It is impossible for the prover to construct a proof without knowing a witness (ex: the hash).

Through this protocol, the identity and amount being spent can be hidden and assets can be transferred across a distributed blockchain network with secrecy.

Finally, ZKPs are also used in authentication systems. Often, many secure systems are made from unreliable devices such as public kiosks that could be compromised. These compromises include (but are not limited to) hardware keylogging, software keylogging, cameras, or even human intervention. There are numerous ways to monitor devices and there is no concrete method to prevent them all. The only reliable method is to log in through systems while not passing any sensitive information through the untrusted device. Currently, the most prominent ZKP system used for authentication is the Socialist Millionaires Problem (SMP Protocol).