# Classic Cryptographic Systems

Basel Halak



## Compression & Encryption

- Data is not Information
- Information is not Wisdom

- Compression and Encryption are about manipulating information.

# What is the difference?

- *Compression*:

- Extract the information from the data and encode as efficiently as possible with a public algorithm.

- *Encryption*:

- Diffuse a key into the information as much as possible & encode with a public algorithm

- (most systems diffuse the key into the data. DES does this and the DES cracks rely on this to break into the code)



# The best approach

The best way to practically transmit secure data safely is to:

1. Compress the data
2. Encrypt
3. Add error detection & recovery

Cryptanalysis relies (ultimately) on exploiting redundancies in the plaintext (data).

- Compression removes these.

Encryption is usually slow

- compressing first (which is fast) means that you can pipeline the information flow.
- Data is contracted by the compressor, so the volume of data into the encryption system is less.

## Key Terminology

| Term | Description |
|---|---|
| Plaintext | Message in "clear" form |
| Steganography | Conceals the existence of the message.<br>The message itself may or may not be encoded. |
| Cryptography | Message in plain view, but the meaning is concealed. |
| Cipher | Ciphertext=f(plaintext,key)<br>Operates on groups of characters |



## Learning Outcomes

- At the end of this unit, you should be able to:

1. Describe the principles of classic ciphers
2. Perform cryptanalysis on classic ciphers
3. Describe the principles of Steganography
4. Describe basic terminology of modern cryptosystems

# Learning Outcomes

- At the end of this unit, you should be able to:

1. Describe the principles of classic ciphers
2. Perform cryptanalysis on classic ciphers
3. Describe the principles of Steganography
4. Describe basic terminology of modern cryptosystems



# Hieroglyphics

- Almost 4,000 years ago, a scribe in the town of Menet Khufu drew out the story of the life of his master in hieroglyphics.
- At the same time, though, he also brought into the world the first documented use of Cryptography.

The scribe used a simple code of hieroglyphic substitution, changing one symbol for another, less well-known one. Plaintext is on right, above.

However, this scribe did not use a comprehensive system of encryption; he just substituted hieroglyphs here and there, mostly at the end of his document. Why?

# Caesar Cipher (40 AD)

- It is one of the simplest encryption techniques, it uses monoalphabet substitution technique, in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- It is named after Julius Caesar, who used it with a shift of three to protect messages of military significance, but other substitution ciphers are known to have been used earlier.



# Caesar Cipher

Map a plain alphabet to a single shifted cipher alphabet:

**P abcdefghijklmnopqrstuvwxyz**
**C defghijklmnopqrstuvwxyzabc**

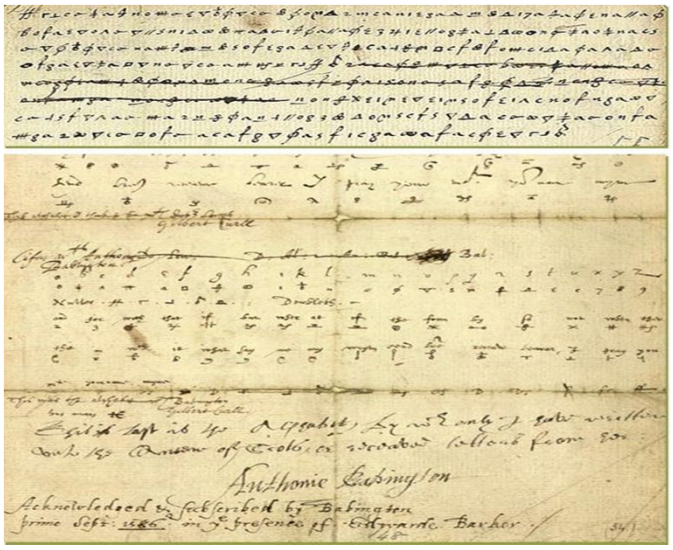
Thus to encipher a plaintext of "veni vidi vici", simply look up the character in the plaintext and pick out the corresponding ciphertext character:

Veni => YHQL
Vidi => YLGL
Vici => YLFL

A variation of this approach is to start the cipher alphabet with a *keyword* e.g. ROME and then complete the remainder of the cipher alphabet as before

The trouble with the monoalphabetic ciphers is that they can be easily cracked using frequency analysis.

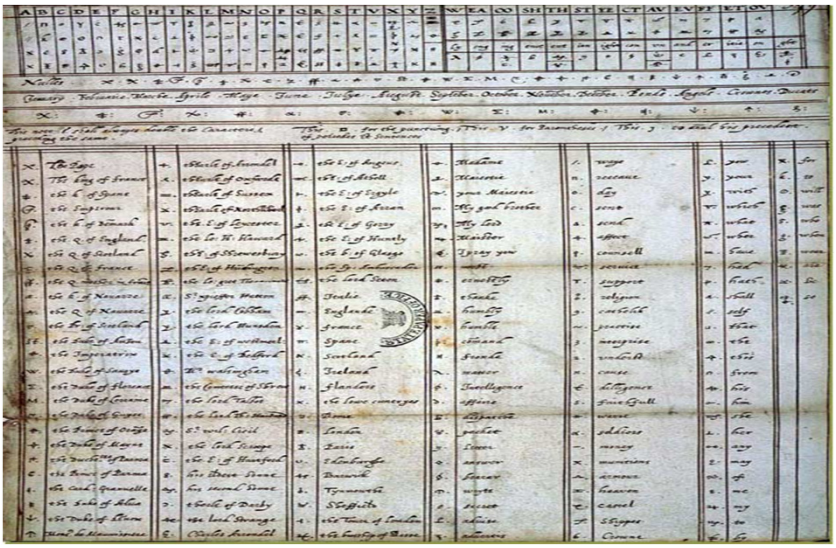
# A deadly letter: 1586





# The Babington Plot

- **It was the event which most directly led to the execution of Mary, Queen of Scots by Queen Elizabeth the First.**
- **Mary had her letters encrypted by her cipher secretary and then smuggled out in casks of ale**. The secret code substituted symbols for letters of the alphabet and also some words. The cipher also included some "nulls", or symbols which represented nothing at all, to confuse anyone trying to decipher the letters.
- **Unfortunately for Mary, the courier was a Catholic double-agent**, Gilbert. The letter was passed to Walsingham's master codebreaker, Sir Thomas Phelippes, a Cambridge-educated language specialist.
- **Phelippes used frequency analysis to work out how to break the code**. He started by looking for the symbols used to denote the most commonly used letters of the alphabet – "e" and "a" – and worked from there.
- **Mary's encryption was not only deciphered, her letter to Babington was also amended;** Phelippes shrewdly added a postscript, asking Babington for the names of others involved in the plot. The conspirators were rounded up and executed.

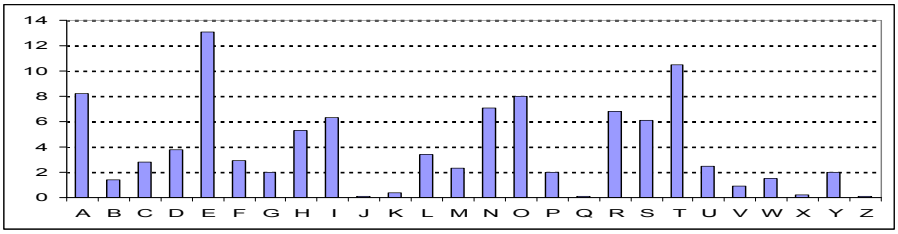


http://www.nationalarchives.gov.uk/spies/ciphers/mary/ma1.htm



# Frequency Analysis

- In English, the frequency of each letter within a piece of standard text is typically:




- This can be used to indicate *possible substituted* letters

# Basic Cryptanalysis

- **Cryptanalysis usually involves a variety of techniques.**

1. For example, **frequency analysis** can indicate possible letters, but this can be enhanced by cribs.
2. **A "crib"** is a known sequence of letters or words – e.g. in English the letter u almost always follows q.
3. We can also **make guesses** about common words based on the context of letters.

- Unfortunately a strength of a short message is its difficulty of applying frequency analysis, Moreover, cipher maker usually employ some additional tricks on the text for example: break up the text into words that do not match the original word lengths – or even a stream of characters



# Basic Cryptanalysis

- **Take the following ciphertext:**

**JUWWTDRU MDH ZPIU FDWIUT GZKF FHQFGKGHGKDR SKAZUC PRT GZU JDCT GZPG MDH RUUTVDC GZU PRFJUC KF PWAZPQUG**

# Basic Cryptanalysis

- If we carry out a basic frequency analysis the most common letter in the message is U – could this be E (the most common letter in the alphabet)?

JeWW TDRe MDH ZPIe FDWIeT GZKF FHQFGKGHGKDR SKAZUC PRT GZe JDCT GZPG MDH ReeT VDC GZe PRFJeC KF PWAZPQeG



# Basic Cryptanalysis

JeWW TDRe MDH ZPIe FDWIeT GZKF FHQFGKGHGKDR SKAZUC PRT GZe JDCT GZPG MDH ReeT VDC GZe PRFJeC KF PWAZPQeG

- The second most frequent letter is G, could this be t?

JeWW TDRe MDH ZPIe FDWIeT tZKF FHQFtKtHtKDR SKAZUC PRT tZe JDCT tZPt MDH ReeT VDC tZe PRFJeC KF PWAZPQet

- Note the 3 letter sequence tZe – could Z be h? – it happens more than once.

JeWW TDRe MDH hPIe FDWIeT thKF FHQFtKtHtKDR SKAhUC PRT the JDCT thPt MDH ReeT VDC the PRFJeC KF PWAhPQet

# Basic Cryptanalysis

JeWW TDRe MDH hPIe FDWIeT thKF FHQFtKtHtKDR SKAhUC PRT the JDCT thPt MDH ReeT VDC the PRFJeC KF PWAhPQet

- Some words can be easily inferred: *thPt* is clearly *that*, so P is a.

JeWW TDRe MDH haIe FDWIeT thKF FHQFtKtHtKDR SKAhUC aRT the JDCT that MDH ReeT VDC the aRFJeC KF aWAhaQet

- We can return to our frequency analysis and note that the letters *e t h a* have been allocated. The next most commonly used letter is D. Does this correspond to the 4th most common letter in common English usage o?



# Basic Cryptanalysis

JeWW ToRe MoH haIe FoWIeT thKF FHQFtKtHtKoR SKAhUC aRT the JoCT that MoH ReeT VoC the aRFJeC KF aWAhaQet

- We can also notice the letter sequence JeWW. Pairs of letters are unusual, and we have a clue letter that is follows e. Possible letters could be eLL, eMM, eNN, eOO, eTT, eSS.
- We can make a guess that LL looks more likely: W->L

Jell ToRe MoH haIe FolIeT thKF FHQFtKtHtKoR SKAhUC aRT the JoCT that MoH ReeT VoC the aRFJeC KF alAhaQet

# Basic Cryptanalysis

- We now can observe another interesting word: alAhaQet

Is this "alphabet" ? A -> p and Q -> b

Jell ToRe MoH haIe FolIeT thKF FHbFtKtHtKoR SKphUC aRT the JoCT that MoH ReeT VoC the aRFJeC KF alphabet

- The next most common letters are N, R and I.
- Take the triplet aRF could R-> r – ar? – but the possible letters after *ar* are *e* and *t* – both taken. What about I – ai? – limited scope for words – N seems better. Try it and see…

Jell Tone MoH haIe FolIeT thKF FHbFtKtHtKon SKphUC anT the JoCT that MoH neeT VoC the anFJeC KF alphabet



# Basic Cryptanalysis

Jell Tone MoH haIe FolIeT thKF FHbFtKtHtKon SKphUC *anT* the JoCT that MoH neeT VoC the anFJeC KF alphabet

- Now we see an interesting triplet again *anT*. Could the T-> d ? Try it and see…

Jell done MoH haIe FolIed thKF FHbFtKtHtKon SKphUC and the JoCd that MoH need VoC the anFJeC KF alphabet

- There are three letters left, C, F & K, that occur frequently, and the main common letters not used thus far are S, R, & I. Is there a match?
- Take the text thKF – K could be a vowel – therefore K is i?

## Basic Cryptanalysis

**Jell done MoH haIe FolIed *thiF* FHbFtitHtion SiphUC and the JoCd that MoH need VoC the anFJeC iF alphabet**

- **Now we have the word thiF – F could be S or R – but S makes a proper word, thus F->s**

**Jell done MoH haIe solIed this sHbstitHtion SiphUC and the JoCd that MoH need VoC the ansJeC is alphabet**

- **By this stage, you could probably solve the rest intuitively, but we can still replace another letter straight away – C -> r, by elimination**

**Jell done MoH haIe solIed this sHbstitHtion SiphUr and the Jord that MoH need Vor the ansJer is alphabet**



## Basic Cryptanalysis

**Jell done MoH haIe solIed this sHbstitHtion SiphUr and the Jord that MoH need Vor the ansJer is alphabet**

- **Another simple word to identify (big words are useful as they have fewer common denominators) is sHbstitHtion**

**Obviously H -> u**

**Jell done Mou haIe solIed this substitution SiphUr and the Jord that Mou need Vor the ansJer is alphabet**

# Basic Cryptanalysis

- **And we can complete the message with a bit more intuition and using common words:**

***well done you have solved this substitution cipher and the word that you need for the answer is alphabet***



# Some Cryptanalysis Tips

1. **The vowels A E I O are normally high frequency, U is moderate and Y is low frequency.**

   Try to analyze a cipher in terms of vowels and consonants alone before committing values to letters.
2. **Letters contacting low frequency letters are usually vowels.**
3. **Letters showing a wide variety of contact are usually vowels.**
4. **In repeated digrams (pairs of letters), one letter is usually a vowel.**
5. **In reversed digrams - eg VX, XV, one letter is normally a vowel.**
6. **Doubled consonants are usually bordered by vowels and vice versa.**
7. **It is unusual to find more than five consonants in a row.**
8. **Vowels do not often contact one another.**

# Polyalphabetic Ciphers

System: Alternate between the cipher alphabets C1 and C2

```
P  abcdefghijklmnopqrstuvwxyz
C1 fzbvkixaymeplsdhjorgnqcutw
C2 goxbfwthqilapzjdesvycrkuhn
```

| Plaintext | C1 | C2 | Ciphertext |
|---|---|---|---|
| H | a | | a |
| E | | f | f |
| L | p | | p |
| L | | a | a |
| O | d | | d |

It is interesting to note that the same letter *L* in the plaintext does not appear as the same letter in the ciphertext



# Vigenère Cipher (1585 )

- It is a 26 alphabet form of the polyalphabetic substitution cipher.
- It is more secure than Cesare cipher against frequency analysis.
- It is attributed to Blaise de Vigenère, a French diplomat and cryptographer.

# Vigenère Cipher

- The idea is to have 26 cipher alphabets all shifted by a different amount from the plaintext alphabet, this is managed using a Vigenère square:

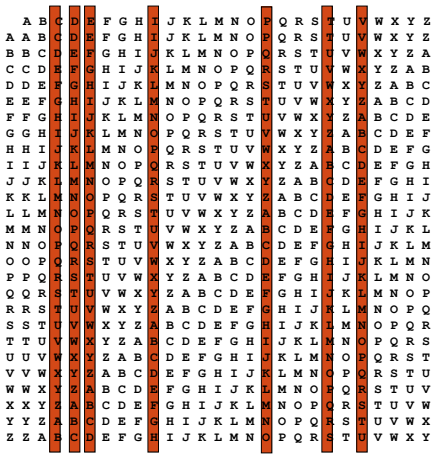
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T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



# Vigenère Cipher

1. Consider the set of all 26 Caesar ciphers:

   $\{ C_a, C_b, C_c, \ldots, C_z \}$
2. Choose a Key: e.g. *deceptive*
3. Encrypt the plaintext letter by letter using $C_d$, $C_e$, $C_c$, $C_e$, $C_p$, $C_t$, $C_i$, $C_v$ , $C_e$ in turn.
4. Repeat from start after the last $C_e$
5. Decryption simply works in reverse.



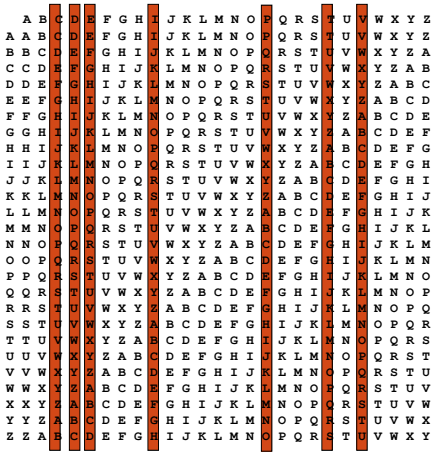
# Vigenère Cipher

**Example:**

**key:**
deceptivedeceptivedeceptive

**plaintext:**
wearediscoveredsaveyourself

**ciphertext:**
ZICVTWQNGRZGVTWAVZHCQYGLMGJ





# Vigenere Cipher: Cryptanalysis

- **Basic approach**:

1. Find the length of the key.
2. Divide the message into that many simple substitution encryptions.
3. Use frequency analysis to solve the resulting simple substitutions.

- **How to find the key**:

1. Kasisky test
2. Index of coincidence (Friedman)

# Kasisky Test

- **Concept**: plaintext words separated by multiples of the key length are encoded in the same way.
- **Example:**

Key: **C A T** C A T C A T **C A T** C A T C A **T C A** T C A T

PT: **T H E** S U N A N D **T H E** M A B I N **T H E** M O O N

CT: **V H X** U U G C N W **V H X** O A G K N **M J E** F Q O G

- It is worth noting if the distance between the repeated pattern is not a multiple of the key, this test will reveal nothing



# Kasisky Test

- **Algorithm**:

1. Search for pairs of identical segments of length at least 3
2. Record distances between the two segments: $\Delta 1, \Delta 2, \ldots$ –
3. The key length (m) should divide the gcd( $\Delta 1, \Delta 2, \ldots$)

# Index of Coincidence (Friedman)

- **IC is defined** to be the probability that two randomly selected letters from a text will be identical.

- **Example**

For a truly random text , the chance of pulling out an A is $^1/_{26}$. The probability of pulling out two As simultaneously is $(^1/_{26})*(^1/_{26})$. In general, the chance of drawing any pair of letters is $26*(^1/_{26})*(^1/_{26}) = (^1/_{26}) = 0.0385$

- **The formula for the index of coincidence for any text is**:

$$IC = \frac{\sum_{i=a}^{z} f_i(f_i - 1)}{N(N-1)} \approx \frac{\sum_{i=a}^{z} {f_i}^2}{N^2} = \sum_{i=a}^{z} {P_i}^2$$

**Where**

$N$ is the total number of letters in the text

$f_i$ is the frequency of letter $i$ in the text (i.e the number of occurrences )

$P_i$ is the probability of letter $i$ in the text



# Frequency Analysis

- If a ciphertext is a mono-alphabetic substitution then it will have similar frequencies (albeit not the same letters) as the standard alphabet.
- There is a measure called the 'Index of Coincidence' which can be calculated to see whether our cryptogram frequencies are similar to normal English.

The formula for the index of coincidence is:

$$IC = \frac{\sum_{i=A}^{Z} f_i (f_i - 1)}{N (N - 1)}$$

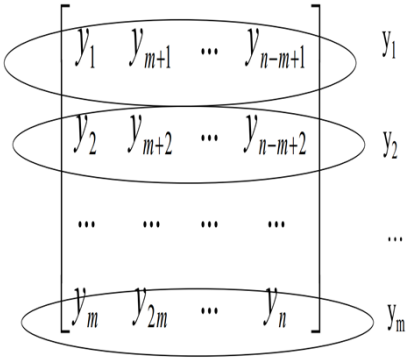
- Normal written English has an IC of around 0.066.
- Entirely random text has an IC near 1/26 = 0.038

# Index of Coincidence (Friedman)

- IC is unique for each language, for an English text, IC is around 0.065
- You can use this metric to find the key length of a Vigenere cipher, as follows:

Assume you have a text $Y=y_1y_2y_3..y_n$

1. Make a guess of the key length (e.g. m)
2. Arrange the text into a matrix of m rows



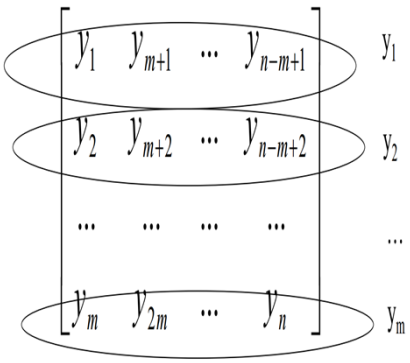



# Index of Coincidence (Friedman)

- IC is unique for each language, for an English text, IC is around 0.065
- You can use this metric to find the key length of a Vigenere cipher, as follows

Assume you have a text $Y=y_1y_2y_3..y_n$

1. Make a guess of the key length (e.g. m)
2. Arrange the text into a matrix
3. Calculate IC for each row ($Y_1$-$Y_m$)



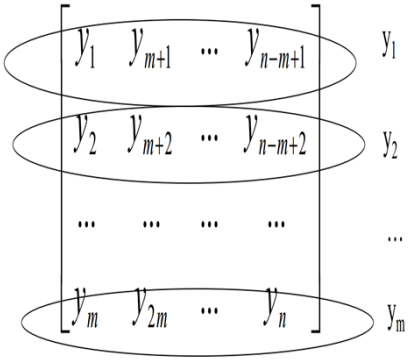
# Index of Coincidence (Friedman)

- IC is unique for each language, for an English text, IC is around 0.065
- You can use this metric to find the key length of a Vigenere cipher, as follows

Assume you have a text $Y=y_1y_2y_3..y_n$

1. Make a guess of the key length (e.g. m)
2. Arrange the text into a matrix
3. Calculate IC for each row ($Y_1$-$Y_m$)
4. If the width of the matrix is a multiple of the actual key length than:

$$IC\ (Y_i) \approx 0.065 \ \forall\ 1 \le i \le m$$






# Index of Coincidence (Friedman)

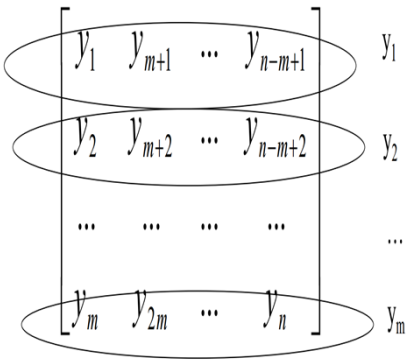
- IC is unique for each language, for an English text, IC is around 0.065
- You can use this metric to find the key length of a Vigenere cipher, as follows

Assume you have a text $Y=y_1y_2y_3..y_n$

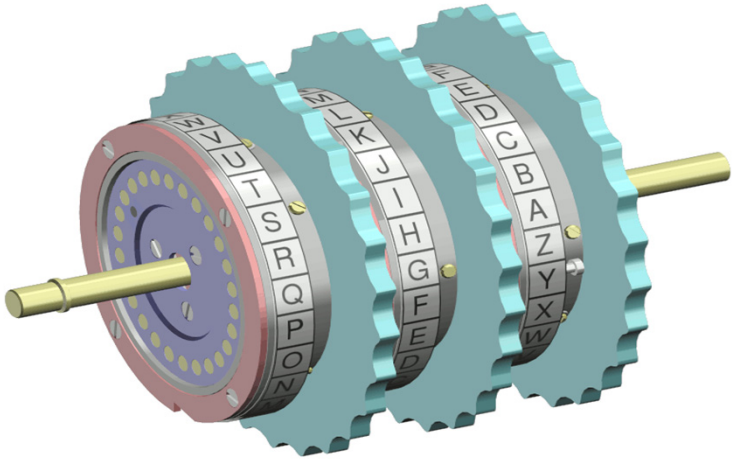
1. Make a guess of the key length (e.g. m)
2. Arrange the text into a matrix with m rows
3. Calculate IC for each row ($Y_1$-$Y_m$)
4. If the width of the matrix is a multiple of the actual key length than:

$$IC\ (Y_i) \approx 0.065 \ \forall\ 1 \le i \le m$$

5. Repeat the process again until you find the correct key



# Rotor Cipher Machines

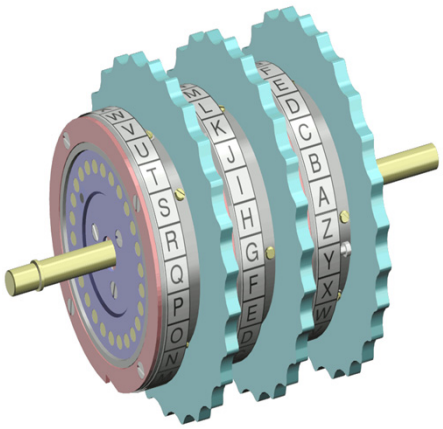


41

# Rotor Cipher Machines

- **Operation Principles**

These are rotating disks with an array of electrical contacts on either side. the rotors advance positions after encrypting each letter, which changes the substitution, this produces a complex polyalphabetic substitution cipher which changes for every letter.

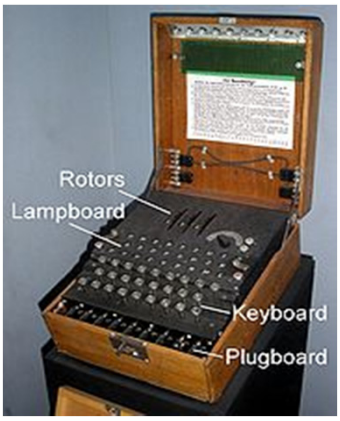


- Widely used in WW2.
- With 3 cylinders, $K = 26^3 = 17,576$.
- With 5 cylinders, $K = 26^5 = 12 \times 10^6$.

42

# The Enigma Machine

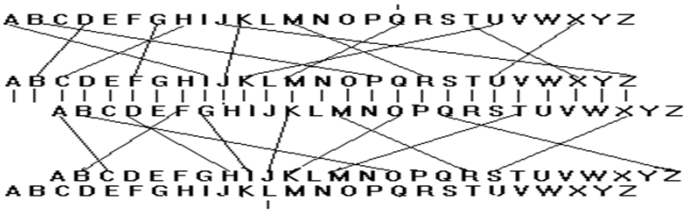
- Before and during WW II almost all German communications were enciphered on the Enigma cipher machine.
- It was based **on rotors whose movement produced ever-changing alphabetic substitutions.**





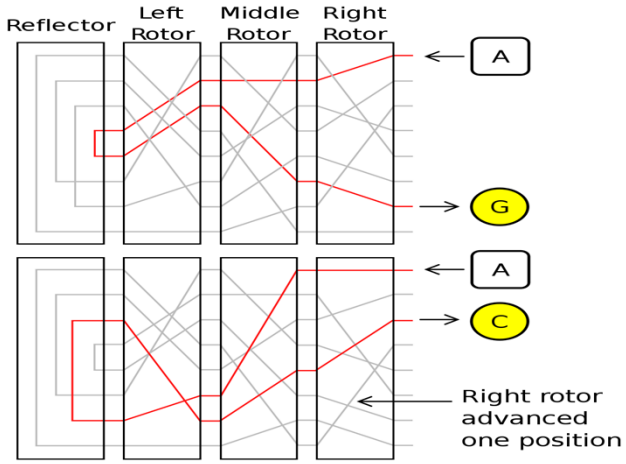

# The Enigma Machine

- The Enigma machine worked by using a series of substitution ciphers where the connections were made electrically.
- The idea behind the rotors was to shift each substitution alphabet in addition to making the connections




- **In this example, if you applied a voltage to the letter Q, then the lamp L would light up.**

# The Enigma Machine

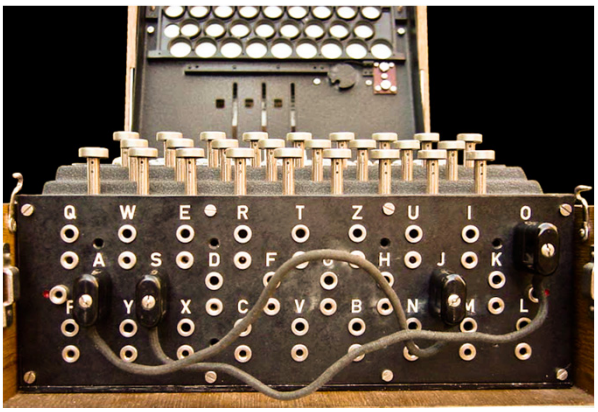



Encryption for two consecutive letters with the right rotor moving one position between them.



# The Enigma Machine

- In its military use, the basic machine **was greatly enhanced by a plugboard**, visible on the front of the machine.



# The complexity of the Enigma Machine

- **The number of setting of the Enigma mashie is calculated as**:

**Possible choice of routers * Possible starting positions* number of plugbord settings**

- **Example:**

Assume an enigma machine with 6 rotors, 10 plugboard connections

Rotors = 6 x 5 x 4 =120

Initial position: 26 x 26 x 26 =17576

Plugboard settings: $\frac{26!}{6! \times 10! \times 2^{10}} = 150{,}738{,}274{,}937{,}250$

So the total number $3.179 * 10^{21}$



# The complexity of the Enigma Machine

- **More abstractly**: **the number of ways of choosing m pairs out of n objects is:**

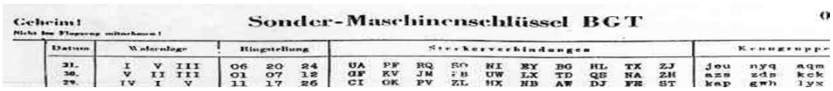
Plugboard settings: $\frac{n!}{(n-2m)! \times m! \times 2^m}$

- **For example**: There are 3 different ways of putting 2 pairs of wire into 4 plugboard sockets
- **From this formula we can find** that possible plugboard pairings is greatest for 11 pairs, and then decreases:

1 pair: 325
2 pairs: 44.850
3 pairs: 3,453,450
4 pairs: 164,038,875
5 pairs: 5,019,589,575
6 pairs: 100,391,791,500
7 pairs: 1,305,093,289,500
8 pairs: 10,767.019,638,375
9 pairs: 58,835.098,191,875
10 pairs: 150,738,274,937,250
11 pairs: 205,552,193,096,250
12 pairs: 102,776,096,548,125
13 pairs: 7,905,853,580,625

# The Enigma Machine

- The message key, (the complete and exact configuration of the machine in its starting position), had to be conveyed to the intended recipient of the message.
- This was done using a 24 hour settings sheet as shown here. (*Geheim* means secret!)



Geheim! Sonder-Maschinenschlüssel BGT

| Datum | Walzenlage | Ringstellung | Steckerverbindungen | Kenngruppe |
|---|---|---|---|---|
| 31. | I V III | 06 20 24 | UA PF RQ SO NI EY BG HL TX ZJ | jou nyq aqn |
| 30. | V II III | 01 07 12 | QF KV JM IB UW LX TD QS NA ZH | azs zdn kck |
| 29. | IV I V | 11 17 26 | CI OK PV ZL HX NB AW DJ FR ST | kap gwh lyx |

- From these settings the rotors could be configured as well as the plugboard connections.



# Breaking Enigma

## Breaking Enigma

- Due to treachery by Hans-Thilo Schmidt, the French, and then the Poles, had access to the Enigma instruction manual, from which the method of operation and the wheel wirings could be deduced.
- The key problem was the interaction between the (big) collection of plugboard settings and the (manageable) collection of wheel settings.
- **One of the key flow of the enigma machine which allows it to be broken is that a letter is never encrypted as its self**



## Breaking Enigma

- The bombe was designed by Alun Turing , a mathematician who worked for the British secret service
- With a set of tricks, weak keys, knowledge of some information from code books "pinched" and short weather messages with standard information, the basic settings could be plugged into a mechanical system , called a bombe, that ran combinations of settings to decode the messages.

## Transposition Ciphers

- Transposition or permutation ciphers form the second basic building block of ciphers.
- The core idea is to rearrange the order of basic units (letters/bytes/bits) without altering their actual values.
- Examples: Rail Fence Cipher, Columnar Transposition Cipher,

## An Example from History

# An Example from History

- **One of the earliest transposition encryption** devices was the the Scytale
- **Ancient Spartans and Greeks** are told to have used this cipher to communicate during military campaigns.
- **The Scytale consists of a ribbon wrapped around a dowel of a particular diameter and length.**
- **Encryption**: The secret message was written on the ribbon while the ribbon was wrapped on the rod. The ribbon was then removed and transported to the other field commander who had an identical rod. If the ribbon was intercepted it look like jumble of letters.
- **Decryption**: The recipient uses a rod of the same diameter on which he wraps the parchment to read the message.


# Rail Fence cipher

- **Encryption** : the message letters out diagonally over a number of rows (rails)then read off cipher row by row. The key for the rail fence cipher is just the number of rails
- **Example**:

defend the east wall of the castle

**1. the message letters out diagonally over 3 rails**

d . . . n . . . e . . . . .t . . . . .l . . . . .h . . . s . . .

. e . e . d . h . e . s . .w . . l . .o . t . e . a . t . .e

. . f . . . t . . . a . . . .. a . . . . .f . . . c . . . .l .

**2.The ciphertext is read off along the rows:**

dnetlhseedheswloteateftaafcl

# Columnar Transposition Cipher

- **Encryption**: the message is written out in rows of a fixed length and then read out again column by column. Both the width of the rows and the permutation of the columns are usually defined by a keyword.

- **Example**: Encrypt the message M below using the keyword ***Destiny***

  M= {Attack postponed until two}

57

# Columnar Transposition Cipher

1. The message is written out in rows of a fixed length 7 indicated by the number of letters in the key word "**Destiny**". **The empty spaces at the end are usually filled with random letters**

| Alphabetical Order | 1 | 2 | 5 | 6 | 3 | 4 | 7 |
|---|---|---|---|---|---|---|---|
| **K** | D | E | S | T | I | N | Y |
| **PT** | a | t | t | a | c | k | p |
| | o | s | t | p | o | n | e |
| | d | u | n | t | i | l | t |
| | w | o | ***f*** | ***g*** | ***x*** | ***n*** | ***u*** |

2. The columns are read according to the alphabetical order of the letters in the keyword. In this case, the order would be "1 2 4 5 3 6 7 ".
3. Therefor the cipher text will be

AODW TSUO COIX KNLN TTNF APTG PETU

58

## Columnar Transposition Cipher: Cryptanalysis

- **Example**

Decrypt the following message:

Trsao rih oivo geetn tootn bfsafme cepcogep ihtetcen



## Columnar Transposition Cipher: Cryptanalysis

- The number of letter in this message is 45 (without spaces)
- The key length should be a divisor of 45 (i.e. 3, 5,9 or 15).

# Columnar Transposition Cipher: Cryptanalysis

- We assume the key length is three and we write the ciphertext in a table with 3 columns
- If the key guess is correct than we should find anagrams if we read the text row by row
- We need to look of patterns or combinations that suggest a reordering of columns

| Column 1 | Column 2 | Column 3 |
|---|---|---|
| T | t | e |
| r | n | p |
| s | t | c |
| a | o | o |
| o | o | g |
| r | t | e |
| i | n | p |
| h | b | i |
| o | f | h |
| i | s | t |
| v | a | e |
| o | f | t |
| g | m | e |
| e | e | c |
| e | c | n |



# Columnar Transposition Cipher : Cryptanalysis

- Our first guess does not seem to be correct
- We repeat the process assuming the key length is 5.

| Column 1 | Column 2 | Column 3 | Column 4 | Column 5 |
|---|---|---|---|---|
| t | i | o | m | p |
| r | v | o | e | i |
| s | o | t | c | h |
| a | g | n | e | t |
| o | e | b | p | e |
| r | e | f | c | t |
| i | t | s | o | c |
| h | n | a | g | e |
| o | t | f | e | n |

# Columnar Transposition Cipher : Cryptanalysis

- Our first guess does not seem to be correct.
- We repeat the process assuming the key length is 5.
- The last row seems to be an anagram for the word often. We re-order the columns accordingly.

| Column 1 | Column 2 | Column 3 | Column 4 | Column 5 |
|---|---|---|---|---|
| t | i | o | m | p |
| r | v | o | e | i |
| s | o | t | c | h |
| a | g | n | e | t |
| o | e | b | p | e |
| r | e | f | c | t |
| i | t | s | o | c |
| h | n | a | g | e |
| o | t | f | e | n |



# Columnar Transposition Cipher : Cryptanalysis

- If we read the table along the row , we can obtain the following text:

*To improve is to change to be perfect is to change often*

| Column 1 | Column 3 | Column 2 | Column 4 | Column 5 |
|---|---|---|---|---|
| t | o | i | m | p |
| r | o | v | e | i |
| s | t | o | c | h |
| a | n | g | e | t |
| o | b | e | p | e |
| r | f | e | c | t |
| i | s | t | o | c |
| h | a | n | g | e |
| o | f | t | e | n |

# Fitness Metrics

- **A Fitness Metric is a measure to determine how similar a piece of text is to English text.**
- **Examples:**

1. Single letters frequencies (not very relevant to the Hill Climbing method, Why?)
2. Digrams statistics: the probability of a sequence of two letters
3. Trigrams statistics: the probability of a sequence of three letters
4. Quadgrams statistics: the probability of a sequence of four letters

- A piece of text very similar to English will get a high fitness score while a text of random characters will get a low score .
- A ciphertext or an incorrectly deciphered (i.e. using the wrong key) message will probably contain sequences e.g. 'QGKZ' which are very rare in normal English.



# Columnar Transposition Cipher Cryptanalysis Methods

- **Brute Force attacks assuming Short Key words**

Try all possible short words (up to 9 letters) and look for anagrams, patterns , possible combinations for words…

- **Dictionary attacks**

1. Compile a list a of dictionary words including place names, famous people, mythological names, historical names and so on(1,000,000 dictionary words would be a good comprehensive target.
2. Generate a text file of possible keys.
3. Try to decrypt the ciphertext with all possible dictionary words and record the keyword that resulted in plaintext with the highest fitness.

## Columnar Transposition Cipher Cryptanalysis Methods

- **Hill Climbing Method**

1. Assume the key length is N =10, then choose a random starting keyword of this length. This is called the parent key.
2. Apply the parent key and measure the fitness of the resulting text
3. Generate a child key by making random swaps in the parent keyword.
4. Apply the child key measure the fitness of the resulting text, if the latter is higher than fitness obtained in step 2, than replace the parent with the child that beat it.
5. Rank different decryption keys until you find the one that produces deciphered text with the fewest rare sequences.
6. If the decryption key is not found after several hundred times, choose another random word of length N and go back to step 2
7. If the key is still not found increment N by 1 and repeat the whole process .

- Keys of length 20 are very difficult to crack, and it gets much more difficult to crack keys as they become longer than 20.

67

## Product Ciphers

- Ciphers using only substitutions or transpositions are not secure because of language characteristics
- The fore number of ciphers in succession to make harder:
  - two substitutions make a more complex substitution
  - two transpositions make more complex transposition
  - but a substitution followed by a transposition makes a new much harder cipher
- Product Ciphers is bridge from classical to modern ciphers

# Learning Outcomes

- At the end of this unit, you should be able to:

1. Describe the principles of classic ciphers
2. Perform cryptanalysis on classic ciphers
3. Describe the principles of Steganography
4. Describe basic terminology of modern cryptosystems



# What is Steganography?

- Origin of the term: ***steganos:*** **secret** – ***graphy:*** **writing**

## Steganography & Cryptography

- **The difference is in their goals…**
  - Cryptography: although encypted and unreadable, the existence of data is not hidden
  - Steganography: no knowledge of the existence of the data

## First Stenographic Methods

- An ingenious method was to shave the head of a messenger and **tattoo a message or image on the messengers head.** After allowing his hair to grow, the message would be undetected until the head was shaved again.
- **Invisible inks.** Ancient Chinese wrote messages on fine silk, which was then crunched into a tiny ball and covered in wax. The messenger then swallowed the ball of wax. This method was used as recently as WWII.
- **Microdots.** It is based on photographically shrinking a page of text into a dot less than one millimeter in diameter, and then hide this microdot in an apparently innocuous letter. This methods have been used since early 20th century.
- **Null ciphers** (unencrypted messages) were also used, where the real message is "camouflaged" in an innocent sounding message.

## Steganosystems Principles

- **A Steganosystem consist of**

1. A cover-object is an original unaltered medium (text, image or video).
2. Embedding process in which the sender hides a message by embedding it into a cover-text, usually using a key, to obtain a stego-object.
3. Stego-object
4. Recovering process in which the receiver extracts the hidden message from the stegotext using the key.

- **Security requirement** stego-objects should be indistinguishable from cover-objects.



## Textual Steganography

- **It consists of hiding messages in formatted texts**
- **Examples:**

1. Every n-th character: A message is concealed into certain letters of the text, for example into the first letters of some words. An improvement of the previous method is to distribute plaintext letters randomly in the cover-text and then use a mask to read it.
2. Using errors or stylistic features at predetermined points in the cover data such as word shifting encoding by altering the amount of whitespace

# An Null Cipher Example

- The following message was actually sent by a German Spy in WWII [1]:

*"Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils."*

- Taking the second letter in each word the following message emerges: *Pershing sails from NY June 1.*

[1] David Kahn, *The Codebreakers*, The Macmillan Company. New York, NY 1967.

# Visual Steganography

Most "real" images or audio files can lose ~3% of information before degradation becomes noticeable

In this example, would you notice the effect of noise in the picture?

0% Noise

3% Random Noise

# Visual Steganography

- Visual Steganography can be used to **hide messages in images**
- **For example**, take a 24-bit full colour image = 3x8-bit RGB or CMY . Take the LSB of each plane and use for a secret message



# Visual Steganography

- GIF uses an 8-bit pointer to a palette of 256 colours.
- The colour distribution and palette are optimised for the specific image
- Subtract LSB for your message and the palette is reduced to 128 colours.
- While this is OK for some types of image, the human eye is particularly sensitive to certain types of image and colours – e.g. skin colours.
- JPEG splits the scene into 8x8 blocks of pixels and applies a cosine transform.
- It is inherently noisy, but you can hide data within the cosine frequencies and amplitudes

## Free Steganography Tools

- **Steganography Studio**

http://steganography_studio.en.softonic.com/

- **OpenStegno**

http://www.openstego.info/



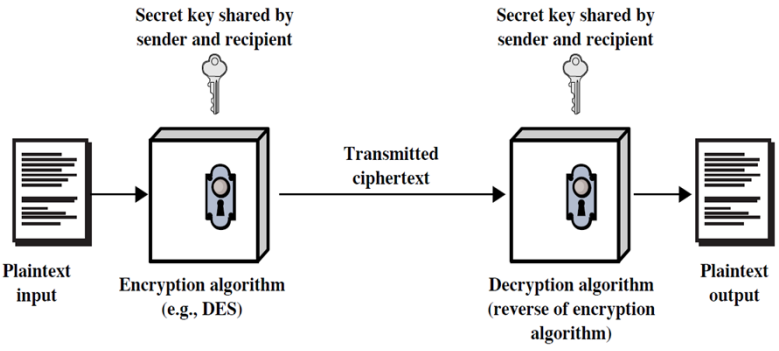
## Learning Outcomes

- At the end of this unit, you should be able to:

1. Describe the principles of classic ciphers
2. Perform cryptanalysis on classic ciphers
3. Describe the principles of Steganography
4. Describe basic terminology of modern cryptosystems

# Private Key Cryptography

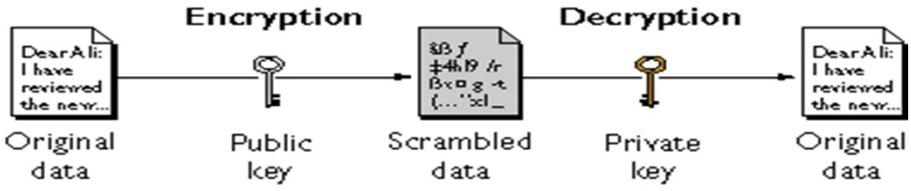
- Symmetric algorithms have a single key for reading and writing
  - Like a safe – you have the key, you open the door, you can read, write, delete, copy etc.






# Public Key Cryptography

- Asymmetric algorithms have 2 keys: one for writing and one for reading.
  - E.g. a letter box – easy to put things in, hard to get them out.
  - These rely on trap door one way functions

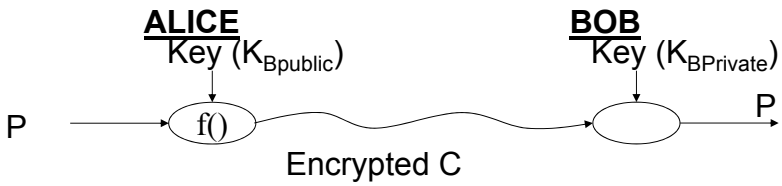


# Public Key Cryptography

- **One way functions:**
  - Given x, evaluation of f(x) is trivial
  - Given y, evaluation of $f^{-1}(y)$ computationally difficult
  - e.g. *discrete log*
- **Trap Door One Way Functions**:
  - Construct a pair of functions f, $g=f^{-1}$
  - Given x, evaluation of f(x) is trivial
  - Given only f and y, evaluation of $f^{-1}(y)$ computationally difficult
  - But, given g & y, evaluation of $f^{-1}(y) = g(y)$ is trivial
  - For example: *RSA*



# Public Key Cryptography




- $K_{Bpublic}$ is freely available to anyone & f()
- $K_{Bprivate}$ is never sent over a communications channel
- Attack of C knowing P (known plaintext attack) will only ever reveal $K_{Bpublic}$ which is freely available anyway.
- $K_{Bprivate}$ is (we hope) computationally difficult to generate from $K_{Bpublic}$

# Hybrid Cryptosystems

**BUT: asymmetric (i.e. public key) algorithms are generally $10^3$ slower than symmetric algorithms**

**Asymmetric systems are vulnerable to exhaustive enumeration of plaintexts.**

- i.e. keep generating pairs of plaintexts and ciphertexts until the ciphertexts match – when this happens, the plaintext has been found
- Salt the plaintexts to avoid this.

**Hybrid approach for speed:**

- A generates a random session key, R, encrypts with B's public key and sends to B
- A decrypts with private key
- A & B have a secure dialogue using R & a symmetric algorithm and then discard R
- Be very careful how you generate R