

Introduction

- The difference between public and private cryptography: the key exchange protocol and implicitly the information manipulation. The two domains are not disjoint but rather complementary.

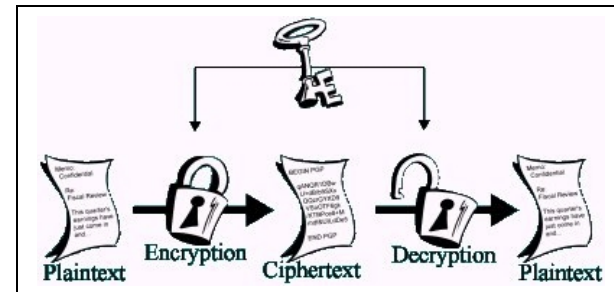
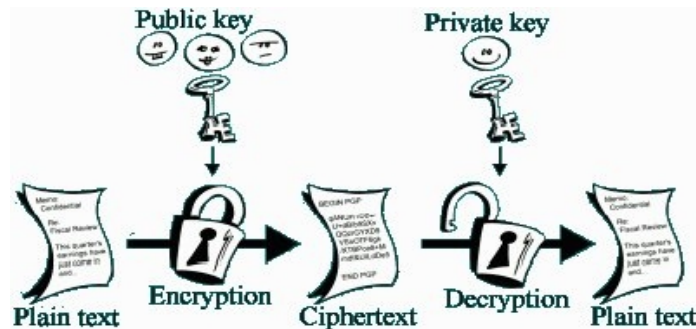


Fig.1: Public (key) cryptography: (asymmetric) the key used to perform the encryption is made public so that any person can decipher the message using a private key. **Fig.2: Private (key) Cryptography** (symmetric): a key is "symmetrically" used to encrypt and respectively decrypt a message.

- The concept of a cryptographic protocol: the procedure through which data and keys are exchanged..
 - **Private cryptography** is the strict mathematic method through which data is encrypted or decrypted.
 - **Public cryptography** refers largely to the procedure through which the keys are exchanged (some being able to be stored in a public repository).

Alph

- **Cryptography from the Greek language: "kryptos" hidden and "logos" word. The purpose being to transform data into an intermediary form in which the initial content is present but hidden (and theoretically impossible to recover).**
- **Two domains in cryptography: **Criptography** as the art of writing secrets and **cryptographic analysis** through which one attempts to recover the message from the intermediary form. Alph and its role in the two sub-domains:**
 - **Implementation of **antique cyphers and codes** as well as **new algorithms**:**
CAESAR, ATBASH, ZIGZAG, RAILFENCE, VIGENERE MDELVAYO, BEALE, VERNAM, NIHILIST, PLAYFAIR, BIFID, USASS, PURPLE, ENIGMA, NAVAJO, TWOSQUARE, LEWIS, ALBERTI, ADFGX, MORSE, ROT-13 and FEISTEL.
 - **Hashes:** One time cypher or generation of a unique identifier: *UNIXELF, WEINBERGER, SDBM, DJB2, XOR si MD5.*
 - **Methods of **cryptographic analysis**:** *Frequency analysis and the "Hill-Climbing" method.*
 - **Various **cryptographic utilities**:** permutations and combinations of data, the use of a "stecker" (implemented for the PURPLE and ENIGMA cyphers) and filter or file operation mode.

Implementation of Categories

- Difference between *codes* and *cyphers*: Codes substitute words, phrases or whole expression whereas a cypher implies a transformation at unit level. Alph presently contains two codes: *NAVAJO* and *MORSE*.
- Difference between *substitution* si *transposition* in cryptography: Substitution replaces parts of data through a symbol following a precise algorithm and transposition re-arranges plain text in a different manner. Alph contains several "clean" transposition cyphers: *ZIGZAG*, *RAILFENCE* and *BIFID*.
- Definition of a *hash*: transformation of a block of data in an intermediary form so that the original block can't be re-created. A supplementary condition is that each time the data block is processed, it will wield the same result. Hashes in Alph: *UNIXELF*, *WEINBERGER*, *SDBM*, *DJB2*, *XOR*, *MD5*.
- The other cyphers are rather hybrids from all categories using advanced substitution and transposition techniques and even hashes (ie. *FEISTEL* and the linear forward shift).
- Classification of the cyphers in historical orders such as *classical*, *modern cyphers* and *hashes*.

First Cyphers

- **Cesar**: First set of cyphers found in ancient Rome. A simple 3-letter alphabetic shift:

$$\begin{aligned}f(x) &= x+3 \\ C(x) &= f(x) \\ D(x) &= f^{-1}(x)\end{aligned}$$

- The power of the cypher is based rather on the inability of third parties to read or write at all. The cypher was used to dispatch orders from Rome to armies in various Roman provinces.
- **Atbash**: Based on the Hebrew alphabet, a simple substitution replacing each letter with its alphabetic mirror counterpart. For example "*aleph*", first letter in the Hebrew alphabet, is replaced by "*tav*" last letter in the alphabet.

$$f(x) = 27 - x \pmod{26} \quad (x=1,26)$$

$$C(x) = D(x) = f(x) = f^{-1}(x)$$

- The first cypher perfectly satisfying the symmetry condition of certain cyphers. The encryption function is completely reversible. Atbash is used a lot in the interpretation of mystic Hebrew texts and has been associated with the esoteric Jewish methodologies such as the Kabbalah.

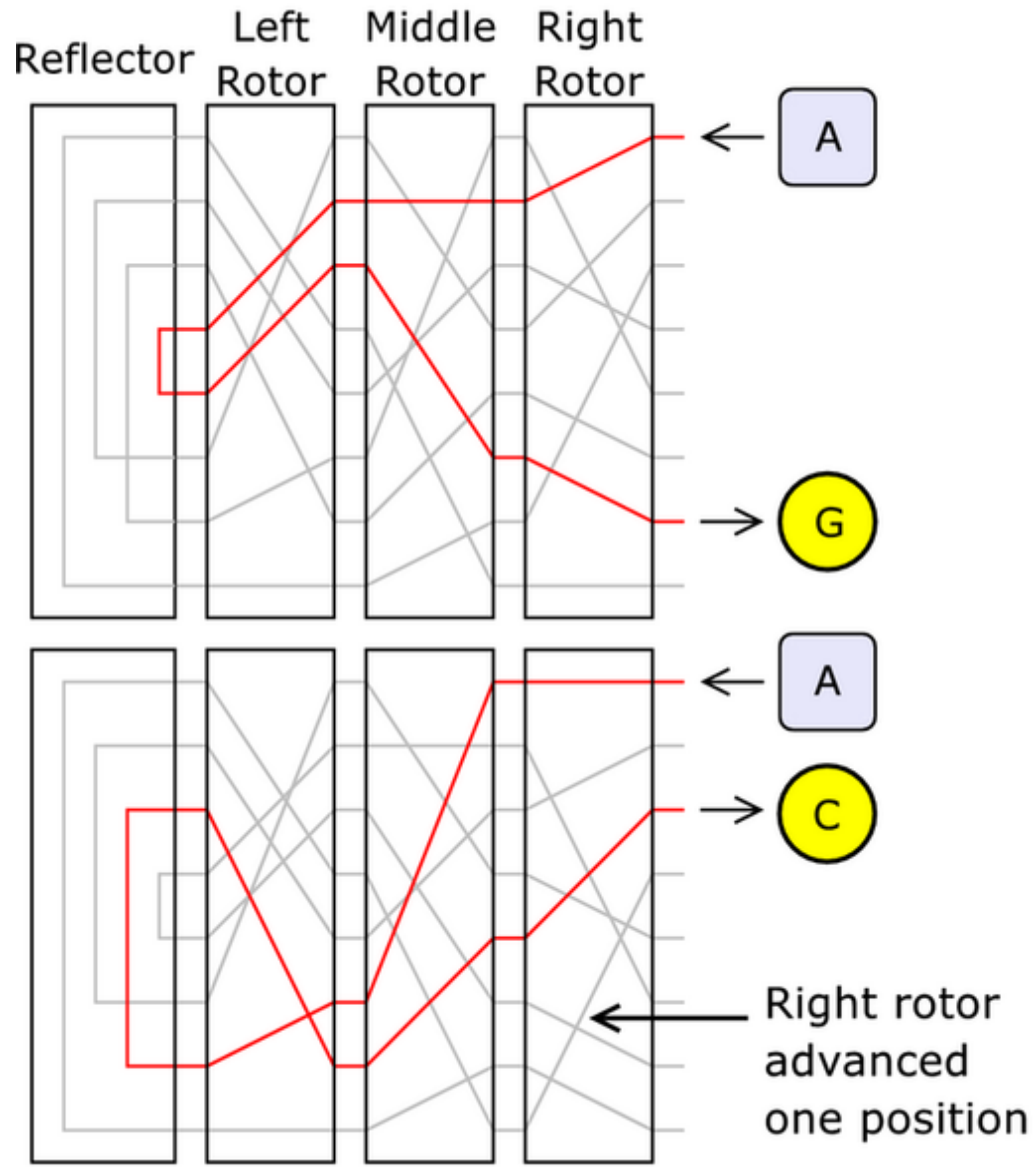
Enigma

- **Probably the most controversial cypher and employed by the German army during the Second World War.**
- **Unique mechanical system implementing a perfect symmetrical function with the property that each symbol from the original plain text is gradually replaced by another symbol even if it repeats.**

$$C(x)=D(x)=PRMLUL^rM^rR^rP^r$$

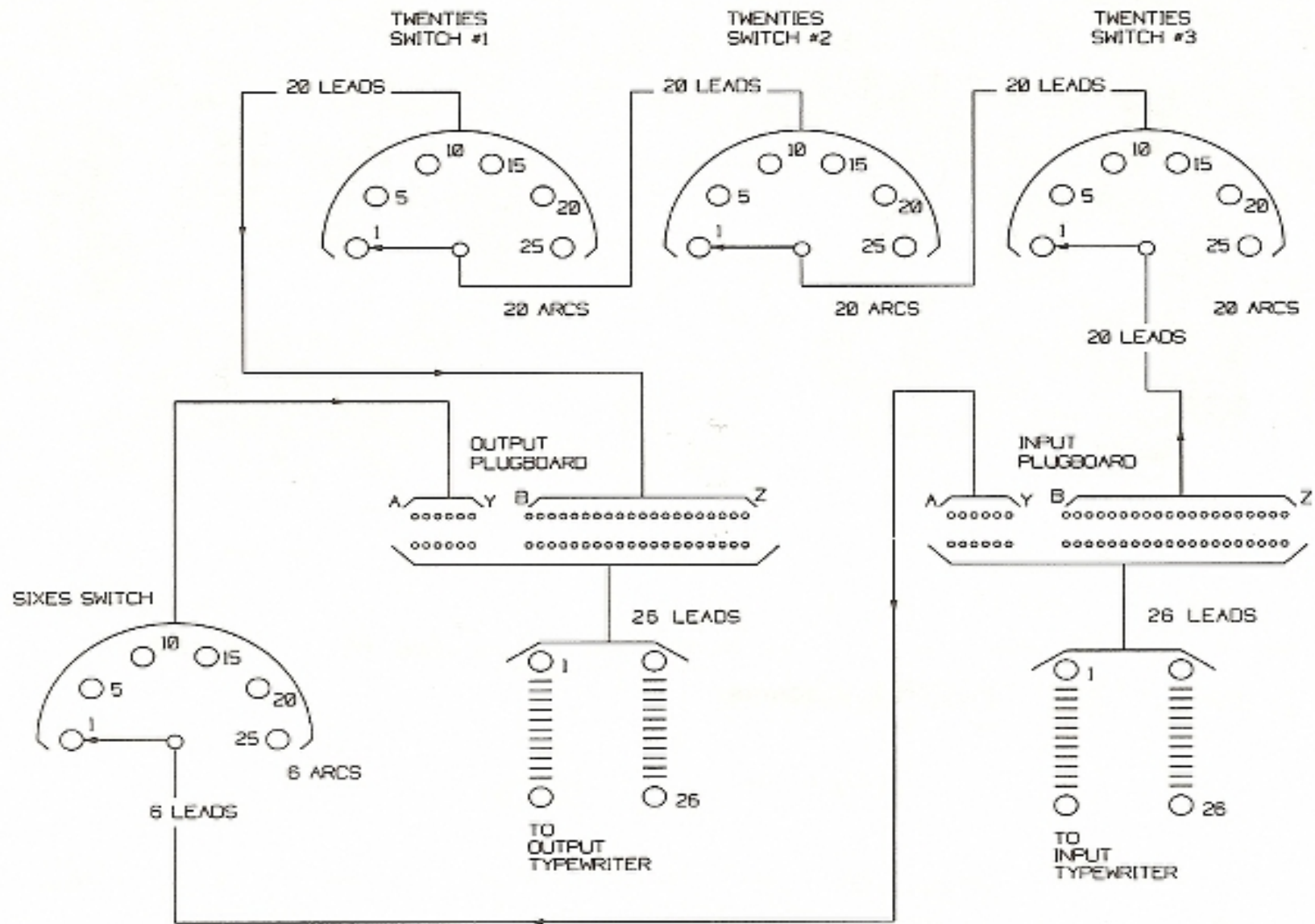
where P is the "stecker", R, M, L right, middle and left rotors and the rest, the corresponding inverted functions.

- **A modular mechanical system allowing the addition or replacement of the alphabetical sets; implemented through a system of rotors and deflectors.**
- **The key is highly variable and depends on: the rotors used, the deflector used, the ring (initial rotor position setting) and a plugboard setting.**



Purple

- **Purple or Angooki - taipu B is a machine similar to ENIGMA used to cypher the Japanese transmissions and was the machine used to write the 14 multi-part message through which Japan declares war to the United States.**
- **A system based on ciphering the vowels and the rest of the letters separately using the "sixes", respectively "twenties" rotors. Major flaw of concept which leads to weak resistance to frequency analysis.**
- **A system which employs a series of fixed rotors which can not be reconfigured and which are hard to practically reproduce.**
- **Uses a complex key algorithm: initial position of the sixes, position of the twenties and motion sequence of the rotors (switches) and a plugboard setting.**
- **Is not symmetric: at decryption one must reverse the twenties rotors and keep the motion sequence intact.**



Enigma vs. Purple

Possible combinations:

ENIGMA:

- Variable number of rotors and one variable deflector. Up to six rotors in some cases but usually three.
- Total combinations (in case of single notched machines): $26^n * 26^2$ where n is the number of rotors used times the plugboard. In case of common machines (three rotors and a deflector) 11,881,376 combinations.

PURPLE:

- Fixed number of switches meaning no possibility for extensions.
- Three twenties switches and a sixes switch: twenties($26^2 * (20*20)^3$) + sixes($26^2 * (6*6)$) = 4,326e10 combinations.

Cryptographic:

- Full confusion in both cases (substitutions).

Enigma:

- SAC (the rotors are the keys).

Purple:

- Partial SAC depending on motion and initial positions (which is present in ENIGMA).

"Unbreakable" Cipher

- **Conceived by an engineer at AT&T Bell Labs in 1917 concerning research of an ideal cypher which should be unbreakable and resistant to all analysis.**
- **Based on the first most important concept of key cryptography: "the resistance of a cypher is given by the strength of the key".**
- **Using the above concept, the new cipher uses a key equal to the length of the message. This key is then combined with the plain text unit by unit.**
- **With a large enough key one can apply an unique transformation to all the units of the plain text. The key entropy is not to be found in every unit but rather in the message itself. If one knows the key in part, one can determine just a part of the message...**
- **Main drawbacks are the data block length required to be transmitted: one needs to send double the message length. On the other hand, one can determine part of the message if one knows part of the key.**

Feistel Networks

- Invented by Horst Feistel in the 70s, Feistel networks become a standard for various cyphers. (Ex: DES).
- A cypher based on Feistel has the following properties:
 - Bit mangling (P-box permutations)
 - Simple non-linear functions (S-box permutations)
 - Linear mangling (XOR functions)
 - SAC (Strict Avalanche Criterion).
 - Any function may be used, even irreversible ones.

- **Algorithm:**

To *encrypt*:

1. The plain text is divided into two equal blocks (L_0, R_0)
2. For each round $i = 1, 2, \dots, n$:

$$\begin{aligned}L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i)\end{aligned}$$

3. The result is (L_n, R_n) .

To *decrypt*:

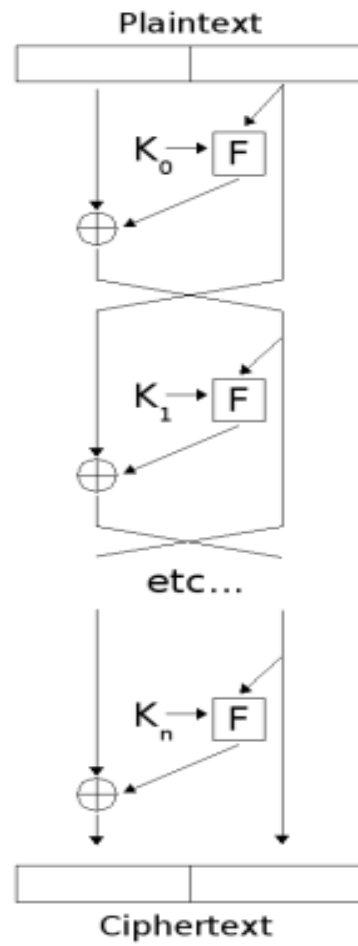
Regardless of the function f :

$$\begin{aligned}R_{i-1} &= L_i \\ L_{i-1} &= R_i \oplus f(L_i, K_i)\end{aligned}$$

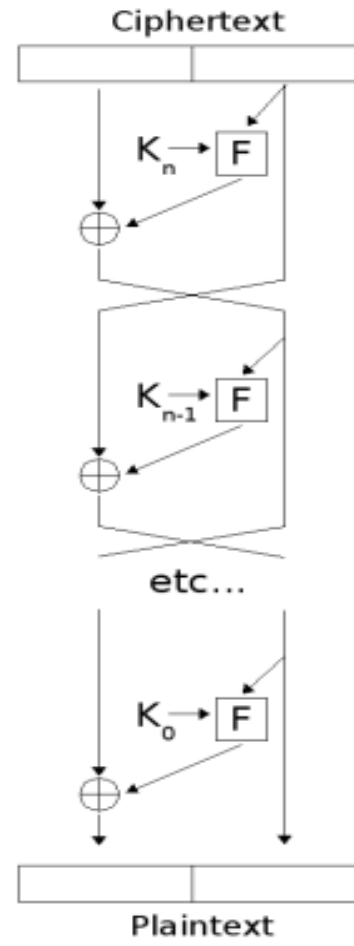
The only difference to encryption is following the elements i of the key K in reverse order.

The result is (L_0, R_0)

Encryption:



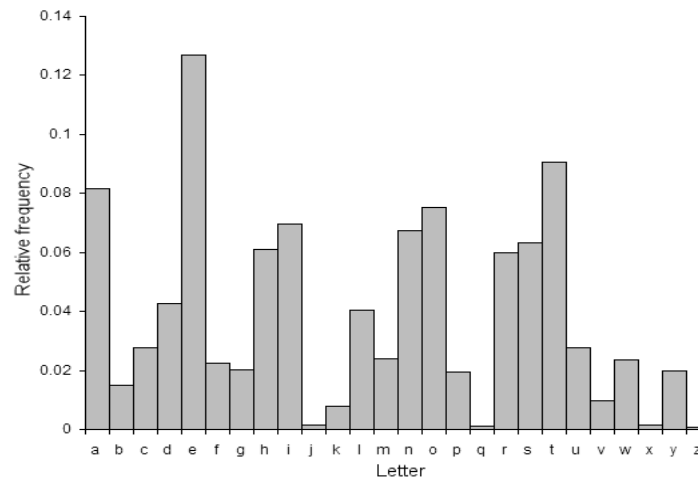
Decryption:



Feistel Cipher

Frequency Analysis

- Around 750a.d. during a manuscript is found signed by Al Kindi in which he proposes a method named "**Frequency analysis**" and is based on the fact that the letters appear with different frequencies in various languages.
- Using a frequency table, and knowing in what language a certain text was written, we can count occurrences of certain bigrams, trigrams and unique letters. The following is a common distribution of unique letters in the English language:



- In certain intermediary forms (ciphertexts) we can distinguish certain patterns or position of certain letters which would yield information about the plaintext and eventually break the cipher.

Index of Coincidence and Frequencies

- Each language uses different letters with a specific frequency depending on the nature of the text. The index of coincidence of a text is the probability that two letters randomly selected are identical.
- **Definition:** Suppose a text containing n_0 A's, n_1 B's ... n_{25} Z's and α the finite alphabet set, then $n=n_0 + n_1 + \dots + n_{25}$ is the numbers of the letters in a given text so that:

$$Ic = \frac{\sum_{i=1}^{\alpha} n_i(n_i - 1)}{n(n - 1)}$$

we can express the above formula using combination theory:

$$Ic = \frac{\sum_{i=1}^{\alpha} C_{n_i}^2}{C_n^2}$$

which actually means the ratio between one favorable event and the pool of events; where the nominator represents the sum of all probabilities of two letter pairs divided by all pairs which might occur in the given text.

Example: Suppose we have a set of 2600 letters of 100 A's, 100 B's... 100 Z's, then we can calculate the index of coincidence:

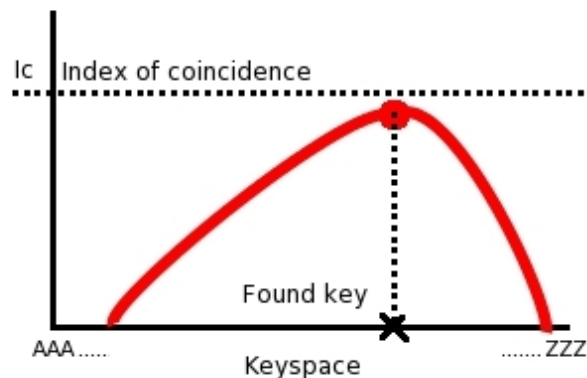
$$Ic = \frac{C_{100}^2 + C_{100}^2 + \dots + C_{100}^2}{C_{2600}^2} = \frac{26 * \frac{100!}{2!(100-2)!}}{2600!} = 26 * \frac{\frac{99 * 100}{2}}{2599 * 2600} = \frac{26 * 99 * 100}{2600 * 2599} \approx 0.0380915737$$

For simplicity, the value is multiplied by a factor of roughly 26 to obtain an Ic value of 1 in case of a perfect random distribution.

- **Combining the index of coincidence with a frequency table, we can determine the index of coincidence of every language represented by a unique number.**
- **This number can help in determining whether two texts put side by side are in the same language.**
- **The method of coincidence calculations in cryptography and applied cryptography is very important since we eliminate the human factor and have one programmatically number to compare a result with.**

Hill-Climbing and Frequency Analysis

- The Hill-Climbing method is based on the fact that a key is generally considered to be the strength of the whole cipher. The key is also the only piece of information which would wield a precise ciphertext.
- If we encrypt a text using a certain key and obtain a ciphertext. If then we vary the key slightly, in case SAC is not accomplished by the cipher, we obtain a similar ciphertext which is comparable to the first ciphertext and distinguished only by a delta error.
- The following graph presents a theoretic and simplified evolution of a plaintext during a decryption process using variable keys from all keyspaces. The distribution is similar to a Gaussian bell distribution:

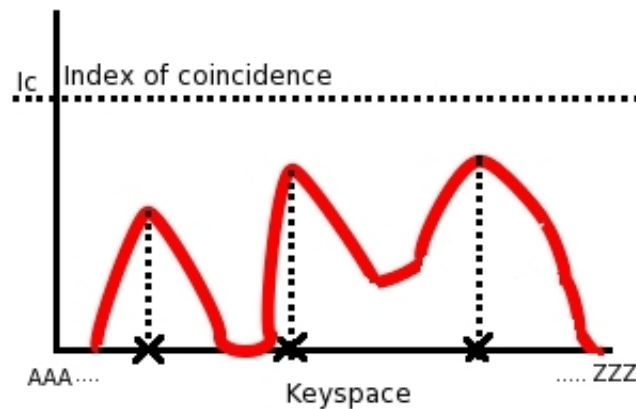


[Edit: Since SAC is excluded the key alterations would still "slide" the plaintext but the actual evolution would proceed on n-dimensions depending on the key length. - M. Ryan]

- We use the index of coincidence to programmatically detect whether the obtained result is actual plaintext and not gibberish.

Classic algorithm:

- 1.) Get a key $1/4^{\text{th}}$ of the way to the final key in the keyspace.
- 2.) Decrypt text and compare it to the index of coincidence and store in a variable X.
- 3.) Increment the chosen key by one bit in the keyspace.
- 4.) Decrypt text with the new key and compare it to the index of coincidence storing the result in a variable Y.
- 5.) Compare X and Y:
 - a.) If $Y > X$ then overwrite the key by the incremented key at step 3 and go to step 2.
 - b.) If $X < Y$ then decrement the key and go to to step 2.



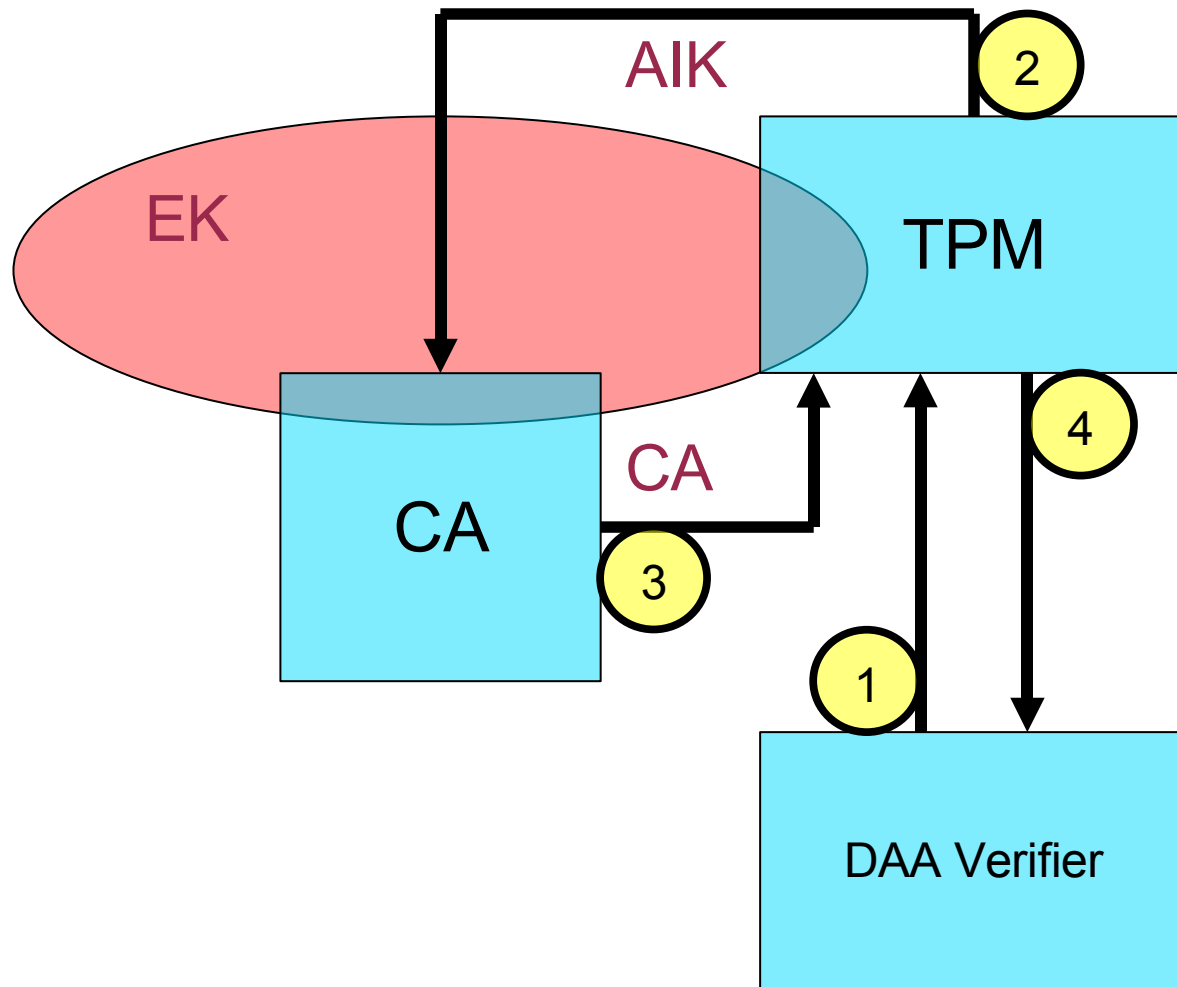
- In order to avoid “fake hills”, Alph uses a Monte-Carlo algorithm. It detects if the program is in an endless loop state and generates a random key, jumping to another place in the keyspace while keeping a statistic set of indexes of coincidence.

Protocols

- A protocol can be defined as the rules governing the syntax, semantics and the synchronization of communication.
- They usually specify one or more of the following properties:
 - Detection of the physical connection or the existence of another endpoint.
 - Handshaking
 - Negotiation of various characteristics
 - Message formating and breaks
 - Error correction and fallbacks
 - Session termination.
- In cryptography a "cryptographic protocol" usually incorporates at least one of the following properties:
 - Key negotiation or establishment
 - Entity authentication
 - Symmetric encryption
 - Data transport
 - Non-repudiation methods

DAA

- The DAA algorithm is based on three phases on two different levels in three steps and implies three communicating parts: the TPM, the CA and the DAA verifier.
- Compared to normal certificate authentications, in this case, the server (DAA verifier) checks the client (the TPM machine) based on a common Endorsement Key shared between the TPM and the CA.
- The protocol allows differing degrees of privacy. The transactions are always anonymous between the DAA verifier and the TPM machine (zero knowledge proof). However there is a shared "secret" between the TPM and the CA.
- The CA must remain maintain high availability while remaining secure since it takes part in every transaction.
- The common protocol implies that the CA would send the certificate back to the TPM and the TPM would use that certificate to authenticate to the DAA verifier. This can be tampered with (*).
- In some specification, the CA is able to connect-back to the TPM machine and verify if the TPM is a rogue.



- 1.) The DAA Verifier (server) requests the TPM machine to authenticate.
- 2.) The TPM generates an RSA key pair AIK based on the hardwired EK.
- 3.) The CA issues a certificate and sends it back to the TPM.
- 4.) The TPM machine uses the certificate to authenticate to the DAA Verifier.