Final consideration about columnar transposition cipher

- Clearly using more keys (in the double transposition cipher we just use 2 keys) the bad guy needs more time (i.e., computation) in order to deciphering a message.

- However columnar transposition ciphers are breakable mainly because they do not change characters but just their positions.

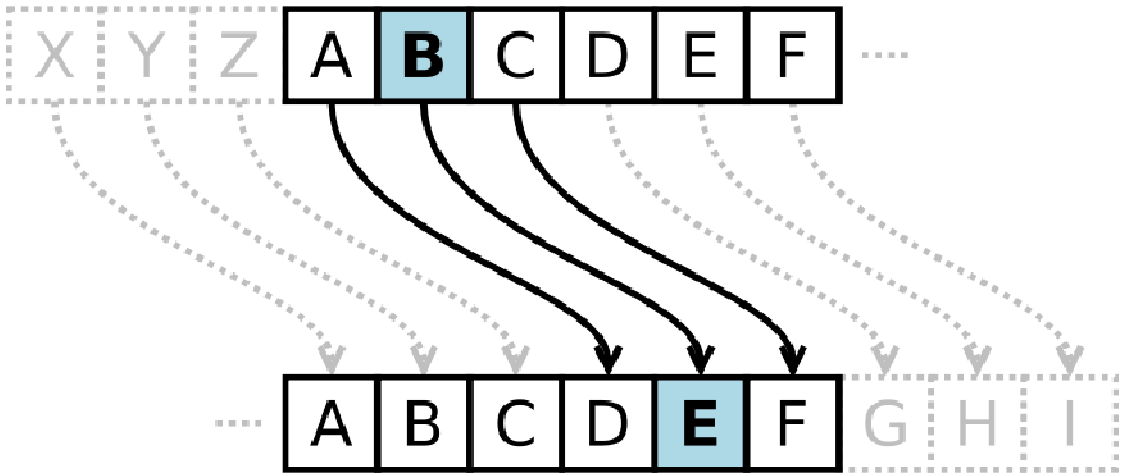
Recall coding

- Recall that for now (later we will use ASCII or binary code) in order to make things simpler we suppose to codify the english alphabet in the following way:

- A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Caesar cipher

- It is one of the simplest and most widely known encryption techniques.
- It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- Let us suppose A is codifed with 0, B with 1,…, Z with 25. Moreover let us indicate with *a mod b* as the reminder of the division of *a* by *b*
- c = (m+k) mod 26
- m = (c-k) mod 26
- Example:
  - k = 3
  - “Data” → “Gdwd ”



Caesar chipher

- Only 26 possibilities with English Alphabet
- Linear with the size of the alphabet
- Brute Force search can decrypt

Exercise

- English alphabet:
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Decrypt the following sentence
  - AOL XBPJR IYVDU MVE QBTWZ VCLY AOL SHGF KVN
    - k=1: ZNK…
    - k=2: YMJ…
    - k=3: XLI…
    - K=4: WKH…
    - …
    - K=7: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

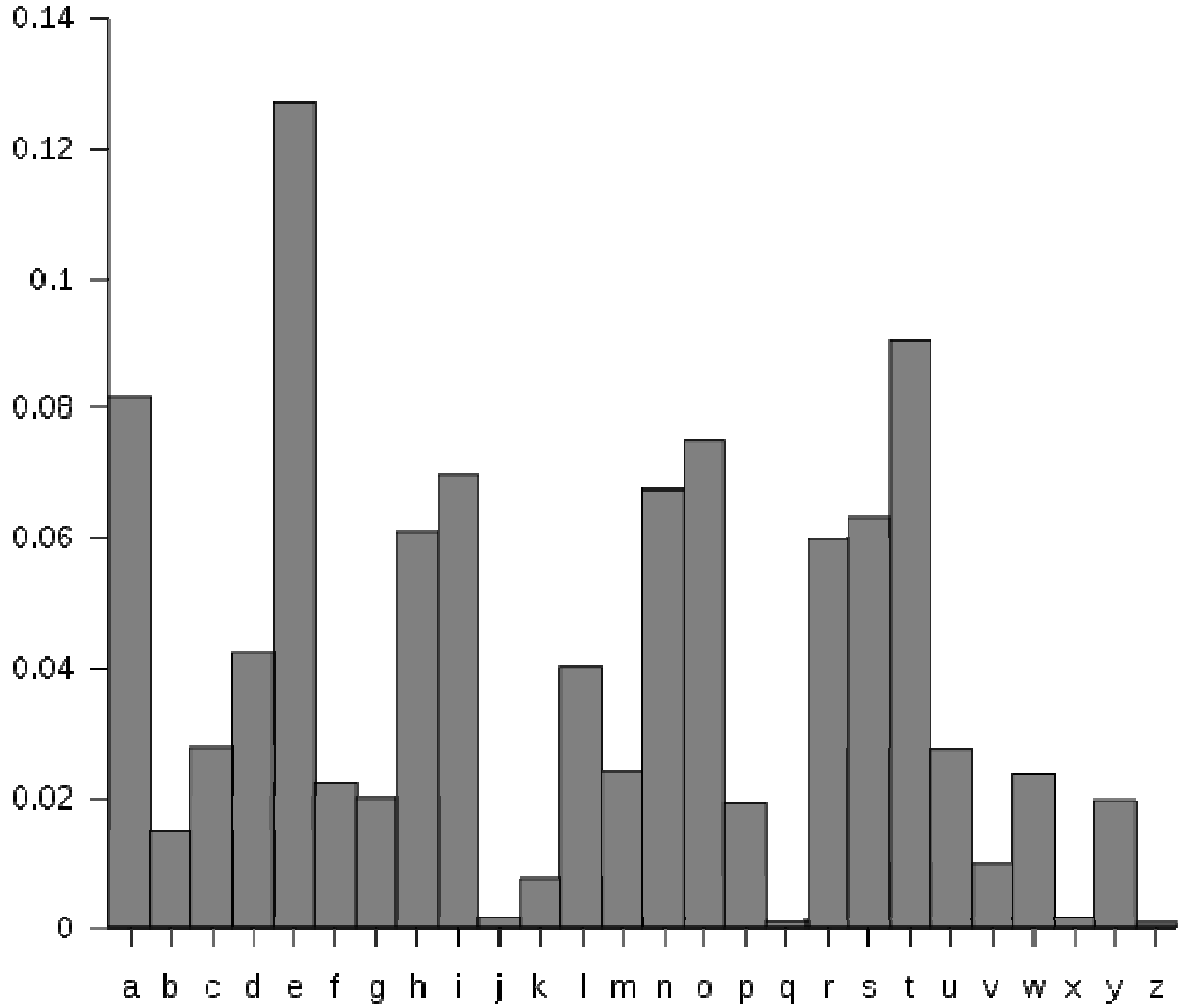
Monoalphabetic cipher

- It uses random letter substitution
- Key is 26 letters long
- Example:
  - Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - Cipher:DKVQFIBJWPESCXHTMYAUOLRGZN
  - “DATA”=“QDUD”
- Caesar cipher is a particular case of monoalphabetic cipher

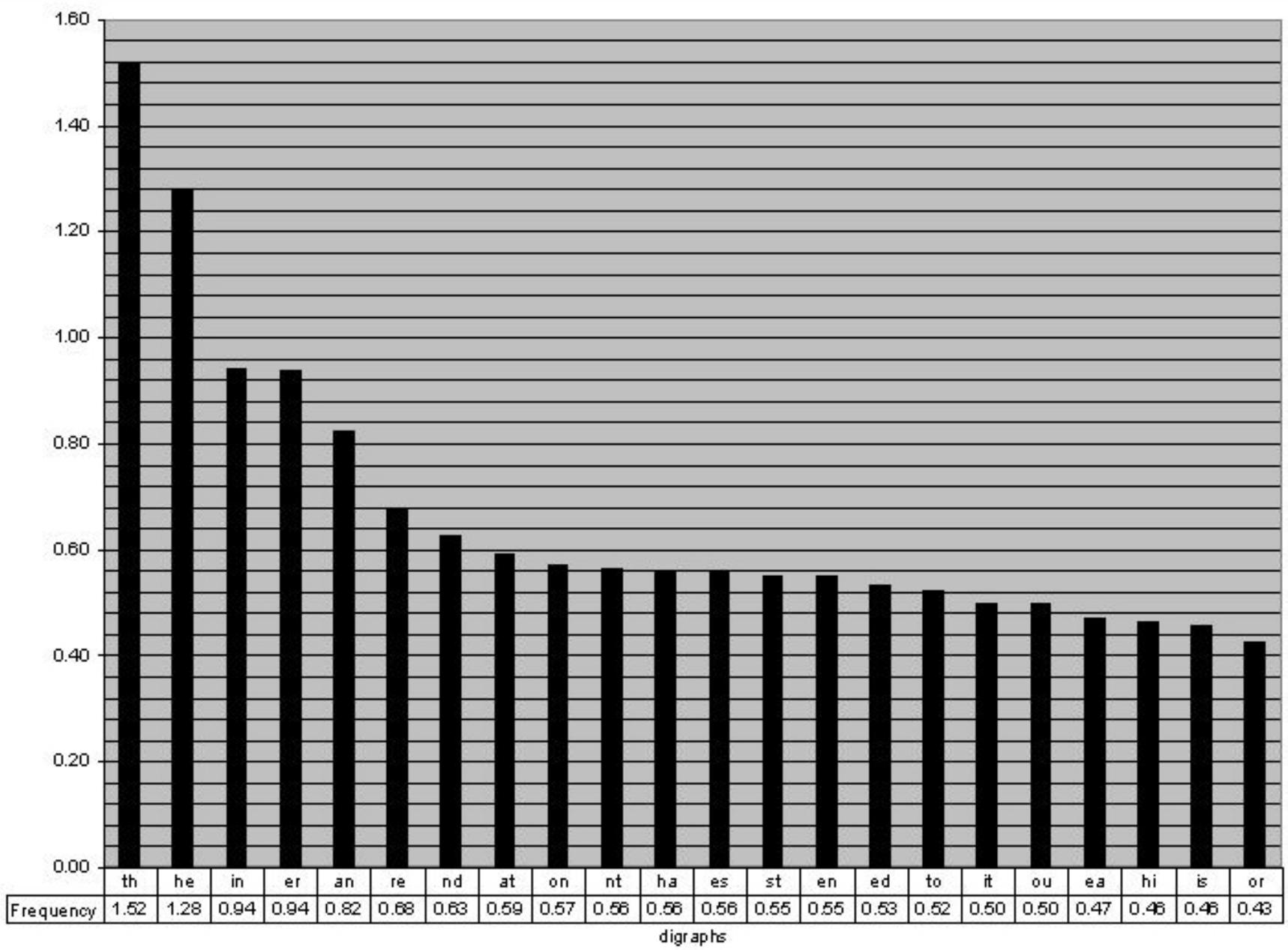
Monoalphabetic cipher

- $26! \approx 2^{88}$ Combinations, an huge number!
- However, it is easy to decrypt because language gives lots of hints:
  - Most common letters: E and T
  - Most common single letters: I or A
  - Most common first letters in a word: T and A
  - Most used digraph (pairs of letters): TH and HE
- Using the letter frequency statistics and clever observations, make guesses at probable letter substitutions.

English letters frequency



English digraphs frequency



Other statistics

- Most Common Digraphs (Listed in order of frequency)
  - TH HE AN IN ER ON RE ED ND HA AT EN ES OF NT EA TI TO IO LE IS OU AR AS DE RT VE SE OR AL TE CO
- Most Common Trigraphs (Listed in order of frequency)
  - THE AND THA ENT ION TIO FOR NDE HAS NCE TIS OFT MEN ING EDT STH
- Most Common Double Letters (Listed in order of frequency)
  - SS EE TT FF LL MM OO
- Most Common Two-Letter Words (Listed in order of frequency)
  - OF TO IN IT IS BE AS AT SO WE HE BY OR ON DO IF ME MY UP AN GO NO US AM
- Most Common Three-Letter Words (Listed in order of frequency)
  - THE AND FOR BUT NOT YOU ALL ANY CAN HAD HER WAS ONE OUR OUT DAY GET HAS HIM HIS HOW MAN NEW NOW OLD SEE TWO WAY WHO BOY DID ITS LET PUT SAY SHE TOO USE
- Most Common Four-Letter Words (Listed in order of frequency)
  - THAT WITH HAVE THIS WILL YOUR FROM THEY KNOW WANT BEEN GOOD MUCH SOME TIME VERY WHEN COME HERE JUST LIKE LONG MAKE MANY MORE ONLY OVER SUCH TAKE THAN THEM WELL WERE

Polyalphabetic ciphers

- A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets.
  - Al Kindi cipher (about 850)
  - Alberti cipher (about 1467)
  - Vigenère cipher (1553 and later 19th century)
  - The Enigma (WW II)

Vigenère cipher

- The Vigenère cipher (16 Century) is a method of encrypting alphabetic text by using a series of different Caesar ciphers based on the letters of a keyword.
- For three centuries it resisted all attempts to break it!
- To encrypt, a table of alphabets can be used, termed a **tabula recta, Vigenère square, or Vigenère table**.
- It consists of the alphabet written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar ciphers.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Recall coding

- Recall that for now (later we will use ASCII or binary code) in order to make things simpler we suppose to codify the english alphabet in the following way:

- A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Vigenère cipher

- Let us choose a key
- The key is repeated until it matches the length of the plaintext, we obtain the actual key K
- Let us denote as
  - $m=m_1m_2…m_n$
  - $K=k_1k_2…k_n$
  - $c=c_1c_2…c_n$
- $c_i = (m_i+k_i) \bmod 26$
- $m_i = (c_i-k_i) \bmod 26$
- Example:
  - Key = KEY plaintext = HELLO K= KEYKE
  - $c_1= m_1 + k_1 = (7 + 10) \bmod 26 = 17$ corresponding to the character R
  - ……
  - ciphertext = RIJVS

  Or we can use the table.

Vigenère cipher

- How do we get back the plaintext from a ciphertext obtained by using the Vigenère cipher if we now the key?
- Example:

  c = TELFPFF Key = BCD

  k = BCDBCDB

m_1 = (19 – 1) mod 26 = 18 corresponding to S
m_2= (4 – 2) mod 26 = 2 corresponding to C
m_3 = (11 – 3) mod 26 = 8 corresponding to I
m_4 = (5 – 1) mod 26 = 4 corresponding to E
m_5 = (15 – 2) mod 26 = 13 corresponding to N
m_6 = (5 – 3) mod 26 = 2 corresponding to C
m_7 = (5 – 1) mod 26 = 4 corresponding to E

m = SCIENCE

Or we can use the table (exercise).

Vigenère cipher

- The same letter of the plaintext can be enciphered as different letters in the ciphertext letters.
  - Simple frequency analysis does not work.
- Weakness: the key repeats.
  - If we know the key's length (let us say *n*), then the cipher text can be treated as *n* Caesar ciphers, which individually are easily broken.
- The Kasiski test can help to deduce the key length (especially when the length of the key is much smaller than the length of the plaintext).
  - If two identical chunks of plaintext are separated by some multiple of the keylength, they will generate identical chunks of ciphertext.
  - Look for repeated groups of ciphertext letters, chart their separation distance.
  - All factors of the distance are possible key lengths – a key of length one is just a simple Caesar cipher

Example

- K=CAN
- Plaintext: `TO BE OR NOT TO BE`
- Key: `CA NC AN CAN CA NC`
- Chiphertext: `VO OG OE POG VO OG`

Digraph	Distance	Factors
VO	9	3,9
OO	9	3,9
OG	9	3,9
OG	4	2,4
OG	5	5

3 and 9 (since they appear more times) are more likely to be the length of the key than 2, 4 and 5.

One-Time pad (Vernam cipher): XOR

XOR (exclusive or) is a logical operation that outputs true whenever both inputs differ (one is true, the other is false).

XOR Truth Table (A XOR B)

Input		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

0 = FALSE
1 = TRUE

One-Time pad (Vernam cipher)

- Choose a symmetric random key k as long as the message (uniformly distributed in $\{0,1\}^n$)
- $E_K(m) = m \text{ XOR } K = c$
- $D_K(c) = c \text{ XOR } K = m$

Why XOR?

- The idea behind exclusive-OR encryption is that it is impossible to reverse the operation without knowing the initial value of one of the two arguments.
- For example, if you XOR two variables of unknown values, you cannot tell from the output what the values of those variables are. For example, if you take the operation A XOR B, and it returns TRUE, you cannot know whether A is FALSE and B is TRUE, or whether B is FALSE and A is TRUE. Furthermore, even if it returns FALSE, you cannot be certain if both were TRUE or if both were FALSE.
- If, however, you know either A or B it is entirely reversible, unlike logical-AND and logical-OR. For exclusive-OR, if you perform the operation A XOR TRUE and it returns a value of TRUE you know A is FALSE, and if it returns FALSE, you know A is true. Exclusive-OR encryption works on the principle that if you have the encrypted string and the encryption key you can always decrypt correctly. If you don't have the key, it is impossible to decrypt it without making entirely random keys and attempting each one of them until the decryption program's output is something akin to readable text.

Perfect secrecy

- Space of messages $\{0,1\}^n$
- Shannon Secrecy
  - $Pr(M = m \mid E_K(m) = c) = Pr(M = m)$
  - Probability of guessing the plaintext knowing the ciphertext = probability of guessing plaintext without knowing ciphertext
- Perfect secrecy
  - $Pr(E_K(m) = c) = Pr(E_K(m') = c)$
  - Probability of any message giving a ciphertext is the same
- If the sizes of the space of messages and keys are equal the above definitions are equivalent

Perfect secrecy

Theorem: One-time pad has perfect secrecy.

- Claude Shannon proved, using information theory considerations, that the one-time pad has the perfect secrecy property; that is, the ciphertext C gives absolutely no additional information about the plaintext.
- This is because, given a truly random key which is used only once, a ciphertext can be translated into *any* plaintext of the same length, and all are equally likely.
- Given perfect secrecy, in contrast to conventional symmetric encryption, One time pad is immune even to brute-force attacks. Trying all keys simply yields all plaintexts, all equally likely to be the actual plaintext. Even with known plaintext, like part of the message being known, brute-force attacks cannot be used, since an attacker is unable to gain any information about the parts of the key needed to decrypt the rest of the message. The parts that are known will reveal only the parts of the key corresponding to them, and they correspond on a strictly one-to-one basis; no part of the key is dependent on any other part.

One-Time pad (Vernam cipher): example

We will use ASCII code:

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

One-Time pad (Vernam cipher): example

m = “HELLO”

Hexadecimal ASCII codes:	Binary:
H = 48	01001000
E = 45	01000101
L = 4c	01001100
L = 4c	01001100
O = 4f	01001111

k = “ABCDE”

Hexadecimal ASCII codes:	Binary:
A = 41	01000001
B = 42	01000010
C = 43	01000011
D = 44	01000100
E= 45	01000101

m= 01001000 01000101 01001100 01001100 01001111
k= 01000001 01000010 01000011 01000100 01000101

C= 00001001 00000111 00001111 00001000 00001010

One-Time pad (Vernam cipher)

Despite Shannon's proof of its security, the one-time pad has serious drawbacks in practice because it requires:

- Each key works only once
- Works with fixed length messages
- Key length = message length
- Not very practical (preventive agreement on how to generate and use keys).

The secure generation and exchange of the one-time pad values is not very practical (notice that the security of the one-time pad is only as secure as the security of the one-time pad exchange).

In practice…

- Security in cryptography is measured by
  - Time resources
  - Memory resources
  - Computational resources
- (Almost) any currently used cipher can be broken by using enough time and computation
- If the best possible scheme will take 10 million years to break using all of the computers in the world, then it can be considered reasonably secure!

In practice…

- Security depends on the space of possible keys
- If it is big enough, then a lot of time and computational resources are required to guess a key
  - 20 bits, about 1 million keys: currently not safe
  - 1024 bits, more than 10^{102} keys: currently safe, for how long?

In practice…

- The adversary has limited
  - Time
  - Computational resources
  - Memory