Classic Ciphers I

Lectore 02

Byte-wise Shift Cipher

Byte-wise Shift Cipher

- Instead of $a, b, c, d, \ldots, z$ have (for example) 0000, 0001,. . .,1111.
- Works for an alphabet of *bytes* rather than (English, lowercase) *letters*
  - Data in a computer is stored this way anyway. So works natively for arbitrary data!
- Use XOR instead of modular addition. Fast!
- Decode and Encode are both XOR.
  - Essential properties still hold

Hexadecimal (base 16)

Hex	Bits ("nibble")	Decimal	Hex	Bits ("nibble")	Decimal
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	A	1010	10
3	0011	3	B	1011	11
4	0100	4	C	1100	12
5	0101	5	D	1101	13
6	0110	6	E	1110	14
7	0111	7	F	1111	15

Hexadecimal (base 16)

Notation: 0x before a string of $\{0, 1, \ldots, 9, A, B, C, D, E, F\}$ means that the string will be base 16.

- 0x10
  - 0x10 = 16*1 + 0 = 16
  - 0x10 = 0001 0000
- 0xAF
  - 0xAF = 16*A + F = 16*10 + 15 = 175
  - 0xAF = 1010 1111

ASCII

- Characters (often) represented in ASCII with TWO hex-digits.
- Potentially 256 characters via $\{0, \ldots, 9, A, \ldots, F\} \times \{0, \ldots, 9, A, \ldots, F\}$
- Only use 128 characters via $\{0, \ldots 8\} \times \{0, \ldots, 9, A, \ldots, F\}$

Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

Source: http://benborowiec.com/2011/07/23/better-ascii-table/

ASCII

- '1' = 0x31 = 0011 0001
- 'F' = 0x46 = 0100 0110

Useful observations

- Only 128 valid ASCII chars (128 bytes invalid)
- 0x20-0x7E printable
- 0x41-0x7A includes upper/lowercase letters
  - Uppercase letters begin with 0x4 or 0x5
  - Lowercase letters begin with 0x6 or 0x7

Byte-wise shift cipher

- $\mathcal{M}$ = {strings of bytes}
- *Gen*: choose uniform byte $k \in \mathcal{K} = \{0, \ldots, 255\}$
- $Enc_k(m_1 \ldots m_t)$: output $c_1 \ldots c_t$, where $c_i := m_i \oplus k$
- $Dec_k(c_1 \ldots c_t)$: output $m_1 \ldots m_t$, where $m_i := c_i \oplus k$
- Verify that correctness holds…

Example

Key is 11001110.
Alice wants to send $00011010, 11100011, 00000000$
She sends

$$00011010 \oplus 11001110, 11100011 \oplus 11001110, 00000000 \oplus 11001110$$

$$= 11010100, 00101101, 11001110$$

Example

Key is 11001110.
Alice wants to send $00011010, 11100011, 00000000$
She sends

$$00011010 \oplus 11001110, 11100011 \oplus 11001110, 00000000 \oplus 11001110$$

$$= 11010100, 00101101, 11001110$$

Question: Should it worry Alice and Bob that the key itself was transmitted? Discuss

Example

Key is 11001110.
Alice wants to send $00011010, 11100011, 00000000$
She sends

$$00011010 \oplus 11001110, 11100011 \oplus 11001110, 00000000 \oplus 11001110$$

$$= 11010100, 00101101, 11001110$$

Question: Should it worry Alice and Bob that the key itself was transmitted? Discuss
No. Eve has no way of knowing that.

Is this cipher secure?

- No – only 256 possible keys!
  - Given a ciphertext, try decrypting with every possible key
  - If ciphertext is long enough, only one plaintext will "look like English" (use the vector method of the last set of slides).
- Can further optimize
  - First nibble of plaintext likely 0x4, 0x5, 0x6, 0x7 (assuming letters only)
  - Can reduce exhaustive search to 26 keys (how?)
  - Talk to your friends or blood enemies about this.

Sufficient key space principle

- The key space must be large enough to make exhaustive-search attacks impractical
  - How large do you think that is?
- Note: this makes some assumptions. . .
  - English-language plaintext
  - Ciphertext sufficiently long so only one valid plaintext

Is this cipher secure if we are transmitting numbers?

If Alice sends Bob a Document in English via Byte-Shift then insecure!

What if Alice sends Bob a credit card number? Discuss

Is this cipher secure if we are transmitting numbers?

If Alice sends Bob a Document in English via Byte-Shift then insecure!

What if Alice sends Bob a credit card number? Discuss
Credit Card Numbers also have patterns:

1. Visa cards always begin with 4
2. American Express always begins 34 or 37
3. Mastercard starts with 51 or 52 or 53 or 54.

Upshot: If Eve knows what kind of information is being transmitted (English, Credit Card Numbers, numbers on checks) she can use this to make any cipher with a small key space insecure.

Affine, Quadratic, Cubic, and Polynomial Ciphers

Affine Cipher

Recall: Shift cipher with shift s:

1. Encrypt via $x \to x + s \pmod{26}$.
2. Decrypt via $x \to x - s \pmod{26}$.

We replace $x + s$ with more elaborate functions

Definition: The Affine cipher with a, b:

1. Encrypt via $x \to ax + b \pmod{26}$.
2. Decrypt via $x \to a^{-1}(x - b) \pmod{26}$

Affine Cipher

Recall: Shift cipher with shift s:

1. Encrypt via $x \to x + s \pmod{26}$.
2. Decrypt via $x \to x - s \pmod{26}$.

We replace $x + s$ with more elaborate functions

Definition: The Affine cipher with a, b:

1. Encrypt via $x \to ax + b \pmod{26}$.
2. Decrypt via $x \to a^{-1}(x - b) \pmod{26}$

Does this work? Vote YES or NO or OTHER

Affine Cipher

Recall: Shift cipher with shift s:

1. Encrypt via $x \to x + s \pmod{26}$.
2. Decrypt via $x \to x - s \pmod{26}$.

We replace $x + s$ with more elaborate functions

Definition: The Affine cipher with a, b:

1. Encrypt via $x \to ax + b \pmod{26}$.
2. Decrypt via $x \to a^{-1}(x - b) \pmod{26}$

Does this work? Vote YES or NO or OTHER Answer: OTHER

Affine Cipher

Recall: Shift cipher with shift s:

1. Encrypt via $x \to x + s \pmod{26}$.
2. Decrypt via $x \to x - s \pmod{26}$.

We replace $x + s$ with more elaborate functions

Definition: The Affine cipher with a, b:

1. Encrypt via $x \to ax + b \pmod{26}$.
2. Decrypt via $x \to a^{-1}(x - b) \pmod{26}$

Does this work? Vote YES or NO or OTHER Answer: OTHER
$2x + 1$ does not work: 0 and 13 both map to 1.

Affine Cipher

Recall: Shift cipher with shift s:

1. Encrypt via $x \to x + s \pmod{26}$.
2. Decrypt via $x \to x - s \pmod{26}$.

We replace $x + s$ with more elaborate functions

Definition: The Affine cipher with a, b:

1. Encrypt via $x \to ax + b \pmod{26}$.
2. Decrypt via $x \to a^{-1}(x - b) \pmod{26}$

Does this work? Vote YES or NO or OTHER Answer: OTHER
$2x + 1$ does not work: 0 and 13 both map to 1.
Need the map to be a bijection so it will have a unique inverse.

Affine Cipher

Recall: Shift cipher with shift s:

1. Encrypt via $x \rightarrow x + s \pmod{26}$.
2. Decrypt via $x \rightarrow x - s \pmod{26}$.

We replace $x + s$ with more elaborate functions

Definition: The Affine cipher with a, b:

1. Encrypt via $x \rightarrow ax + b \pmod{26}$.
2. Decrypt via $x \rightarrow a^{-1}(x - b) \pmod{26}$

Does this work? Vote YES or NO or OTHER Answer: OTHER
$2x + 1$ does not work: 0 and 13 both map to 1.
Need the map to be a bijection so it will have a unique inverse.

Condition on a, b so that $x \rightarrow ax + b$ is a bij: a rel prime to 26.
Condition on a, b so that a has an inv mod 26: a rel prime to 26.

Shift vs Affine

Shift: Key space is size 26

Affine: Key space is
$|\{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}| \times 26 = 12 \times 26 = 312$

In an Earlier Era Affine would be harder to crack than Shift.

Shift vs Affine

Shift: Key space is size 26

Affine: Key space is
$|\{1,3,5,7,9,11,15,17,19,21,23,25\}| \times 26 = 12 \times 26 = 312$

In an Earlier Era Affine would be harder to crack than Shift.

Today They are both easy to crack.

Both Need: The Is English algorithm. Reading through 312 transcripts to see which one looks like English would take A LOT of time!

The Quadratic Cipher

Definition: The Quadratic cipher with a, b, c:

1. Encrypt via $x \to ax^2 + bx + c \pmod{26}$.

The Quadratic Cipher

Definition: The Quadratic cipher with a, b, c:

1. Encrypt via $x \to ax^2 + bx + c \pmod{26}$.

Does this work? Vote YES or NO

The Quadratic Cipher

Definition: The Quadratic cipher with a, b, c:

1. Encrypt via $x \rightarrow ax^2 + bx + c \pmod{26}$.

Does this work? Vote YES or NO Answer: NO

The Quadratic Cipher

Definition: The Quadratic cipher with a, b, c:

1. Encrypt via $x \rightarrow ax^2 + bx + c \pmod{26}$.

Does this work? Vote YES or NO Answer: NO

1. No easy test for Invertibility (depends on def of easy).
2. It turns out that every quadratic function mod 26 is an affine function.

The Polynomial Cipher

Definition: Poly Cipher with poly p (coefficients in $\{0, \ldots, 25\}$.

1. Encrypt via $x \to p(x) \pmod{26}$.
2. Decrypt via $x \to p^{-1}(x) \pmod{26}$.

Given a polynomial over mod 26 (or any mod) does it have an inverse? What is the complexity of this problem?
Vote: P, NP-complete, unknown to science.

The Polynomial Cipher

Definition: Poly Cipher with poly p (coefficients in $\{0, \ldots, 25\}$.

1. Encrypt via $x \rightarrow p(x) \pmod{26}$.
2. Decrypt via $x \rightarrow p^{-1}(x) \pmod{26}$.

Given a polynomial over mod 26 (or any mod) does it have an inverse? What is the complexity of this problem?
Vote: P, NP-complete, unknown to science.
Unknown to Science but if over mod a prime then in P.

The Polynomial Cipher

Definition: Poly Cipher with poly p (coefficients in $\{0, \ldots, 25\}$.

1. Encrypt via $x \rightarrow p(x) \pmod{26}$.
2. Decrypt via $x \rightarrow p^{-1}(x) \pmod{26}$.

Given a polynomial over mod 26 (or any mod) does it have an inverse? What is the complexity of this problem?
Vote: P, NP-complete, unknown to science.
Unknown to Science but if over mod a prime then in P.

Course website, Notes, has pointer to blog of mine on this. Some of the comments have theorems and pointers to the literature.

The Polynomial Cipher

Definition: Poly Cipher with poly p (coefficients in $\{0, \ldots, 25\}$.

1. Encrypt via $x \rightarrow p(x) \pmod{26}$.
2. Decrypt via $x \rightarrow p^{-1}(x) \pmod{26}$.

Given a polynomial over mod 26 (or any mod) does it have an inverse? What is the complexity of this problem?
Vote: P, NP-complete, unknown to science.
Unknown to Science but if over mod a prime then in P.

Course website, Notes, has pointer to blog of mine on this. Some of the comments have theorems and pointers to the literature.

The first place The Polynomial Cipher appeared was

The Polynomial Cipher

Definition: Poly Cipher with poly p (coefficients in $\{0, \ldots, 25\}$.

1. Encrypt via $x \rightarrow p(x) \pmod{26}$.
2. Decrypt via $x \rightarrow p^{-1}(x) \pmod{26}$.

Given a polynomial over mod 26 (or any mod) does it have an inverse? What is the complexity of this problem?
Vote: P, NP-complete, unknown to science.
Unknown to Science but if over mod a prime then in P.

Course website, Notes, has pointer to blog of mine on this. Some of the comments have theorems and pointers to the literature.

The first place The Polynomial Cipher appeared was

my 3-week summer course on crypto for High School Students.

So, as the kids say, its not a thing.

General Substitution Cipher

Shift and Affine were good for Alice and Bob since

1. Easy to encrypt, Easy to decrypt
2. Short Key: Roughly 5 bits for Shift, 10 bits for Affine.

Definition: Gen Sub Cipher with perm f on $\{0, \ldots, 25\}$.

1. Encrypt via $x \to f(x)$.
2. Decrypt via $x \to f^{-1}(x)$

1. Key is now permutation, roughly 125 bits.
2. Encrypt and Decrypt slightly harder

General Substitution Cipher

Shift and Affine were good for Alice and Bob since

1. Easy to encrypt, Easy to decrypt
2. Short Key: Roughly 5 bits for Shift, 10 bits for Affine.

Definition: Gen Sub Cipher with perm f on $\{0, \ldots, 25\}$.

1. Encrypt via $x \to f(x)$.
2. Decrypt via $x \to f^{-1}(x)$

1. Key is now permutation, roughly 125 bits.
2. Encrypt and Decrypt slightly harder

Uncrackable! Eve has to go through all 26! possibilities!!

General Substitution Cipher

Shift and Affine were good for Alice and Bob since

1. Easy to encrypt, Easy to decrypt
2. Short Key: Roughly 5 bits for Shift, 10 bits for Affine.

Definition: Gen Sub Cipher with perm f on $\{0, \ldots, 25\}$.

1. Encrypt via $x \rightarrow f(x)$.
2. Decrypt via $x \rightarrow f^{-1}(x)$

1. Key is now permutation, roughly 125 bits.
2. Encrypt and Decrypt slightly harder

Uncrackable! Eve has to go through all 26! possibilities!!

NOT EVEN CLOSE! Eve can use Freq Analysis

Freq Analysis

Alice sends Bob a LONG text encrypted by Gen Sub Cipher.
Eve finds freq of letters, pairs, triples, ...

Text in English.

1. Can use known freq: *e* is most common letter, *th* is most common pair.
2. If Alice is telling Bob about Mid East Politics than may need to adjust: *q* is more common (Iraq, Qatar) and some words more common.

Silly Counter Example – Pangrams

Pangrams: Sentence where each letter occurs at least once.

Short Panagrams ruin Freq analysis. Here are some:

1. The quick brown fox jumps over the lazy dog.
2. Pack my box with five dozen liquor jugs.
3. Amazingly few discotheques provide jukeboxes.
4. Watch Jeopardy! Alex Trebek's fun TV quiz game.

Silly Counter Example – Lipograms

Lipograms: A work that omits one letter

1. Gadsby is a 50,000-word novel with no *e*.
2. Eunoia is a 5-chapter novel, indexed by vowels. Chapter A only use the vowel A, etc.
3. How I met your mother, Season 9, Episode 9: Lily and Robin challenge Barney to get a girl's phone number without using the letter *e*.

We are not going to deal with this sillyness!
We assume long normal texts!

Alternatives to Gen Sub (History)

In the Year 2018 Alice can easily generate a random permutation of $\{a, \ldots, z\}$ and send it to Bob.

In the Year 1018 Alice needs a way to encode a random-looking permutation of $\{a, \ldots, z\}$ and transmit it to Bob. So need SHORT description of random-looking perm.

1. Two ways to do this will be on the HW.
2. Foreshadowing the need for a short description of a random-looking string of bits which we will be central later in this course.

The Vigenére Cipher

The Vigenère cipher

EDUCATION NOTE: In class we started but did not finish Vig Cipher. I include everything on Vig Cipher in both this set of slides and the next.

The Vigenère cipher

Key: A word or phrase. Example: *dog* = *(3,14,6)*.
Easy to remember and transmit.
Example using *dog*.
Shift 1st letter by 3
Shift 2nd letter by 14
Shift 3nd letter by 6
Shift 4th letter by 3
Shift 5th letter by 14
Shift 6th letter by 6, etc.

Jacob Prinz is a Physics Major
jacob prinz isaph ysics major

encrypts to

MOIRP VUWTC WYDDN BGOFG SDXUU

The Vigenère cipher

Key: $k = (k_1, k_2, \ldots, k_n)$.
Encrypt (all arithmetic is mod 26)

$$Enc(m_1, m_2, \ldots, m_N) =$$

$$m_1 + k_1, m_2 + k_2, \ldots, m_n + k_n,$$

$$m_{n+1} + k_1, m_{n+2} + k_2, \ldots, m_{n+n} + k_n,$$

$$\ldots$$

Decrypt Decryption just reverse the process

The Vigenère cipher

- Size of key space?
  - If keys are 14-char then key space size $26^{14} \approx 2^{66}$
  - If variable length keys, even more.
  - Brute-force search infeasible
- Is the Vigenère cipher secure?
- Believed secure for many years. . .
- Might not have even been secure then. . .

Cracking Vig cipher: Step One-find Keylength

Assume T is a text encoded by Vig, key length L unknown.
For $0 \le i \le L-1$, letters in pos $\equiv i \pmod{26}$ – same shift.
Look for a sequence of (say) 3-letters to appear (say) 4 times.

Example: aiq appears in the
57-58-59th slot, 87-88-89th slot 102-103-104th slot
162-163-164th slot

Important: Very likely that aiq encrypted the same 3-letter sequence and hence the length of the key is a divisor of
87-57=30 102-87=15 162-102=60
The only possible L's are 1,3,5,15.

Good Enough: We got the key length down to a small finite set.

Important Point about letter Freq

Assume (and its roughly true): In an English text of length N:

e occurs $\sim 13\%$ t occurs $\sim 9\%$ a occurs $\sim 8\%$

Etc- other letters have frequencies that are true for all texts.

Important Point about letter Freq

Assume (and its roughly true): In an English text of length N:

e occurs $\sim 13\%$ t occurs $\sim 9\%$ a occurs $\sim 8\%$

Etc- other letters have frequencies that are true for all texts.

Assume (and its roughly true): In an English text of length N, if $i \ll N$, then if you take every ith letter of T:

e occurs $\sim 13\%$ t occurs $\sim 9\%$ a occurs $\sim 8\%$

Etc- other letters same frequencies as normal texts.

Important Point about letter Freq

Assume (and its roughly true): In an English text of length N:

e occurs $\sim 13\%$ t occurs $\sim 9\%$ a occurs $\sim 8\%$

Etc- other letters have frequencies that are true for all texts.

Assume (and its roughly true): In an English text of length N, if $i \ll N$, then if you take every ith letter of T:

e occurs $\sim 13\%$ t occurs $\sim 9\%$ a occurs $\sim 8\%$

Etc- other letters same frequencies as normal texts.

Relevant to us:

$\vec{q}$ freq of every Lth letter: then $\sum_{i=1}^{26} q_i^2 \approx 0.065$.

$\vec{q}$ is NOT (we won't define that rigorously): $\sum_{i=1}^{26} q_i^2$ MUCH lower.

Cracking Vig cipher: Step One-find Keylength

Let K be the set of possible key lengths. K is small. For every $L \in K$:

- Form a stream of every Lth character.
- Find the frequencies of that stream: $\vec{q}$.
- Compute $Q = \sum q_i^2$
- If $Q \approx 0.065$ then YES L is key length.
- If Q much less than 0.065 then NO L is not key length.
- One of these two will happen
- Just to make sure, check another stream.

Note: Differs from Is English:

Is English wanted to know if the text was actually English

What we do above is see if the text has same dist of English, but okay if diff letters. E.g., if z is 13%, a is 9%, and other letters have roughly same numbers as English then we know the stream is SOME Shift. We later use Is English to see which shift.

A Note on Finding Keylength

We presented one method:

1. Find phrase of length x appearing y times. Differences D.
2. K is set of divisors of all $L \in D$. Correct keylength in K.
3. Test $L \in K$ for key length until find one that works.

Alternative just try all key lengths up to a certain length:

1. Let $K = \{1, \ldots, 100\}$ (I am assuming key length ≤ 100).
2. Test $L \in K$ for key length until find one that works.

Note: With modern computers use Method 2. In days of old eyeballing it made method 1 reasonable.

Cracking the Vig cipher: Step Two-Freq Anal

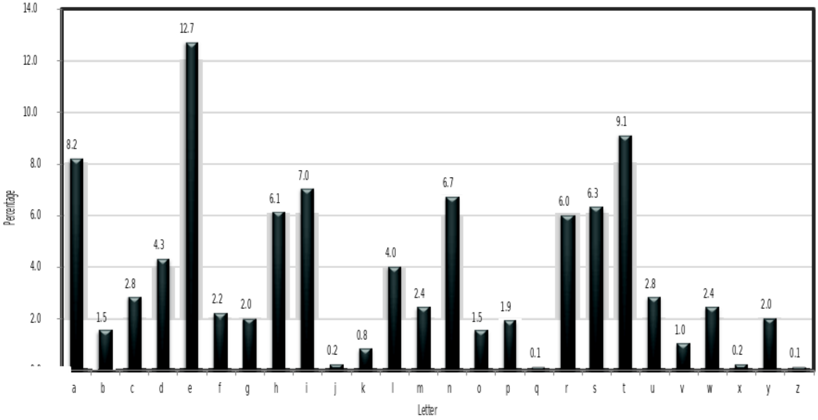
After Step One we have the key length L. Note:

- Every L^{th} character is "encrypted" using the same shift.
- Important: Letter Freq still hold if you look at every L 14th letter!

Step Two:

1. Separate text T into L streams depending on position mod L
2. For each steam try every shift and use Is English to determine which shift is correct.
3. You now know all shifts for all positions. Decrypt!

Using plaintext letter frequencies



Byte-wise Vigenère cipher

- The key is a string of bytes
- The plaintext is a string of bytes
- To encrypt, XOR each character in the plaintext with the next character of the key
  - Wrap around in the key as needed
- Decryption just reverses the process.

Note: Decryption and Encryption both use XOR with same key.
Note: Can be cracked as original Vig can be cracked.