

UMASS AMHERST MATH 471 FALL 2006, F. HAJIR

SOLUTIONS FOR HOMEWORK 7: CRYPTOGRAPHY

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).

We find d such that $de \equiv 1 \pmod{100 \cdot 112}$ by using Bézout; it turns out to be $d = 3$. We then compute $M \equiv C^d \equiv 5859^3 \equiv 1415 \pmod{n}$ so the message is 1414.

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?

This is the story of the suitcase with the 2 locks. Stephanie's lock "is" s and her key is s' , etc. Anyway, we have

$$e \equiv d^{s'} \equiv (c^v)^{s'} \equiv c^{vs'} \equiv m^{svs'} \equiv (m^{ss'})^v \pmod{p}.$$

But since $ss' = 1 + k(p-1)$ for some integer k , $m^{ss'} = m \cdot m^{k(p-1)}$, and by Fermat, $m^{k(p-1)} = (m^{p-1})^k \equiv 1^k \equiv 1 \pmod{p}$. Thus, $m^{ss'} \equiv m \pmod{p}$. Now Vivien applies the same trick, i.e. to recover m from $e \equiv m^{ss'v} \equiv m^v \pmod{p}$, all Vivien has to do is to raise both sides to the v' power:

$$m^{vv'} \equiv m \equiv e^{v'} \pmod{p}.$$

Kevin heard c, d, e but he doesn't know m, s, v , so even if he knows p , in the equation $c \equiv m^s$, he doesn't know either m or s , all he knows is c and there are potentially very many possible pairs (m, s) that give the same $c = m^s$.

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?

This scheme doesn't really change the level of security for Ara. Namely, $C_2 = (M^{e_1})^{e_2}$ or $C_2 = M^{e_1 e_2}$. Thus his new system is equivalent to RSA with encryption key (n, e) where $e = e_1 e_2$.

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 $m^f \equiv 1 \pmod{q}$.

The idea is that $(p-1)/2$ and $(q-1)/2$ are both integers, so $f = \phi(n)/2 = (p-1)(q-1)/2$, is a multiple of $p-1$ as well as a multiple of $q-1$; this'll ensure that we will win by Fermat, namely,

$$m^f = (m^{p-1})^{(q-1)/2} \equiv 1^{\text{an integer}} \equiv 1 \pmod{p}$$

. The exact same argument with p, q reversing roles gives the other congruence.

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

By Chinese Remainder Theorem, $x \equiv y \pmod{n}$ if and only if $x \equiv y \pmod{p}$ and $x \equiv y \pmod{q}$ so we win by (i).

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

We have $ed = 1 + kf$, $k \in \mathbb{Z}$, so $m^{ed} = m \cdot (m^f)^k \equiv m \pmod{n}$.

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

Well, now to find the decryption key d , we compute the inverse of e modulo $\varphi(n)/2$ instead $\varphi(n)$ so that saves us just a teeny bit of time. [But a small savings over a span of a million computations can mean a lot to our bottom line].

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

Well, $e = 1$ is really bad, because then $M = C$ so not much scrambling of the message has taken place, eh? Dr. Evil would have a field day. Now for every integer $n > 2$, $\phi(n)$ is even (why?!) and since $\gcd(e, \phi(n)) = 1$ is needed for the decryption key to exist in the standard implementation of RSA, $e = 2$ (or e even) is not a good choice. You could modify the search for the decryption key by looking for an inverse of e modulo $\varphi(n)/2$ only as in the previous problem, but by the previous problem, even $\varphi(n)/2$ is even. Another reason that $e = 2$ isn't so good, is that you are challenging Dr. Evil to find a square root modulo n . He could search for an integer k such that $C + kn$ is a perfect square m^2 and then $m \pmod{n}$ would be the message. Since perfect squares are more common than cubes, etc. $e = 2$ is not such a great choice even if you could make it work.

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 .

Since n is odd, 2 is invertible modulo n , so we can multiply the second congruence by the inverse of 2 to get $93861^2 \equiv 7 \pmod n$. Now we have two squares being congruent mod n ,

$$516107^2 \equiv 93861^2 \pmod n,$$

or $n|(x^2 - y^2) = (x - y)(x + y)$ with $x = 516107$, $y = 93861$. We compute $\gcd(n, x + y)$ say and find it is 569. If we had used $x - y$, we'd get the other factor 1129. Anyway, $642401 = 1129 \cdot 569$, yea verily.

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

Since $x \not\equiv \pm y \pmod n$, n does not divide $x \pm y$ i.e. n is not a divisor of $x + y$, nor of $x - y$. Thus, $\gcd(x + y, n)$ cannot be n and nor can $\gcd(x - y, n)$ be n . On the other hand, if $\gcd(x + y, n) = 1$, then $n = \gcd((x + y)(x - y), n) = \gcd(x - y, n)$ gives a contradiction. So we must have $\gcd(x + y, n) > 1$. We have shown that $1 < \gcd(x + y, n) < n$ and clearly $\gcd(x + y, n)$ is a divisor of n .

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

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

We have by Fermat, since $\gcd(m, n) = 1$ implies $p \nmid m$, that

$$m^k = m^{\varphi(n)r} = m^{(p-1)(q-1)r} = (m^{p-1})^{(q-1)r} \equiv 1^{\text{an integer}} \equiv 1 \pmod p.$$

The other one follows by symmetry.

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

By the first form of Fermat, $m^{k+1} = m^{(p-1)(q-1)r+1}$. In the exponent of the previous expression, we will isolate the multiples of p as in

$$(p-1)(q-1)r+1 = p[(q-1)r] - [(q-1)r] + 1.$$

When we raise m to the above power, $m^p \equiv m \pmod p$ will simplify things a lot, as in

$$m^{p[(q-1)r]-[(q-1)r]+1} \equiv (m^p)^{(q-1)r} m^{-[(q-1)r]} m \equiv m^{(q-1)r} m^{-[(q-1)r]} m \equiv m \pmod p.$$

No sweat. The other congruence follows by symmetry.

(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.

Once again, we apply (ii) together with Chinese Remainder.

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

Recall that for a fixed n , $\gcd(m, n)$ depends only on $m \bmod n$. There are n integers m in $[0, n - 1]$, and $\varphi(n) = (p - 1)(q - 1)$ of them are coprime to n , thus the probability that a random integer is coprime to n is

$$\frac{\varphi(n)}{n} = \frac{(p-1)(q-1)}{pq} = \left(1 - \frac{1}{p}\right) \left(1 - \frac{1}{q}\right).$$

If p, q are large, this is pretty close to 1.

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?

We're supposed to have $C = M^d \pmod n$ and if that is true, then $C^e = M^{de} = M \pmod n$, so Alden checks to see if $F = M \pmod n$. If that's the case, then the person sending the message knows d so it must be Farshid. If it's not the case, then someone is trying to mess with us and we distrust the message.

(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).

No, he shouldn't be able to fool Matt because even though he knows M , he doesn't know d so he doesn't know how to compute C .