MAT 302: LECTURE SUMMARY

We began lecture with the following problem: find all solutions $x \in \mathbb{Z}_{21}^{\times}$ to the equation

$$x^{17} = 2. \tag{1}$$

We first noted that by Euler's theorem, $a^{\varphi(21)} = 1$ for any $a \in \mathbb{Z}_{21}^{\times}$. Since

$$\varphi(21) = \varphi(3 \times 7) = \varphi(3) \times \varphi(7) = 12$$

this means that $a^{12} = 1$ for any $a \in \mathbb{Z}_{21}^{\times}$. This immediately implies that $x^{17} = x^5$, so we need to solve the equation

$$x^5 = 2. \tag{2}$$

Of course, we could raise each element of $\mathbb{Z}_{21}^{\times}$ to the fifth and see whether we get 2, but there's a much better approach to the problem. Let $d = 5^{-1} \pmod{12}$. Then by definition, $5d \equiv 1 \pmod{12}$, which is the same as saying that $5d = 12k + 1$. It follows that if x is a solution to (2), then

$$(x^5)^d = 2^d.$$

But $(x^5)^d = x^{5d} = x^{12k+1} = (x^{12})^k x = x$ by Euler's theorem. Therefore, the only solution to (2) is

$$x = 2^d.$$

It remains only to evaluate $d = 5^{-1} \pmod{12}$. One can do this by trial and error fairly quickly, since it is particularly easy to check with a particular integer $\equiv 1 \pmod{12}$ is a multiple of 5. Another way to do this is to realize that since $(5, 12) = 1$, we can find integers a and b (using the Euclidean algorithm) such that $5a + 12b = 1$; reducing this equation modulo 12 shows that a is the desired inverse of 5. In any event, one sees that $d = 5$, whence the only solution to (1) is $x = 2^5 = 11$.

With the insight gleaned from this example, we approached in a more organized way the problem of solving

$$x^{13} \equiv 41 \pmod{69}. \tag{3}$$

Step 0: Recall Euler's Theorem

> For any $a \in \mathbb{Z}_{69}^{\times}$, we have $a^{\varphi(69)} = 1$. Since $69 = 3 \times 23$, we see that $\varphi(69) = 44$, whence $a^{44} = 1$ for any a.

Step 1: Determine $d = 13^{-1} \pmod{44}$.

> Use the Euclidean algorithm to find integers a and b such that $13a + 44b = 1$. Then a is the desired inverse. In this example, it turns out that $d = 17$. Make sure you understand how to derive this from the Euclidean algorithm!

Date: March 3rd, 2011.

Step 2: Raise both sides of (3) to the d^{th} power.

By definition of d, we have $13d = 44k + 1$ for some integer k. Therefore,

$$(x^{13})^d = x^{13d} = (x^{44})^k x \equiv x \text{ (mod 69)}$$

On the other hand, if x is a solution to (3), then $(x^{13})^d \equiv 41^d$ (mod 69). Thus, we see that

$$x \equiv 41^d \text{ (mod 69)}.$$

Step 3: Evaluate 41^d (mod 69).

In principle, this is a straightforward computation. Of course, in another example with enormous numbers this could still be a difficult task. We'll discuss this more next lecture.

1. RSA

In 1977, Rivest, Shamir, and Adleman used the above idea to generate an ingenious cipher, which has been called RSA ever since and (in slightly modified form) has been in use ever since. Here's how it works.

Bob chooses two gigantic primes P and Q, which he keeps private. He then announces publicly $N = PQ$ and e, a large positive integer of his choice (there are some restrictions on e which will be discussed below). If Alice wishes to send a message X to Bob, she first evaluates $Y = X^e$ (mod N), then sends Bob the ciphertext Y. How does Bob decode this? Simple: he receives the text Y from Alice, and he knows that whatever her original plaintext X is, it satisfies

$$X^e \equiv Y \text{ (mod } N). \tag{4}$$

Following our scheme from the first part of lecture, Bob first determines $d = e^{-1}$ (mod $\varphi(N)$), then raises both sides of (4) to the d^{th} power, which yields the congruence

$$X \equiv Y^d \text{ (mod } N)$$

Evaluating Y^d (mod N) thus allows Bob to recover the plaintext X.

But there's something fishy about this. Why couldn't Oscar do exactly the same thing? After all, both N and e are publicly known. It turns out there are only two essential difficulties:

(1) it turns out that determining $\varphi(N)$ is as hard as determining the factorization $N = PQ$, and it is unknown how to efficiently accomplish this;
(2) it is unknown how to efficiently solve the congruence (4) for X without knowing $\varphi(N)$.

Once $\varphi(N)$ is known, all other steps of the algorithm – calculating $d = e^{-1}$ (mod $\varphi(N)$) and Y^d (mod N) – are computationally feasible. This, then, is Bob's great advantage over Oscar: he alone has the knowledge of the factorization of N, allowing him to successfully implement the steps of the above algorithm.

There are several small technical points which should be mentioned. First, when Bob decodes Alice's message, he only determines it $\pmod N$. It is therefore crucial that $X \le N$. Thus, P and Q must be chosen so large that any conceivable message will be smaller than N. Second, to guarantee that the decoding algorithm works, e must be invertible $\pmod{\varphi(N)}$. This is easily arranged, since Bob knows $\varphi(N) = (P-1)(Q-1)$ and needs only choose any e relatively prime to this value.

RSA is entirely different from any cryptosystem we saw previously in this course: at no point do Alice and Bob have to share a secret key! This is a major advantage over any symmetric algorithm. However, there is a serious disadvantage to RSA (and all other publicly known asymmetric ciphers): it's significantly slower than current symmetric ciphers. In practice, people use the best of both worlds by employing RSA (or another asymmetric cipher) to do an initial key exchange, and subsequently employ AES (or another symmetric cipher) to communicate securely based on the shared key.

Finally, I wish to emphasize that RSA is not provably secure: any theoretical advance in factoring large integers or in solving congruences of the form (4) would have serious repercussions on the security of RSA. On the other hand, 30 years of intense research have failed to provide any serious mathematical attacks on RSA[*], which may be taken as evidence that the underlying mathematical problems are difficult.

[*] There have, however, been several ingenious attacks devised which capitalize on faulty implementation; see Chapter 7 of Paar-Pelzl.