

Seguridad y Privacidad:

Tecnologías criptográficas para la protección de la
privacidad

Contenido

- **Tecnologías criptográficas para la protección de la privacidad**
 - Introducción y motivación:
 - ¿Big Data is a Big Problem?
 - Algunas ventajas y algunos problemas de Big Data.
 - Problemas éticos de Big Data:
 - Nos centraremos en privacidad.
 - Fundamentos criptográficos de la protección de la información
 - Como se puede obtener seguridad y privacidad con tecnologías de la información:
 - Un método posible es mediante funciones criptográficas:
 - Algunas funciones criptográficas que proporcionan seguridad.
 - Intercambio de claves.
 - Problema de la gestión de la identidad digital mediante certificados digitales: el estándar X.509
 - Definición de los conceptos de trazabilidad, enlazado, anonimato y pseudo-anonimato: firmas grupales
 - Navegación anónima: introducción a las redes de mezcla de tráfico, *onion routing* y ofuscación de tráfico

Introducción y Motivación

- El Big Data representa una gran solución para las empresas hoy en día, transformando el negocio de las mismas y generando una serie de ventajas:
 - Implementación de mejoras tecnológicas para adquirir datos, permitiendo la mejora de la compañía por el descubrimiento de las necesidades de la misma.
 - Análisis de datos → mejora de toma de decisiones de la compañía.
 - Análisis de datos → facilita la evaluación de productos (nuevos productos, rediseño, etc.).
 - División de clientes para acciones personalizadas.
 - Mejora la accesibilidad y fluidez dentro de la propia empresa.
 - Etc.
- También el Big Data empieza a cobrar un papel importantísimo en todos los campos:
 - en salud biomédica como ya os ha contado Pablo,
 - en energía como os ha contado Julia,
 - en investigación como por ejemplo en genética como os ha contado Irene,
 - en seguridad nacional,
 - Etc.

Introducción y Motivación

- La otra cara de Big Data son los importantes riesgos que puede producir:
 - asociados a sus procesos análisis predictivo y recolección indiscriminada de información y especialmente la violación de datos.
- Big Data puede violar claramente la privacidad ([K. Crawford & J. Schultz, 2014](#)), y puede plantear muchas cuestiones éticas.
- Algunos ejemplos reales de violación de privacidad (I):
 - [El 16 de febrero de 2012, el New York Times](#) publicó un artículo acerca de la capacidad de la empresa americana Target de identificar cuando una cliente está embarazada, a través de detalles íntimos sobre sus patrones de consumo con los minoristas. Todo empezó con Andrew Pole que acababa de empezar a trabajar como **técnico de estadística** para Target en 2002.

Introducción y Motivación

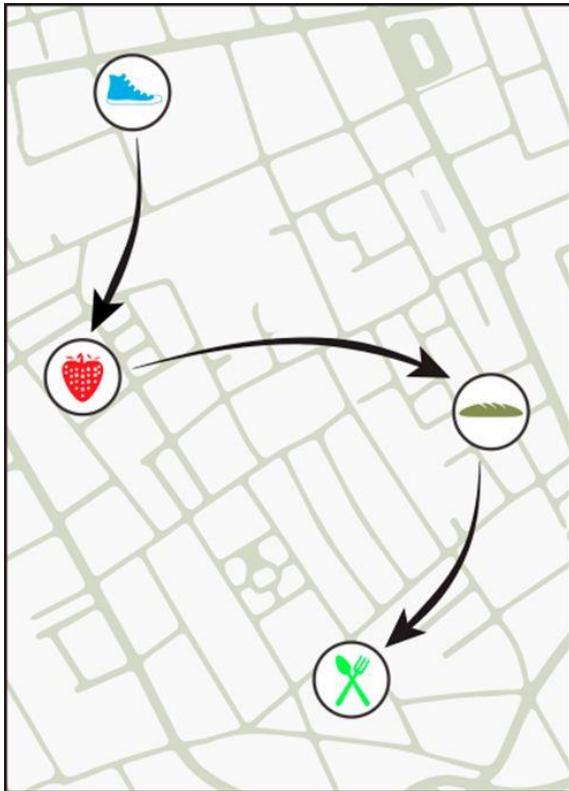
- Algunos ejemplos reales de violación de privacidad (II):
 - Raytheon Riot Software predice comportamiento basado en Social Media. Para ello utiliza los datos y fotos colgados en Facebook y otras redes sociales, los mini-mensajes de Twitter y además aprovecha la localización por GPS que la gente se activa en sus *smartphones* y con aplicaciones como Foursquare u otras. Con todo ello, y ciertas rutinas de las personas que se aprenden y se extraen de los datos, el software de Raytheon podría predecir comportamientos y desplazamientos de las personas rastreadas ¿Quién es Raytheon? Es el 5º fabricante mundial de armas.
 - El software no se vende al público aunque la compañía ha reconocido que la tecnología es compartida por el Gobierno de EE.UU., para seguimiento de delincuentes (ver el video para una demostración de como funciona: "Sabemos donde está Nick y cómo es Nick", explica Urch en el video. "Ahora queremos tratar de predecir dónde puede estar en el futuro").
 - Según Centro Electrónico de Información de Privacidad (EPIC), este hecho no ha generado más seguridad entre la población, si no que al contrario ha generado mucha inquietud y preocupación como pueden ser utilizados los datos personales sin ninguna legislación vigente seria en estos momentos.

Introducción y Motivación

- Algunos ejemplos reales de violación de privacidad (III):
 - The end of privacy (número especial de la prestigiosa revista Science de Enero 2015).
 - Por ejemplo en el trabajo (Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, Alex "Sandy" Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. Science 30 January 2015: Vol. 347 no. 6221 pp. 536-539 DOI: 10.1126/science.1256297) se estudian y analizan 3 meses de registros de tarjetas de crédito de 1,1 millones de personas y los autores demuestran que cuatro puntos espacio-temporales son suficientes para reidentificar el 90% de los individuos (ya existen técnicas para gestionar el riesgo de reidentificación como por ejemplo la k-anonimización). Se demuestra que saber el precio de una transacción aumenta el riesgo de reidentificación en un 22%, en promedio.
 - El conjunto de datos fue simplemente anonimizado, lo que significa que no contiene ningún nombres, números de cuenta, o identificadores obvios. Cada transacción fue etiquetada con marca de tiempo con una resolución de 1 día y asociada a una tienda (distribuidas en todo el país).
 - Un ejemplo de datos la siguiente transparencia:

Introducción y Motivación

Foto extraída de: [The end of privacy](#) (número especial de la prestigiosa revista Science de Enero 2015).



shop	user_id	time	price	price_bin
boot	7abc1a23	09/23	\$97.30	\$49 – \$146
strawberry	7abc1a23	09/23	\$15.13	\$5 – \$16
bread	3092fc10	09/23	\$43.78	\$16 – \$49
fork/knife	7abc1a23	09/23	\$4.33	\$2 – \$5
swimmer	4c7af72a	09/23	\$12.29	\$5 – \$16
bread	89c0829c	09/24	\$3.66	\$2 – \$5
fork/knife	7abc1a23	09/24	\$35.81	\$16 – \$49

La [reidentificación](#) es el análisis de ficheros anonimizados con el fin de identificar a personas específicas a partir de ellos

Introducción y Motivación

- La violación de la privacidad en Big Data genera una gran controversia, ya que también se puede proponer también como una herramienta fundamental para luchar contra el crimen:
 - Big Data puede revelar ideas increíbles e incluir predicciones en donde el crimen va a suceder: [Andrey Bogomolov, Bruno Lepri, Jacopo Staiano, Nuria Oliver, Fabio Pianesi, and Alex Pentland. Once Upon a Crime: Towards Crime Prediction from Demographics and Mobile Data. 2014.](#)
 - ¿[Minority Report](#) puede ser real?



=



Introducción y Motivación

Foto extraída de: Andrey Bogomolov, Bruno Lepri, Jacopo Staiano, Nuria Oliver, Fabio Pianesi, and Alex Pentland. Once Upon a Crime: Towards Crime Prediction from Demographics and Mobile Data. 2014.

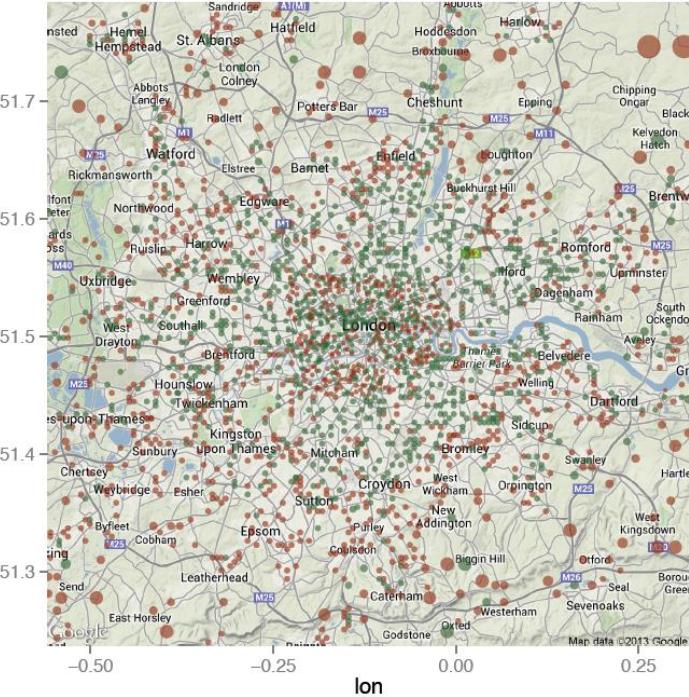


Figure 2: *Ground Truth of Crime Hotspots*

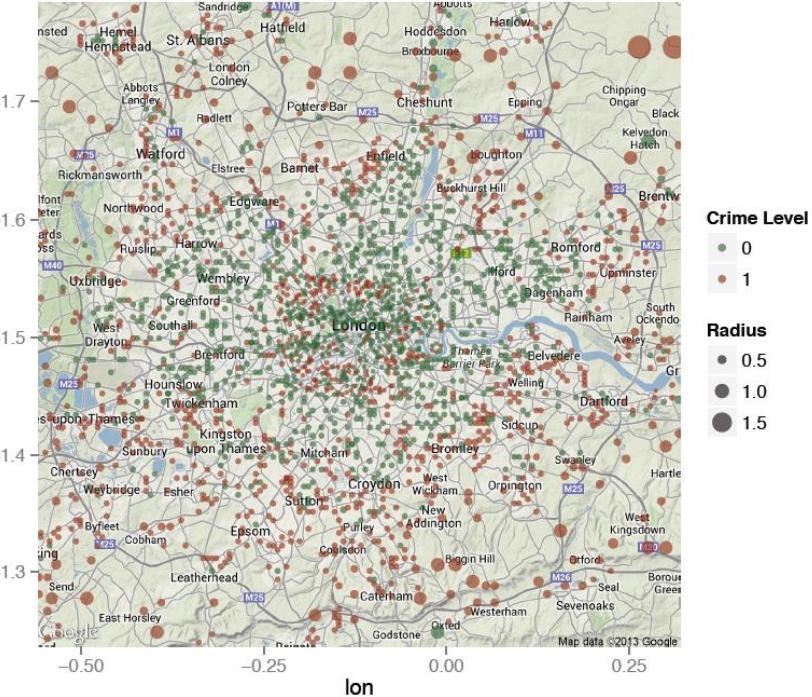


Figure 3: *Predicted Crime Hotspots*

Introducción y Motivación

- La Agencia Española de Protección de Datos (AEPD) aconseja K-Anonimidad para conseguir privacidad (nota de prensa del 14 de junio 2019).
- ¿Tenemos algún sitio que nos informe de la violaciones de privacidad actuales?
 - SI, el mejor sitio es EPIC:
 - Avisos de violación de privacidad y noticias del momento relacionadas con la privacidad a todos los niveles, con actualización cada dos semanas:
<https://www.epic.org/alert/>
- ¿Se preocupan las empresas por este problema?
 - SI, ya empiezan poco a poco, y por poner un ejemplo el de KPMG.
 - KPMG ha elaborado recientemente un informe que analiza la seguridad y privacidad en el contexto de Big Data:
 - [Navigating Big Data's Privacy and Security Challenges, kpmg.com \(KPMG 2014\)](#).

Introducción y Motivación

- En este informe se identifican los cinco grandes retos en seguridad y privacidad que las empresas deben abordar para asegurar un correcto control de su programa de Big Data:
 1. La gobernanza del Big Data.
 2. Mantener los requisitos originales de privacidad y seguridad a lo largo del ciclo de vida de la información.
 3. Re-identificación del riesgo.
 4. Terceras partes—uso y respeto de las obligaciones contractuales.
 5. Interpretar la normativa vigente y anticiparse a futuras regulaciones.

Introducción y Motivación

- La gobernanza del Big Data:
 - La implantación **Big Data** → puede conducir a la creación o al **descubrimiento de información confidencial** o desconocida hasta el momento, combinando diferentes conjuntos de datos.
 - Las organizaciones que buscan instaurar programas de Big Data sin contar con un **marco fuerte de gobernanza** se arriesgan a enfrentar dilemas éticos ante la ausencia de procedimientos y directrices concretas que seguir.
 - Por lo tanto, disponer de un sólido **código ético** junto con procesos, entrenamiento, personal y medidas adecuadas es imperativo para gobernar todo lo relacionado con el programa de Big Data.

Introducción y Motivación

- Mantener los requisitos originales de privacidad y seguridad a lo largo del ciclo de vida de la información:
 - La **información es dinámica** e interacciona con otra información.
 - La información recopilada que se incluye en los programas de Big Data guardará relación con otros conjuntos de datos, que, en última instancia, podrán generar nueva información o alterar los datos originales de diferentes maneras, a menudo impredecibles.
 - Las organizaciones deben asegurarse de que todos los requisitos de seguridad y privacidad que se aplican a los conjuntos originales de datos sean monitorizados y mantenidos en los procesos de Big Data a lo largo del ciclo de vida de la información, desde la recopilación de los datos hasta su divulgación o destrucción

Introducción y Motivación

- Re-identificación del riesgo (muy relacionado con el punto anterior):
 - Aquellos datos procesados, gestionados o modificados por los programas de Big Data pueden representar beneficios tanto internos como externos para las organizaciones.
 - A menudo, los datos deben ser anónimos para proteger la privacidad de la fuente original de información, como es el caso de consumidores o vendedores.
 - La información que no atraviesa un proceso de anonimato adecuado antes de divulgarse externamente (o en algunos casos, también internamente) puede **comprometer información confidencial al combinarse con datos complejos previamente recopilados**, que incluyen geolocalización, reconocimiento de imágenes y seguimiento del comportamiento.
 - Si la información es anónima, se debe evaluar la correlación existente entre los distintos conjuntos de datos, pues, de lo contrario, terceras partes con acceso a diferentes informaciones podrían ser capaces de re-identificar a individuos anónimos

Introducción y Motivación

- Terceras partes—uso y respeto de las obligaciones contractuales:
 - Haciendo uso del Big Data, combinar los datos propios con los de otras organizaciones puede esclarecer perspectivas que sería imposible descubrir si se contara solo con la información particular.
 - No obstante, esto puede comportar grandes **riesgos** si los **terceros** a los que se acude no cuentan con sistemas de protección de datos adecuados, arriesgando en consecuencia, la privacidad y la seguridad de la información.
 - De este modo, antes de compartir datos con terceros, las organizaciones deben evaluar si estos cuentan con las prácticas adecuadas en términos de confidencialidad y seguridad de los datos.

Introducción y Motivación

- Interpretar la normativa vigente y anticiparse a futuras regulaciones:
 - Ni Estados Unidos ni la Unión Europea poseen normativas específicas de Big Data; sin embargo, existen leyes que regulan la recopilación, el uso y el almacenamiento de ciertos tipos de información personal, como es el caso de datos financieros, sanitarios y de menores.
 - Así mismo, se aprecia una creciente supervisión normativa, evidenciada por el énfasis que pone la [Comisión Federal de Comercio](#) en los denominados [Data Brokers](#) y en base al grupo de trabajo del [artículo 29](#) sobre la protección de datos ([Directiva 95/46/CE del Parlamento Europeo y del Consejo](#), ver [documento](#)) y el principio de limitación de los fines del [Big Data](#). A partir del 25 de mayo de 2018, el Grupo de trabajo del artículo 29 dejó de existir y fue reemplazado por el [Consejo Europeo de Protección de Datos \(EDPB\)](#).
 - Con el objetivo de cumplir con la normativa vigente y mantenerse actualizadas en cuestiones regulatorias, las organizaciones deben definir un conjunto inicial de leyes que necesitarán renovar de forma periódica.

Introducción y Motivación

- **Ya está claro hoy en día que Big Data plantea problemas de privacidad** personales, genera nuevas preguntas sobre la identidad personal, sobre todo quien es dueño de nuestros datos personales y cómo el aumento de la presencia y la disponibilidad de más datos influyen en nuestra fama o reputación:
 - Como hemos visto antes, las empresas empiezan a tomar parte activa en la regulación de la privacidad y seguridad en el contexto de Big Data planteando una serie de retos.
 - The end of privacy (número especial de la prestigiosa revista Science de Enero 2015).
 - The ethics of big data: Focus Feature (en la prestigiosa revista PLOS Computational Biology): es un foro para la discusión de temas de vanguardia en el campo, con especial énfasis en los desafíos éticos que acompañan al uso de "grandes datos" en biología y medicina.
 - The 5 worst big data privacy risks.
 - Etc.

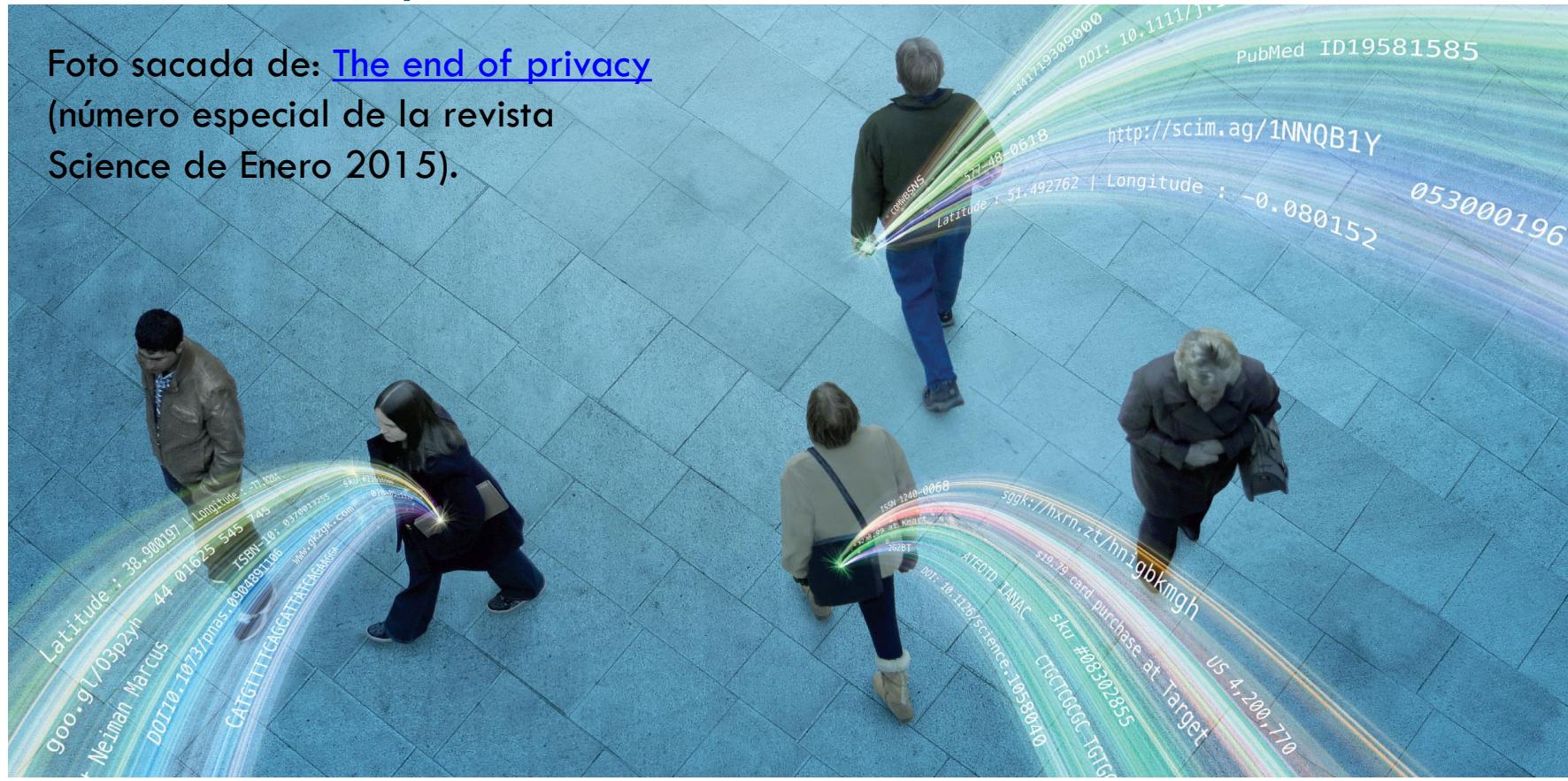
Introducción y Motivación

- Se pueden definir cuatro **elementos comunes** que puede ser considerado como un marco general para la ética de Big Data:
 - Identidad: ¿Cuál es la relación entre nuestra identidad en línea y nuestra identidad fuera de línea?
 - **Privacidad: ¿Quién debe controlar el acceso a los datos?**
 - Propiedad de los datos: ¿Quién posee los datos, se puede transferir los derechos a la misma, y cuáles son las obligaciones de las personas y organizaciones que generan y utilizar esos datos?
 - Reputación: ¿Cómo podemos determinar qué datos son dignos de confianza? Ya sea sobre nosotros mismos, los demás, o cualquier otra cosa ya que Big Data aumenta exponencialmente la cantidad de información y formas en las que pueden interactuar con él. Este fenómeno aumenta la complejidad del ser humano modulando como este es percibido y juzgado.
- En esta primera parte nos vamos a centrar en que tipo de metodologías del campo de “Computer Science” existen para intentar atacar el problema de Seguridad y Privacidad en Big Data, problema que aún no está resuelto.
- **Porque en verdad la realidad de hoy en día se puede resumir en la siguiente transparencia**

Introducción y Motivación

Foto sacada de: [The end of privacy](#)

(número especial de la revista Science de Enero 2015).



Introducción y Motivación

- No obstante se va avanzando en la protección de los datos privados de los ciudadanos:
[GDPR/RGPD: General Data Protection Regulation/Regulación General de Protección de datos.](#)
- El 25 de mayo de 2018 ha entrado en vigor en toda Europa esta nueva ley de protección de datos: [GDPR](#) (General Data Protection Regulation).
- Realmente entro en vigor el 24 de mayo de 2016, pero a partir de el 25 de mayo de 2018 es obligado a cumplir.
- La Ley Orgánica de Protección de Datos ([LOPD](#)) se adapta a la normativa europea para no contradecirla.
- Básicamente es una normativa que afecta a todas aquellas empresas que traten datos de los ciudadanos europeos aunque sean de Estados Unidos, como por ejemplo Google o Facebook.
- Ahora **hay que dar tu consentimiento inequívoco para que las empresas puedan usar tus datos** si eres ciudadano europeo. Así unifica, los derechos como las obligaciones de los ciudadanos europeos en esta materia tan importante.

Introducción y Motivación

- Y esto va más allá, te tendrán que decir qué datos están utilizando, cómo los están tratando, para qué y quién es la persona responsable de los mismos.
- Hay una nueva estrategia en el tratamiento de datos, hay que informar obligatoriamente a los usuarios sobre qué información cedemos y para qué se usa.
- Grandes multas para las compañías que violen esta ley.
- Más información:
 - https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_es.htm
 - https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=uriserv:OJ.L_.2016.119.01.0001.01.ENG&toc=OJ:L:2016:119:TOC
 - <https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=celex:32009L0136>
 - https://ec.europa.eu/info/law/law-topic/data-protection/reform/rights-citizens_es
 - Hasta los bancos ya ponen la información al respecto: <https://www.bbva.com/es/gdpr-nueva-ley-europea-proteccion-datos/>

Privacidad y Seguridad

- Una de las formas de proporcionar privacidad y seguridad es mediante funciones criptográficas.
- Para ello vamos ha realizar una introducción a seguridad y criptografía, para poder entender como generar seguridad y privacidad mediante funciones criptográficas que proporcionan anonimia:
 - Introducción.
 - Criptografía simétrica y asimétrica.
 - Intercambio de claves.
 - Funciones hash.
 - Autenticación con firmas digitales.
 - Certificados de Clave Pública Certificados X.509.
 - Firmas grupales: autenticación con privacidad.
 - Navegación anónima: redes de mezcla de tráfico.

Introducción

- La seguridad de los sistemas de información está relacionada con la informática y comunicación en presencia de **adversarios**.
- **Sistemas de información:**
 - PC
 - Teléfonos
 - Redes de computadores
 - Cajeros automáticos
 - RFID
 - Puntos de Wireless
 - Dispositivos médicos
 - Email
 - Coches
 -
- Todo es **digital** hoy en día, o casi todo.....
- La seguridad informática se refiere a los objetivos de seguridad o las políticas de seguridad: que se quiere proteger, que actividades o eventos deberían ser prevenidos o detectados.

Introducción

- Los métodos básicos para obtener la seguridad a través de criptografía pueden agruparse en 4 principales áreas:
 - Cifrado Simétrico
 - Cifrado Asimétrico
 - Algoritmos de integridad de datos
 - Protocolos de seguridad (engloban a los anteriores)
- [El libro del NIST \(actualizado\)](#) define el término seguridad informática más o menos como:
 - *la protección que se otorga a un sistema de información automatizado con el objetivo de alcanzar la preservación de la integridad, disponibilidad y confidencialidad de la información y los recursos del sistema (incluye hardware, software, firmware, información / datos, y telecomunicaciones).*
- [NIST](#): National Institute of Standards and Technology

Introducción

Confidencialidad



SEGURIDAD
*Informática y las
comunicaciones*



Integridad

Disponibilidad

- Para chequear y testear CIA (Confidentiality Integrity Availability): **Auditorías** (verificar que se cumplen las normas de seguridad apropiadas).

Introducción

- Los mecanismos de seguridad o control de seguridad son las componentes técnicas o métodos para asegurar los servicios de seguridad, típicamente dos formas:
 - **Prevención:** Mantener la política de seguridad para no ser violada.
 - Passwords, cifrados, etc.
 - **Detección:** Detectar cuando la política de seguridad es violada.
 - Detección de intrusión en redes, chequeos de virus, etc.
- Quienes son los **adversarios:**
 - Puede ser interior/exterior al sistema de información, vendedor, empresas, etc
 - Un **vendedor** puede instalar un *rootkit* en el sistema (es un programa que se ejecuta en un ordenador con medidas para **mantener su presencia oculta** y para **impedir su eliminación**).
 - [El rootkit más conocido fue desarrollado y diseminado subrepticiamente por Sony en 2005.](https://www.schneier.com/blog/archives/2005/11/sonys_drm_rootk.html) (https://www.schneier.com/blog/archives/2005/11/sonys_drm_rootk.html). Y lo más grave que OJO que los virus se pueden aprovechar de esto.
 - Una **escucha ilegal** de un canal puede manipular las comunicaciones.

Introducción

- Los servicios de seguridad junto con los mecanismos para alcanzar esos servicios están definidos en el documento de recomendación [X.800](#).
- Esta recomendación forma parte de la [Unión Internacional de Telecomunicaciones](#), aprobada el 22 de marzo de 1991 en Ginebra, que fue elaborado por el Comité Consultivo Internacional Telegráfico y Telefónico.
- No es una especificación de implementación, sino una descripción de los servicios de seguridad junto con los mecanismos para alcanzar estos servicios.
- X.800 define en qué capa del Modelo [OSI](#) (Open System Interconnection) se deben aplicar los servicios de seguridad junto con los mecanismos o funciones que pueden ser implementados para ofrecer esos servicios.
- También se hace en este documento una recomendación de la administración de la seguridad.

Introducción

- En los Servicios de Seguridad, **Parte, Actor o Entidad**: Puede ser un usuario, proceso, sistema,
 - **Autenticación:**
 - Asegura que la identidad de un actor, entidad o entidades conectadas a un actor, entidad o entidades sea autentica y verdadera.
 - Asegura y corrobora a una entidad que la información proviene de otra entidad es autentica y verdadera.
 - **Control de acceso:**
 - Protege a una entidad contra el uso no autorizado de sus recursos.
 - Se puede aplicar a varios tipos de acceso:
 - uso de medios de comunicación, la lectura, escritura o eliminación de información y la ejecución de procesos, etc.
 - **Confidencialidad:**
 - Protege a una entidad contra la revelación deliberada o accidental de cualquier conjunto de datos a entidades no autorizadas.
 - Cuando el conjunto de datos a proteger se refiere a información propia de un individuo (dirección postal, entorno familiar, cuentas bancarias, actividades personales, etc.) generalmente se suele hablar de **privacidad**.

Introducción

➤ **Integridad:**

- Asegura que los datos almacenados en las computadoras y/o transferidos en una conexión no fueron modificados.
- Su aplicación es variable: se puede aplicar a un flujo de mensajes, un mensaje solo, o campos seleccionados dentro de un mensaje.
- Asegura que los mensajes son recibidos tal como se enviaron, sin duplicación, inserción, modificación, reorganización, o repeticiones.
- En general, proporciona protección contra toda alteración de mensaje no autorizado.

➤ **No repudio:**

- Protege contra usuarios que quieran negar falsamente que enviaran o recibieran un mensaje.
- Cuando se envía un mensaje, el receptor puede probar que el emisor de hecho, ha enviado el mensaje (Origen).
- Cuando se recibe un mensaje, el emisor puede demostrar que el receptor de hecho recibió el mensaje (Destino).

Introducción

- Mecanismos de seguridad específicos (se pueden aplicar a las diferentes capas del modelo OSI).
 - Cifrado
 - Filma Digital
 - Integridad de datos
 - Control de acceso
 -
- Mecanismos de seguridad pervasivos (no son específicos de ninguna capa del modelo OSI)
 - Funcionalidad de confianza
 - Etiquetas de seguridad
 - Detección de eventos
 - Registro de auditoría de seguridad
 - Recuperación de seguridad
 -
- En casi todos los mecanismos que proporcionan los servicios de seguridad están implicados los **algoritmos criptográficos** y los algoritmos de **integridad de datos** (algunos de estos utilizan la criptografía también).

Introducción

Foto extraída de X.800 : Security architecture for Open Systems

Interconnection for CCITT applications:

https://www.itu.int/rec/dologin_pub.asp?lang=e&id=1-REC-X.800-199103-11PDF-S&type=items

Mecanismo	Cifrado	Firma digital	Control de acceso	Integridad de datos	Intercambio de automatización	Relleno de tráfico	Control de encaminamiento	Notarización
Servicio								
Autenticación de la entidad para	S	S				S		
Autenticación del origen de los datos	S	S						
Servicio de control de acceso	.	.	S					
Confidencialidad en modo con conexión	S	.	.				S	
Confidencialidad en modo sin conexión	S	.	.				S	
Confidencialidad de campos seleccionados	S	.	.				.	
Confidencialidad del flujo de tráfico	S	.	.			S	S	
Integridad en modo con conexión con recuperación	S	.	.	S			.	
Integridad en modo con conexión sin recuperación	S	.	.	S			.	
Integridad de campos seleccionados en modo con conexión	S	.	.	S			.	
Integridad en modo sin conexión	S	S	.	S			.	
Integridad de campos seleccionados en modo sin conexión por campos selectivos	S	S	.	S	.	.	.	S
No repudio. Origen	.	S	.	S	.	.	.	S
No repudio. Entrega	.	S	.	S	.	.	.	S

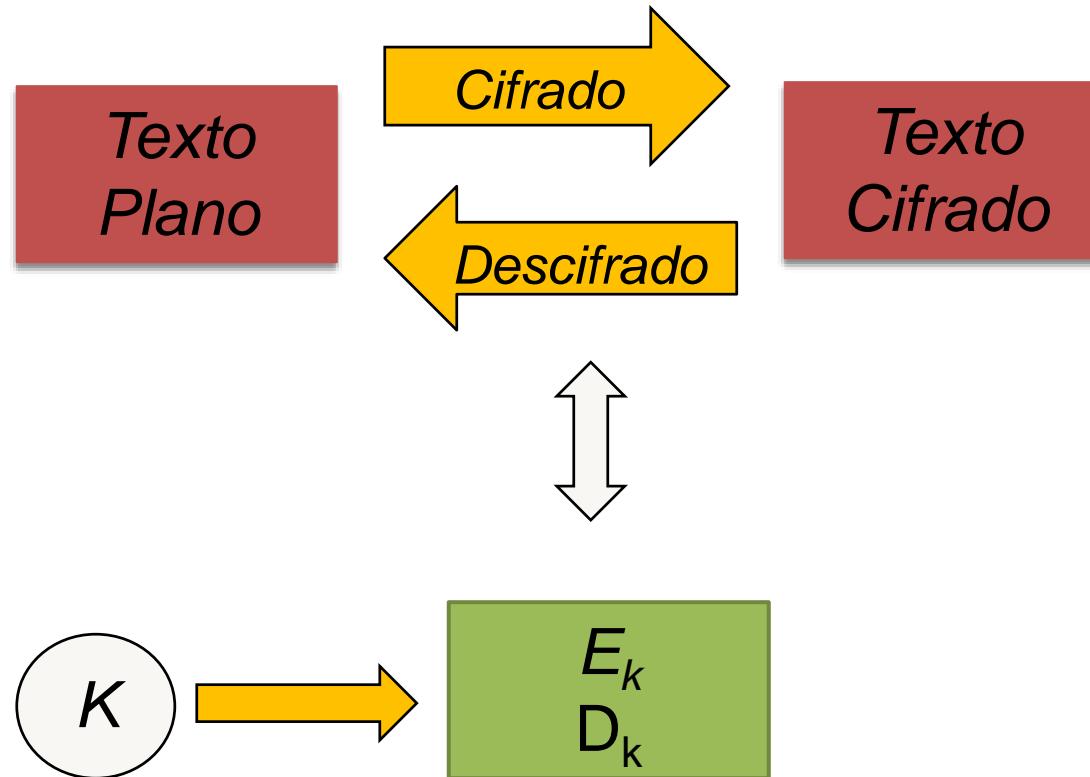
Criptografía Clásica

- Formas básicas de cifrado:
 - Utilizar un **lenguaje** inventado.
 - **Permutar** los símbolos.
 - **Sustituir** los símbolos por otros (el primer sistema es semejante a éste).
 - **Ocultar** el mensaje dentro de otro más grande (por ejemplo, en las primeras letras de los versos de una poesía).
 - **Varias** de ellas son compatibles entre sí.
 - El **algoritmo de cifrado es secreto** (en la **criptografía moderna no**).
- Por ejemplo:
 - Esparta, siglo V a.c.: Escritura sobre una cinta enrollada alrededor de un cilindro u otro cuerpo sólido alargado (**Permutación**). ([Excítalo de Lacedemonios](#)).
 - [Kamasutra, India, siglo IV a.c.](#): El número 45 de la lista es *mlecchitavi-kalpa*, que habla sobre el arte de la escritura secreta (por ejemplo emparejar las letras al azar y sustituir cada una por la correspondiente **Sustitución**).
 - [Julio César](#), siglo I a.C.:
 - **Sustitución** de caracteres latinos por griegos.
 - Utilización sistemática de cifrados, en particular por **desplazamiento** de las letras.
 - La **sustitución** de letras por otras fue utilizada sistemáticamente por la administración de los califatos abasidas en los siglos VIII y IX. También criptoanálisis.

Criptografía: Conceptos Básicos

- Texto Plano
 - Texto cifrado
 - Cifrado o encriptación
 - Descifrar o descifrado
 - Criptoanálisis
 - Criptología
 - Esteganografía
 - Seguridad Perfecta
 - Algoritmos y claves
-
- Protagonistas del proceso de cifrado:
 - Encriptador (A)
 - Descifrador (B)
 - Atacante (Mata Hari)
 - Datos y útiles:
 - Texto inicial
 - Texto cifrado
 - Algoritmo de cifrado
 - Algoritmo de descifrado (depende del de cifrado)

Criptografía: Conceptos Básicos



Criptografía: Conceptos Básicos

- **Esteganografía:**
 - Xerxes, emperador de los persas, planeó un ataque por sorpresa sobre Grecia en 480 a.c.
 - Un griego exiliado vio los preparativos y decidió enviar un mensaje de aviso escrito bajo el barniz de un par de tablas de madera.
 - Su hermana, según Heterodoxo, lo adivinó y eso permitió a los griegos organizarse y tender una emboscada a los persas en su ataque.
 - Esteganografía: Comunicación mediante ocultación.
 - Formas históricas de utilización de la esteganografía:
 - Escribiendo bajo la cabellera de un mensajero.
 - Escribiendo en un trozo de seda aplastado formando una pelota cubierta de cera tragada por un mensajero.
 - Escribiendo con un compuesto adecuado sobre la cáscara de un huevo cocido, de manera que el mensaje es visible en la clara y no en la cáscara.
 - Utilizando tinta invisible.
 - Miniaturizándolo en un punto de un documento.

Criptografía: Conceptos Básicos

➤ Algoritmos y claves:

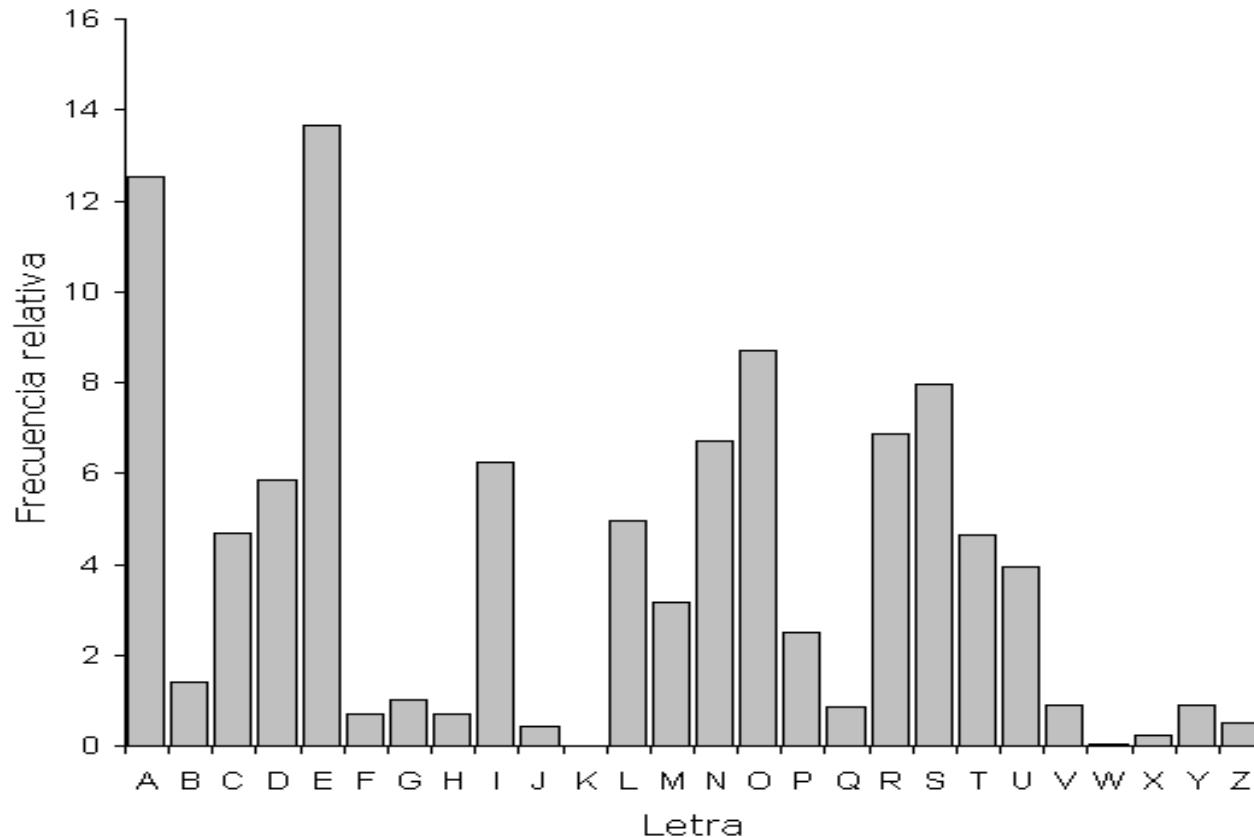
- Los métodos de cifrado y descifrado admiten **muchas variantes, correspondientes a los diferentes valores de las claves**:
- El cifrado mediante permutación del contenido del texto inicial **tiene una variante por cada permutación**.
- El cifrado mediante la **sustitución** de letras **tiene una variante por cada tabla de sustitución**.
- La utilización de algoritmos genéricos y públicos de cifrado con claves tiene ventajas:
 - A Mata Hari no le basta con descubrir el algoritmo para poder descifrar un mensaje cifrado, sino que también necesita conocer la clave con la que ha sido encriptada.
 - Si Mata Hari llega a conocer la clave con que ha sido cifrado un mensaje, y Paco y Sandra tienen conocimiento de ello, no tienen que cambiar el algoritmo, sino que les basta con utilizar otra clave diferente en mensajes subsiguientes.
 - La seguridad del algoritmo se puede evaluar por expertos, ya que el algoritmo no es secreto.
- Un peligro de utilizar algoritmos con clave es que es posible que Mata Hari **pueda descubrir la clave si llega a descubrir el mensaje inicial**.
- La idea de que los algoritmos de cifrado y descifrado deben ser públicos y la clave secreta fue propuesta por A. Kerckoffs, criptógrafo holandés del siglo XIX, y se conoce como Principio de Kerckoffs. La fuerza reside en la clave, y no en el algoritmo que sea secreto. Es decir los algoritmos criptográficos son públicos.

Criptografía: Conceptos Básicos

➤ Criptoanálisis:

- El Criptoanálisis es el arte o ciencia de descifrar mensajes cifrados.
- A lo largo de la historia los criptoanalistas han tenido una consideración a nivel social, especialmente en círculos de gran poder social.
- Es una actividad que, pese a que hoy día se realiza con el apoyo de tecnologías avanzadas, sigue manteniendo un cierto espíritu artesano (a pesar de todas las herramientas que existen para tal efecto).
- El trabajo de un criptoanalista tiene algo en común con la resolución de un jeroglífico. Se trata de utilizar diversas ideas para ir obteniendo información complementaria que permita completar el análisis por partes.
- Actualmente hay grandes empresas especializadas en criptoanálisis, cuyos clientes son a su vez grandes corporaciones de todos los sectores.
- En el siglo IX, el filósofo y científico árabe Al Kindi (su obra abarca la Medicina, la Astronomía, las Matemáticas, la Lingüística y la Música) escribió un tratado sobre El Descifrado de Mensajes Encriptados.
- En ese tratado se expone por primera vez la idea de que en un texto cifrado por sustitución de letras, **la letra más frecuente corresponde a la más frecuente del idioma inicial** y así sucesivamente.

Criptografía: Conceptos Básicos



Criptografía: Conceptos Básicos

➤ Sistemas criptográficos perfectos:

- C. Shannon estudió las propiedades que debe tener un algoritmo criptográfico óptimo.
- Un sistema de **cifrado** es **perfecto** si cualquier intento de descubrir la clave de cifrado es estadísticamente tan (in)eficiente como el intento por fuerza bruta.
- La propiedad fundamental que caracteriza a los sistemas criptográficos perfectos es que no dan ninguna información, es decir que si miramos a las propiedades de las cadenas cifradas desde un punto de vista estadístico, **no haya correlaciones** entre ellas y propiedades correspondientes de los mensajes originales.
- **Seguridad perfecta** $\leftrightarrow P_p(X | Y) = P_p(X)$
- Por ejemplo, los algoritmos de sustitución simple son muy imperfectos porque mantienen las proporciones de aparición de letras (no esconde la estadística).
- Un sistema en el que los mensajes cifrados tengan más probabilidad de tener más ‘aes’ si su mensaje original contiene más veces la letra s detrás de otra consonante no es perfecto, pues esto se puede explotar para intentar encontrar la clave o descifrar mensajes con más facilidad que por la fuerza bruta.

Criptografía: Conceptos Básicos

➤ Sistemas criptográficos perfectos:

- Shannon demostró que para que un sistema sea perfecto es necesario utilizar claves de la longitud de los mensajes que se cifran: cifrado de Vernam o One Time Pad ([OTP](#))
- Los algoritmos de cifrado más potentes, incluyendo los estándar, optimizan la dificultad de descubrimiento de la clave dentro de mantener criterios fijos de tamaño de las claves y de eficiencia de los algoritmos en sí.
- Hay formas bien establecidas de mejorar los algoritmos de encriptación. Por ejemplo una de las más clásicas es el encadenamiento de bloques:

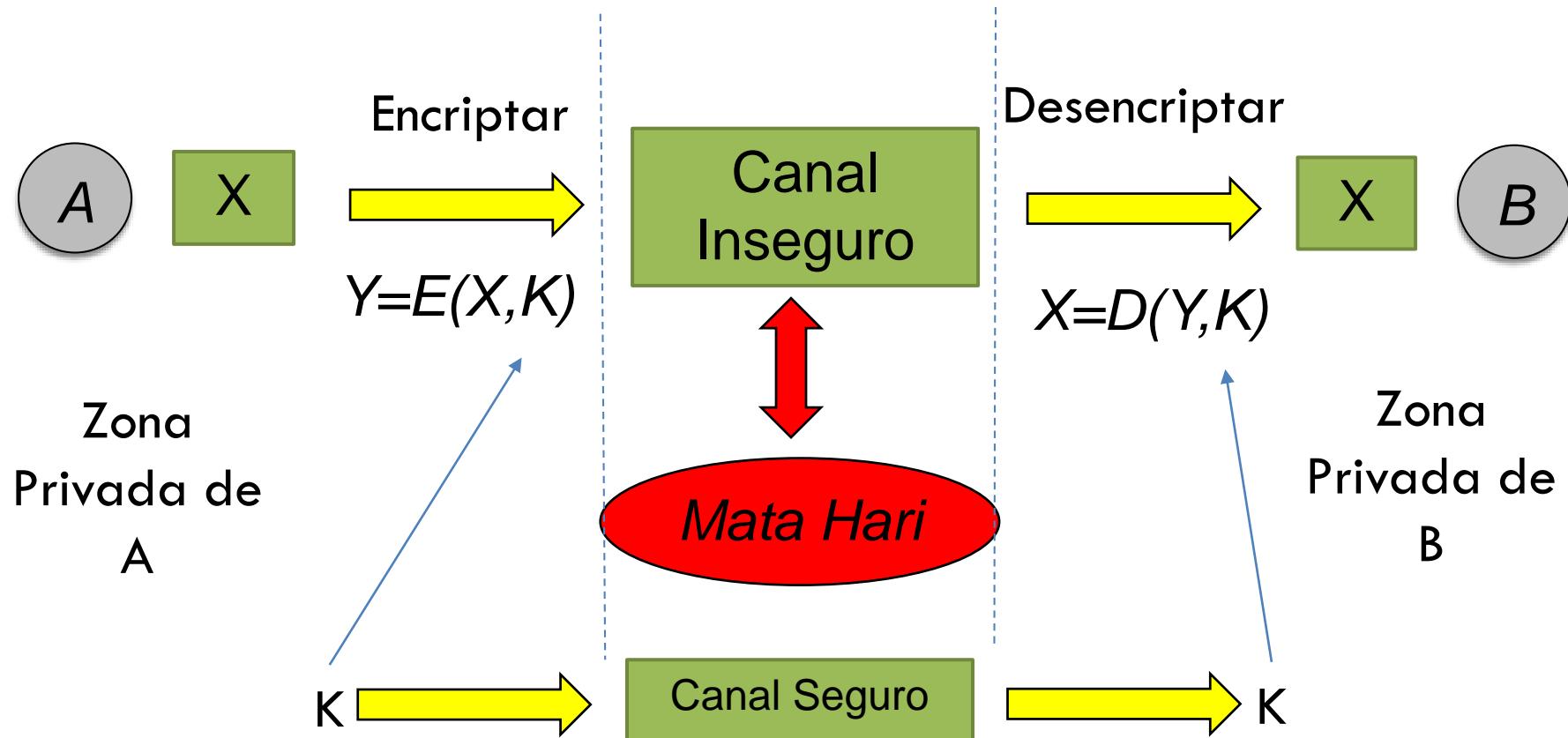
- A partir de un algoritmo de cifrado por bloques se construye otro más seguro cifrando cada bloque (excepto el primero) aplicando el primer cifrado al resultado de una operación que combine el bloque con el resultado anterior: $G(B_{i+1}) = F(C(G(B_i), B_{i+1}))$

➤ Concepto de producto de criptosistemas.

- Por ejemplo: si multiplicamos $Y = aX \text{ mod } m$ ($n \leq m$ claves, **multiplicativos inversos**) por el criptosistema $Y = X + b \text{ mod } m$ (m claves), obtenemos un criptosistema $Y = ax + b \text{ mod } m$ (m^*n claves).
 - Ojo que esto no siempre pasa, hay veces que el producto de criptosistemas no aumenta la fuerza del criptosistema resultante (**criptosistema idempotente**).

- **AHORA** pasamos a los algoritmos criptográficos de hoy en día.

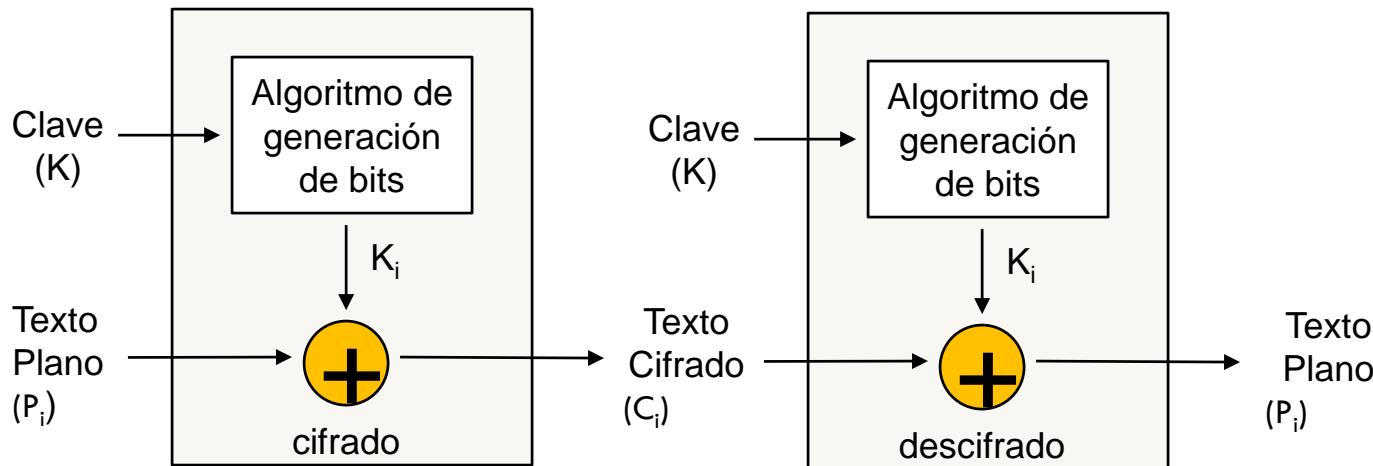
Cifrado Simétrico: Esquema General



Cifrados Simétricos: Flujo vs Bloques

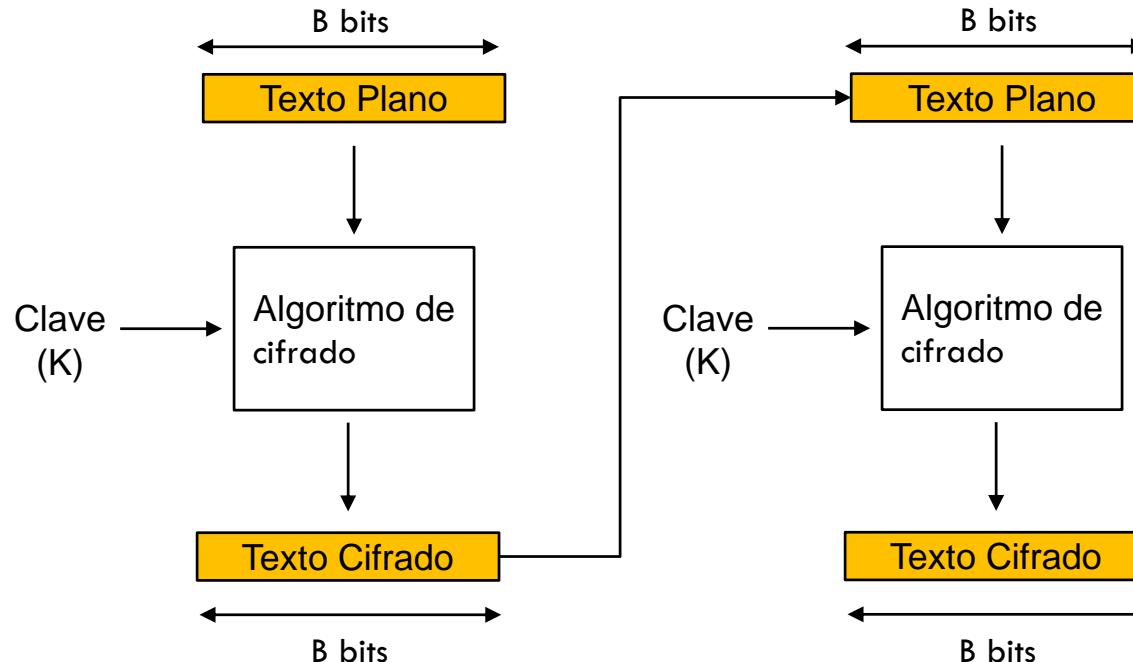
- **Cifrado de Flujo:**
 - es el que cifra un flujo de datos digitales de un bit o un byte a un tiempo (cifrados clásicos **Vigenère** de flujo de periodo constante, **Vernam**).
 - El flujo de claves (**keystream**) es tan largo como el texto a cifrar.
 - Si el flujo de clave es realmente aleatorio entonces este sistema de **cifrado es irrompible**
- **Cifrado de Bloque:**
 - Un bloque de texto claro se trata como un todo y se utiliza un algoritmo de cifrado para producir un bloque de texto cifrado de igual longitud.
 - Típicamente, un tamaño de bloque es de 64 o 128 bits.
 - Al igual que con un cifrado de flujo, los dos usuarios comparten una simétrica.
 - Al final el cifrado de bloques se puede utilizar para lograr el mismo efecto que el de flujo (modos de operación en cadena).

Cifrados Simétricos: Flujo vs Bloques



Cifrado de Flujo

Cifrados Simétricos: Flujo vs Bloques



Cifrado de Bloques

Cifrados Simétricos: Cifrado Feistel

- Algunos conceptos fundamentales:
 - Redes Feistel: [Horst Feistel, H. W. Notz, J. Lynn Smith. "Some cryptographic techniques for machine-to-machine data communications." IEEE proceedings, 63\(11\), 1545–1554, 1975.](#) (para visualizar el artículo corta y pega este link: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1451934>).
 - Cifrado Feistel
 - Reversible: las operaciones de cifrado y descifrado son idénticas
 - Tipos: Blowfish, Camellia, CAST-128, DES, FEAL, GOST 28147-89, ICE, KASUMI, LOKI97, Lucifer, MARS, MAGENTA , MISTY1, RC5, Simon, TEA, Triple DES, Twofish, XTEA, CAST-256, CLEFIA, MacGuffin, RC2, RC6, Skipjack, SMS4,
 - Confusión y difusión: Introducidos por Claude Shannon como dos bloques básicos de construcción para cualquier sistema criptográfico.
 - Frustrar el criptoanálisis basado en análisis estadístico análisis.
 - Supongamos que Mata Hari tiene un poco de conocimiento de las características estadísticas del texto claro (la distribución de frecuencias de los símbolos).
 - Si el cifrado fuera ideal, todas las estadísticas de la texto cifrado son independientes de la clave utilizada.

Cifrados Simétricos: Cifrado Feistel

- Confusión y difusión:
-
- Difusión: la estructura estadística del texto plano se disipa a lo largo de las rondas, es decir cada bit del texto plano afecta al máximo número de bits del texto cifrado, i.e. si se cambia un bit en el texto sin cifrar, deberían cambiarse la mayor cantidad posible de bits en el texto cifrado para que hubiese una buena difusión del criptograma (se obtiene a través Permutaciones).
- Confusión: busca hacer la relación entre las estadísticas de el texto cifrado y el valor de la clave de cifrado tan complejo como sea posible (se obtiene a través de Sustituciones).

Cifrados Simétricos: Cifrado Feistel

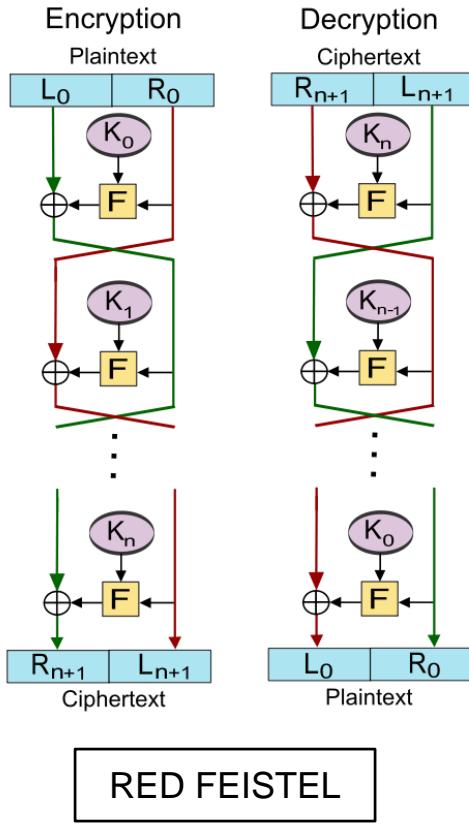
➤ Algunos conceptos fundamentales:

- La confusión difusión es la piedra angular del diseño moderno de cifrado por bloques.
- Producto de criptosistemas de sustitución y permutación.
- Rondas.
- Proceso avalancha.
- Generación de subclaves.
- Función F de cada ronda.
- Tamaño de la Clave.
- Tamaño de Bloque.
- Número de rondas.
- En contraste a todas estas redes Feistel están las Redes de sustitución-permutación SPNs, (AES, 3-Way, SAFER, SHARK, Square, etc.):
 - Intrínsecamente, son más paralelizables y la función inversa puede ser diferente a la de cifrado, en contraste con las redes Feistel.
 - Todos los conceptos fundamentales anteriores se suele cumplir también en estas estructuras de sustitución y permutación criptográficas.

Cifrados Simétricos: Cifrado Feistel

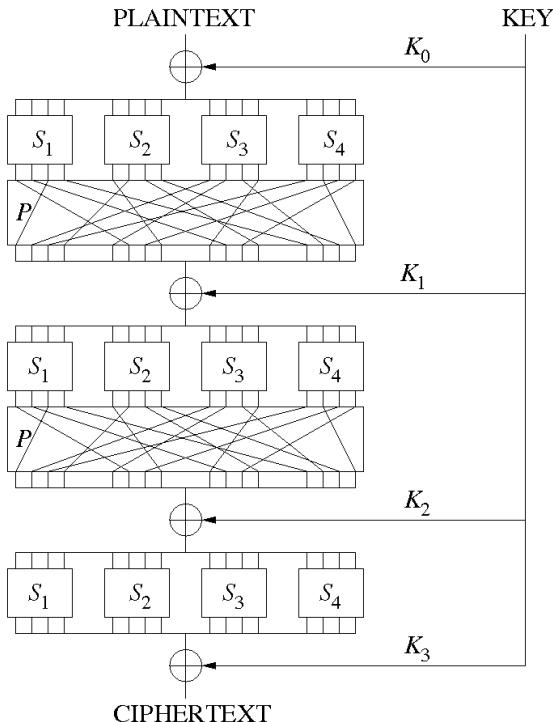
Imagen extraída de
http://commons.wikimedia.org/wiki/File:Feistel_cipher_diagram_en.svg

En una red Feistel (del criptógrafo alemán Horst Feistel de IBM), la entrada se divide en dos bloques (L_0 y R_0) que interactúan entre sí.



RED FEISTEL

En una SPN, la entrada se divide en múltiples bloques pequeños, aplicados a un S-box (sustitución), entonces las salidas de los bits se mezclan (permutación). La adición clave puede ocurrir antes o después de estas dos operaciones.



RED SP

Imagen extraída de
<https://upload.wikimedia.org/wikipedia/commons/c/cd/SubstitutionPermutationNetwork2.png>

Cifrados Simétricos: Cifrado Feistel

Imagen extraída de: Horst Feistel, H., W. Notz, J. Lynn Smith. "Some cryptographic techniques for machine-to-machine data communications." IEEE proceedings, 63(11), 1545–1554, 1975.

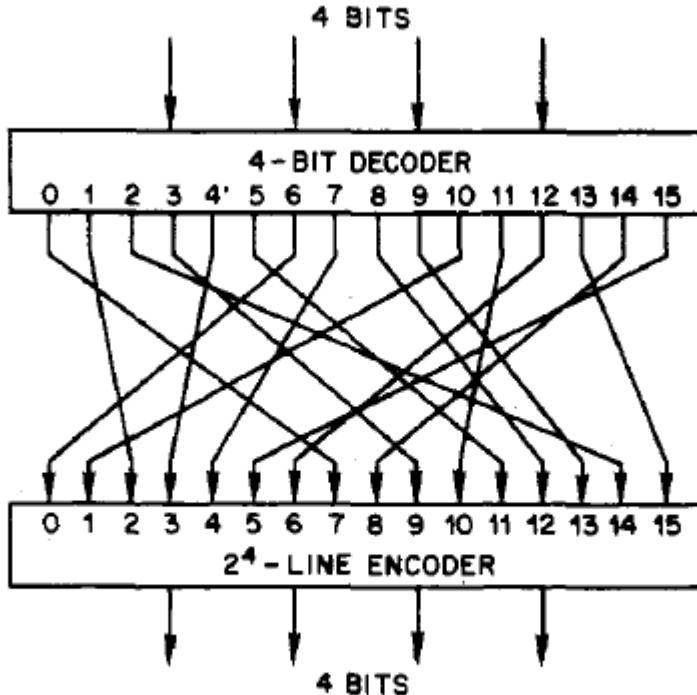
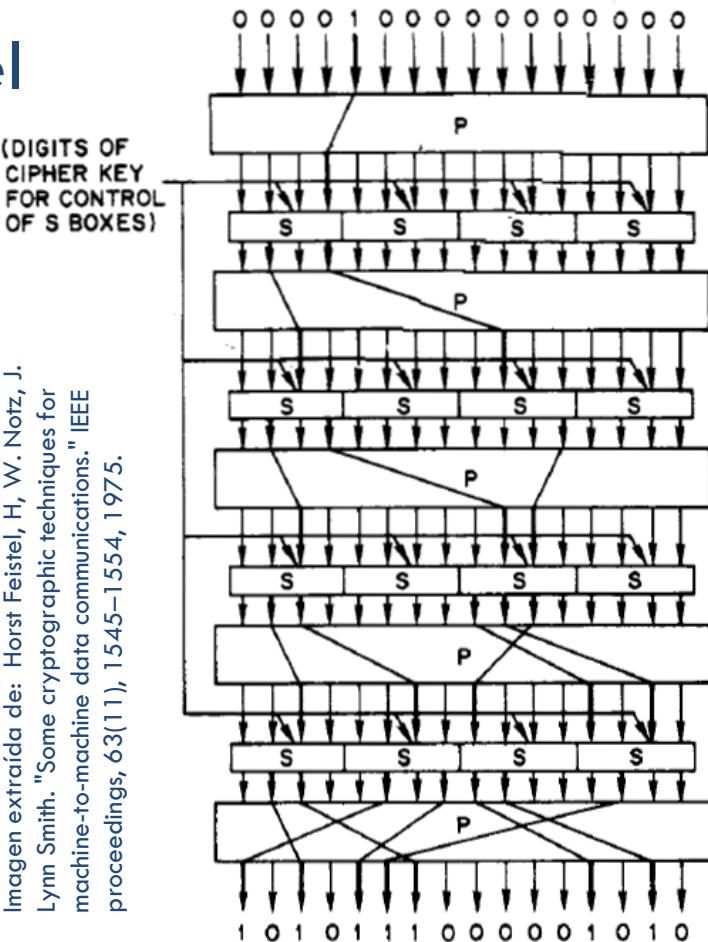


Fig. 1. General n -bit- n -bit block substitution, S (here shown for $n = 4$, connected to produce a particular nonsingular transformation).

- Una sustitución se puede ver como esta caja.
- En este caso de 4 bits.

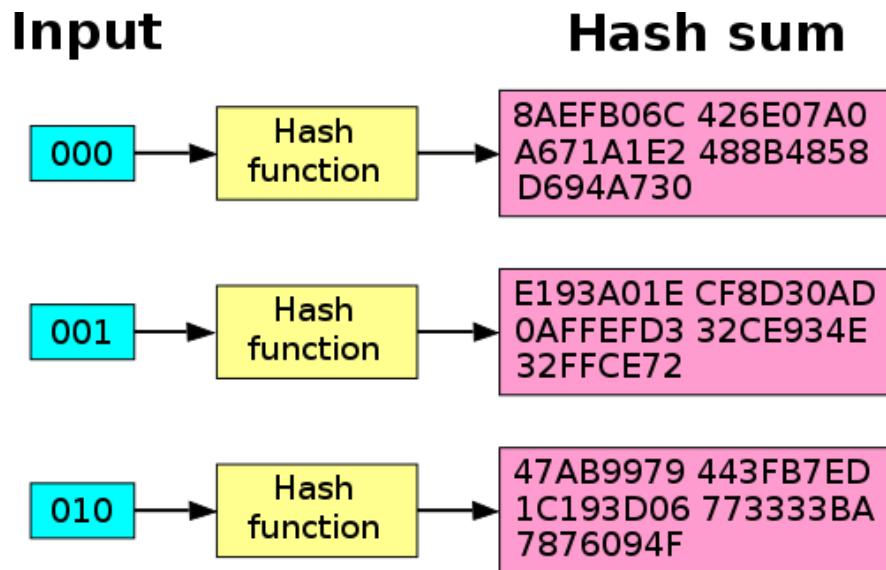
Cifrados Simétricos: Cifrado Feistel

- Este tipo de cifrado ejecuta un producto de criptosistemas de sustitución y permutación.
- A través de la generación de las rondas se produce un fenómeno de avalancha.
- Esto es un principio de diseño de cifrado de bloques.



Cifrados Simétricos: Cifrado Feistel

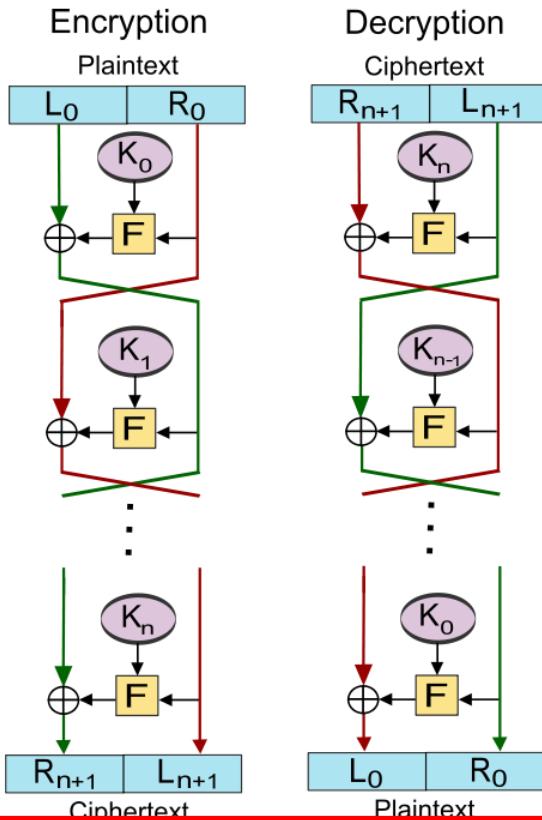
- El efecto avalancha se utiliza en más funciones, como Hash, no solo en el DES.
- Es un principio fundamental de seguridad.
- Por ejemplo en SHA-1, con un solo cambio de bit la salida es completamente diferente.
- Este principio se consigue mediante, fundamentalmente mediante:



- Strict avalanche criterion (SAC): Si se complementa un solo bit de entrada, cada uno de los bits de salida cambia con un 50 % de probabilidad.
- Bit independence criterion (BIC): Los bits de salida j y k deben cambiar de manera independiente cuando un solo bit de entrada i se complementa, para todo i, j y k .

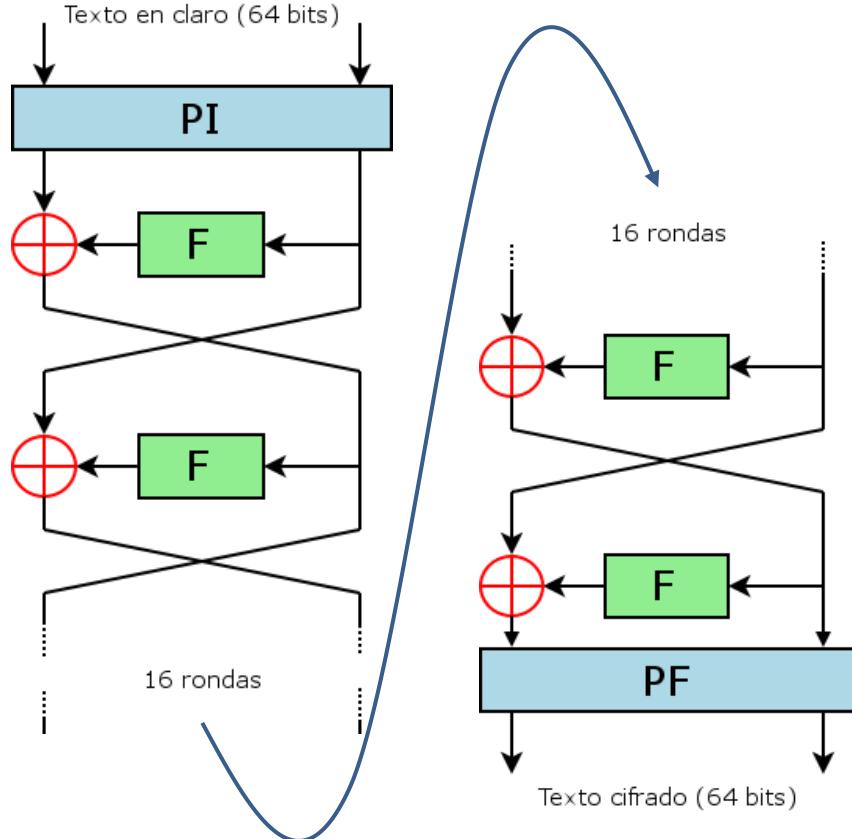
Cifrados Simétricos: Cifrado Feistel

Imagen extraída de
http://commons.wikimedia.org/wiki/File:Feistel_cipher_diagram_en.svg



- Los cífrados de tipo Feistel son siempre de esta forma.
- Un cífrado de tipo Feistel clásico es el DES.

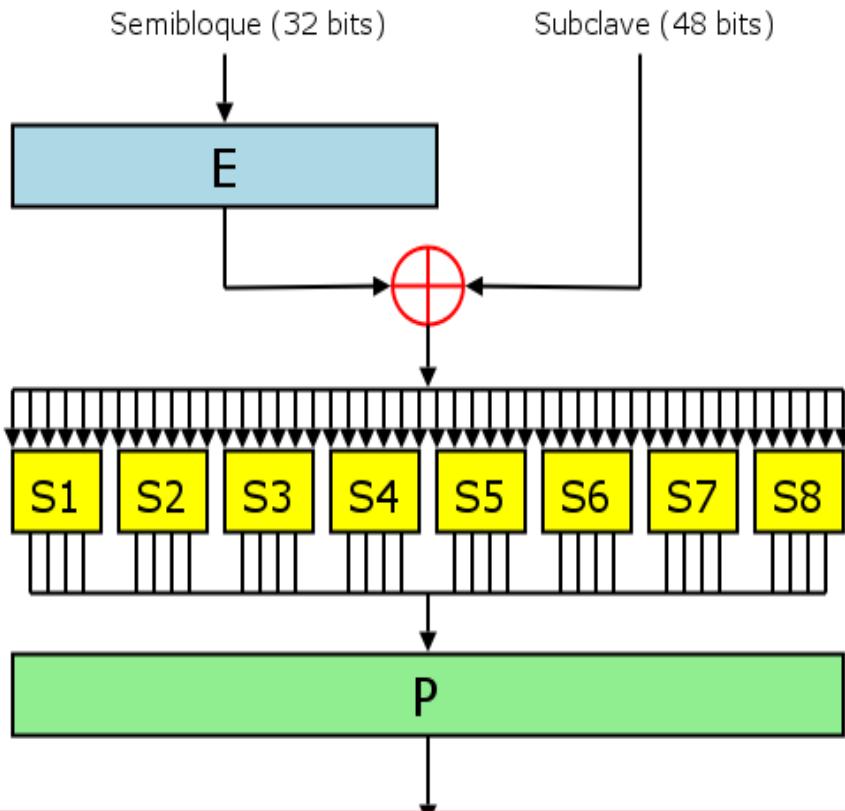
Cifrados Simétricos: DES



- Hasta el AES en 2001, el Data Encryption Standard (DES, en 1977 por el NIST, [FIPS PUB 46](#)) fue intensamente utilizado.
- Los datos se cifran en bloques de 64 bits usando una clave de 56 bits (los bits de paridad están fuera).
- Los mismos pasos, con la misma clave invertida, se utilizan para el descifrado.
- En 1999, el NIST publicó el TDES o ([FIPS PUB 46-3](#)).
- Porque 16 Rondas: se observa que para 16 rondas un ataque diferencial es un poco menos eficiente que la fuerza bruta. El criptoanálisis diferencial requiere $2^{55.1}$ operaciones, mientras que la fuerza bruta requiere 2^{55} .

Cifrados Simétricos: Función F del DES

Imagen extraída de <http://commons.wikimedia.org/wiki/File:DES-funcion-f.png>



- El corazón es la función F no lineal cumpliendo los principios SAC y BIC:
 - ▣ SAC: Establece que cualquier bit de salida j de una caja S debe cambiar con probabilidad $1/2$, cuando un solo bit de entrada i se invierte para todo i, j .
 - ▣ BIC: Establece que los bits de salida j y k deben cambiar de forma independiente cuando un solo bit de entrada i se invierte para todo i, j , y k .

Cifrados Simétricos: Subclaves DES

Imagen extraída de
<http://commons.wikimedia.org/wiki/File:DES-key-schedule.png>

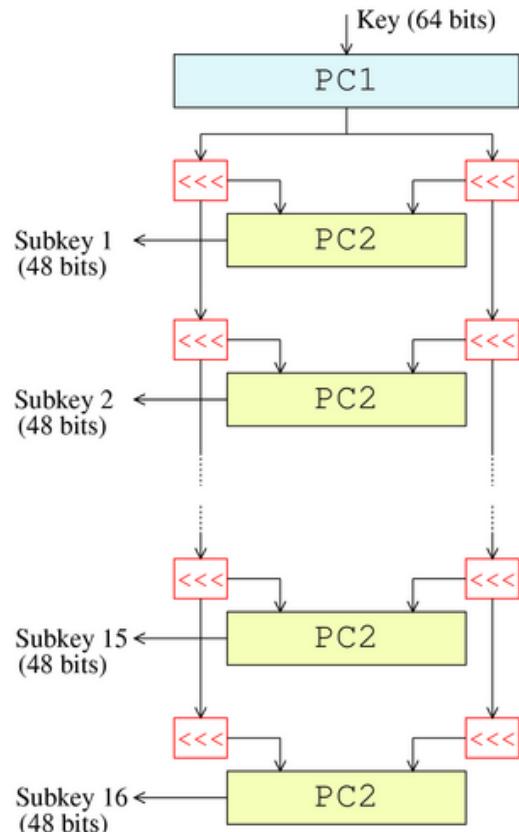
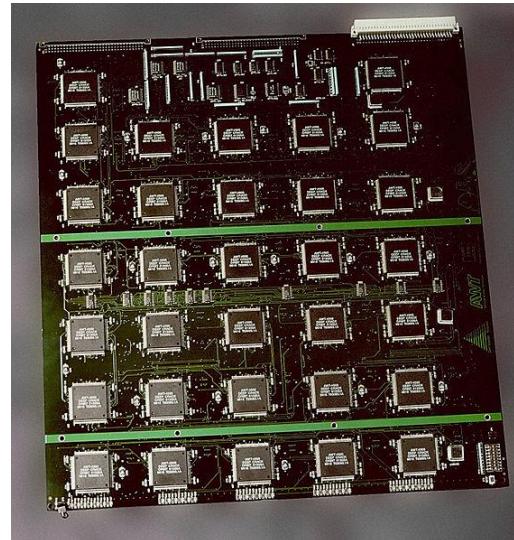


Imagen extraída de
<http://commons.wikimedia.org/wiki/File:Board300.jpg>



DEEP CRACK, The
DES challenges

- Se generan 16 claves con una clave inicial.
- En el descifrado las claves se introducen en orden reverso.

Cifrados Simétricos: DES, versiones

➤ El DES doble:

- $C = EK2(EK1(P))$
- $P = DK1(DK2(C))$

➤ Se ataca en el medio:

$$E_{K1}(P) = D_{K2}(C)$$

➤ El TDES o 3DES:

- $C = EK3(DK2(EK1(P)))$
- $P = DK1(EK2(DK3(C)))$

➤ Opciones de claves:

- Las tres claves independientes ($3 \times 56 = 168$ bits).
- $K3 = K1$ ($2 \times 56 = 112$ bits).
- $K1 = K2 = K3$ (56 bits, por compatibilidad con el antiguo DES).

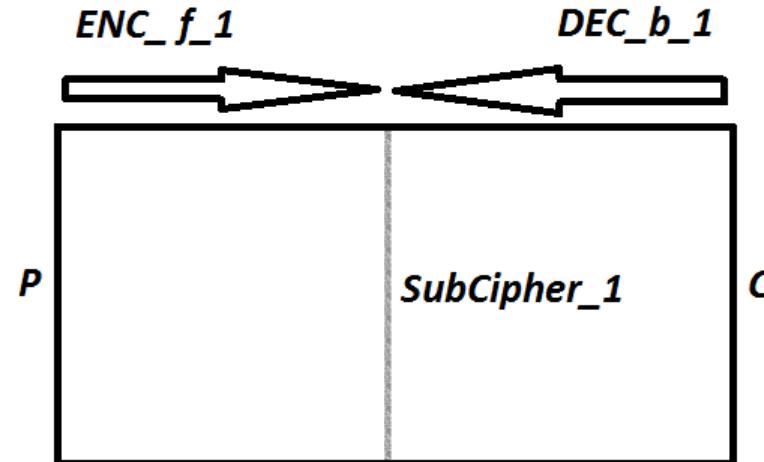


Imagen extraída de
http://en.wikipedia.org/wiki/File:1D_MITMNEW.png

Cifrados Simétricos: AES

- El NIST propone un concurso para un nuevo estándar en 1997, ya que se estaban probando las vulnerabilidades del DES mediante ataque por fuerza bruta en 1999 ([distributed.net](#) + [Electronic Frontier Foundation, EFF](#)).
 - Petición del concurso: dominio público, simétrico y 128 bits, variabilidad de tamaños de claves, eficiente en SW y HW
- En 1999 Electronic Frontier Foundation (EFF) con el Deep Crack quedó demostrada el ataque por fuerza bruta del DES.
- En la primera fase del concurso se admiten 15 algoritmos.
- En la segunda fase se seleccionan 5:
 - MARS
 - RC6
 - RIJNDAEL
 - SERPENT
 - TWOFRISH
- MARS: 13 votos, RC6: 23 votos, RIJNDAEL: 86 votos, SERPENT: 59 votos, TWOFRISH: 31 votos.
- Gana el RIJNDAEL que será el AES hasta nuestros días (en noviembre de 2001 se publicó [FIPS 197](#)).

Cifrados Simétricos: AES

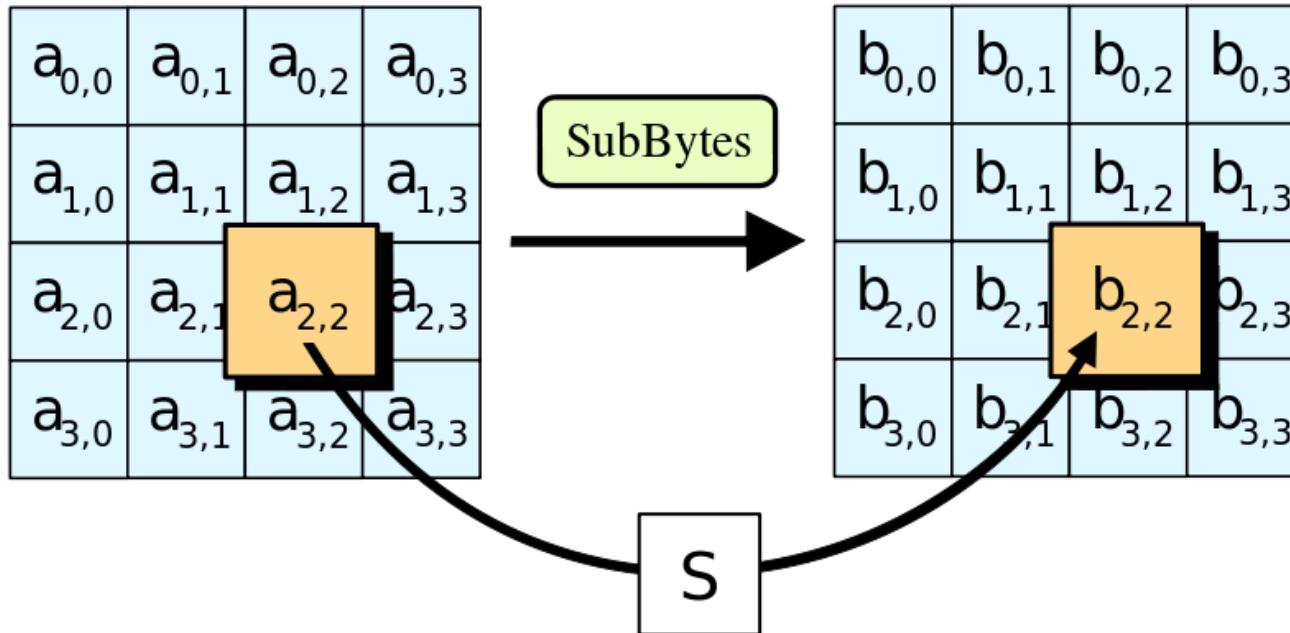
- Este algoritmo es el nuevo estándar de cifrado simétrico desde 2001.
- No es está basado en el cifrado Feistel, sino en redes de sustitución y permutación.

Tamaño de clave (W/B/b)	4/16/128	6/24/192	8/32/256
Tamaño de bloque (W/B/b)	4/16/128	4/16/128	4/16/128
Numero de Rondas	10	12	14
Clave expandida (W/B)	44/176	52/208	60/240

- El bloque sufre cuatro transformaciones en cada ronda:
 - SubBytes
 - ShiftRows
 - MixColumns
 - AddRoundKey

Cifrados Simétricos: AES-SubBytes

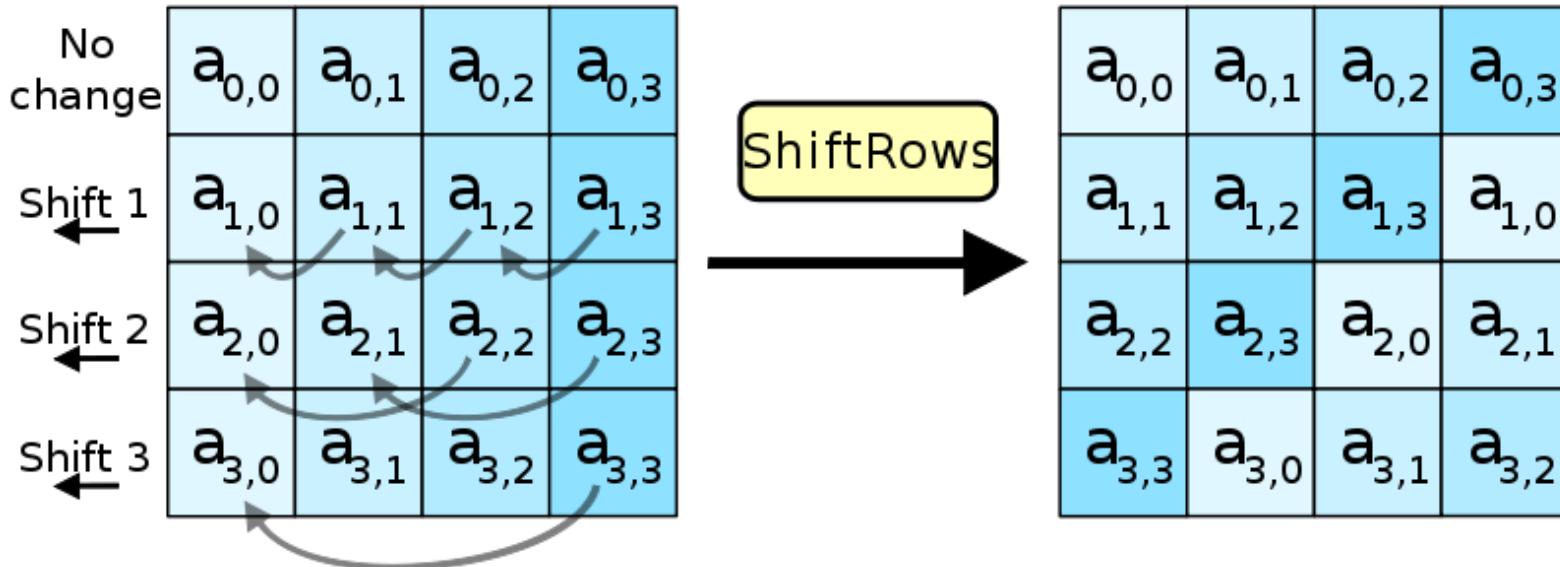
Imagen extraída de
[http://commons.wikimedia.org/wiki/
File:AES-SubBytes.svg](http://commons.wikimedia.org/wiki/File:AES-SubBytes.svg)



- State: transformaciones que sufre el bloque a lo largo del cifrado.
- Caja de sustitución por bytes del bloque.
- Operaciones matemáticas basadas en campos finitos (polinomios de Galois)

Cifrados Simétricos: AES-ShiftRows

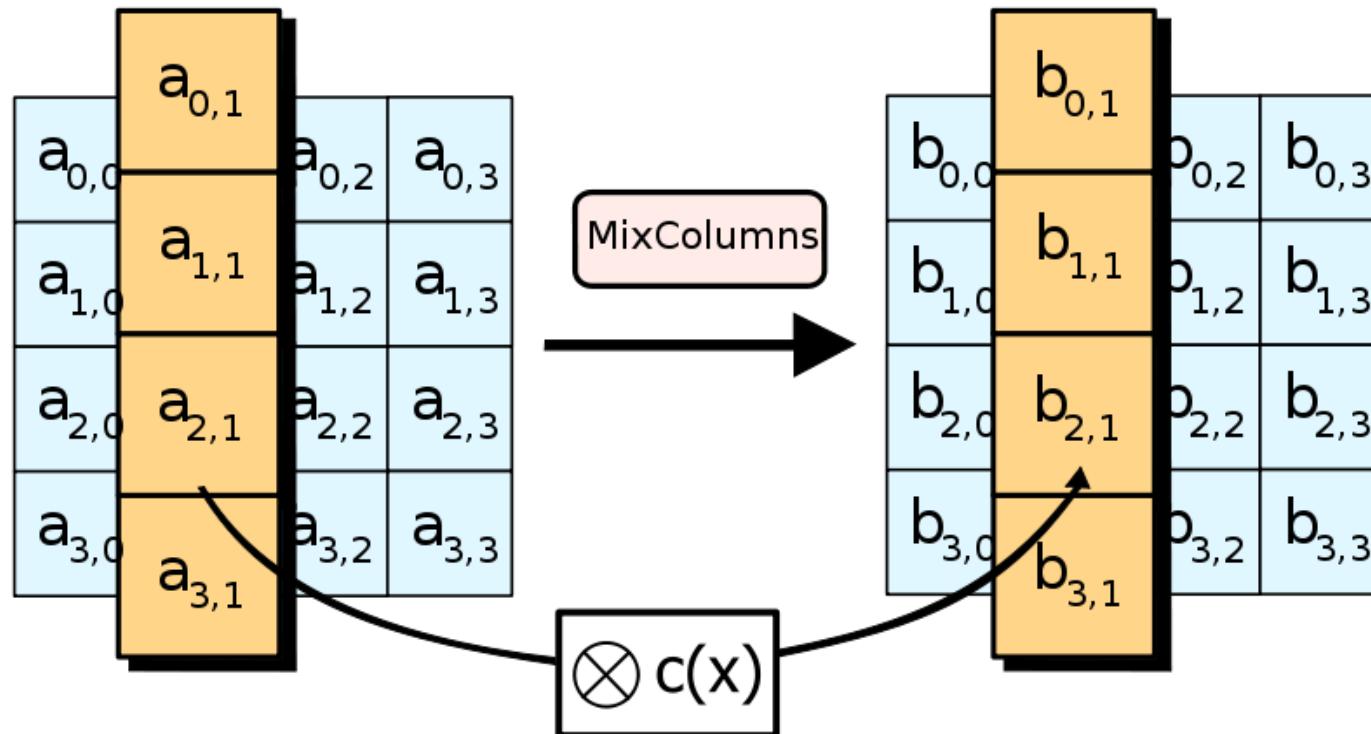
Imagen extraída de
<http://commons.wikimedia.org/wiki/File:AES-ShiftRows.svg>



- Los bytes del state son rotados por filas, es decir se descolocan los bytes en el state, pero no se cambia su valor.

Cifrados Simétricos: AES-MixColumns

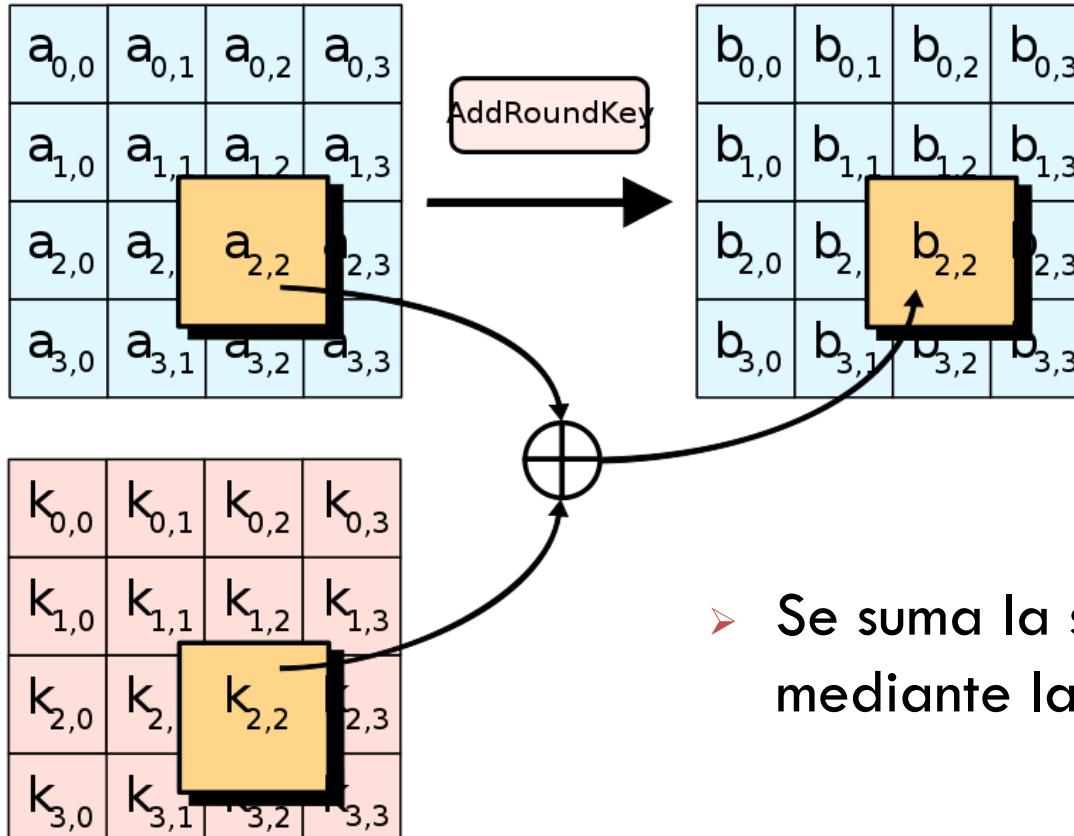
Imagen extraída de
<http://commons.wikimedia.org/wiki/File:AES-MixColumns.svg>



- Cada columna se multiplica por valor fijo o polinomio constante (polinomio de Galois), mezclándose así todos los bytes por columnas.

Cifrados Simétricos: AES-AddRoundKey

Imagen extraída de
https://commons.wikimedia.org/wiki/File:AES_AddRoundKey.svg

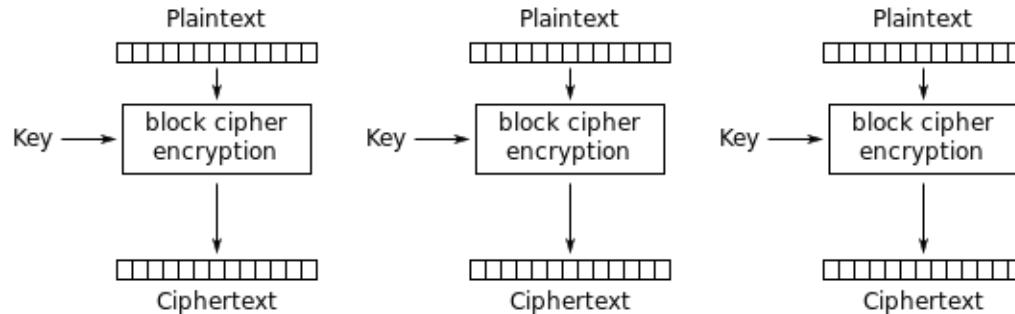


➤ Se suma la subclave mediante la función XOR.

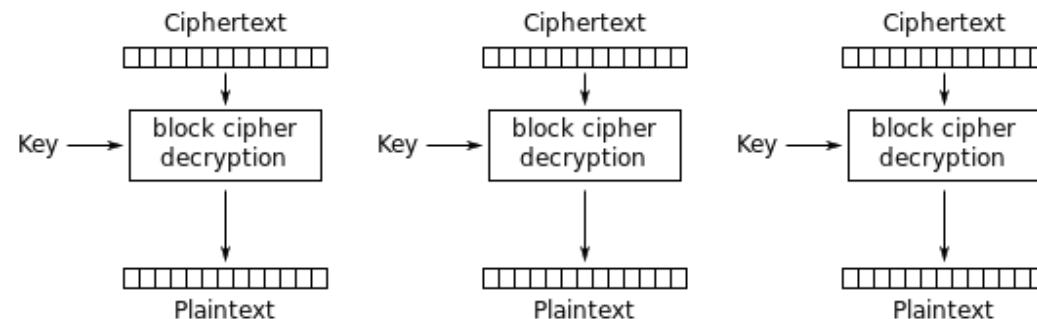
Cifrados Simétricos: AES-Pseudocódigo

- AES-CIFRA (State, K)
 - Expansión de la clave inicial
 - AddRoundKey
 - for i=1 to Rondas-1
 - RondaAES
 - RondaFinalAES
- RondaAES
 - SubBytes
 - ShiftRows
 - MixColumns
 - AddRoundKey
- RondaFinalAES
 - SubBytes
 - ShiftRows
 - AddRoundKey
- Para cada función del AES existe su inversa bien definida:
 - InvSubBytes
 - InvShiftRows
 - InvMixColumns
 - InvAddRoundKey
- Por lo tanto en el descifrado se realiza el proceso inverso con las funciones inversas.
- Referencias interesantes:
 - FIPS 197
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
 - Gráfico animado AES
<http://www.formaestudio.com/rijndaelinspector/>
 - Códigos de diferentes cifrados:
http://embeddedsw.net/Cipher_Reference_Home.html

Cifrados Simétricos por bloques: Modos de Operación



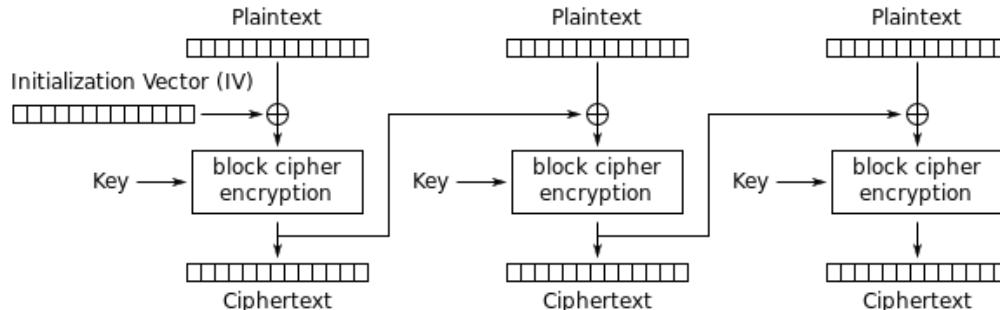
Electronic Codebook (ECB) mode encryption



Electronic Codebook (ECB) mode decryption

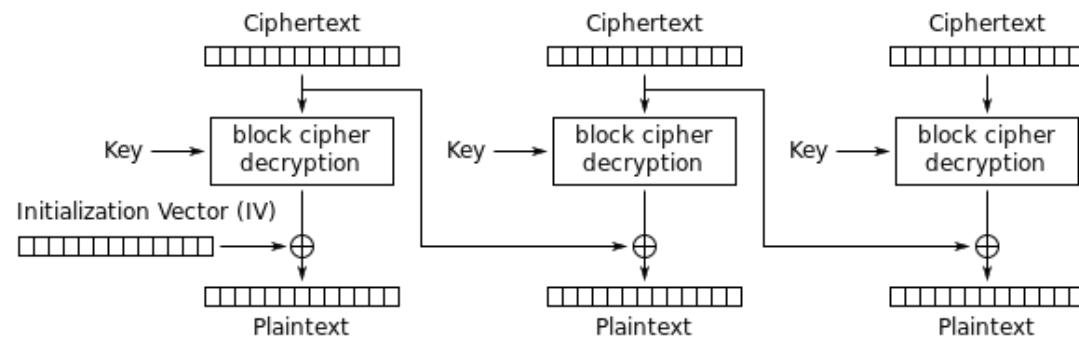
Imagen extraída de
http://en.wikipedia.org/wiki/File:ECB_encryption.svg

Cifrados Simétricos por bloques: Modos de Operación



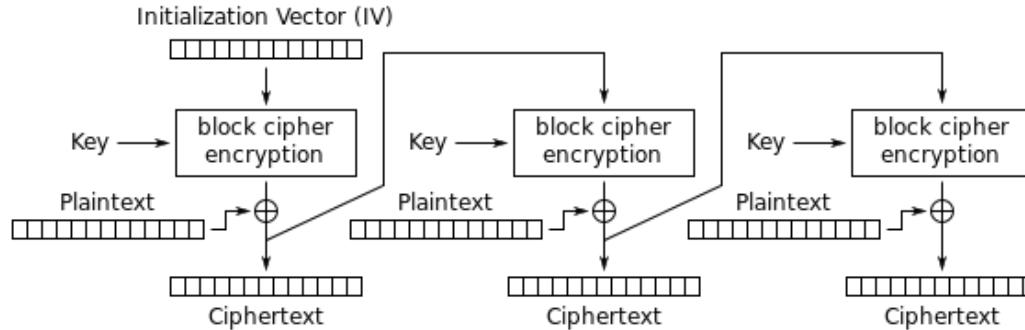
Cipher Block Chaining (CBC) mode encryption

[Volver](#)

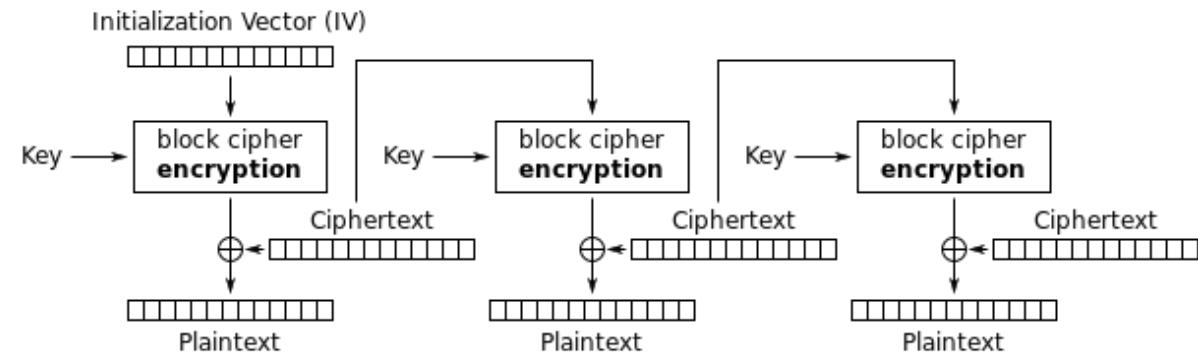


Cipher Block Chaining (CBC) mode decryption

Cifrados Simétricos por bloques: Modos de Operación



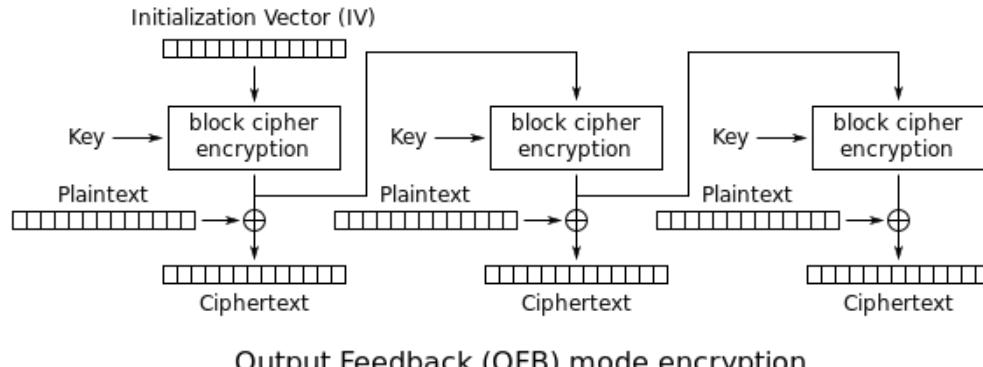
Cipher Feedback (CFB) mode encryption



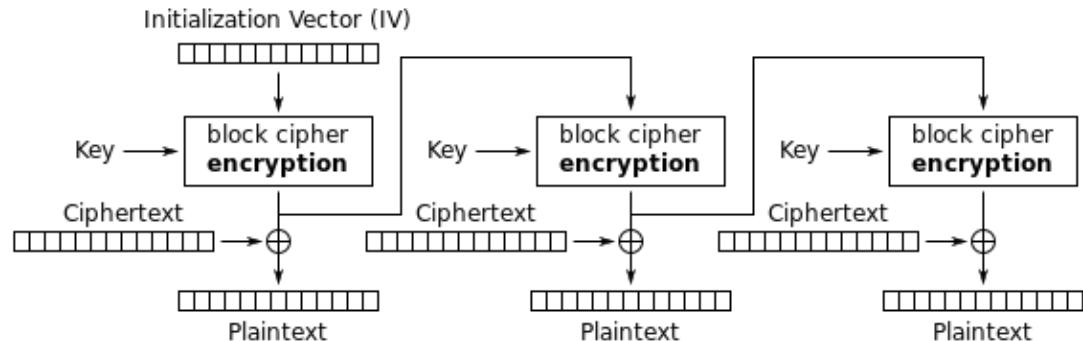
Cipher Feedback (CFB) mode decryption

Cifrados Simétricos por bloques: Modos de Operación

Imagen extraída de
http://en.wikipedia.org/wiki/File:OFB_encryption.svg

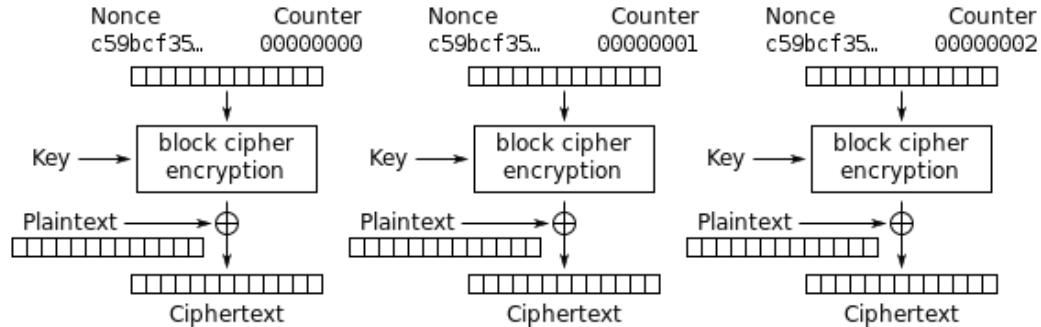


Output Feedback (OFB) mode encryption

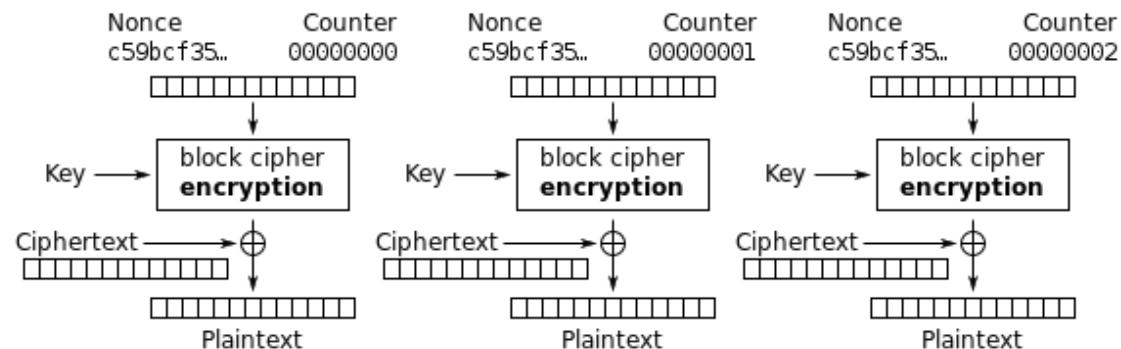


Output Feedback (OFB) mode decryption

Cifrados Simétricos por bloques: Modos de Operación



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Cifrados Simétricos: Cifrados de Flujo

Generación de Números Aleatorios

- La generación de números aleatorios en seguridad y criptografía es fundamental:
 - Distribución de claves.
 - Generación de claves en una sesión.
 - Generación de claves en criptografía pública.
 - Generación del flujo de claves en cifrados de flujo simétricos.
- Generación de números aleatorios en seguridad y criptografía sigue los principios fundamentales:
 - La aleatoriedad:
 - Distribución uniforme: la distribución de los bits en la secuencia debe ser uniforme, es decir, la frecuencia de aparición de unos y ceros debe ser aproximadamente igual.
 - Independencia: es decir que no exista una sub-secuencia en la secuencia completa de bits que se pueda inferir de otras sub-secuencias.
 - La impredicibilidad: no es sólo la exigencia que la secuencia de números sean estadísticamente distribuidos al azar, sino que esta secuencia es impredecible (recordar que las secuencias de números aleatorios las generan algoritmos deterministas). Tiene que cumplirse hacia delante y atrás.

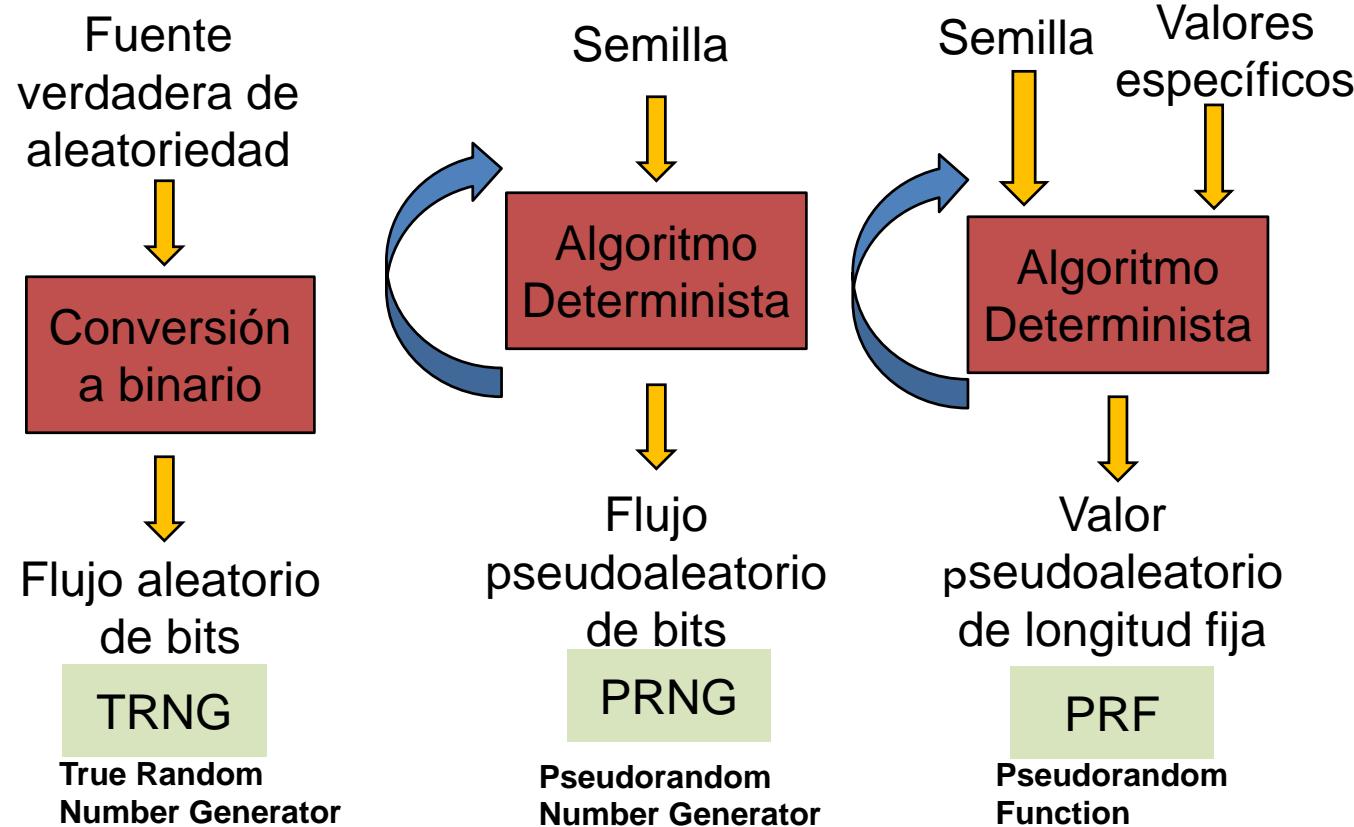
Cifrados Simétricos: Cifrados de Flujo

Generación de Números Aleatorios

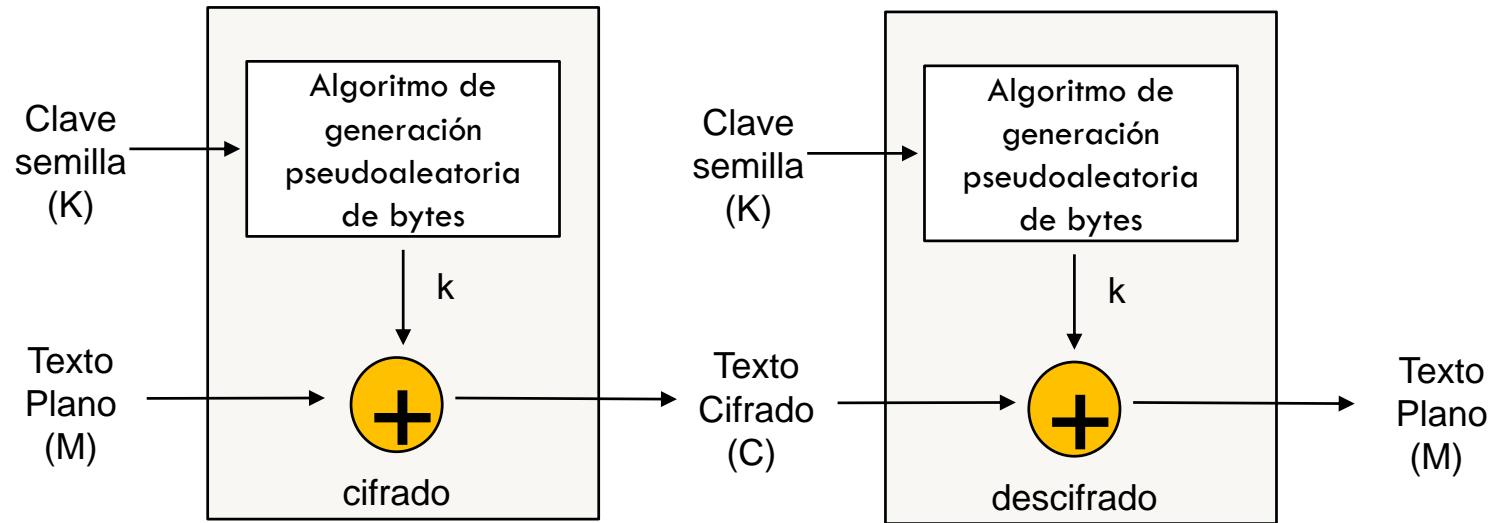
- Realmente lo que tenemos en el mundo real son números pseudoaleatorios:
 - Las aplicaciones criptográficas típicamente utilizan algoritmos para la generación de números aleatorios.
 - Estos son deterministas y por tanto producen secuencias de números que no son completamente estadísticamente aleatorios.
 - Si el algoritmo es bueno, entonces las secuencias resultantes pasan ciertas pruebas de aleatoriedad y estas se conocen como secuencias de números pseudoaleatorios.
- Generadores de números pseudoaleatorios (Número o Función):
 - Entrada: Semilla (normalmente es un número aleatorio real).
 - Salida: Secuencia de bits pseudoaleatorios con un algoritmo determinista.
 - El flujo de bits de salida se determina únicamente por el valor de entrada o valores, por lo que un adversario que conoce el algoritmo y la semilla puede reproducir todo el flujo de bits.
- Un número aleatorio real toma como entrada una fuente que es efectivamente al azar, la fuente se refiere a menudo como una fuente de entropía.

Cifrados Simétricos: Cifrados de Flujo

Generación de Números Aleatorios



Cifrados Simétricos: Cifrados de Flujo-RC4



- Cifrado de Flujo (en cada paso se cifra un Byte para el RC4)

Cifrados Simétricos: Cifrados de Flujo-RC4 (principios diseño)

- El cifrado de flujo es similar a [OTP](#), la diferencia es que utiliza un flujo de números pseudoaleatorios como clave y que es orientado a bytes.
- Algunos principios de diseño:
 - La secuencia de la clave para el cifrado debe tener un período de gran tamaño (todo generador pseudoaleatorio utiliza una función que produce un flujo de bits que finalmente repite).
 - El flujo de claves debe aproximarse a las propiedades de un número aleatorio verdadero. Por ejemplo aproximadamente igual número de 1s y 0s. Si la cadena de claves se trata como una secuencia de bytes, entonces todos los posibles valores de 256 bytes debe uniforme (Seguridad Perfecta).
 - La salida del generador de números pseudoaleatorios está condicionada sobre el valor de la clave de entrada o semilla. Esta debe ser lo suficientemente grande frente a ataques de fuerza bruta.
- Con los apropiados diseños de los métodos de generación de números aleatorios, los cifrados de flujo pueden ser tan robustos como los de bloque y más rápidos.

Cifrados Simétricos: Cifrados de Flujo-RC4 (principios diseño)

- La desventaja en este tipo de cifrados de flujo es que reusar claves no es posible ya que el XOR de dos textos cifrados es el XOR de los planos.
- Por lo tanto es más fácil criptoanalizar:
 - Dawson, E., and Nielsen, L. "Automated Cryptoanalysis of XOR Plaintext Strings." Cryptologia, 1996.
- Es decir $Y_1 \text{ xor } Y_2$ es lo mismo que $(X_1 \text{ xor } K) \text{ xor } (X_2 \text{ xor } K) = X_1 \text{ xor } X_2$ (recordar que $A \text{ xor } A = 0$).
- Esto **no** ocurre con los cifrados por bloque.
- **A efectos prácticos como RC4 no toma un nonce o un IV, si es necesario cifrar varios mensajes con la misma clave de largo plazo, debe crearse un nonce independiente distinto para cada mensaje y una clave de corto plazo se debe derivar de la combinación de la clave de largo plazo y el nonce.**
- Debido al algoritmo de programación de clave débil de RC4 (key scheduling algorithm) se podría sacar una parte de la clave, es decir la parte que se repite de la combinación que es la clave de largo plazo.
- Por lo tanto la combinación debe llevarse a cabo con una función compleja (por ejemplo, un hash criptográfico) y no simplemente concatenando clave y nonce.
- Es decir se debe generar confusión en la clave de largo plazo y el nonce a través del hash.
- **Un nonce es un número aleatorio que solo se utiliza una única vez.**

Cifrados Simétricos: Cifrados de Flujo-RC4

- El ejemplo típico de cifrado por flujo es el RC4.
- Diseñado en 1987 por Ron Rivest para RSA Security (división de seguridad de EMC Corporation).
- ARCFOUR, ARC4 o Alleged-RC4 en su implementación no oficial, ya que RSA Security nunca ha liberado el algoritmo de su RC4.
 - Inicialmente el algoritmo era un secreto registrado, pero en septiembre de 1994 una descripción del algoritmo fue posteada anónimamente en una lista de correo de Cypherpunks.
- Clave variable con operaciones orientadas a bytes.
- Basado en el uso de permutaciones aleatorias.
- Ocho a dieciséis operaciones de código máquina por byte de salida.

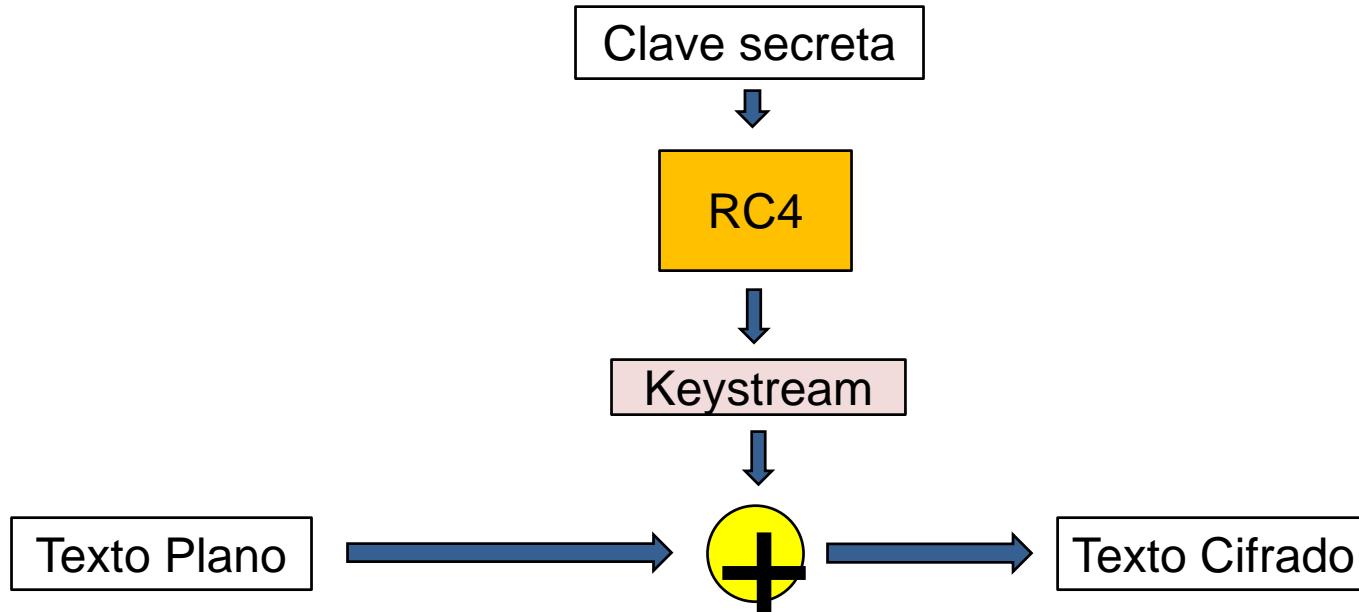
Cifrados Simétricos: Cifrados de Flujo-RC4

- Utilizado en el Secure Sockets Layer / Transport Layer Security (SSL /TLS) que se han definido para la comunicación entre navegadores y servidores Web de manera segura.
- A partir de 2015, se especula que algunas agencias criptológicas, como la NSA, pueden poseer la capacidad de romper RC4 cuando se utiliza en el protocolo TLS.
- Por tanto Internet Engineering Task Force ha publicado el RFC 7465 para prohibir el uso de RC4 en TLS, al igual que Mozilla y Microsoft han emitido recomendaciones similares, y la versión Chrome 48 elimina completamente el soporte RC4 para TLS.
- Así ahora se aconseja el TLS 1.2 y mejor 1.3 que sustituye el RC4 por el AES-GCM, o AES-GCM-SIV.
- También se utiliza en la Wired Equivalent Privacy (WEP) y el protocolo WiFi Protected Access nuevo protocolo (WPA) que forman parte del estándar de LAN inalámbrica IEEE 802.11.

Cifrados Simétricos: Cifrados de Flujo-RC4

- Enseñar con <https://vpn2.uam.es/> que seguridad tiene (tecla F12 y menú Security) y enseñar otro ejemplos de otras páginas:
 - <https://www.bancosantander.es/>
 - <https://www.bbva.es/particulares/index.jsp>
 - <https://www.citapreviadnie.es/>
 - <https://www.google.es/>
- Su seguridad está sujeta a gran debate:
 - <http://www.isg.rhul.ac.uk/tls/>
 - <http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-2566>
 - <http://www.isg.rhul.ac.uk/tls/RC4biases.pdf>
 - <http://www.rc4nomore.com/>
 - [Google Chrome 48 elimina el soporte para el cifrado RC4 \(21 Enero 2016\).](#)
 - https://en.wikipedia.org/wiki/Transport_Layer_Security#RC4_attacks

Cifrados Simétricos: Cifrados de Flujo-RC4



Cifrados Simétricos: Cifrados de Flujo-RC4

- Una clave variable o semilla de 1 a 256 bytes se utiliza para inicializar un vector estado S de 256 bytes (típicamente entre 5 y 16 bytes, o 40 a 128 bits).
- El vector de estado S, en todo momento, contiene una permutación de todos los números de 8 bits del 0 al 255.
- Para el cifrado y descifrado se genera un byte de una de las posibles entradas de S.
- La inicialización de S consiste en estos dos procesos simples (llamada *Key Scheduling Algorithm*):

/* Initialization */

for i = 0 to 255 do

 S[i] = i;

 T[i] = K[i mod keylen];

Clave
entre 5 y
16 bytes

/* Initial Permutation of S */

j = 0;

for i = 0 to 255 do

 j = (j + S[i] + T[i]) mod 256;

Swap (S[i], S[j]);

- El vector temporal T, tiene la semilla repetida hasta las 256 posiciones.
- Se usa T para producir la permutación inicial de S.
- Para cada S[i] se intercambia por otro byte S[j] siguiendo la permutación pseudoaleatoria. El índice j se calcula con el procedimiento de arriba.

Cifrados Simétricos: Cifrados de Flujo-RC4

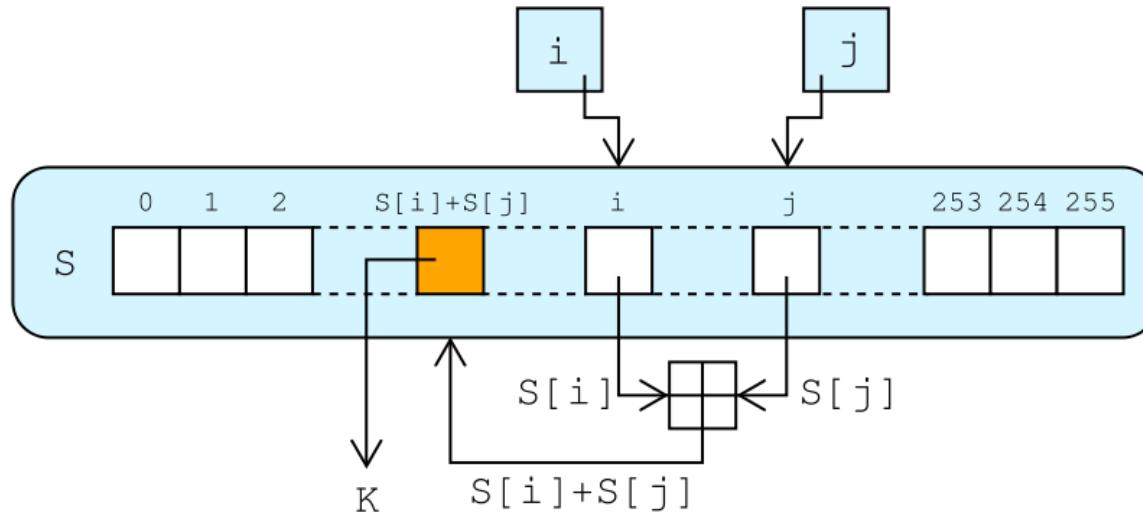
- La semilla ya no se vuelve a utilizar (es decir la clave K anterior).
- La generación del keystream implica cada $S[i]$ un intercambio con otro elemento j de S dictado por el siguiente código. Cuando se termina de recorrer S , se vuelve a empezar por el principio (aritmética modular 256), esta fase es llamada *Pseudo-Random Generation Algorithm*:

```
➤ /* KeyStream Generation */  
➤ i, j = 0;  
➤ while (true)  
➤     i = (i + 1) mod 256;  
➤     j = (j + S[i]) mod 256;  
➤     Swap (S[i], S[j]);  
➤     t = (S[i] + S[j]) mod 256;  
➤     k = S[t];
```

- Para cifrar/descifrar hacemos un XOR de k con texto plano/cifrado.

Cifrados Simétricos: Cifrados de Flujo-RC4

Imagen extraída de
<http://commons.wikimedia.org/w/index.php?title=File:RC4.svg>



```
/* KeyStream Generation */  
i, j = 0;  
while (true)  
    i = (i + 1) mod 256;  
    j = (j + S[i]) mod 256;  
    Swap (S[i], S[j]);  
    t = (S[i] + S[j]) mod 256;  
    k = S[t];
```

- Fue excluido enseguida de los estándares de alta seguridad por los criptógrafos. No está recomendado su uso en los nuevos sistemas, sin embargo, algunos sistemas basados en RC4 son lo suficientemente seguros para un uso común. Se usa por ser rapidísimo.
- WEP es vulnerable pero no por RC4 sino por el protocolo que utiliza el mismo (la forma como se generan las claves).

Criptografía Pública

- Es **más segura** desde el punto de vista del **criptoanálisis** de cifrado simétrico.
- Por esta razón el cifrado de clave pública es una técnica de uso general que ha hecho al cifrado simétrico obsoletos, aunque por el contrario no podemos dejar el cifrado simétrico totalmente ya que el asimétrico tiene una gran carga computacional.
- La distribución de claves está resuelta cuando se utiliza el cifrado de clave pública, en contraste con lo engorroso que es en el cifrado simétrico ([Needham, R., and Schroeder, M. "Using Encryption for Authentication in Large Networks of Computers." Communications of the ACM, December 1978.](#))
- **Claves asimétricas:** dos claves relacionadas, una clave pública y una clave privada , que se utilizan para realizar operaciones complementarias, como cifrado y descifrado o la generación de firma y verificación de la firma.

Criptografía Pública

- **Certificado de Clave Pública:** documento digital emitido y firmado digitalmente con la clave privada de una CA que se une al nombre de un suscriptor de la clave pública y por tanto lo une de manera única a su clave privada. Es decir el certificado indica que el suscriptor identificado tiene el control exclusivo y el acceso a la clave privada correspondiente.
- **Algoritmo criptográfico de clave pública:** utiliza los dos tipos de claves (pública y privada) y cumple la propiedad fundamental que derivar la clave privada de la clave pública es computacionalmente imposible.
- **Infraestructura de Clave Pública (PKI):** un conjunto de políticas, procesos, plataformas de servidores, software y puestos de trabajo utilizados para el propósito de administrar certificados y pares de claves públicas y privadas, incluyendo la capacidad para expedir, renovar y revocar los certificados de clave pública.

Criptografía Pública

- El concepto de criptografía de clave pública aparece y evoluciona por el intento de solucionar dos de los problemas más difíciles relacionados con el cifrado simétrico:
 - **Distribución de claves:** Cómo tener comunicaciones seguras en general sin tener que confiar en una tercera parte o un centro de distribución de claves simétricas (por ejemplo protocolos [Wide-Mouth Frog](#), [Needham-Schroeder Secret-Key](#), Otway-Rees, Kerberos o cualquier otro protocolo de claves simétricas). Diffie, se pregunta ¿De qué serviría desarrollar sistemas criptográficos impenetrables, si sus usuarios se ven obligados a compartir sus claves con un centro de distribución de claves (KDC) que podría verse comprometido por ataques?
 - **Firma digital:** en general es mecanismo criptográfico que permite al receptor de un mensaje firmado digitalmente determinar la entidad originadora de dicho mensaje (autenticación de origen y no repudio), y confirmar que el mensaje no ha sido alterado desde que fue firmado por el originador (integridad).

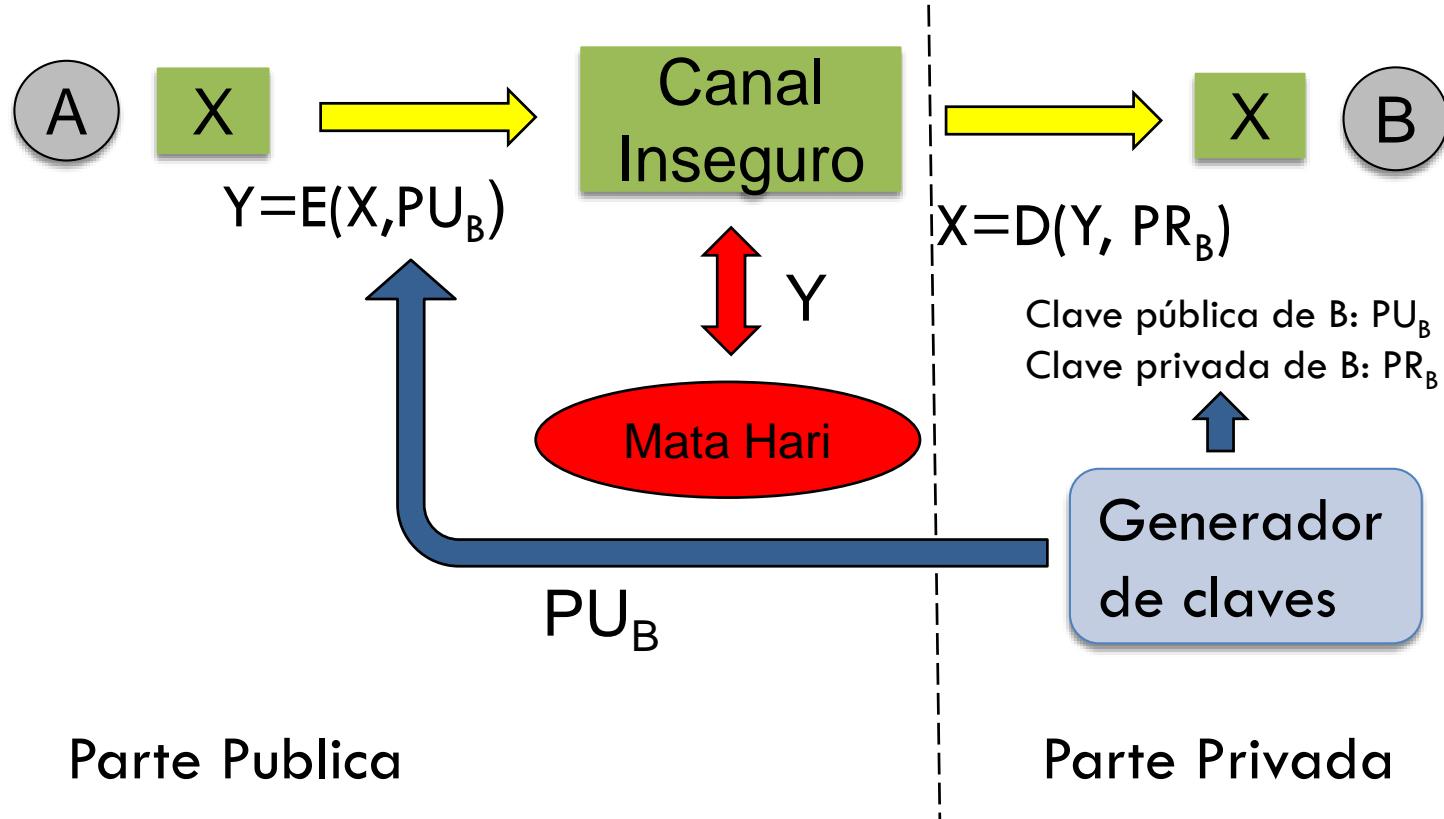
Criptografía Pública

- **Texto Plano:**
- **Texto Cifrado:**
- **Algoritmo de cifrado:**
- **Algoritmo de descifrado:**
- **Claves públicas: PU_A , PU_B :**
- **Claves privadas: PR_A , PR_B :**
- **Propiedades:**
 1. El algoritmo criptográfico cumple la propiedad fundamental que derivar la clave privada de la clave pública es computacionalmente imposible .
 2. Clave para el cifrado y una diferente pero relacionada con esta para el descifrado.
 3. Es computacionalmente imposible determinar la clave de descifrado dado sólo el conocimiento del algoritmo de cifrado y la clave de cifrado.

Criptografía Pública

- Cada usuario genera un par de claves que se utilizará para el cifrado y el descifrado de los mensajes (PU_A , PU_B , PR_A , PR_B).
- Cada usuario publica su clave pública.
- Cada usuario mantiene su clave privada completamente secreta.
- Si **Alicia** quiere enviar un mensaje confidencial a **Bernardo** entonces cifra el mensaje con la clave pública de **Bernardo** (PU_B).
- **Bernardo** descifra el mensaje cifrado enviado mediante sus clave privada (PR_B).
- Ningún otro destinatario puede descifrar el mensaje porque sólo **Bernardo** sabe su clave privada.
- Para enviar un mensaje **Bernardo** a **Alicia** se realiza el mismo procedimiento, pero en el otro sentido.

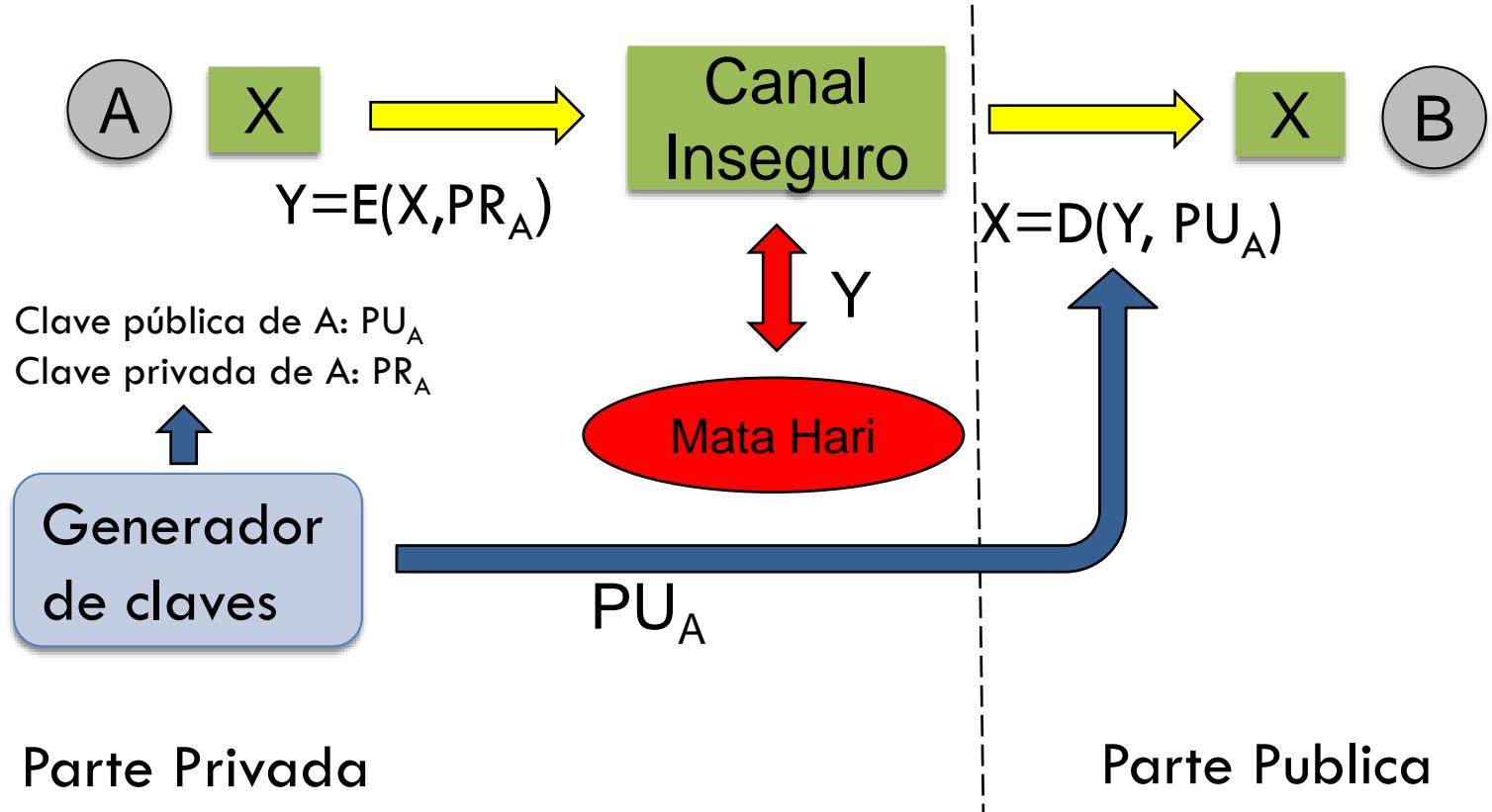
Cifrado Asimétrico: Confidencialidad



Parte Pública

Parte Privada

Cifrado Asimétrico: Autentificación



Cifrado Simétrico vs Asimétrico

CIFRADO SIMÉTRICO

- Necesidad para el **Funcionamiento:**
 - Mismo algoritmo con la misma clave para el cifrado y el descifrado.
 - El emisor y el receptor deben compartir la algoritmo y la clave.
- Necesidad para la **Seguridad:**
 - La clave debe mantenerse en secreto.
 - Debe ser imposible descifrar un mensaje si la clave se mantiene en secreto.
 - El conocimiento del algoritmo + textos cifrados es insuficiente para determinar la clave.

CIFRADO ASIMÉTRICO

- Necesidad para el **Funcionamiento:**
 - Un algoritmo se utiliza para el cifrado y descifrado con un par de claves, una para el cifrado y otra para descifrado.
 - El emisor y el receptor deben tener cada uno una del par de claves.
- Necesidad para la **Seguridad:**
 - Una de las dos claves debe mantenerse en secreto.
 - Debe ser imposible descifrar un mensaje si una de las claves se mantiene en secreto.
 - El conocimiento del algoritmo + textos cifrados + una de las claves es insuficiente para determinar la otra clave.

Cifrados Asimétricos: Aplicaciones

- Los criptosistemas de criptografía pública se pueden clasificar en tres categorías:
 - **Cifrado y descifrado:** Alicia cifra un mensaje con la clave pública de Bernardo.....
 - **Firma Digital:** Alicia firma un mensaje con su clave privada.....
 - **Intercambio de claves:** dos partes se ponen de acuerdo para el intercambio de claves.
- No todos los algoritmos basados en criptografía pública se pueden utilizar para todas las aplicaciones:

Algoritmo	Cifrado/Descifrado	Firma Digital	Intercambio de claves
RSA	SI	SI	SI
Curvas Elípticas	SI	SI	SI
Diffie-Hellman	NO	NO	SI
DDS	NO	SI	NO

Cifrados Asimétricos: Requerimientos de los Algoritmos

- Es computacionalmente **fácil** para las partes **generar** los pares de **claves públicas y privadas**.
- Es computacionalmente **fácil** para el emisor, conociendo la clave pública del receptor, **generar el texto cifrado** correspondiente.
- Es computacionalmente **fácil** para el receptor para **desencriptar el texto cifrado** resultante usando la **clave privada** para recuperar el mensaje original.
- Es **computacionalmente imposible** para un adversario, a través de la **clave pública**, **determinar la clave privada asociada**.
- Es **computacionalmente imposible** para un adversario si conoce la **clave pública** y un texto cifrado, **recuperar el mensaje original**.
- La pareja de claves se pueden aplicar en cualquier orden.
- Los algoritmos anteriores y sus relacionados cumplen estas características y requerimientos.

Cifrados Asimétricos: Requerimientos de los Algoritmos (funciones Trap-door y One Way)

- Estas condiciones anteriores se traducen en los conceptos de funciones **One Way** y **Trap-door**.
- Función One Way:
 - $Y = f(X)$ fácil computacionalmente hablando
 - $X = f^{-1}(Y)$ es imposible computacionalmente hablando
- Función Trap-door: es una familia de funciones invertibles que cumplen:
 - $Y = f_k(X)$ fácil, si k y X son conocidos
 - $X = f_k^{-1}(Y)$ fácil, si k y Y son conocidos
 - $X = f_k^{-1}(Y)$ inviable, si Y es conocido pero no k
- Por ejemplo: la aritmética modular como principio de diseño de estas propiedades (con aritmética modular para sacar x solo por fuerza bruta).
 - $\begin{array}{rcccccc} x & - & 1 & 2 & 3 & 4 & 5 & 6 \\ 3^x & - & 3 & 9 & 27 & 81 & 243 & 729 \\ 3^x \bmod 7 & - & 3 & 2 & 6 & 4 & 5 & 1 \end{array}$

Cifrados Asimétricos: Criptoanálisis

- **Lo ataques clásicos que hoy en día no se dan:**
 - Ataque de función de Euler, Exponente común pequeño, Ataque las Vegas, Timming attacks.
- **Fuerza Bruta:** Se sube el tamaño de clave tanto como sea necesario.
 - Esa es la razón de que los criptosistemas públicos sean lentos para cifrar y generalmente se suelen utilizar para gestión de claves y firma digital.
 - La recomendación para tamaños de claves según diferentes organizaciones es:
<http://www.keylength.com/>
 - [Transición algoritmos y tamaños de claves](#).
- Otra forma de ataque es encontrar algún **método para encontrar la clave privada** a través de la clave pública. Hasta el momento esto no se ha logrado pero esto no quiere decir nada.
- Por último esta el **ataque de mensaje**: si sabemos que tipo de mensaje va cifrado con criptografía pública, y ese mensaje tiene un número de posibilidades lo suficientemente pequeñas como para atacarlo por fuerza bruta: **es decir probamos todas la posibilidades del mensaje con la clave pública.**
 - Por ejemplo imaginemos que mandamos una clave de DES de 56 bits cifrada con la clave pública (la solución añadir algo aleatorio).

Cifrados Asimétricos: RSA (Rivest-Shamir-Adleman)

- Fue desarrollado en 1977 en el MIT por Ron **Rivest**, Adi **Shamir** & Len **Adleman**.
- Es el más usado de **propósito general**.
- Es una cifra en la que el texto plano y texto cifrado son enteros entre 0 y $n-1$ para algún n .
- Un tamaño típico para n es de 1024 bits, o 309 dígitos decimales, aunque hoy en día el NIST aconseja 2048 bits (<http://www.keylength.com/>).
- RSA hace uso de la función matemática de **potenciación modular**.
- El texto plano se cifra por bloques, cada uno con un valor binario de menos de un número n .
- $C = M^e \text{ mod } n$.
- $M = C^d \text{ mod } n = (M^e)^d \text{ mod } n = M^{ed} \text{ mod } n$.
- Tanto el emisor como el receptor deben conocer el valor de n .
- El remitente sabe el valor de e , y sólo el receptor conoce el valor de d .
- Se trata de un algoritmo de cifrado de clave pública con una clave pública (suponiendo que B recibe) $PU_B = \{e, n\}$ y una clave privada de $PR_B = \{d\}$.

Cifrados Asimétricos: RSA, Requerimientos y Algoritmo

- Los requerimientos para el algoritmo del RSA son:
 - Es posible **encontrar valores de e, d, n** tal que $M^{ed} \bmod n = M$ para todo $M < n$
 - Es relativamente **fácil de calcular** $M^e \bmod n$ y $C^d \bmod n$ para todos los valores de $M < N$
 - Es **factible** para determinar d dada e y n
- Generación de las claves pública y privada:
 - Seleccionamos p, q primos distintos ([mediante algoritmos de primalidad](#))
 - **Calculamos $n = p \times q$ (PUNTO CRÍTICO)**
 - Calculamos $F(n) = (p - 1)(q - 1)$ (denominada F como [función de Euler](#)).
 - Seleccionamos e como el mcd ($F(n), e$) = 1; es decir $1 < e < F(n)$
 - Calculamos d como $d = e^{-1} \pmod{F(n)}$ ([algoritmo de Euclides extendido](#))
 - Clave pública PU = {e, n}
 - Clave privada PR = {d, n}
- Cifrado/Descifrado:
 - Texto plano: $M < n$, $C = M^e \bmod n$
 - Texto cifrado: C , $M = C^d \bmod n$

Cifrados Asimétricos: RSA, Ejemplo Numérico

- Seleccionamos dos números primos, $p = 17$ and $q = 11$.
- Calculamos $n = pq = 17 \times 11 = 187$.
- Calculamos la función de Euler $F(n) = (p - 1)(q - 1) = 16 \times 10 = 160$.
 - $F(n) = |\{m \in \mathbb{N} \mid m \leq n \wedge \text{mcd}(m, n) = 1\}|$
 - TFA: Todo entero positivo $n > 1$ puede ser representado exactamente de una única manera como un producto de potencias de números primos:
 - $n = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$, con p_i primos distintos. Por ejemplo $1000 = 2^3 \times 5^3$.
 - $F(n) = (p_1 - 1) p_1^{(a_1 - 1)} \dots \dots (p_k - 1) p_k^{(a_k - 1)}$
- Seleccionamos un e primo relativo con $F(n) = 160$ menor que el; así por ejemplo elegimos $e = 7$
- Determinamos $d = e^{-1} \pmod{160}$, es decir $d = 23$ (mediante el algoritmo de Euclides extendido).
- Clave pública PU = {7, 187}.
- Clave privada PR = {23, 187}.

Cifrados Asimétricos: RSA, Ejemplo Numérico

- Supongamos que el mensaje que queremos mandar es $M=88$.
- Para cifrar $C = 88^7 \text{ mod } 187$:
 - $88^7 \text{ mod } 187 = [(88^4 \text{ mod } 187) \times (88^2 \text{ mod } 187)$
 - $\times (88^1 \text{ mod } 187)] \text{ mod } 187$
 - $88^1 \text{ mod } 187 = 88$
 - $88^2 \text{ mod } 187 = 7744 \text{ mod } 187 = 77$
 - $88^4 \text{ mod } 187 = 59,969,536 \text{ mod } 187 = 132$
 - $88^7 \text{ mod } 187 = (88 \times 77 \times 132) \text{ mod } 187 = 894,432 \text{ mod } 187 = 11$
- Para descifrar $M = 11^{23} \text{ mod } 187$:
 - $11^{23} \text{ mod } 187 = [(11^1 \text{ mod } 187) \times (11^2 \text{ mod } 187) \times (11^4 \text{ mod } 187)$
 - $\times (11^8 \text{ mod } 187) \times (11^8 \text{ mod } 187)] \text{ mod } 187$
 - $11^1 \text{ mod } 187 = 11$
 - $11^2 \text{ mod } 187 = 121$
 - $11^4 \text{ mod } 187 = 14,641 \text{ mod } 187 = 55$
 - $11^8 \text{ mod } 187 = 214,358,881 \text{ mod } 187 = 33$
 - $11^{23} \text{ mod } 187 = (11 \times 121 \times 55 \times 33 \times 33) \text{ mod } 187 = 79,720,245 \text{ mod } 187 = 88$

Cifrados Asimétricos: RSA, Potenciación Modular

POTENCIACIÓN MODULAR (Z, e, n)

$$X = 1$$

FOR $i = l-1 \rightarrow 0$

$$e = \sum_{i=0}^{l-1} a_i 2^i$$

$$X = X^2 \bmod n$$

IF ($a_i == 1$)

$$\text{THEN } X = (X \cdot Z) \bmod n$$

DECOMPOSICIÓN
BINARIA DEL
EXPONENTE

RETURN (X)

INPUT $\begin{cases} z \\ e \\ n \end{cases}$ { COMPUTA EFICIENTEMENTE } OUTPUT $X = z^e \bmod n$ } $z^e \bmod n$

- Para la potenciación modular podemos usar la propiedad:
 - $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$
- Hay que darse cuenta que los números son muy grandes y tenemos que utilizar un algoritmo eficiente para la potenciación modular como por ejemplo:

EJEMPLO: $z^{11} \bmod n = z^{(1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)}$

$$x = 1$$

$$i = 3$$

$$x = 1^2 \bmod n$$

$$a_3 = 1 \rightarrow x = 1^2 \times z \bmod n$$

$$i = 2$$

$$x = z^2 \bmod n$$

$$a_2 = 0 \rightarrow \text{NO HACE NADA}$$

$$i = 1$$

$$x = (z^2)^2 \bmod n$$

$$a_1 = 1 \rightarrow x = (z^2)^2 \times z \bmod n$$

$$i = 0$$

$$x = ((z^2)^2 \times z)^2$$

$$a_0 = 1 \rightarrow x = ((z^2)^2 \times z)^2 \times z \bmod n$$

$$x = z^{11} \bmod n$$

Cifrados Asimétricos: Requerimientos de los Algoritmos

- Para la operación de potenciación modular normalmente se elige un exponente de cifrado con cierta propiedades de eficiencia computacional:
 - La elección más común es $65537 (2^{16} + 1)$, es el cuarto número de Fermat ($F_n = 2^{2n} + 1$ con $n = 4$) llamado F_4 . Es un número primo y tiene un peso de Hamming muy bajo. Es lo suficientemente grande para no sufrir ataques.
 - Otras dos opciones populares son $e = 3$ y $e = 17$ (inseguros, ya que para un mensaje pequeño M^e puede ser más pequeño que m y por tanto con hacer la raíz e -ésima tienes M).
 - Cada una de estas opciones tiene sólo dos bits de 1's, por lo que el número de multiplicaciones necesarias para realizar la exponenciación se minimiza.
 - Con una muy pequeña clave pública, como $e = 3$, RSA se vuelve vulnerable a un ataque sencillo.
- Para la clave de descifrado esta no puede ser pequeña para que no sea atacada por fuerza bruta, pero puede ser más eficiente por el teorema del Resto Chino. Se puede hacer hasta 4 veces más rápido.

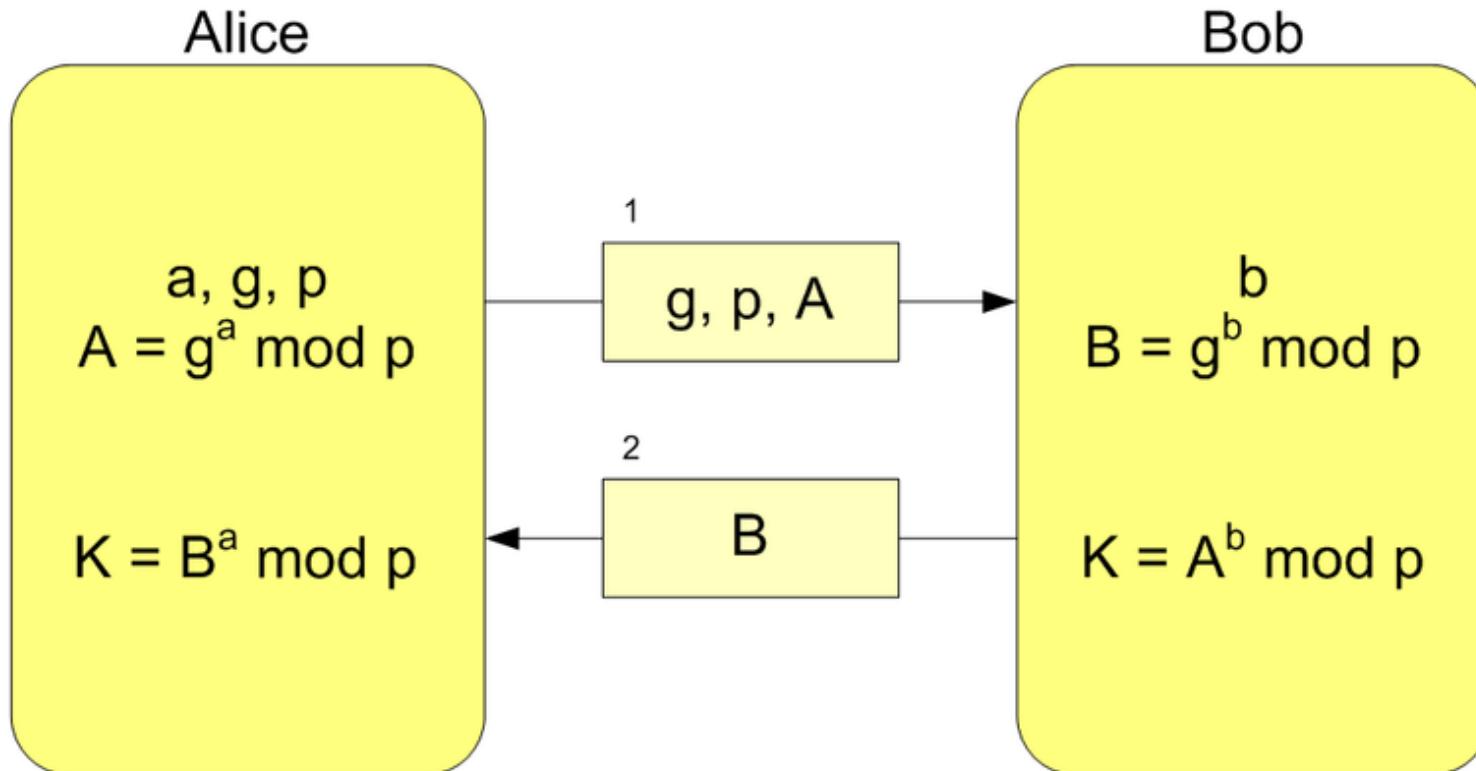
Cifrados Asimétricos: Intercambio de Claves Diffie-Hellman

- Publicado por primera vez como algoritmo de clave pública (Diffie, W., and Hellman, M. "Multiuser Cryptographic Techniques." IEEE Transactions on Information Theory, November 1976).
- Un gran número de productos comerciales emplean esta técnica de intercambio de claves (por ejemplo **ORACLE**, o la red para anonimato **Tor** usa el protocolo Diffie Hellman, sobre una conexión TLS).
- El propósito es permitir que dos usuarios puedan intercambiar de forma segura una clave que se puede utilizar después para el cifrado simétrico en los siguientes mensajes.
- El algoritmo en sí está limitado al **intercambio de los valores secretos**.
- Su eficacia depende de la dificultad de cálculo de **logaritmos discretos**.

Cifrados Asimétricos: Intercambio de Claves Diffie-Hellman

- Podemos definir los logaritmos discretos como sigue:
- Una raíz primitiva a de un número primo p es aquella que cuyas potencias en módulo p generan **todos** los números enteros de 1 a $p-1$.
 - Es decir, si a es raíz primitiva de p todos entonces estos números son distintos:
 - $a \text{ mod } p, a^2 \text{ mod } p, \dots, a^{p-1} \text{ mod } p$ (i.e. el resultado es una permutación del conjunto $\{1\dots p-1\}$)
- Para cualquier número entero b y una raíz primitiva a de un número primo p , podemos encontrar un exponente único i tal que $b = a^i \pmod{p}$, donde el exponente se encuentra en $0 \leq i \leq (p - 1)$.
- El exponente i se conoce como el logaritmo discreto de b para la base a , en aritmética **mod p**.
- Expresamos este valor como $i = \text{dlog}_{a, p}(b)$.
- Ya sabemos que el calculo del logaritmo discreto es computacionalmente inviable, al igual que pasaba en RSA. En esta computación inviable se basa el intercambio de claves DH.

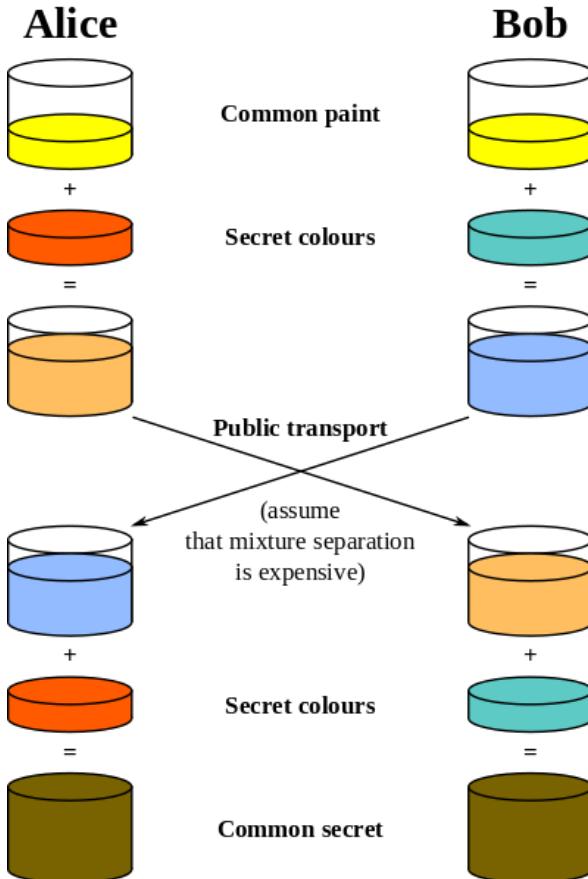
Cifrados Asimétricos: Intercambio de Claves Diffie-Hellman



$$K = A^b \text{ mod } p = (g^a \text{ mod } p)^b \text{ mod } p = g^{ab} \text{ mod } p = (g^b \text{ mod } p)^a \text{ mod } p = B^a \text{ mod } p$$

Imagen extraída de
<https://es.wikipedia.org/wiki/Diffie-Hellman>

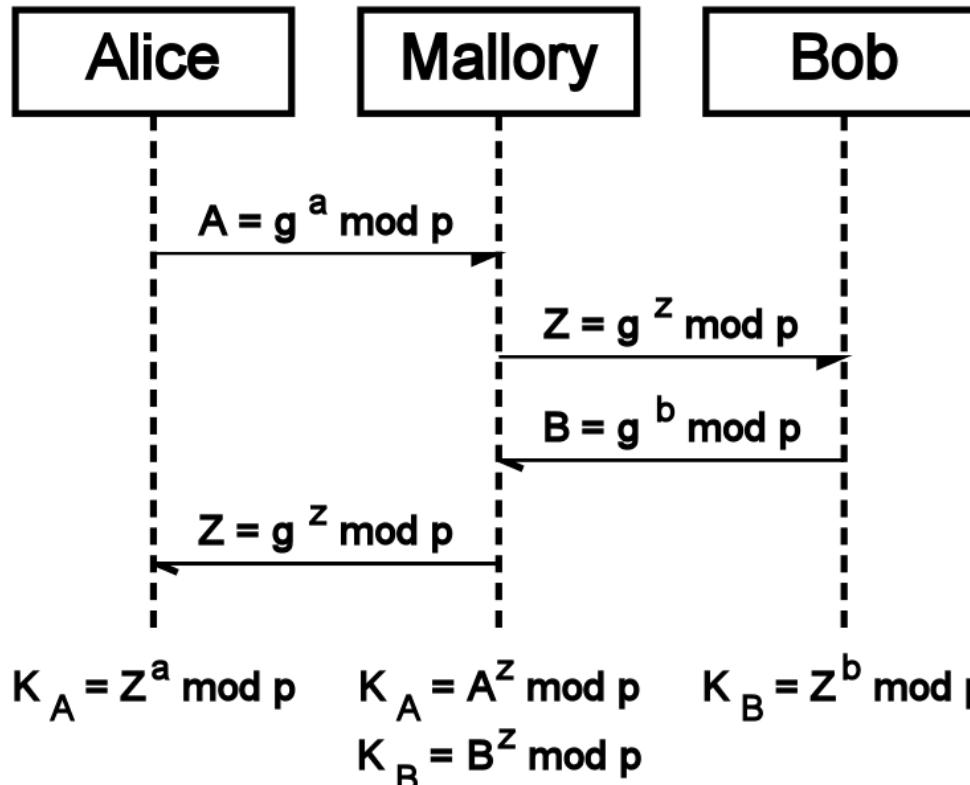
Cifrados Asimétricos: Intercambio de Claves Diffie-Hellman



- Alice y Bob acuerdan el primo $p = 23$ y la $g = 5$ base (es una raíz primitiva del numero primo p).
- Alice elige un secreto $a = 6$, y envía a Bob $A = g^a \text{ mod } p$
 - $A = 5^6 \text{ mod } 23$
 - $A = 15,625 \text{ mod } 23$
 - $A = 8$
- Bob elige un secreto $b = 15$, y envía a Alice $B = g^b \text{ mod } p$
 - $B = 5^{15} \text{ mod } 23$
 - $B = 30,517,578,125 \text{ mod } 23$
 - $B = 19$
- Alice calcula el secreto compartido $s = B^a \text{ mod } p$
 - $s = 19^6 \text{ mod } 23$
 - $s = 47,045,881 \text{ mod } 23$
 - $s = 2$
- Bob calcula el secreto compartido $s = A^b \text{ mod } p$
 - $s = 8^{15} \text{ mod } 23$
 - $s = 35,184,372,088,832 \text{ mod } 23$
 - $s = 2$
- Alice y Bob ahora comparten el secreto $s = 2$.

Imagen extraída de http://en.wikipedia.org/wiki/File:Diffie-Hellman_Key_Exchange.svg

Cifrados Asimétricos: Intercambio de Claves Diffie-Hellman



- Control de tiempos.
- Autenticación previa de las partes.
- Autenticación del contenido. Por ejemplo podríamos usar MAC sobre el contenido de los mensajes.
- Mediante el uso de las firmas digitales y certificados de clave pública ([protocolo STS](#)).

Imagen extraída de http://commons.wikimedia.org/wiki/File:Man-in-the-middle_attack_of_Diffie-Hellman_key_agreement.svg

Cifrados Asimétricos: Elgamal

- Anunciado en 1984 por T. Elgamal:
 - "Elgamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." Proceedings, Crypto 84, 1984.
 - Elgamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." IEEE Transactions on Information Theory, July 1985.
- Esquema de clave pública basado en logaritmos discretos estrechamente relacionados con la técnica de Diffie-Hellman.
- Se utiliza en el estándar de firma digital (DSS) y el estándar de correo electrónico S/MIME (es un estándar para criptografía de clave pública y firmado de correo electrónico encapsulado en MIME o Multipurpose Internet Mail Extensions).
- Elementos globales públicos: un número q primo y un número a que es una raíz primitiva de q .
- La seguridad se basa en la dificultad de cálculo de logaritmos discretos.

Cifrados Asimétricos: Elgamal, Protocolo

- **Elementos globales públicos:** q número primo y α con $\alpha < q$ siendo una raíz primitiva de q.

- **Generación de claves por Alicia:**

- Seleccionar una clave privada X_A con $X_A < q - 1$
- Calcular Y_A como $Y_A = \alpha^{X_A} \text{ mod } q$
- Clave pública: PU={q, α , Y_A }
- Clave privada: PR={ X_A }

Este es el proceso
clave:

$$X_A = \text{dlog}_{\alpha,q}(Y_A).$$

- **Cifrado por Bernardo con la clave pública de Alicia:**

- Texto plano: $M < q$
- Seleccionar un entero k aleatorio: $k < q$
- Calcular K: $K = (Y_A)^k \text{ mod } q$ (es decir mod q)
- Calcular C_1 : $C_1 = \alpha^k \text{ mod } q$
- Calcular C_2 : $C_2 = KM \text{ mod } q$
- Texto cifrado: (C_1, C_2)

- **Descifrado de Alicia con su clave privada:**

- Texto cifrado: (C_1, C_2)
- Calcular K: $K = (C_1)^{X_A} \text{ mod } q$ (darse cuenta $K = (\alpha^k)^{X_A} \text{ mod } q$)
- Texto plano: $M = (C_2 K^{-1}) \text{ mod } q$

$$\begin{aligned} K &= (\alpha^{X_A})^k \text{ mod } q = \\ &= (\alpha^k)^{X_A} \text{ mod } q = \\ &= \alpha^{k X_A} \text{ mod } q = K \end{aligned}$$

Cifrados Asimétricos: Elgamal, Ejemplo

- **Elementos globales públicos:** $q=19$ que tiene como raíces primas $\{2, 3, 10, 13, 14, 15\}$, y elegimos $\alpha=10$.
- **Generación de claves por Alicia:**
 - Seleccionar una clave privada $X_A = 5$, con $X_A < q - 1$
 - Calcular Y_A como $Y_A = \alpha^{X_A} \bmod q = 10^5 \bmod 19 = 3$
 - Clave pública: PU = $\{q=19, \alpha=10, Y_A=3\}$
 - Clave privada: PR = $\{X_A=5\}$
- **Cifrado por Bernardo con la clave pública de Alicia:**
 - Texto plano: $M=17 < q$
 - Seleccionar un entero k aleatorio: $k=6 < q$
 - Calcular K: $K=(Y_A)^k \bmod q = 3^6 \bmod 19 = 729 \bmod 19 = 7$
 - Calcular C_1 : $C_1 = \alpha^k \bmod q = 10^6 \bmod 19 = 11$
 - Calcular C_2 : $C_2 = KM \bmod q = 7 \times 17 \bmod 19 = 119 \bmod 19 = 5$
 - Texto cifrado: $(C_1, C_2) = (11, 5)$
- **Descifrado de Alicia con su clave privada:**
 - Texto cifrado: $(C_1, C_2) = (11, 5)$
 - Calcular K: $K = (C_1)^{X_A} \bmod q = 11^5 \bmod 19 = 161051 \bmod 19 = 7$
 - Texto plano: $M = (C_2 K^{-1}) \bmod q = 5 \times 11 \bmod 19 = 55 \bmod 19 = 17$

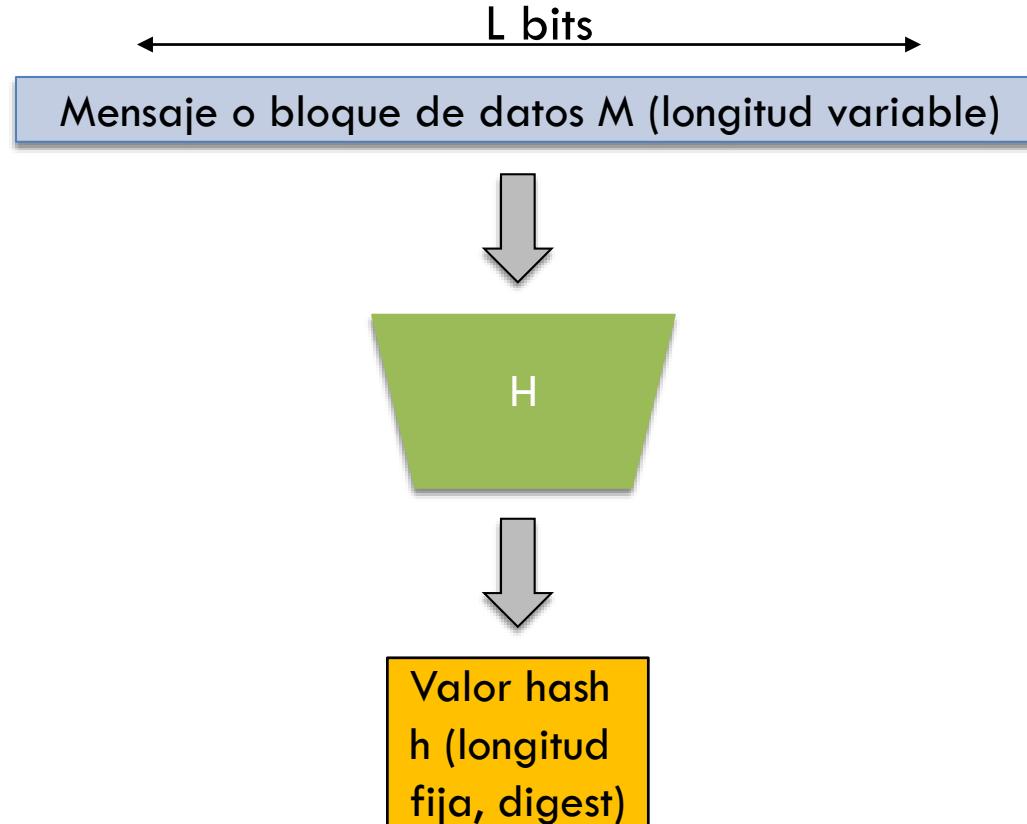
Cifrados Asimétricos: Seguridad

- La seguridad reside en el coste computacional de los logaritmos discretos.
- Para recuperar la clave privada de Alicia, un adversario tendría que calcular
$$X_A = \text{dlog}_{\alpha,q}(Y_A).$$
- Como alternativa, para recuperar la clave de una sola vez, un adversario tendría que determinar el número aleatorio k , y esto requeriría la computación $k = \text{dlog}_{\alpha,q}(C_1)$ del logaritmo discreto.
- En el libro de Stinson, D. Cryptography: Theory and Practice. Boca Raton, FL: CRC Press, 2006, señala que estos cálculos se consideran inviables si el primo q es de al menos 300 dígitos decimales y $(q-1)$ tiene por lo menos un factor primo "grande".
- Hay que tener **cuidado cuando se cifran diferentes bloques no utilizar un número k aleatorio igual en dos bloques** porque sino tenemos roto el criptosistema cuando conozcamos por alguna razón uno de los bloques:
 - Supongamos $C_{1,1} = \alpha^k \bmod q; C_{2,1} = \alpha^k \bmod q = KM_1 \bmod q$ (bloque 1)
 - Supongamos $C_{1,2} = \alpha^k \bmod q; C_{2,2} = \alpha^k \bmod q = KM_2 \bmod q$ (bloque 2)
 - $C_{2,1} / C_{2,2} = (KM_1 \bmod q) / (KM_2 \bmod q) = (M_1 \bmod q) / (M_2 \bmod q)$
 - Si M_1 entonces M_2 se puede computar fácilmente:
 - $M_2 = (C_{2,2})^{-1} C_{2,1} M_1 \bmod q$

Funciones Criptográficas HASH

- Es un mecanismo o servicio que se utiliza para verificar la **integridad de los mensajes**.
- La **autenticación de mensajes** asegura que los datos recibidos son exactamente como enviados (es decir, no contienen ninguna modificación, inserción, eliminación o reproducción).
- También en muchos casos, hay un requisito de que el mecanismo de autenticación asegura que la supuesta identidad del remitente es válida.
- Cuando se utiliza una función hash para proporcionar autenticación de mensajes, el valor de la función hash se refiere a menudo como un resumen del mensaje.
- Autenticación de mensajes:
 - el remitente calcula un valor hash como una función de los bits en el mensaje y transmite tanto el valor hash y el mensaje
 - el receptor realiza la misma hash de cálculo de los bits de mensaje y compara este valor con el entrante valor hash. Si hay una falta de coincidencia , el receptor sabe que el mensaje (o posiblemente el valor hash) ha sido alterado.

Funciones Criptográficas HASH



Funciones Criptográficas HASH

Input

Digest

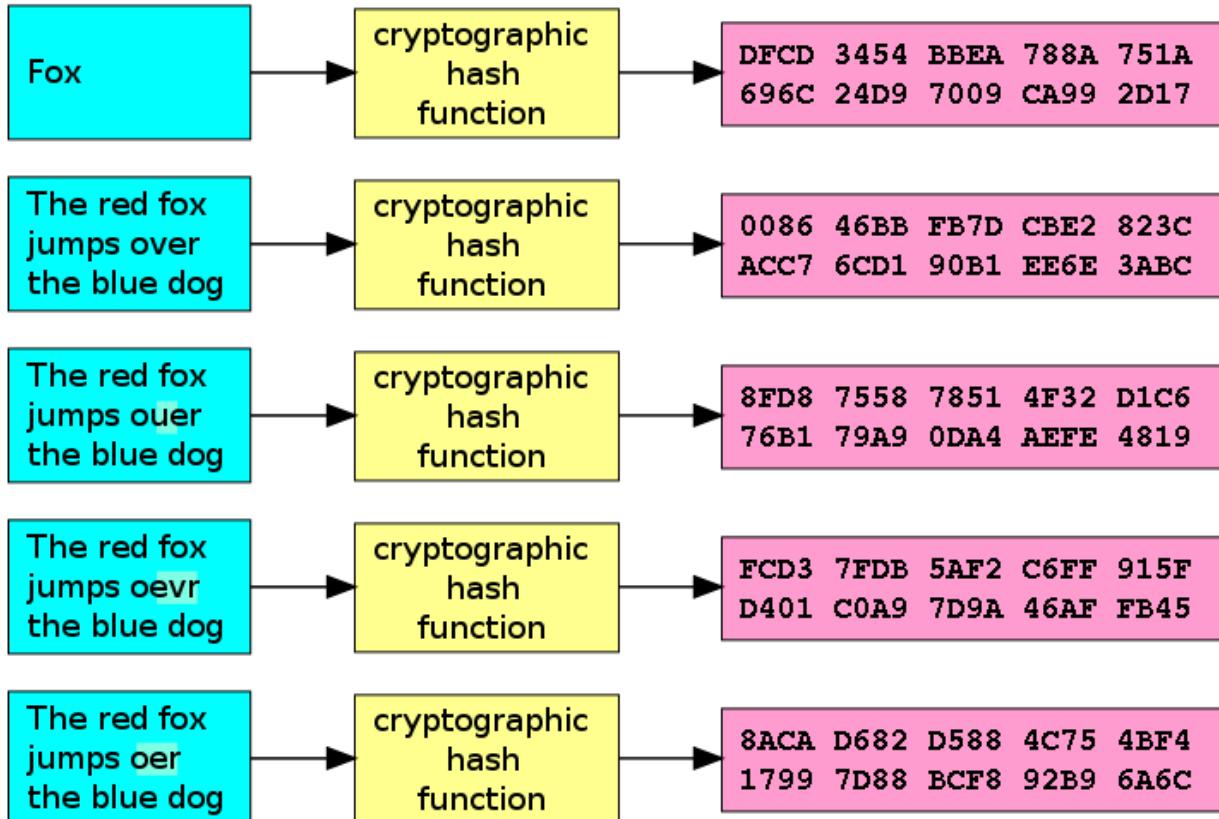


Imagen extraída de
http://en.wikipedia.org/wiki/File:Cryptographic_Hash_Function.svg

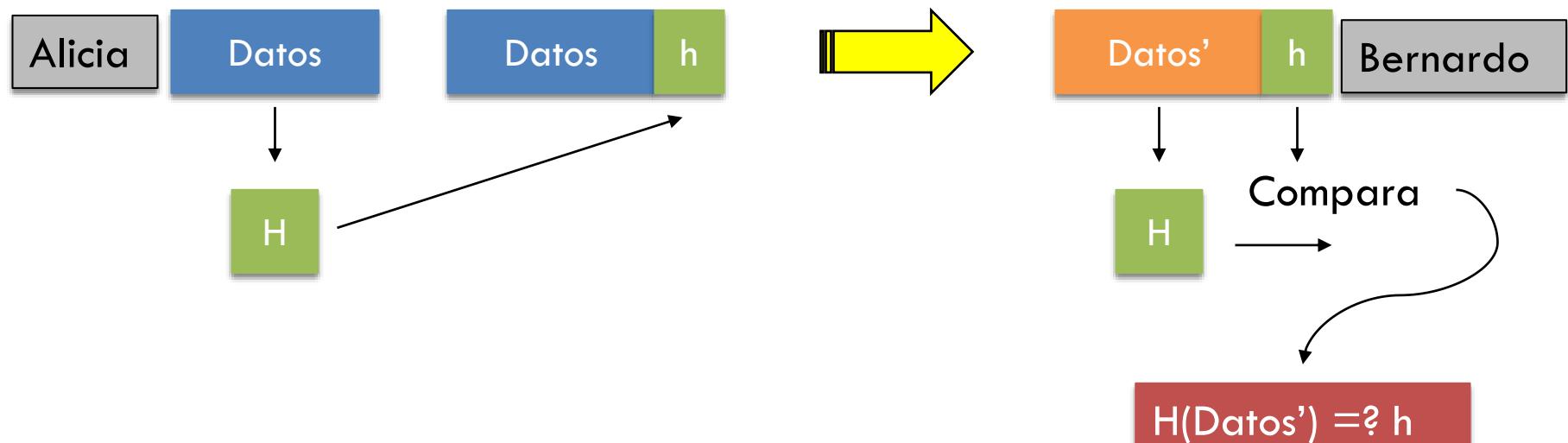
Funciones Criptográficas HASH

- Una función hash H acepta un bloque de longitud variable de los datos M como entrada y produce un valor hash de tamaño fijo:
 - $h = H(M)$
- Es la primitiva principal es la **integridad de datos**:
 - un cambio a cualquier bit o los bits en de M , resulta con alta probabilidad, un cambio en el código hash h (*digest*).
- La función hash criptográfica es un algoritmo para el que es imposible computacionalmente encontrar:
 - un M de datos que se asigna a un resultado h (la propiedad de un solo sentido, *one-way*)
 - dos M iguales de datos que se correlacionan con el mismo resultado h (la propiedad libre de colisiones, *collision-free*)

Aplicaciones de las Funciones Criptográficas HASH

- Las funciones criptográficas Hash tienen muchos usos, entre los más comunes están:
 - Autentificación de mensajes.
 - Firmas digitales.
 - One-Way Passwords.
 - Detección de intrusos.
 - Detección de virus.
 - Generación de números pseudoaleatorios o funciones pseudoaleatorias.
- Vamos a ver todas estas funcionalidades con un poco más de detalle.

Funciones Criptográficas HASH

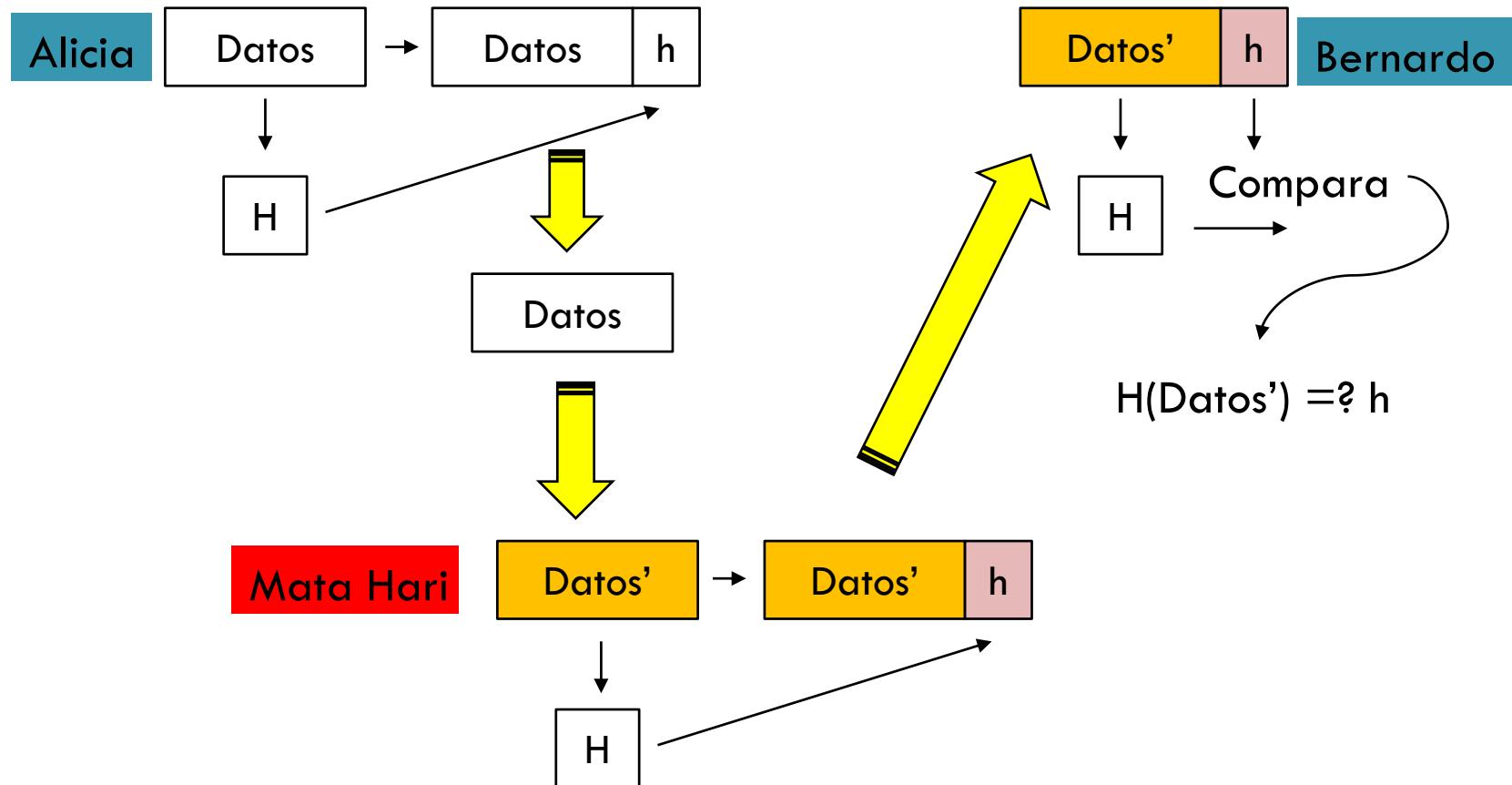


Integridad o Autenticación de Mensaje

Funciones Criptográficas HASH Man-in-the-middle-attack

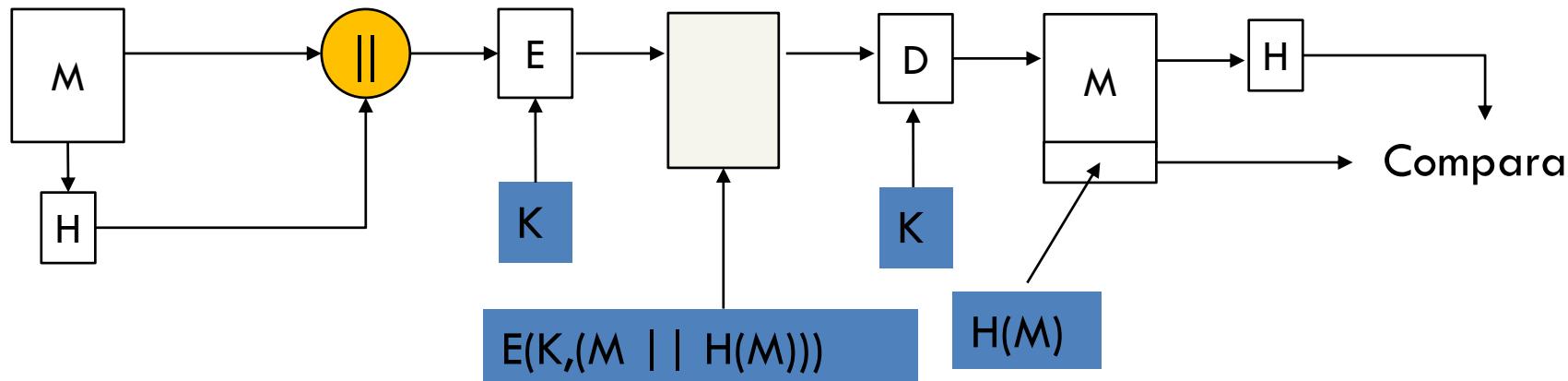
- La función de hash debe ser transmitida de una manera segura.
- El hash se protegerá de forma que si un adversario altera o reemplaza el mensaje, no es factible para adversario alterar también el valor hash para engañar al receptor.
- Este tipo de ataques se llaman del tipo **Man-in-the-middle-attack**.
- Alicia transmite un bloque de datos y produce un valor hash.
- Mata Hari intercepta el mensaje y altera o reemplaza el bloque de datos, y calcula y se une un nuevo valor hash.
- Bernardo recibe los datos alterados con el nuevo valor de hash y no detecta el cambio.
- Para evitar este ataque, el **valor hash generado** por Alicia debe ser **protegido** (lo vemos a continuación).

Funciones Criptográficas HASH Man-in-the-middle-attack



Funciones Criptográficas HASH: Protegiendo el hash

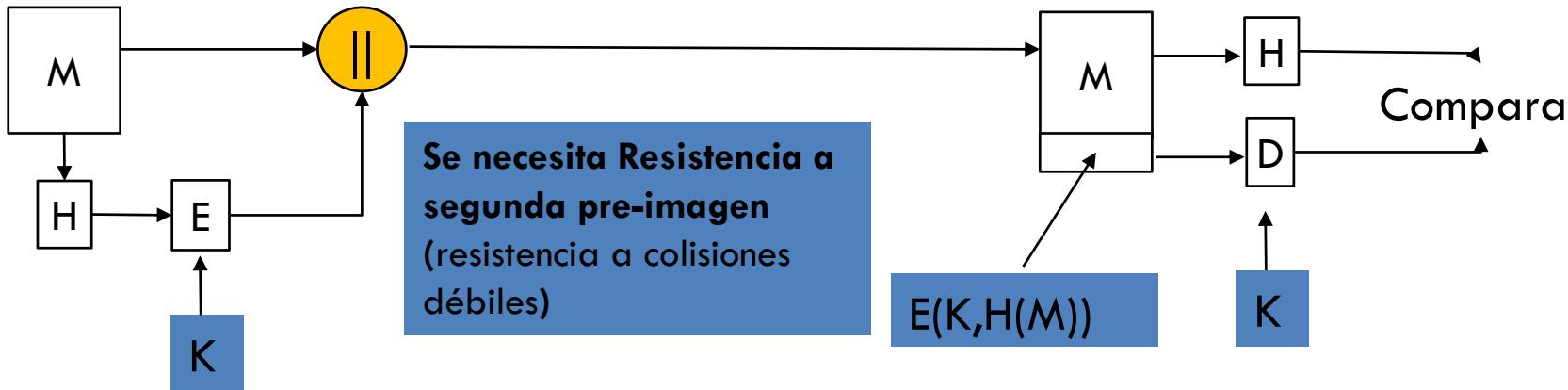
Ejemplos de Autenticación de Mensaje



- Debido a que sólo A y B comparten la clave secreta, el mensaje que ha venido de A no ha sido alterado.
- El código hash proporciona la estructura o la redundancia requerida para lograr la **autenticación**.
- Dado que el cifrado es aplicado a todo el mensaje más código hash, también se proporciona **confidencialidad**.

Funciones Criptográficas HASH : Protegiendo el hash

Ejemplos de Autenticación de Mensaje

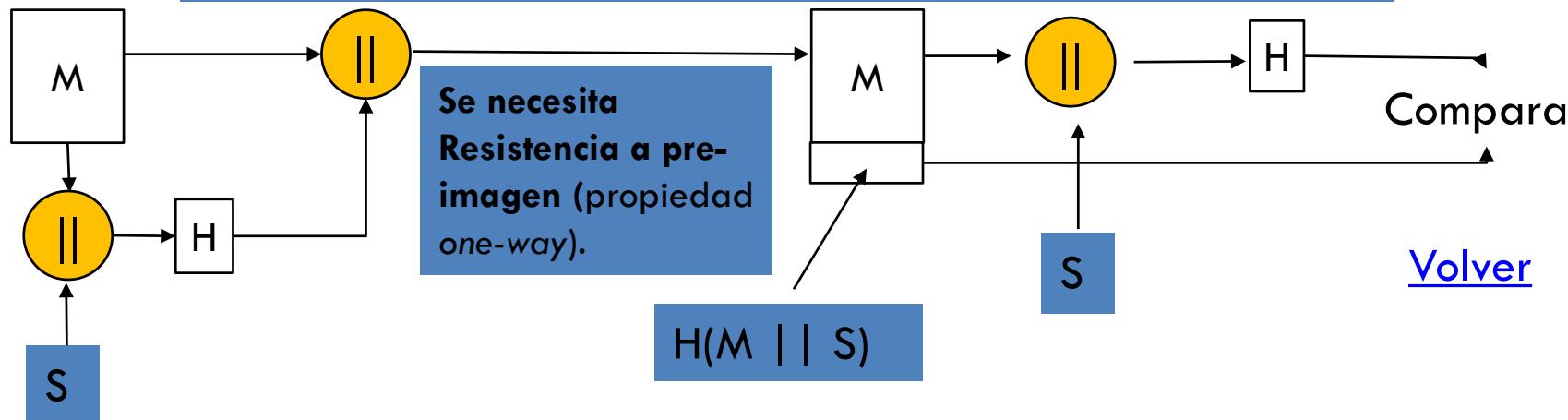


- Sólo se cifra el código hash mediante el cifrado simétrico.
- Esto reduce la carga de procesamiento para aquellas aplicaciones que no requieren la **confidencialidad del mensaje**.
- Pero si se protege el código hash (evitar Man-in-the-middle-attack).

Funciones Criptográficas HASH : Protegiendo el hash

Ejemplos de Autenticación de Mensaje: SIN CIFRAR

Este protocolo se podría considerar como una función MAC

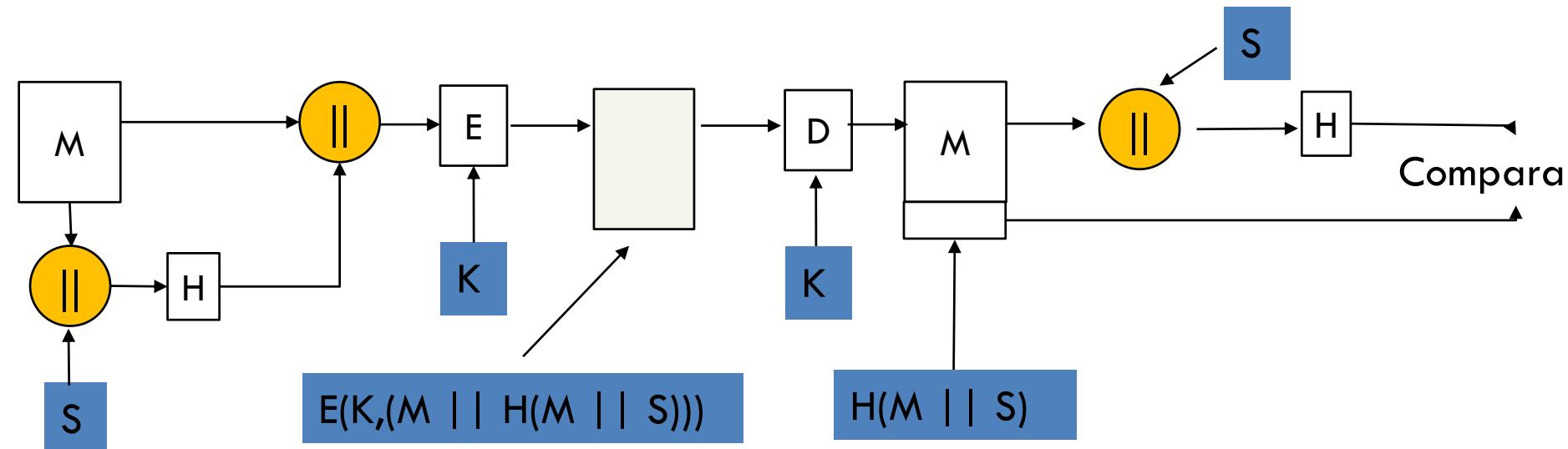


[Volver](#)

- Es posible utilizar una función hash, pero no la encriptación para la autenticación de mensajes.
- La técnica supone que las dos partes que se comunican comparten un **secreto común, el valor S**.
- **El código hash se protege mediante el secreto común** (compartido con DH por ejemplo).
- Debido a que el secreto no se envía, un oponente no puede modificar un mensaje interceptado y no se puede generar un mensaje falso.

Funciones Criptográficas HASH : Protegiendo el hash

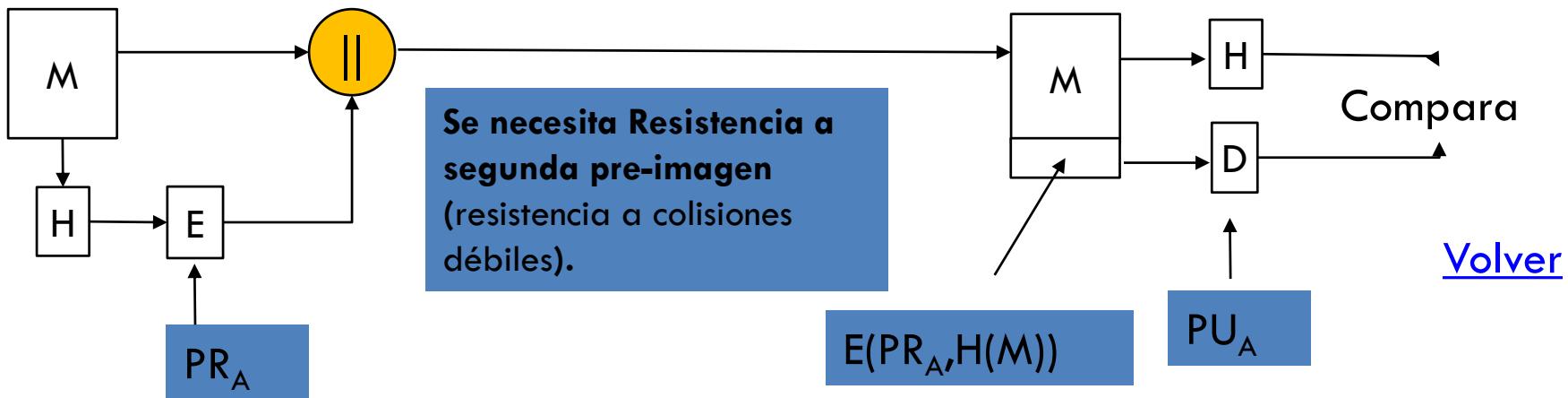
Ejemplos de Autenticación de Mensaje: SIN CIFRAR



- La **confidencialidad** se puede añadir a la aproximación del método anterior mediante el cifrado de la mensaje completo más el código hash.

Funciones Criptográficas HASH: Firma Digital

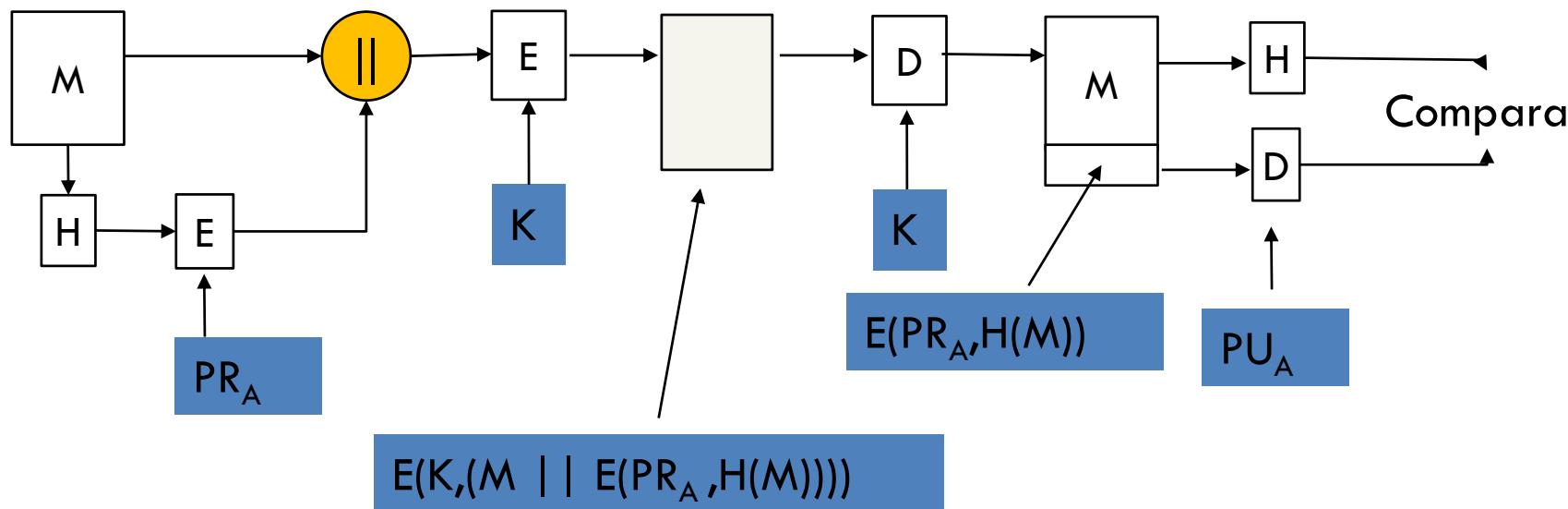
- El funcionamiento es similar a la de la MAC.
- El valor de hash de un mensaje se cifra con la clave privada de un usuario.
- Cualquiera que conozca la clave pública del usuario puede verificar la integridad del mensaje.
- Un atacante que quiere modificar el mensaje tendría que saber la clave privada del usuario.
- El código hash se cifra mediante el cifrado de clave pública con la privada del remitente. Esto proporciona la **autenticación**. También proporciona una **firma digital**, ya que sólo el remitente podría haber producido el cifrado hash de código. De hecho, esta es la esencia de la técnica de firma digital



Funciones Criptográficas HASH

Firma Digital y Confidencialidad

- Si se desea la **confidencialidad**, así como una firma digital, entonces el mensaje más el código hash de clave privada cifrada se puede cifrar utilizando una simétrica clave secreta. Esta es una técnica común.



Funciones Criptográficas HASH: Otras funcionalidades

- Se suelen utilizar para crear un archivo de **contraseñas** de un solo sentido.
- Se utiliza un esquema en el que un hash de la contraseña se almacena en un sistema operativo en lugar de la propia contraseña.
- Por lo tanto, la contraseña actual no es recuperable por un hacker que tenga acceso al archivo de contraseñas:
 - Cuando un usuario introduce una contraseña, entonces es el hash de la contraseña el que se compara con el valor hash almacenado para su verificación.
 - Este enfoque de la protección de contraseña es utilizada por la mayoría de los sistemas operativos.

Funciones Criptográficas HASH: Otras funcionalidades

- Las funciones hash se pueden utilizar para la **detección de intrusiones** y detección de virus:
 - Se almacena la $H(F)$ para cada archivo en un sistema de ficheros en un lugar seguro.
 - Uno puede después puede determinar si un archivo ha sido modificado por recalcular de nuevo el $H(F)$ ya que un intruso podría haber cambiado F sin haber cambiado $H(F)$.
- Una función de hash criptográfica se puede utilizar para construir una **función pseudoaleatoria** (por ejemplo SHA-3).
- O un generador de números pseudoaleatorios.
- Una aplicación común de las funciones pseudoaleatorias basadas en hash es su utilización para la generación de las claves simétricas.

Funciones Criptográficas HASH: Requerimientos y Seguridad

- Antes ver dos conceptos:
 - **pre-imagen**
 - **colisión**
- Para un valor hash $h = H(x)$, decimos que x es la imagen inversa de h (**o pre-imagen**, recordar que la imagen en una función es el campo de valores o rango de una función, i.e $\text{Im}_f := \{y \in Y \mid \exists x \in X, f(x) = y\}$).
- Es decir, la pre-imagen x es un bloque de datos cuya función hash, utilizando la función H , es h .
- Debido a que **H es un mapeo muchos-a-uno**, para cualquier valor determinado hash h , habrá en general **múltiples pre-imágenes**.
- Una **colisión** se produce cuando tenemos $x \neq y$, y además las imágenes son iguales, $H(x) = H(y)$.
- Como estamos usando funciones hash para la integridad de los datos, las colisiones son claramente indeseables.

Funciones Criptográficas HASH: Requerimientos y Seguridad

- Las propiedades deseables de una función criptográfica HASH:
 1. **Tamaño de entrada variable:** H se puede aplicar a un bloque de datos de cualquier tamaño.
 2. **Tamaño de salida fija:** H produce una salida de longitud fija.
 3. **Eficiencia:** $H(x)$ es relativamente fácil de calcular para cualquier x dado, siendo posible hacer ambas implementaciones de hardware y software.
 4. **Resistente a pre-imagen** (propiedad solo de ida, o propiedad one-way): Para cualquier valor de hash dado, h , es computacionalmente inviable encontrar una pre-imagen y tal que $H(y) = h$.
 5. **Resistente a la segunda pre-imagen** (resistencia a colisiones débiles): para cualquier bloque dado x , es computacionalmente inviable encontrar un $y \neq x$ con $H(y) = H(x)$.
 6. **Resistencia a colisiones** (resistente a colisiones fuertes): Es computacionalmente imposible encontrar cualquier par (x, y) tales que $H(x) = H(y)$.
 7. **Pseudoaleatoriedad:** La salida de H cumple las pruebas estándares para pseudoaleatoriedad.

Funciones Criptográficas HASH: Requerimientos y Seguridad

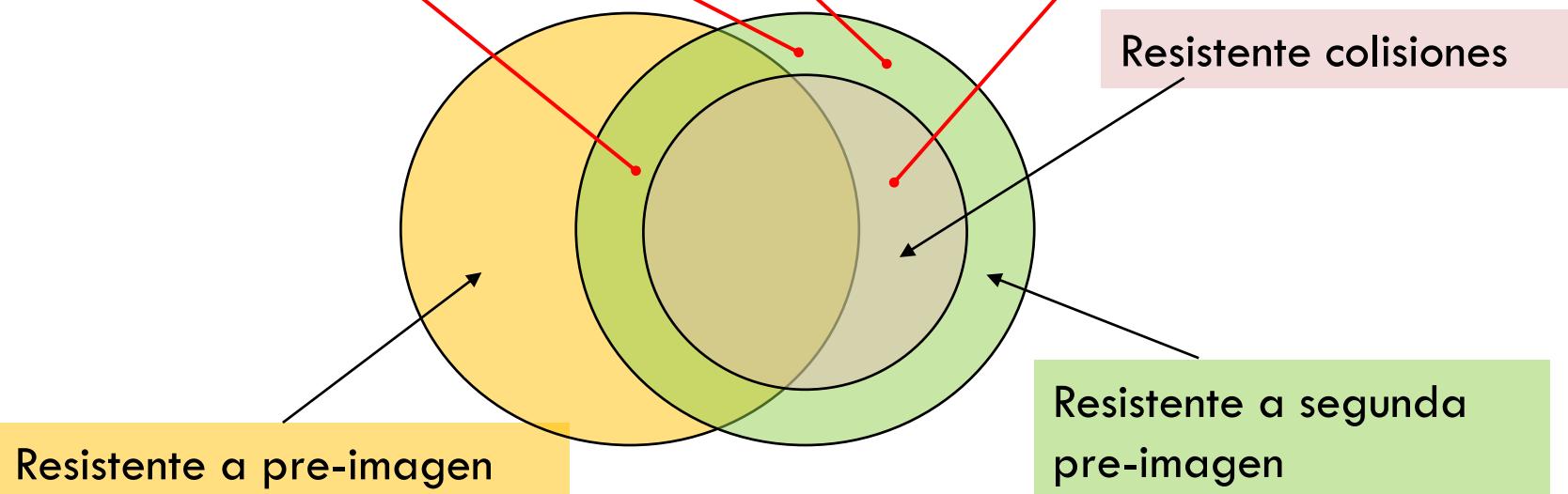
- **Las tres primeras propiedades** son requisitos para la **aplicación práctica** de un función hash.
- La **cuarta propiedad de imagen inversa resistente**, es la propiedad de un solo sentido: es fácil para generar un código dado a un mensaje, pero prácticamente imposible generar un mensaje dado un código. Por ejemplo, esta propiedad es importante si la técnica de autenticación implica el uso de un valor secreto o **secreto compartido** como el en [el protocolo anterior](#).
- La **quinta propiedad, segunda pre-imagen resistente**, garantiza que es imposible encontrar un mensaje alternativo con el mismo valor hash a un mensaje dado. Esta propiedad **evita la falsificación** cuando se utiliza un código hash cifrado como [habíamos visto anteriormente](#). Si esta propiedad no se cumpliese, un atacante sería capaz de:
 - Observar o interceptar un mensaje más su código hash cifrado
 - Generar un código hash no cifrado del mensaje (lo podemos hacer ya que la función hash es pública y el mensaje puede ir libre como en los protocolos anteriores).
 - Generar un mensaje alternativo con el mismo código hash (notar que como es el mismo h no hace falta saber la clave secreta del cifrado).

Funciones Criptográficas HASH: Requerimientos y Seguridad

- Una función de hash que satisface las **cinco primeras propiedades** se denomina **función de hash débil**. Si cumple la **sexta propiedad**, resistente a las colisiones se conoce como una **función de hash fuerte**.
- Una función **hash fuerte protege de un ataque en el que una de las partes genera un mensaje para que otra persona lo firme** (ataque del cumpleaños).
 - Supóngase que Bernardo escribe un mensaje de reconocimiento de Alicia de una deuda (es decir que Alicia le debe a Bernardo una cierta cantidad), y lo envía a Alicia para que lo firme (notar que solo firma el hash).
 - Pero Bernardo puede encontrar dos mensajes con el mismo hash, uno de los cuales requiere que Alicia pague una pequeña cantidad (la deuda verdadera) y el otro es un mensaje que reconoce una gran deuda.
 - Alicia firma el primer mensaje (pequeña deuda) y Alicia ya está tranquila, pero Bernardo es capaz de afirmar que el segundo mensaje de la deuda grande es auténtico, ya que había encontrado una colisión.

Funciones Criptográficas HASH: Requerimientos y Seguridad

- Una función que es resistente a colisiones también es resistente a segunda pre-imagen, pero al contrario no es necesariamente cierto.
- Una función puede ser resistente a colisiones, pero no tiene porque ser resistente pre-imagen y viceversa.
- Una función puede ser resistente a pre-imagen, pero no resistente a la segunda pre-imagen y viceversa.



Funciones Criptográficas HASH: Ataques

- Al igual que con los algoritmos de cifrado, hay dos categorías de ataques: ataques de **fuerza bruta** y **criptoanálisis**.
- Un ataque de **fuerza bruta** no depende en el algoritmo específico, pero sólo depende de la longitud **de bits de la salida** del hash.
- Un ataque por **criptoanálisis**, por el contrario, es un ataque basado en las **debilidades del algoritmo** criptográfico.
- Por ejemplo el **Ataque de cumpleaños** explota la paradoja del cumpleaños: la posibilidad de que en un grupo de personas dos compartirán el mismo cumpleaños no es tan pequeña (sólo se necesitan 23 personas para una $\text{Pr} > 0.5$).
- Se puede generalizar el problema para buscar coincidente de dos conjuntos, y se necesitan solo probar $2^{m/2}$ para conseguir una colisión en un hash de m bits.
- Yuval (Gideon Yuval - How to Swindle Rabin Cryptologia3:187-189, 1979) propuso la estrategia mostrada a explotar la paradoja del cumpleaños en un ataque resistente a colisión. Tener en cuenta que la creación de muchos mensaje de variantes es relativamente fácil, ya sea por reformulación o simplemente variando la cantidad de espacios en blanco en el mensaje. Todo lo cual indica que se necesitan mayores salidas de MACs / Hashes

Funciones Criptográficas HASH: Ataques

¿Cómo funciona el **ataque de cumpleaños**?

- La fuente (A) está dispuesto a firmar un mensaje legítimamente añadiendo el código de m bits hash apropiado y cifrar ese código hash con la clave privada de A.
- Un rival genera $2^{n/2}$ x' variaciones de x, todos con esencialmente el mismo significado, y almacena los mensajes y sus valores de hash. Hay que tener en cuenta que la creación de muchos mensajes de variantes es relativamente fácil, ya que simplemente variando la cantidad de espacios en blanco en el mensaje o añadiendo conectores que no varían el significado se puede obtener (m es el "digest").
- El rival genera un mensaje fraudulento para el que se desea la firma de A y todas sus variaciones.
- Los dos conjuntos de mensajes se comparan para encontrar un par con el mismo hash.
- Esto es posible por la paradoja de cumpleaños: para un hash de tamaño n es posible encontrar una colisión generando solo $2^{n/2}$ "hashes".
- El rival ofrece la variación válida a A para que la firma (el hash cifrado con la clave privada) pueda ser unido a la variación fraudulenta para la transmisión al destinatario previsto. Debido a que las dos variantes tienen el mismo código hash, producirán la misma firma y el oponente tiene asegurado el éxito a pesar de que la clave de cifrado no se sabe.

Funciones Criptográficas HASH: Estructura General

- La estructura general es conocida como una **función resumen repetida**, y se propuso por Merkle:
 - Merkle,R. *Secrecy, Authentication, and Public Key Systems.* Ph.D.Thesis, Stanford University, June 1979.
 - Merkle,R.“One Way Hash Functions and DES.” *Proceedings, CRYPTO '89*, 1989; published by Springer-Verlag.
- Es la estructura de la mayoría de las funciones de hash en uso hoy en día, incluyendo SHA, MD4, MD5, SHA1, SHA2, SHA3, etc.
- La función hash toma un mensaje de entrada y se divide en bloques de tamaño fijo. Si es necesario, el bloque final se rellena con bits que falten.

Funciones Criptográficas HASH: Estructura General

- El bloque final también incluye el valor de la longitud total de la entrada a la función hash. La inclusión de la longitud hace que el trabajo del oponente más difícil.
- El algoritmo implica el uso repetido de una función de compresión, f , que tiene dos entradas:
 - una entrada de la etapa anterior, variable de encadenamiento,
 - y otra entrada el bloque del mensaje,
- y produce una salida para la siguiente función, que es la variable de encadenamiento para el siguiente.
- Al comienzo de hash, hay un valor inicial (IV) que se especifica como parte del algoritmo. Al final el valor de la variable de encadenamiento es el valor hash.

Funciones Criptográficas HASH

Estructura General: Merkle–Damgård

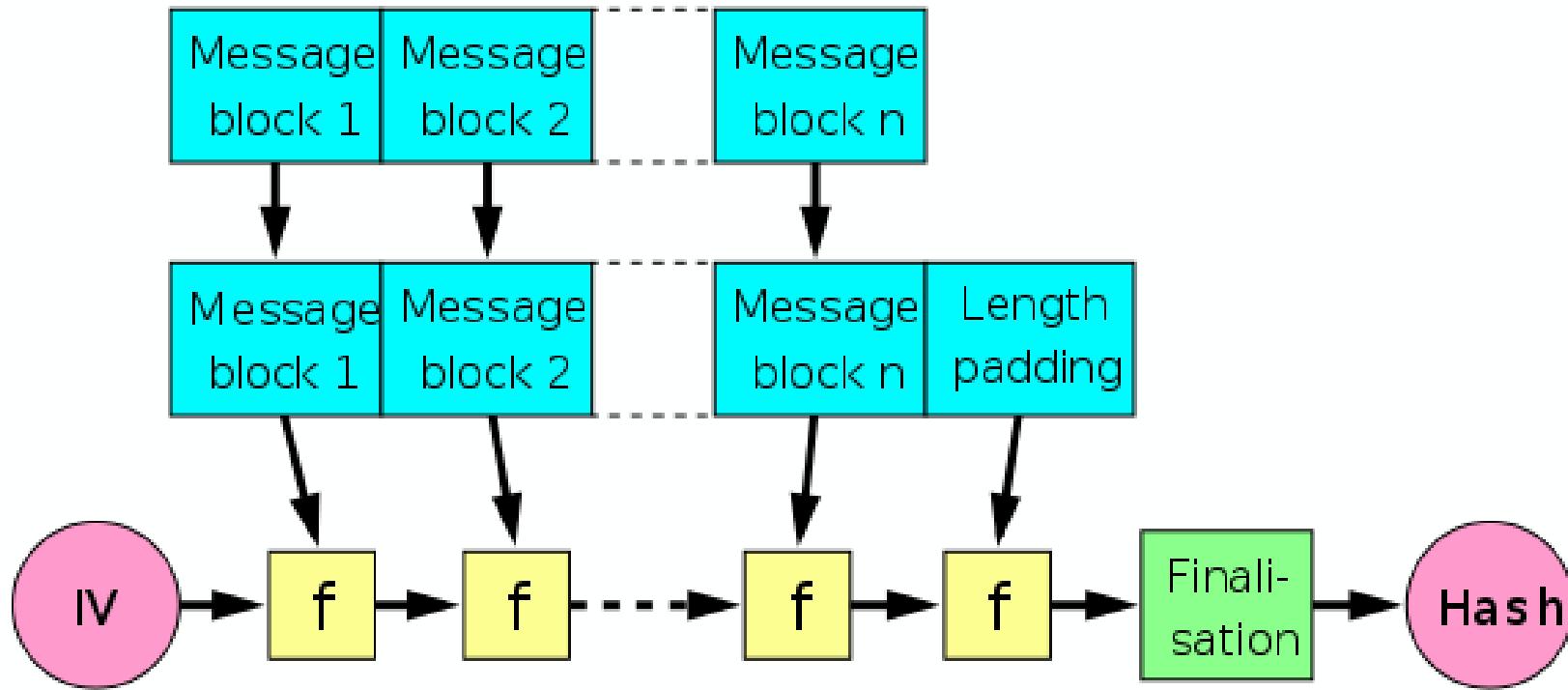
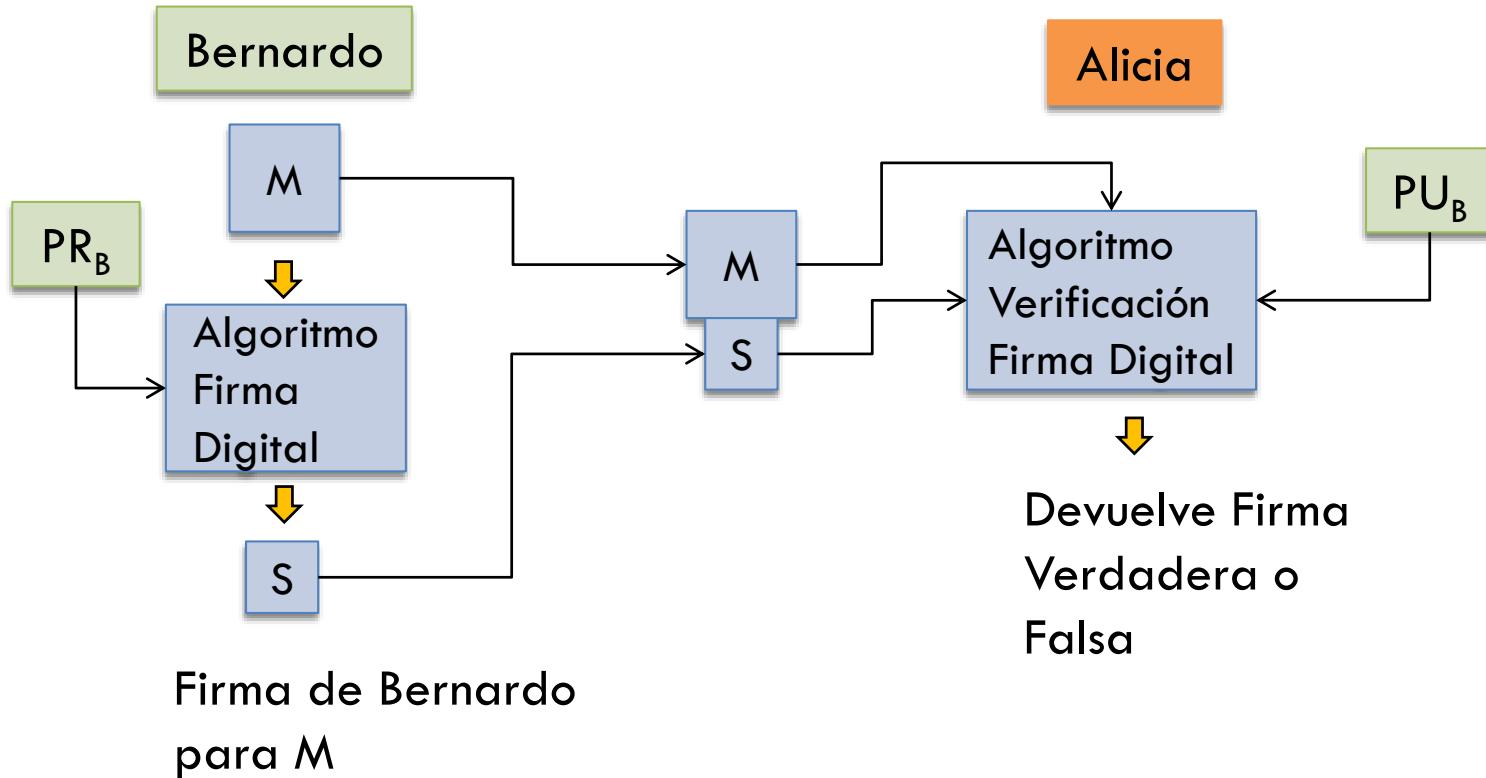


Imagen extraída de http://en.wikipedia.org/wiki/File:Merkle-Damgard_hash_big.svg

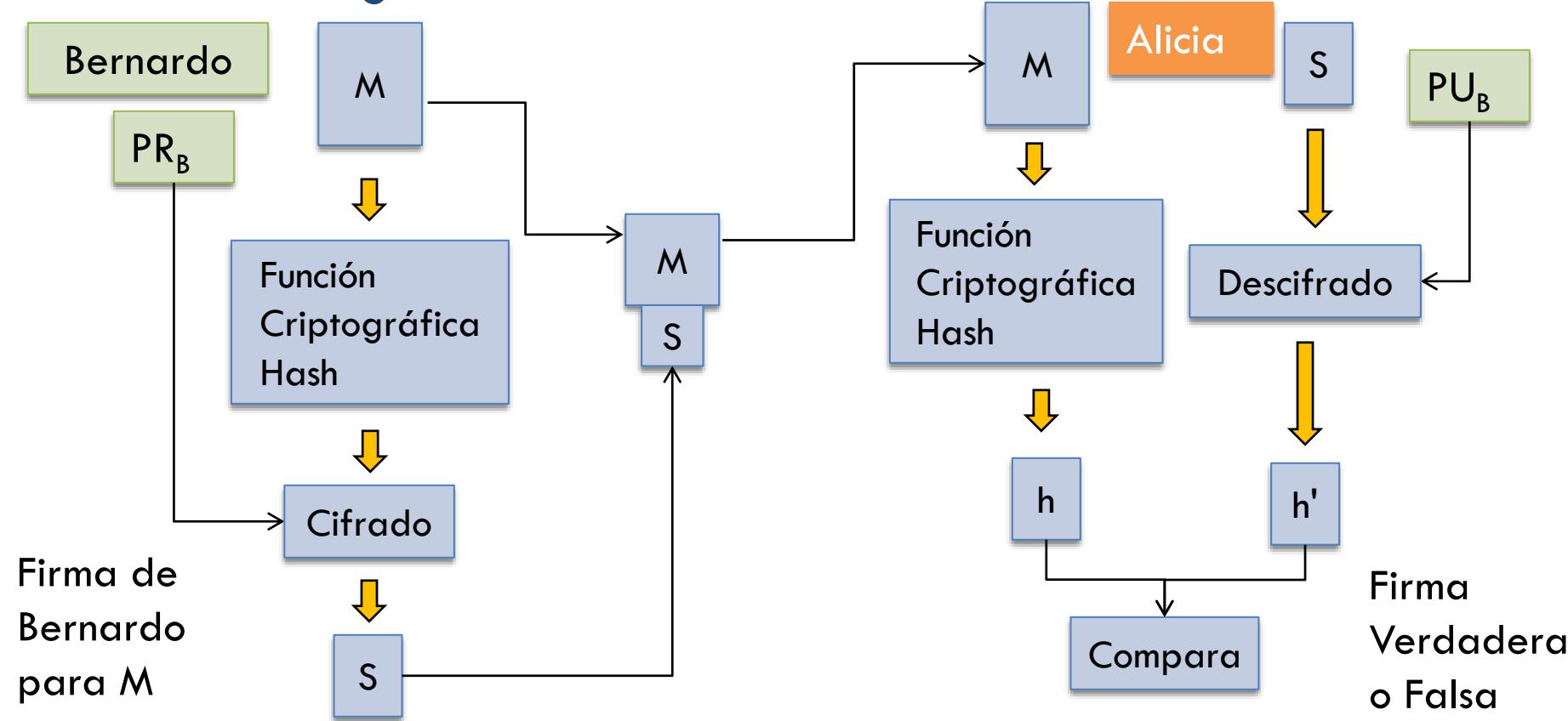
Funciones Criptográficas HASH: Basadas en el modo CBC

- Funciones hash basados en el uso de una técnica de encadenamiento de bloques de cifrado, pero sin la clave secreta.
- Rabin (Rabin, M. "Digitalized Signatures." *Foundations of Secure Computation*, DeMillo, R.; Dobkin, D.; Jones, A.; and Lipton, R., eds. New York: Academic Press, 1978) divide un mensaje M en bloques de tamaño fijo M_i , y usa un sistema de cifrado simétrico como DES para calcular el código hash.
 - Usando $H_0 = 0$ y cero de relleno de bloque final.
 - Calcular: $H_i = E(M_i, H_{i-1})$.
 - Usa el resultado del bloque final como el valor hash.
 - De manera similar a CBC, pero sin una clave.
- Este esquema está sujeta al ataque de cumpleaños, y si el algoritmo de cifrado DES es y se produce sólo un código hash de 64 bits, entonces el sistema es vulnerable.
- También vulnerable a Man-in-the-middle-attack.

Firmas Digitales: Proceso General



Firmas Digitales: Proceso General



Firmas Digitales: Propiedades

- La **autenticación de mensajes protege dos partes** que intercambian mensajes de cualquier tercero.
- Sin embargo hay que tener en cuenta que **no protege a las dos partes de uno contra otro**.
- Se pueden dar varias formas de **disputa entre las dos son partes**.
- En consecuencia, en situaciones donde **no hay confianza plena** entre el remitente y el receptor se necesita algo más que la autenticación.
- Una solución atractiva para este problema es la **firma digital**.
- La firma digital debe tener la siguiente propiedades:
 - Se debe **verificar el autor, la fecha y hora de la firma**.
 - Se debe **autenticar el contenido** en el momento de la firma.
 - Debe ser **verificable por terceros**, para resolver disputas.
- Por lo tanto, la función **firma digital incluye la función de autenticación**.

Firmas Digitales: Propiedades

- La firma debe ser un **patrón de bits que depende del mensaje que se firmó**.
- La firma debe **usar algo de información única del emisor** para evitar tanto la falsificación y la negación.
- La firma digital debe ser relativamente **fácil de producir**.
- Debe ser relativamente **fácil de reconocer y comprobar la firma digital**.
- Debe ser **computacionalmente imposible de falsificar** una firma digital, ya sea por la construcción de un nuevo mensaje para una firma digital existente o construyendo una firma digital fraudulenta para un mensaje dado.
- Debe ser **práctico guardar una copia de la firma digital** en cualquier tipo de almacenamiento.
- Una **función de hash segura**, incrustada en el protocolo anterior es la base para cumplir todos estos requerimientos.
- Sin embargo, se debe **tener cuidado en el diseño del protocolo de firma**.

Firmas Digitales: DSA, NIST

- El NIST publicó el procesamiento de Información Federal estándar FIPS 186, conocida como DSA (Digital Signature Algorithm).
- Se usa en el Estándar de Firma Digital (DSS).
- **DSS=DSA.**
- DSA hace uso de SHA (Secure Hash Algoritmo).
- La DSA fue propuesta originalmente en 1991 y revisado en 1993, en relación con la seguridad del esquema.
- Hubo una revisión menor aún más en 1996.
- En 2000, un versión ampliada de la norma se publicó como FIPS 186-2, posteriormente actualizado FIPS 186-3 en 2009, ahora FIPS 186-4 en 2013.
- Esta última versión también incorpora la firma digital algoritmos basados en RSA y en criptografía de curva elíptica.
- El DSA utiliza un algoritmo que está diseñado para proporcionar sólo la función de firma digital. Sin embargo, es una técnica de clave pública.

Firmas Digitales: DSA, NIST

- **DSA está basado en los esquemas Elgamal y en el de Schnorr de firmas digitales:**
 - Elgamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." Proceedings, Crypto 84, 1984.
 - Elgamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." IEEE Transactions on Information Theory, July 1985.
 - Schnorr, C. "Efficient Identification and Signatures for Smart Cards." EUROCRYPT, 1988.
 - Schnorr, C. "Efficient Signature Generation by Smart Cards." Journal of Cryptology, No. 3, 1991.
- El esquema de firma **Elgamal implica el uso de la clave privada** para el cifrado y la clave pública para el descifrado.
- El esquema **Schnorr minimiza la cantidad dependiente del mensaje** del cálculo necesario para generar una firma:
 - El computo principal para la generación de firma no depende de el mensaje y se puede hacer durante el tiempo de inactividad de los procesadores (la potenciación que es costosa).
- La generación de la firma requiere solo multiplicar dos números enteros (s^*e).
- También existen otros esquemas de DSA basados en curvas elípticas (ECDSA).

Distribución de Claves Simétricas

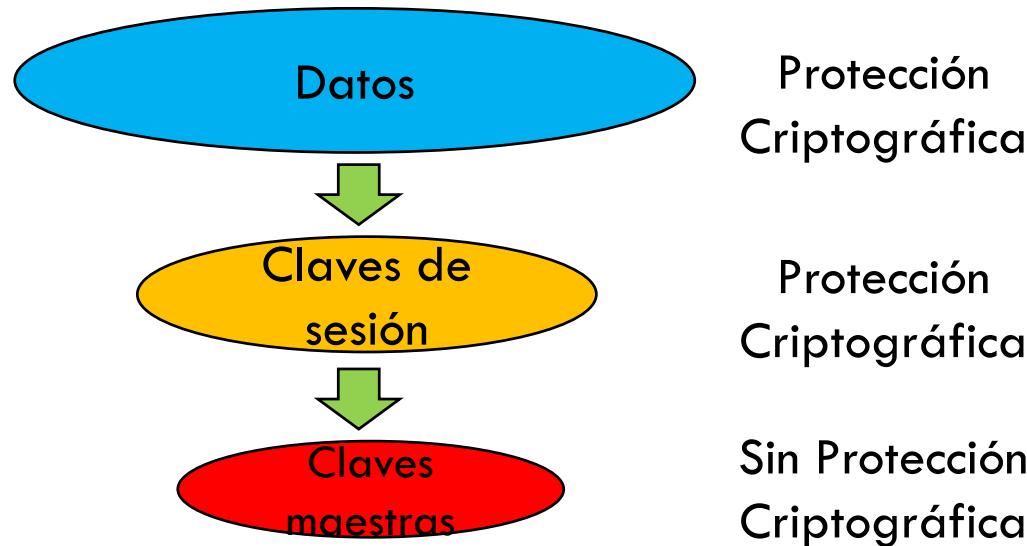
- Término que se refiere a los **medios de la entrega de la llave de dos partes** que desean intercambiar datos sin permitir que otros vean la clave.
- Para el cifrado simétrico, las dos partes deben compartir la misma clave, y esa **clave debe ser protegida contra el acceso de otras personas**.
- Los **cambios frecuentes de clave son deseables**.
- Para dos partes A y B, de distribución de claves puede ser:
 1. A puede seleccionar una clave y entregarla físicamente a B.
 2. Un tercero puede seleccionar la clave y entregarla físicamente a A y B.
 3. Si A y B se han comunicado anteriormente y recientemente con una clave, una de las partes puede transmitir la clave a la otra, cifrada mediante la clave antigua.
 4. **Si A y B cada uno tiene una conexión cifrada a un tercero C, este puede enviar una clave en los enlaces cifrados hacia A y B.**
- **El problema de distribución es complejo**, si tenemos **N usuarios** y queremos comunicar todos necesitamos **(N*(N-1)/2) claves**.
- Si el cifrado es realizado a nivel de aplicación, una clave se utiliza para cada pareja de puntos terminales.
- Así para 1000 puntos terminales necesitamos un poco menos de medio millón de claves (con 10000 aplicaciones necesitamos aprox. **50 millones de claves**).

Distribución de Claves Simétrica

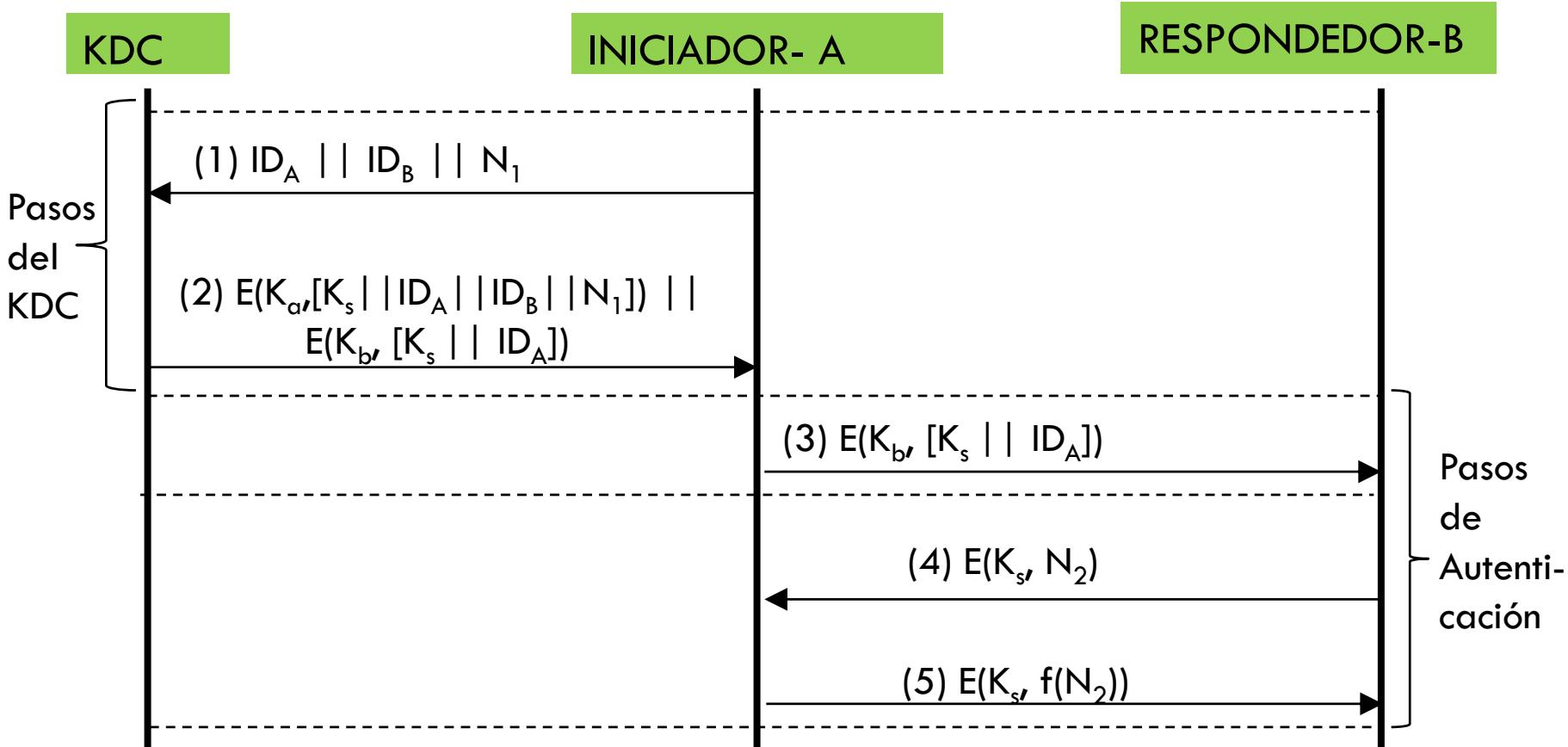
- Para el cifrado de extremo a extremo, se han implementado ampliamente alguna variación en la opción 4.
- En este esquema, un **centro de distribución de claves (KDC)** es responsable de la distribución llaves de pares de usuarios (hosts, procesos, aplicaciones, ets.) según sea necesario.
- Cada usuario debe compartir una clave única con el KDC para los propósitos de la distribución de claves.
- El uso de un KDC se basa en el uso de una **jerarquía de claves**.
- La comunicación entre los sistemas finales se cifra utilizando una clave temporal (**clave de sesión**).
- Típicamente, la clave de sesión se utiliza para la duración de una conexión lógica.
- Cada clave de sesión se obtiene desde el KDC a través de las mismas instalaciones de redes.
- Las claves de sesión se transmiten en forma encriptada, usando una **clave maestra** que es compartido por el KDC y un sistema final o usuario .
- Para cada sistema o usuario final, hay una única clave maestra que comparte con KDC.

Distribución de Claves Simétrica

- Por supuesto, estas llaves maestras deben ser distribuidos de alguna manera segura.
- Sin embargo, la magnitud del problema se reduce enormemente:
 - Recordar que si hay entidades N que se quieren comunicar por pares, entonces, como se mencionó, se necesitan $[N(N - 1)] / 2$ claves de sesión en un momento dado.
- Sin embargo ahora, sólo se requieren **N llaves maestras**, uno para cada entidad.
- Las claves maestras se pueden distribuir de alguna forma no criptográfica, así, por ejemplo: la entrega física, etc.



Distribución de Claves Simétrica KDC



Distribución de Claves Simétrica Jerarquías de KDCs

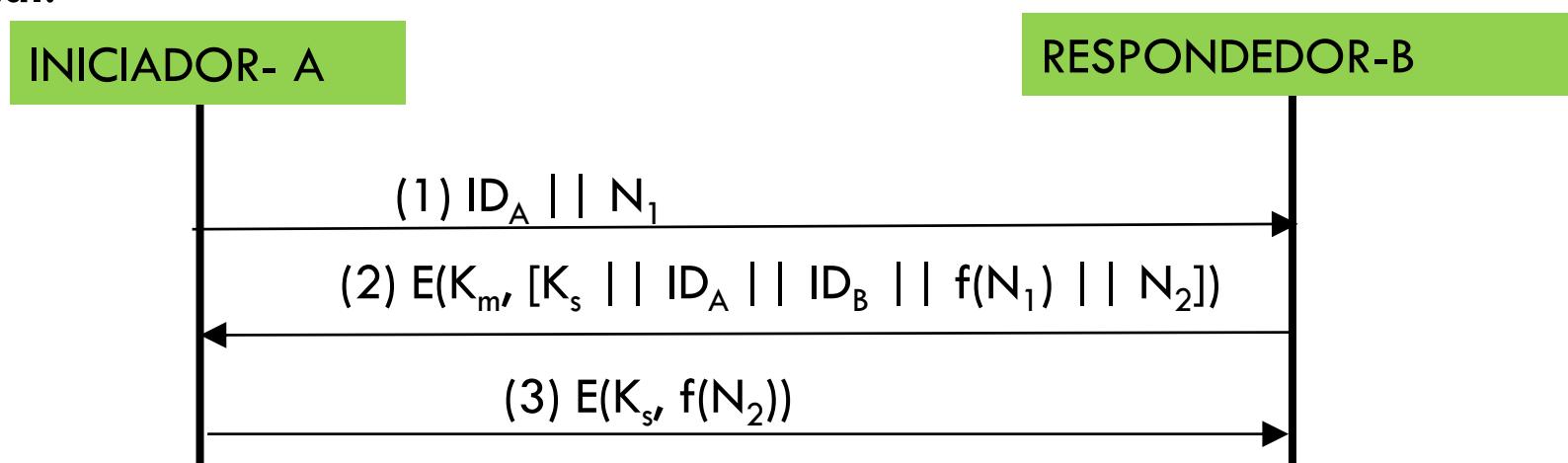
- El nuevo concepto de “**nonce**” es fundamental (**autenticando con “nonces”**)
- Darse cuenta que el mensaje inicial incluye la identidad de A y B y un identificador único, N1, para esta transacción, que es el “nonce”.
- Un “nonce” puede ser una marca de tiempo, un contador o un número aleatorio. El requisito mínimo es que es que difiera con cada solicitud, por lo tanto de un solo uso para que sea único. Además, para evitar el falseo del mensaje, debe ser difícil que un oponente adivine ese “nonce”.
- Así, un **número aleatorio** es una buena elección.
- Así en el paso 2, A puede verificar que su solicitud original no fue alterada antes de la recepción por el KDC y, debido al nonce, y el mensaje 2 no se origina por alguna repetición de alguna solicitud anterior por parte de un oponente.
- En el paso 4, utilizando la clave de sesión recién suministrada por el KDC para el cifrado, B envía un nuevo “nonce”, N2, a A.
- Además, usando K_s , A responde con $f(N2)$, donde f es una función que realiza alguna transformación de N2 (por ejemplo, añadir uno).
- Estos pasos anteriores aseguran a B que el mensaje original que recibió (paso 3) no fue una repetición por algún oponente.
- Hablando estrictamente, la distribución de claves real implica sólo los pasos 1 a 3.
- Los pasos 4 y 5, además del paso 3 a través de K_b , realizan una función de autenticación.

Distribución de Claves Simétrica Jerarquías de KDCs

- **No es necesario limitar la función de distribución de claves a un solo KDC.**
- Para las redes muy grandes, puede no ser práctico.
- Como una alternativa, se debe establecer una **jerarquía de KDCs**:
 - Por ejemplo, debería haber KDCs locales para un pequeño dominio de la interconexión de redes en general, tales como una sola LAN. Así para la comunicación entre entidades dentro del mismo dominio local, el KDC local es responsable de la distribución de claves .
 - Si dos entidades en diferentes dominios desean una clave compartida, los **KDC locales** correspondientes se puede comunicar a través de un **KDC global**.
 - En este caso, cualquiera de los tres KDC involucrados puede seleccionar la clave.
- El concepto jerárquico puede extenderse a tres o incluso más capas, dependiendo del tamaño de la red.
- Un esquema jerárquico reduce al mínimo el esfuerzo que supone la distribución de las claves principal, porque la mayoría de las llaves maestras son aquellas que comparten un KDC local con sus entidades locales.
- Por otra parte, **este esquema limita el daño de un KDC defectuoso o atacado solo a su área local**.

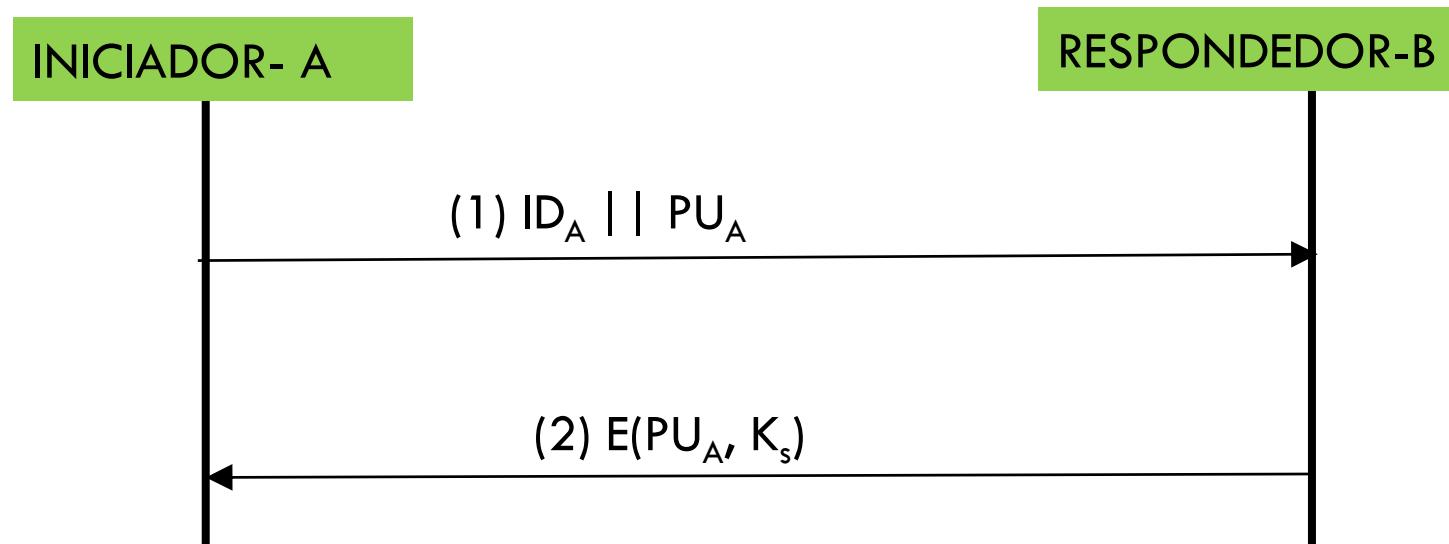
Distribución de Claves Simétrica: Claves Descentralizadas

- El uso de un centro de distribución de claves impone el requisito de que el KDC sea de confianza. Este requisito se puede evitar si la distribución claves es totalmente descentralizada.
- Como se necesita una Km, para N usuarios se necesitaran **(N*(N-1)/2) claves**.
- Aunque la **plena descentralización no es práctica** para redes más grandes para utilizar solo el cifrado simétrico, puede ser **útil** dentro de un **contexto local**.



Simple Distribución de Claves

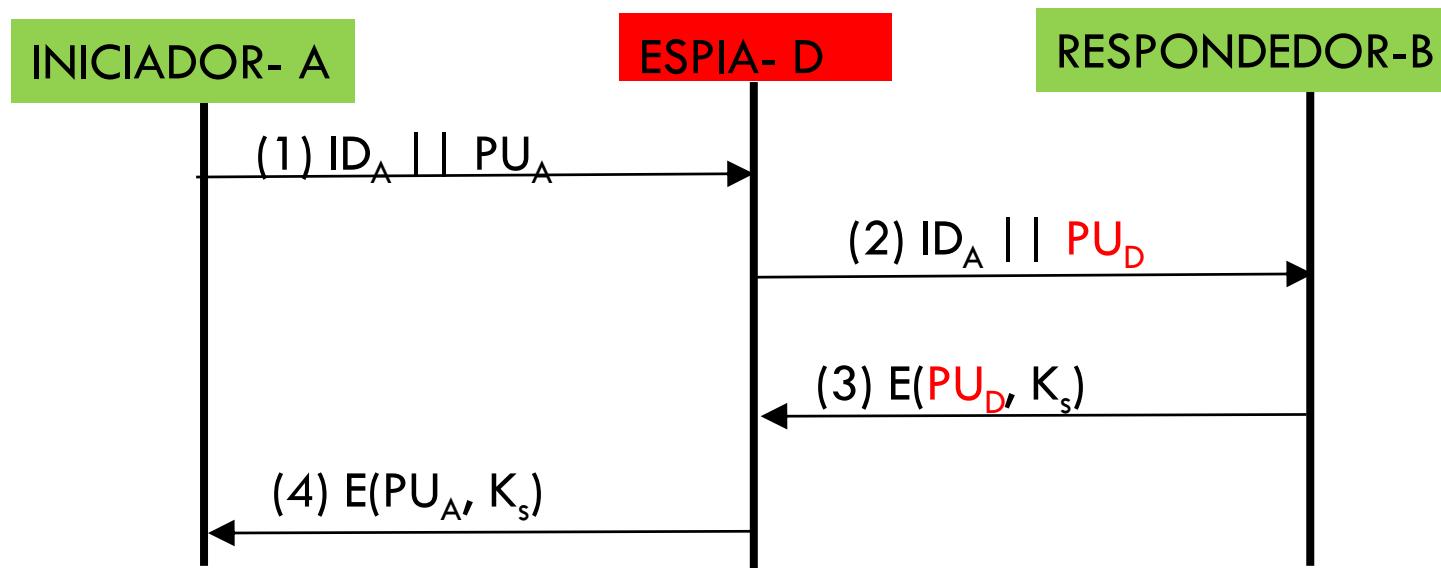
- A pesar de su simplicidad, este es un protocolo es muy atractivo.
- **No existen claves maestras, K_m , antes del inicio de la comunicación y tampoco después de la finalización de la comunicación.**
- Por lo tanto, el riesgo de compromiso de las claves es mínima.
- Al mismo tiempo, la comunicación no puede ser espiada (no se puede averiguar K_s).



Simple Distribución de Claves: Man-in-the-Middle Attack

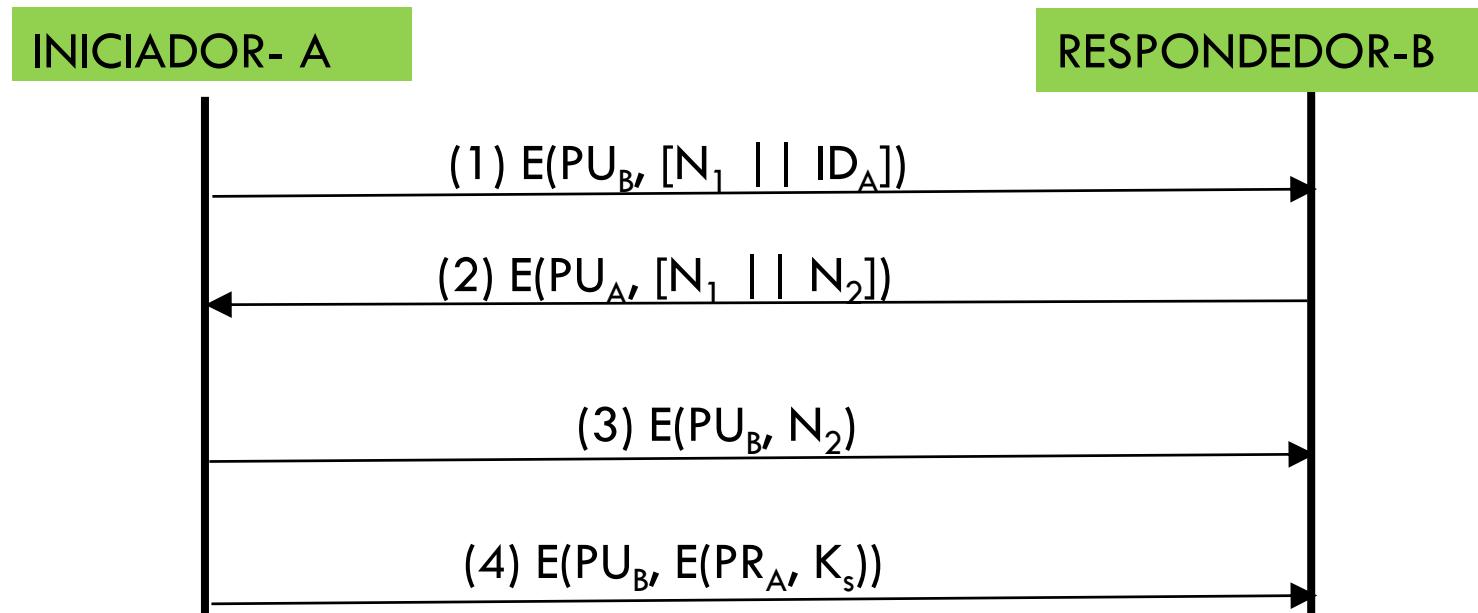
- El problema con este simple esquema es el Man-in-the-Middle Attack.
- Si un adversario D, tiene el control del canal de comunicación que interviene, entonces D puede poner en peligro la comunicación de la siguiente manera, sin saber B que está siendo detectado.
- A y B no tienen noción de que existe un espía D que conoce la clave K_s .

En el paso 3 la clave de sesión es $K_s = D(PR_D, E(PU_D, K_s))$



Distribución de Clave Secreta con Confidencialidad y Autenticación

- Basado en [Needham, R., and Schroeder, M. "Using Encryption for Authentication in Large Networks of Computers." Communications of the ACM, December 1978](#) el siguiente protocolo proporciona protección contra ataques activos y pasivos



Distribución de Claves Híbrido Simétrico y Asimétrico

- El iniciador A utiliza la clave pública de B para cifrar un mensaje que contiene un identificador de A (ID_A) y un “nonce” (N_1), que se utiliza para identificar esta transacción única.
- B envía un mensaje a A cifrado con PU_a y que contiene el “nonce” N_1 recibido de A más un nuevo “nonce” generado por B (N_2).
 - Debido a que sólo B podría haber descifrado el mensaje 1, la presencia de N_1 en el mensaje 2 asegura a A que es B el único que recibió el mensaje 1 descifrándolo con la clave privada de B.
- A devuelve N_2 cifrado usando la clave pública de B, para asegurar a B que es el que ha recibido el mensaje (es el único que puede descifrar el mensaje 2 con su clave privada, PR_A).
- A continuación A selecciona una clave secreta K_s y envía el mensaje $M = E(PU_b, E(PR_a, K_s))$ a B. De esta forma se asegura que sólo B puede leerlo.
- Con la clave privada de A se asegura que sólo A podría haber enviado.
- Por último B calcula $D(PU_a, D(PR_b, M))$ para recuperar la clave secreta K_s .

Distribución de Claves Híbrido Simétrico y Asimétrico

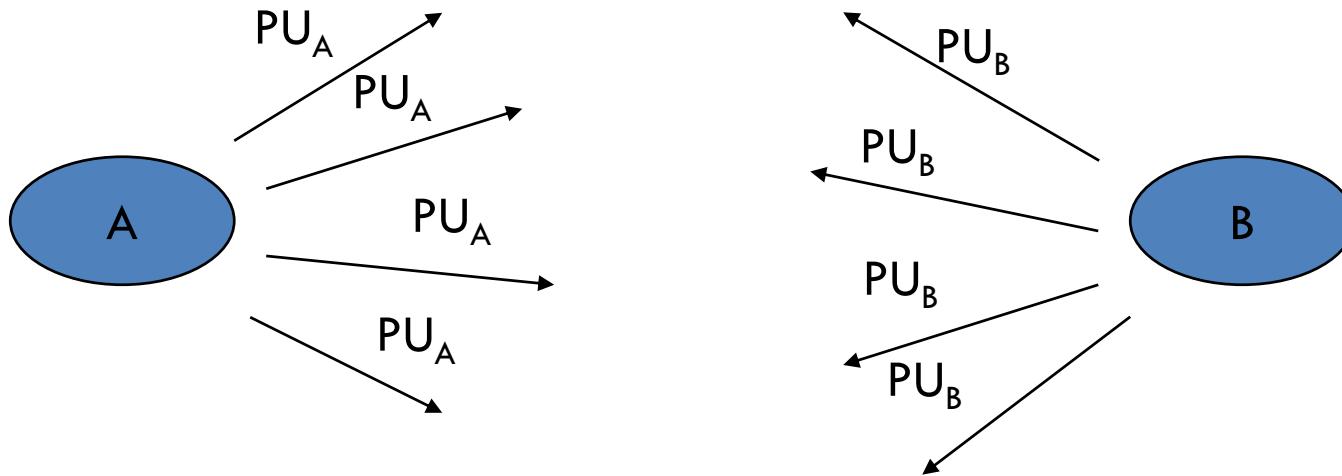
- Por ejemplo se utiliza en los mainframes de IBM:
- Consiste en el uso de un centro de distribución de claves (KDC) que comparte una clave maestra secreta con cada usuario y distribuye claves secretas de sesión cifradas con la clave maestra.
- Sin embargo utiliza un esquema de clave pública para distribuir las claves maestras.
- La justificación para el uso de este enfoque es la siguiente:
 - **Rendimiento:**
 - Hay muchas aplicaciones, en el que las claves de sesión cambian con frecuencia.
 - Por tanto la distribución de las claves de sesión por el cifrado de clave pública podrían degradar el rendimiento general del sistema (debido a la relativamente alta carga computacional de cifrado de clave pública y el descifrado).
 - Con este enfoque, se utiliza el cifrado de clave pública sólo de vez en cuando para actualizar la llave maestra entre un usuario y el KDC.
 - **Compatibilidad con versiones anteriores:**
 - El esquema híbrido se superpone muy fácilmente en un esquema ya existente de KDCs, con interrupción y cambios mínimos software.

Distribución de Claves Públicas

- Se han propuesto varias técnicas para la distribución de claves públicas.
- Prácticamente todas estas propuestas se pueden agrupar en los siguientes esquemas generales:
 - Anuncio público.
 - Directorio disponible públicamente.
 - Autoridad de clave pública.
 - Certificados de clave pública.

Anuncio Público

- Aunque este enfoque es conveniente, aunque tiene una **gran debilidad**. Cualquier persona puede **falsificar un tal anuncio público**. Es decir, algún usuario podría hacerse pasar por el usuario A y enviar una clave pública a otro participante o difundir dicha clave pública.
- Hasta el momento en que el usuario A descubre la falsificación y alerta a los demás participantes, el falsificador es capaz de leer (no descifrarlos) todos los mensajes cifrados destinados a A y puede utilizar las claves públicas para autenticación (en un protocolo que se cifre primero con la privada).



Directorio Disponible Públicaamente

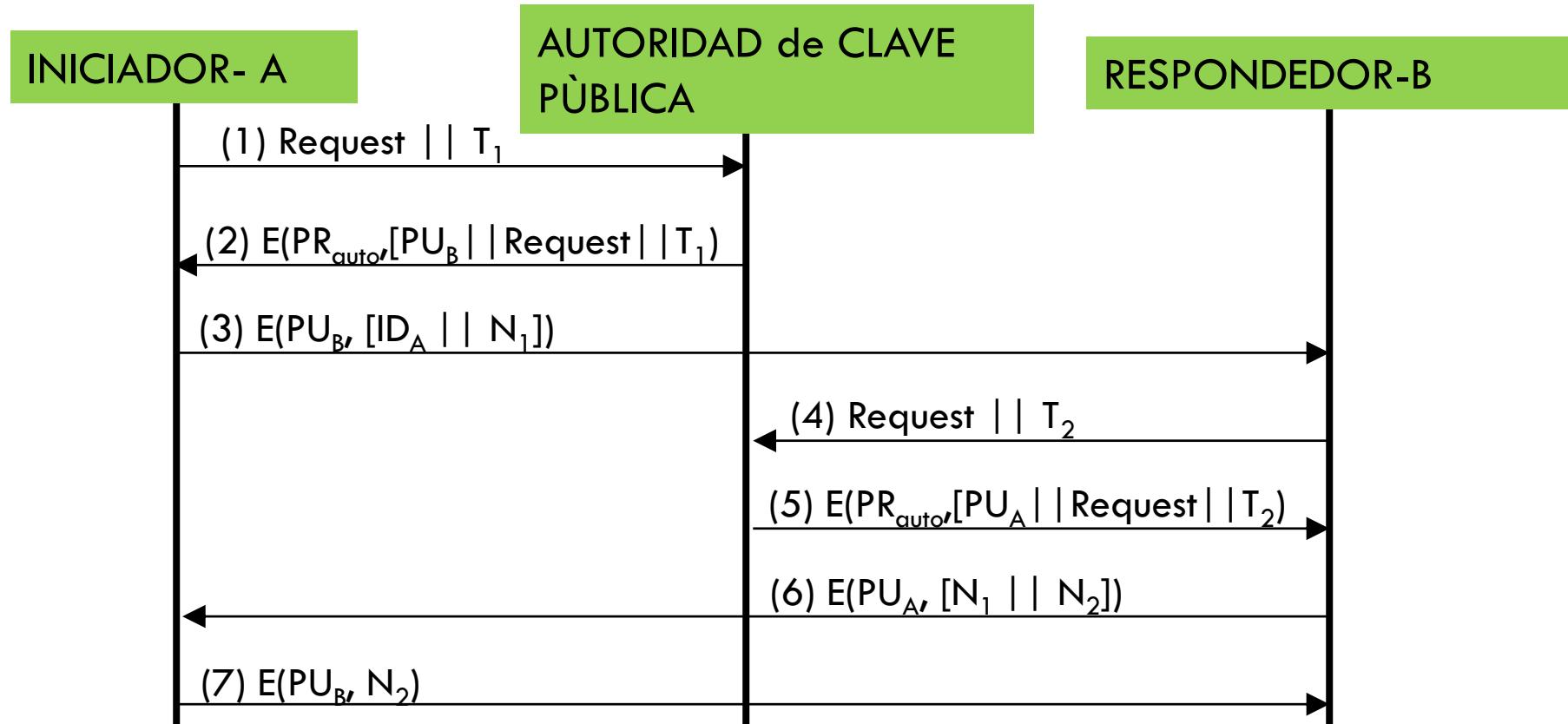
- Un mayor grado de seguridad se puede lograr mediante el mantenimiento de directorio dinámico público de claves públicas.
- El mantenimiento y distribución del directorio tendría que ser la responsabilidad de alguna entidad u organización de confianza y debería incluir los siguientes elementos:
 - La autoridad mantiene un directorio con una entrada {nombre, clave pública} para cada participante.
 - Cada participante registra una clave pública con la autoridad del directorio.
 - El registro tendría que ser en persona o por algún tipo de seguro de comunicación autenticada.
 - Un participante podrá sustituir la clave existente por una nueva en cualquier momento, ya sea por el deseo de reemplazar una clave pública que ya ha sido utilizado para una gran cantidad de datos, o porque la clave privada correspondiente ha sido comprometida de algún modo.
 - Los participantes también pueden acceder al directorio en forma electrónica. Para este propósito, es obligatoria la comunicación autenticada de la autoridad con el participante.

Directorio Disponible Públicamente

- Este sistema está claramente más seguro que los anuncios públicos individuales.
- Pero todavía tiene vulnerabilidades:
 - Si un adversario tiene éxito en la obtención o el cálculo de la clave privada de la autoridad de directorio, el adversario podría poner claves públicas falsificadas y, posteriormente, hacerse pasar por cualquier participante para espiar todos los mensajes enviados a cualquier participante.
 - Otra forma de lograr el mismo fin es alterando los registros llevados por la autoridad.



Autoridad de Clave Pública



Certificados de Clave Pública Autoridades Certificadoras (CA)

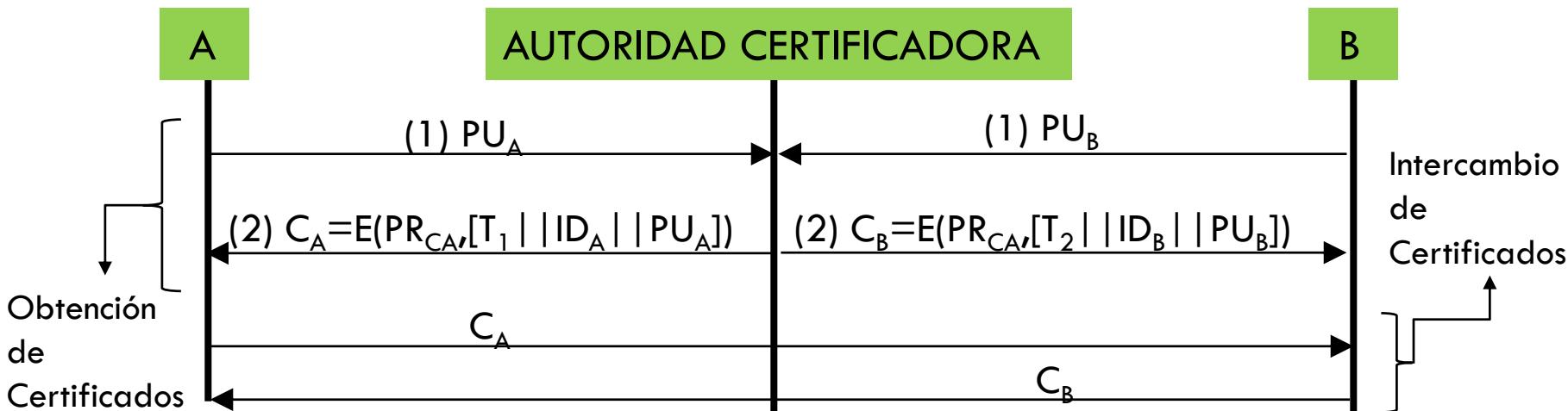
- Este escenario anterior es atractivo, sin embargo, tiene algunos **inconvenientes**:
 - La petición de clave pública a la **autoridad** podría ser un **cuello de botella** en el sistema, ya que un usuario siempre debe apelar a la autoridad de una clave pública para contactar con otro usuario.
 - Y como antes, el directorio de nombres y claves públicas gestionadas por la autoridad puede ser **vulnerable a la manipulación**.
- Un enfoque alternativo, sugerido por primera vez por Kohnfelder (Loren M Kohnfelder, Towards a Practical Public-key Cryptosystem, Thesis MIT, 1978), es el **uso de certificados** que pueden ser utilizados por los participantes para intercambiar claves sin contactar con una autoridad de clave pública, de una manera que es tan fiable como si las claves se obtuvieron directamente de una autoridad de clave pública.
- **En esencia, un certificado consta de (i) una clave pública, (ii) un identificador del propietario de la clave, (iii) y todo el bloque firmada por un tercero de confianza (CA).**
- Por lo general, el tercero es una autoridad de certificación (CA), como una agencia gubernamental o una institución financiera, que es la confianza de la comunidad de usuarios.

Certificados de Clave Pública Autoridades Certificadoras (CA)

- Un usuario puede presentar su clave pública a la autoridad certificadora de un modo seguro y obtener un certificado.
- El usuario puede entonces publicar el certificado.
- Cualquier persona que necesite la clave pública de este usuario puede obtenerla del certificado y verificar su validez a través de la firma de confianza adjunto.
- Un participante también puede transmitir su información clave a otra, mediante la transmisión de su certificado.
- Otros participantes puedan verificar que el certificado fue creado por la autoridad.
- Podemos colocar los siguientes **requisitos** en este esquema:
 - Cualquier participante puede leer un certificado para determinar el nombre y la clave pública del propietario del certificado.
 - Cualquier participante puede verificar que el certificado fue originado por la CA y además no es falso.
 - Sólo la autoridad de certificación puede crear y actualizar los certificados .

Certificados de Clave Pública Autoridades Certificadoras (CA)

- Todos estos requisitos se cumplen en la propuesta original en Kohnfelder (Loren M Kohnfelder, Towards a Practical Public-key Cryptosystem, Thesis MIT, 1978).
- Sin embargo Denning (Denning, D. Protecting public keys and signature keys. IEEE Computer. 1983) añadió el siguiente requisito adicional :
 - Cualquier participante puede verificar la vigencia del certificado.
- Cada participante suministra a la CA una clave pública y la solicitud del certificado. **La solicitud debe ser en persona o por alguna forma de comunicación segura y autenticada.**



Certificados de Clave Pública Certificados X.509

- La recomendación IUT-T de X.509 forma parte de la serie X.500 de recomendaciones que definir un servicio de directorio.
- El directorio es, un servidor o un conjunto distribuido de servidores que mantiene una base de datos de información sobre los usuarios.
- La información incluye una asignación de nombre de usuario a la dirección de red, así como otros atributos y la información sobre los usuarios.
- X.509 define un marco para la prestación de servicios de autenticación por el directorio X.500 a sus usuarios.
- El directorio puede servir como un repositorio de claves públicas mediante certificados del tipo descrito.
- **Cada certificado contiene la clave pública de un usuario y se firma con la clave privada de la autoridad de certificación de confianza.**
- Además, **X.509 define protocolos de autenticación alternativos basados en el uso de certificados de clave pública.**

Certificados de Clave Pública Certificados X.509

- X.509 es un estándar importante porque la estructura certificado y protocolos de autenticación definidos en mismo X.509 se utilizan en una variedad de contextos:
 - En S / MIME, Seguridad IP, SSL / TLS, Etc....
- X.509 se publicó inicialmente en 1988 . El estándar fue revisado posteriormente para abordar algunas de las preocupaciones de seguridad documentadas en
 - I'Anson,C.,and Mitchell,C.“Security Defects in CCITT Recommendation X.509—The Directory Authentication Framework” Computer Communications Review, April 1990.
 - Mitchell, c. , Walker, M., and Rush, D. "CCITT/ISO Standards for Secure Message Handling. " IEEE Journal on Selected Areas in Communications, 1989.
- Se publicó una recomendación revisada se publicó en 1993.
- Una tercera versión se publicó en 1995 y fue revisada en 2000.
- X.509 se basa en el uso de la criptografía de clave pública y firmas digitales.
- La norma no impone el uso de un algoritmo específico, pero recomienda RSA.

Certificados de Clave Pública Certificados X.509

Certificate:

Data:

Version: 1 (0x0)
Serial Number: 7829 (0xe1e95)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
OU=Certification Services Division,
CN=Thawte Server CA/Email=server-certs@thawte.com

Validity

Not Before: Jul 9 16:04:02 1998 GMT
Not After : Jul 9 16:04:02 1999 GMT

Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org

Subject Public Key Info:

Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
e8:35:1c:9e:27:52:7e:41:8f
Exponent: 65537 (0x10001)

Signature Algorithm: md5WithRSAEncryption
93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
ab:2f:4f:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
68:9f

Imagen extraída <http://es.wikipedia.org/wiki/X.509>

- Al final está la firma del certificado.
- Para poner la firma, la CA calcula un hash MD5 (en este caso) de la primera parte del certificado (la sección de Data: los datos del mismo más la clave pública).
- Se cifra ese hash con la clave privada, PR, de la CA.
- Si nos conectamos a www.freesoft.org y el sitio devuelve el certificado de la izq.
- Para validar este certificado, necesitamos el certificado de la CA (Thawte Server CA).
- Se toma la PU del certificado de la CA para decodificar la firma del primer certificado, obteniéndose un hash MD5.
- Este hash MD5 debe coincidir con el hash MD5 calculado sobre la primera parte del certificado.
- Si no se valida OK no se asegurara que el certificado de www.freesoft.org está vinculado con esa clave pública.

Certificados de Clave Pública Certificados X.509

(Autofirmados)

```
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 1 (0x1)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
OU=Certification Services Division,
CN=Thawte Server CA/Email=server-certs@thawte.com
Validity
Not Before: Aug 1 00:00:00 1996 GMT
Not After : Dec 31 23:59:59 2020 GMT
Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
OU=Certification Services Division,
CN=Thawte Server CA/Email=server-certs@thawte.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:d3:a4:50:6e:c8:ff:56:6b:e6:cf:5d:b6:ea:0c:
68:75:47:a2:aa:c2:da:84:25:fc:a8:f4:47:51:da:
85:b5:20:74:94:86:1e:0f:75:c9:e9:08:61:f5:06:
6d:30:6e:15:19:02:e9:52:c0:62:db:4d:99:9e:e2:
6a:0c:44:38:cd:fe:be:e3:64:09:70:c5:fe:b1:6b:
29:b6:2f:49:c8:3b:d4:27:04:25:10:97:2f:e7:90:
6d:c0:28:42:99:d7:4c:43:de:c3:f5:21:6d:54:9f:
5d:c3:58:e1:c0:e4:d9:5b:b0:b8:dc:b4:7b:df:36:
3a:c2:b5:66:22:12:d6:87:0d
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints: critical
    CA:TRUE
Signature Algorithm: md5WithRSAEncryption
07:fa:4c:69:5c:fb:95:cc:46:ee:85:83:4d:21:30:8e:ca:d9:
a8:6f:49:1a:e6:da:51:e3:60:70:6c:84:61:11:a1:1a:c8:48:
3e:59:43:7d:4f:95:3d:a1:8b:b7:0b:62:98:7a:75:8a:dd:88:
4e:4e:9e:40:db:a8:cc:32:74:b9:6f:0d:c6:e3:b3:44:0b:d9:
8a:6f:9a:29:9b:99:18:28:3b:d1:e3:40:28:9a:5a:3c:d5:b5:
e7:20:1b:8b:ca:a4:ab:8d:e9:51:d9:e2:4c:2c:59:a9:da:b9:
b2:75:1b:f6:42:f2:ef:c7:f2:18:f9:89:bc:a3:ff:8a:23:2e:
70:47
```

Anonimia y privacidad

- Como comentamos en las primeras transparencias, la privacidad que se puede conseguir a través de la anonimia es un servicio y característica de seguridad fundamental que cada vez se está prestando más atención.
- Algunos puntos motivantes para que la anonimia y privacidad sean tomadas en cuenta pueden ser:
 - Entrado en vigor en toda Europa la nueva ley de protección de datos: [GDPR/RGPD: General Data Protection Regulation/Regulación General de Protección de datos](#) (25 de mayo de 2018).
 - La alerta social generada por la pérdida total de privacidad en la cual nos encontramos hoy en día ([The end of privacy](#) (número especial de la revista Science de Enero 2015).
 - Pruebas de la importancia de la privacidad es la organización [Wikileaks](#): sin anonimia y privacidad no existiría.
 - También las últimas filtraciones por parte de Edward Snowden de como somos completamente investigados mediante herramientas muy potentes de vigilancia masiva como son: [PRISM](#) y [XKeyscore](#).
 - La información derivada de la asistencia médica, debe ser privada.
 - Fundamental en voto electrónico.
 - Y no olvidemos el comercio electrónico, debería mantener la privacidad.
 - O las redes sociales genéticas.
 - En el big data la anonimización es muy complicada (recordemos la reidentificación en transacciones de tarjetas de crédito, ([Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, Alex "Sandy" Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. Science 30 January 2015: Vol. 347 no. 6221 pp. 536-539 DOI: 10.1126/science.1256297](#)). El truco es la masiva cantidad de datos.
- Por todos estos motivos y más que no se presentan hay que tratar en serio la anonimia y privacidad.

Algunos conceptos sobre anonimia y privacidad

- Recordar algunos conceptos (generales):
 - Confidencialidad: la información se entiende sólo por aquellos que están autorizados para hacerlo.
 - Integridad: asegura que la información no sea alterada, ya sea consciente (y tal vez maliciosamente) o accidentalmente.
 - Autenticidad: garantiza la legitimidad de la información.
 - autenticación de entidad: que una entidad es quien dice que es,
 - y autenticación del origen de datos: que la información proviene de la fuente esperada.
 - Para los conceptos relacionados con anonimia y privacidad nos basaremos en trabajo de A. Pfitzmann and M. Hansen, Anonymity, unlinkability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology, 2005. Accesible desde:
 - https://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.28.pdf.

Algunos conceptos sobre anonimia y privacidad

- Recordar algunos conceptos (anonimia y privacidad):
 - **Privacidad:** es la calidad de un sistema o protocolo de asegurar que ninguna información privada, ni información que pueda ser procesada para inferir datos de otro modo privado, para no filtrar a otras partes que no sean los destinatarios deseados. Depende mucho del contexto (en algunos sistemas puede que informaciones privadas no lo sean en otros).
 - **Anonimato:** es el estado de ser no identifiable dentro de un conjunto de sujetos (los del conjunto de anonimato).
 - **Seudonimia:** es el uso de seudónimos como identificadores, principalmente para la autenticación.
 - **Unlinkability (Imposibilidad de vinculación):** es la propiedad garantizar que un adversario no puede hacerlo mejor que al azar para determinar si o no dos o más elementos están vinculados de alguna manera predefinida. Es decir, en el contexto de firmas grupales, dados dos mensajes y sus firmas, no podemos decir si las firmas eran del mismo firmante o no.

Algunos conceptos sobre anonimia y privacidad

- Realmente la anonimia y privacidad son conceptos muy amplios y existen en todas las partes de las tecnologías de la información.
- Así la anonimia y privacidad puede hacer referencia:
 - A la información que se envía:
 - Una posible metodología criptográfica para esto son las Firmas Grupales.
 - O puede hacer referencia a las comunicaciones:
 - Redes de comunicaciones anónimas: redes de mezcla de tráfico, onion routing y ofuscación de tráfico.
 - Siendo Tor la más famosa.

Firmas Grupales: Autenticación con Privacidad

- Para que la información sea anónima y privada se puede quitar todo lo que identifique a la persona de esa información.
- Pero hay un problema:
 - Si se quiere acceder a cualquier servicio en las tecnologías de la información, siempre se requiere algo de información personal.
 - Si no das esa información personal no puedes usar los servicios que necesitas.
 - Por lo tanto esto sería incompatible con la privacidad y anonimía.
 - Por lo tanto necesitamos autentificarnos en ese servicio con privacidad:
 - Firmas grupales: una posible solución criptográfica.

Firmas Grupales: Autenticación con Privacidad

- En la autenticación basada en grupo, los usuarios pueden **autenticarse en nombre de algún grupo**, en lugar de sobre la base individual.
- Es decir, el proceso de autenticación no revela ninguna información que pueda utilizarse para identificar a algún usuario en particular.
- Desde toda la información revelada sólo puede estar vinculado a algún grupo de usuarios, la autenticación basada en grupo es un enfoque adecuado para la consecución de la privacidad del usuario.
- Con este enfoque los usuarios se **consideran autenticados** si pueden proporcionar una **prueba de la pertenencia al grupo**.
- Hay que tener en cuenta que la autenticación basada en grupo se utiliza a menudo con el propósito de control de acceso, donde los individuos a menudo se asignan a los grupos y permisos para acceder y operar en ciertos recursos se otorga en base a estas tareas.
- En nuestro contexto estamos interesados en técnicas de autenticación basadas en grupos aplicando las firmas digitales.
- El concepto de las firmas del grupo, adopta la autenticación basada en grupo para lograr privacidad de firmantes contra verificadores potenciales.

Firmas Grupales: Autenticación con Privacidad

- La firmas grupales son las primitivas criptográficas que nos permiten construir protocolos de seguridad con privacidad respetuosa, y permitir la gestión justa del anonimato.
- Las firmas grupales están concebidas para proporcionar privacidad a través de anonimato.
- Las propiedades más generales de este nuevo sistema de firmas grupales son:
 - Sólo los miembros de un grupo pueden firmar los mensajes.
 - El receptor puede verificar que se trata de una firma válida dentro del grupo, pero no puede descubrir qué miembro de grupo fue quien firmó. Es decir dados dos mensajes y dos firmas no se puede decir si corresponde al mismo firmante o no.
 - Si fuese necesario, la firma podría ser "abierta", de manera que la persona que firmó el mensaje se revela (revelación de identidad). Solo lo puede hacer el gestor de grupo. Esta propiedad no la tienen todos los tipos de firmas grupales que existen.

Firmas Grupales: Autenticación con Privacidad

- Como ilustración del esquema de privacidad con apertura anterior supongamos el siguiente ejemplo para clarificar el problema:
 - Una empresa tiene varios ordenadores, cada uno de ellos conectado a la red local.
 - Cada departamento de ese empresa tiene su propia impresora (también conectado a la red) y sólo las personas de ese departamento se les permite utilizar la impresora de su departamento.
 - Por lo tanto, antes de imprimir cualquier hay que estar convencido de que el usuario está trabajando en ese departamento de la impresora (solo se imprimen trabajos relacionados con ese departamento).
 - Al mismo tiempo, la empresa quiere privacidad para impresión de documentos: el nombre del usuario no puede ser revelada.
 - Sin embargo, si alguien descubre al final del día que una impresora ha sido utilizada muy a menudo y de manera abusiva, el director debe ser capaz de descubrir quién abusa esa impresora, al que le enviara una factura.

Firmas Grupales: Autenticación con Privacidad

- Firmas grupales, primeramente propuestas Chaum, son como las firmas digitales convencionales, que hemos visto, en que se utilizan para **demostrar que el propietario de un secreto específico ha sido la fuente de información**.
- La diferencia fundamental con su homólogo convencional, es que las firmas grupales (FGs) ocultan a este propietario entre un conjunto (grupo) de posibles propietarios y por tanto la firma es anónima.
- Se han propuesto diversas variaciones de las FGs, que permiten añadir funciones adicionales:
 - Por ejemplo y por lo general, hay un administrador de grupo (AG), quien controla alguna información secreta que le permite revocar este anonimato, y buscar la identidad del emisor de la firma en el grupo (esto se llama la apertura de una firma de grupo).
 - Sin embargo las firmas de anillo proporcionan anonimato incondicional, lo que significa que la funcionalidad de apertura de firma no se puede realizar.
 - Otros esquemas añaden una “trampilla adicional” además de la que utilizado para la apertura de las firmas del grupo, para que una autoridad (el AG o alguna autoridad subsidiaria) sea capaz de vincular las firmas hechas por el mismo miembro del grupo, pero en lugar de usar su identidad, se utiliza una “trampilla de trazado” exclusivamente para esta tarea. Este procedimiento se llama trazado, y por esto este tipo de firmas se denominan firmas trazables.
 - Etc.

Firmas Grupales: Autenticación con Privacidad

- Vamos a ver un ejemplo muy sencillo de esquema de firma grupal:
 - El AG elige un determinado sistema de clave pública y da a cada persona una lista de claves secretas (estas listas son todos disjuntas) y publica la lista completa de las correspondientes claves públicas (en orden aleatorio) en un directorio público de confianza. Esto es lo que se llama la clave pública del grupo.
 - Cada persona puede firmar un mensaje con una clave secreta de su lista, y el destinatario puede verificar la firma con su correspondiente clave pública de la lista pública del directorio de confianza. Cada clave se usará sólo una vez, de lo contrario las firmas creadas con esa clave estarían vinculadas.
 - AG conoce todas las listas de claves secretas, para que en caso de la disputa, él sabe la identidad del que hizo la firma de disputa. Por lo tanto AG es necesario para la configuración y para la “apertura” una firma. Un problema de este esquema es que AG sabe todas las claves secretas de los miembros del grupo y por tanto, también puede crear firmas. Esto se puede prevenir mediante el uso de claves públicas ciegas. Así en general las claves secretas no las conoce AG.
 - Si cada miembro del grupo recibe el mismo número de claves secretas, entonces la longitud de la clave pública de este grupo (es decir, la longitud del directorio público de confianza) es lineal en el número de personas; pero el número de mensajes que una persona puede firmar es fijo (el número de claves secretas asignadas).
 - [Chaum, David; van Heyst, Eugene \(1991\). Group signatures. Advances in Cryptology - EUROCRYPT '91. Lecture Notes in Computer Science. Vol. 547. pp. 257–265. doi:10.1007/3-540-46416-6_22.](#)
- Este es un esquema muy sencillo pero ilustrativo para entender el problema.

Navegación anónima: redes de mezcla de tráfico

- Pasamos ahora a anonimia en las comunicaciones.
- En la aplicación de estos diferentes tipos de anonimato, es frecuente recurrir a redes de comunicaciones anónimas (redes con una configuración especial que proporcionan algún tipo de anonimato).
- Las redes más conocidas de este tipo son las **redes de mezcla** y aquellas basadas en el **enrutamiento cebolla**.
 - Mix network (red de mezcla): proporciona el anonimato de las comunicaciones enrutada por medio de que cada proxy intermedio (conocido como mezcla) baraja al azar todos los paquetes cifrados que recibe de varios destinatarios. Al encaminar los paquetes a través de varias mezclas, el rastreo de ellos se hace más difícil.
 - Onion routing (enrutamiento cebolla): es una comunicación que mejora el anonimato debido a que un paquete se encamina a través de varios “proxies” intermedios conocidos como “routers” de cebolla. El origen encripta cada paquete con la clave pública de cada uno de estos “routers” cebolla. Primero, el ordenador A, que quiere enviar el mensaje a B, calcula una ruta aleatoria al destino pasando por varios nodos intermedios (“routers” cebolla). Consigue las claves públicas de todos ellos usando un directorio de nodos. Primero cifrará el mensaje con la clave pública del último nodo de la ruta, para que sólo él lo pueda descifrar. Además del mensaje, incluye (también cifradas) instrucciones para llegar al destino, B. Así se hará lo mismo sucesivamente con todos los nodos de la ruta. Este mensaje cifrado varias veces, se envía al primer nodo “cebolla” que lo descifra con su clave privada, y así sucesivamente hasta el final hasta el último nodos que saca el mensaje. Por ejemplo [TOR](#).

Navegación anónima: Mix Network

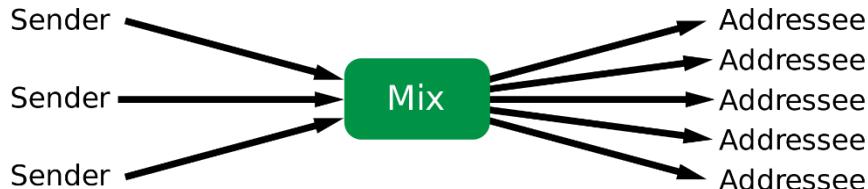
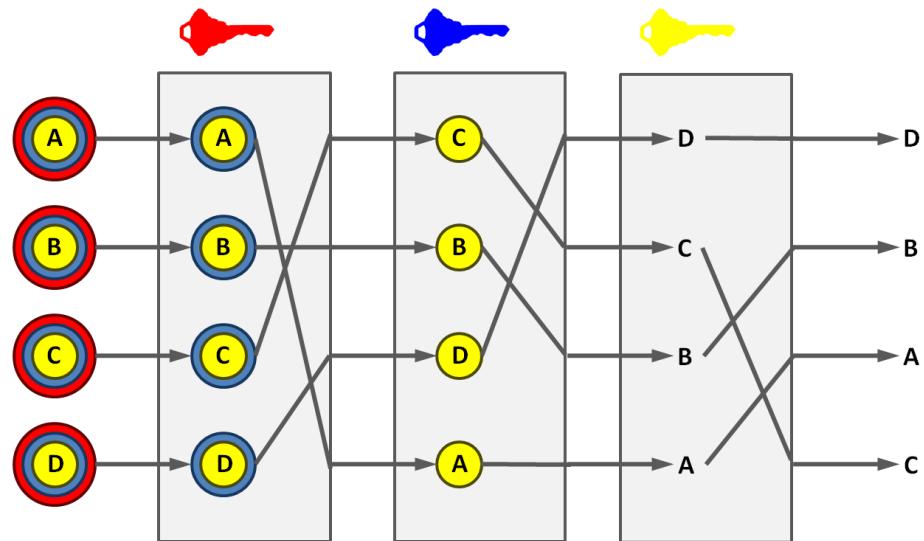


Imagen extraída de
https://en.wikipedia.org/wiki/Mix_network#/media/File:Chaum_Mix.svg

Los mensajes se cifran bajo una secuencia de claves públicas. Cada nodo de mezcla elimina una capa de cifrado utilizando su propia clave privada. El nodo baraja el orden de los mensajes y transmite el resultado al siguiente nodo.

Imagen extraída de
https://en.wikipedia.org/wiki/Mix_network#/media/File:Red_de_mezcla.png



Navegación anónima: TOR

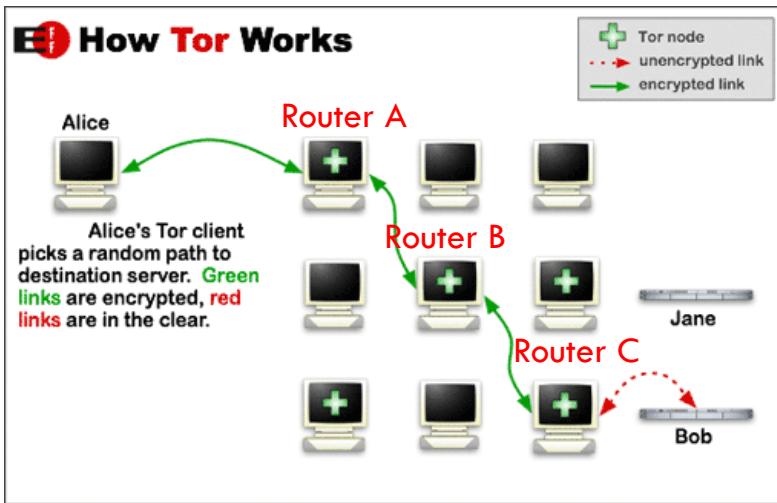
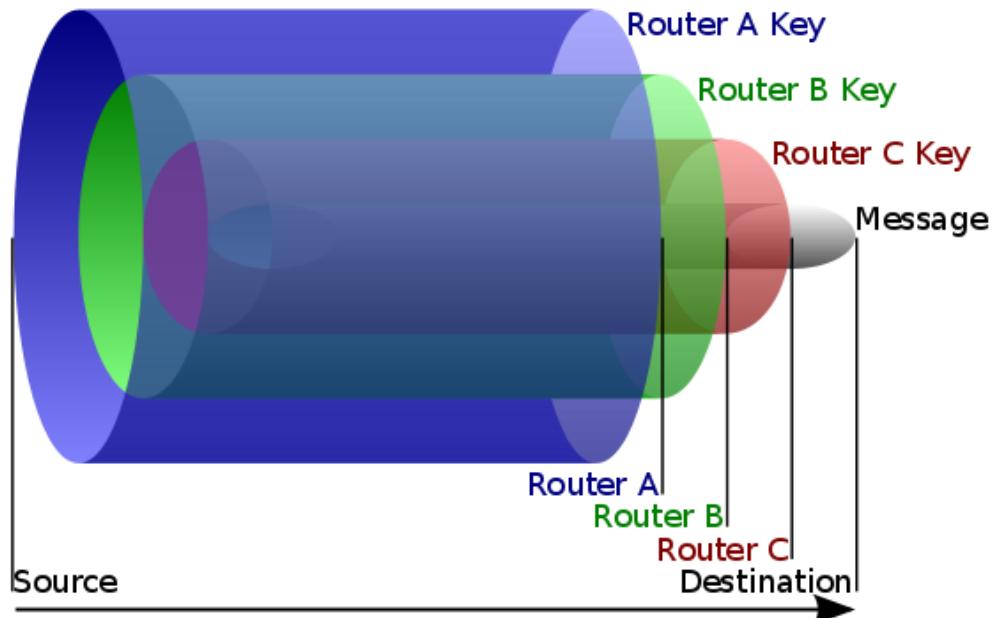


Imagen extraída de
[https://en.wikipedia.org/wiki/Tor_\(anonymity_network\)#/media/File:Tor-onion-network.png](https://en.wikipedia.org/wiki/Tor_(anonymity_network)#/media/File:Tor-onion-network.png)

Imagen extraída de
https://upload.wikimedia.org/wikipedia/commons/e/e1/Onion_diagram.svg



Navegación anónima: TOR

- La tecnología Tor, es una red de baja latencia superpuesta sobre internet.
- En esta red en el enrutamiento de los mensajes no revela la identidad de los usuarios en la red e comunicaciones, es decir su dirección IP.
- Por lo tanto como hemos dicho anteriormente ofrece anonimato en el nivel de las comunicaciones.
- Por esta razón se dice que Tor pertenece a la web profunda:
 - Existe la Internet superficial: cuyas páginas se indexan por los típicos motores de búsqueda de Google, Yahoo, Bing, etc.
 - Y existe la Internet profunda o web profunda: cuyas páginas son difíciles de indexar o incluso privadas. Es difícil estimar su cantidad pero se estima que es más de un 90% de la información residente en Internet.
- Así los servidores de Tor están alojados en la red, pero se mantienen fuera del alcance de las redes convencionales.
- Así estos sitios, con dominio .onion, forman parte de la denominada web profunda.
- Ejemplo en <https://www.genbeta.com/web-20/47-paginas-onion-para-visitar-el-lado-amable-de-la-deep-web>

Navegación anónima: TOR

- Tor surgió en los laboratorios de investigación naval de los Estados Unidos ([Dingledine et al. in 2004](#)).
- Pero hoy en día es una organización sin ánimo de lucro: [Tor Project](#).
- Es utilizada por miles de usuarios distintos en todo el mundo y ofrece software libre y gratuito.
- TOR Project provee un navegador de manera totalmente gratuita para poder navegar de forma anónima.
- Básicamente lo que hace el software es convertir nuestro ordenador en un nodo OP (Onion Proxy) y redirige nuestro tráfico a través de varios nodos OR (Onion Router).
- El navegador viene con varias herramientas para proteger la información de la navegación: posibilidad de desactivar scripts, cookies; navegar por https por defecto, etc.
- En definitiva el navegador Tor genera un Proxy local en tu computador como entrada a la red anónima Tor.
- Una vez que estas con el navegador Tor puedes navegar de manera anónima por la red normal de las páginas web habituales, o por la red profunda o servicios escondidos como los dominios .onion.

Navegación anónima: TOR

- Elementos de la red Tor:
 - Nodos OR (Onion Router): son enrutadores de los paquetes de datos, y determinan la ruta de los paquetes de datos (servidores DNS). Mantienen un servicio de directorio replicado lo la red TOR.
 - Nodos OP (Onion Proxy): ejecutan el software final y obtienen información del servicio de directorio con el fin de determinar la ruta a seguir. Los servicios de directorio también contienen las claves públicas de los nodos de Tor.
 - Células: Cuando se abre la conexión TLS, los nodos mandan paquetes de información prototipados (células) entre ellos:
 - Células de control: interpretadas por el nodo que las recibe y controlan la comunicación.
 - Células de transmisión (relay cell): son usadas para la comunicación entre el OP y cualquiera de los OR del circuito.
 - Para ver la información de la red y estatus de TOR: <https://torstatus.rueckgr.at/>

Navegación anónima: TOR

➤ Enrutamiento de paquetes:

- En Internet cuando navegamos nos conectamos a un punto de información que nos dirige hacia el servidor que deseamos para consultar.
- En Tor la petición se dirige por una ruta de nodos aleatoria que llega hasta el servidor que deseamos para consultar. El servidor no sabe la ip del origen de la información.
- El usuario de Tor, por medio del software adecuado (contenido en un nodo OP), pregunta al servidor de directorio la información necesaria para poder trazar una ruta hasta el destino. Generalmente la ruta se selecciona de manera aleatoria.
- Una vez determinado el camino a seguir, el OP negocia con cada OR por el que se va a pasar las claves para cifrar el mensaje. La negociación implica el intercambio de las claves de sesión mediante Diffie-Hellman. De esa manera se obtienen las claves de intercambio simétricas para el AES128 entre el OP y los tres ORs. En cada paso de los ORs los mensajes encebolados son cifrados (AES128) con las claves de sesión negociadas previamente con DH.
- El mensaje se va cifrando con las claves públicas siguiendo el orden inverso a los nodos por los que va pasando (comenzando por la clave del nodo de salida y terminando por la clave del nodo de entrada) y se envía al primer nodo.
- Así cada nodo recibe el paquete y descifra con su clave privada la “capa” más externa del paquete y lo reenvía al siguiente nodo, hasta que el mensaje llega al último nodo (nodo de salida).
- El servidor destino de la petición responderá al nodo de salida de TOR, así todas las respuestas del destino serán enviadas a ese nodo salida. Este es el encargado de mandarlo hacia atrás hasta llegar al Origen.

Navegación anónima: TOR

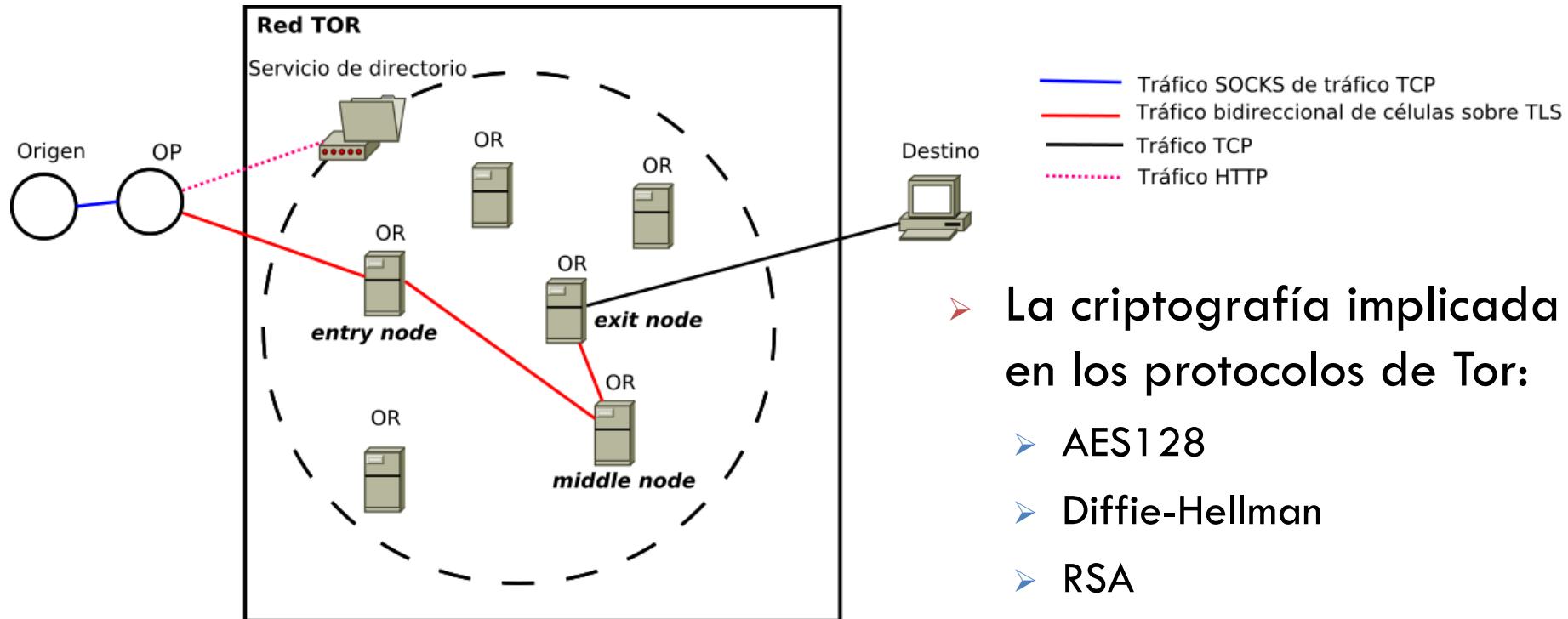


Imagen extraída de [https://es.wikipedia.org/wiki/Tor_\(red_de_anonimato\)](https://es.wikipedia.org/wiki/Tor_(red_de_anonimato))

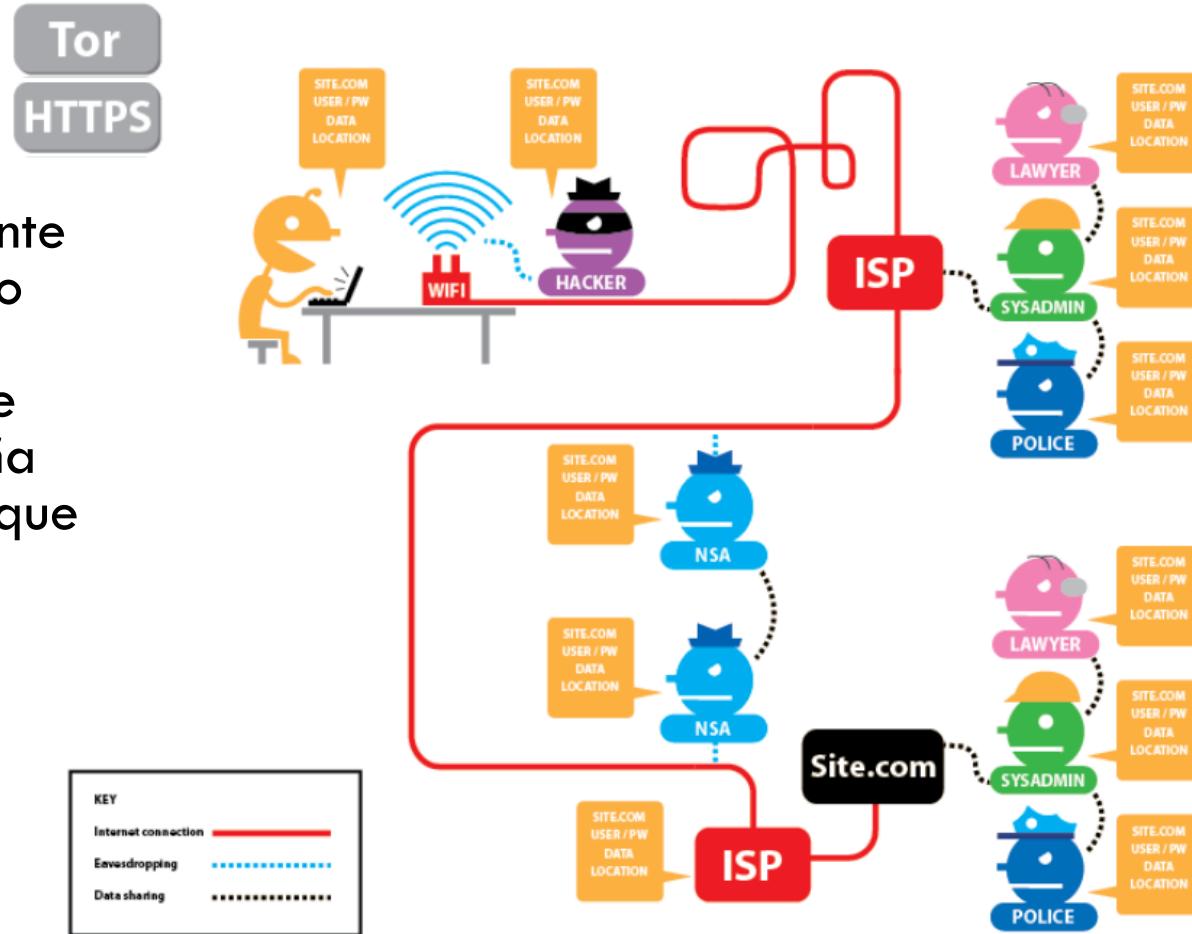
Navegación anónima: TOR

- Primero cifrará el mensaje con la clave pública del último nodo de la ruta, para que sólo él lo pueda descifrar. Además del mensaje, se cifran las instrucciones para llegar al destino final a través de una ruta aleatoria predefinida.
- Segundo, todo este paquete cifrado (mensaje + instrucciones para llegar al último nodo de la lista), se cifra de nuevo con la clave pública del penúltimo nodo para que sólo lo pueda descifrar este nodo de la ruta. Y así sucesivamente.
- Cuando el mensaje parte hacia el destino con todas las capas de cifrado, el primer nodo descifra el paquete con su clave privada, y encuentra las instrucciones para enviar hasta el siguiente nodo. Así sucesivamente, hasta que el último nodo descifra la última capa y la envía al destino.
- El destino solo sabe que la información viene del último nodo y no de un usuario que ha generado una ruta por la red TOR.
- **OJO, la red TOR no te anonimiza la información que se envía, solo las comunicaciones.**

Navegación anónima: Sin TOR ni HTTPS

Imagen extraída de
<https://www.eff.org/es/pages/tor-and-https>

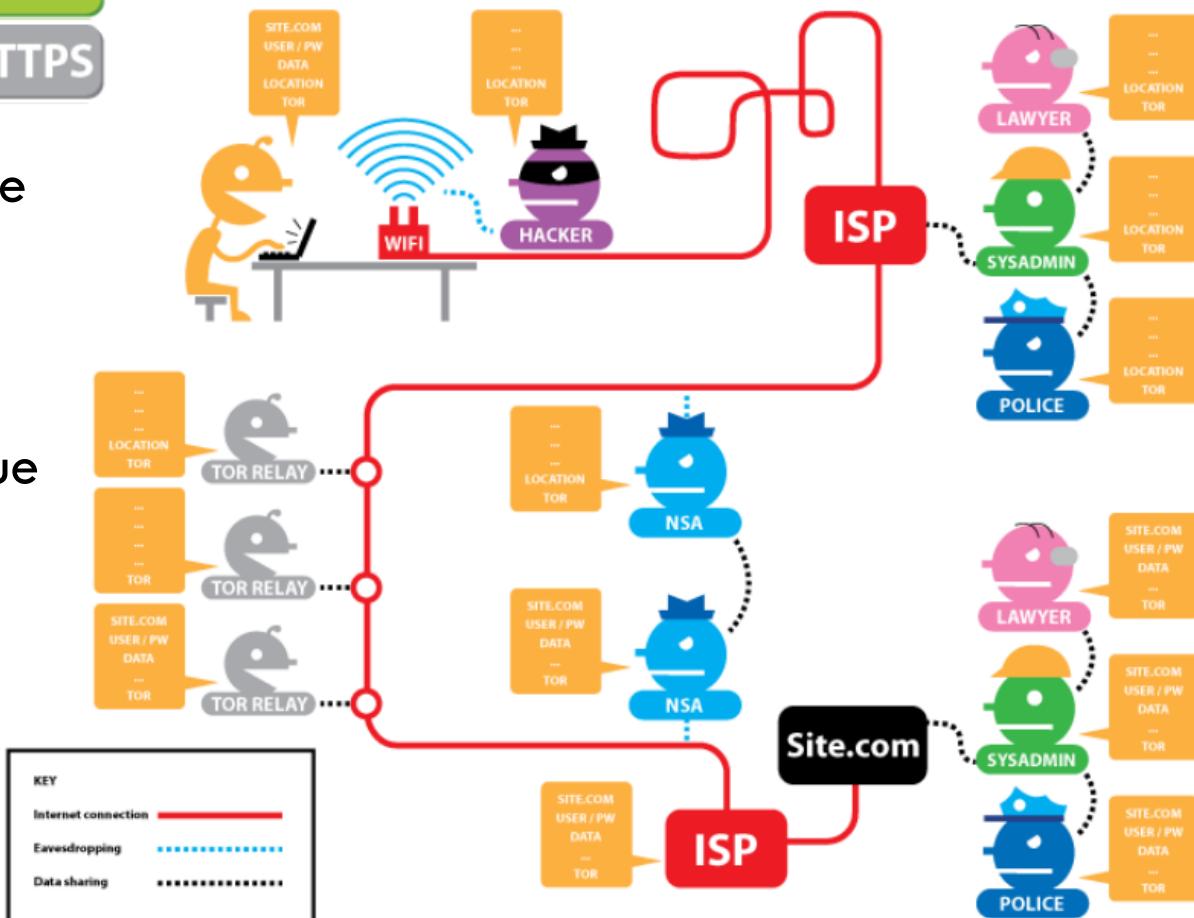
Los datos potencialmente visibles incluyen: el sitio que está visitando (**SITE.COM**), su nombre de usuario y contraseña (**USER/PW**), los datos que está transmitiendo (**DATA**), su dirección IP (**LOCATION**) y si está utilizando Tor (**TOR**)



Navegación anónima con TOR, sin HTTPS

Tor
HTTPS

Los datos potencialmente visibles incluyen: el sitio que está visitando (**SITE.COM**), su nombre de usuario y contraseña (**USER/PW**), los datos que está transmitiendo (**DATA**), su dirección IP (**LOCATION**) y si está utilizando Tor (**TOR**)



Navegación anónima: sin TOR, con HTTPS

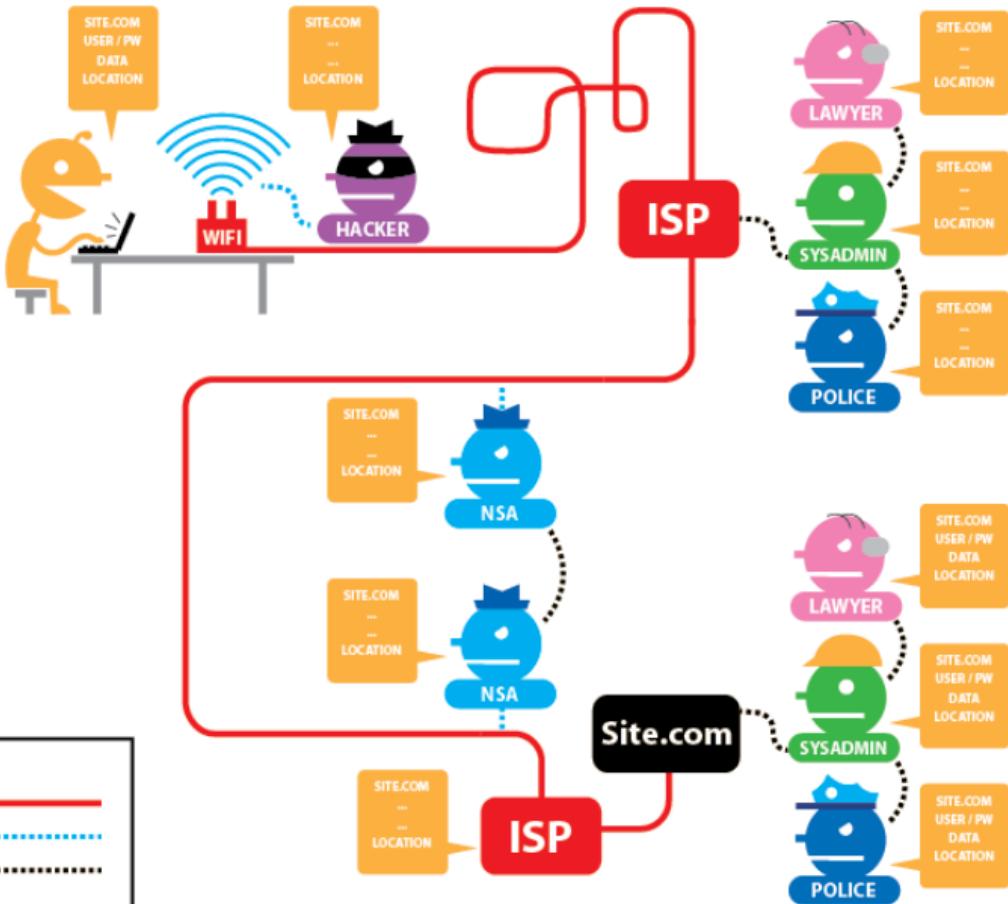


Imagen extraída de
<https://www.eff.org/es/pages/tor-and-https>

Los datos potencialmente visibles incluyen: el sitio que está visitando (**SITE.COM**), su nombre de usuario y contraseña (**USER/PW**), los datos que está transmitiendo (**DATA**), su dirección IP (**LOCATION**) y si está utilizando Tor (**TOR**)

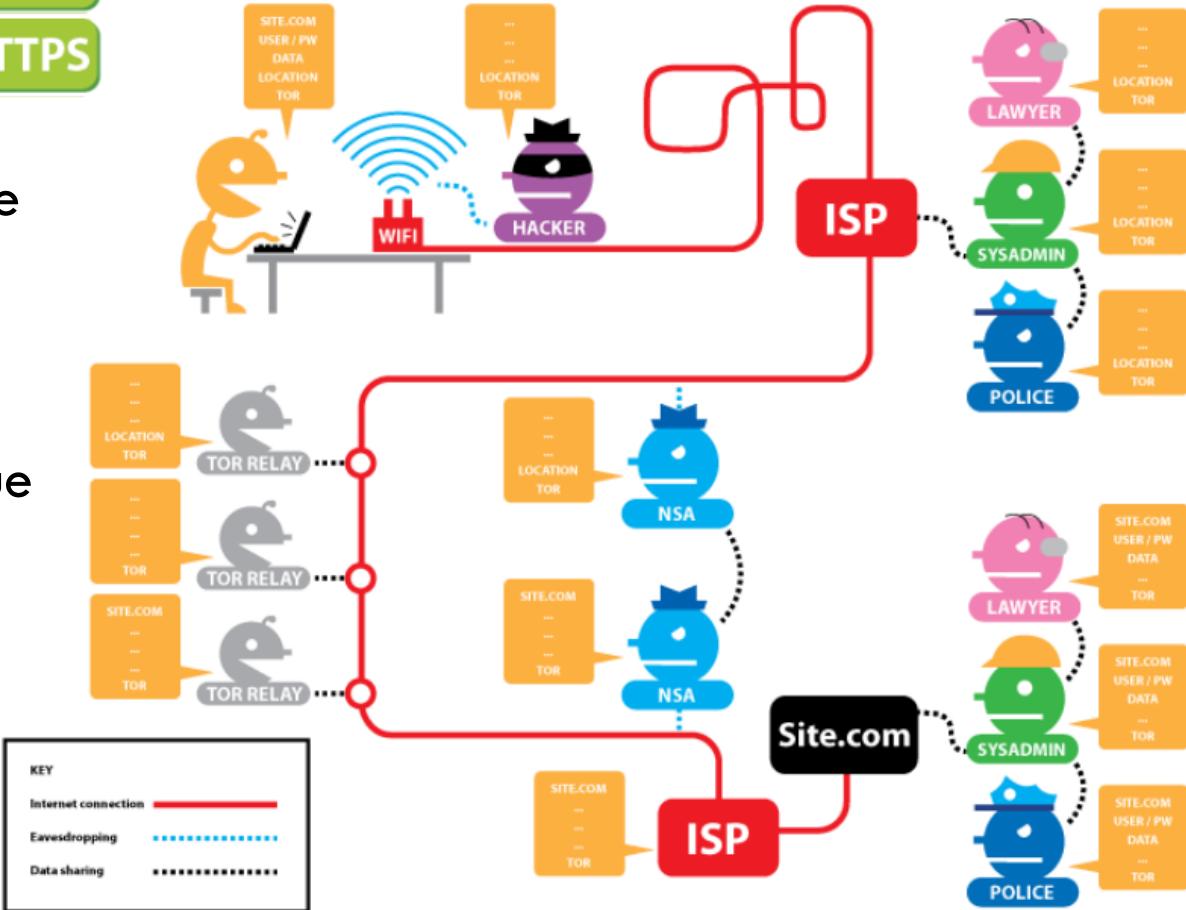
Navegación anónima: con TOR y con HTTPS

Tor

HTTPS

Imagen extraída de
<https://www.eff.org/es/pages/tor-and-https>

Los datos potencialmente visibles incluyen: el sitio que está visitando (**SITE.COM**), su nombre de usuario y contraseña (**USER/PW**), los datos que está transmitiendo (**DATA**), su dirección IP (**LOCATION**) y si está utilizando Tor (**TOR**)



Bibliografía y lecturas relacionadas:

- W. Stallings, "Cryptography and Network Security: Principles and Practice" (Básica).
- W. Stallings, L. Brown "Computer Security: Principles and Practice" (Básica).
- Ross Anderson, "Security Engineering", (Básica).
- A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, "Handbook of Applied Cryptography" (Básica).
- A. Pfitzmann and M. Hansen, Anonymity, unlinkability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology, 2005. Accesible desde https://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.28.pdf (Básica).
- Al Sweigart. Hacking Secret Ciphers with Python: A beginner's guide to cryptography and computer programming with Python (Complementaria).
- TJ O'Connor. Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers (Complementaria).
- K. Rannenberg, J. Camenisch, A. Sabouri (eds.). Attribute-based Credentials for Trust Identity in the Information Society, Springer, 2015 (Avanzado).
- Daniel Echeverri, Deep Web TOR, FreeNET & I2P Privacidad y Anonimato, ZeroXword Computing, 2016 (Avanzado).

Bibliografía y lecturas relacionadas:

- [Navigating Big Data's Privacy and Security Challenges, kpmg.com \(KPMG 2014\).](#)
- [K. Crawford & J. Schultz. Big Data and Due Process: Toward a Framework to Redress Predictive Privacy Harms. Boston College Law Review, Volume 55, Issue 1, Article 4. 2014.](#)
- [The end of privacy \(número especial de la prestigiosa revista Science de Enero 2015\).](#)
- [The ethics of big data: Focus Feature \(Focus Feature en la prestigiosa revista PLOS Computational Biology\): es un foro para la discusión de temas de vanguardia en el campo.](#)
- [Ethics of Big Data, Balancing Risk and Innovation. By Kord Davis. O'Reilly Media. 2012.](#)
- [Data and Ethics: Etiquette and Law for an Always-On World – Recorded Online Conference](#)
- Richards, Neil M. and King, Jonathan H., Big Data Ethics (May 19, 2014). Wake Forest Law Review, 2014. Available at SSRN: <http://ssrn.com/abstract=2384174>
- W. Stallings, “Cryptography and Network Security: Principles and Practice”. 6º edición.
- Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In ASIACRYPT, pages 552–565, 2001.
- David Chaum and Eugène van Heyst. Group signatures. In EUROCRYPT, pages 257–265, 1991.

Bibliografía y lecturas relacionadas:

- [M. Manulis, N. Fleischhacker, F. Günther, F. Kiefer, B. Poettering. Group Signatures: Authentication with Privacy. Bundesamt für Sicherheit in der Informationstechnik 2012.](#)
- Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity - A proposal for terminology. In *Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, USA, July 25-26, 2000, Proceedings, pages 1–9, 2000.
- Seung Geol Choi, Kunsoo Park, and Moti Yung. Short traceable signatures based on bilinear pairings. In *IWSEC*, pages 88–103, 2006.
- Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In *EUROCRYPT*, pages 571–589, 2004.
- Jesus Diaz, David Arroyo, and Francisco B. Rodriguez. Anonymity revocation through standard infrastructures. In *EuroPKI*, pages 112–127, 2012.
- Jesus Diaz, David Arroyo, and Francisco B. Rodriguez. Fair anonymity for the Tor network. *CoRR*, abs/1412.4707, 2014.
- Jesus Diaz, David Arroyo, and Francisco B. Rodriguez. libgroupsig: an extensible c library for group signatures. *submitted*, 2015.
- Jesus Diaz, David Arroyo, and Francisco B. Rodriguez. New X.509-based mechanisms for fair anonymity management. *Computers & Security*, 46:111–125, 2014.
- Tesis: "Design and implementation of secure protocols for practical authentication and fair anonymity systems". Autor: Diaz Vico, Jesus. Universidad: AUTÓNOMA DE MADRID, Departamento: ESCUELA SUPERIOR DE INFORMÁTICA.