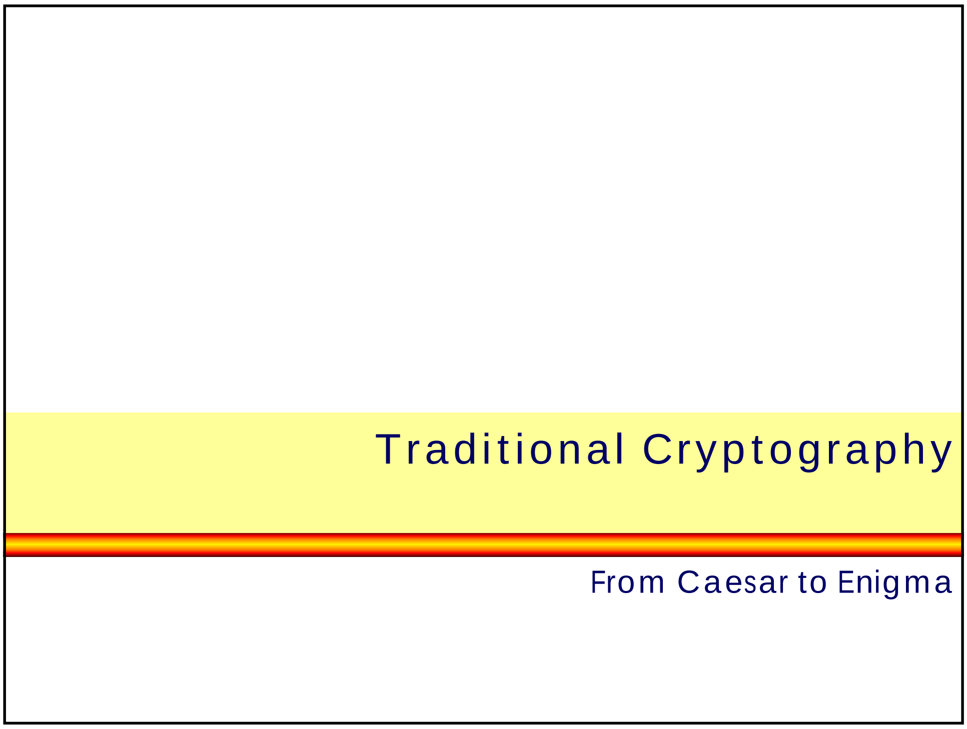
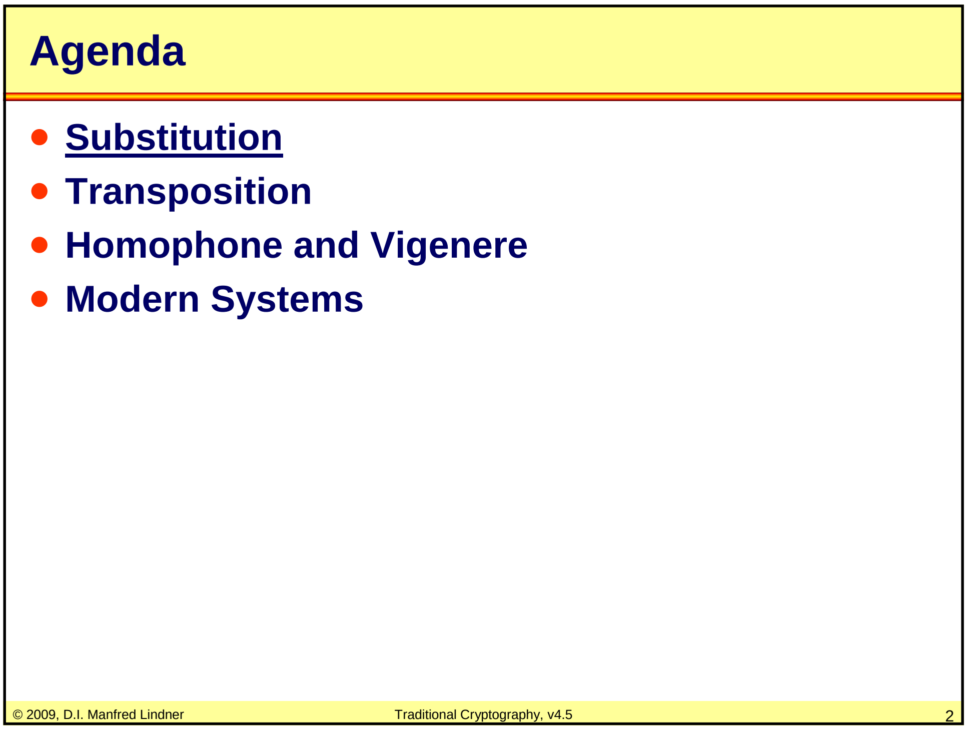


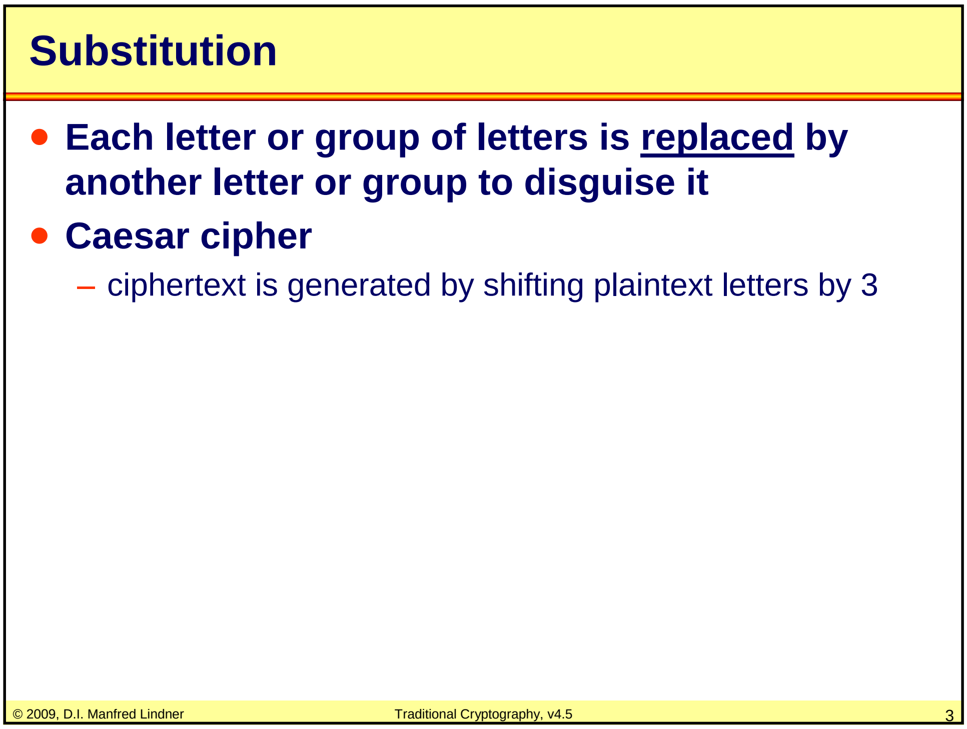
Traditional Cryptography

From Caesar to Enigma



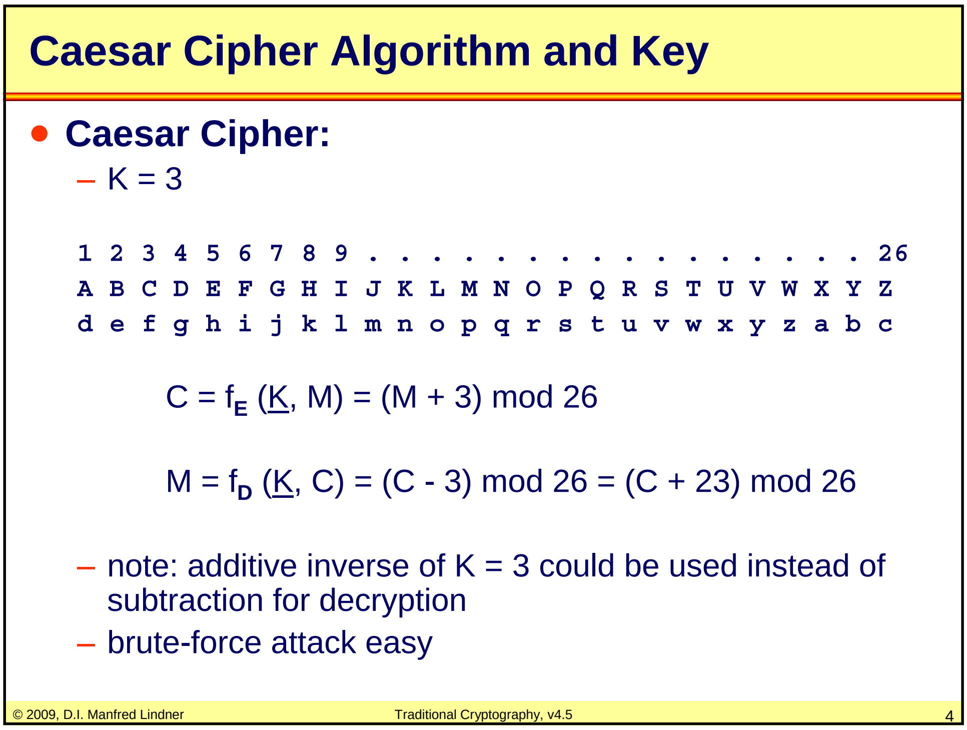
Agenda

- **Substitution**
- **Transposition**
- **Homophone and Vigenere**
- **Modern Systems**



Substitution

- **Each letter or group of letters is <u>replaced</u> by another letter or group to disguise it**
- **Caesar cipher**
  - ciphertext is generated by shifting plaintext letters by 3



Caesar Cipher Algorithm and Key

- **Caesar Cipher:**
  - K = 3

1	2	3	4	5	6	7	8	9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	26
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
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

$$C = f_E\ (\underline{K}, M) = (M + 3) \bmod 26$$

$$M = f_D\ (\underline{K}, C) = (C - 3) \bmod 26 = (C + 23) \bmod 26$$

  - note: additive inverse of K = 3 could be used instead of subtraction for decryption
  - brute-force attack easy

Substitution

- **Generalization of Caesar cipher**
  - ciphertext is generated by shifting plaintext letters by secret value k (k = 1 … 26)
- **General system of substitution technique is called mono-alphabetic substitution**
  - each letter of the plaintext is mapped onto some other letter, but in every case always mapped to the same
    - a letter in the ciphertext represents a single cleartext letter
  - there are 26! = 4x10^{26} possibilities
    - 400 000 000 000 000 000 000 000 000

Codeword and Mono-alphabetic

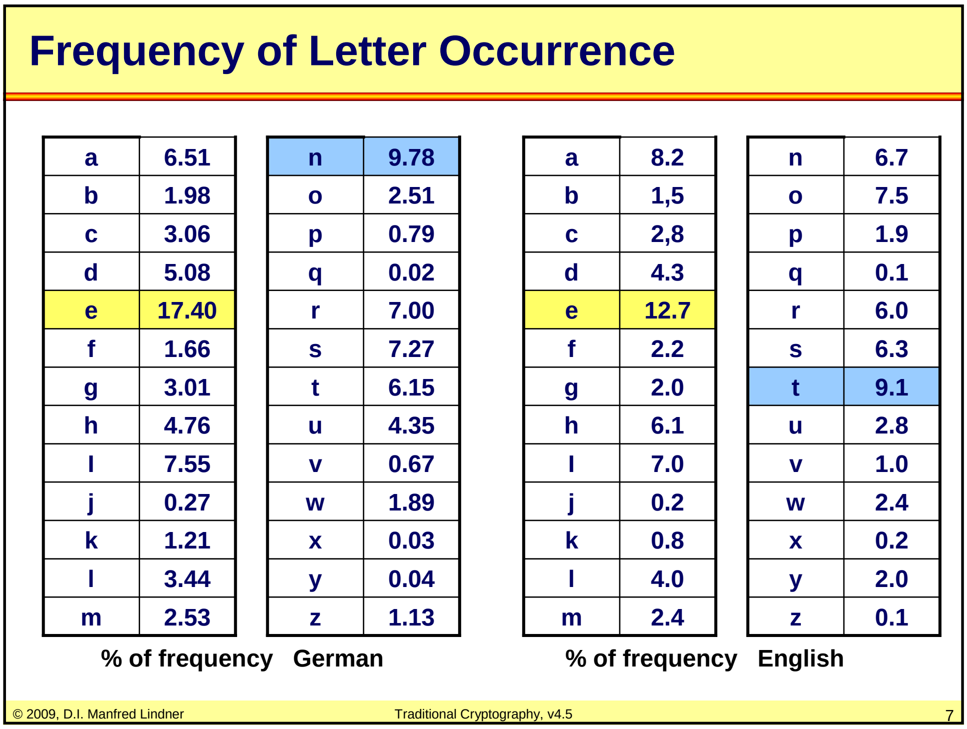
- **General system:**
  - random pattern for matching (code) table

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
x	v	r	q	u	w	s	y	z	g	h	k	m	o	p	c	a	f	t	c	b	d	j	i	l	n

  - brute-force attack will take very long
- **Codeword:**
  - GEHEIMSCHRIFT, Key = Letter K

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
n	o	p	q	u	v	w	x	y	z	g	e	h	i	m	s	c	r	f	t	a	b	d	j	k	l

  - brute-force attack not so easy



Frequency of Letter Occurrence

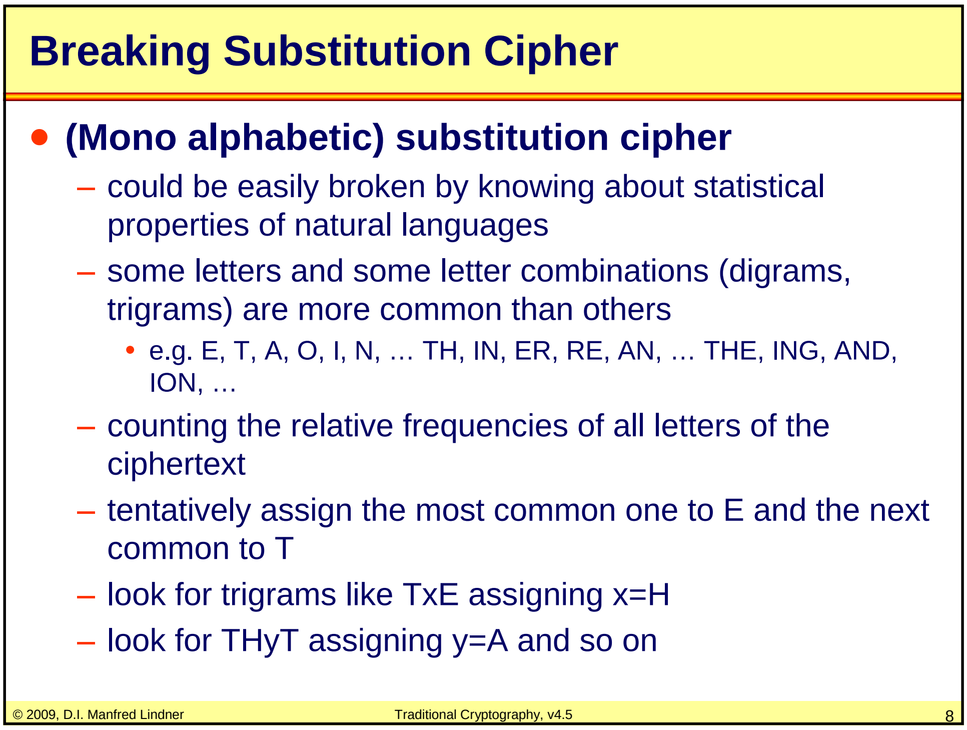
a	6.51	n	9.78
b	1.98	o	2.51
c	3.06	p	0.79
d	5.08	q	0.02
e	17.40	r	7.00
f	1.66	s	7.27
g	3.01	t	6.15
h	4.76	u	4.35
I	7.55	v	0.67
j	0.27	w	1.89
k	1.21	x	0.03
l	3.44	y	0.04
m	2.53	z	1.13

% of frequency German

a	8.2	n	6.7
b	1,5	o	7.5
c	2,8	p	1.9
d	4.3	q	0.1
e	12.7	r	6.0
f	2.2	s	6.3
g	2.0	t	9.1
h	6.1	u	2.8
I	7.0	v	1.0
j	0.2	w	2.4
k	0.8	x	0.2
l	4.0	y	2.0
m	2.4	z	0.1

% of frequency English

© 2009, D.I. Manfred Lindner Traditional Cryptography, v4.5 7



Breaking Substitution Cipher

- **(Mono alphabetic) substitution cipher**
  - could be easily broken by knowing about statistical properties of natural languages
  - some letters and some letter combinations (digrams, trigrams) are more common than others
    - e.g. E, T, A, O, I, N, … TH, IN, ER, RE, AN, … THE, ING, AND, ION, …
  - counting the relative frequencies of all letters of the ciphertext
  - tentatively assign the most common one to E and the next common to T
  - look for trigrams like TxE assigning x=H
  - look for THyT assigning y=A and so on

© 2009, D.I. Manfred Lindner Traditional Cryptography, v4.5 8

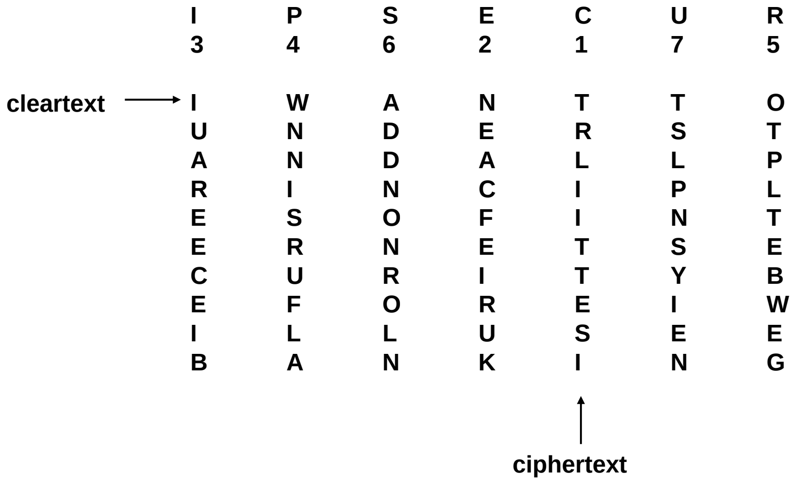
Agenda

- Substitution
- Transposition
- Homophone and Vigenere
- Modern Systems

Transposition (Permutation)

- Substitution cipher preserve order of the plaintext symbols but disguise them
- Transposition cipher reorder the plaintext symbols but do not disguise them
- one example
  - columnar transposition
  - cipher is keyed by a phrase not containing any repeated letters
  - purpose of the key is to number the columns
    - first column is under the key letter closest to the start of alphabet
  - plaintext is written horizontally
  - ciphertext is generated by reading the columns

Transposition Cipher Example



I	P	S	E	C	U	R
3	4	6	2	1	7	5
I	W	A	N	T	T	O
U	N	D	E	R	S	T
A	N	D	A	L	L	P
R	I	N	C	I	P	L
E	S	O	F	I	N	T
E	R	N	E	T	S	E
C	U	R	I	T	Y	B
E	F	O	R	E	I	W
I	L	L	U	S	E	E
B	A	N	K	I	N	G

Breaking Transposition Cipher 1

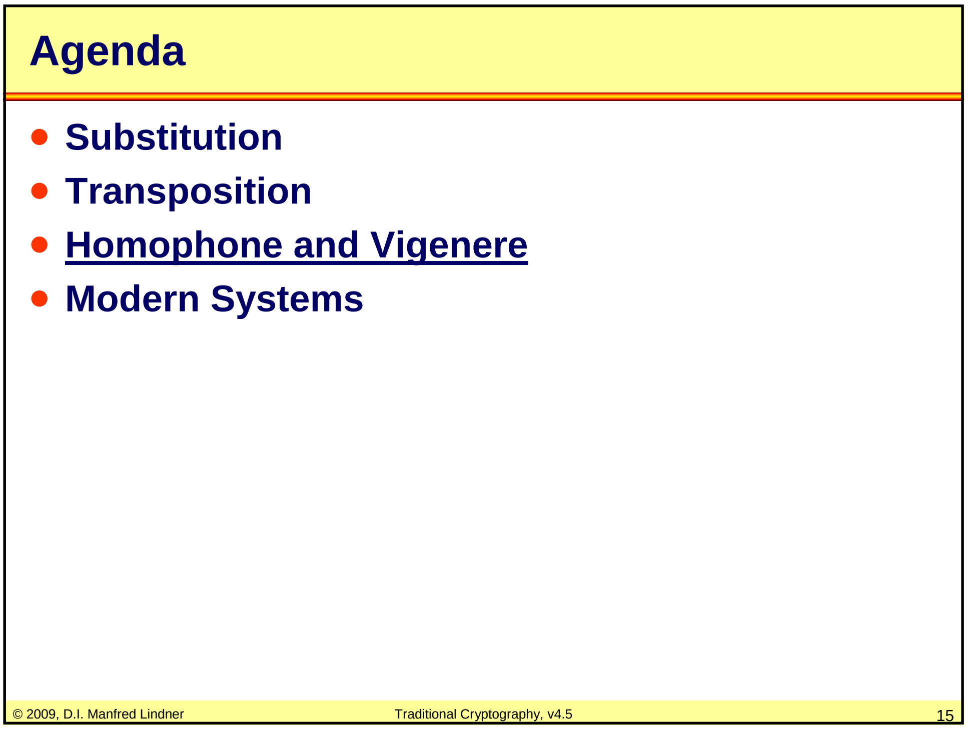
- First cryptanalyst must be aware about transposition cipher method
- Again can be broken by knowing about statistical properties of natural languages
  - because letters are not disguised in this case the frequency of ciphertext letters contains the statistic of the plaintext
  - next step is to guess number of columns
    - helpful are guessed words or guessed phrases which are suspected in the context of the original message and which are longer than the key phrase hence resulting in key phrase wrap-around
    - e.g. **I**NTERNE**T**SECURITY results in digrams IT, NS, TE, EC, RU, NR, EI, TT, SY because of key length 7
    - if key length would be 8 then digrams IS, NE, TC, EU, RR, NI, ET would have occurred instead

Breaking Transposition Cipher 2

- remaining step is to order the columns
  - when the number of columns k is small, each of the column pairs k(k-1) can be examined to see if digrams frequencies match those of the assumed plaintext language
  - the pair with the best match is assumed to be correctly positioned
  - now each remaining column is tentatively tried as successor to this pair
  - the column whose digram and trigram frequencies give the best match is assumed to be correct
  - the predecessor column is found in the same way
  - the entire process is continued until a potential ordering is found
- the chance that the plaintext will be recognizable (readable with some minor errors) at this point is very high and you can make adaptation to avoid these minor errors

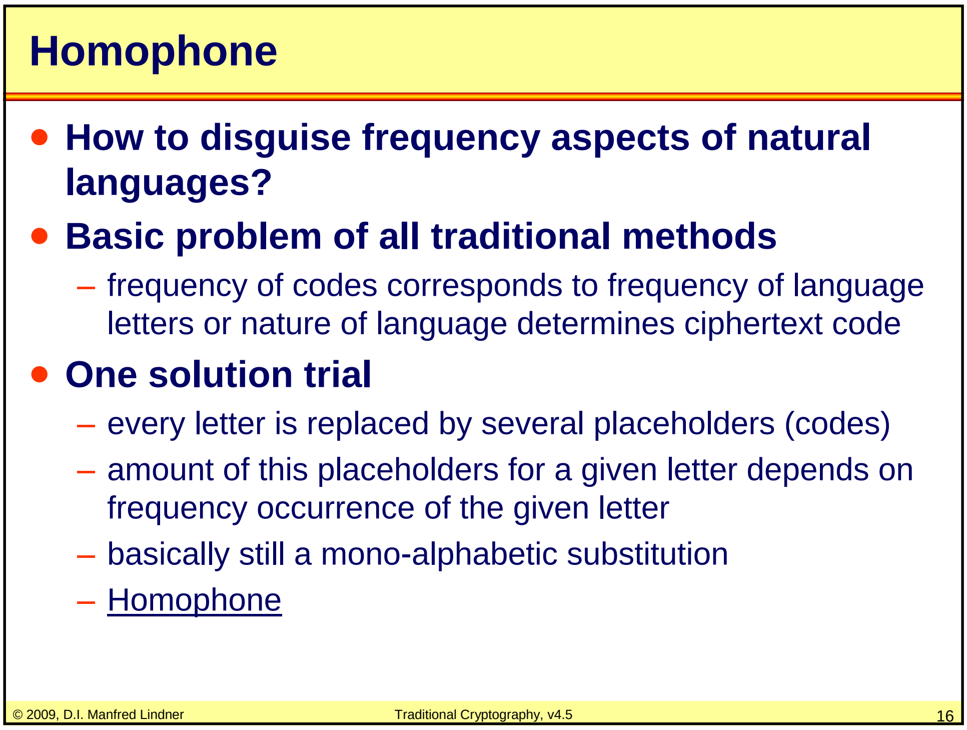
Transposition Cipher

- **Some transposition ciphers**
  - accept a fixed-length block of input and produce a fixed-length block of output
- **Such ciphers could be described**
  - by a list telling the order in which the letters are to be output
- **in our example:**
  - 70 character block cipher
  - with output order 5, 12, 19, 26, 33, 40, 47, 54, 61, 68, 4, 11, 18, …
    - that means plaintext input 5 is first to be output, 12 is second to be output and so on



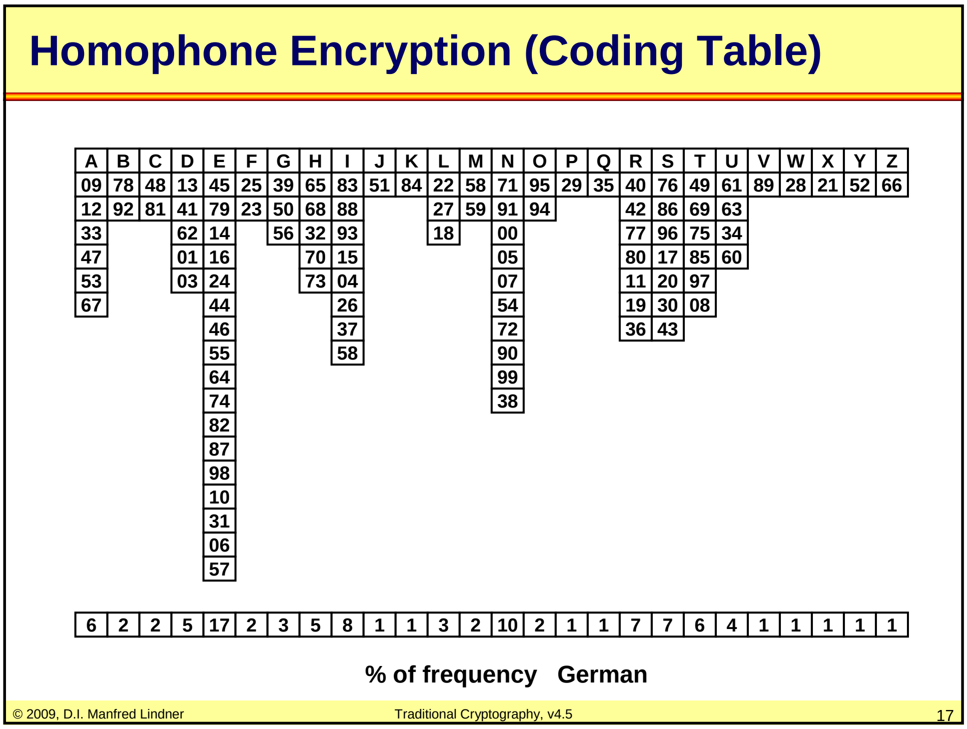
Agenda

- Substitution
- Transposition
- Homophone and Vigenere
- Modern Systems



Homophone

- How to disguise frequency aspects of natural languages?
- Basic problem of all traditional methods
  - frequency of codes corresponds to frequency of language letters or nature of language determines ciphertext code
- One solution trial
  - every letter is replaced by several placeholders (codes)
  - amount of this placeholders for a given letter depends on frequency occurrence of the given letter
  - basically still a mono-alphabetic substitution
  - Homophone

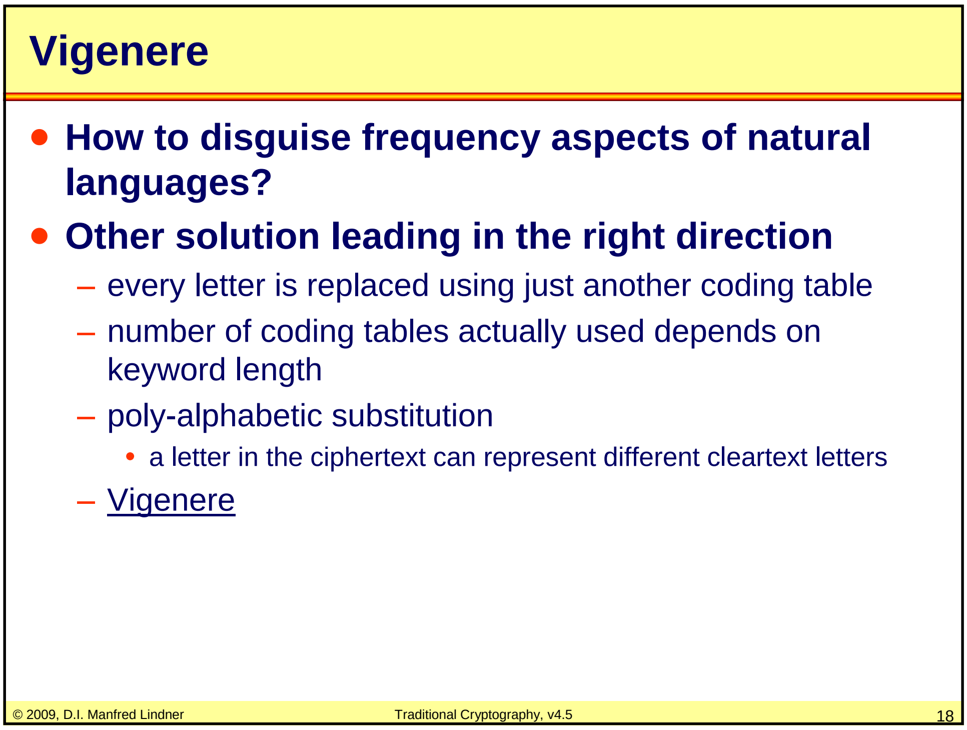


Homophone Encryption (Coding Table)

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
09	78	48	13	45	25	39	65	83	51	84	22	58	71	95	29	35	40	76	49	61	89	28	21	52	66
12	92	81	41	79	23	50	68	88			27	59	91	94			42	86	69	63					
33			62	14		56	32	93			18		00				77	96	75	34					
47			01	16			70	15					05				80	17	85	60					
53			03	24			73	04					07				11	20	97						
67				44				26					54				19	30	08						
				46				37					72				36	43							
				55				58					90												
				64									99												
				74									38												
				82																					
				87																					
				98																					
				10																					
				31																					
				06																					
				57																					

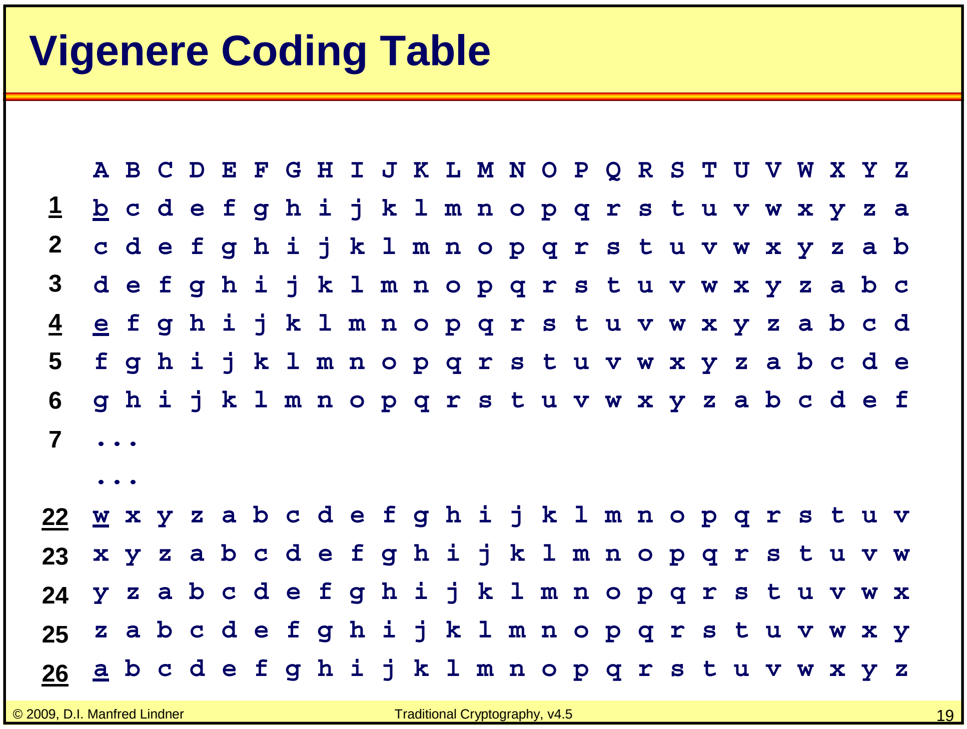
6	2	2	5	17	2	3	5	8	1	1	3	2	10	2	1	1	7	7	6	4	1	1	1	1	1

% of frequency German



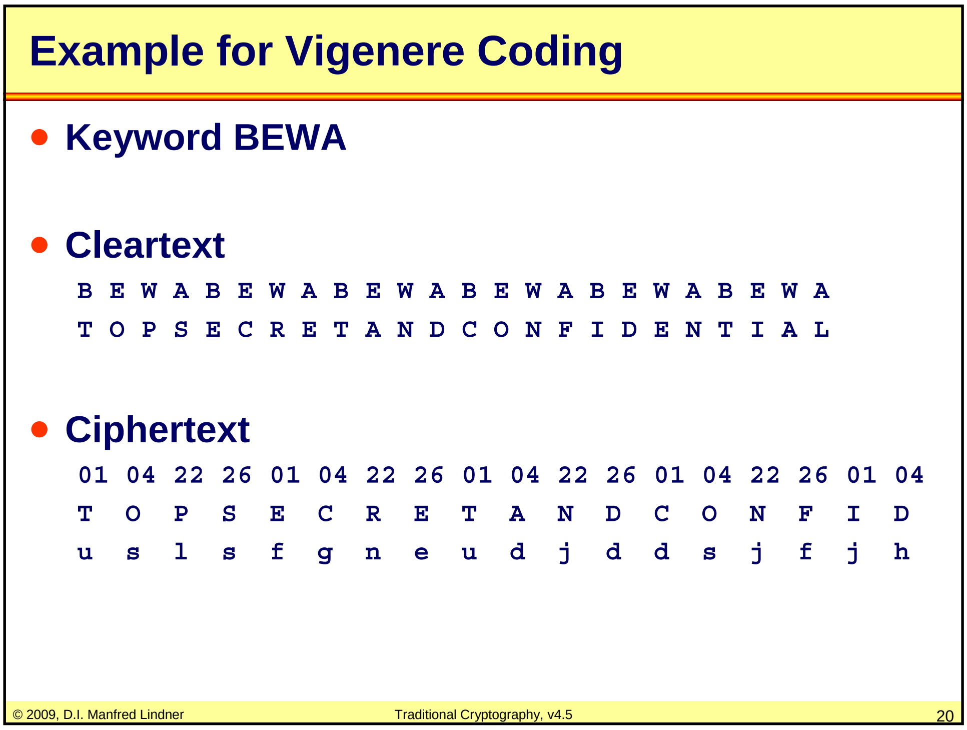
Vigenere

- **How to disguise frequency aspects of natural languages?**
- **Other solution leading in the right direction**
  - every letter is replaced using just another coding table
  - number of coding tables actually used depends on keyword length
  - poly-alphabetic substitution
    - a letter in the ciphertext can represent different cleartext letters
  - Vigenere



Vigenere Coding Table

	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
1	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
2	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
3	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
4	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
5	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
6	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
7	...																									
	...																									
22	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
23	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
24	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
25	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
26	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



Example for Vigenere Coding

- Keyword BEWA

- Cleartext

  BEWABEWABEWABEWABEWABEWA

  TOPSECRETANDCONFIDENTIAL

- Ciphertext

01	04	22	26	01	04	22	26	01	04	22	26	01	04	22	26	01	04
T	O	P	S	E	C	R	E	T	A	N	D	C	O	N	F	I	D
u	s	l	s	f	g	n	e	u	d	j	d	d	s	j	f	j	h

Breaking Vigenere Coding (Babbage, Kasiski)

- Look for repetitions in the ciphertext
  - Could result with high probability because the same cleartext part is encoded in relation to the same position of the keyword
- Cleartext

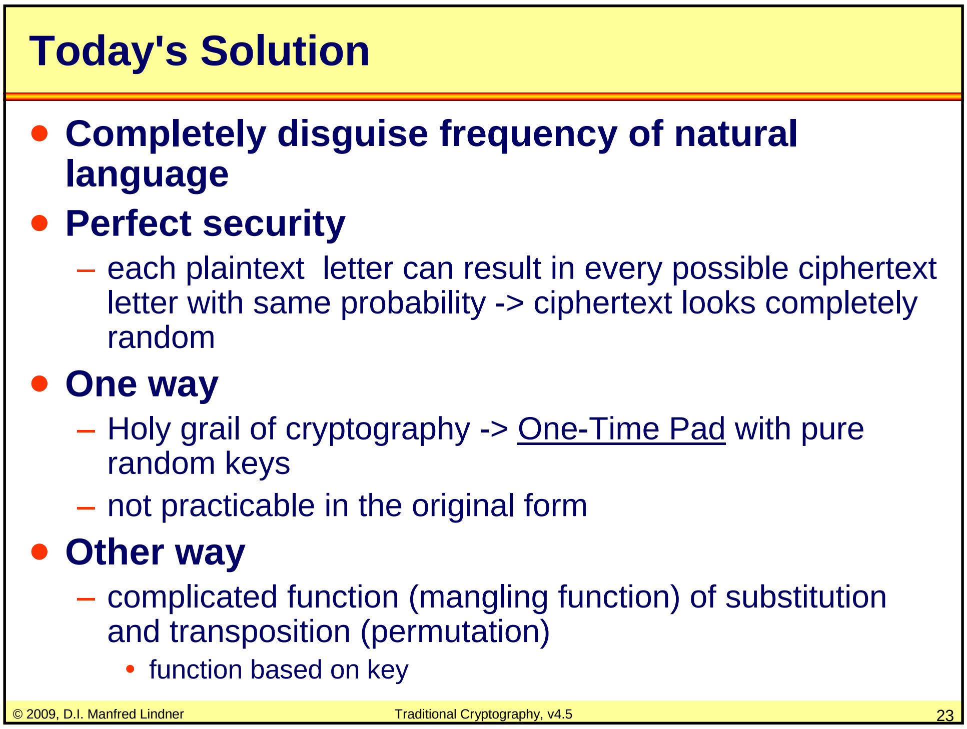
```
B E W A B E W A B E W A B E W A B E W A B E W A
T O P S E C R E T A N D T O P G E H E I M
```

  - space between repetitions is 12
  - keyword length must divide 12 without rest
- Ciphertext

```
01  04  22  26  01  04  22  26  01  04  22  26  01  04  22  26  01  04
T   O   P   S   E   C   R   E   T   A   N   D   T   O   P   G   E   H
u   s   l   s   f   g   n   e   u   d   j   d   u   s   l   g   f   l
```

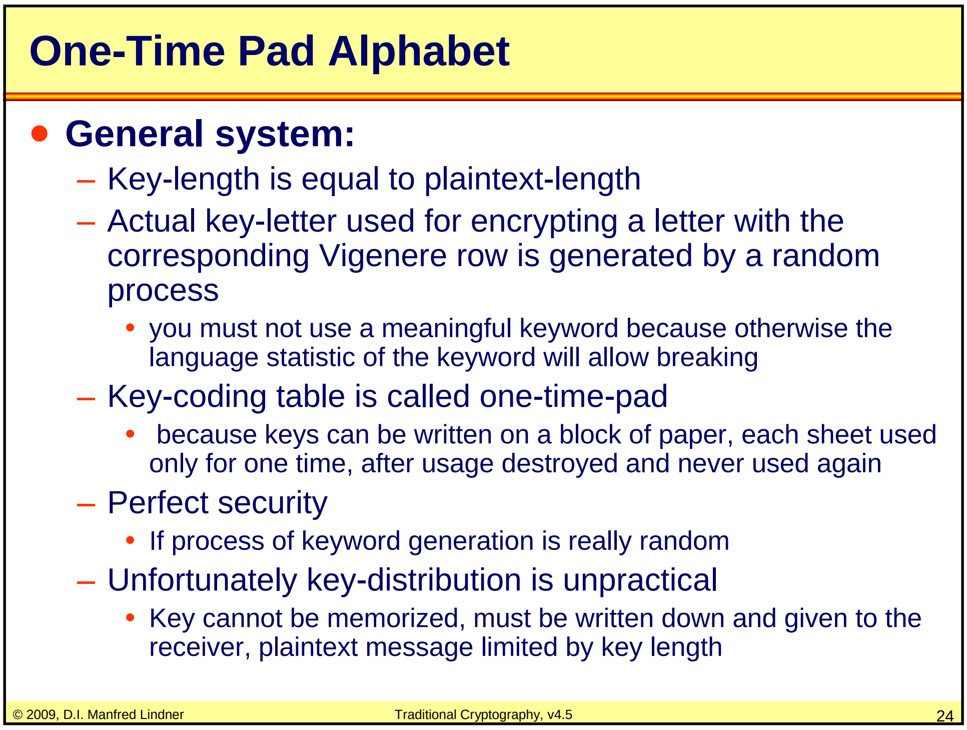
Agenda

- Substitution
- Transposition
- Homophone and Vigenere
- Modern Systems



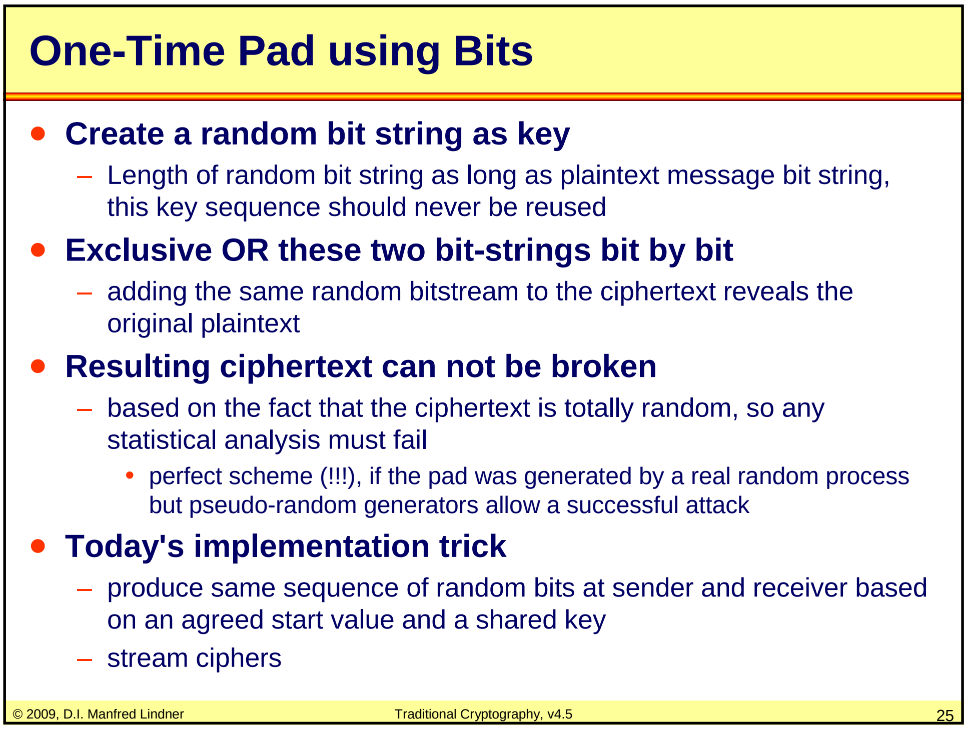
Today's Solution

- Completely disguise frequency of natural language
- Perfect security
  - each plaintext letter can result in every possible ciphertext letter with same probability -> ciphertext looks completely random
- One way
  - Holy grail of cryptography -> One-Time Pad with pure random keys
  - not practicable in the original form
- Other way
  - complicated function (mangling function) of substitution and transposition (permutation)
    - function based on key



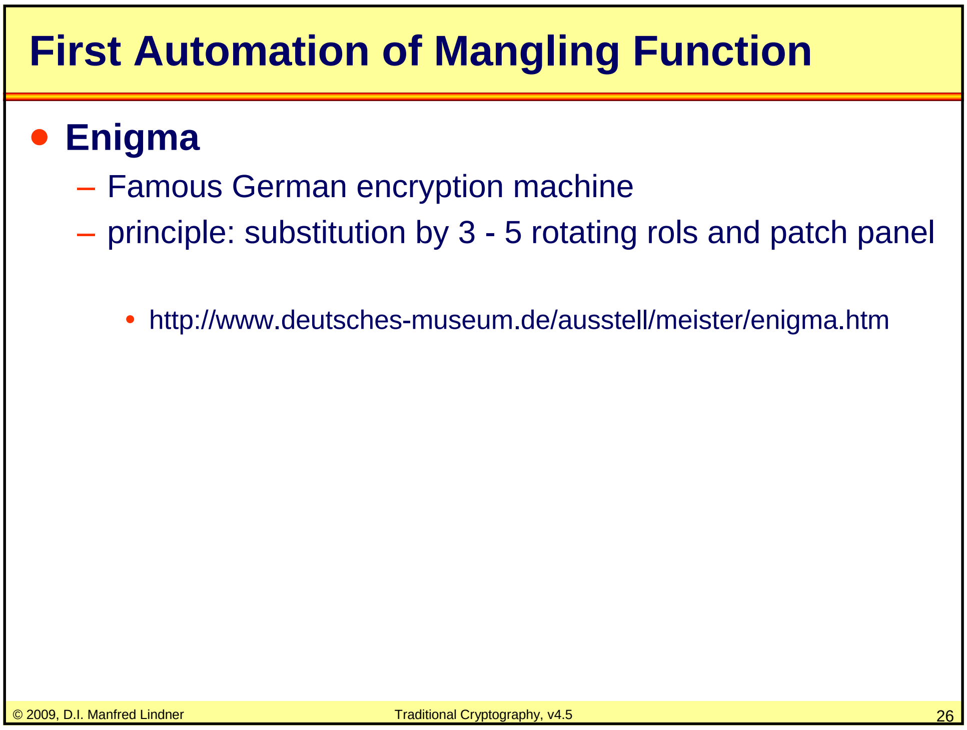
One-Time Pad Alphabet

- General system:
  - Key-length is equal to plaintext-length
  - Actual key-letter used for encrypting a letter with the corresponding Vigenere row is generated by a random process
    - you must not use a meaningful keyword because otherwise the language statistic of the keyword will allow breaking
  - Key-coding table is called one-time-pad
    - because keys can be written on a block of paper, each sheet used only for one time, after usage destroyed and never used again
  - Perfect security
    - If process of keyword generation is really random
  - Unfortunately key-distribution is unpractical
    - Key cannot be memorized, must be written down and given to the receiver, plaintext message limited by key length



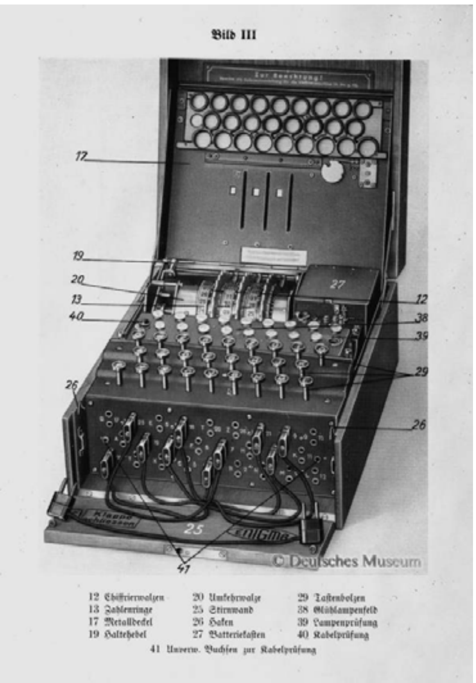
One-Time Pad using Bits

- **Create a random bit string as key**
  - Length of random bit string as long as plaintext message bit string, this key sequence should never be reused
- **Exclusive OR these two bit-strings bit by bit**
  - adding the same random bitstream to the ciphertext reveals the original plaintext
- **Resulting ciphertext can not be broken**
  - based on the fact that the ciphertext is totally random, so any statistical analysis must fail
    - perfect scheme (!!!), if the pad was generated by a real random process but pseudo-random generators allow a successful attack
- **Today's implementation trick**
  - produce same sequence of random bits at sender and receiver based on an agreed start value and a shared key
  - stream ciphers

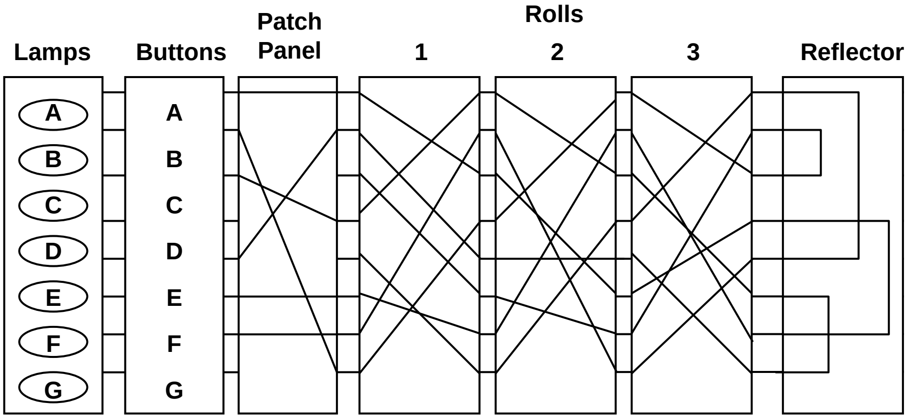


First Automation of Mangling Function

- **Enigma**
  - Famous German encryption machine
  - principle: substitution by 3 - 5 rotating rols and patch panel
    - http://www.deutsches-museum.de/ausstell/meister/enigma.htm



Principle of Enigma



Substitution

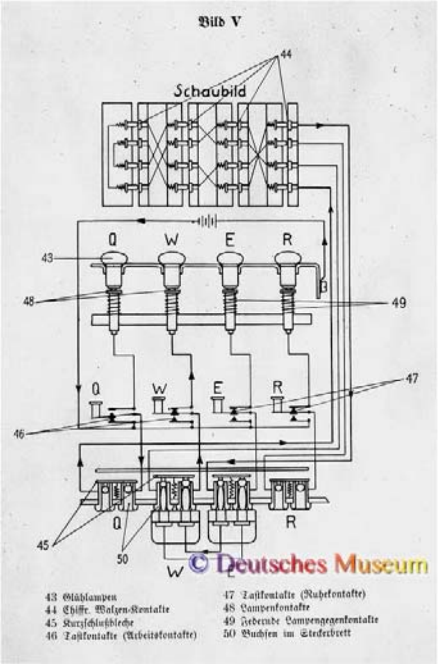
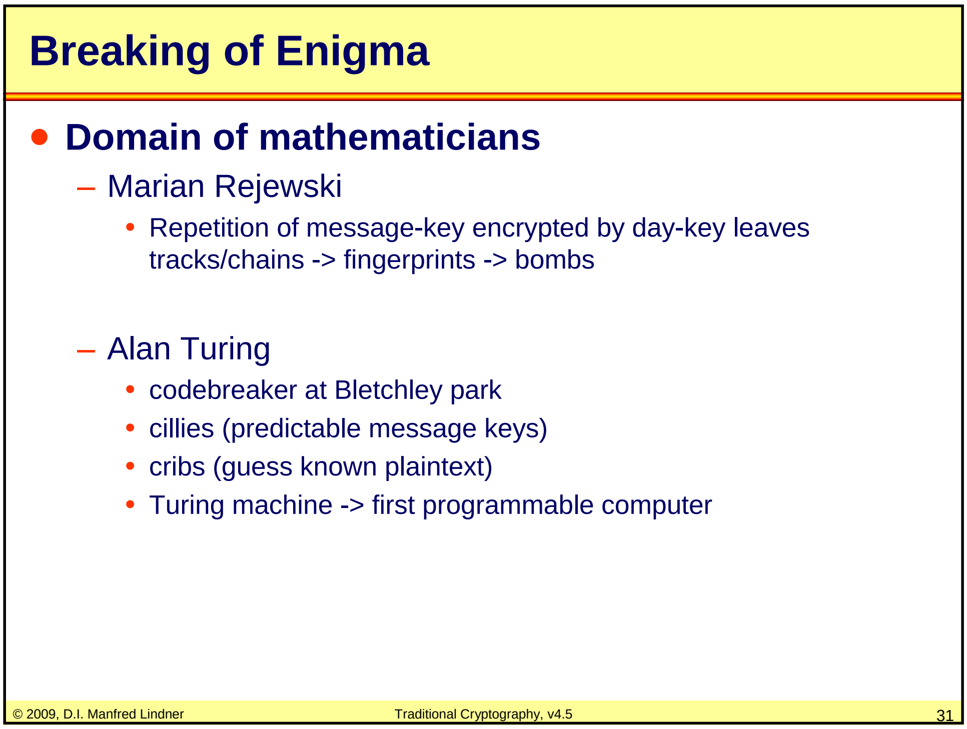


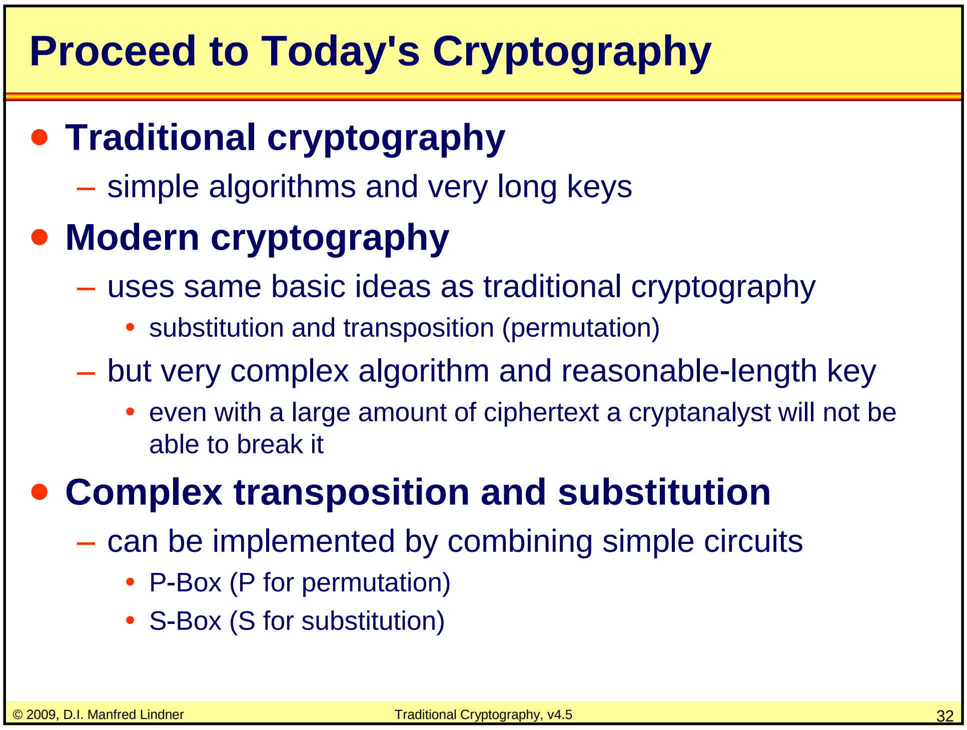
Bild V
Schaubild
44
Q
W
E
R
43
48
49
47
46
45
50
© Deutsches Museum
43 Glühlampen
44 Chiffr. Walzen-Kontakte
45 Kurzſchlußbleche
46 Taſtkontakte (Arbeitskontakte)
47 Taſtkontakte (Ruhekontakte)
48 Lampenkontakte
49 Federnde Lampengegenkontakte
50 Buchſen im Steckerbrett

Zur Beachtung!
ENIGMA
© Deutsches Museum



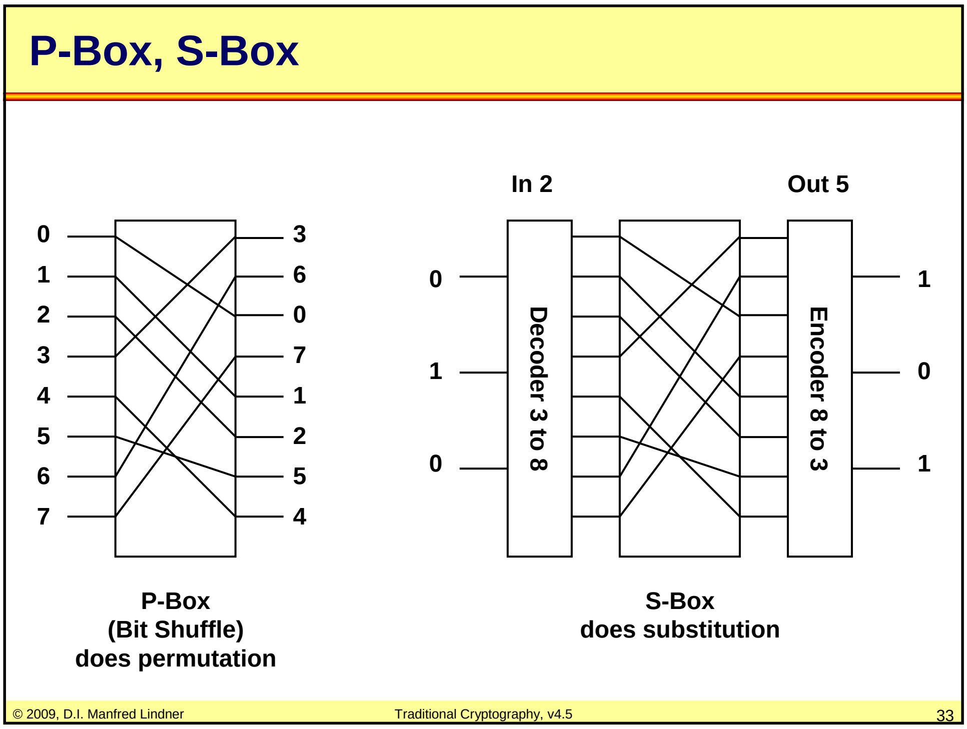
Breaking of Enigma

- **Domain of mathematicians**
  - Marian Rejewski
    - Repetition of message-key encrypted by day-key leaves tracks/chains -> fingerprints -> bombs
  - Alan Turing
    - codebreaker at Bletchley park
    - cillies (predictable message keys)
    - cribs (guess known plaintext)
    - Turing machine -> first programmable computer



Proceed to Today's Cryptography

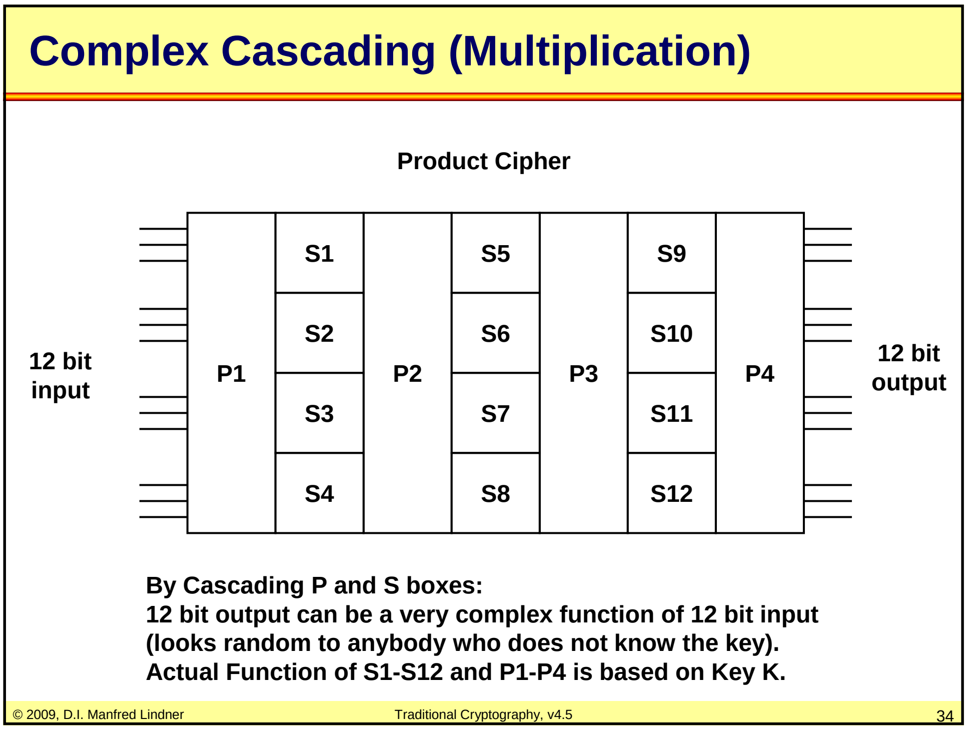
- **Traditional cryptography**
  - simple algorithms and very long keys
- **Modern cryptography**
  - uses same basic ideas as traditional cryptography
    - substitution and transposition (permutation)
  - but very complex algorithm and reasonable-length key
    - even with a large amount of ciphertext a cryptanalyst will not be able to break it
- **Complex transposition and substitution**
  - can be implemented by combining simple circuits
    - P-Box (P for permutation)
    - S-Box (S for substitution)



P-Box, S-Box

P-Box
(Bit Shuffle)
does permutation

S-Box
does substitution



Complex Cascading (Multiplication)

Product Cipher

By Cascading P and S boxes:
12 bit output can be a very complex function of 12 bit input (looks random to anybody who does not know the key).
Actual Function of S1-S12 and P1-P4 is based on Key K.