MATH 75, FINAL CIPHER CHALLENGE

EDGAR COSTA

The solution to each of the following ciphers is a codeword: this codeword is either the keyword (or key phrase) or a secret word or phrase contained in the plaintext. Your mission is to discover these codewords.

In a few cases, the plaintexts of earlier ciphers contain clues for the later ones.

You have two options to turn in your final exam:

(1) Deliver your completed solution to my office (Kemeny 339) on Tuesday, June 5th, 2018, between 10:00 a.m. and noon; or

(2) Email me (`edgar.costa@dartmouth.edu`) your completed solution in one file by noon on Tuesday, June 5th, 2018. (I am a bit more relaxed about this, but please do not spam me with a bunch of different documents.)

No late exams will be accepted.

Show your work, and please be neat! No need to be laboriously detailed, but please indicate clearly your method of attack.

If you use any computational resources, please print out and attach all code and output. (References to code can be given, but you must also show the run and output.) Feel free to use the SageMath notebook server:

`https://doob.dartmouth.edu/`

You are free to (re)use any code, algorithm, or method from classwork or homework, and you may use any programming language you like. However, the rules for cooperation are completely different than for the homework. You may not work with anyone else. No communication about the exam is permitted with anyone except the instructor. In particular, do not give away solutions and do not share code. You may *not* use code that is not yours: for example, do not use random web applets. Exceptions to this rule will be allowed on a case-by-case basis—but contact me first! The intent of this policy is to ensure that you are the only author of the work you turn in, not to obstruct good work. If you consult or use any resource other than the textbooks or classwork, state this clearly.

If you are stuck—or just anxious—about one of your solutions, please come talk to me or send me an e-mail! I will be happy to help (at no penalty).

Have fun and good hunting!

Date: Due noon on Tuesday, June 5th, 2018.

Cipher 1: Substitution cipher

```
NOCRCFSKSDATRGTSDJFHTXXCOOCCSFJSKCRRHDICVXSJCNNRFATXPCNLFJSP
TCJTNRFATXTRXCJGDKNCCSFJSXCCZTSSCOJCJTFKMFRFKDICVXSNCOLTRTDN
PTJNDJNLTVJSTRGRCONLNLTJNCCZNLTCNLTRFKQVKNFKMFDRFJSLFADJGETR
LFEKNLTPTNNTRIXFDWPTIFVKTDNOFKGRFKKHFJSOFJNTSOTFRNLCVGLFKMCR
NLFNNLTEFKKDJGNLTRTLFSOCRJNLTWRTFXXHFPCVNNLTKFWTFJSPCNLNLFNW
CRJDJGTBVFXXHXFHDJXTFATKJCKNTELFSNRCSSTJPXFIZCLDZTENNLTMDRKN
MCRFJCNLTRSFHHTNZJCODJGLCOOFHXTFSKCJNCOFHDSCVPNTSDMDKLCVXSTA
TRICWTPFIZNLTZTHOCRSDKRCPTRNMRCKNDJFJRKFERCPXTWNLTCRSTRCMKTA
TJWCSVXCEDKKWFXX
```

Cipher 2: Affine cipher

Eve intercepts the ciphertext

```
36333, 28512, 64818, 20428, 47277, 59369, 47116, 45798, 5832,
17660, 61146, 53877, 15849, 4382, 52990, 27892, 48922, 50914,
13506, 24094, 59369, 64818, 56435, 46740, 19320, 52990, 52427,
52990, 27892, 48922, 50914, 63538, 63894, 24094, 48761, 27892,
55522, 2327, 61873, 28478, 50914, 27726, 15787, 43074, 48922,
62724, 58778, 21375, 25012
```

and the first part of the corresponding plaintext:

```
19561, 27769, 11296, 27753 = "Lily, li"
```

Charlie, an enemy agent, was also captured. Using enhanced interrogation techniques, Eve was able to able to extract the following information: Alice uses an affine cipher, and the plaintext alphabet consists of blocks of two letters written as ASCII bytes and then interpreted as an integer modulo n. Unfortunately, Charlie suffered a medical incident before he could disclose n.

Cipher 3: Diffie–Hellman key exchange

```
alice> hey bob hwru
bob> im gr8
alice> hv d secret g?
bob> ys
alice> prv it
alice> p = 126765060022822940149670320633 1
alice> a = 49999
alice> compute g^a
bob> 29645751720482123999806757206 29
eve> LULZ, gg!
```

Cipher 4: RSA

```
bob> xo alice, i need da secret key
bob> but eve can here us
eve> yellow!
bob> n = 47173362901027805827488943908212254131882888891l3
bob> e = 2^150+1
bob> did u here abt charlie? cr8zy m8!1!
alice> that exponent has been compromised;
alice> a decryption exponent is the RSA patent number
bob> whatevs, FINE
bob> e = 65537
alice> y = 46618754094228625131914564009141629507205472331 73
bob> c u l8r
eve>  Oh My!
```

Cipher 5: RSA

```
alice> hey bob hwru
bob> im gr8
bob> you ready? charlie want's to send us a message!
bob> n = 49355407517652738747237894951599690330136219174468963851
         68710088518492057924984638275768694615065921707914550359
alice> so c00l that we have the same modulus, so much more secure
alice> i dont really understand how to make my own primes :(
bob> no prob babe
alice> e = 65537 for me
bob> f = 100003 for me
charlie> hi guys, here is the wink wink nudge nudge for the you-know-what
charlie> alice: 156596570957509534853917274525581256548041829681950944 56
                106801545948062150087793862600869709870244190518156668 57
charlie> bob:   257878075607486792033505379533904879297654804024438200 81
                428672836463154471484031078999639790768964227554725047 83
charlie> i just ASCII'ed the whole thing, no blocks or anything fancy
eve> LOL
```

Cipher 6: RSA

```
n = 38363377337643649482566149302846355910907128165 5
    79457714343557347024260093135367748083611439063
e = 65537
y = 19758787036898556648955350327907478583773573858 7
    62934796891942172502739582925252245424966766 04
```

The plaintext is written in base 26.

Cipher 7: Diffie–Hellman key exchange

```
alice> Dude, look at this p!
alice> 6323291470921996460000000000000000000000000000000954817012109221465 47
bob> WILD! let's use it
bob> g = 2 as always!
alice> 11824133099297351280364249735666699054153772699598949884103771866349
bob> 18958445374903586524651007339890684505843652541505582097475315564406
alice> OK, I will use our common secret to send you a message
alice> I don't trust ASCII--I think it's rigged.
alice> I'm just going to write my message as an integer like
alice>    h e l l o = 07 04 11 11 14 = 704111114
alice> Here it is:
alice> 534569730837141629686997031912074692716829004201603855907855053174 28
```

Cipher 8: Elliptic curve discrete logarithm

The elliptic curve $E : y^2 = x^3 + x$
having field of definition $\mathbb{F}_p$ with $p = 2^{89} - 1$
is used in an elliptic curve cryptosystem by Alice in Wonderland.
"No wise fish would go anywhere without a porpoise."
Known is the point $P = (408358146153463289622115164, 160614043710190468600357775)$;
she hides a secret codeword in the
multiple a of P written in base 26, not the
ordinate or anything. This multiple is
observed by the trolling bellman to be

$$aP = (553896511907353224105489574, 594394227295626927732292957).$$

The discrete logarithm problem is supposed to be
hard, right?
"¡To grow larger and reach the key, just keep doubling!," advises the doorknob.

Cipher 9: Elliptic curve ElGamal

Elliptic curve ElGamal is used with the following parameters:

- $p = 2^{31} - 1 = 2147483647$
- $E : y^2 = x^3 + x + 1$ over $\mathbb{F}_p$
- $G = (493644129, 2069524559)$

Bob sends the point

$$(928984375, 811839063) \in E(\mathbb{F}_p).$$

Alice encodes her keyword as the x-coordinate of a point on E, and she sends the pair of points

$$((1004499824, 1398370638), (270353101, 1452301961)).$$