

Homework

Problems beginning CW are primarily computer work. On homework problems not involving the computer, show enough work so that I can see what you're doing. Obviously you won't show everything you did on your calculator on longer, repetitive problems. On homework involving the computer, there will often be little work to show.

Hist-1. Encrypt the following message. Playfair cipher system, key SUBHARMONIC, plaintext: CHRISTIANS

Hist-2. Decrypt the following message. Playfair cipher system, key FACETIOUSLY, ciphertext: HQSMLFTO

Hist-3. Decrypt the following message. ADFGVX ciphersystem, key CREAMY, ciphertext: AFDFVXFAXXDVGXGXGXVGXXFG

Hist-4. You are an ancient Greek and you intercept a thin strip of paper with the following letters. Decrypt the message. If you prefer not to do this by hand, the message is also in a file called greek.html

ATAIWSRTSIPTSI LAHWNETHLINRHGROHDNDOERRSEBEJWNOONSUAESACDAEL
FRINKARNLAKTASNEDTRSGNIDTSHIOAGTCHTANUSLSAEHTTTPEWSSEGOAIRIT
MHUTFNOTSAOAHIGNHHLRRESSAHILNDWHHJISEOSAAOUSBRFHTNRTTAFAlEDE

In real life it would look like $\begin{array}{|c} A \\ T \\ \vdots \end{array}$

NT-1. Find $\gcd(720, 450)$ i) using the Euclidean algorithm, ii) by factoring each.

NT-2. For each of the following pairs of numbers, find the gcd using the Euclidean algorithm and then write the gcd as an integer linear combination of the pair: i) 21, 30, ii) 126, 129.

NT: 3 - 6: if you're working Mod m or in $\mathbf{Z}/m\mathbf{Z}$ all answers should be between 0 and $m - 1$.

NT-3. Find the multiplicative inverses to all elements in $\mathbf{Z}/13\mathbf{Z}^*$. Your answer should look like: $1^{-1} = 1$, $2^{-1} = \dots$

NT-4. Use the Euclidean algorithm to find 35^{-1} in $\mathbf{Z}/73\mathbf{Z}$. What's 35^{-2} in $\mathbf{Z}/73\mathbf{Z}$? (We could say these are $35^{-1} \pmod{73}$ and $35^{-2} \pmod{73}$.)

NT-5. Make an addition table for $\mathbf{Z}/6\mathbf{Z}$. Make a multiplication table for $\mathbf{Z}/6\mathbf{Z}$.

NT-6. Make a multiplication table for $\mathbf{Z}/8\mathbf{Z}^*$. Make a multiplication table for $\mathbf{Z}/10\mathbf{Z}^*$.

NT-7. Show that the sum of the squares of 2 odd integers is not the square of an integer. Hint: work Mod 4.

NT-8. Show $X^3 + X + 1$ is always odd if X is an integer.

NT-9. Consider the 10 functions $f_n(x) = nx$ from $\mathbf{Z}/12\mathbf{Z}$ to $\mathbf{Z}/12\mathbf{Z}$ where $n = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$ (consider them one at a time, the first is $f_2(x) = 2x$). How many elements are in the

range (the range is the set of outputs) of f_n for these 9 n 's? Come up with a formula predicting these values based on n and 12. So for example, the elements of $\mathbf{Z}/12\mathbf{Z}$ are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. The function $2x$ sends them to 0, 2, 4, 6, 8, 10, 0, 2, 4, 6, 8, 10, so the size of the range is 6. The function $3x$ sends them to 0, 3, 6, 9, 0, 3, 6, 9, 0, 3, 6, 9 so the size of the range is 4. Do this up to 11.

NT-10. Find i) $\varphi(32)$, ii) $\varphi(100)$, iii) $\varphi(3600)$, iv) $\varphi(35)$, v) $\varphi(77)$.

NT-11. If $n = pq$ where p and q are different primes, find a formula for $\varphi(n)$ in terms of p and q .

NT-12. This is an incredibly difficult problem: Solve the MU problem. All strings allowed are made up of the letters M,I,U. You start with the string MI. You can apply any of the following 4 rules to change your string. You can use the rules repeatedly, and in any order you like. 1) If last letter of string is I can add a U at the end (so MI can become MIU). 2) Suppose you have Mx (where x is any string of M,I,U's), can change to Mxx. (from MIU can get MIUIU, from MUM can get MUMUM). 3) If III occurs in string can replace with U (from UMIIIMU can get UMUMU, but can't change IIMII with this rule). 4) If UU occurs in a string you can drop it. Puzzle: can you get the string MU? Don't spend more than 30 minutes on this.

NT-13. Solve the following (if possible): i) $3x \equiv 2(\text{Mod } 14)$, ii) $3x \equiv 2(\text{Mod } 15)$, iii) $3x \equiv 6(\text{Mod } 15)$, iv) $37x \equiv 51(\text{Mod } 100)$.

NT-14. For which positive integers m is the following statement true? $27 \equiv 5(\text{Mod } m)$?

NT-15. Reduce $3^{1,000,000}(\text{Mod } 7)$.

NT-16. Find all positive integers n with $\varphi(n) \leq 12$. You need not prove your result.

NT-17. For which positive integers n is $\varphi(3n) = 3\varphi(n)$? No need to explain, just find the pattern. I want you to describe the set of ALL positive integers with this property; don't just give me a few examples.

SmC-1. Use frequency analysis to decrypt the following. I used a simple shift transformation: $C \equiv P + b(\text{Mod } 26)$. I have inserted spaces and some punctuation. Z CVRIEVU KYV WFCCFNZEX AFBV KYV CRJK KZDV Z NFIBVU R IVRC AFS. NYRK UZU KYV WZJY JRP NYVE YV YZK YZJ YVRU? URD

SmC-2. Encrypt TRANSFER FUNDS using $C \equiv 7P + 11(\text{Mod } 26)$.

SmC-3. You intercept NGIPNGZO YGWB DQIQP GE. You know it was encrypted with $C \equiv aP + b(\text{Mod } 26)$ (so you'll use $P \equiv a'C + b'(\text{Mod } 26)$ to decrypt). You know this sender ends every message with the plaintext OK. i) Find the decrypting keys a' and b' . ii) Decrypt the message. iii) Find the encrypting keys a and b and encrypt COME SOON OK.

SmC-4. If I used a 38 symbol alphabet (the letters, the digits, a period and a blank space), how many different key-pairs (a, b) would there be for affine encryption: $C \equiv aP + b(\text{Mod } 38)$?

SmC-5. Let's use a 27-letter alphabet consisting of the standard 26 letters and a blank space, which we'll denote $-$. We encode the letters in the usual way with $A = 0, \dots, Z = 25$,

and $- = 26$. Then we encode digraphs as in class (though for a 27-letter alphabet). You intercept some ciphertext encrypted with $C \equiv aP + b(\text{Mod } 729)$. From frequency analysis, you determine that the ciphertexts corresponding to the two most frequent digraphs in English (namely $S-$ and $-T$) are NG and KX, respectively. Decrypt this piece of the ciphertext: TCUGARXKOK. Don't write anything up for the following three sentences. Observe the last two plaintext and ciphertext digraphs. Think about whether that will always happen by thinking simply Mod 27. For comparison, note that this didn't happen with the second and fifth digraphs. (Note I found another document that says that the two most common digraphs are $E-$ and $S-$.)

LM-1. Read the computer labs manual. (This homework problem has no writeup).

CW-LM-2. Use CryptoSoft to encode 'all is well' (without apostrophes) as an integer. Put it in the file t.txt (in newer Windows versions you just create a file called t). (This computer work problem has no writeup).

CW-LM-3. Use GP-PARI to compute the following:

Let a be the nextprime above 2^{50} , b be the nextprime above 3^{50} and $m = 11^{27}$.

i) Reduce $a^b \text{Mod } m$ (remember to do $\text{Mod}(a,m) \wedge b$). Put it in your writeup.

ii) Find $a^{-1} \text{Mod } m$. Put it in your writeup.

iii) Confirm a^{-1} is right by doing the following. Lift $a^{-1}(\text{Mod } m)$ to an integer that is not inside a modular expression. Multiply this integer by a , subtract 1 and then divide by m . You'd better get an integer (think about why), what is it? Put that integer in your writeup.

iv) Find $\text{gcd}(m, 8689142)$. Put it in your writeup.

v) Read file t.txt and subtract the number 15735122002826882864906240 from it and write it to the file u.txt. **REMEMBER: to get out of PARI** you type $\backslash q$. No need to write anything up for this one.

vi) Use CryptoSoft to decode the number in u.txt to plaintext and put it in your writeup.

FF-1. 2 is a generator of \mathbf{F}_{13}^* . i) For $i = 1$ to 12 compute 2^i in \mathbf{F}_{13}^* .

ii) For each element $b (= 2^i)$ of \mathbf{F}_{13}^* , find the smallest positive integer r so that $b^r = 1$ (0 is not positive). Make a chart $i, b = 2^i, r$.

In the chart, i is called the discrete $\log_2(b)$ since $2^i = b$.

iii) Find a formula for r given i .

iv) Multiply $3 \cdot 12$ in \mathbf{F}_{13}^* . Find $\log_2(3)$, $\log_2(12)$ and $\log_2(3 \cdot 12)$ (those are **DISCRETE logs**, so $\log_2(3) = 4$). Find $9 \cdot 10$ in \mathbf{F}_{13}^* . Find $\log_2(9)$, $\log_2(10)$ and $\log_2(9 \cdot 10)$. Find $11 \cdot 5$ in \mathbf{F}_{13}^* . Find $\log_2(11)$, $\log_2(5)$ and $\log_2(11 \cdot 5)$. Recall with usual (non-discrete) logs, that $\log(ab) = \log(a) + \log(b)$. For discrete logs, how does it seem that the log of the product is related to the two other logs? So if $a, b \in \mathbf{F}_{13}^*$, how are $\log_2(a)$, $\log_2(b)$ and $\log_2(a \cdot b)$ related?

FF-2i) 2 is not a generator of \mathbf{F}_{17}^* ; find the smallest one, call it g .

ii) What is the smallest positive power i of g that gives you 2 (i.e. what's $i = \log_g(2)$)? What power r of 2 gives you 1? Is r as predicted by the formula in FF-1-iii)? (generalized from $p = 13$ to $p = 17$).

For problems SC-1 and SC-2, our random bit generator will be the first one I taught you. 83 is prime and 2 generates \mathbf{F}_{83}^* . $2 \cdot 83 + 1 = 167$ is prime and 5 generates \mathbf{F}_{167}^* . Our symmetric/shared/secret key is $k=7$. We have $s_1 \equiv g^k \equiv 5^7 \pmod{167}$. For $i = 1, 2, \dots$ we have $s_{i+1} \equiv s_i^2 \pmod{167}$. For $i = 1, 2, \dots$ we have $k_i \equiv s_i \pmod{2}$ where we consider s_i to have been reduced Mod 167 already. To start you off, $s_1 = 136$, $s_2 = 126$, $s_3 = 11$ and so $k_1 = 0$, $k_2 = 0$, $k_3 = 1$. As a check, you should have $s_{16} = 62$. We will use the same random stream for problems SC-1 and SC-2.

SC-1. Read the paragraph above. Stream cipher: The ciphertext is 0101 0101 1110 1001 (I put in spaces to make it easier to read). Those are the bits $c_1c_2 \dots c_{16}$. The plaintext will be called $p_1p_2 \dots p_{16}$. The rule is $p_i = c_i \oplus k_i$ for $i = 1, 2, \dots, 16$. Decrypt and decode using ASCII.

SC-2. Self-synchronizing stream cipher: The ciphertext is 0110 1101 1111 0111 Those are the bits $c_1c_2 \dots c_{16}$. The plaintext will be called $p_1p_2 \dots p_{16}$. We will define $p_{-1} = p_0 = 0$ which are not part of the plaintext but show up in the computations. The rule is shown below for $i = 1, 2, \dots, 16$. Decrypt and decode using ASCII.

$$p_i = k_i \oplus c_i \oplus \begin{cases} p_{i-2} & \text{if } p_{i-1} = 0 \\ p_{i-3} & \text{if } p_{i-1} = 1 \end{cases}$$

FF-4. Factor (mod 2) all eight polynomials of the form $x^3 + b_2x^2 + b_1x + b_0$ completely, where $b_i \in \{0, 1\}$. For example, $x^3 + x^2 = x^2(x + 1)$, now you do the other 7.

FF-5. Find the first 7 powers of x^2 in $\mathbf{F}_8 = \mathbf{F}_2[x]/(x^3 + x + 1)$. Is x^2 a generator of \mathbf{F}_8^* ?

FF-6. $x^4 + x + 1$ is irreducible over \mathbf{F}_2 (trust me). (a) Find $(x^3 + x + 1) \cdot (x^3 + x^2 + 1)$ in $\mathbf{F}_{16} = \mathbf{F}_2[x]/(x^4 + x + 1)$. (b) Find the first 15 powers of x in \mathbf{F}_{16} . (c) Does x^3 generate \mathbf{F}_{16}^* ?

FF-7. In $\mathbf{F}_{32} = \mathbf{F}_2[x]/(x^5 + x^2 + 1)$ find the multiplicative inverse of $x^3 + x^2 + 1$.

AES-1. Read section 10 of the AES handout.

AES-2. Verify by hand that SBOX(1100) = 1100. By this I mean first invert $x^3 + x^2$ and turn that into a nibble. Turn that nibble into an element $N(y)$ of $\mathbf{F}_2[y]/(y^4 + 1)$ and find $(y^3 + y^2 + 1)N(y) + (y^3 + 1)$. Turn it back into a nibble; it ought to be 1100.

AES-3. We know that MC^{-1} is multiplying by $c(z)^{-1} = xz + (x^3 + 1)$ in $\mathbf{F}_{16}[z]/(z^2 + 1)$. It would be nicer to have this in a form

$$\begin{bmatrix} b_0b_1b_2b_3 \\ b_4b_5b_6b_7 \end{bmatrix} \text{ to } \begin{bmatrix} b_3 \oplus b_5 & & b_1 \oplus b_4 \oplus b_7 \\ & b_2 \oplus b_4 & & b_0 \oplus b_6 \oplus b_7 \end{bmatrix}.$$

So multiply $(b_0x^3 + b_1x^2 + b_2x + b_3)z + (b_4x^3 + b_5x^2 + b_6x + b_7)$ with $xz + (x^3 + 1)$ in $\mathbf{F}_{16}[z]/(z^2 + 1)$ by hand to fill in this table. (I gave you half of them as a check.)

AES-4i) Expand the key 1011 1101 0010 0101 using Simplified AES. As a check, the last four bits of the expanded key should be 1100 (that is $k_{44}k_{45}k_{46}k_{47}$).

ii) Find $c(z)^{-1}K_1$. (Hint, that's really $MC^{-1}K_1$).

iii) Decrypt the string 0111 0001 0011 1001 with the key 1011 1101 0010 0101 using Simplified AES. Use $A_{K_0} \circ SR^{-1} \circ NS^{-1} \circ A_{c(z)^{-1}K_1} \circ MC^{-1} \circ SR^{-1} \circ NS^{-1} \circ A_{K_2}$. (Remember that means A_{K_2} is the first step. Just do ECB Mode). Then decode to a pair of ASCII characters (the message will tell you that you got it right). Give the homework grader a break and write this up neatly with little notes about which step you're doing.

As a midway check: After MC^{-1} , you should have 1000 0001 1100 1011.

AES-5. Encrypt the 32-bit message COEN (in ASCII) using Simplified AES and Cipher Block Chaining (CBC) with initialization vector 1000 1101 0000 1011 (IV) and key 1100 1011 1111 1110. Here are some checks. CT1 is 1000 0110 1101 110*. The final ciphertext (CT2) is *1* 1 *0*1 *0*1 *001. (I'm not telling you the bits under the *'s)

AES-6. Decrypt 1100 0010 1011 0011 0010 0101 1011 0011 using the output feedback (OFB) stream cipher. Use the same key as in the last problem, namely 1100 1011 1111 1110. Use the initialization vector 1100 1110 0100 0100 (Look in the last problem and you'll see that this string was the XOR of the first two bytes of plaintext and the initialization vector from that problem. In that last problem, that XOR is what you encrypted. That's the first step of this problem. So that saves a LOT of work. So you need not show any work to get z_1). Decode to ASCII characters.

AES-7. Explain the middle-in-the-middle attack that shows that triple-DES with three different keys is not much safer than triple-DES with two different keys. For simplicity, assume that the former is $CT = E_{K_3}(E_{K_2}(E_{K_1}(PT)))$ and the latter is $CT = E_{K_1}(E_{K_2}(E_{K_1}(PT)))$.

NT-18. Use repeated squares to reduce $17^{53} \pmod{97}$. Do this problem twice. Use the calculator method first. Then do the computer algorithm (though with your calculator).

RSA-1. Do this problem by hand and calculator. Don't use PARI (though you could check work with it). For RSA, your p is 7, your q is 97 and your e is 257. Find $\varphi(n)$ and d and include them in the writeup. We will use RSA to agree on a key for the affine $C \equiv aP + b \pmod{26}$ cipher. I encode the key(s) a and b by computing $26a + b$ (we didn't do that with the key when discussing this affine cipher; we did it with digraphs when working mod 26^2 , which we're NOT doing). I then encrypt the key using your e and n and send you the reduction, namely 146. Decrypt to a number and then determine a and b and include them in your writeup. I have encrypted a message with $C \equiv aP + b \pmod{26}$ and got PBEXBI. I send you PBEXBI. Decrypt the message and include in your writeup.

The next day, you want to send me a message using $C \equiv aP + b \pmod{26}$. For safety you decide to use a new a and b , namely $a = 11$, $b = 21$. Turn that into a single number, namely $26a + b$ and send me that number (the key) using RSA. My $n, e = 681, 19$.

LM-4. In CryptoSoft is an implementation of Simplified AES in CBC mode with the initialization vector 1100 1011 1110 1000. Click on the tab 'Encrypt' to get there. For plaintext

it allows upper and lower case letters, digits, spaces, commas and periods only. Don't use enter/carriage return.

For practice, encrypt 'do not go gentle into that good night' using the key 1010 1010 1010 1010 (don't type the spaces). On your homework write out the last four Hex ciphertext characters. If you don't know Hex, there's an aside at the end of this problem.

Copy the ciphertext (into a buffer) and click on the tab 'Decrypt' and past the ciphertext into the appropriate field. Enter the same key and decrypt and make sure you get back the original plaintext. You don't turn anything for this paragraph's work, I just want to be sure you're using the software correctly.

Aside on Hex. Hex is an encoding of nibbles 0000, 0001, ..., 1001 encode to 0, 1, ..., 9. 1010, 1011, ..., 1111 encode to A, B, ... F.

For problems CW-RSA 2 - 4, at the top of your homework, write your name, your and your partner's colors.

R.S.A. directory, (n,e), for problems CW-RSA 2 - 4

Blue (bl): $n = 231440235891660906053817934904337175936493389690535863833485$
 $2359112541601646303546661419068500664331$

$e = 72693169641294392054324089725009715723775238103809310693108$
 $2582280108084558760457289206461584448191$

Green (gr): $n = 8982739355658980450954505563786378902219911689145959$
 $45263169996375998665839252877482856394980059807$

$e = 440795955818489272223347981691856786085455386333305489849$
 $776281698474322567978204145020005718349271$

Orange (or): $n = 45970003270264989688908974131921753900627792978445589$
 $494936157700926851903384528223275277440227877$

$e = 596131982616448003103784710980344171151668293418683956$
 $626438045113323689246259881668637506766829519$

Red (re): $n = 1859486070465862481250084737025244499643330450742207640$
 $127531631364580491764319392630687618608632729$

$e = 57463814268791658055979510125582566711041821111604849929$
 $1854170685862245899201436141624525447657277$

Yellow (ye): $n = 3466436991338570577407597275451416873137600057687491154$
 $54693238560026239259246326048573361535111137$

$e = 8760737903908348835948966304524580678073899022132796982046$
 $30864651783688734202322795082344727508911$

There are files with these numbers if you don't want to type them in, namely n.txt and e.txt CW-RSA-2. Above is your n, e . There are files called xxp.txt where xx is the first two letters of your color, blue has blp.txt, etc. This file contains your p . Read this into Pari. Find your q and write it on your homework. Now find $\varphi(n)$, using the formula and write it on your homework. Now find your d and write it on your homework. I recommend that you *lift* your d and write it to the file d.txt so you don't have to keep retyping it. Feel free to write only the first four digits of each long number in your writeup.

CW-RSA-3. There's a file called xxAESkey1.txt where xx is the first two letters of your color. This is a key for SAES in CryptoSoft that I have encrypted using your R.S.A. keys. Decrypt our shared key. It is an integer less than 2^{16} . In the file xxAESCT1.txt there is a message from me which I encrypted using SAES and our shared key. Find the message. Put our key and the plaintext message in your homework. Usually R.S.A. is used for key exchange for a faster conventional cryptosystem as in this problem.

CW-RSA-4. Find your partner's color and encrypt the SAES key 51913 for him/her using RSA (and his/her RSA keys, of course). Put the ciphertext in your writeup. As practice, your partner should decrypt the ciphertext and be sure s/he gets 51913. In this problem, you won't actually use the SAES key to encrypt a message.

Now encrypt 51914 for your partner using RSA (include its encryption in your writeup). Notice the difference in ciphertexts from the two very similar plaintexts. Feel free to write only the first four digits of each ciphertext in your writeup.

DH-1. Do this problem by hand/calculator. Diffie-Hellman key exchange. Work in \mathbf{F}_{677} , 677 is prime. $g = 2$ is the generator of \mathbf{F}_{677}^* we'll use. Your private key is 13. (a) What's your public key? (b) My public key is 287. Find our shared key. (c) Turn our shared key into a pair of keys (a,b) where our shared key is $a26 + b$ where $0 \leq a, b \leq 25$. You will encrypt for me using $C \equiv aP + b(\text{Mod } 26)$. Encrypt YO (not as a digraph, but as a two letters, i.e. two numbers in $[0,25]$) for me and decode to a pair of letters. That is the ciphertext pair you would send to me.

For the problems DH 2 - 3, at the top of your homework, put your color and name, and your partner's color

Discrete log directory of public keys for problems DH 2, 3, EIG 2, 3. We are working with $p = \text{nextprime}(10 \wedge 24)$ and $g=5$ (a generator of \mathbf{F}_p^*). This directory is in the file dlkey.txt.

Ed: 483535020765083780845831	Blue: 258020014459286684891269
Green: 794096565547742926108933	Orange: 977726291542055604902486
Red: 635364114936087527279104	Yellow: 891128110299929524095054

Your secret key (your a) is in a file called xxFFkey.txt where xx is the first two letters of your color. Note, that if you compute $\text{Mod}(5,p) \wedge a$, you should get your public key.

CW-DH-2. Diffie Hellman: Find your shared key with your partner. Reduce it mod 2^{16} (and put the reduction in your writeup). Use this integer (that is less than 2^{16}) as the key to pass messages back and forth with your partner using CryptoSoft encryption and decryption. (don't turn in anything for that).

CW-DH-3. Diffie Hellman: Find your shared key with Ed. Reduce it mod 2^{16} (and put the reduction in the writeup). Use this integer (that is less than 2^{16}) as the key for CryptoSoft SAES to decrypt the message in xxDHCT1.txt where xx is the 2 letter code for your color. Put output in writeup.

CW-DH-4. Diffie Hellman: We will not work with the public keys in the table for this problem. We will also not break the class up by color. We will instead work in $\mathbf{F}_{2^{25}}$. The polynomial $x^{25} + x^3 + 1$ is irreducible over \mathbf{F}_2 . A generator of $\mathbf{F}_{2^{25}}^*$ is x , which in Pari is

Mod(Mod(1, 2) * x, x \wedge 25 + x \wedge 3 + 1). We want to use Diffie Hellman to agree on a SAES key. My public key is in the file dhkey.txt. Your private key is 8675309. Find our shared key. It will be of the form $a_{24}x^{24} + a_{23}x^{23} + \dots a_1x + a_0$ with $a_i \in \{0, 1\}$. We want to turn that into a 16-bit SAES key. It will help to lift it twice. Our simplified AES key will be $a_{24}a_{23} \dots a_9$ (in that order, left to right). In the file allDHCT2.txt is a ciphertext. Decrypt using CryptoSoft's SAES. Put the 16 bit SAES key and the plaintext in your writeup.

CW-RSA-5. There's a file you downloaded called xxRSACT1.txt where xx is the first two letters of your color. This is a message I encrypted for you using R.S.A. and your keys n, e , given earlier. Read this into Pari, decrypt, lift, and use CryptoSoft to decode this integer to a plaintext message. Write the plaintext in your homework.

CW-RSA-6. Find your partner's color and pick the one out of the five following messages that includes his or her color (do not include the comma). blue moon rising, the corn is green, ripe juicy oranges, the lady in red, yellow rose of texas. First use CryptoSoft to encode this message as an integer. Put it in the file msg.txt. Go into Pari and use R.S.A. to encrypt each line with your partner's n and e . Write on your homework the first four digits of the encrypted number and your partner's color.

ElG-1. Do by hand/calculator. ElGamal message exchange. You and I have the same keys as in problem DH-1. You want to send the message PK (abbrev. for public key cryptography) to me. Encode this digraph as an integer in $[0, 675]$. You choose a random $k = 11$. (a) What pair of numbers will you send? (b) I send you the pair (310, 407). Decrypt and decode.

CW-ElG-2. ElGamal message exchange: In the file xxElGCT1.txt, where xx is your color, is a pair of numbers (g^k, Mg^{ak}) , where $g = 5$, a is your secret key (in xxFFkey.txt), M is my plaintext message and k is a random number I picked (you'll never need or know k). Find the number M , lift and use CryptoSoft to decode the plaintext message. Put output in writeup. Pari is flexible with division with Mod: Mod(10,11)/3 would give you Mod(7,11) also Mod(10,11)/Mod(3,11) would give you Mod(7,11). Recall for this and CW-ELG-3 we are using the discrete log directory given earlier.

CW-ElG-3. Now practice with the ElGamal message exchange with each other. I suggest that you keep your messages short. Nothing to turn in here.

CW-ElG-4. ElGamal again, but now we work in $\mathbf{F}_{2^{25}}$. The polynomial $x^{25} + x^3 + 1$ is irreducible over \mathbf{F}_2 . A generator of $\mathbf{F}_{2^{25}}^*$ is Mod(Mod(1, 2) * x, x \wedge 25 + x \wedge 3 + 1). I will use ElGamal to send you a simplified AES key. Your secret key is a=321321. I send you a pair (g^k, Mg^{ak}) , which is in the file elgtwo.txt. Find M and then lift it once and then lift it again to get a polynomial $x^{24} + \dots$. That polynomial is $a_{24}x^{24} + \dots a_1x + a_0$ where $a_i \in \{0, 1\}$. Again, to get the simplified AES key, take the first 16 bits, namely $a_{24}a_{23}a_{22} \dots a_9$. Decrypt the message in ElGCT2.txt with this key using SAES in CryptoSoft. Put the key and the plaintext message in your writeup.

CW-MO-1. Massey-Omura Cryptosystem. This must be done with a partner. Again we use $p = \text{nextprime}(10 \wedge 24)$. The order of the colors is Blue, Green, Orange, Red, Yellow. The partner whose color appears first in that list is the first partner. Both partners will find their randomly chosen encrypting key for this session in the file xxMOkey.txt where xx is

the first two letters of your color. Find your decrypting key (remember it is the inverse Mod $p-1$) and put this in your writeup. The first partner wants to communicate the AES key 1010 0101 0100 0000 to the second partner. Use CryptoSoft to turn this binary key (without spaces) into an integer. (Aside, the integer you get, in base 2, is the 16 bit key. So that integer equals $1 \cdot 2^{15} + 0 \cdot 2^{14} + 1 \cdot 2^{13} + \dots + 0 \cdot 2^1 + 0 \cdot 2^0$.) Read that integer into Pari and encrypt it. Both partners should put the encrypted number in their writeups. Get this message to the second partner. Now the second partner should encrypt that. Put that in both writeups. Now the first partner should start decrypting that. Put that number in both writeups. Now the second partner should decrypt and decode and hope that you got the same integer representing the key. On the top of your homework include your name, your color and your partner's color. Feel free to just write down the first four digits of each number.

Try the same exercise (with a different message, if you like) starting with the second partner. Nothing to turn in here.

MO-2. Alice and Bob use Massey-Omura. Alice is the initial sender. She sends message M_1 to Bob. Alice uses e_A and d_A . The next day, Alice uses Massey-Omura with Eve and she (Alice) sends a different message M_2 to Eve and uses the **same** e_A and d_A . How can Eve trick Alice and determine the message from the Alice-Bob exchange? (It's OK if Eve doesn't ever determine the message that Alice is trying to send her.) Alice should not receive anything from Eve that she (Alice) has seen before. This shows why Alice should use a different e_A and d_A pair each time. Assume that all three users use the same finite field.

For the following problems, write your color and your partner's color at the top. Do EC-1 and EC-2 by hand and with a calculator.

EC-1. Add the point $[-2,3]$ to the point $[2,-5]$ on the elliptic curve $y^2 = x^3 + 17$ using line intersections. Now do the same problem using the addition formulas.

EC-2. Double the point $[-2,3]$ on the same elliptic curve using line intersections.

Working with elliptic curves in PARI. Represent the elliptic curve $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ as a vector $[a_1, a_2, a_3, a_4, a_6]$. You can give the elliptic curve a name like $E=[0,0,0,0,1]$. Represent the point $(0,1)$ on the curve by the vector $[0,1]$. You can name points $Q=[0,1]$, $R=[2,-3]$. To find $Q+R$ you compute **elladd(E,Q,R)**. To find $Q-R$ you compute **ellsub(E,Q,R)**. To find $7Q$ (Q added to itself 7 times) you compute **ellpow(e,Q,7)**. PARI calls the 0 point $[0]$.

CW-EC-3. Use the elliptic curve $y^2 = x^3 + 17$. Let Q be the point $[-2,3]$ and R be the point $[2,5]$. Verify in your head that both points are on the curve. Find $2Q$ (I would first type $E=[0,0,0,0,17]$ and $Q=[-2,3]$ and then at this point write $q2=ellpow(E,Q,2)$ so I can use $2Q$ in the future). Then find $Q+R$, $3Q$, $4Q$, $2R$, $Q-R$, $2Q-R$, $3Q-R$, $4Q-R$, $2Q-2R$. (There are old versions of PARI on campus that want: $ellpow(e,2,Q)$. If in doubt, after ? type ?ellpow).

EC-4. Find all points of $y^2 = x^3 - 4$ over \mathbf{F}_2 , \mathbf{F}_3 and \mathbf{F}_5 (don't forget the 0-point). Do by hand.

In PARI: let's say $E=[a1,a2,a3,a4,a6]$. You can work over the finite field \mathbf{F}_p by saying $E=Mod(1,p)*E$. Now E is the same elliptic curve over the finite field \mathbf{F}_p . Once you've done

that, you don't need to make the points Mod p .

CW-EC-5. Now we'll work with the elliptic curve $y^2 + y = x^3 - x$. Let Q be the point $[0,0]$. We will work over \mathbf{F}_7 (so $p = 7$). Find $2Q, 3Q, \dots$ until you get the 0-point. Which multiple of Q is 0? (In the writeup, call a point $[a,b]$ instead of $[\text{Mod}(a,7), \text{Mod}(b,7)]$).

For EC 6, 7, 8, we will be working with elliptic curve discrete log cryptosystems. The elliptic curve is

$EE=[0,0,0,0,-4]$ ($y^2 = x^3 - 4$). The point we'll work with is $G=[2,2]$, the finite field we'll work over is \mathbf{F}_p where $p=\text{nextprime}(10\wedge 25)$ (not 24). Let $E=\text{Mod}(EE,p)$, now the elliptic curve is defined over the finite field. My public key point on this elliptic curve is $[166985550562940165155251, 1303821074460599731155518]$, it is the file `eckey.txt`. Your private key is $a=\text{nextprime}(10\wedge 22)$, so your public key is $ag=\text{ellpow}(E,G,a)$ (which you don't need to compute).

CW-EC-6. Diffie-Hellman key exchange. Find our shared key. Our shared key is a point on the elliptic curve. If Q is a point on an elliptic curve, you can get its x -coordinate by typing $Q[1]$. Lift the x -coordinate of our shared key. Reduce it mod 2^{16} . That represents an SAES key. In the file `ECCT1.txt` is a message that I encrypted with CryptoSoft's SAES using the SAES key you just got. Put the (integer) SAES key and plaintext in your writeup.

CW-EC-7. ElGamal message exchange part 1. For this problem, I am sending a message to you. So I send you two points, the first one is kG where k is a random number I found and G is the point $[2,2]$. That point is $[1603542159342002120731182, 6878152744582340991914779]$. It is in the file `kg.txt`. The second point I send you is $Q+kaG$ where Q is my plaintext point, and a is your secret key. The second point is $[7307569311613305606365950, 1187389910207131103923637]$, it is also in the file `kg.txt`. Find the point Q . Get its x -coordinate as in EC-6 and remove its last digit (omit that last 1). Now use CryptoSoft to decode this integer. Put plaintext in your writeup.

CW-EC-8. El Gamal message exchange part 2. For this problem, you will send me the message 'Honduras' (without apostrophes). First use CryptoSoft to encode it as an integer that we will call m . Now read that file into Pari. If starting a new session you need to tell PARI what p , E and G are. Now let $n = 10 * m$ and plug n into $x^3 - 4 \text{ Mod } p$ (the left side of the elliptic curve equation) and use the `sqrt` function to see if $n^3 - 4 \text{ Mod } p$ is a square in \mathbf{F}_p or not. Feel free to use the `subst` command (type `?subst` to learn how it works). (I wrote the on-line help for `subst`, Frenchmen reworded it :) If you get a square root, that means there is a point on the elliptic curve with x -coordinate n . If you don't get a `sqrt`, then compute $n1=n+1$. Do all the same steps with $n1$ instead of n . If this doesn't work, go to $n2=n1+1$ (hint, you don't need very many n 's). Once you get a square root, let y be that square root. Create a point $Q=[ni,y]$ (both ni and y are Mod expressions). This is a point on the curve. Hope you see why. Now you have encoded your plaintext as a point q on the elliptic curve. You choose a random $k=\text{nextprime}(10\wedge 20)$. Now compute kg . Put that in your writeup (first 4 digits each is O.K.). Now compute $Q+kaG$ (remember that's my ag , which is in `eckey.txt`). Put that in your writeup. Those are the 2 points you would send me. As a check, the last point is $(6\dots, 8\dots)$.

CW-EC-9. This is the only realistic elliptic curve cryptography homework problem. We will do ECDH over the finite field $\mathbf{F}_{2^{16}}$. The only unrealistic part is the 16 (which should be a prime at least 163). Let $f = t^{16} + t^6 + t^2 + t + 1$. We represent $\mathbf{F}_{2^{16}}$ as $\mathbf{F}_2[t]/(f)$. I use t here instead of the usual x so as not to confuse with the x in the equation of the elliptic curve. The elliptic curve is $y^2 + xy = x^3 + 1$ or in Pari: $E = [1, 0, 0, 0, 1] * \text{Mod}(\text{Mod}(1, 2), f)$. My public key is the point $[t^{11} + t^8 + t^4 + t^3 + t, t^{13} + t^9 + t^8 + t^7 + t^6 + t^5 + t^4 + t^3 + 1]$. The

file ECDHkey.txt has two lines. The first is E , the second is my public key point. Your private key is 31415. Find our shared Diffie Hellman Key. Lift its x -coordinate twice to get a 16 bit SAES key. The x -coordinate is $a_{15}t^{15} + a_{14}t^{14} + \dots + a_1t + a_0$. Let the SAES key be $a_{15}a_{14} \dots a_1a_0$. Go to CryptoSoft and decrypt the message in ECAESCT1.txt. In your writeup put the key and plaintext.

Not a homework problem: You have two files (that I didn't tell you about before - heh, heh). They are called renc and rdec. Those stand for Rijndael encryption and Rijndael decryption. They do simplified AES encryption and decryption, respectively. While in Pari, you can read those in: for example `\r renc`. You won't see much happen in Pari and that's OK. If you read in renc then you can type

`aesenc([1,0,1,0,1,1,1,0,1,0,0,0,1,0,1,0],[1,1,0,1,1,1,0,0,1,0,0,1,1,1,0,0])`. The first vector is the plaintext; the second vector is the key. The output (on the next line) will be the corresponding ciphertext. Notes: remember, if the plaintext vector is called %1 and the key is called %2 then you could just type `aesenc(%1,%2)`. It is important (for later) that these vectors not look like `[Mod(1,2),...]`. If you have a vector like that, then *lift* it.

If you read in rdec then you can type `aesdec([1,1,1,0,0,1,0,0,1,1,0,0,1,0,1,1],[1,1,0,1,1,1,0,0,1,0,0,1,1,1,0,0])` and it will use simplified AES to decrypt the first vector (the ciphertext) with the second vector (the key).

CW-HASH-1. Using AES as shown in class does not create a secure hash function. It does not have the one way property. I want you to demonstrate that you understand how to break the one way property by doing an example with simplified AES. Assume $IV = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. Find M_1 and M_2 , both 16 bit strings, so that the hash is $[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$. To be precise, find M_1 , a 16 bit string, so that when you encrypt it using SAES with IV as the key, you get an output C . Now encrypt M_2 , a 16 bit string, so that when you encrypt it using SAES with C as the key, you get as output $[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$. Explain to the homework grader how you did it (so s/he does not have to verify all of them using a computer). Use the **Not a homework problem** immediately before this problem to help.

CW-MAC-1. MAC problem. The plaintext is *memory* (converted to ASCII). The MAC key is $[0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1]$. That key is in a file you downloaded called mackey.txt Find the MAC using simplified AES. Again you can use renc. As a check: the answer is 1011*****0110. Copy your answer to your notes as you will need it again when you do CW-MAC-2.

Hash-2. Let $f : X \rightarrow Y$ be a hash function and assume $|X|/|Y|$ is very large. I want you to give an informal proof that if f has the weakly collision free property then f has the one-way property. We will do this by contradiction. So Assume that f does NOT have the one-way property. Given an informal proof that f will NOT have the the weakly collision free property.

Recall: One-way property: Given $y \in Y$ it is not feasible to find $x \in X$ such that $f(x) = y$. Weakly collision free property: Given $x \in X$ it is infeasible to find $x' \in X$ with $x' \neq x$ such that $f(x') = f(x)$.

RSA-7. This will give you practice at using RSA for signatures. The two parties involved are Dr. Schaefer, whose enciphering keys are $n_S = 65, e_S = 11$ and Dr. Appleby $n_A = 77, e_A = 13$.

Dr. Appleby has an ID number known to everyone and it is 32; Dr. Schaefer's is 30. First S wants to send a msg notifying A that he's about to send a signed message, so he sends his ID number (30), but encrypted, with A's keys, to A. It is simply encrypted with no digital signature. Without this notification, A would have no idea who the upcoming message was from. a) How will this message look to the enemy (Eve, perhaps), i.e. what number will S send? (In your writeup of a), show your work computing d_S and d_A). b) Now A knows who the upcoming message is allegedly from; notice the enemy does not, not knowing A's deciphering key. S now wants to send the message 21 to A; he will sign it and encrypt it for A. This way the enemy can't read it and A can be sure it came from S. Pay attention to the sizes of n_S and n_A when you decide whether to sign first or encrypt first. How will this message look to the enemy. (Back of the book: The answer is '3', but I want you to show work getting it, of course). c) Now have A decrypt both messages (and with the 2nd, confirm it's from S). Do all work for a)-c) by hand and with a calculator, show your work in the writeup. This will give you practice with repeated squares, finding inverses, etc. for the midterm. You may check your work with PARI. You may use PARI (no need for calculator or much writeup) for d)-f) or do them with a calculator if you hate computers.

Now we'll have A return a message to S. First A simply encrypts his ID number 32 for S, so S will know a signed message from A is about to arrive. d) what number will A send? Now A wants to send S the message 15. He will sign it so S knows it's really from A and he'll encrypt it so the enemy can't read the message. Pay attention to the sizes of n_S and n_A when you decide which to do first. e) what number will A send? f) now have S decrypt both messages and confirm the second is from A.

CW-ElG-5) For ElG 5, at the top of your homework, write your name, your and your partner's colors. ElGamal signature problem. In this problem both partners should send and receive signatures. Use the discrete log directory from right before problem CW-DH-2: $p = \text{nextprime}(10^{24})$ and $g=5$ and $a = \text{your secret key in xxFFkey.txt}$. Encode 'It is I' (without apostrophes) as an integer using CryptoSoft. That's your signature S. (This problem is not very realistic. Usually you sign a MAC.) Use the number in xxMOkey.txt as your random k . Compute $r = g^k \text{ Mod } p$ and lift. Find your $x = k^{-1}(S - ar) \text{ Mod } p-1$ and lift. Put r , x and S in your writeup. Send your partner (r,x,S) .

ii) Receive (r,x,S) from your partner. The signature receiver works only Mod p . Compute r^x , g^{ar} (where g^a is your partner's public key; it is in dlkey.txt and the directory in earlier homework), and $g^{ar}r^x$. Then compute g^S . It should be the same as $g^{ar}r^x$. Include r^x and g^{ar} in your writeup.

ElG-6 Let's say that on Monday and Tuesday, Alice uses ElGamal signatures to send two different signatures, S_1 and S_2 , to Bob but uses the same k both days. Explain how Eve can determine a_A (Alice's private key). For simplicity, you may assume that the numbers appearing during the problem are relatively prime to p and $p - 1$. This explains why Alice should use a different k each time she signs something.

CW-MAC-2. This and MAC-3 are more realistic signature problems since usually what gets signed is a MAC. We want to sign the MAC from problem CW-MAC-1 using RSA. The MAC above can be considered to be a base two number. Turn it into a base 10 number $S = 1 \cdot 2^{15} + 0 \cdot 2^{14} + 1 \cdot 2^{13} + 1 \cdot 2^{12} + \dots + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0$ (CryptoSoft can help). Sign

that using your private key from problem CW-RSA-2. Put the signature in your writeup (first 4 digits is OK). Have your partner verify the signature using your public key. Copy the value of S to your notes as you will need it for CW-MAC-3.

CW-MAC-3. We want to sign the MAC from problem CW-MAC-1 using ElGamal signatures. Take the number S that you determined in problem CW-MAC-2. Use the p, g , private key and k from problem CW-ElG-5). Put the signature S, r, x in your writeup (first 4 digits is OK). Have your partner verify the signature.

Cert-1. I want you to explain why, when you check a certificate, you should hash the top of the certificate and confirm that you get the hash listed on the certificate. Let's say you want to check a certificate signed by Verisign. If your browsers do not compute the hash the hash, then the entity with the certificate could cheat. Explain how they could still fill in a string for the hash and a string for the signature so that $\text{signature}^{e_{\text{ver}}} \pmod{n_{\text{ver}}} = \text{hash}$.

SSL-1. Nothing to turn in on this one. Go through the simplified SSL protocol with your partner. Choose an Amazon and a Bob. Have Bob find a message that you will encrypt using simplified AES. Use simplified AES for the MAC. Use RSA or Diffie Hellman initially to agree on the AES and MAC keys. Have both parties do their parts of the whole process. If you're really gung ho, do it again switching roles (possibly with a new message). Don't bother with certificates.