

UMASS AMHERST MATH 471 FALL 2006, F. HAJIR

HOMEWORK 7: CRYPTOGRAPHY

1. AUSTIN, BASIL, AND DR. EVIL

Consider the following problem. A secret agent (namely *Austin Powers*, sometimes referred to as “A”) needs to send some highly sensitive information to his Boss (by the name of *Basil Expedition*, or “B”) but he knows that an evil person (*Mr. Evil*¹) will be intercepting the message and discover its highly sensitive contents. Austin and Basil come up with a brilliant plan to foil Dr. Evil. Before Austin goes on assignment, they come up with what is commonly called a “code” or “cipher” (technically known as a “cryptosystem”). According to this cipher, Austin will compose a message, then encode the message according to a previously agreed-upon *encryption key*, that is to say transform the message M into a coded message C . He will transmit C to Basil, who knows how Austin encrypted the message and so can reverse the process to go from C to M , the original intended message. When Dr. Evil intercepts the coded message C , he does not know the key that was used for encrypting the message and so he does not know how to decode the message – he will be left chewing his pinky with vexation.

For example, Austin and Basil agree ahead of time to substitute “The Bald Eagle” for every occurrence of “Doctor Evil” in the message and to use the phrase “Hot Dog with Relish and Onions” to stand for “Nuclear Bomb stolen from Kazakhstan,” as well as “Eat” for “Detonate” etc. Thus, when Basil receives the message “The Bald Eagle has managed to get his hands on a Hot Dog with Relish and Onions and is going to eat it at Dodger Stadium tomorrow morning!” he decrypts the message to find its true meaning, and swings into action immediately to prevent disaster by putting the Los Angeles Police on Alert. Meanwhile, Dr. Evil simply thinks that Austin has some bizarre interest in the dietary habits of endangered fowl.

What are some drawbacks of this cipher system? The biggest is that A and B have to exchange “the key” of how to encode/decode messages ahead of time. For example, you can imagine how lucky Austin and Basil were for having thought of phrases to substitute for “Nuclear Bomb stolen from Kazakhstan”! Once Austin enters Dr. Evil’s lair, he might realize that he has forgotten to invent a substitute for “Giant Laser” for example and by then it’s too late. One way around this is to create a whole scheme for encoding any message into a seemingly non-sensical message which the intended receive will be able to decode back into the original message. For example, if we simply shift each letter over a fixed number of letters, say three, then GIANT LASER becomes JLDQW ODVHU (if I didn’t make any mistakes). The decoding algorithm of course is very easy, you “subtract three” from each letter so to speak. I doubt this would fool the wily Dr. Evil for long, however. Even for more sophisticated systems, if Dr. Evil intercepts a few lengthy messages, he may be able to discover the key by performing a statistical analysis of the encoded messages. Namely,

¹DOCTOR Evil, I didn’t go to ten years of Evil Graduate School to be called “Mister!”

english text has certain statistical characteristics which will leave a fingerprint on the encoded message in many cipher schemes. For instance, the letters E and T are the most commonly occurring letters in the English language. By analyzing the ciphertexts (encoded messages), Dr. Evil can guess which letter has been substituted for E and do the same for T fairly easily. By continuing in this way, he will then be well on his way to deciphering the message. This is the method Edgar Allan Poe apparently used to baffle his readers by unscrambling the coded messages they sent him.

For Austin and Basil, there is a provably secure cryptosystem, by the way, called “The one-time pad.” First we convert the alphabet into numbers to make manipulation easier (this is not really needed but is convenient), for example, A is 01, B is 02, C is 03, etc. until Z is 26. So the word CAB becomes the number 131112. Now what we require is that Austin carry a notebook on each page of which a series of random numbers have been written every three lines. Austin writes his message above the first line of random numbers, then below that line he “adds” in columns and performs the addition modulo 26 to get a new sequence of numbers, which he then sends to Basil together with an indication of which page of the notebook he used to encrypt the message. Back at headquarters, Basil has the only other identical copy of the notebook with random numbers in it. He writes Austin’s encoded message on the appropriate page and now subtracts the random numbers. For example, let’s say the line of random numbers begins 2797713578192870... Austin converts BOMB to 02151302 which then becomes

```

02151302
27977135
03120611

```

In other words, BOMB becomes CLFK. Note that the first C turned into a C and the second B turned into a K. Thus, frequency analysis will not help Dr. Evil to decrypt the ciphertext; even if he is able to determine for sure one segment of the message, he won’t be able to use that success to determine the rest of the message! There is a cardinal sin of the one-time pad, however, that Austin has to watch out for. He absolutely cannot use any given one-time pad more than once! Reportedly, the hotline between the Presidents of the Soviet Union and the United States used the one-time pad technique.

2. DR. EVIL GOES GLOBAL

Now let us suppose we need to provide information security for a much larger population, not just between an agent and the boss. Suppose a network of banks need to communicate with each other regarding bank account transactions. It is way too cumbersome for each pair of banks to share keys. If there are only 1000 banks in the network for example, that would require the exchange and distribution of $\binom{1000}{2} = 999 \cdot 500 = 499500$ or nearly half a million keys! Since keys need to be changed periodically, that would be a big nightmare for the banks. In the sixties and seventies, this problem of key distribution became more and more of a pressing issue as more and more businesses (and military outfits) needed reliable and convenient security for transfer of information.

This need led to the development of a revolutionary idea in cryptography, one that had not been explored much in the previous 2000 years of this subject: the idea is nowadays called “Open Key Cryptography,” and the credit for it generally goes to a pair of Stanford

researchers, Diffie and Hellman. You might be thinking “Dude, that’s dumb, if the key is open for all to see, then intercepted messages can be decoded so your whole system is compromised.” The point is that while in principle that is true, in practice, the cryptosystem is built in such a way that it takes so long for the unauthorized interceptor (Dr. Evil) to decrypt your message that you don’t care: our sun will have gone supernova by then, so to speak and our little bank transactions will be of interest only to some intergalactic historians.

So, what is the Diffie-Hellman idea? They noted that although we have been talking about “the key” in the singular most of the time, there are in fact two keys in a cryptosystem, one for encrypting (coding) and the other for decrypting (decoding). We would like each user (each bank, say) in the network to publish their encryption key to the whole community so that anyone can send them a message without having to exchange keys ahead of time. The problem, as Diffie-Hellman saw it, is that if we publish the encryption key, then in principle, any interceptor can figure out the decryption key as well, in principle, because all they have to do is “reverse-engineer” the encryption method. In all the examples we have studied (not many), this is very easy to accomplish. But Diffie and Hellman thought that just because something always has been a certain way, it doesn’t mean that it **has** to be that way always. So they set out to find a cryptosystem with an *asymmetric key*, i.e. one where the encryption process is “easy” but the decryption process is “hard.” You might be thinking that even if they could find such a thing, then it would be no good, because Basil would have to be sitting there scratching his head for a long time to try to decrypt Austin’s message. Ah, but that is also an assumption that need not hold: the idea is that Basil might be privy to some secret information that will help him decrypt Austin’s message in a jiffy. (I had to look up the spelling of jiffy). The kind of process Diffie and Hellman needed is called a “trapdoor function” easy to enter from one side, but hard to break through from the other.

In number theory, we’ve got such functions in great supply, and shortly after Diffie and Hellman’s publication of their work, three researchers (Rivest, Shamir, and Adleman) used one of them to make an open encryption key cryptosystem (called RSA) that helps run the secure transfer of credit card numbers and all kinds of other information on the internet today. RSA made internet shopping possible. So, finally, let’s get down to the details of it.

Austin Powers wants to buy dental floss from `dentalfloss.com`. The latter website has chosen and made public their encryption key which is a pair of integers (n, e) . Before sending his credit card number M (or any message in English which Austin would first convert into a number by some accepted scheme, say according to the ASCII assignment), Austin encrypts it using the key (n, e) by computing

$$C = M^e \bmod n.$$

In other words, the encoded message C is just $C = \text{Rem}(M^e \div n)$, where Austin supplies the message M and `dentalfloss.com` provides the e and the n .² Now Austin sends C over the internet. Note that it is illegal in most circumstances for Dr. Evil to listen in on a phone line that Austin is using, but if Austin is broadcasting his credit card on the internet, he is not nearly as well-protected legally, so he needs to arm himself with cryptography. Okay, so Dr. Evil intercepts the coded message C – the credit card number has been scrambled. Can Dr. Evil unscramble it to find Austin’s credit card number?! In principle, yes, in practice, no (as long as `dentalfloss.com` has done a decent job choosing their key (n, e) .)

²By the way, the number n will be rather large, so that $M < n$. If the message M is a very large number, we chop it up into smaller bits and send them each separately.

Let's see why. What does Dr. Evil know? He knows n, e, C . He wants M . (He's the boss, he needs the info!) He knows that $M^e \equiv C \pmod n$ so all he has to do is to solve for M : easy, he just takes the e th root of C to get M right? Well, he has to do it modulo n . How do you do that?! Hmm, let's let Dr. Evil puzzle over that for a while, perhaps he can use sharks with lasers attached to their fins, and in the meantime let's visit the headquarters of dentalfloss.com

How did these guys choose n ? Well, following RSA's advice, they chose two large random primes p and q . What "large" really means changes with the situation. I think for today's technology, you'd want to choose p and q to have at least a hundred digits to feel safe. If you choose them to have 200 digits each, you'll sleep even better at night. Anyway, after choosing p and q , you put $n = pq$. Multiplying is easy, so figuring out n takes hardly any time at all for a computer. BUT, AND THIS IS THE BIG BUT THAT MAKES RSA WORK!! If someone is just given n without knowing p and q first, it is COMPUTATIONALLY HARD to figure out p and q though in principle, it's not a big deal, you just factor instead of multiply. This is a great trap-door function: think of the two primes as two pieces of metal; if multiplying them together means welding them together, that can be done pretty easily if you know your way around a metal shop, but then if I give you the welded metal and ask you to separate the fused metal, your job is a lot harder now, even with heavy equipment at your disposal. So, the secret, privileged information that the online shop has is the knowledge of p and q . How can those be useful for figuring out M from $C = \text{Rem}(M^e \div n)$?

To answer that, recall our good friend the lion-hearted Leonhard Euler. Inspired by Fermat, Euler proved that if $\gcd(C, n) = 1$, then $C^{\varphi(n)} \equiv 1 \pmod n$. Now, to compute $\varphi(n)$ is easy for dentalfloss.com because they know that $\varphi(n) = \varphi(pq) = (p-1)(q-1)$. But for Dr. Evil, it's just as hard to compute $\varphi(n)$ as it is to factor n completely. (See the exercises). Why does knowing $\varphi(n)$ matter? Well, consider this little argument: How did dentalfloss.com choose the number e anyway? We haven't said yet. They chose e to be a largish number coprime to $\varphi(n) = (p-1)(q-1)$ (We're talking big numbers here, so in fact, it's quite hard to make $\gcd(e, (p-1)(q-1)) > 1$ if you choose e randomly. Since $\gcd(e, \varphi(n)) = 1$, by Bézout, we can find an integer d such that $de \equiv 1 \pmod{\varphi(n)}$. Okay, that means that $de = 1 + k\varphi(n)$ for some integer k . Ready for the kicker? to find M , we have to extract an e th root of C , right? But we'll accomplish this by actually raising C to the d th power! Why does this work? Remember Euler says $C^{\varphi(n)} \equiv 1 \pmod n$, so

$$C^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{1+k\varphi(n)} \equiv M \pmod n!!$$

Presto! Ta da! Voilà! The operation C^d is not hard to carry out, by the way, because we do everything modulo n .

Let's review what dentalfloss.com does: they choose two large random primes p, q (they shouldn't be too close to each other!): this is not hard to do, you just pick a random odd number and there is a fast way to check if it's prime; if it's not, you then add 2 to it and check again; pretty soon, you'll hit a prime probably; if you add 2 a thousand times and you don't get a prime, just start over. You do this twice to get p and q , then you take $n = pq$ and you choose e then use Bézout to find d such that $de \equiv 1 \pmod{(p-1)(q-1)}$. In practice, you probably want to choose d to be a largish prime and then use Bézout to find e . You keep d secret, you destroy any record of p, q and you publish n, e for all to see. That's all there is to it. To decrypt a received coded message C , you raise it to the d th power and take

the remainder modulo n , i.e. $M = \text{Rem}(C^d \div n)$. For the encoder, life is easy too, because $C = \text{Rem}(M^e \div n)$.

For Dr. Evil, life sucks because he knows C, n, e so in principle he should be able to figure out M but in order to do so, he needs to know d i.e. the inverse of e modulo $\varphi(n)$ but to figure out $\varphi(n)$, he needs to factor n and that is HARD, needel-in-a-haystack-hard.

Let's do two examples to see all this in practice: for the first one, the numbers are kept deliberately very small so the system is not secure, but you can follow all the calculations easily on a calculator; in the second, I'll use bigger numbers so you get a feel for how hard life might be for Dr. Evil, though I'm not even approaching 100-digit p and q .

Example 1. You want to send a secure text message to your bud about when to meet to study. We'll keep the message short, it's just $M = 10$. Now you look up your bud's encryption key, say it's (n, e) where $n = 23501$ and $e = 67$. To compute the coded message C , we must find $C = M^e \pmod n$. To do this, let's note that $e = 67 = 1 + 2 + 64$ is how it decomposes into a sum of powers of 2, so to compute M^e it's enough to compute M^1, M^2 and M^{64} then multiply all these together. We find by successive squaring six times that

$$M^2 = 100, M^{64} = 9587, \text{ so that } C = M^{67} = MM^2M^{64} = 22093.$$

We're doing everything here modulo n of course. Now we send $C = 22093$ to our bud. Our bud has the secret information that $n = 71 \cdot 331$ where 71 and 331 are primes, so that $\varphi(n) = 70 \cdot 330 = 23100$, and also the information that $de \equiv 1 \pmod{\varphi(n)}$ where $d = 3103$. Now our bud has the seemingly unpleasant task of computing

$$M \equiv C^d \equiv 22093^{3103} \pmod n.$$

which may seem daunting. But it's not bad at all, we compute the base 2 expansion of 3103 to be

$$3103 = 1 + 2 + 4 + 8 + 16 + 1024 + 2048.$$

(You just take away as big a power of 2 as you can until you run out of room). By successive squaring eleven times and then multiplying we find

$$M \equiv 22093^{3103} \equiv 10 \pmod{23501},$$

so we're supposed to meet at 10, and this highly sensitive information has been kept safe!

Example 2. Let's say the message is MEETATDCATELEVEN. Using the 01 is A, 02 is B etc scheme, we convert this to a number $M = 13050520012004030120051205220514$. We look up our bud's published encryption key and find it is

$$\begin{aligned} n &= 86235320551695174717678592798468704942601 \\ e &= 19287013241 \end{aligned}$$

Our coded message then is

$$C = M^e \pmod n = 36818249479175534712099148820495662970760.$$

To decrypt it, our bud knows that $de \equiv 1 \pmod{\varphi(n)}$ where

$$d = 18604104988918776638073808824707077774337.$$

So,

$$M = C^d \pmod n = 13050520012004030120051205220514$$

which converts to MEETATDCATTEN.

1. The ciphertext $C = 5859$ was obtained from the RSA algorithm with $n = 11413$ and $e = 7467$. Using the factorization $n = 101 \cdot 113$, find the original message (also called plaintext).

2. Stefanie and Vivien have previously agreed on the choice of a large prime p (they both know what p is). Stefanie has secret information which she converts to a number m , $1 \leq m \leq p-1$. Stefanie wants to convey m to Vivien without actually telling her the number over the phone. Here is what they do. Stefanie chooses an integer s coprime to $p-1$ and Vivien chooses an integer v coprime to $p-1$; they don't talk about s and v . Stefanie computes $c = \text{Rem}(m^s \div p)$ and tells Vivien what c is. Vivien then computes $d = \text{Rem}(c^v \div p)$ and tells Stefanie what d is. Now Stefanie uses Bézout to compute s' such that $ss' \equiv 1 \pmod{p-1}$, then computes $e = \text{Rem}(d^{s'} \div p)$ and tells Vivien what e is. Explain what Vivien must now do to determine m as well as why it works. If Kevin has been listening, he would have heard c, d, e : can he figure out m easily? What if he happens to know p also?

3. Ara has been using RSA with encrypting key (n, e) to receive messages from Rob and is happy with it, but he gets a little paranoid so he decides he will double his security by choosing two encrypting exponents e_1, e_2 (keeping the same n) and requiring Rob to encrypt his message M to him twice, first by computing $C_1 = M^{e_1} \pmod{n}$ and then re-encrypting C_1 by putting $C_2 = C_1^{e_2} \pmod{n}$; Rob is then supposed to send C_2 to Ara. Does this scheme double Ara's security? Why or why not?

4. Let p, q be distinct odd primes, and put $n = pq$. Suppose m is an integer coprime to n . Let $f = \varphi(n)/2$.

(i) Show that $m^f \equiv 1 \pmod{p}$ and $m^f \equiv 1 \pmod{q}$.

(ii) Show that $m^f \equiv 1 \pmod{n}$.

(iii) Show that if $ed \equiv 1 \pmod{f}$, then $m^{ed} \equiv m \pmod{n}$.

Explain how (iii) can be used to simplify RSA a tiny bit.

5. Explain why the exponents $e = 1, 2$ should not be used as the encryption exponent in RSA.

6. You are trying to factor the number $n = 642401$. Suppose you discover that

$$516107^2 \equiv 7 \pmod{n} \text{ and } 187722^2 \equiv 2^2 \cdot 7 \pmod{n}.$$

Use this information to factor n .

7. Show that if $x^2 \equiv y^2 \pmod{n}$, and $x \not\equiv \pm y \pmod{n}$, then $\text{gcd}(x+y, n)$ is a non-trivial factor of n , i.e. $\text{gcd}(x+y, n) | n$ and $1 < \text{gcd}(x+y, n) < n$.

8. Let $n = pq$ be the product of two distinct primes.

(i) Let k be a multiple of $\varphi(n)$. Show that if $\text{gcd}(m, n) = 1$, then $m^k \equiv 1 \pmod{p}$ and $m^k \equiv 1 \pmod{q}$.

(ii) Let k be as in (i), but now we do not assume that $\text{gcd}(m, n) = 1$. Show that $m^{k+1} \equiv m \pmod{p}$ and $m^{k+1} \equiv m \pmod{q}$.

(iii) Let e, d be the encryption, decryption exponents for RSA with modulus n . Show that $m^{ed} \equiv m \pmod{n}$ for all integers m . This shows that we do not need to assume that $\gcd(m, n) = 1$ where m is the message (plaintext) in order for RSA to work properly.

(iv) For a fixed $n = pq$, how likely is it that $\gcd(m, n) = 1$ when you pick a random integer m ?

9. RSA Signatures Suppose I want to send my 471 students their grades at the end of the semester via e-mail, and the students want to have a way to verify that the e-mail they will receive will actually have been sent by me and not by some bozo who hacked into my computer. Here is a scheme for doing it. At the final, I put on the board for all to see my chosen RSA encryption key (n, e) as usual, with $n = pq$, p, q being distinct primes and $1 < e < \varphi(n)$ with $\gcd(e, \varphi(n)) = 1$. I chose p, q to start with, then deleted all knowledge of them from my computer right after I calculated d such that $de \equiv 1 \pmod{n}$. I keep d memorized and I only tell the students n and e ; the number d is not stored in my computer. Now when I send a student (say Alden) his grade, I do it by sending him a ten-digit number M where the first nine digits are Alden's social security number and the tenth digit is 9,8,7,6,5,4,3,2,1,0 according to whether he got A,A-,B+,B-,C+,C-,D,F. But I don't just send Alden M , I send him the pair (M, C) where $C = M^d \pmod{n}$. To verify that it's really Farshid sending the message, Alden calculates $F = C^e \pmod{n}$.

(i) After he calculates F , how does Alden decide whether the message came from the real Farshid or from a faker and why does he decide that way?

(ii) Suppose Alden decides the real Farshid did send the message, but now he hacks into my machine (tsk tsk) in order to send a fake grade to Matt. Let's say he even knows Matt's social security number. If Alden cannot factor n , can he fool Matt? (You can assume that Matt remembers his own social security number).

Extra Credit

If you want to win some cold hard CASH (not from me), check out

<http://www.rsasecurity.com/rsalabs/node.asp?id=2094>

If you beat one of the challenge numbers, I'll even give you extra credit ... but you'll have so much money you won't care.