Problem 10): *Any* **RSA** implementation where the modulus N is prime will always be insecure. There are two reasons for this insecurity. These reasons are rooted in the fundamental mathematics that underlies *all* **RSA** based crypto-systems. While each of these reasons is sufficient to cause a break in security on its own, choosing the modulus as N a prime forces both of these to occur.

The first reason has to do with the elements in the group defined by N and used in any **RSA** crypto-system, $\mathbb{G} = \mathbb{Z}_N^\star$. When N is prime, then $\mathbb{Z}_N^\star$, is automatically known because $\mathbb{Z}_N = \mathbb{Z}_N^\star$, by the definition of primality. The problem is that this every element in $[0, N]$ will be in the group eliminating the ambiguity about which elements belong to the group in use. Furthermore, this choice of N eliminates the ability to use any element of the group as a generator. This makes the *Discrete Logarithm Problem* easier to solve. Since the difficulty of the *Discrete Logarithm Problem* constitutes a critical part of the fundamental mathematical assumptions relied upon by all **RSA** crypto-systems, this choice of N causes any implementation using it to be insecure.

The other reason setting the modulus N as a prime is problematic has to do with the order of $\mathbb{Z}_N^\star$, $|\mathbb{Z}_N^\star|$. When N is prime, then $|\mathbb{Z}_N^\star|$ cannot be prime for any $N > 3$, by the definition primality. The problem here is that the *Decisional Diffe-Hellman Problem* is not hard for groups with a non-prime order. The security **RSA** relies upon the fundamental mathematical difficulty of solving the *Decisional Diffe-Hellman Problem*. Since this choice of N makes the *Decisional Diffe-Hellman Problem* easier to solve, it will also break the security of any **RSA** implementation using that choice.