ZK Bootcamp: Day 2 Problem Set

Brian Justin Stout*

July 25, 2023

Problem 1. *Two parts. Part a: Is it true that all odd squares are* $\equiv 1(\text{mod}8)$*? Part b: What about even squares? Are they* $\equiv 1(\text{mod}8)$*?*

Let a be an odd square. Then for an odd number α we have $\alpha^2 = a$. Subtract 1 from both sides and factor $(\alpha+1)(\alpha-1) = b$ and we wish to show $8|b$. Because α is odd both $\alpha+1$ and $\alpha-1$ are both even. We claim that for any α odd it is true that $\alpha+1$ or $\alpha-1$ is divisible by 4 and hence all even squares are congruent to $1(\text{mod}8)$. We proceed inducively. If $\alpha = 3$, then $\alpha+1$ is divisible by 4. Suppose the result is true for the n^{th} odd number. Arguing by symmetry, assume that $\alpha+1$ is divisible by 4. Then the next odd number is $\alpha+2$. But then $(\alpha+2)-1 = \alpha+1$ which is divisible by 4.

If a is an even square, we claim $a = 0(\text{mod}8)$ or $a = 4(\text{mod}8)$. Clearly $4 = 2^2 = 4(\text{mod}8)$ and $16 = 4^2 = 0(\text{mod}8)$. Assume that $(2k)^2$ is congruent to 0 or 4 modulo 8. Then the next even square is $(2(k+1))^2$ which is congruent to $4(k^2+2k+1) = (4k^2+8k+4) = 4k^2+4(\text{mod}8)$, which is either 0 or 4 depending on the last even square.

Problem 2. *Try out the vanity bitcoin address example at asecurity or for Ethereum.*

I started running an in-browser vanity address generator to find an address with "brian" at the beginning. I started running on 4 threads at 12:56pm EST. At 2 hours later I have not found my vanity address and had about 1% chance so far of finding it.

Problem 3. *What what do you understand by (a)* $O(n)$*, (b)* $O(1)$*, and (c)* $O(\log(n))$*? For a proof size, which of these would you want?*

The terms $O(n), O(1), \text{and} O(\log(n))$ represent complexity classes of algorithms. They mean that for an algorithm the number of computations on an input of size n grows approximately like n, constant, or $\log(n)$. It does not mean that the algorithm uses exactly this number of computations every time, just that the worst case, dominant term is that function, respectively. $O(1)$ means the number of computations is constant size, or doesn't depend on the input size n. A proof of size $O(1)$ would be considered very small, because no matter what the statement (large, very large, or extremely large), the proof size is basically the same size.

*email: bstout.eth@ethermail.io