WHAT ARE RSA AND K NUMBERS

In a remarkable 1978 paper by Rivest, Shamir, and Adleman it was shown that public key cryptography can be used to communicate secretly between two parties on the open airwaves. The success of the method arises from the fact that large semi-primes N=pq are not easily factored in a short time when the digit length exceeds 200 or more. Recently, however, with the advent of ever faster super-computers, one is reaching the point where semi-primes of this length can often be factored in a few seconds. This fact calls into question the security of all RSA public key cryptography .

We want in this article to look at some of the larger public key semi-primes being used and discuss how they may be factored using a new approach far simpler than other existing methods. The method is based on the point function S=sigma(N)-N-1.

Let us begin our discussion with the well known RSA semi-primes RSA-100 through RSA-2048 which you can find listed at-

https://en.wikipedia.org/wiki/RSA_numbers

There are some 54 of these with 19 of them having already been factored by 2017 mostly using the generalized number field sieve approach.The number following the RSA designation represents the number of digits D in the number. Thus the smallest reads-

RSA-100 =**1522605027922533360535618378132637429718068114961380688657908494580122963258952897654000350692006139**

This semi-prime was first factored in 1991 by A. Lenstra. At that time it took a solid four month of computer run time to produce the following factoring-

37975227936943673922808872755445627854565536638199 × 40094690950920881030683735292761468389214899724061

Today this number can be factored in just a few minutes using generalized sieve methods. Typically the larger the number of digits contained in N=pq the longer it will take to produce a factoring. Things can be speeded up by running computers in parallel but even this has not produced a successful factoring for most numbers above RSA-220 and especially for the largest listed RSA number. of RSA-2048. In the early days there was actually a contest established offering monetary rewards to anyone capable of factoring any of the RSA numbers. This contest, however, was suspended by 2007 when it became apparent that

factoring of some of these semi-primes was becoming possible with reasonable computer run times.

The standard method for generating an RSA number of D digit length is to start with a string of D/2 random digits and adjust the last few digits until a prime is created. Call this prime p. Next repeat the procedure with a new random number sequence of D/2 length to generate a second prime q. Then multiply p and q together to get the semi-prime RSA-D. Let us demonstrate things for D=24 where a pair of twelve digit long random numbers read 427419669081 and 321110693270. Adjusting the end digits produces the primes –

p=427419669071 and q=321110693273

Multiplying these together produces the RSA number-

N= RSA-24=137249026253905045859383

A number of this relatively small size is easy to factor by existing methods with perhaps the simplest approach being our own which simply asks for the two roots of-

x^2-Sx+N=0

, where x=[p,q] and S={sigma(N)-N-1}=748530362344. Here sigma(N) is the sigma function of number theory. It took only a split second to calculate S on our home PC using the math program MAPLE V. Solving the quadratic, then leads at once to the above values for p and q.

An alternate way to generate large semi-primes is our own approach first introduced on this Web Page about six years ago. Instead of random numbers we rather use the product F of several irrational numbers written out to D digit length. We then take a sequence of D/2 digits out of this expansion specifying which is the first digit after the decimal point for the first digit in the sequence. With some end digit adjustment this produces the prime p. Repeating the procedure with a different starting point and adjusting the last few digits we get a second prime q. The new semi-prime N=pq will be referred to as K-D.

Let us demonstrate how one generates a K-24 number. Start with the 24 digit long expansion of the irrational product-

F=evalf(exp(-3)*sqrt(Pi)/17,24)=0.005190898885524508845247…

Assuming we want a twelve digit long p approximation starting with the 4^{th} digit after the decimal in F , we have p ≈ 190898885524. On adjusting the last digit we get the prime-

p=190898885521

To get the second prime of digit length 12 we start with the 12 digit q approximation of q≈552450884524. After adjustments this yields the prime-

q=552450884521

taking the product of p and q , we get the 24 digit long semi-prime-

N=K-24=105462258160149569920441

Here S=sigma(N)-N-1= 743349770042 is found in a split second. Solving

x^2-Sx+N=0 then produces the above values of x=[p,q].

Although the K-D numbers are not constructed from a string of random numbers this should not matter in using them as public keys. They have a big advantage over RSA semi-primes in that all their digits are contained in the product function F of irrational numbers known to the sender and friendly receiver. This allows for easy storage and transmission. Here are a few of these semi-primes larger than K-24 which we have constructed-

K-32 =22780948237690657892498545869677

K-38 =79259796022025569219580682646178858411

K-72=
4702467291450675664813883626348043467405473087530879579461319752699185 11

K-100=
4232645070331114502389141379272067425948811706763611499183494431736227 63134452332592218954322366097 7

K-191=
4924101399098474368810691739772255073040815158698838316935202523323070351 7626782849911820364583705222271807483359386351007941543360388146295342591 6067144812412950116302145632490117694989861 1

K-614=
1017896104727357477398783880737638288535675208745672064629871026745216871 2495222515643180506037409999704105127366045345376709856244024782159295061 8855103001693997808591907849897337319190980251607220548850254539025470490 1555884921168795290473324675791003567516773575764686186161748192468138858 0619201828814512572763235164905681170006408224152625718516087855823443562 9265337305958371435350451232336725819781360972863637766737912010330709691

447984191477902062096569214678665855813197204472966401365619258963025491584409118118689395356416373352154552359454774482526210254929781929735818767556710696652424734374770115 31

and-

K-1419=

831015614177066402847308220869907378005268537632530746940210713395960268989669189603503273121693085810785207178287016660831515055557300691996566415526348523856728230469228860147073949112686103998068294140157668605192825524372067878677617379349353724335081659782609152007548518915772523949719550201959761252834368499119515927946156992353127547243242427257733941757567507865332428476100905079102507587569661923702608043479241536356723414228953006820503816848287604982272768139736271851033285743956577498520296604987059713664855002333933557710511855923393318203521048487531419504382709694747718307170527637401599858233279510236951867677936856074278828228114108664821362293621033754804417061002342588412247589242678995781867108153346410570601014103835149672638233521792206090262393200968112491834232419608744000963915358310290277559287672245038207244602778993760572460146246002367519265660446625230108295540084435539261717785821432063026419524179976520781366705599167492675950365289207032717735217950703811078441485401480658221578021756136598265295691284689526470474964059985773019877152990887695045067438550087099639344190161971272098794679704586936801686593707937437878928542509734998144074110681294312409959328723965542015360840593658236148604579770777332932089042109829279458201969846321994810668682784985830350714374606163433996209631440144630713860445111636542548208515780469077948346357427389604990109182201007760467

Of all of the above K-Ds found, it was the last two which were the most time-consuming to establish. For K=614 we used F=π^3sqrt(2) to generate p and F=πexp(2) to generate q. The product of primes p and q then generated the 614 digit number shown. The K-1419 was also generated by a group of irrational numbers. I first submitted this largest of our K-D numbers as a factoring challenge on the internet in 2012. So far no answer has been received. I would not expect to receive one before 2025.

One notes that these K-D semi-primes (as well as the RSA-D numbers) are not unique for a given D since there are many different random number strings and different irrational product functions F one can use to generate different semi-primes with the same D.

Finally a few words concerning factoring procedures for both RSA-D and K-D semi-primes. The standard approach has been to use the Lenstra elliptic curve factorization and more frequently the generalized sieve method. We have come up with a third factoring approach employing the point function S=sigma(N)-N-1.

For smaller semi-primes up to about 100 digits length, the last technique is the simplest and fastest. It involves essentially finding the value of sigma(N) and then solving the quadratic-

$$x^2-\{sigma(N)-N-1\}x+N=0$$

This approach has remarkable factoring speeds for RSA-Ds and K-Ds when the digit size is D=100 or less. On my home PC using my old MAPLE V computer program, numbers such as K-38 or less can be easily factored in split seconds. However above about D=40 the computer run times become prohibitive. My colleague Dr.Richard Fearn(Professor Emeritus University of Florida) has used his much newer MATHEMATICA11.3.0.0 program to find K-100 in a couple hours of computer run-time on his home PC. I have no doubt that with the use of the latest super-computers one will be able to factor RSA- 230 and K-191 using the present approach in just a few seconds. If this indeed turns out to be the case, then public key cryptography, used almost exclusively in modern day secret electronic communications, will no longer be secure from adversaries.

U.H.Kurzweg
July 18, 2018
Gainesville, Florida