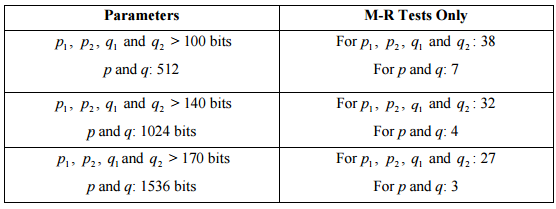
**What is the general procedure for generating primes with in the library?**

The library first determines the number of rounds of the Miller-Rabin primality testing using the National Institute of Standards and Technology's Federal Information Processing Standards Publication. This Standard allows two alternatives for testing primality: either using several iterations of only the Miller-Rabin test, or using the iterated Miller-Rabin test, followed by a single Lucas test [1]. Python-RSA utilizes the standard set by Table C.3 for the minimum number of rounds of M-R testing when generating primes for use in RSA Digital Signatures (error probability ).



Are they following industry standard?

Is it secure?

[1]

They use M-R tests only because of the speed advantage. Next, the library tests the primality using Miller-Rabin primality testing of some odd integer with number of iterations = minimum M-R tests only + 1. The implementation is consistent with the FIPS standard established on pages 71 and 72. The only dependency that needs to be verified is the RGB (random bit generator) used in the primality testing.

**Are they following industry standard?**

Yes, it seems so.

**References:**

[1] <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf#page=62>