Seguridad

Fernando Díaz Gómez [1]

[1] University of Valladolid – C/Plaza de Santa Cruz, 8, 47002 Valladolid, Spain
fdiaz@infor.uva.e

Resumen: En este capítulo el lector encontrará una introducción a la protección de los sistemas informáticos, característica que en los últimos tiempos se ha convertido en una cues-tión muy relevante para todos aquellos sistemas abiertos, es decir, aquellos sistemas informáticos que proporcionan sus servicios a través de una red de comunicaciones como es, por ejemplo, Internet. En cuanto a la organización del capítulo, la primera sección de este capítulo introduce el panorama general de la seguridad, introduciendo las propiedades de la información que deben salvaguardarse para garantizar la seguri-dad, las amenazas a las que está expuesta la información, los servicios de seguridad requeridos para contrarrestar estas amenazas, los mecanismos que implementan los servicios requeridos. Por este motivo, se recomienda su lectura con el fin de obtener una visión global de este ámbito, que puede complementarse con el primer apartado de la sección dedicada a la criptografía, técnica que constituye la base fundamental sobre la que se basan la mayoría de los servicios de seguridad requeridos. El resto de apar-tados de este capítulo, es decir, los centrales de la sección dedicada a la criptografía son los más técnicos, por lo que se recomienda su lectura sólo a aquellos lectores que estén interesados en profundizar en los fundamentos de la criptografía y en la descrip-ción de los algoritmos criptográficos más comunes.

Palabras clave: Marketing

Abstract. In this chapter the reader will find an introduction to the protection of computer systems, a feature that has recently become a very relevant issue for all those open systems, that is, those computer systems that provide their services through a communications network such as, for example, the Internet. As for the organization of the chapter, the first section of this chapter introduces the general panorama of security, introducing the properties of the information that must be safeguarded to guarantee security, the threats to which the information is exposed, the security services required to counteract these threats, the mechanisms implemented by the services required. For this reason, it is recommended to read it in order to obtain a global vision of this area, which can be complemented with the first section of the section dedicated to cryptography, a technique that constitutes the fundamental basis on which most of the required security services are based. The rest of the sections of this chapter, i.e. the central sections dedicated to cryptography are the most technical, so it is recommended to read them only to those readers who are interested in delving deeper into the fundamentals of cryptography and the description of the most common cryptographic algorithms.

Keywords: Marketing

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

1 Información y Seguridad

1.1 Introducción

Hoy en día, el éxito de una empresa depende en gran medida de la calidad de la información que genera y gestiona. La información comprende los datos propios que gestiona, los mensajes que se intercambian entre las personas y/o las máquinas de la organización, el historial de clientes y proveedores, de productos, etc. En definitiva, la información representa el *know-how* de la organización y si ésta se pierde o deteriora, le será muy difícil recuperarse y seguir siendo competitiva. Se dice que una empresa posee información de calidad si, además de ser útil en su operativa, presenta otras características adicionales como son la confidencialidad, la integridad y la disponibilidad. Es en este momento, cuando se trata de dotar de seguridad a los sistemas de información que la empresa utiliza. Así pues, puede decirse que la seguridad es el conjunto de medidas que las organizaciones adoptan con el fin de garantizar la confidencialidad, integridad y disponibilidad de la información. Cuando el acceso a la información de la organización se realiza a través de servicios telemáticos (como ocurren en el caso de Internet) la necesidad de adoptar estas medidas es insoslayable. En concreto, son cuatro los elementos fundamentales de la seguridad informática cuyas definiciones se han extraído de CSI (1996):

- *Confidencialidad*, que puede definirse como la característica que previene contra la divulgación no autorizada de la información.

- *Autenticación*, definida como la propiedad de dar y reconocer la autenticidad de la información y/o la identidad de los actores (típicamente una fuente y un destinatario de los datos) y/o la autorización por parte de los autorizadores, así como, la verificación de dichas tres cuestiones.

- *Integridad*, definida como la característica que previene contra la modificación o destrucción no autorizadas de la información de una organización.

- *Disponibilidad*, o característica que previene contra la denegación no autorizada de acceso a la información.

La necesidad de adoptar medidas de seguridad que protejan la información de una empresa se debe a que en el entorno en que ésta se desenvuelve (personas, máquinas, sucesos o ideas) existen amenazas, es decir, condiciones de dicho entorno que, dada una oportunidad, podrían dar lugar a que se produjese una violación de la seguridad. Ejemplos de amenazas de seguridad son: la divulgación no autorizada de la información, la modificación no autorizada de la información, el acceso no autorizado a recursos, la denegación de un servicio. Para hacer frente a estas amenazas y, por lo tanto, para protegerse o paliar los efectos de la materialización de una de estas amenazas, se dispone de servicios de seguridad, los cuales establecen qué hacer frente a estas amenazas con el fin de satisfacer los requisitos de seguridad de la organización. Ejemplos de servicios de seguridad son: la confidencialidad de los datos, la integridad de los datos y el control de acceso. Una vez identificadas las medidas a adoptar ante las posibles amenazas, es necesario implementar dichas medidas, para lo cual se acudirá a las técnicas y mecanismos de seguridad concretos que soportan la lógica y los algoritmos que implementan los servicios considerados. Ejemplos de técnicas y mecanismos de seguridad son: la criptografía, los cortafuegos (*firewalls*), las firmas digitales, etc. Por último, es importante resaltar que la identificación de las amenazas y la definición

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

e implementación de las oportunas medidas, debe hacerse en el marco de un plan de seguridad de la empresa que defina una política de seguridad, es decir, un documento que defina los objetivos de seguridad en función de las necesidades propias de la organización y asigne responsabilidades en la gestión de la seguridad [1-10].

1.2 Amenazas de seguridad

El estudio de la seguridad informática puede plantearse desde dos enfoques diferentes:

- *Seguridad física*, o protección del sistema ante amenazas físicas (desastres naturales, incendios, agua, robo y pirateo informático, etc.) y requiere la definición de planes de contingencia, así como, el establecimiento de una política de control de acceso físico al sistema, de una política de copias de seguridad, etc.

- *Seguridad lógica*, o protección de la información en su propio medio, es decir, en las aplicaciones informáticas que soportan tal información. Por lo general, se emplearán técnicas criptográficas para lograr este fin.

En cualquier caso deben asumirse los siguientes principios de cara a garantizar la seguridad.

- En primer lugar hay que ser conscientes de que el atacante utilizará cualquier artilugio que haga más fácil su acceso y su posterior ataque. Esto conduce a identificar los puntos débiles de cualquier sistema informático, y evaluar mediante un análisis de riesgos, no sólo la posibilidad de la amenaza, sino el perjuicio ocasionado en caso de materializarse.

- También hay que tener presente que sólo debe protegerse la información mientras ésta tenga valor, por lo que tiene sentido hablar de la caducidad de los sistemas de protección.

- Por último, las medidas de seguridad deberán estar implementadas de forma que su funcionamiento sea eficaz (que proporcionen la respuesta apropiada en el momento oportuno), eficiente (que optimicen los recursos del sistema) y fáciles de usar (que pasen desapercibidas para el usuario).

Por lo tanto, una de las primeras cuestiones a la hora de abordar el problema de la seguridad es identificar las debilidades del sistema. Estas pueden referirse básicamente a cinco entidades distintas: el hardware (conexiones sueltas, desconexión de tarjetas, etc.), el software (robo de programas, modificación, ejecuciones incorrectas, etc.), los datos (alteración de contenidos, introducción de datos falsos, manipulación fraudulenta, etc.), la memoria (virus, mala gestión de memoria, bloqueo del sistema, etc.) y los usuarios (suplantación de la identidad, accesos no autorizados, etc.). De estos cinco elementos, los tres primeros son los que sufren un mayor número de ataques y constituyen lo que se denomina el *triángulo de las debilidades del sistema* (véase la Figura 1).

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

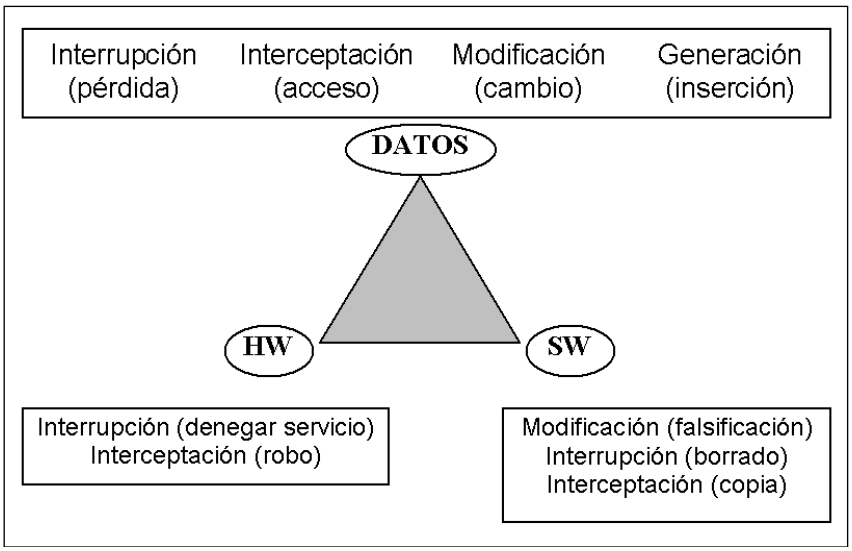
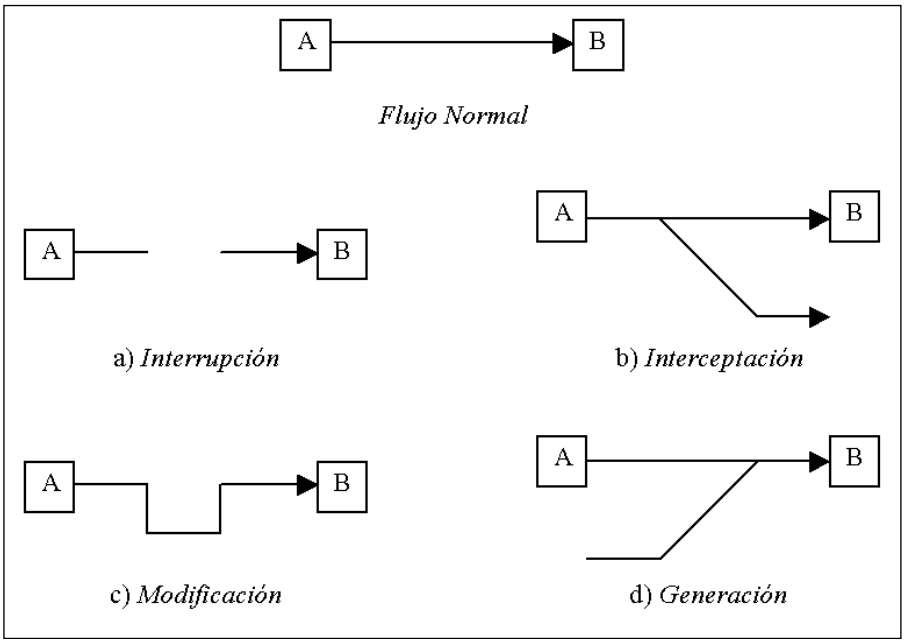


Figura 1: Triángulo de las debilidades de los sistemas informáticos.

Así pues, interesa conocer cuales son las amenazas que afectan a los tres elementos claves (hardware, software y datos), entendiendo genéricamente por amenaza cualquier condición del entorno del sistema de información (persona, máquina, suceso o idea) que, dada una oportunidad, podría dar lugar a que se produjese una violación de la seguridad. Las amenazas a la seguridad en una red pueden caracterizarse modelando el sistema como un flujo de información desde una fuente *A* (como por ejemplo un fichero o una región de la memoria principal) a un destino *B* (como por ejemplo otro fichero o un usuario). Las cuatro categorías generales de amenazas son las siguientes y se ilustran en la Figura 2.



ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242

Figura 2: Clasificación de las amenazas según la modificación realizada sobre el flujo normal de datos entre una fuente y un destino.

- *Interrupción*, un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son la destrucción de un elemento hardware, como un disco duro, cortar una línea de comunicación, eliminar un programa o un conjunto de datos, etc.

- *Interceptación*, una entidad no autorizada consigue acceso a un recurso. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa o un ordenador. Ejemplos de este ataque son pinchar una línea para hacerse con datos que circulen por la red y la copia ilícita de ficheros o programas (intercepción de datos), o bien la lectura de las cabeceras de paquetes para desvelar la identidad de uno o más de los usuarios implicados en la comunicación observada ilegalmente (intercepción de identidad).

- *Modificación*, una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad. Ejemplos de este ataque son el cambio de valores en un archivo de datos, alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que están siendo transmitidos a través de la red.

- *Generación*, una entidad no autorizada inserta objetos falsificados en el sistema. Este es un ataque contra la autenticidad. Ejemplos de este ataque son la inserción de mensajes espurios en una red o añadir registros a un archivo.

A la vista de la Figura 1 se comprende que los datos son la parte del sistema más vulnerable, por lo que en este capítulo se centrará la atención en el estudio de los mecanismos disponibles para garantizar la seguridad de los datos.

Por abuso del lenguaje, muchas veces se emplea el término ataque como sinónimo de amenaza, si bien, un ataque es realmente la materialización de una amenaza. Genéricamente, se diferencian dos tipos de ataques:

- *Ataques pasivos*. En los ataques pasivos el atacante no altera la comunicación, sino que únicamente la escucha o monitoriza con el fin de obtener información que está siendo transmitida. Sus objetivos son la intercepción de datos y el análisis de tráfico. Este último ataque constituye una técnica más sutil para obtener información relativa a la obtención del origen y destinatario de la comunicación (mediante la lectura de las cabeceras de los paquetes monitorizados), el control del volumen de tráfico intercambiado entre las entidades monitorizadas (obteniendo así información acerca de los períodos de actividad e inactividad normales o inusuales). Los ataques pasivos son difíciles de detectar, ya que no provocan ninguna alteración de los datos, y pueden realizarse mediante aplicaciones conocidas como *sniffers*.

- *Ataques activos*. Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos y son los que habitualmente suelen sufrir los sistemas informáticos.

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

1.3 Servicios de seguridad

Para hacer frente a las amenazas a la seguridad del sistema se definen una serie de servicios para proteger los sistemas de proceso de datos y de transferencia de información de una organización. Estos servicios hacen uso de uno o varios mecanismos de seguridad. Los servicios de seguridad suelen agruparse en función de la propiedad de la información que tratan de salvaguardar.

- *Confidencialidad*. El servicio de confidencialidad de datos se aplica a todos los datos intercambiados por las entidades autorizadas o, tal vez, sólo a porciones o segmentos seleccionados de los datos, por ejemplo, mediante cifrado. El servicio de confidencialidad de flujo de tráfico protege la identidad del origen y destino(s) del mensaje, por ejemplo enviando los datos confidenciales a muchos destinos además del verdadero, así como el volumen y el momento de tráfico intercambiado.

- *Autenticación*. Este tipo de servicios son imprescindibles cuando se requiere una identificación correcta del origen o destino del mensaje, asegurando que no se trata de falsas entidades. Se distinguen dos tipos: autenticación de entidad (o comprobación de que una entidad es la que se presupone) y de datos de origen (o comprobación de que la fuente de los datos recibidos es la afirmada). Este servicio trata de combatir fundamentalmente la amenaza del enmascaramiento.

- *Integridad*. En este caso, se requiere que la información sólo pueda ser modificada por las entidades autorizadas. La modificación incluye escritura, cambio, borrado, creación y reenvío de los mensajes transmitidos. La integridad de datos asegura que los datos recibidos no han sido modificados de ninguna manera, por ejemplo anexando al mensaje original un resumen cifrado del mismo (o firma digital), mientras que la integridad de secuencia de datos asegura que la secuencia de los bloques o unidades de datos recibidas no ha sido alterada y que no hay unidades repetidas o perdidas, por ejemplo, mediante *timestamps*.

- *Control de acceso*. También se requiere que el acceso a los recursos (información, capacidad de cálculo, nodos de comunicaciones, entidades físicas, etc.) sea controlado y limitado por el sistema destino, mediante el uso de contraseñas, llaves hardware, cortafuegos, etc.

1.4 Mecanismos de seguridad

Los servicios de seguridad definen qué medidas son necesarias adoptar para garantizar los requisitos de seguridad de un sistema, mientras que las técnicas y mecanismos de seguridad determinan cómo se implementan tales medidas. Así pues, una técnica o mecanismo de seguridad es la lógica o algoritmo que implementa un servicio de seguridad particular, bien sea en hardware o software. Aunque no existe un único mecanismo capaz de proveer todos los servicios de seguridad, la mayoría de ellos hacen uso de técnicas criptográficas (Lucena, 2002; Schneier, 1996) basadas en el cifrado de la información [11-18].

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

A continuación se ofrece un recorrido por el "bazar" de las técnicas y mecanismos disponibles para lograr cada uno de los servicios de seguridad comentados en el apartado anterior. Así, para el caso de los servicios orientados a garantizar la confidencialidad de los datos o del flujo de datos se emplean técnicas y métodos como:

- *Cifrado*. Garantiza que la información es inteligible para individuos, entidades o procesos no autorizados (confidencialidad). Consiste en transformar mediante un proceso de cifrado un texto plano en un texto cifrado, gracias a una información secreta o clave de cifrado. Cuando se emplea la misma clave en las operaciones de cifrado y descifrado, se dice que el criptosistema es simétrico. Cuando se utiliza una pareja de claves para separar los procesos de cifrado y descifrado, se dice que el criptosistema es asimétrico o de clave pública. Los criptosistemas de clave pública, aunque más lentos que los simétricos, resultan adecuados para implementar otros mecanismos como la autenticación, la distribución de claves y las firmas digitales.

- *Etiquetas de seguridad*. Los recursos, incluyendo los datos, pueden tener asociadas etiquetas de seguridad, por ejemplo, para indicar el nivel de sensibilidad (a la difusión).

- *Relleno de tráfico*. Los mecanismos de relleno del tráfico se pueden utilizar para proporcionar diversos niveles de protección contra los análisis del tráfico. Se trata de enviar tráfico espurio junto con los datos válidos para que el "adversario" no sepa si se está enviando información o qué cantidad de datos útiles se está transfiriendo. Estos mecanismos sólo pueden ser efectivos si el relleno del tráfico está protegido mediante un método de cifrado.

Para implementar otros servicios, los mecanismos o técnicas empleadas son muy variadas:

- Códigos MAC (*Message Authentication Codes*). Este tipo de códigos se emplean para garantizar la integridad de un mensaje de datos. Los códigos MAC son también etiquetas de autenticación que se obtienen de la aplicación de un mecanismo de autenticación (básicamente, funciones resumen[5] o algoritmos de cifrado de clave secreta), junto con una clave secreta, a partir de los datos a transmitir. Los códigos MAC se caracterizan porque los procesos de generación y verificación utilizan la misma clave.

- *Firmas digitales*. La autenticación del origen de datos puede garantizarse mediante este mecanismo. Se basa en el uso de técnicas criptográficas y suele implementarse con técnicas de cifrado de clave pública. La posesión de una clave privada identifica a un usuario ya que ésta sólo es conocida por el propietario, y, sólo él puede cifrar con ella. Todo el mundo puede verificar la identidad de un usuario descifrando con la clave pública los datos cifrados con la privada. Si son iguales, la firma es correcta, en caso contrario se rechaza. La característica esencial del mecanismo de firma, es que dicha firma sólo puede haber sido generada con la información privada del signatario. Por lo tanto cuando se

[5] Las funciones resumen (*hash functions*) son funciones sin inversa, por lo que si se aplican sobre un mensaje de datos a transmitir, es imposible reconstruir el mensaje original a partir del resumen generado y, además, por su diseño es muy improbable que dos mensajes diferentes generen el mismo resumen. Estas funciones pueden emplearse para implementar un servicio que garantice la integridad (códigos MAC) o la autenticación de un origen de datos (firmas digitales).

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242

verifica la firma, se puede probar que sólo el poseedor de la información privada puede haberla generado [19-25].

- *Terceras partes de confianza* (*TTP*, *Trusted Third Parties*). Para implementar un servicio de no repudio o un servicio de acuse de recibo, es necesario, además de la utilización de firmas digitales, la participación de una tercera parte en la que ambos interlocutores (origen y destino) confían y cuya misión es arbitrar en situaciones de conflicto (existencia o no de un mensaje, negación de la existencia de un mensaje por una de las partes, etc.), de forma que, a requerimiento de uno o ambos interlocutores, la TTP emite un informe de resolución de tales conflictos.

- *Intercambio de autenticaciones*. La autenticación de entidades puede realizarse mediante técnicas de intercambio basadas en la utilización de una información de autenticación (como contraseñas proporcionadas por la entidad emisora y comprobadas por la entidad receptora) o técnicas criptográficas. Si el mecanismo no proporciona una autenticación positiva de la entidad, puede producirse un rechazo o la finalización de la conexión, además de una entrada en el programa de auditoria de seguridad y un informe al centro de gestión de la seguridad. La selección de técnicas de autenticación dependerá de las circunstancias en que deben utilizarse, y, en la mayoría de los casos, deben emplearse junto con sellos de tiempo y de secuencia de los mensajes de autenticación (para evitar el reenvío de estos mensajes), la utilización de protocolos con fase de saludo (*handshake*) de dos o tres vías (para la autenticación unilateral y la autenticación mutua, respectivamente) y servicios de no repudio.

Conviene resaltar que todos los mecanismos poseen tres componentes principales que los caracterizan:

- Una información secreta, como claves y contraseñas, conocidas por las entidades autorizadas.
- Un conjunto de algoritmos, para llevar a cabo el cifrado, descifrado, generación de resúmenes y generación de números aleatorios.
- Un conjunto de procedimientos, que definen cómo se usarán los algoritmos, quién envía qué a quién y cuándo.

2 Criptografía

2.1 Introducción

Según el diccionario de la R. A. E., la palabra criptografía se define como el “Arte de escribir mensajes con una clave secreta o de modo enigmático". Esta definición es un poco desafortunada para definir lo que hoy en día se entiende por criptografía. En primer lugar, no se trata de un arte sino de una ciencia, los sistemas actuales usan más de una clave, no todos los sistemas utilizan claves secretas (los sistemas de clave pública utilizan dos: una privada o secreta y otra pública), y por último la representación final de los mensajes cifrados es binaria que aunque se trata de una representación enigmática para los seres humanos, no lo es para los computadores (pues es su lenguaje natural).

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242

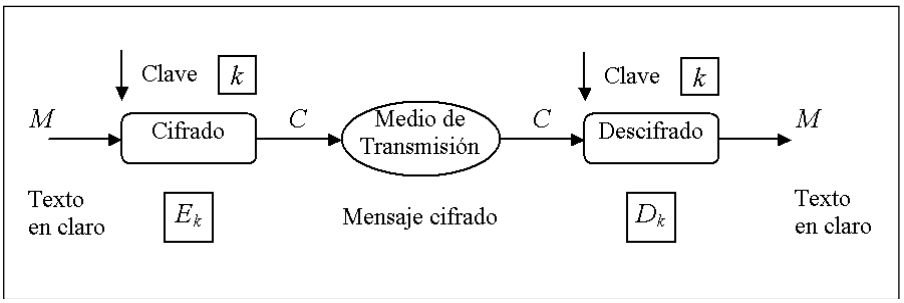


Figura 3: Esquema genérico de un criptosistema.

Así pues, es necesario dar una definición más adecuada del concepto de criptografía (Lucena, 2002; Schneier, 1996). Una posible definición es la siguiente:

> *Rama de las Matemáticas, y en la actualidad de la Informática y la Telemática, que hace uso de métodos y técnicas matemáticas con el objetivo principal de cifrar un mensaje o archivo por medio de un algoritmo, usando una o más claves Esto da lugar a los criptosistemas que permiten asegurar cuatro aspectos fundamentales de la seguridad informática: la confidencialidad, integridad, disponibilidad y no repudio de emisor y receptor.*

Surge así la noción de criptosistema, cuyo esquema básico se representa en la Figura 3, y que consta de cinco elementos distintos:

- El espacio de mensajes $\boldsymbol{M}$ representa el conjunto de todos los posibles mensajes sin cifrar (lo que se denomina texto en claro, *plaintext*) que se pueden transmitir, es decir, $\boldsymbol{M} = \{m_1, m_2, ..., m_{|M|}\}$.

- El espacio de criptogramas $\boldsymbol{C}$ representa el conjunto de todos los posibles mensajes cifrados.

- El espacio de claves $\boldsymbol{K}$ contiene todas las posibles claves que pueden utilizarse en el criptosistema, es decir, $\boldsymbol{K} = \{k_1, k_2, ..., k_{|K|}\}$. Si el espacio de claves $\boldsymbol{K}$ fuera tan grande como el espacio de mensajes $\boldsymbol{M}$, es decir, si $|\boldsymbol{M}| \approx |\boldsymbol{K}|$, se podría obtener un criptosistema perfecto. En la práctica, es imposible mantener un espacio de claves tan grande como el espacio de mensajes, por lo que puede considerarse que $|\boldsymbol{M}| >> |\boldsymbol{K}|$.

- Una vez elegida una clave k del espacio de claves $\boldsymbol{K}$, cualquier función que, basándose en dicha clave, permiten transformar un elemento de $\boldsymbol{M}$ (un mensaje de texto en claro) en un elemento de $\boldsymbol{C}$ (un mensaje cifrado) se denomina transformación de cifrado, E_k. Por lo general, esta transformación viene dada por un algoritmo o procedimiento y debe asumirse que dicho algoritmo es siempre de dominio público, por lo que habitualmente su código fuente estará disponible.

- Las transformaciones inversas, es decir, cualquier aplicación que para una clave $k \in \boldsymbol{K}$, sobre un elemento de $\boldsymbol{C}$ (un mensaje cifrado) devuelve un elemento de $\boldsymbol{M}$ (un mensaje en texto claro), se denominan transformaciones de descifrado, D_k. Por lo general, la transformación D_k es la operación inversa de E_k, aunque también es posible que ambas transformaciones coincidan, pero se aplique la transformación de descifrado empleando una

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

clave *k*', que sea la inversa (bajo determinadas condiciones) de la clave *k* utilizada en el proceso de cifrado. Dado que en la mayoría de criptosistemas, los espacios de mensajes y de criptogramas son de igual magnitud, es posible que existan claves para las cuales la transformación de cifrado sobre un mensaje genere el mismo mensaje (es decir, que $E_k(m) = m$) o que con el cifrado de un mensaje ya cifrado se obtenga el mensaje original (es decir, $E_k(E_k(m)) = m$). A estas claves se las denomina claves débiles (*weak keys*) [26-35].

Es aconsejable que los criptosistemas cumplan además una serie de características adicionales:

- El algoritmo de cifrado/descifrado debe ser rápido y fiable.
- No debe existir un retardo significativo debido al proceso de cifrado o descifrado.
- La seguridad del sistema deberá residir sólo en el secreto de una clave y no de las funciones de transformación.
- La fortaleza del sistema se entenderá como el coste computacional asociado al proceso de romper la cifra o encontrar la clave secreta.

Existen dos criterios principales a la hora de clasificar los sistemas de cifrado. Por un lado, y según el tratamiento que de los mensajes a cifrar hacen los algoritmos se habla de:

- *Métodos de cifrado en bloque*. En estos sistemas, el mensaje original se trocea en bloques del mismo tamaño sobre los que se aplica el mismo algoritmo de cifrado, sucesivas veces, y empleando la misma clave. El cifrado en bloque presenta ventajas como la alta difusión de los elementos en el criptograma o que es fácilmente detectable la inserción de bloques falsos. Como principales desventajas destacan, la baja velocidad de cifrado (con respecto a los métodos de cifrado en flujo) al tener que leer el bloque y que un error en el cifrado de un elemento de información se propagará a todo el bloque. Algoritmos de esta clase son, por ejemplo, DES, IDEA, RSA.

- *Métodos de cifrado en flujo*. Estos métodos se caracterizan por aplicar el algoritmo de cifrado a un elemento de información (carácter, *bit*) mediante un flujo de clave siendo ésta (en teoría) aleatoria y mayor que el mensaje. Las ventajas de este tipo de métodos son: la alta velocidad de cifra y que es resistente a los errores (ya que cada elemento se cifra independientemente del resto). Por el contrario, presenta como deficiencias, la baja difusión de los elementos en el criptograma y que es vulnerable a la modificación de elementos por separado. Entre los algoritmos de esta categoría destaca el algoritmo A5 empleado en telefonía móvil.

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242

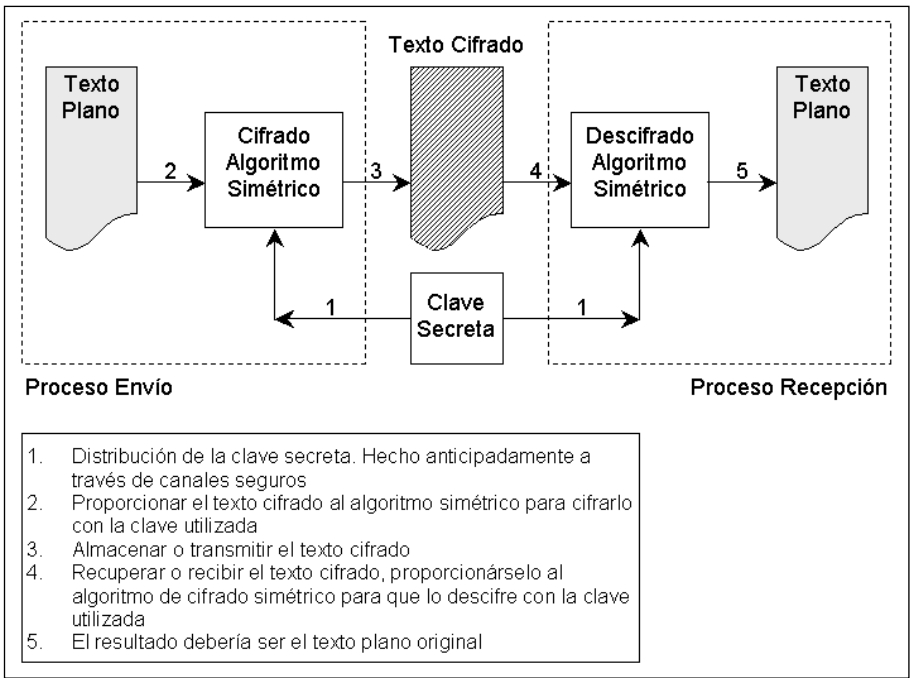


Figura 4: Esquema general de un algoritmo de cifrado de clave secreta.

En segundo lugar, y de acuerdo con el uso de las claves que utilizan los algoritmos, se habla de:

- *Criptosistemas simétricos o de clave secreta* (véase Figura 4). Son aquellos que emplean la misma clave *k* tanto para cifrar como para descifrar. Presentan el inconveniente de que para ser empleados, el emisor y el receptor deben de conocer la clave *k*, lo cual plantea el problema de cómo transmitir tal clave sin comprometerla (es decir, cómo se distribuyen las claves). Por el contrario, son mucho más rápidos que los métodos de clave pública (q) y resultan apropiados para el cifrado de grandes volúmenes de datos. Algoritmos de cifrado simétricos son DES, IDEA, 3DES, AES.

- *Criptosistemas asimétricos o de clave pública* (véase Figura 5). Este tipo de criptosistemas emplean un par de claves $\langle k_P, k_S \rangle$. La clave k_S se conoce como clave privada y la clave k_P como clave pública. Una de ellas sirve para la transformación de cifrado *E* y la otra para la transformación *D* de descifrado, que en este tipo de algoritmos coinciden. En muchos casos son intercambiables, esto es, si se utiliza una para cifrar la otra sirve para descifrar y viceversa. Estos criptosistemas deben cumplir además, que el conocimiento de la clave pública k_P no permite calcular la clave privada k_S. Ofrecen un abanico superior de posibilidades, pudiendo emplearse para establecer comunicaciones seguras por canales inseguros (puesto que únicamente viaja por el canal la clave pública, que sólo sirve para cifrar), o para llevar a cabo autenticaciones. Los algoritmos RSA, Diffie-Hellman pertenecen a esta clase de algoritmos. En la práctica, se emplea una combinación de estos dos tipos de criptosistemas, de forma que los mensajes se cifran mediante un algoritmo de clave secreta (del tipo 3DES o IDEA, que son más rápidos que los algoritmos asimétricos) y la clave empleada por estos se transmite mediante criptografía asimétrica.

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

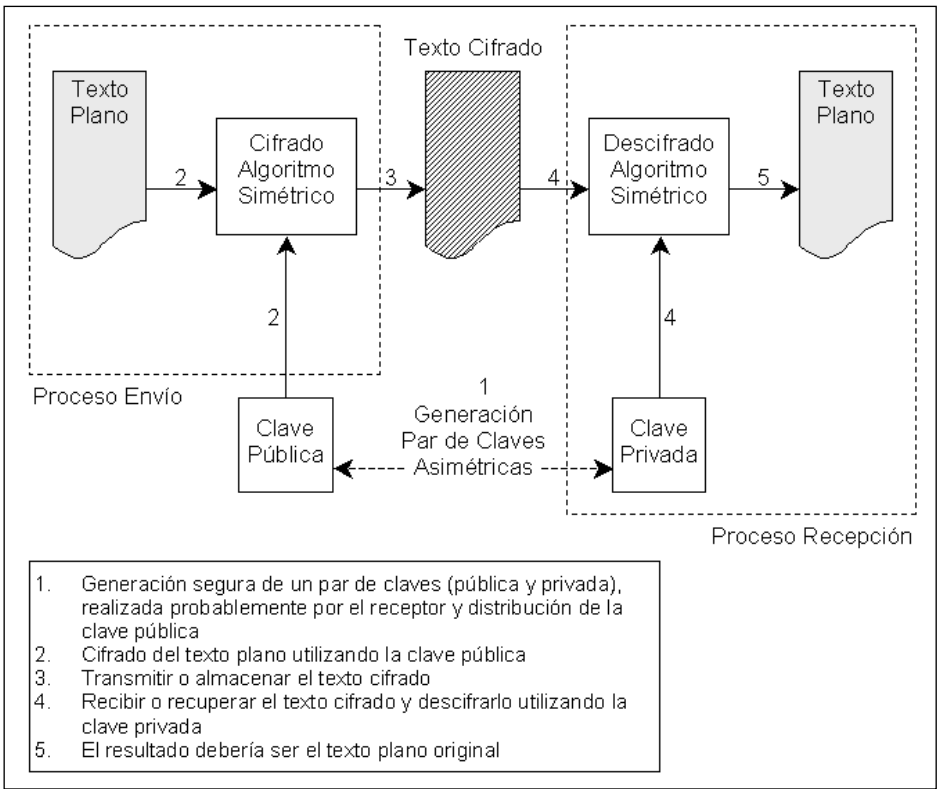


Figura 5: Esquema general de un algoritmo de cifrado de clave pública.

Por último, y para finalizar este apartado, conviene introducir la noción de criptoanálisis que consiste, básicamente, en el conjunto de técnicas y procedimientos encaminados a comprometer la seguridad de un criptosistema. Esto se puede hacer descifrando un mensaje sin conocer la clave, o bien obteniendo a partir de uno o mas criptogramas la clave que ha sido empleada en el proceso de cifrado. No se considera criptoanálisis el descubrimiento de un algoritmo secreto de cifrado. De hecho, éste es un principio estable de la criptografía (principio de Kerckhoffs, 1883): "La seguridad de un criptosistema no debe depender de mantener secreto el algoritmo. La seguridad depende sólo de mantener secreta la clave".

Existen sistemas idealmente seguros, capaces de resistir cualquier ataque, pero en la práctica carecen de interés (ya que implican que la clave sea aleatoria y de longitud infinita, lo cual plantea problemas de almacenamiento y distribución). Así pues, es necesario adoptar una solución de compromiso que conjugue el coste del sistema (computacional, espacial y económicamente) y su resistencia a diferentes ataques criptográficos [36-42].

2.2 Criptografía simétrica o de clave secreta (*secret key cryptography*)

Este apartado se centra en los algoritmos de cifrado en bloques y simétricos. La inmensa mayoría de estos algoritmos se apoyan en los conceptos de confusión y difusión inicialmente propuestos por Shannon y que se combinan para dar lugar a los denominados *cifrados de producto*. La confusión consiste en tratar de ocultar la relación que existe entre el texto en claro, el texto cifrado y la clave. Un buen mecanismo de confusión hará demasiado complicado extraer relaciones estadísticas entre las tres partes. Por un lado, la difusión trata de repartir la influencia de cada *bit* del mensaje original lo más posible entre el mensaje cifrado. Conviene destacar que la confusión por

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242

sí sola será suficiente, ya que si se establece una tabla de sustitución completamente diferente para cada clave con todos los textos en claro posibles se tendrá un sistema extremadamente seguro. Sin embargo, dichas tablas ocuparían grandes cantidades de memoria, por lo que en la práctica serán inviables. Lo que en realidad se hace para conseguir algoritmos fuertes (sin necesidad de almacenar tablas enormes) es intercalar la confusión (sustituciones simples, con tablas pequeñas) y la difusión (permutaciones). Esta combinación se conoce como cifrado de producto. La mayoría de los algoritmos se basan en diferentes capas de sustituciones y permutaciones, estructura que se denomina Red de Sustitución-Permutación. En muchos casos el criptosistema no es mas que un paso simple de sustitución-permutación repetido *n* veces, como ocurre con el algoritmo DES.

El algoritmo DES ha sido uno los algoritmos simétricos más ampliamente extendido al haber sido adoptado como estándar para las comunicaciones seguras por el Gobierno de los EE.UU. En realidad, la NSA (*National Security Agency*) lo diseño para ser implementado por hardware, con la intención de mantenerlo en secreto, pero al parecer por un malentendido entre la Agencia y la Oficina Nacional de Estandarización, su especificación se hizo publica con suficiente detalle como para que pudiera ser implementada por software. A mediados de 1998, se demostró que un ataque por la fuerza bruta a DES era viable, debido a la escasa longitud que emplea en su clave. No obstante, el algoritmo a un no ha demostrado ninguna debilidad grave desde el punto de vista teórico, por lo que su estudio sigue siendo plenamente interesante.

El algoritmo Lucifer, diseñado por Feistel en IBM durante la década de los 70 y adoptado por el NIST (*National Institute of Standards and Technology*) como estándar de cifrado de datos comerciales (DES, *Data Encryption Standard*). Ha sido uno de los algoritmo de cifrado simétrico más utilizado. Se ha demostrado su fortaleza ante diferentes ataques, si bien su mayor debilidad ha sido la escasa longitud de la clave de cifrado empleado. En la década de los 90 ha dejado de ser el estándar empleado por la NIST. Otros algoritmos de cifrado en bloque simétricos son: el algoritmo IDEA (que se considera uno de los mejores sistemas disponibles para uso comercial en la actualidad) y el nuevo estándar de cifrado avanzado del NIST, el AES [43-50].

El algoritmo DES presenta algunas claves débiles. En general, todos aquellos valores de la llave que conducen a una secuencia inadecuada de subclaves K_i serán poco recomendables. Se distingue entre claves débiles, que son aquellas que generan un conjunto de dieciséis valores iguales de K_i (y que cumplen $E_K(E_K(M)) = M$), y claves semidébiles, que generan dos valores diferentes de K_i, cada uno de los cuales aparece ocho veces. En cualquier caso, el numero de claves de este tipo es tan pequeño en comparación con el numero total, que el riesgo puede considerarse insignificante. Sin embargo, a mediados de julio de 1998, una empresa sin animo de lucro, denominada EFF (*Electronic Frontier Foundation*), logró fabricar una maquina capaz de descifrar un mensaje DES en menos de tres días. *DES–Cracker* costó menos de 240.000 €.

Para tratar de paliar las debilidades que se derivan de la escasa longitud de clave del algoritmo DES se han propuesto varias alternativas que se recogen a continuación:

- DES múltiple. Consiste en aplicar varias veces el algoritmo DES con diferentes claves al mensaje original. Esto es posible puesto que DES no presenta estructura de grupo[6] y por

[6] Un criptosistema se dice que tiene estructura de grupo si se cumple siempre que cifrar un mensaje M con una clave K_1 y luego el resultado con una clave K_2, es equivalente a cifrar el mensaje original con una única clave K_3.

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

lo tanto admite cifrado múltiple con lo cual se aumenta el tamaño efectivo de la clave. El más común de todos ellos es 3DES, que responde a la siguiente estructura $C = E(K_1, E^{-1}(K_2, E(K_1, M)))$. Es decir, se codifica con la subclave K_1, se decodifica el resultado con la subclave $K2$ y se vuelve a codificar con la subclave K_1. De esta forma, la clave resultante es la concatenación de K_1 y K_2 y tiene una longitud de 112 bits.

- DES con subclaves independientes. Consiste en emplear subclaves diferentes para cada una de las 16 rondas de DES. Puesto que estas subclaves son de 48 bits, la clave resultante tendría 768 bits en total.

- DES con *S-box* alternativas. Consiste en utilizar tablas de sustitución diferentes a las de la versión original de DES. En la práctica no se han encontrado *S-box* mejores que las propias de DES.

Independientemente del algoritmo de cifrado simétrico que se utilice (DES, 3DES, IDEA, CAST, Blowfish, Rijndael, etc.) su utilización para cifrar mensajes en bloque se hace siguiendo diferentes modos de operación (NIST, 1980) que se describen a continuación:

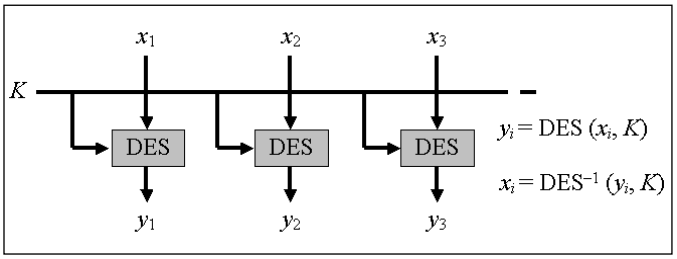


Figura 6: Diagrama de operación del modo ECB.

- Modo ECB (*Electronic CodeBook*). Es el método mas sencillo y obvio de aplicar un algoritmo de cifrado por bloques (Figura 6). Simplemente se subdivide la cadena que se quiere codificar en bloques del tamaño adecuado y se cifran todos ellos empleando la misma clave. Por lo tanto, el resultado es similar al que se obtendría si se codificase mediante un gran libro electrónico de códigos, donde la entrada al libro sería el bloque a cifrar y el código que lo sustituye (en este sentido conviene recordar que codificar no es lo mismo que cifrar) sería el criptograma asociado.
  Este modo presenta como principales debilidades que se podría reconstruir ese libro electrónico sin necesidad de conocer la clave, que presenta el problema de comienzos y finales fijos que permiten un tipo de ataque sencillo y que el sistema puede atacarse a través de la repetición de bloques similares.

- Modo CBC (*Cipher Book Chaining*). Este modo incorpora un mecanismo de retroalimentación en el cifrado por bloques (véase la Figura 7). Esto significa que la codificación de los bloques anteriores condiciona la codificación del bloque actual, por lo que será imposible sustituir un bloque individual en el mensaje cifrado. Esto se consigue efectuando una operación XOR entre el bloque del mensaje que quiere cifrarse y el último criptograma obtenido.

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

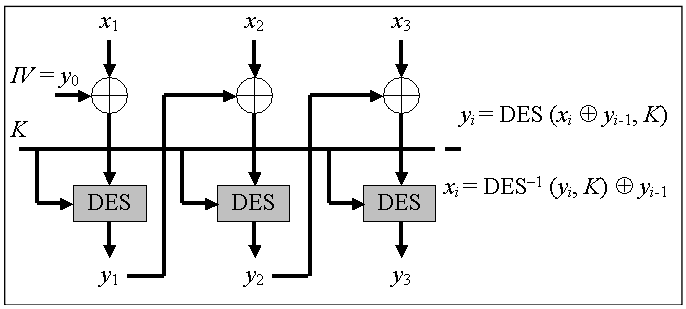


Figura 7: Diagrama de operación del modo CBC.

Con este modo de operación se evita el ataque por repetición de bloque ya que se enmascara el mensaje con la propia secuencia cifrante. Ahora bien, un error de cifrado se propaga a los bloques contiguos.

- Modo CFB (*Cipher FeedBack*). En este modo se permite la codificación de la información en unidades inferiores al tamaño del bloque, lo cual permite aprovechar totalmente la capacidad de transmisión del canal de comunicaciones, manteniendo además un nivel de seguridad adecuado (véase la Figura 8).

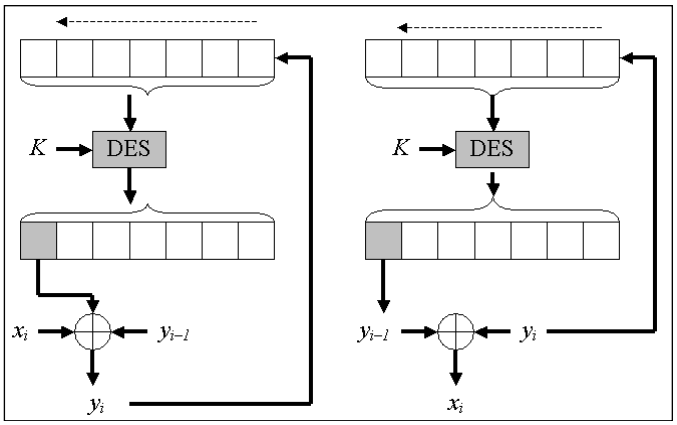


Figura 8: Diagrama de operación del modo CFB.

Otro algoritmo de cifrado simétrico bien conocido es IDEA (Lai y Massey, 1991). El historial de IDEA es el siguiente:

- En 1990 Xuejia Lai y James Massey proponen el algoritmo PES (*Proposed Encryption Standard*).
- En 1991 y debido a los avances de Biham y Shamir en el criptoanálisis diferencial, los autores proponen el IPES (*Improved Proposed Encryption Standard*).
- En 1992 Lai y Massei proponen finalmente el algoritmo IDEA (International Data Encryption Algorithm).
- En 1999 el algoritmo IDEA, mucho más seguro que el algoritmo DES o cualquiera de sus versiones, se comienza a usar ampliamente en el sistema de correo electrónico PGP.

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

Básicamente, el algoritmo IDEA cifra bloque de 64 bits en 8 vueltas. Para ello divide la entrada *M* en cuatro bloques de 16 bits y genera 52 subclaves a partir de una única clave de 128 bits. En cada vuelta se usan 6 claves para modificar el mensaje y al final hay una transformación adicional con 4 claves con el fin de invertir la operación inicial. Las transformaciones realizadas están definidas en función de la operación XOR, de la suma (modulo 2^{16}) y multiplicación (módulo 2^{16}).

IDEA ha demostrado ser inmune ante el criptoanálisis diferencial ya que sus autores conocían la debilidad de DES respecto a este ataque y diseñaron IDEA de forma que fuera resistente. En 1992 Joan Daemen descubre una clase de claves débiles aunque la probabilidad de que se elija de forma aleatoria una clave de este grupo es sólo de una entre 2^{96} (además, la utilización de estas claves pueden evitarse en la implementación del algoritmo). Hasta la fecha no se conoce todavía ningún sistema o algoritmo de ataque que haya criptoanalizado con éxito el algoritmo IDEA, por lo que se considera como uno de los algoritmos simétricos más seguro que existe en la actualidad.

En 1997 el NIST no certifica al algoritmo DES (que hasta entonces se había empleado como estándar de cifrado en bloque para todas las aplicaciones comerciales en los Estados Unidos) y saca a concurso público el diseño de un nuevo estándar de cifrado que se denomina AES (*Advanced Encryption Standard*) (NIST, 2001). Desde entonces los pasos seguidos han sido los siguientes:

- En 1998 el NIST anuncia la preselección de 15 candidatos con posibilidades de alcanzar el estándar: Cast–256 Crypton, Deal, Dfc, E2, Frog, Hpc, Loki97, Magenta, MARS, RC6, Rijndael, Safer+, Serpent, Twofish.
- En 1999 el NIST depura el número de candidatos a tan sólo cinco: MARS (IBM), RC6TM (RSA Laboratories), Rijndael (Joan Daemen y Vincent Rijmen), Serpent (Ross Anderson, Eli Biham y Lars Knudsen) y Twofish (Bruce Schneier, John Kesey, Doug Whiting, David Wagner, Chris Ha y Niel Ferguson).
- Tras sucesivas reuniones, pruebas y debates, en Octubre de 2000 se adopta el sistema Rijndael como nuevo estándar.
- Finalmente, en 2001 se pone en marcha el nuevo estándar publicándose toda la documentación y especificaciones necesarias para su implementación.

Como características principales de Rijndael (Daemen y Rijmen, 2000) cabe destacar que no se trata de un algoritmo del tipo Feistel, el tamaño de la clave es variable: 128, 192 y 256 bits (estándar), con tamaño del bloque de texto también variable: 128 bits o un múltiplo de 4 bytes, realiza operaciones modulares a nivel de byte, posee un número de etapas flexible según las necesidades del usuario y usa un conjunto de tablas de sustitución (*S–box*) al estilo de las utilizadas por el algoritmo DES.

Las diferentes transformaciones operan sobre un resultado intermedio denominado estado y que puede ser representado por un matriz rectangular de bytes, con cuatro filas y un número variable de columnas N_b que coincide con el tamaño del bloque considerado dividido entre 32. La clave de cifrado puede representarse de forma análoga como una matriz de 4 filas y N_k columnas. Los bytes de entrada al algoritmo (bloque de texto en claro a cifrar en el modo ECB) se disponen sobre la matriz de estado en el siguiente orden a_{00}, a_{10}, a_{20}, a_{30}, a_{01}, a_{11}, a_{21}, a_{31}, a_{41}, ... , y los bytes de la clave de cifrado también se disponen en ese mismo orden (es decir, se rellenan las matrices por columnas). El número de etapas del algoritmo N_r, depende de los valores de N_b y de N_k.

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242

En cuanto a la generación de claves empleadas en cada etapa, éstas se derivan a partir de la clave de cifrado. El proceso se realiza en dos pasos: expansión de la clave y selección de las claves intermedias empleadas en cada etapa. El número total de bits de la claves intermedias debe ser igual al tamaño del bloque de datos a cifrar multiplicado por el número de rondas más una (por ejemplo, si el tamaño del bloque es de 128 bits y se realizan 10 etapas, el total de bits de la clave expandida debe ser 1408). Por este motivo se requiere un proceso de expansión de la clave original para obtener la clave extendida. En cada ronda se toman los bits de la clave expandida como sigue: en la primera ronda se toman los N_b bits primeros bits de la clave expandida, en la segunda ronda los siguientes N_b bits de la clave expandida y así sucesivamente. Las operaciones de cada una de las etapas de transformación y la etapa final son básicamente las siguientes:

- Una operación *ByteSub* que desempeña una función análoga a las de las tablas de sustitución del tipo *S–box*.
- La operación *ShiftRow* traslada cíclicamente cada fila un número de bytes independientes.
- La operación *MixColumn* equivale a una multiplicación de matrices por polinomios.
- Finalmente, la operación *AddRoundKey* aplica la clave intermedia correspondiente a la etapa en curso al resultado de las operaciones anteriores mediante una operación XOR.

2.3 Criptografía asimétrica o de clave pública (*public key cryptography*)

Los sistemas de cifrado con clave secreta presentan varios inconvenientes relacionados con la gestión y distribución de claves y con el hecho de no poseer firma digital. La mala gestión de las claves se debe a que cuando se tiene un número grande de usuarios, *n*, las necesidades de almacenamiento de tales claves son del orden de $O(n^2)$. En cuanto a la distribución de claves, la criptografía simétrica por sí sola no dispone de ningún mecanismo para enviar, de forma segura, una clave a través de un canal de comunicaciones inseguro. Por último, aunque sí es posible autenticar los mensajes mediante una marca, no es posible firmar digitalmente los mismos (proporcionar un elemento que la otra parte pueda cotejar para autenticar el origen del mensaje). Aún así, la criptografía simétrica se utiliza y la razón fundamental se debe a la eficiencia ya que son de 100 a 1000 veces más rápidos que los algoritmos asimétricos [51-55].

Como se ha mencionado con anterioridad, la criptografía de clave pública se basa en que cada usuario tiene dos claves, una secreta o privada y otra pública. Además, cada una de las claves es la inversa de la otra para alguna operación concreta definida sobre un cuerpo finito. La criptografía asimétrica se basa en el uso de "funciones unidireccionales con trampa" (*one–way functions*) que son funciones matemáticas de un solo sentido, ya que su cálculo es fácil en sentido directo (y por eso se usan para cifrar o descifrar), pero de cálculo muy difícil en sentido inverso, es decir, en aquellos casos en los que se quiera atacar o criptoanalizar el sistema de cifra. Un ejemplo de este tipo de funciones es la exponenciación discreta $y = a^x \bmod n$. Esta operación no es costosa de realizar (hablando desde un punto de vista computacional) puesto que existen algoritmos eficientes de exponenciación rápida. Por el contrario, el cálculo de su inverso, lo que se denomina el logaritmo discreto ($x = \log_a y \bmod n$) es, desde un punto de vista computacional muy costoso, puesto que no existe ningún algoritmo que permita calcularlo en un tiempo razonable. Otro ejemplo de función unidireccional es el del producto de dos primos grandes $p*q = n$, ya que existen algoritmos eficientes para calcular el producto, pero no para factorizar un número grande, es decir, partiendo de *n*, encontrar los dos factores tal que $n = p*q$. Las funciones unidireccionales que desde el punto de vista de la criptografía son útiles son las que se denominan funciones unidireccionales con trampa, es decir, aquellas que conociendo una determinada información permiten

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242

deshacer la función fácilmente (básicamente aplicando la misma función al criptograma, pero con otros parámetros de la función, convenientemente elegidos).

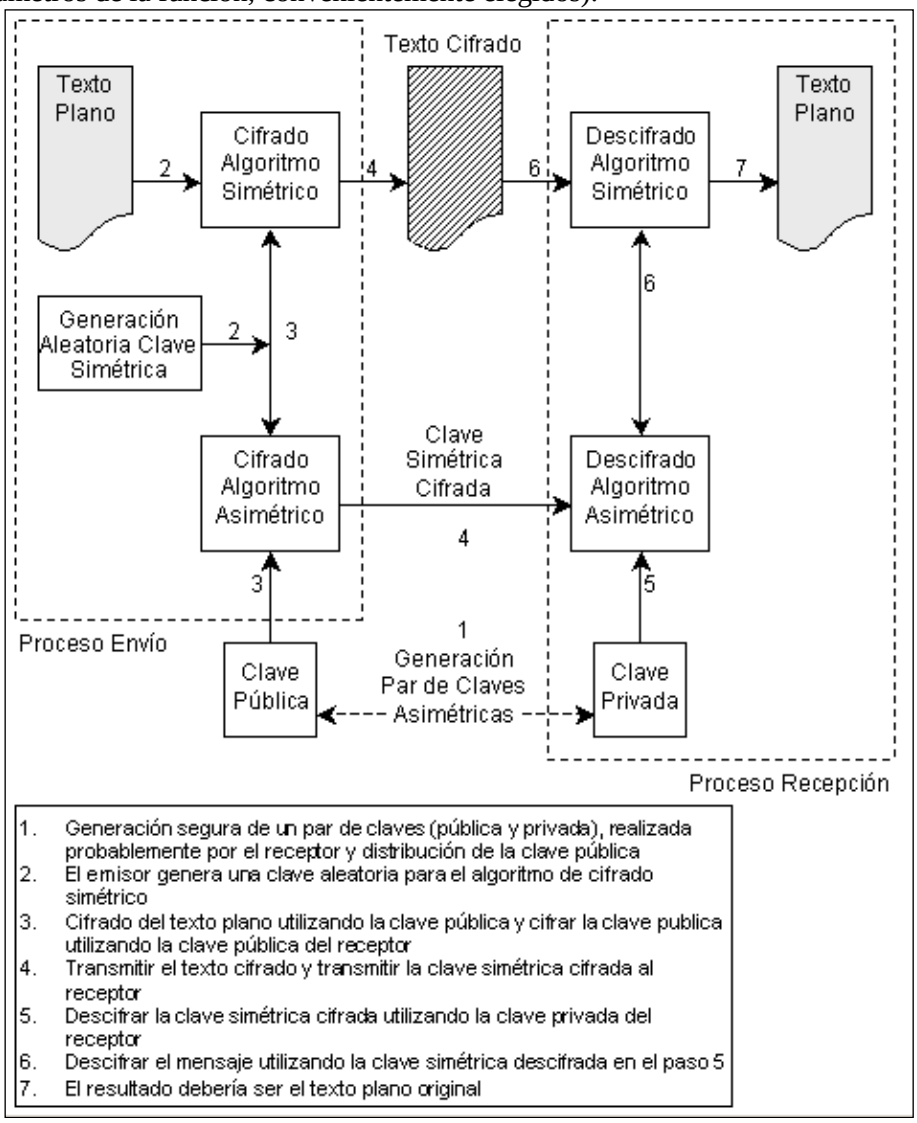


Figura 9: Envoltorio digital o cifrado clave de sesión.

Así pues, en un sistema de clave pública en el que intervienen dos interlocutores *A* y *B*, entrarán en juego dos pares de claves: el par $\langle p_A, s_A \rangle$ representa a las claves pública y privada de *A*, respectivamente, y el par $\langle p_A, s_A \rangle$ a las claves pública y privada de *B*. Como las claves privadas se mantienen en secreto sólo es posible proceder de una de las tres formas que se citan a continuación:

- El emisor del mensaje (por ejemplo *B*) cifra con la clave pública p_A del receptor *A*. De esta forma se garantiza la confidencialidad del mensaje enviado puesto que el receptor *A* puede descifrar el criptograma recibido utilizando su clave privada s_A (conviene recordar que su clave privada será la inversa de la clave pública utilizada en origen para cifrar el mensaje).

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242

- El caso en que el emisor del mensaje *B* utiliza su propia clave pública p_B para cifrar un mensaje no tiene sentido bajo el punto de vista de los sistemas de clave pública ya que sólo el emisor sería capaz de descifrar el criptograma (deshacer la operación de cifra) con su propia clave privada. Esta situación solo sería de interés para cifrar de forma local su propia información, pero para ello ya están los sistemas de clave secreta que son mucho más rápidos.

- La otra posibilidad factible es que el emisor *B* utilice su clave privada s_B para cifrar un mensaje. En este caso se obtiene una firma digital que le autentica como emisor ante el destinatario (sólo el poseedor de la clave pública que permite descifrar el criptograma puede ser la fuente del mensaje, aunque no necesariamente su identidad) y, además, a este último le permitirá comprobar la integridad del mensaje.

Recapitulando y comparando ambos sistemas, los sistemas asimétricos son más eficientes en la gestión de claves (ya que sólo es necesario que un usuario memorice su clave privada), los espacios de claves no son comparables en ambos sistemas (puesto que en un sistema simétrico será del orden de la centena, típicamente mayor o igual a 128 bits, mientras que en los asimétricos será del orden de miles de bits, típicamente se recomienda una longitud de clave igual o superior a 1024 bits). En lo que se refiere a la vida de las claves, en los sistemas simétricos el tiempo de vida de una clave es corta y comprende típicamente el tiempo de una sesión o comunicación (de segundos a minutos), mientras que la duración de una clave pública es mucho mayor (de meses a años). En cuanto a la autenticación, los sistemas asimétricos, a diferencia de los simétricos, permiten la firma digital de los mensajes. Por último y en relación con la velocidad de cifra, los sistemas simétricos son de 100 a 1000 veces más rápidos que los asimétricos. Ante esta situación, la solución comúnmente adoptada es mixta y se conoce como cifrado de la clave de sesión o sobre digital (*digital envelope*). Básicamente (véase Figura 9), consiste en que una de las partes genera una clave de forma aleatoria, que será utilizada como clave secreta por los algoritmos de cifrado simétricos que utilicen ambas partes, pero que el emisor dará a conocer al receptor mediante el uso de criptografía asimétrica (es decir, cifrando esta clave con la clave pública del receptor) [56-60].

Los antecedentes de la criptografía asimétrica se encuentran en el protocolo de intercambio de claves propuesto por Diffie y Hellman (1976). El protocolo propuesto se basa en la aritmética modular, en concreto en los grupos finitos multiplicativos, asociados a un número primo *Q* y a un generador *P* de dicho número primo, que se hacen públicos. Una vez seleccionados los parámetros (en el ejemplo, $Q = 11$ y $P = 7$), los pasos a seguir por ambas partes son los siguientes:

A	B
i) *A* elige al azar un número que mantiene en secreto *a*, por ejemplo $a = 3$.	i) Elige otro número *b*, por ejemplo, $b = 6$.
ii) Calcula $P^a \bmod Q = 7^3 \bmod 11 = 2$	ii) Calcula $P^a \bmod Q = 7^6 \bmod 11 = 4$
iii) El resultado se denomina α	iii) El resultado se denomina β
A y B se intercambian los valores de α y β. Es posible que una tercera parte C intercepte estos números y que también esté en posesión de los valores de P y Q, pero le será difícil invertir las funciones empleadas.	
iv) Se calcula $\beta^a \bmod Q = 4^3 \bmod 11 = 9$	iv) Se calcula $\alpha^a \bmod Q = 2^6 \bmod 11 = 9$
El número 9 puede emplearse como clave de la comunicación entre A y B.	

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

Por lo tanto, el secreto compartido por ambas partes es el valor de P^{ab} mod Q, por lo que si un intruso C se hace con las claves públicas P y Q y alguno de los mensajes enviados (α o β) se enfrentará al problema del logaritmo discreto para descubrir la clave. Este problema es intratable (no existe un método de calculo eficiente desde un punto de vista computacional) cuando el número primo elegido es grande (de al menos 512 bits).

Un año después de la aparición del protocolo de intercambio de claves propuesto por Diffie y Hellman, otros tres investigadores propusieron el algoritmo de cifrado asimétrico que hasta la fecha de hoy se ha convertido en el algoritmo simétrico más ampliamente utilizado. Este algoritmo conocido como RSA (Rivest *et al.*, 1978) y cuyo nombre está formado a partir de las iniciales de sus tres autores: Ron Rivest, Adi Shamir y Leonard Adleman) obedece al siguiente algoritmo:

1. Cada usuario elige un número N tal que $N = P * Q$, siendo P y Q dos números primos que se mantienen en secreto. El algoritmo realiza operaciones modulares en $\mathbf{Z}_N$.
2. Cada usuario calcula el siguiente valor $\phi(N) = (P - 1)*(Q–1)$.
3. Cada usuario elige una clave pública e que esté en $\mathbf{Z}_N$ de modo que e y $\phi(N)$ sean primos entre sí, es decir, que el m.c.d $[e, \phi(N)] = 1$.
4. Bajo estas condiciones está garantizado que existe un número d en $\mathbf{Z}_N$ que es el inverso e, es decir, $d = \text{inv}[e, \phi(N)]$ tal que $e*d \bmod \phi(N) = 1$. Este valor puede calcularse de forma eficiente mediante el algoritmo extendido de Euclides.
5. El usuario da a conocer su clave pública compuesta por la pareja de números $\langle N, e\rangle$, mientras que la clave secreta es el valor de d. Una vez elegida el valor de e y calculado el valor de d, los valores P, Q y $\phi(N)$ no tienen que memorizarse.

Así por ejemplo, suponiendo el grupo generado por $N = 187 = 17 * 11$, de donde se obtiene que $\phi(N) = (17–1)*(11–1) = 160$. Suponiendo que se elige como clave pública el valor de $e = 7$ (valor que cumple que m.c.d(7, 160) = 1), entonces existe un único inverso d, en el ejemplo, $d = \text{inv}[7, 160] = 23$. Así pues, la clave pública viene dada en este ejemplo por el par $\langle 187, 7\rangle$. Se supone además, que se quiere enviar un mensaje que consiste únicamente en un número (por ejemplo, el valor de una clave que se empleará posteriormente por un algoritmo de cifrado simétrico), por ejemplo, $M = 88$.

- Para cifrar el mensaje basta con que el emisor del mismo realice la siguiente operación $C = M^e \bmod N$, donde los valores de e y N son conocidos al ser la clave pública. En el ejemplo, $C = 88^7 \bmod 187 = 11$ siendo este valor el que se transmite.

- Para descifrar el criptograma recibido, es necesario realizar la operación de exponenciación con la clave privada, es decir, $M = C^d \bmod N = 11^{23} \bmod 187 = 88$, con lo que se recupera el valor original de M a partir del valor de C.

La fortaleza del algoritmo RSA se fundamente en que si un intruso quiere conocer la clave secreta d a partir de los valores públicos N y e se enfrentará al problema de la factorización de números

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242

grandes[7], ya que la solución para conocer la clave privada pasa por conocer el valor de $\phi(N)$ y así poder encontrar el inverso de la clave pública.

3 Autenticación

Esta sección se centra en una de las aplicaciones más interesantes de la criptografía asimétrica, que permite firmar digitalmente un mensaje y, por lo tanto, proporcionar un mecanismo para llevar a cabo la autenticación. En esta sección se introducirá la noción de firma digital, así como, conceptos como funciones resumen (*hash functions*), certificados digitales, autoridades de certificación, etc.

Como ya se ha mencionado, la firma digital se obtiene cifrando un mensaje con la clave privada del emisor de un mensaje. De esta forma, el receptor puede descifrar dicho mensaje con la clave pública del emisor y comprobar así la integridad y autenticación del mensaje. Dado que los sistemas de clave pública son relativamente lentos, en vez de firmar digitalmente el mensaje completo, la firma que se adjunta a los mensajes se realiza sobre un resumen (típicamente del orden de la centena de bits) del mensaje original.

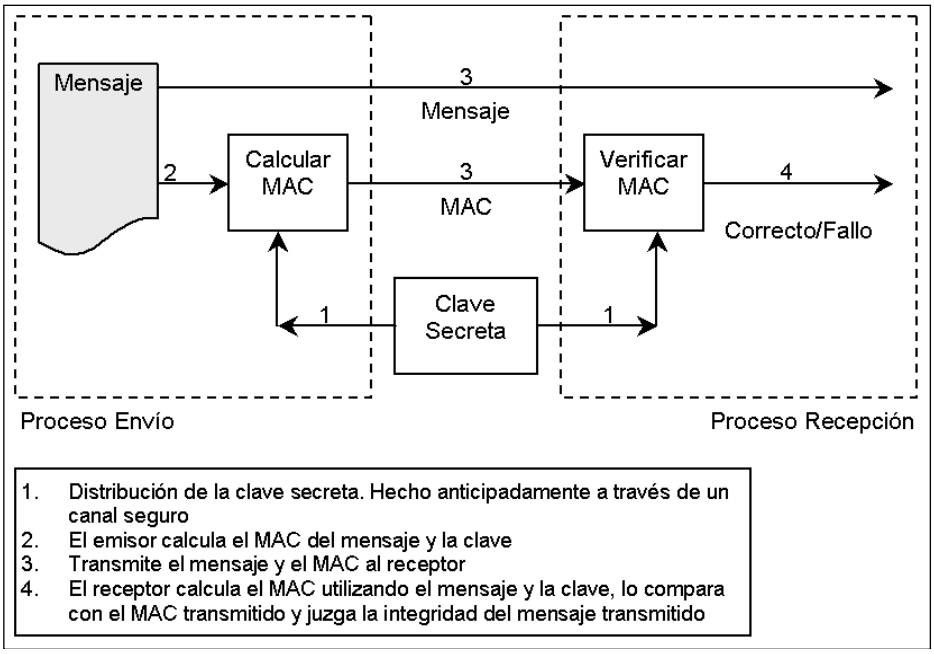


Figura 10: Autenticación mediante códigos MAC.

3.1 Funciones resumen (*hash functions*)

[7]El orden de ejecución de los algoritmos disponibles para resolver este problema es del orden de $O\left(e^{\sqrt{\ln(N)\ln[\ln(N)]}}\right)$, que en términos prácticos se traduce en que para factorizar un número N de 60 dígitos son necesarios $2{,}7*10^{11}$ pasos (3 días de cálculo suponiendo que cada paso tarda 1 μs en cada paso), $2{,}3*10^{15}$ pasos (74 años) para un número N de 100 dígitos o $1{,}2*10^{23}$ ($3{,}8*10^{9}$ años) para un número de 200 dígitos.

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

Una función resumen *H* se utiliza para generar un código *H*(*M*) que "resume" un determinado mensaje *M*. Ahora bien, el "resumen" generado no es cualquier valor sino que la función que lo genera debe cumplir una serie de propiedades que se enumeran a continuación:

- *Unidireccionalidad*. Conocido un resumen *H*(*M*), debe de ser computacionalmente imposible encontrar el mensaje original *M* a partir de dicho resumen.

- *Compresión*. A partir de un mensaje de cualquier longitud, el resumen *H*(*M*) debe tener una longitud fija. Lo normal es que la longitud de *H*(*M*) sea menor, típicamente de 128 ó 256 bits.

- *Facilidad de cálculo*. Debe ser fácil calcular *H*(*M*) a partir de un mensaje *M*.

- *Difusión*. El resumen *H*(*M*) debe ser una función compleja de todos los bits del mensaje *M*.

- *Colisión simple*. Conocido el mensaje *M*, será computacionalmente imposible encontrar un mensaje *M'* tal que coincidan sus resúmenes, es decir, tal que *H*(*M*) = *H*(*M'*). Si *H* cumple esta propiedad se dice que la función presenta resistencia débil frente a las colisiones.

- *Colisión fuerte*. Será computacionalmente difícil encontrar un par de mensajes *M* y *M'* tal que *H*(*M*) = *H*(*M'*). En este caso se dice que la función presenta una resistencia fuerte a las colisiones.

Existen multitud de algoritmos de generación de resúmenes: MD5, SHA-1, RIPEMD, Snefru, N-Hash, etc. Entre estos algoritmos, los más extendidos son el algoritmo MD5 (Rivest, 1992) y el algoritmo SHA-1 (NIST, 1993, 1994b). Brevemente, las operaciones del algoritmo MD5 pueden resumirse como sigue:

1. Se ajusta el tamaño de cualquier mensaje *M* de forma que su tamaño sea un múltiplo de 512 bits (para ello se añaden bits de relleno al final del mensaje si es necesario).
2. Con los 128 bits de cuatro vectores iniciales de 32 bits cada uno y el primer bloque del mensaje de 512 bits, se realizan diversas operaciones lógicas entre ambos.
3. La salida de esta operación (128 bits) se convierte en el nuevo conjunto de vectores y se realiza la misma función con el segundo bloque de 512 bits, y así, sucesivamente.
4. Al terminar, el algoritmo devuelve un resumen que se corresponde con los últimos 128 bits generados.

Las funciones que generan un resumen de 128 bits tienen una complejidad algorítmica de tan solo 2^{64} (un valor que en la actualizad puede verse comprometido). La función SHA-1 genera un resumen de 160 bits (y por tanto tiene una complejidad algorítmica del orden de 2^{80}, que hoy en día es segura). Básicamente, el algoritmo SHA-1 es similar al algoritmo MD5 con la diferencia de que genera resúmenes 160 bits de longitud.

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

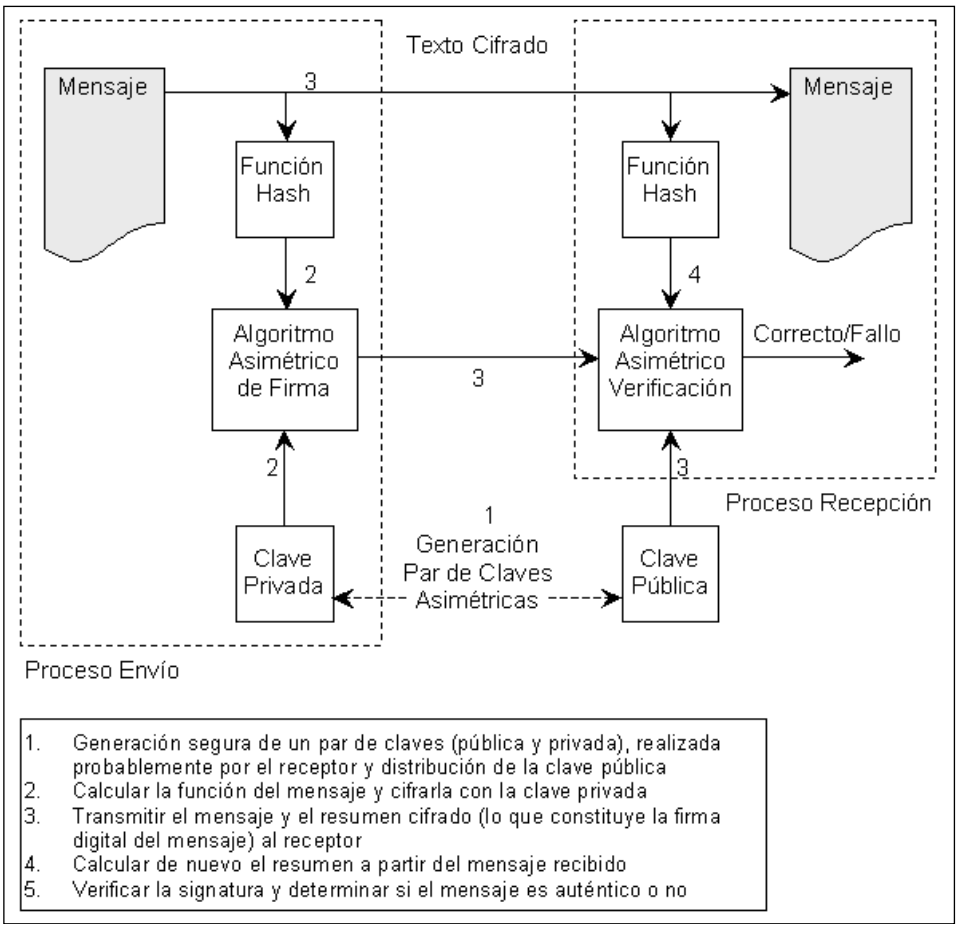


Figura 11: Autenticación mediante firmas digitales.

3.2 Autenticación y firma digital

Si bien en muchas aplicaciones es muy importante mantener la confidencialidad de la información (si ésta lo requiere), en otras muchas situaciones tiene quizás más trascendencia, el poder certificar la autenticidad entre un cliente y un servidor. Así pues, los problemas asociados a la integridad pueden enumerarse de la siguiente forma:

- *Autenticidad del emisor*, es decir, cómo comprueba el receptor (usuario *B*) que el mensaje recibido del emisor, que dice ser el usuario *A*, es efectivamente el usuario *A*.

- *Integridad del mensaje*, es decir, cómo comprueba el receptor que el mensaje recibido de *A* es el auténtico y no un mensaje falso.

- *Actualidad del mensaje*, es decir, cómo comprueba el receptor que el mensaje recibido de *A* no es un mensaje de fecha anterior y reenviado.

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

- *No repudio del emisor*, cómo demuestra el receptor *B* que el mensaje recibido ha sido enviado por *A*, cuando este niega haberlo enviado.

- *No repudio del receptor*, es decir, cómo comprueba *A* que el mensaje enviado al receptor *B*, efectivamente se envió cuando el receptor niega haberlo recibido.

- *Usurpación de la identidad del emisor o el receptor*, es decir, cómo comprueba el usuario *A* que cualquier otro usuario no están enviando mensajes firmados como él.

Para conseguir la autenticación existen diferentes funciones de autenticación que pueden clasificarse como sigue:

- *Autenticación mediante el cifrado de mensajes con criptografía simétrica*. En este caso, si la clave del sistema simétrico es segura (no está comprometida), se puede afirmar que, además de la confidencialidad del sistema, se obtienen también la integridad del mensaje y la autenticidad del emisor, en tanto que sólo el usuario emisor (en quien se confía por el modelo de cifra) puede generar ese mensaje. Ahora bien, en este caso se presentan los problemas característicos de los criptosistemas simétricos puesto que éstos no proporcionan, por sí mismos, un mecanismo para intercambiar las claves de forma segura (que es la base de la relación de confianza).

- *Autenticación mediante códigos MAC (Message Code Authentication)*. En el caso anterior, la autenticación descansa sobre el mensaje cifrado en sí mismo, siempre y cuando la clave compartida por ambas partes sea segura. En este caso, véase Figura 10, la solución es similar, pero en vez de descansar sobre el mensaje cifrado como tal, descansa sobre un código generado a partir del mensaje original y una clave compartida y secreta, es decir, el código *MAC* será la aplicación de una función *C* al mensaje *M* junto con la clave secreta *K*, es decir, $MAC = C_K(M)$. El emisor el mensaje en claro y el código MAC al receptor *B*, que puede comprobar la integridad del mensaje si el valor del código que calcula en destino a partir del mensaje recibido coincide con el calculado en origen. También puede comprobar la autenticidad del emisor por la misma razón que antes, la clave que genera el código es común, secreta y no comprometida.

- *Autenticación mediante el cifrado de mensajes con criptografía asimétrica*. La autenticación se consigue mediante lo que se denomina firma digital (véase Figura 11), que es un elemento que se anexa a los mensajes y que se debe caracterizar por ser fácil de generar, irrevocable (no rechazable por su propietario), única (sólo la puede haber generado su propietario), fácil de autenticar o reconocer (por su propietario o por los usuarios receptores) y depender del mensaje y del autor. En los sistemas asimétricos la firma digital se obtiene cifrando (con la clave privada del emisor) un resumen del mensaje enviado, habiendo obtenido éste mediante la aplicación de una función *hash*. El resultado de esta operación se adjunta al mensaje enviado y se proporciona así al receptor un elemento que puede cotejar (ya que puede descifrar y obtener el resumen calculado en origen) para autenticar completamente el mensaje. Los sistemas de firma digital basados en criptografía asimétrica son los más ampliamente extendidos y destacan la firma RSA y la firma DSA (NIST, 1992, 1994a).

3.3 Certificados digitales y autoridades de certificación

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242

Conviene hacer notar que, aunque una firma digital válida garantiza que, necesariamente, el remitente de un mensaje es el poseedor de una clave pública, ésta, por sí sola, no es suficiente para garantizar la identidad del remitente. Esto es así dado que, en ningún momento se puede constatar que el poseedor de una clave pública concreta sea quien dice ser. Este hecho hace que cualquier sistema de cifrado asimétrico sea susceptible de sufrir un ataque de intermediario. Suponiendo que *A* es el emisor de un mensaje, que *B* es el receptor y que *C* quiere espiar la comunicación, el ataque de intermediario consiste en que cuando *A* solicite a *B* su clave pública, *C* se interpone, obteniendo la clave de *B* y enviando a *A* una clave falsa creada por él. Cuando *A* codifique el mensaje, *C* lo interceptará de nuevo, decodificándolo con su clave propia y empleará la clave pública de *B* para cifrarlo y enviárselo de nuevo a *B*. De esta forma, ni *A* ni *B* serán conscientes de que sus mensajes han sido interceptados por *C*.

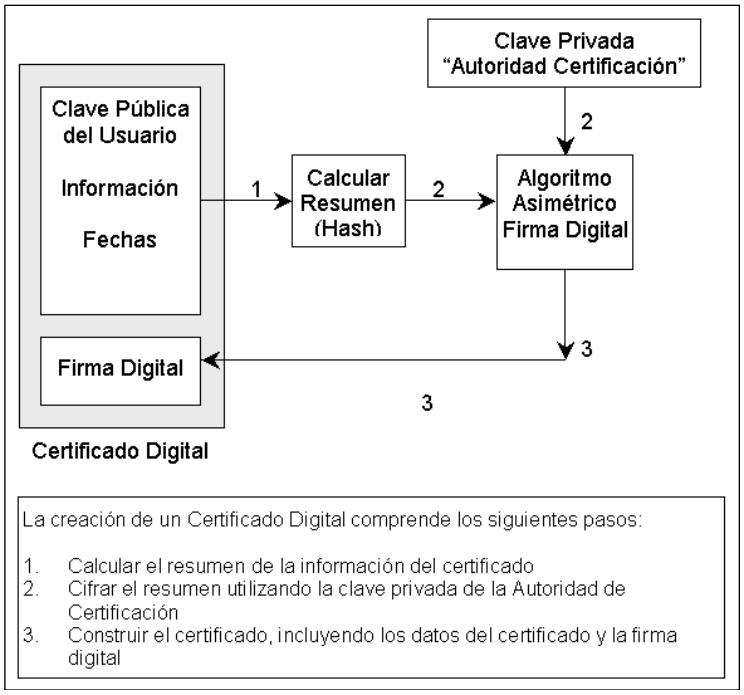


Figura 12: Proceso de generación de un Certificado Digital.

Para solucionar este problema se requiere la utilización de un certificado digital (*digital certificate*), o documento electrónico emitido y firmado digitalmente por una tercera parte de confianza (*trusted third party*), en el que se hace constar que una clave pública específica pertenece realmente a una persona concreta (véase Figura 12). Así pues, un certificado digital no es más que un documento que liga una clave pública con la identidad de un usuario (que se entiende su legítimo propietario) y está firmado digitalmente por una tercera entidad en la que ambas partes (emisor y receptor) confían. A partir de la noción de certificado digital y del modelo de confianza basado en terceras partes, se define lo que se conoce como infraestructura de clave pública (*PKI, Public Key Infrastructure*). Sucintamente, una PKI es un conjunto de protocolos, servicios y estándares que soportan aplicaciones basadas en criptografía de clave pública. Los servicios que ofrece la infraestructura PKI son:

- *Registro de claves* o emisión de un nuevo certificado para una clave pública.

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

- *Revocación de certificados*, es decir, la cancelación de un certificado previamente emitido.
- *Selección de claves* o publicación de la clave pública de los usuarios.
- *Evaluación de la confianza*, que se refiere a la determinación sobre si un certificado es válido y qué operaciones están permitidas para dicho certificado.
- *Recuperación de claves* o posibilidad por parte de un usuario de recuperar una clave.

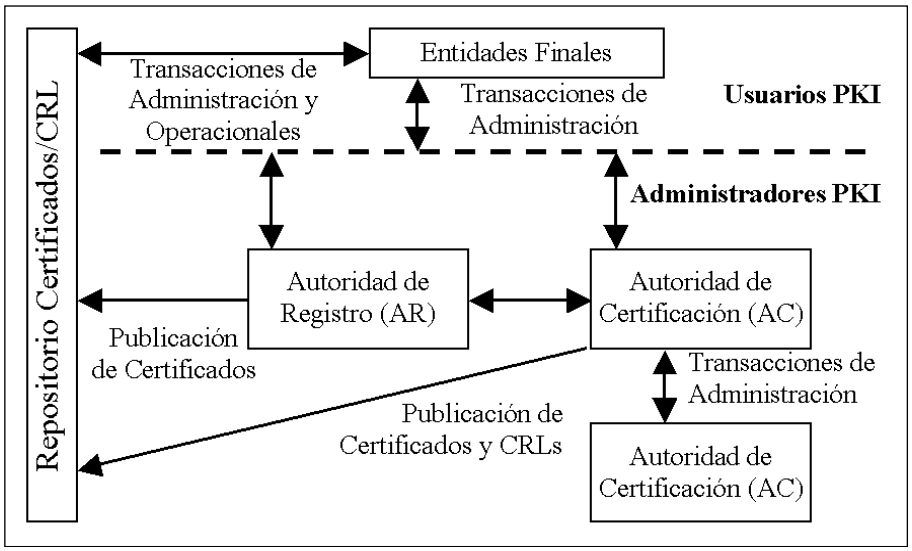


Figura 13: Esquema genérico de una PKI.

En cuanto a la organización de la infraestructura PKI, hay que decir que no es una entidad única y monolítica, sino que es un sistema distribuido, de modo que diferentes PKI interconectadas puedan ínter operar. Ahora bien, los componentes de una infraestructura PKI particular son: un conjunto de entidades finales (bien sean usuarios PKI o usuarios identificados), una autoridad de certificación (AC), una autoridad de registro (AR), otras autoridades de certificación (por ejemplo, autoridades de sellado digital de tiempo) y un repositorio de claves, certificados, listas de revocación de certificados (*CRL, Certification Revocation Lists*). Estos componentes se muestran en la Figura 13.

Por lo que se refiere a las funciones de las autoridades de certificación y registro, éstas comprenden tareas como la identificación de los solicitantes de certificados (que puede delegarse en las AR), la generación y registro de claves, la emisión de certificados, la custodia indiscutible de la clave privada de la autoridad de certificación (fundamental, ya que su compromiso invalidaría el sistema por completo: todos los certificados están firmados con esta clave), el mantenimiento (actualizado y veraz) de las claves vigentes y revocadas (CRLs) y el ofrecimiento de un servicio de directorio [61-65].

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242

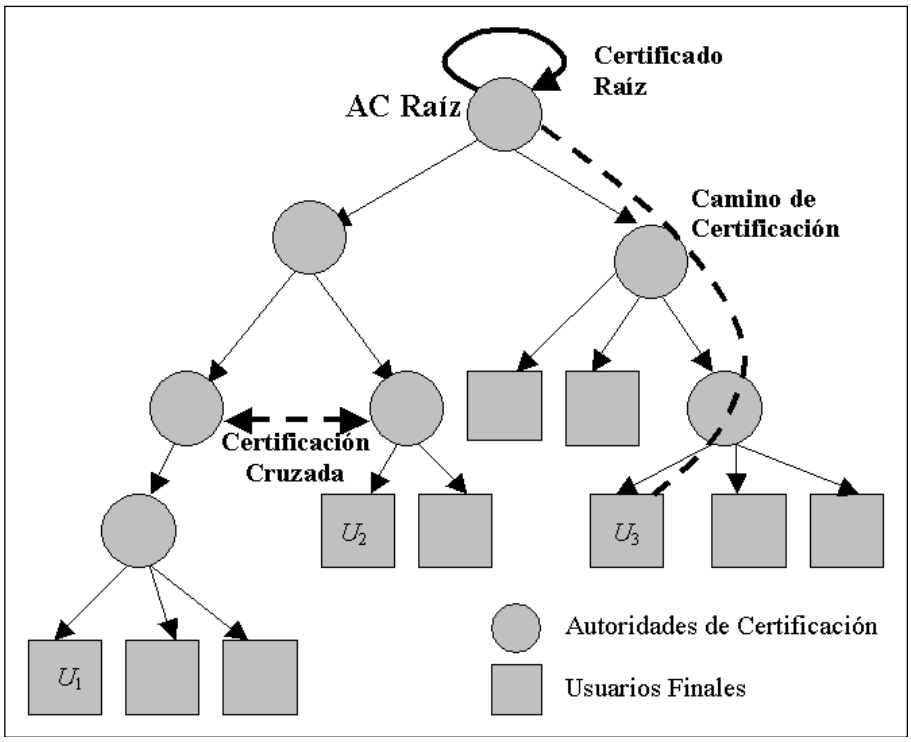


Figura 14: Jerarquía de confianza en una PKI.

La fortaleza del modelo se basa en que la asociación entre la identidad y la clave pública de un usuario está avalada por una autoridad de certificación mediante su firma. Para garantizar la autenticidad de la clave pública de la autoridad, puede requerirse un certificado digital de ésta avalado por otra autoridad de certificación de nivel superior. De esta forma, el sistema se estructura en forma de árbol que representa una jerarquía de confianza, tal y como se representa en la Figura 14. En la misma reflejan también nociones como:

- *Certificado raíz*. Consiste en un certificado para la autoridad de certificación emitida por la propia autoridad de certificación. Son certificados hechos públicos de modo fiable con la finalidad de que los vendedores de software puedan incluirlos en su software.
- *Camino de certificación* (*trust chain*). Para cada usuario final existe un camino que termina en la raíz.
- *Certificación cruzada*. Consiste en la certificación mutua entre autoridades de certificación mediante comunicaciones frecuentes.

Generalmente, la organización en forma de árbol es habitual en las infraestructuras PKI cuyo ámbito es nacional. En un sistema de ámbito multinacional, la jerarquía de confianza se convierte en un grafo con certificación cruzada entre las autoridades de certificación raíz. Las autoridades de certificación se caracterizan porque su funcionamiento debe seguir una política de certificación convenientemente establecida. La política en sí, no es más que el conjunto de criterios que regulan los servicios de certificación relativos a la solicitud de un certificado, la validación de la solicitud, la emisión del certificado, su aceptación y uso, la suspensión, revocación y renovación y la recuperación de claves. Se supone que el usuario es conocedor de la tecnología PKI y que acepta los certificados antes de utilizarlos. Además, la responsabilidad de confiar en un certificado y comprobar si es válido, recae sobre el receptor del certificado. En caso de pérdida o compromiso de

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

la clave privada, el usuario se compromete a notificarlo inmediatamente a la autoridad de certificación.
Finalmente, cabe destacar que los certificados digitales más extendidos siguen la norma X.509. El documento RFC2459 especifica el formato y la semántica de los certificados X.509 v3 y de las listas de revocación de certificados (CRLs) X.509 v2. El documento es parte de la familia de estándares para la infraestructura de clave pública (PKI) X.509 sobre Internet. Los certificados X.509 v3 se estructuran de la siguiente forma:

- *Datos del certificado*. Son los datos del certificado en sí mismo e incluyen:
  - *Versión*. Versión del estándar X.509 empleado (1, 2 ó 3).
  - *Número de serie*. Número único asignado por la AC a cada certificado que emite (Se utilizará en las CRLs).
  - *Algoritmo de firma*. Identifica convenientemente (mediante identificadores OID de la notación ASN.1) el algoritmo utilizado para firmar el certificado (DSA, SHA-1).
  - *Emisor*. Nombre (en forma estructurada) de la autoridad de certificación que emite el certificado.
  - *Validez*. Período de validez en el que la autoridad de certificación garantiza que mantendrá información acerca del estado del certificado.
  - *Asunto*. Nombre (en forma estructurada) de la entidad asociada con la clave pública almacenada en el correspondiente campo.
  - *Clave pública*. Información sobre el algoritmo y la clave pública empleada por la entidad identificada.
  - *Identificadores únicos*. Incluidos por si fuera necesario en un futuro reutilizar los nombres correspondientes.
  - *Extensiones*. Proporciona un medio para asociar atributos adicionales a los usuarios o las claves públicas, así como la gestión de la jerarquía de certificación.
- *Algoritmo de firma*. Coincide con lo descrito con anterioridad.
- *Firma*. Valor de la firma digital de la autoridad de certificación generada a partir del los datos del certificado y gracias a la cual, la autoridad certifica la validez de dichos datos.

En lo que se refiere a las listas de revocación de certificados su estructura es la siguiente:

- Datos CRL X.509 v2
  - *Versión*. Campo opcional.
  - *Algoritmo de firma*. Identifica convenientemente (OIDs ASN.1) el algoritmo utilizado para firmar la lista de certificados (DSA, SHA-1).
  - *Emisor*. Nombre (en forma estructurada) de la autoridad de certificación que firma y emite la lista de certificados.
  - *Fecha de la presente actualización*. Indica la fecha de emisión de esta lista de certificados.
  - *Fecha de la siguiente actualización*. Indica la fecha en la que, como muy tarde, se emitirá la siguiente lista de certificados.
  - *Certificados anulados*. Lista con los certificados anulados. Cada certificado se identifica por el número de serie que asignó la autoridad de certificación al crearlo y la fecha de revocación. Opcionalmente pueden tener asociada información adicional (razón de por qué aparece en la lista) mediante una lista de extensiones para la entrada correspondiente.

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242

  - *Extensiones CRL*. Las extensiones CRL proporcionan un medio para asociar atributos adicionales a la lista de certificados: información necesaria para soportar grupos bien diferenciados, identificación de la clave pública que se corresponde con la clave privada empleada por la autoridad de certificación para firmar la lista (información útil cuando el emisor maneja más de un par de claves).
- *Algoritmo de firma*. Coincide con el descrito anteriormente.
- *Firma*. Valor de la firma digital de la AC generada a partir del los datos de la CRL y gracias a la cual la autoridad certifica la validez de dichos datos.

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

References

1. Biham, E., y Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. Advances in Cryptology – Crypto'90, (págs. 2–21). Springer–Verlag.
2. Biham, E., y Shamir, A. (1993). Differential Cryptanalysis of the Data Encryption Standard. Springer–Verlag.
3. Consejo Superior de Informática, CSI. (Ministerio Administraciones Públicas, MAP). (1996). Seguridad en redes telemáticas, correo electrónico y servicios Internet.
4. Daemen, J., y Rijmen, V. (2000). The Block Cipher Rijndael. En J.-J. Quisquater, J. J., y Schneier, B. (Eds.). Smart Card Research and Applications, LNCS 1820, (págs. 288–296). Springer–Verlag.
5. Diffie, W., y Hellman, M. E. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22: 644–654.
6. Feistel, H. (1973). Cryptography and Computer Privacy. Scientific American (May, 1973).
7. Lucena, M. J. (2002). Criptografía y Seguridad en Computadores. Tercera Edición. Disponible en http://wwwdi.ujaen.es/~mlucena/
8. Lai, X., and Massey, J. L. (1991). A proposal for a new block encryption standard. Advances in Cryptology – Eurocrypt'90 (págs. 389–404). Springer–Verlag.
9. Matsui, M. (1993). Linear cryptanalysis meted for DES cipher. Advances in Cryptology – Eurocrypt'93. (págs. 386–397).
10. National Institute of Standards and Technology, NIST. (1980). FIPS Publication 81: DES Modes of Operation.
11. National Institute of Standards and Technology, NIST (1992). The Digital Signature Standard, proposal and discussion, Communications of the ACM (7), 35:36–54.
12. National Institute of Standards and Technology, NIST (1993). FIPS Publication 180: Secure Hash Standard (SHS).
13. National Institute of Standards and Technology, NIST (1994a). FIPS Publication 186: Digital Signature Standard (DSS).
14. National Institute of Standards and Technology, NIST (1994b). Announcement of Weakness in the Secure Hash Standard.
15. National Institute of Standards and Technology, NIST (2001). FIPS Publication 197: Advanced Encryption Standard (AES).
16. Schneier, B. (1996) Applied Cryptography. Second Edition. John Wiley & Sons.
17. Rivest, R. L., Shamir, A., y Adleman, L. M. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM (2), 21: 120–126.
18. Rivest, R.L. (1992). RFC 1321: The MD5 Message-Digest Algorithm, Internet Activities Board.
19. Shannon, C. E. (1949). Communication Theory of Secrecy Systems. Bell Systems Technical Journal, 28:656–715.
20. Ana Karin Chávez Valdivia (2017). Between the Profiles Pay Per View and the Protection of Personal Data: the Product is You. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 1
21. Ana Oliveira Alves, Tiago Dias, David Silva (2015). A Real-Time, Distributed and Context-Aware System for Managing Solidarity Campaigns. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 2
22. Angelo Costa, Stella Heras, Javier Palanca, Paulo Novais, Vicente Julián (2016). Persuasion and Recommendation System Applied to a Cognitive Assistant. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
23. Canizes, B., Pinto, T., Soares, J., Vale, Z., Chamoso, P., & Santos, D. (2017). Smart City: A GECAD-BISITE Energy Management Case Study. In 15th International Conference on Practical Applications of Agents and Multi-Agent Systems PAAMS 2017, Trends in Cyber-Physical Multi-Agent Systems (Vol. 2, pp. 92–100). https://doi.org/10.1007/978-3-319-61578-3_9
24. Carlos Alberto Ochoa, Lourdes Yolanda Margain, Francisco Javier Ornelas, Sandra Guadalupe Jiménez, Teresa Guadalupe Padilla (2014). Using multi-objective optimization to design parameters in electro-discharge machining by wire. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2
25. Casado-Vara, R., & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386.
26. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
27. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242

28. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
29. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
30. Céline Ehrwein Nihan (2013). Healthier? More Efficient? Fairer? An Overview of the Main Ethical Issues Raised by the Use of Ubicomp in the Workplace. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
31. Chamoso, P., de La Prieta, F., Eibenstein, A., Santos-Santos, D., Tizio, A., & Vittorini, P. (2017). A device supporting the self-management of tinnitus. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 10209 LNCS, pp. 399–410). https://doi.org/10.1007/978-3-319-56154-7_36
32. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
33. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
34. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
35. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
36. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
37. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In Advances in Case-Based Reasoning, Proceedings (Vol. 3155, pp. 533–546).
38. Corchado, J. M., & Aiken, J. (2002). Hybrid artificial intelligence methods in oceanographic forecast models. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 32(4), 307–313. https://doi.org/10.1109/tsmcc.2002.806072
39. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In Industrial Conference on Data Mining (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1
40. David Griol, Jose M. Molina (2016). A proposal to manage multi-task dialogs in conversational interfaces. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
41. David Griol, Jose Manuel Molina (2016). From VoiceXML to multimodal mobile Apps: development of practical conversational interfaces. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
42. Davide Carneiro, Daniel Araújo, André Pimenta, Paulo Novais (2016). Real Time Analytics for Characterizing the Computer User's State. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
43. Eduardo Facchini, Eduardo Mario Dias, Alexandre Pelegi Abreu, Maria Lídia Rebello Pinho Dias (2016). Brazil in Search of Transparency E-Gov. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
44. Eduardo Mario Dias, Eduardo Facchini, Antônio Carlos De Moraes, Mauricio Lima Ferreira, Willian Reginato Este, Maria Lídia Rebello, Pinho Dias (2014). A Future Look. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 3
45. Elton S Siqueira, Patrick Cisuaka Kabongo, Tiancheng Li, Carla D. Castanho, Li Weigang (2016). On Chinese and Western Family Trees: Mechanism and Performance. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
46. Ester Martinez-Martin, Maria Teresa Escrig, Angel P. Del POBIL (2013). A Qualitative Acceleration Model Based on Intervals. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
47. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. Advanced Engineering Informatics, 16(3), 165–178. https://doi.org/10.1016/S1474-0346(02)00008-3

ISBN: 978-84-9012-862-6
Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc

48. Fyfe, C., & Corchado, J. M. (2001). Automating the construction of CBR systems using kernel methods. International Journal of Intelligent Systems, 16(4), 571–586. https://doi.org/10.1002/int.1024
49. García, O., Chamoso, P., Prieto, J., Rodríguez, S., & De La Prieta, F. (2017). A serious game to reduce consumption in smart buildings. In Communications in Computer and Information Science (Vol. 722, pp. 481–493). https://doi.org/10.1007/978-3-319-60285-1_41
50. Giovani Parente Farias, Ramon Fraga Pereira, Lucas W. Hilgert, Felipe Meneguzzi, Renata Vieira, Rafael H. Bordini (2017). Predicting Plan Failure by Monitoring Action Sequences and Duration. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
51. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
52. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
53. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
54. Jean Louis Monino, Soraya Sedkaoui (2016). The Algorithm of the Snail: An Example to Grasp the Window of Opportunity to Boost Big Data. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
55. Johannes Fähndrich, Sebastian Ahrndt, Sahin Albayrak (2014). Formal Language Decomposition into Semantic Primes. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
56. Karel Macek, Jiri Rojicek, Georgios Kontes, Dimitrios V. Rovas (2013). Black-Box Optimization for Buildings and Its Enhancement by Advanced Communication Infrastructure. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
57. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
58. M.ª Belén Aige (2017). The online tourist fraud: the new measures of technological investigation in Spain. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
59. Marco Antonio Ameller, María Angélica González (2016). Minutiae filtering using ridge-valley method. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
60. Merce Teixido, Tomás Palleja, Marcel Tresanchez, Davinia Font, Javier Moreno, Alicia Fernández, Jordi Palacín, Carlos Rebate (2013). Optimization of the virtual mouse HeadMouse to foster its classroom use by children with physical disabilities. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
61. Muhammad Amin Khan, Felix Freitag (2014). Sparks in the Fog: Social and Economic Mechanisms as Enablers for Community Network Clouds. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
62. Palomino, C. G., Nunes, C. S., Silveira, R. A., González, S. R., & Nakayama, M. K. (2017). Adaptive agent-based environment model to enable the teacher to create an adaptive class. Advances in Intelligent Systems and Computing (Vol. 617). https://doi.org/10.1007/978-3-319-60819-8_3
63. Pawel Pawlewski, Kamila Kluska (2017). Modeling and simulation of bus assembling process using DES/ABS approach. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 1
64. Rodriguez-Fernandez J., Pinto T., Silva F., Praça I., Vale Z., Corchado J.M. (2018) Reputation Computational Model to Support Electricity Market Players Energy Contracts Negotiation. In: Bajo J. et al. (eds) Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection. PAAMS 2018. Communications in Computer and Information Science, vol 887. Springer, Cham
65. Roussanka Loukanova (2016). Relationships between Specified and Underspecified Quantification by the Theory of Acyclic Recursion. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4

ISBN: 978-84-9012-862-6

Colección Aquilafuente: AQ-242
Ediciones Universidad de Salamanca - cc by nc dc