MAT 302: LECTURE SUMMARY

We began lecture by reviewing the RSA procedure (the original 1977 paper introducing the algorithm has been posted to the course website, incidentally):

(1) Bob (secretly) picks enormous (e.g. 100-digit) primes P and Q.
(2) Bob publicly announces the quantity $N(= PQ)$ and e. Note that e is any positive integer which is relatively prime to $\varphi(N)$.
(3) Alice encrypts a plaintext X by computing $X^e \pmod{N}$; she then sends this number (call it Y) to Bob.
(4) To decrypt this, Bob must solve the equation $X^e \equiv Y \pmod{N}$. He does so by computing $d = e^{-1} \pmod{\varphi(N)}$ and evaluating $Y^d \pmod{N}$; this recovers the plaintext X.

Oscar can't crack this because (in decreasing order of generality)

(1) it is unknown how to efficiently solve a congruence $X^e \equiv Y \pmod{N}$ without knowing d;
(2) it is unknown how to efficiently determine d without knowing $\varphi(N)$; and
(3) it is unknown how to efficiently determine $\varphi(N)$ without knowing the factorization of N, i.e. without knowing P and Q.

There are various points in the above summary which demand elaboration. Here are a few which we discussed today.

(1) Jonathan brought up the following question: how does Bob find such gigantic primes? This is a serious concern which we will discuss next time (but also, check out Problem 4.4(d) on the latest assignment!).
(2) One potential attack Oscar could mount is to guess d by brute force: given the ciphertext Y he could compute $Y^d \pmod{N}$ for a bunch of d and hope that the plaintext falls out. For this reason, it is important that d is very large (and hence unlikely to be guessed). Therefore, in practice, Bob must first (secretly) select a huge value of d, then compute its inverse e, which he then makes public. Note that e needn't be large. In fact, if e is small, it makes it easier for Alice to encrypt her plaintext.
(3) Last time, in our approach to solving the congruence $X^e \equiv Y \pmod{N}$, we assumed that the plaintext X is relatively prime to N. This will almost certainly be the case, since to *not* be relatively prime to N, X would have to be divisible by P or Q. It is highly unlikely that Alice's plaintext would happen to be divisible by one of these two huge primes, or even that it's *larger* than these primes. However, even in the worst case scenario that $(X, N) \neq 1$, RSA still works! For, suppose $P \mid X$. Then we may assume $Q \nmid X$ (otherwise RSA breaks completely, since the plaintext would be corrupted upon reduction $\pmod{N}$). It follows

Date: March 8th, 2011.

that $X \in \mathbb{Z}_Q^\times$, whence by Euler's theorem, $X^{k\varphi(Q)} \equiv 1 \pmod{Q}$ for any integer k. But then

$$Y^d = (X^e)^d = X^{de} = X^{k\varphi(N)+1} = (1 + jQ)X = X + jQX \equiv X \pmod{N}$$

since $P \mid X$. It follows that the RSA decryption produces the plaintext, for *any* X which is not a multiple of N. And if X *is* a multiple of N, it will be totally obvious to Alice that this is the case (the ciphertext would be just 0) so she would be able to adjust her plaintext accordingly.

(4) The problem of factoring a large integer is presumed to be difficult (on empirical grounds: no one has resolved the problem after several decades of intense research). The security of RSA doesn't depend on factoring *per se*, but rather on computing $\varphi(N)$, and one might imagine that there is a clever way to compute $\varphi(N)$ without knowing the factorization of N. In general, this is an interesting open problem. However, in the specific case of RSA, where N is known to have exactly two prime factors P and Q, factoring N and computing $\varphi(N)$ are of comparable difficulty. To see why, we considered the following problem. Suppose that $N = 2021$, that you know that $N = PQ$ for some primes P and Q, and that $\varphi(N) = 1934$. Can you determine P and Q?

The first insight was that since $\varphi(N) = \varphi(P) \times \varphi(Q) = (P-1)(Q-1)$, we have two equations in two variables:

$$\begin{aligned} PQ &= 2021 \\ (P-1)(Q-1) &= 1934 \end{aligned}$$

Thus, we expect to be able to solve this system for P and Q. There are many ways to do this; the method suggested in the original RSA paper is in three steps: (a) expand $(P-1)(Q-1) = N - (P+Q) + 1$ and use the second equation to deduce $P+Q$; (b) use the identity $(P+Q)^2 - 4N = (P-Q)^2$ to determine $P-Q$; and (c) deduce the values of P and Q individually by adding or subtracting the two quantities $(P+Q)$ and $(P-Q)$.

Would this approach work if N is the product of three primes, rather than just two? This is a good exercise!