Seguridad

La seguridad en la red no es solo sobre las comunicaciones y redes, esta incluye la **física**, **información**, **personal**, **software**, **operaciones** y **comunicaciones y redes**.

Las principales funciones de la seguridad en la red son:

- **Confidencialidad**: Solo el receptor puede "entender" el contenido del mensaje. El mensaje encripta el mensaje y el receptor desencripta el mensaje.
- **Integridad de los mensajes**: El mensaje no ha sido alterado sin detectarlo.
- **Autenticación**: El receptor confirma la identidad del emisor.
- **Accesibilidad y disponibilidad**: Los servicios deben ser accesibles y estar disponibles para los usuarios.

Para hacer los mensajes legibles únicamente entre emisor y receptor, se usan **claves**. Una clave podría definirse como una función la cual cifra un texto. Si tenemos un mensaje **m**, este se cifra desde el emisor A con la clave $\mathbf{K_a}$ de la siguiente forma $\mathbf{K_a(m)}$, para descifrar el mensaje el receptor B usa su clave $\mathbf{K_b}$ sobre el texto cifrado $\mathbf{K_b}$ $\mathbf{(K_a(m))=m}$. Si las claves en ambas partes son iguales, el cifrado es **simétrico**, si fuesen distintas el cifrado sería **asimétrico**.

Existen varios tipos de cifrados:

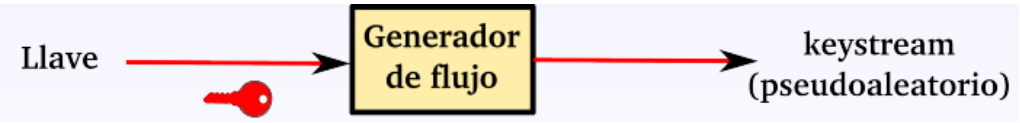
- Cifrado por substitución. Se sustituye una cosa por otra mediante una clave.
- Cifrado polialfabético. Se usan **n** cifrados monoalfabéticos M_1, M_2, M_3… M_n. Estos cifrados se usan cíclicamente, por ejemplo: n = 3; M_2, M_1, M_3 ; M_2, M_1, M_3 ; M_2, M_1, M_3 … La **clave** en este caso son los n cifrados y el patrón cíclico.

También tenemos diferentes tipos de desencriptación:

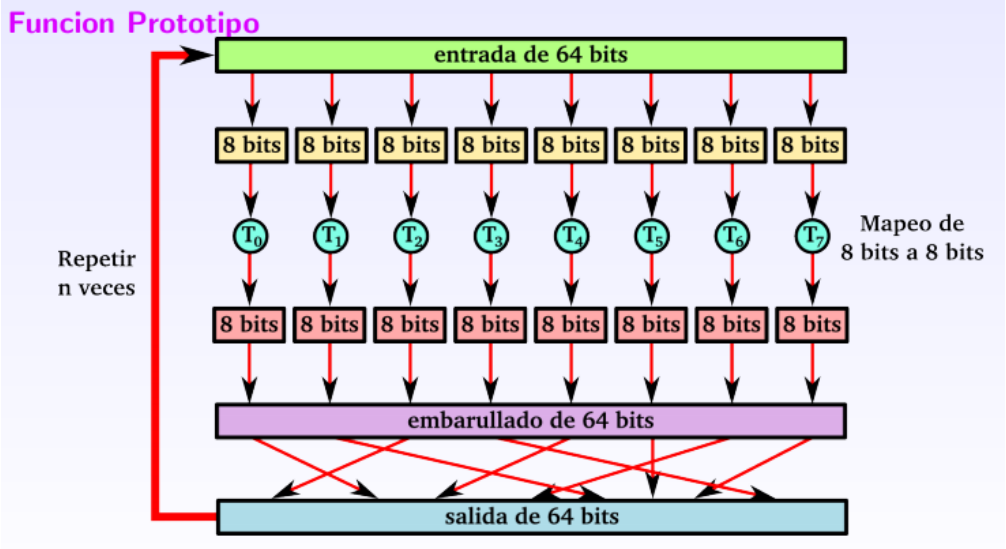
- Ataque basado solo en texto cifrado: El atacante tiene el texto cifrado para analizar, prueba todas las claves posibles distinguiendo texto real de incoherencias. Se hace un análisis estadístico.
- Ataque basado en texto legible conocido: El atacante tiene algo de texto legible correspondiente a texto cifrado, por ejemplo, en cifrado monoalfabético el atacante determina pares.
- Ataque por texto legible seleccionado: El atacante se las ingenia para conseguir que el transmisor envié un texto conocido que él verá en su forma encriptada.

La criptografía normalmente usa un algoritmo conocido por todos y sólo las claves son secretas. Tenemos varios tipos:

- Clave simétrica. Dentro del cifrado simétrico tenemos otros 2 tipos:
  - Cifrado de flujo:
    - Cifrado de flujo de símbolos básicos. Cada bit del texto legible se combina con el bit del flujo de clave, así se obtiene el texto legible. Para combinar dichos bits se usa una puerta XOR.

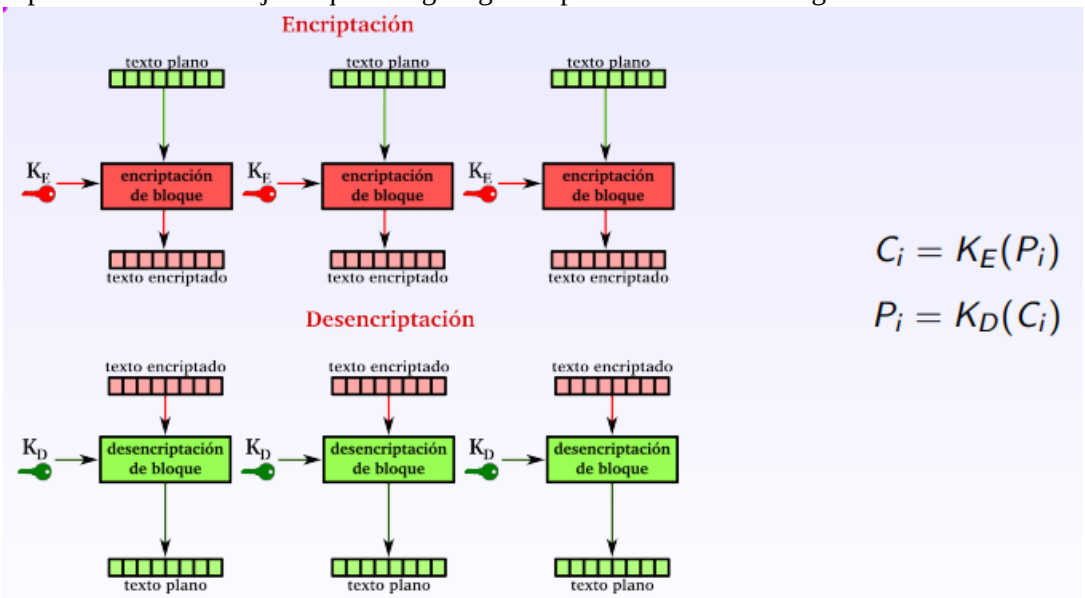


    - Cifrado de flujo RC4. Su uso es bastante común. RC4 es secreto y es una marca registrada, sin embargo, el algoritmo fue descubierto y ahora se usa el algoritmo libre **ARC4**. Se considera bueno. Usa una llave de 1 a 256 bytes, se usa en WEP de 802.11 (Wired Equivalent Privacy), puede ser usado en SSL (secure sockets layer) y, es rápido, simple y eficiente.
  - Cifrado de bloque. El mensaje es procesado en bloques de **k bits**, se usa un mapeo 1-a-1 para mapear k bits del texto a kbits cifrados. El problema es que el numero de entradas en la tablas de asciende muy rápido dependiendo de k, por lo que en vez de usar una tabla conviene usar una función que simule una tabla de permutación aleatoria.

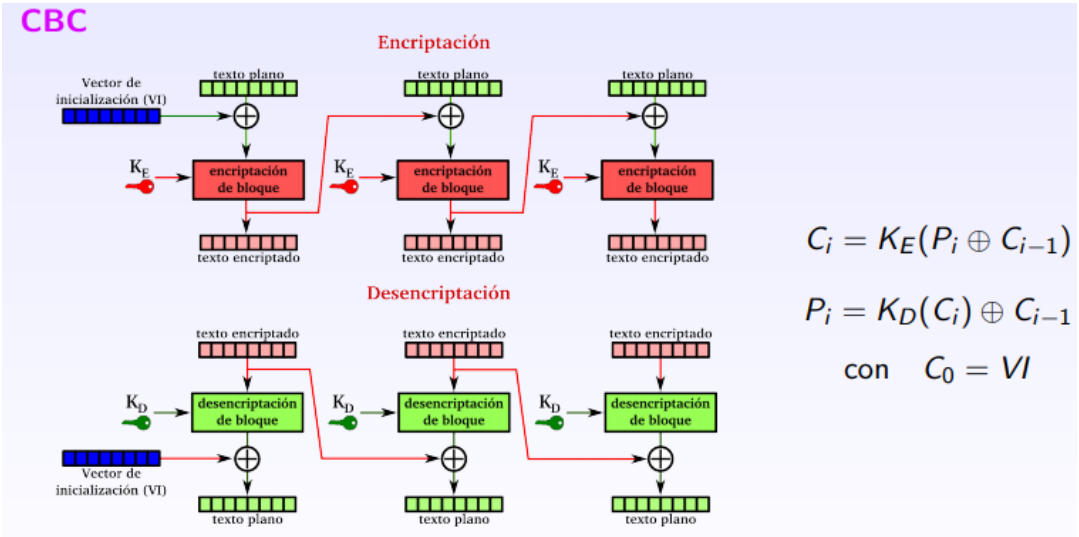


La razón por la que se itera es porque si solo ejecutásemos la función una vez, un cambio de un bit en la entrada sólo altera 8 bits de salida, si lo ejecutamos una segunda vez, los 8 bits afectados se dispersan afectando a otros. El número de ciclos depende del tamaño del bloque.
Existen 3 modos básicos:

- ECB (Electronic CodeBook): Un mensaje es troceado en bloques que se encriptan de forma separada. La desventaja es que códigos iguales producen resultados iguales.



- CBC (Code-Block Chaining): Antes de ser cifrado cada bloque hace un XOR con el bloque previo ya cifrado. Además, se depende de un vector de inicialización que puede hacer único cada mensaje.

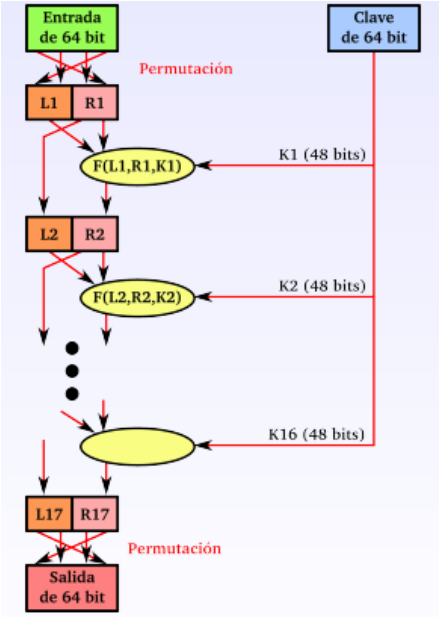


- PCBC (Propagating Code-Block Chaining): Diseñado para que pequeños cambios en el texto cifrado se propaguen más que en el modo CBC.

En cuanto a los vectores de inicialización no necesitan ser secretos, pero son imprescindibles. No se deben reutilizar vectores de inicialización con la misma clave. El vector de inicialización se puede guardar al inicio del fichero.

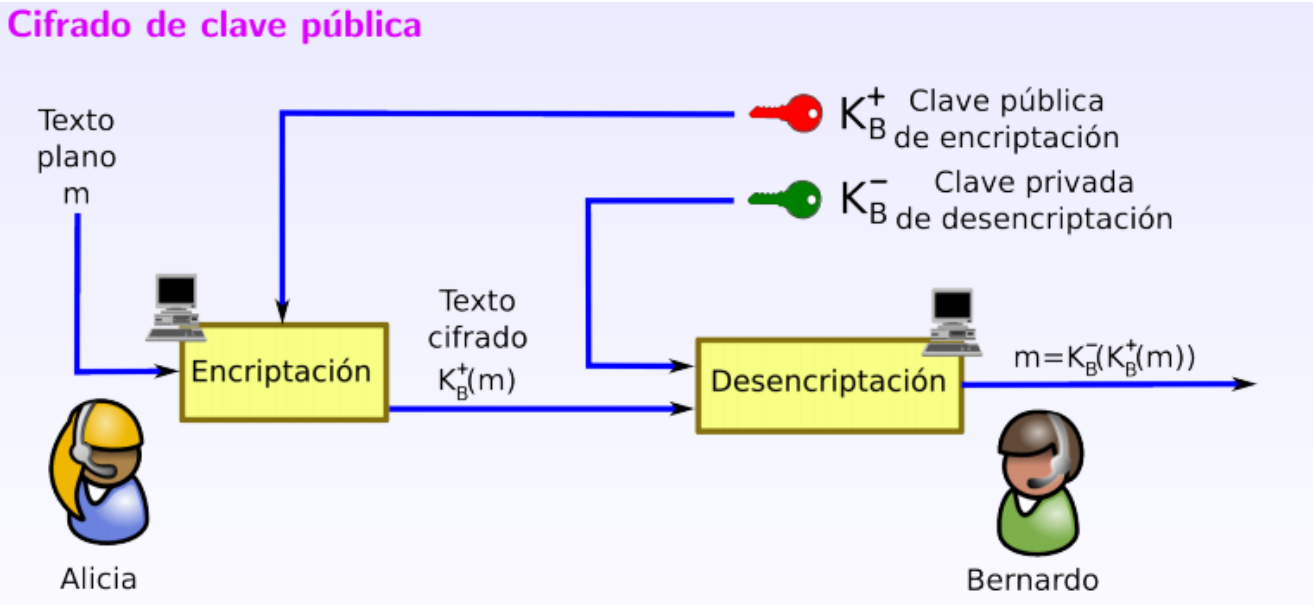
El objetivo es no repetir nunca el vector de inicialización, por lo que como mínimo debe tener una longitud de 16 bytes que son 2128 posibles valores.

- DES (Data Encryption Standard). Estándar americano de 1993 que usa una clave de 56 bit + 8 bit de paridad. 3DES encripta 3 veces con diferentes claves (en realidad cifra, descifra y vuelve a cifrar).



- AES (Advanced Encryption Standard). Nuevo estándar estadounidense de clave simétrica (2001), remplaza a DES. Los bloques de datos son de 128 bits y la longitud de las claves son de 128, 192, 256 bits.

- Clave asimétrica. En esta encriptación está el **cifrado simétrico**, en el cual son necesarios un transmisor y un receptor que compartan un secreto, y el **cifrado asimétrico**, en el cual no se comparte ningún secreto, existe una clave pública conocida por todos y una clave privada que solo la conoce el propietario.



- RSA. Es un protocolo en el cual existen 2 claves, una pública usada para encriptar y otra privada usada para desencriptar. El emisor encripta con la clave pública del receptor, y el receptor desencripta

con su clave privada que solo él debe conocer. RSA es seguro pero costoso ya que implica exponenciación. DES es 100 veces más rápido que RSA.

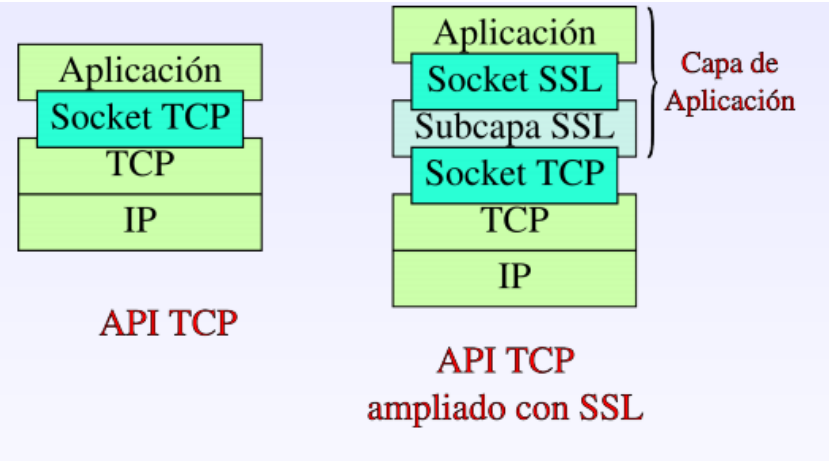
- **Integridad de los mensajes (funciones hash)**. Para garantizar la integridad de los mensajes hay que determinar que el mensaje no ha sido modificado o duplicado por otra fuente.
  Una función hash toma como entrada cualquier mensaje y genera un string de tamaño fijo único. Se considera una función suprayectiva, hay más secuencias de entrada que de salida. Es irreversible y único, es muy difícil que se produzcan colisiones entre 2 funciones hash diferentes.
  Algoritmos de función hash:
  - MD5, es una función hash usada ampliamente. Genera un resumen del mensaje de 128-bit en proceso de 4 pasos.
  - SHA-1, es un estándar EE. UU. Genera un resumen de 160-bit.

  Otra alternativa es el **Checksum**, tiene algunas propiedades de función hash, produce un resumen de tamaño fijo y es una aplicación suprayectiva.
  También tenemos **MAC (Message Authentication Code)**, esta autentica al emisor, verifica la integridad del mensaje y no encripta. Es necesaria una clave secreta.

Existe lo que denominamos **firmas digitales**, estas son análogas a las firmas a mano. El transmisor firma digitalmente un fichero, estableciendo así que él es el dueño (usa su clave privada). Es verificable y no repudiable. La firma digital verifica que el emisor firmo el mensaje, solo el emisor pudo firmar el mensaje, y el mensaje no ha sido modificado.

En internet se usan otros protocolos como **https**, actualmente es uno de los más usados por los navegadores y servidores web. Los objetivos de este protocolo fueron: Permitir comercio electrónico, encriptación, autenticación de servidores web, opcionalmente autenticación de clientes y minimizar riesgos al hacer negocios con nuevos clientes.

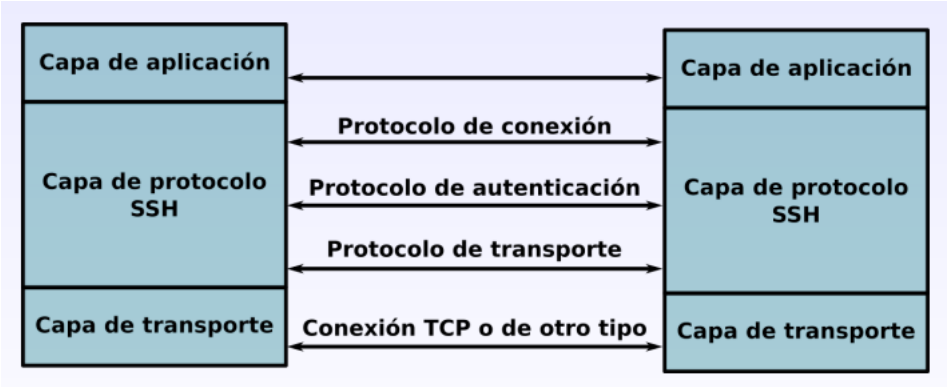
Utiliza el protocolo TCP siguiendo la siguiente estructura:



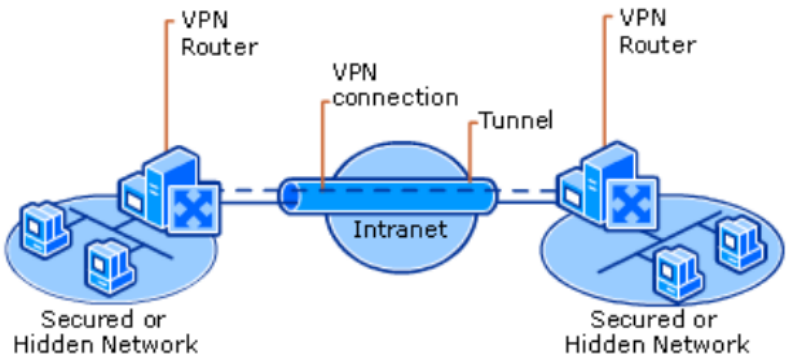
SSL provee una interfaz de programación de aplicaciones (API) para desarrollar aplicaciones. Los mecanismos usados por SSL son:

- Herramientas de cifrado tales como algoritmos de clave pública, algoritmos de encriptación simétrica y algoritmos MAC. SSL permite varios mecanismos de cifrado.
- Negociación, el cliente y el servidor deben acordar mecanismos de cifrado. El cliente ofrece opciones y el servidor elige entre las ofrecidas.

Otra arquitectura es **SSH**. Proporciona varios mecanismos de autenticación: Basada en el uso de criptografía de clave pública, basada en nombre de usuario y contraseña, y basada en procedencia (IP).



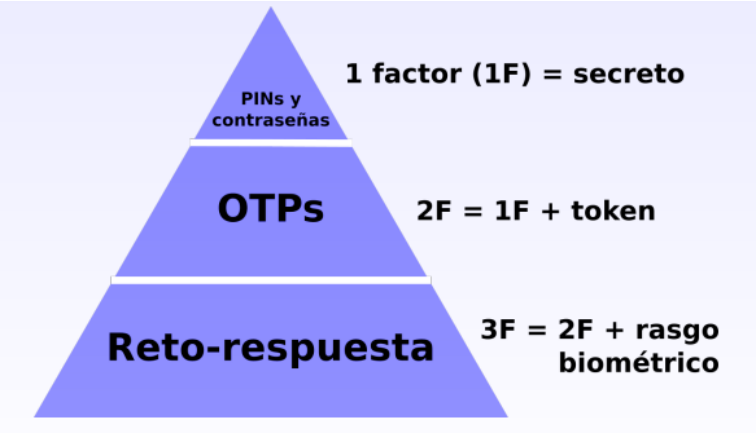
VPN. Virtual private network.



Point-to-point tunneling protocol (PPTP). Funciona a nivel de enlace y está diseñado para conexiones sencillas, únicas entre cliente y servidor.

La **autenticación** trata de probar que la "otra parte" es una persona. Los dos conceptos principales son:

- Identificación: Autentificación especifica de personas. Usa **algo que sabemos** (contraseñas, PIN, claves criptográficas…), **algo que tenemos** (DNIe, Tokens, Teléfonos…) y **algo que somos** (Firma manuscrita, huella dactilar, retina, voz…).



En el sistema **1F**, ambas partes, y solo ellas, conocen un secreto compartido. Las contraseñas son fáciles de implementar, fáciles de usar y no tienen coste, el problema es que son muy vulnerables a ataques.

En el sistema **2F**, el usuario debe demostrar, además del conocimiento de un secreto, posesión de un token. El sistema es más seguro que 1F, pero si se pierde el token o funciona mal no podremos autenticarnos, es incómodo para el usuario y es complejo de implementar.

Por último, en el sistema **3F**, utilizan lo anterior y además un rasgo biométrico para la identificación del usuario. Es la más segura, aunque no suelen estar bien diseñadas, son incomodas, muchos problemas de privacidad, sólo es funcional para personas.

- Autorización: Gestión de permisos de una parte ya autenticada.

Las contraseñas son un gran problema debido a su fragilidad, es por eso por lo que existen **políticas de contraseñas** las cuales tienen en cuenta la longitud, complejidad, expiración y un límite de fallos.

Un método para crear contraseñas puede ser **passphrases**, usar la primera letra de cada palabra.

Método **Persona-Acción-Objeto (P.A.O.)**. Lugar, persona, acción, objeto. "Beyonc´e cocinando una chincheta en la Puerta del Sol" ⇒ beycocchipursol,

Patrones de teclado. 

Gestores de contraseñas, generan contraseñas muy complejas para diferentes servicios y las guardan por ti.

Los **ataques** se pueden producir a:

- Al servidor. Pueden atacar o al sistema operativo o a los servicios y aplicaciones (SQL injection).
- A la red. Atacan a la comunicación entre el cliente y servidor.
    - Atacan la confidencialidad. Pueden obtener el token del usuario en su transmisión por la red (sniffing).
    - Atacan la integridad. Modificación de la comunicación cliente-servidor (ARP spoofing...).
    - Atacan autenticidad. Suplantan la identidad del usuario una vez autenticado y autorizado (hijacking, spoofing).
- Al cliente.
    - Mediante ingeniería social. Manipulando personas.
    - Mediante (Digital) Trashing. Buscando información util para el atacante, post-it, notas en el cuaderno.
    - Mediante shoulder surfing. "Surfea" por encima del hombro del usuario cuando éste escribe su contraseña.

Se atacan también a las contraseñas, ya que en general los seres humanos eligen muy malas contraseñas. Se pueden usar diccionarios que estén estructurados por nombres de familiares, nombre del novio, nombre de la mascota… A parte, al diccionario se le pueden añadir reglas.

Otro tipo de ataque es por diccionario inverso, se fija la contraseña y se varía el usuario.

Existen 2 tipos de cracking de contraseñas:

- Online: Siempre son posibles aunque son lentos y tienen un número máximo de intentos. Los **salvaguardas** que pueden protegernos son: limitar el número de intentos fallidos, activar captcha, no usar contraseñas por defecto, distribuir contraseñas por un medio seguro, no por email.
- Offline: Robos a bases de datos de usuarios. Para prevenir esto no guardamos la contraseña en plano, se guarda el hash de la contraseña. Esto aun así no es suficiente, se debe concatenar un **salt** a la contraseña original y se hace el hash de está nueva cadena. Para verificar la contraseña sería necesario recuperar el salt del usuario, recalcular el hash con la contraseña y se compara el resultado con el almacenado.

Existen las **tablas Rainbow**, estas precalculan todas las posibles combinaciones y aplican el hash sobre estas combinaciones, en resumen, guardan todas los hash.