HOW RSA CRYPTOGRAPHY WORKS

The preferred method of modern day cryptography is the technique first published in 1978 by Rivest, Shamir, and Adleman and referred to in the literature as the RSA technique. Actually the technique was discovered somewhat earlier by the British mathematician C. Cocks (1973) but not published.

The idea behind the technique is the use of a private and public key. Let us explain. Take two parties A and B who want to communicate with each other in secret over the open airwaves. First, party A picks two large prime numbers p and q which form his private key. Next he multiplies these two primes together to generate the unbreakable public key N=pq. Also he constructs a low integer number ε which has the property that the greatest common denominator between it and the number φ=(p-1)*(q-1) is unity. Since φ is always an even number, any low prime number (excluding 2) for ε will do. He now sends out to B (and anyone else who might be listening) the public key N and ε.

Next B receives the public key and wants to send A a coded version C of a message M. First B encodes the message via the mathematical operation-

$$C = M^{\varepsilon} \bmod(N)$$

Here the mod notation comes from number theory and means one divides M^{ε} by N and what remains over is the C. Thus 14 mod 3=2 and 15 mod 5=0. The coded message C is next send over the airwaves and received by A (and anyone else who is listening). However, it is only A who can decipher things via the following operation. First he constructs a number δ based on his private key. It is generated by-

$$\varepsilon\delta \bmod(\varphi) = 1$$

Recall that φ=(p-1) *(q-1) and is known only to A and no one else. To now recover the uncoded message, A needs to simply carry out the operation-

$$M = C^{\delta} \bmod(N)$$

Thus only A and B will know the message unless someone is able to factor the large number N quickly so that p and q can be found. It typically would take several hundred years for the fastest electronic computers to factor public keys of the order of 500 digit length by the presently used brute force approach of division by all odd integers less than the square root of N. Hence such keys are quite secure unless someone clever comes up with a more efficient way of factoring such large numbers. The beauty of the method is that any one can quickly generate an unbreakable public key if one uses a couple of large primes **which are not known to others**. Here is a public key which I generated in about 20 minutes including finding two large primes heretofore unknown

PUBKEY :=
831015614177066402847308220869907378005268537632530746940210713395960268
989669189603503273121693085810785207178287016660831515055557300691996566
415526348523856728230469228860147073949112686103998068294140157668605192
825524372067878677617379349353724335081659782609152007548518915772523949
719550201959761252834368499119515927946156992353127547243242427257733941
757567507865332428476100905079102507587569661923702608043479241536356723
414228953006820503816848287604982272768139736271851033285743956577498520
296604987059713664855002333933557710511855923393318203521048487531419504
382709694747718307170527637401599858233279510236951867677936856074278828
228114108664821362293621033754804417061002342588412247589242678995781867
108153346410570601014103835149672638233521792206090262393200968112491834
232419608744000963915358310290277559287672245038207244602778993760572460
146246002367519265660446625230108295540084435539261717785821432063026419
524179976520781366705599167492675950365289207032717735217950703811078441
485401480658221578021756136598265295691284689526470474964059985773019877
152990887695045067438550087099639344190161971272098794679704586936801686
593707937437878928542509734998144074110681294312409959328723965542015360
840593658236148604579770777332932089042109829279458201969846321994810668
682784985830350714374606163433996209631440144630713860445111636542548208
515780469077948346357427389604990109182201007760467

EPSILON:=65551

I challenge anyone to break this 1419 digit public key!

Let us conclude by carrying out a very simple demonstration of the RSA method following the above steps-

$Let\ p = 5, \quad q = 7$

$then\ N = 5(7) = 35$

$Also\ \varphi = (p-1)*(q-1) = 24\ so\ \varepsilon = any\ prime < \varphi - 1$

$Let\ \varepsilon = 11$

$Thus:\ public\ key\ N = 35\ and\ \varepsilon = 11$

$Bs\ message\ is: M = 16$

$coded\ message: C = 16^{11} (\mathrm{mod}\ 35) = 11$

$Next\ A\ constructs\ \delta:\ (11\delta) \bmod (24) = 1\ so\ \ \delta = 11$

$so\ M = C^{\delta} \bmod (N) = 11^{11} \bmod (35)$

$thus\ M = 16\ which\ is\ Bs\ message!$

In this particular example, the p and q are so small that the public key N=35 is at once recognized as the product of p=5 and q=7 and hence 35 would be an insecure key. Also keys involving products of Mersenne Primes would be insecure since these are recognizable worldwide. The secret is to come up with, simple to generate, primes in the several hundred digit range of which no one else is aware of. Go to http://www.mae.ufl.edu/~uhk/MATHFUNC.htm and look for GENERATING-PRIMES.pdf to see one way to quickly create large ps and qs.

The reason we have the fastest electronic computers in the world located at a nation's national security agencies is the need for the fast number crunching of large numbers required in cryptography and the continued urgency to keep ahead of adversarial code breakers.

The RSA technique has several weaknesses in addition to people being possibly able to factor large Ns quickly by some as yet unknown approach. Since there is a public key , anyone could produce their own coded messages containing false information and send them to A . This would confuse A as to what message is the correct one. Also B could be bombarded with public keys not coming from A. B could thus be tricked into sending information to a third individual. There is also the possibility that Bs encoded message is intercepted by a third party and he just tries different δs until the message becomes decoded. I am quite sure these points have been considered and filters are being applied to avoid such problems. Also it should be kept in mind that an outsider has available the numbers N, ε, and C by just listening. Thus although he may not be able to factor the semi-prime N he could apply brute force to solve $M=[C+n\ N]^{(1/\varepsilon)}$. This is not an impossible task. For the above example where N=35, ε=11, and C=11, we can decipher things into the integer message M=16 by using n=247135881697.