Discrete Logarithms & Cryptography

Nathan Solomon

May 18, 2023

Alice and Bob are in a long-distance relationship, and want to send each other love poems. However, they're paranoid that Eve the Evil Eavesdropper will see the letters and tease them for their amateur poetry. This packet contains a hodgepodge of seemingly unrelated topics, but by the end, you will combine everything you've learned to help Alice and Bob communicate in secret, even if Eve sees every message they send each other.

Problem 1: One way to calculate n^{123} is to take n and multiply by n 122 more times, but that's a huge pain because it requires 122 steps. What is the least number of multiplication problems needed to calculate n^{123}? You can only multiply two number together at a time, and you cannot use division.

Solution: Pretty sure it's 9. One method is to calculate $\{n^2, n^3, n^6, n^{12}, n^{24}, n^{48}, n^{96}, n^{120}, n^{123}\}$. *We should check in with students individually to make sure they can get it in 10 or 11 steps, before showing them all the solution.*

Problem 2: The number line in the figure below has been twisted into an infinitely long helix with 5 points per turn. Prove that for any number $a \in \{..., -3, 2, 7, 12, ...\}$ and any $b \in \{..., -2, 3, 8, 13, ...\}$, the product $a * b$ will be in the set $\{..., -4, 1, 6, 11, ...\}$ and the sum $a + b$ will be in the set $5\mathbb{Z}$.

Solution: Write a as 2 plus a multiple of 5, and write b as 3 plus a multiple of 5. Then use the distributive property to show that $a * b$ is 1 more than a multiple of 5.

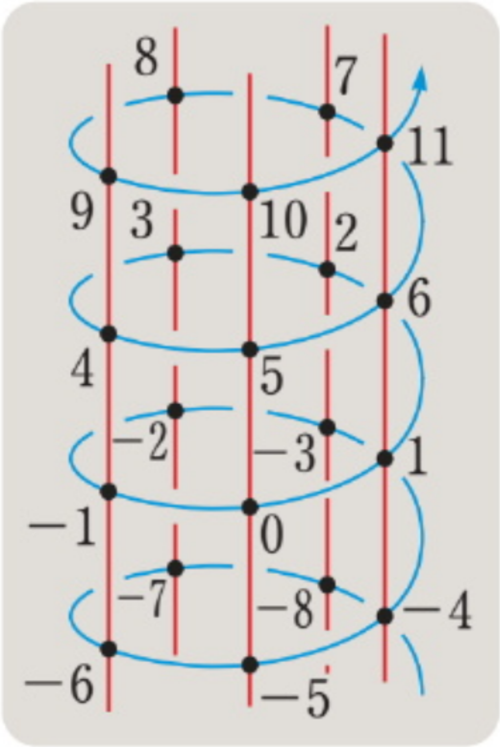


Figure 1: Number line twisted into a helix. Source: `https://i.stack.imgur.com/rnmFz.png`

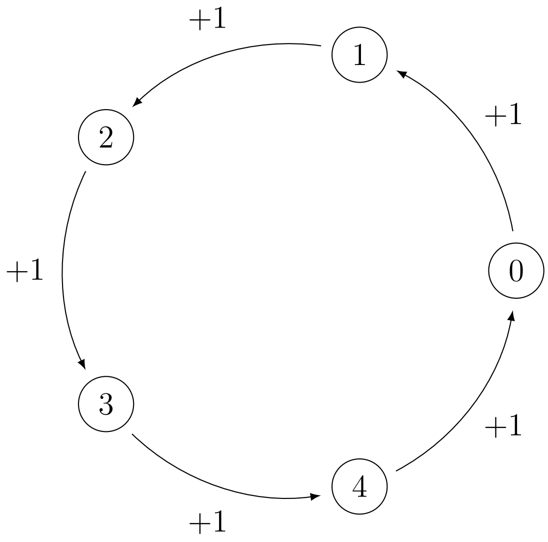


Figure 2: Just like the integers can be drawn on a number line, the integers modulo 5 can be drawn on a "number circle", since 4+1=5=0

If you look straight down at that helix, all the points on one vertical line will be in the same spot, so the infinite helix becomes a circle with only five distinct points. This represents a number system called "integers modulo n" (in this case, $n = 5$). That number system is written as $\mathbb{Z}/n\mathbb{Z}$ (the slash is read as "quotient"), which means, "take the set of all integers, $\mathbb{Z}$, except consider all multiples of n ($n\mathbb{Z} = \{..., -n, 0, n, 2n, ...\}$) to be equal to each other". It's also sometimes called $\mathbb{Z}_n$, because mathematicians are too lazy to write $\mathbb{Z}/n\mathbb{Z}$.

Problem 3: Find all positive integers n such that $3^n \equiv 5 \mod 11$. *Note: "$3^n \equiv 5 \mod 11$" is a clearer and more explicit way of saying "$3^n = 5$ in the $\mathbb{Z}_{11}$ number system".*

Solution: $n \in \{3, 8, 13, 18, ...\}$

The base a logarithm of b, written $\log_a b$, is the number of times you need to multiply a by itself to get to b. For example, $\log_4 512 = 4.5$ because $4^{4.5} = 512$. So long as a and b are both positive, there is exactly one possible value for $\log_a b$. We can define a similar operation, called a discrete logarithm, which works in $\mathbb{Z}_n$ instead of $\mathbb{R}$. In $\mathbb{Z}_n$, a base a logarithm of b is ANY number c such that $a^c = b$ (that is, $a^c \equiv b \mod n$).

Problem 4: In $\mathbb{Z}_{11}$, find $\log_3 5$ and $\log_3 6$.

Solution: $\log_3 5$ *is the same as in the last question, and* $\log_3 6$ *does not exist.*

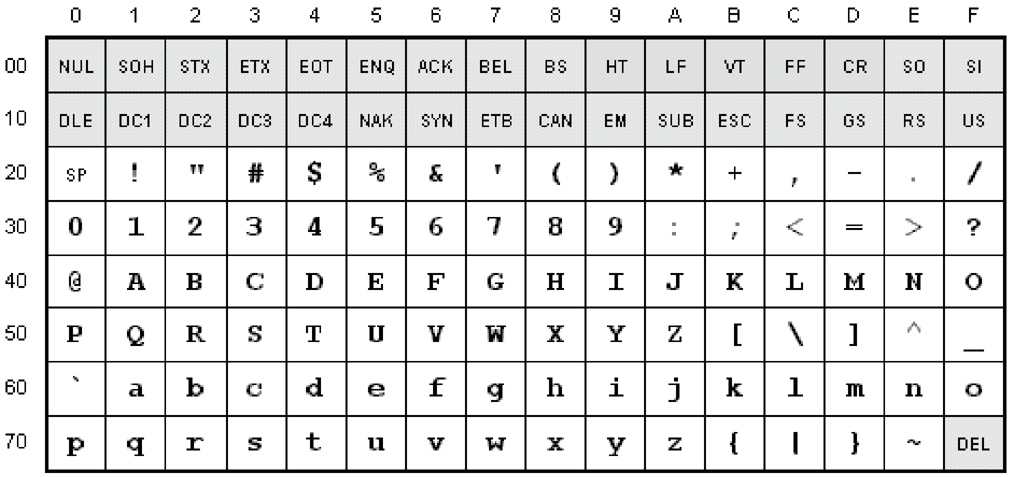
Problem 5: Write the entire addition and multiplication tables for the $\mathbb{Z}/2\mathbb{Z}$ number system. Which boolean operators do these correspond to? Since addition and multiplication are both associative and commutative, this will prove that the corresponding boolean operators are also both associative and commutative.

Solution: Addition is XOR and multiplication is AND.

Problem 6: Using only boolean operators (such as AND, NOT, OR, and XOR), describe a process for adding and for multiplying two positive integers a and b that are written in binary. You are also allowed to use the left-bitshift operator, which adds a zero to the right end, and the right-bitshift operator, which removes the rightmost bit.

*Solution: For addition, calculate (a AND b) shifted left one bit, and also calculate (a XOR b). Store those values in place of a and b respectively, and repeat that process until a=0 and b is the sum of the original two numbers. For multiplication: for each 1 in the binary representation of a, bitshift b left so that it's multiplied by the place value of the 1, and add that shifted version of b to a running total. Once you do that for all the 1s in a, the running total will be a*b.*

The ASCII (American Standard Code for Information Interchange) alphabet is an ordered list of 128 characters, including regular letters, symbols, and special characters (such as delete and tab). That means any character in the following table can be represented by a number from 0 to 127, or by exactly 7 bits (a portmanteau for "binary digits"), although we add a zero at the beginning because computers like to work with 8-bit chunks (bytes). We can also split that into a pair of 4-bit chunks (occasionally called "nybbles", because they're half a byte), and represent each nybble with a hexadecimal digit. For example, "Z" can be written as 90 in decimal, as 5E in hex, or as 01011010 in binary.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
10	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Figure 3: Table of ASCII characters in order. Source: `http://www.goldparser.org/images/ascii-67.gif`

Problem 7: Convert each character in the string “Hi Bob!” to hexadecimal and to binary.

Solution: In hex: 48 69 20 42 6f 62 21. In binary: 01001000 01101001 00100000 01000010 01101111 01100010 00100001.

At the beginning of this year, we learned about the XOR (exclusive or) operator, represented by the symbol ⊕. XOR takes two bits and returns 1 if they are different, and 0 if they're the same. The bitwise-XOR operator, represented by the same symbol, takes two ASCII strings of the same length, converts them to binary, XORs all the bits together, and converts the result back to an ASCII string. For example, “AAAAA” ⊕ “hello” is “)$ – –.”

Problem 8: Alice sends a message to Bob using XOR encryption. The message is $m =$ “Hi Bob!”, the password is $p =$ “AAAAAAA”, and the encrypted message is $e = m \oplus p$. What is e, what is $e \oplus p$, and what is $e \oplus m$?

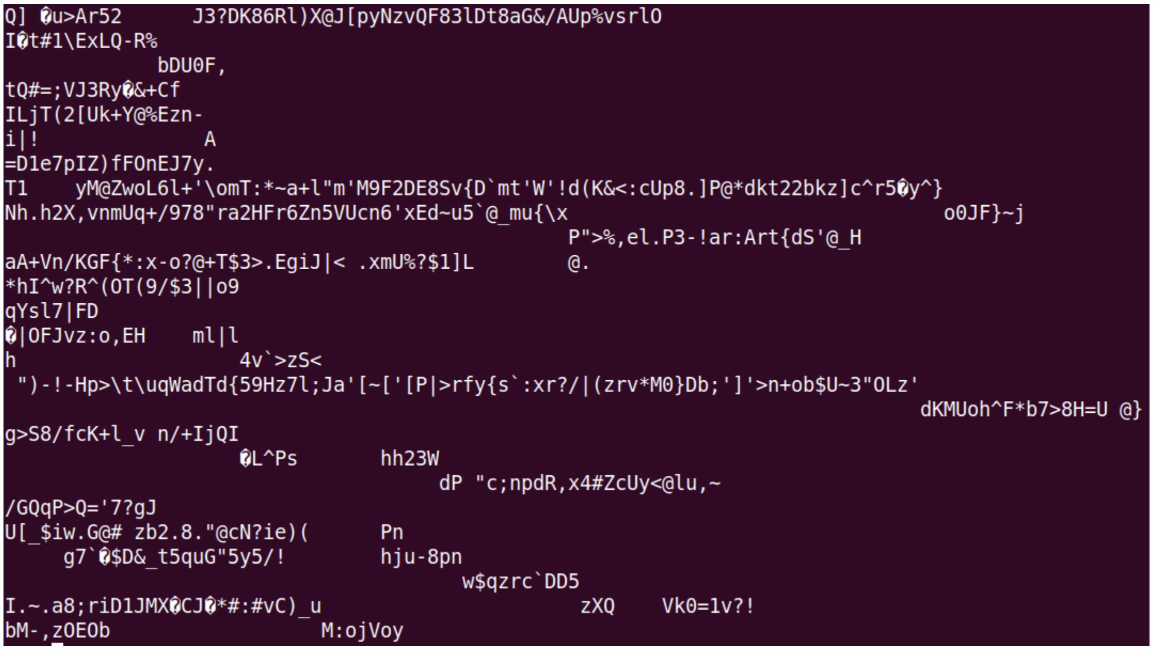
Solution: e = “\t(a\x03.#`” (“\t”=chr(9) is a tab, and “\x03”=chr(3) is the end-of-text character). $e \oplus p = m$ *and* $e \oplus m = p$.

To encrypt longer messages, you can just repeat the password over and over. For example, if Bob uses XOR encryption with the password “abc123” to encrypt a 20-character message m, the encrypted message will be $m \oplus$ “abc123abc123abc123ab”. However, this method makes the encryption less secure.

Problem 9: Eve knows that Alice and Bob are communicating using XOR encryption, and that their password is 7 characters long. She has intercepted a message from Bob, and since she knows Bob is very old-fashioned and formal, she (correctly) predicts that the letter starts with “Dear Alice,”. What is their password?

Solution: $M[:7] \oplus$ "*DearAl*"*, where* $M[:7]$ *is the first seven letters of the message.*

Sometimes small snippets of the decrypted message that you can predict before decrypting, like "Dear Alice", can help you figure out the password. Those snippets, called cillies, were used during WWII in cracking the Enigma code.



```
Q] �u>Ar52      J3?DK86Rl)X@J[pyNzvQF83lDt8aG&/AUp%vsrlO
I�t#1\ExLQ-R%
             bDU0F,
tQ#=;VJ3Ry�&+Cf
ILjT(2[Uk+Y@%Ezn-
i|!              A
=D1e7pIZ)fFOnEJ7y.
T1    yM@ZwoL6l+'\omT:*~a+l"m'M9F2DE8Sv{D`mt'W'!d(K&<:cUp8.]P@*dkt22bkz]c^r5�y^}
Nh.h2X,vnmUq+/978"ra2HFr6Zn5VUcn6'xEd~u5`@_mu{\x                               o0JF}~j
                                                P">%,el.P3-!ar:Art{dS'@_H
aA+Vn/KGF{*:x-o?@+T$3>.EgiJ|< .xmU%?$1]L        @.
*hI^w?R^(OT(9/$3||o9
qYsl7|FD
�|OFJvz:o,EH    ml|l
h                   4v`>zS<
 ")-!-Hp>\t\uqWadTd{59Hz7l;Ja'[~['[P|>rfy{s`:xr?/|(zrv*M0}Db;']'>n+ob$U~3"OLz'
                                                                              dKMUoh^F*b7>8H=U @}
g>S8/fcK+l_v n/+IjQI
                    �L^Ps       hh23W
                                     dP "c;npdR,x4#ZcUy<@lu,~
/GQqP>Q='7?gJ
U[_$iw.G@# zb2.8."@cN?ie)(      Pn
     g7`�$D&_t5quG"5y5/!        hju-8pn
                                       w$qzrc`DD5
I.~.a8;riD1JMX�CJ�*#:#vC)_u                      zXQ    Vk0=1v?!
bM-,zOEOb                  M:ojVoy
```

Figure 4: An excerpt from the suspicious file on Malice's computer

Alice was poking around on her younger sister Malice's computer and found a file called `conspiracies/lawn_gnomes/secret_plans.txt`. When Alice confronted her about it, Malice boasted that no one but her could ever understand it because it's XOR encrypted using a 30-character-long passcode.

Problem 10: How can Alice decrypt `secret_plans.txt`? Assume the file has over 10,000 characters, and that the decrypted version is in ASCII English.

Solution: For each of the 128 characters in the ASCII alphabet, XOR that character with the 1st, 31st, 61st, etc. characters of the file. If that gives you a bunch of spaces and lowercase vowels and very few special characters, it's the correct letter, otherwise, keep trying. Once that's done, repeat the entire process to find out the other 29 characters of the password are.

Note to instructors: check in with students to make sure they figure this out and understand why it works, because this method of breaking XOR encryption is super cool.

To avoid such attacks, Alice and Bob agree to split the letters they send each other into 30-character chunks, and use a different password to encrypt each one (they are VERY insecure about their poetry). But this brings them to another issue: they still haven't figured out how to exchange those passwords!

Problem 11: a and b are two nonzero numbers in $\mathbb{Z}_p$. What criterion must p satisfy to guarantee that $a * b \neq 0$? Prove that if p does not meet that criterion, you will be able to find nonzero numbers a and b such that $a * b = 0$.

Solution: p must be prime. If p is not prime, there exist integers a and b between 1 and $p - 1$ *such that* $a * b = p$*, which in the* $\mathbb{Z}_p$ *system, is the same as* $a * b = 0$.

If p meets the criterion you found above, then we can define another number system called $\mathbb{Z}_p^\times$. This is the same as $\mathbb{Z}_p$, except the number 0 has been removed, and addition and subtraction are no longer allowed.

Important theorem that we aren't gonna prove: For any such p, $\mathbb{Z}_p^\times$ contains at least one value g, called a generator, such that you can reach all the values in $\mathbb{Z}_p^\times$ by raising g to some power. For example, in $\mathbb{Z}_{13}^\times$, the powers of 2 are 1, 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7, 1, 2, 4, 8, etc. That covers all twelve numbers in $\mathbb{Z}_{13}^\times$, so 2 is a generator of $\mathbb{Z}_{13}^\times$.

Problem 12: For any number $a \in \mathbb{Z}_p^\times$, find a value a^{-1} such that $a^{-1}a = 1$.

Solution: let g be a generator of $\mathbb{Z}_p^\times$*, and let* $n = \log_g a$*. Then* $a^{-1} = g^{p-n-1}$*. Note to instructors: consider giving the hint that a can be written in terms of g.*

Problem 13: Prove that for any prime number p and any integer a which is not a multiple of p, $a^{p-1} \equiv 1 \mod p$. This is one version of Fermat's Little Theorem, which is one of the most important theorems in number theory.

Solution: let g be a generator of $\mathbb{Z}_p^\times$*, and let* $n = \log_g a$*. Then* $a^{p-1} = g^{n(p-1)} = (g^{p-1})^n = 1^n = 1$

Bob informs you that his millionaire uncle has passed away, and instead of leaving his fortunes to anyone, he simply left Bob a note saying that the password to his bank account is the smallest positive base g logarithm of A (in the number system $\mathbb{Z}_p$, where p is 40 digit prime number and g is a generator of $\mathbb{Z}_p^\times$). The values of A, g, and p are written on the note, which Bob offers to sell to you for \$419.

Problem 14: If you accept Bob's offer, will you be able to unlock the bank account in your lifetime? Why or why not?

Solution: Probably not. The most obvious method for solving discrete logarithms is trial multiplication, which runs in linear time. There are other methods that run in square root time, or that run faster only if $p - 1$ *has no large factors, but no one has found a method better than those.*

Since Bob's uncle has already done the work of finding an extremely large prime number p and a generator g of $\mathbb{Z}_p^\times$, Alice and Bob decide to use those same values to generate a shared secret key. Here's their plan:

- Alice thinks of a number a and doesn't tell anybody what a is. She uses it to calculate a number A, which she tells everyone.
- Bob thinks of a number b, which he tells to no one. He uses b to calculate B, which he tells to everyone.
- Alice and Bob use what they each know to calculate the same number S, which they will use as their shared secret key for XOR-encrypting messages.
- Eve will know what g, p, A, and B all are, but that's OK

Problem 15: How should Alice and Bob define A, B, and S so that Eve can't figure out what S is?

Solution: $A = g^a, B = g^b, S = g^{ab}$. *To find out what* S *is, Eve would have to know either* a *or* b*, which would require solving an extremely difficult discrete logarithm.*

Check with an instructor to make sure you figured out the correct solution to the last problem. If so, you have figured out the Diffie-Hellman key exchange protocol!

Since XOR encryption is so vulnerable, Alice and Bob had agreed to split their messages into short chunks (shorter than p, that is). They decide to use a slight variation of their earlier plan, where for each chunk, Bob comes up with a new random number b, and sends the new B along with the encrypted chunk of the message. This results in the encrypted message being twice as long as the decrypted message.

Problem 16: Alice and Bob are still worried that XOR encryption is sketchy. Given p, a very large prime number, S, the secret key, and m, a chunk of the message to be encrypted, what is another way they can encrypt and decrypt m? Assume S and m are both around the same order of magnitude as p, although S and m are both slightly smaller than p.

Solution: Convert m *to a numerical value, and calculate the multiplicative inverse of* S*. The encrypted version of* m *is then* $S^{-1}m$*.*

That last method is called Elgamal encryption.

If you enjoyed this packet, you might also be interested in Euler's Theorem, which is a generalization of Fermat's Little Theorem, and RSA encryption, which is a very common encryption method based on Euler's Theorem.