Jiping Zhang
CSE 207B: Applied Cryptography Due:

Homework 6

0. (a) I implemented the Quick pairwise GCD finding algorithm taught on class.

   The most serious problem I ran into, is that: at first, I calculated the product with all the moduli Ns with naive method: multiplying them one by one. This takes a lot of time.

   Later, I realized that doing so is really not wise, I guess that's because sage uses FFT(fast fourier transform) algorithm to multiply big int, so in my naive method's time complexity is $T(n,m) = T(n-1,m) + O(mnlog(mn))$, $T(n,m) = O(n^2mlog(mn))$ for multiplying n m-bits integers.

   I modified it to the method like a tree, that is, first calculate N_1N_2 , N_3N_4 , N_5N_6 , N_7N_8 , ... then calculate $N_1N_2N_3N_4$, $N_5N_6N_7N_8$, ... , then calculate step by step until finally calculating the product of all numbers.

   Here, time complexity is $T(n,m) = 2T(\frac{n}{2},m) + O(mnlog(mn))$, $T(n,m) = O(mnlog(mn)log(n))$, much faster than the naive method.

   (b) there are 2 more problems.

   1. the moduli N used to encrypt the AES key, N-1 can be factored into smaller primes' procuft.

   2. the AES with n bytes of n at end, may be vunerable to padding oracle attack.

1. (a) let $b_i = P_x(2^{i-1})$ $(i \geq 1)$

   meaning of b_i is whether $x - x_{prev} > \frac{N}{2^i}$

   after querying P_x with $2, 4, 8, ..., 2^{\lceil log_2 N \rceil}$

   we get $b_1, b_2, ..., b_{\lceil log_2 N \rceil}$

   we can then compute x: $x = \sum_{i=1}^{\lceil log_2 N \rceil} rounding(\frac{N}{2^i})$

   (b) name the oracle function we can query to be f, that is to say

   $f(z) = 1\ if\ [z^{\frac{1}{e}}\ mod\ N] \geq \frac{N}{2}\ else\ 0$ and we can query f as we like.

   we want to get $[c^{\frac{1}{e}}\ mod\ N]$,

   let $x = [c^{\frac{1}{e}}\ mod\ N]$, and we want to get x.

   we construct a function P so that $P(r) = f((r^e \cdot c)\ mod\ n)$

   $P(r) = f((r^e \cdot c)\ mod\ n)$

   $= 1\ if\ [(r^e \cdot c)^{\frac{1}{e}}\ mod\ N] \geq \frac{N}{2}\ else\ 0$

   $= 1\ if\ [r \cdot c^{\frac{1}{e}}\ mod\ N] \geq \frac{N}{2}\ else\ 0$

$= 1 \ if \ r \cdot x \geq \frac{N}{2} \ else \ 0$

as you can see, P behaves the same as P_x in (a), and we can use the method in (a) to solve x.