RSA

Q1: What are the properties of a random oracle?

A1: The oracle is a black-box, we do not know the internal works. We only give it a binary string as input, and it returns a binary string as output. Everyone (honest and adversarial) can interact with the box, that is: they can query the oracle on x. Such queries are said to be private, and no one else learns what x is during the query. In fact, they do not know that oracle was queried at all. (This is because such queries are done locally in practice.) Call this oracle H for now:

- **Consistency**: H is *consistent*. If returned y to some query $H(x)$, it will always do that for everyone when they query $H(x)$.
- **Uniformity**: If x has not been queried to H, then the value of $H(x)$ is *uniform*.
- **Extractability**: If $\mathcal{A}$ queries $H(x)$, the reduction (i.e. algorithm B that we construct from $\mathcal{A}$) can see the query and learn (*extract*) the query x.
- **Programmability**: The reduction (i.e. algorithm B that we construct from $\mathcal{A}$) can set (*program*) the value of $H(x)$, which is the response to some query x on H, to a value of it's own choice. However, this chosen value must be uniformly distributed.

Q2: Let $GenRSA$ be a PPT algorithm that, on input 1^n , outputs a modulus N that is the product of two n-bit primes, along with integers e, d satisfying $ed = 1 \bmod \phi(N)$. Let H be a function with domain $\{0,1\}^*$ and range $\mathbb{Z}_N^*$ for any N. Construct a signature scheme as follows:

- Gen: on input 1^n, run $GenRSA(1^n)$ to compute (N, e, d) and set the range of H to be $\mathbb{Z}_N^*$. The public key is (N, e) and the private key is (N, d).
- $Sign$: on input a private key (N, d) and a message $m \in \{0,1\}^*$, compute $\sigma := [H(m)^d \bmod N]$.
- $Vrfy$: on input a public key (N, e), a message m, and a signature σ, output 1 if and only if $\sigma^e = H(m) \bmod N$.

Formally prove that if the RSA problem is hard relative to $GenRSA$ and H is modeled as a random oracle, then the construction above is existentially unforgeable under an adaptive chosen-message attack.

A2: Basically we want to show that RSA assumption $\implies {}_RO\Pi$ is CPA-secure. Suppose there is an algorithm $\mathcal{A}$ that breaks Π. Then we will show it would be possible to construct an algorithm B that breaks the RSA assumption. The answer is given in the figure.