Seguridad en Sistemas de Información

Curso en Zacatenco y Tamaulipas [Q2 2013]

Luis J. Dominguez Perez

Cinvestav, Mayo 13 de 2013

Contenido, sección 1

Introducción

Cuando escuchamos la palabra criptografía, las primeras cosas que podrían venirse a nuestra mente podrían ser:

- cifrado de correo
- acceso seguro a sitios web
- smart cards para aplicaciones bancarias
- el rompimiento de códigos durante la segunda guerra mundial, como el ataque a la máquina Enigma.

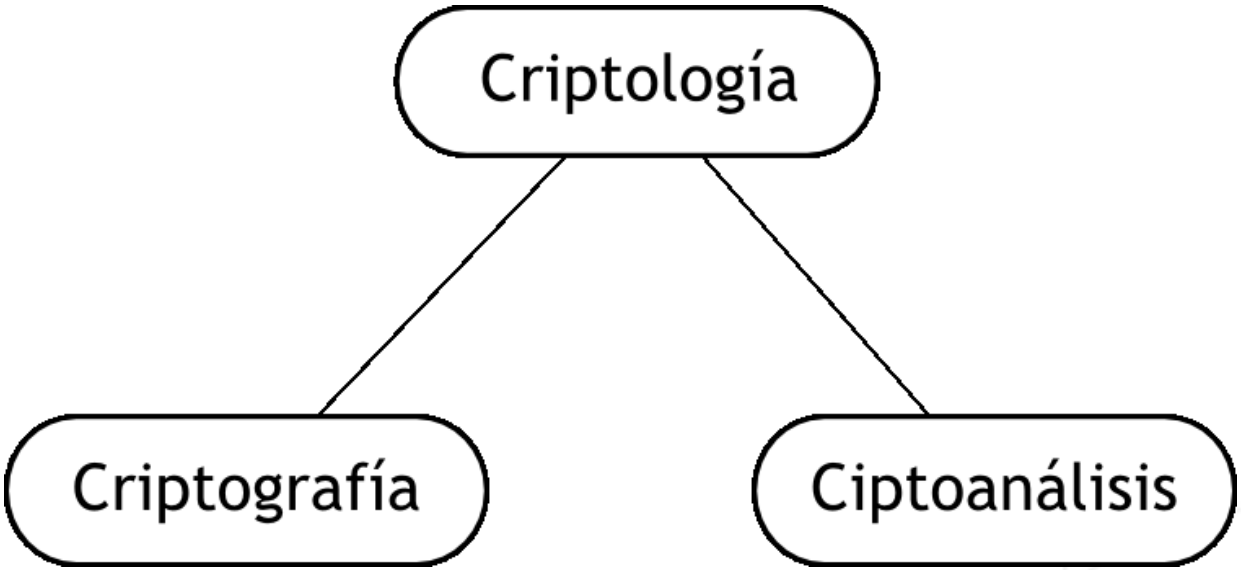

Q W E R T Z U I O
A S D F G H J K
P Y X C V B N M L

Inicios

La criptografía parece estar relacionada a la comunicación electrónica moderna. Sin embargo, la criptografía es un negocio viejo, sus primeros usos datan del año 2000 A.C. en el antiguo Egipto.

En la era moderna, hacemos uso intensivo de la computación. La criptografía es una aplicación de la computación muy importante, de hecho, la criptografía ¡es la primer aplicación de las computadoras! (junto con la balística).

Criptología

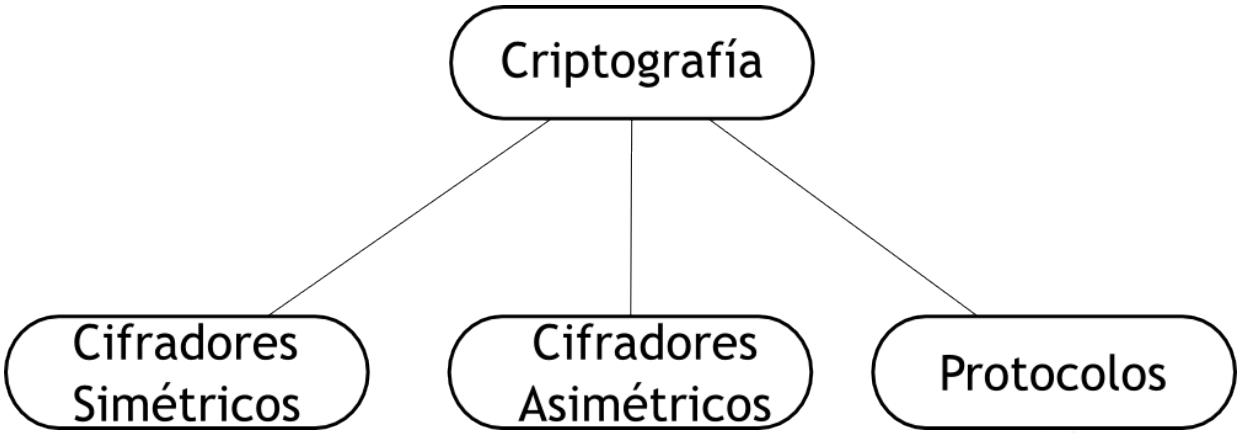


Criptografía y criptoanálisis

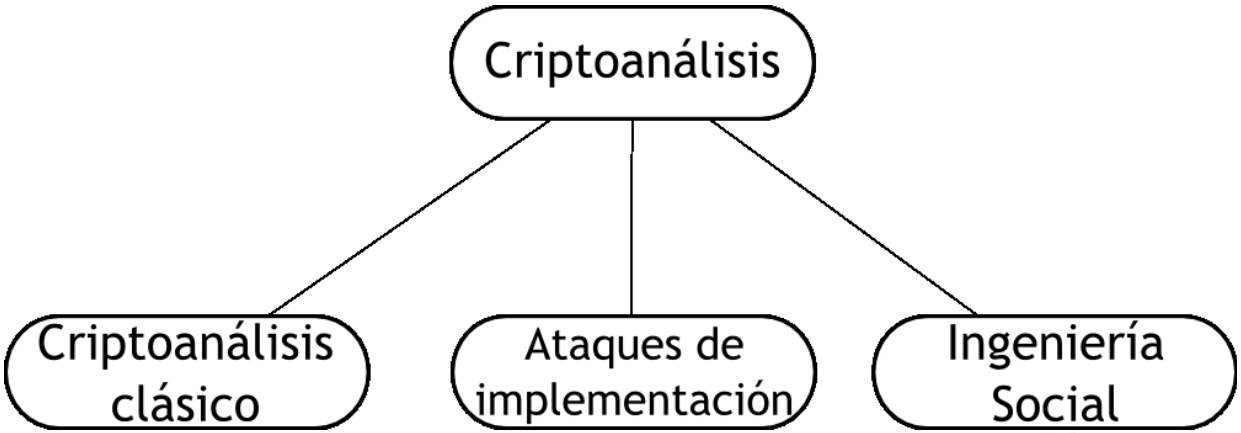
La criptología se divide en criptografía y criptología:

- **Criptografía.** Es la ciencia de escribir secretamente con la finalidad de ocultar el significado de un mensaje.
- **Criptoanálisis.** Es la ciencia y algunas veces el arte de romper criptosistemas.

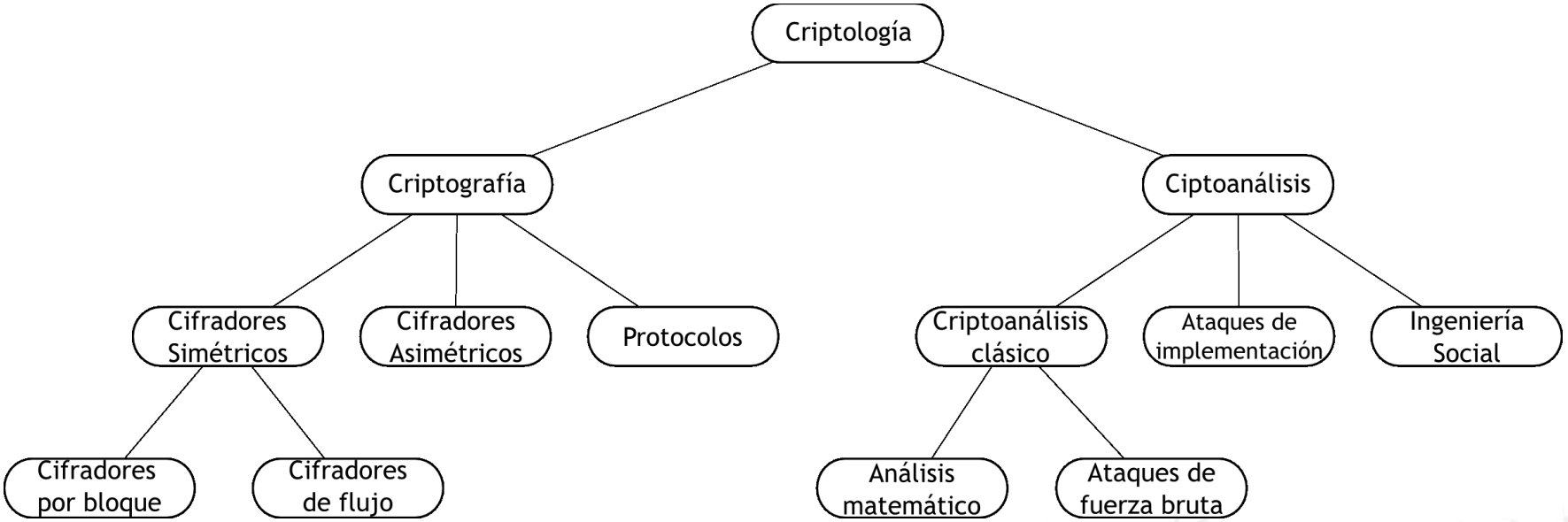
Criptografía



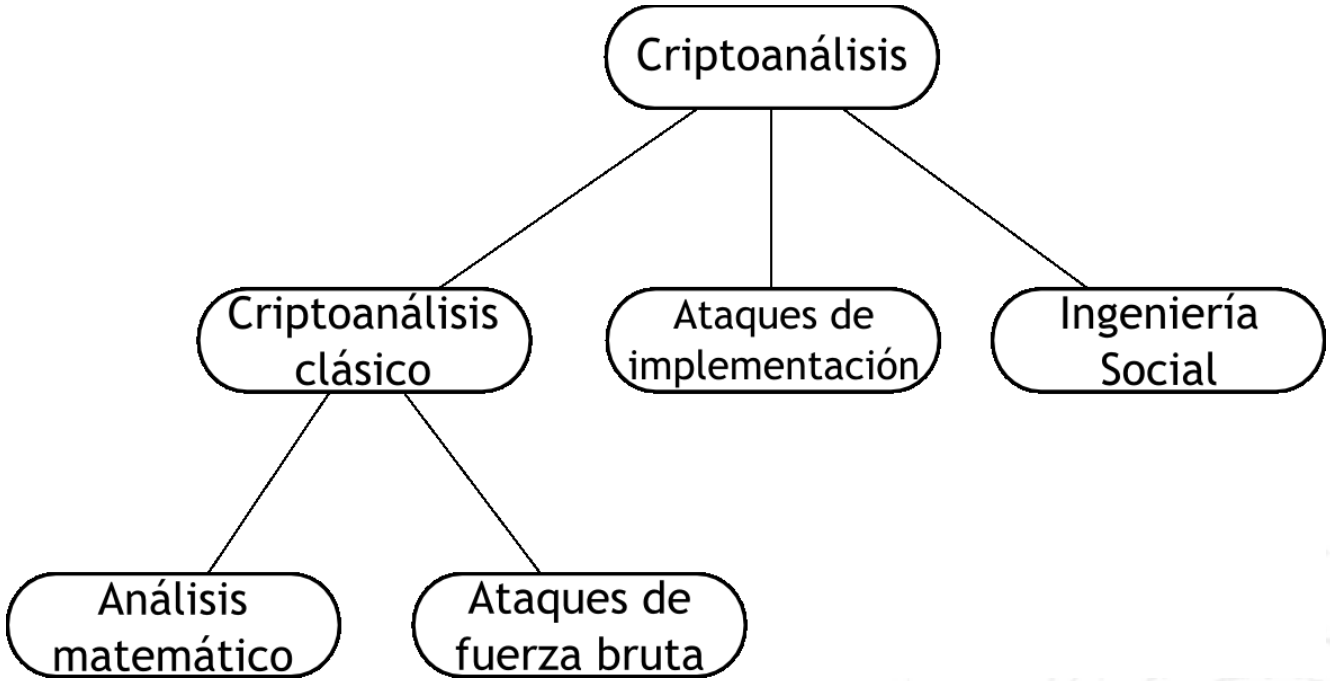
Criptoanálisis



Criptología



Áreas del criptoanálisis

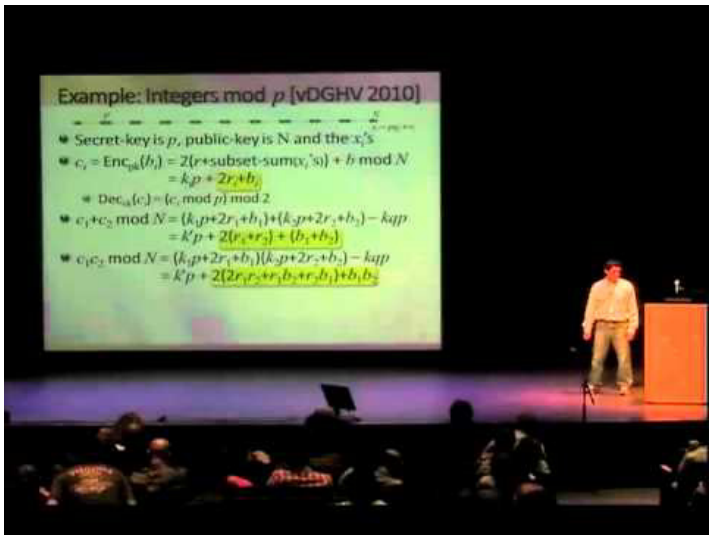


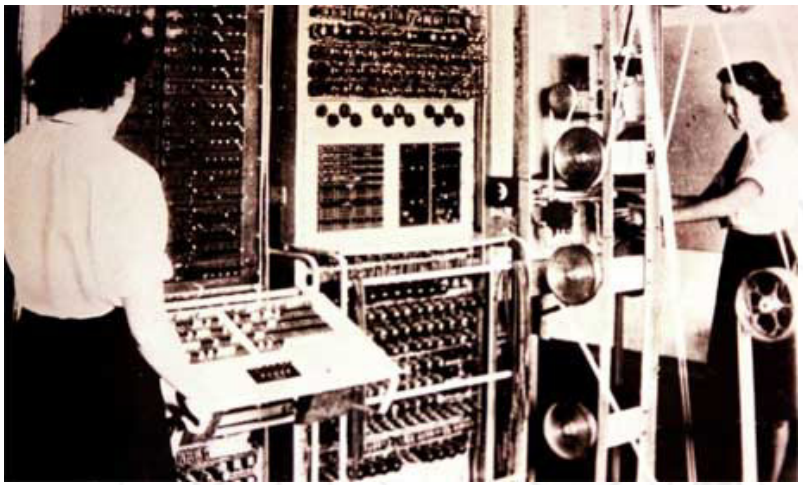
Criptoanálisis

Uno podría pensar que romper códigos es para la comunidad de servicios de inteligencia, o quizá de organizaciones criminales, y no debería de incluirse en una clasificación seria de las disciplinas científicas.

Criptoanálisis

Sin embargo, la mayoría de los criptoanálisis son realizados por investigadores respetables en la academia. El criptoanálisis es un área de importancia vital para los criptosistemas modernos: sin gente que intente romper los métodos de cifrado, nunca sabríamos si un método es realmente seguro o no.





Paradoja del cumpleaños

La probabilidad de que yo tenga el mismo cumpleaños que alguno de ustedes es:

$$1 - \left(\frac{365 - 1}{365}\right)^n .$$

[1] uniformemente distribuida

Paradoja del cumpleaños

La probabilidad de que yo tenga el mismo cumpleaños que alguno de ustedes es:

$$1 - \left(\frac{365 - 1}{365}\right)^n.$$

¿Qué pasa si se replantea el problema a: la probabilidad de que cualquiera de nosotros tenga el mismo cumpleaños que otro?

Si se tienen 367 personas, la probabilidad es del 100%[1], sin embargo, con 57 aún se tiene un 99% de probabilidad.

[1] uniformemente distribuida

Paradoja del cumpleaños

La probabilidad de que yo tenga el mismo cumpleaños que alguno de ustedes es:

$$1 - \left(\frac{365-1}{365}\right)^n.$$

¿Qué pasa si se replantea el problema a: la probabilidad de que cualquiera de nosotros tenga el mismo cumpleaños que otro?

Si se tienen 367 personas, la probabilidad es del 100%[1], sin embargo, con 57 aún se tiene un 99% de probabilidad....y con 23 se tiene cerca del 50%

[1] uniformemente distribuida

Paradoja del cumpleaños

La probabilidad de que yo tenga el mismo cumpleaños que alguno de ustedes es:

$$1 - \left(\frac{365-1}{365}\right)^n.$$

¿Qué pasa si se replantea el problema a: la probabilidad de que cualquiera de nosotros tenga el mismo cumpleaños que otro?

Si se tienen 367 personas, la probabilidad es del 100%[1], sin embargo, con 57 aún se tiene un 99% de probabilidad....y con 23 se tiene cerca del 50% de ahí lo de paradoja.

[1] uniformemente distribuida

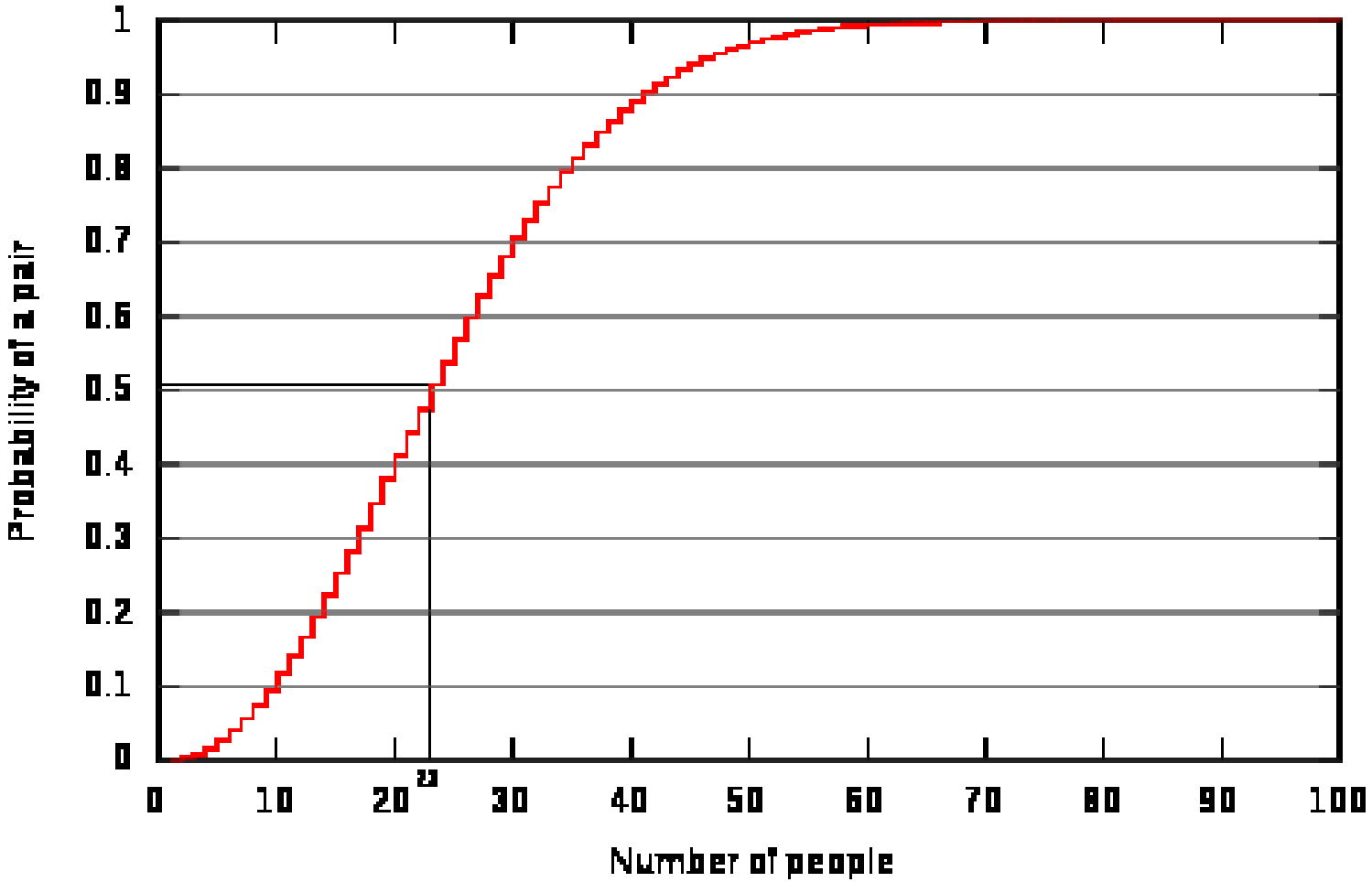
Probabilidad

La probabilidad de que no ocurra la colisión, depende en el número de personas. La fórmula viene dada por:

$$\begin{aligned}\bar{p}(n) &= 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \cdots \times \left(1 - \frac{n-1}{365}\right) \\ &= \frac{365 \times 364 \times \cdots \times (365 - n + 1)}{365^n} \\ &= \frac{365!}{365^n(365-n)!} = \frac{n!\binom{365}{n}}{365^n} = \frac{{}^{365}P_n}{365^n}\end{aligned}$$

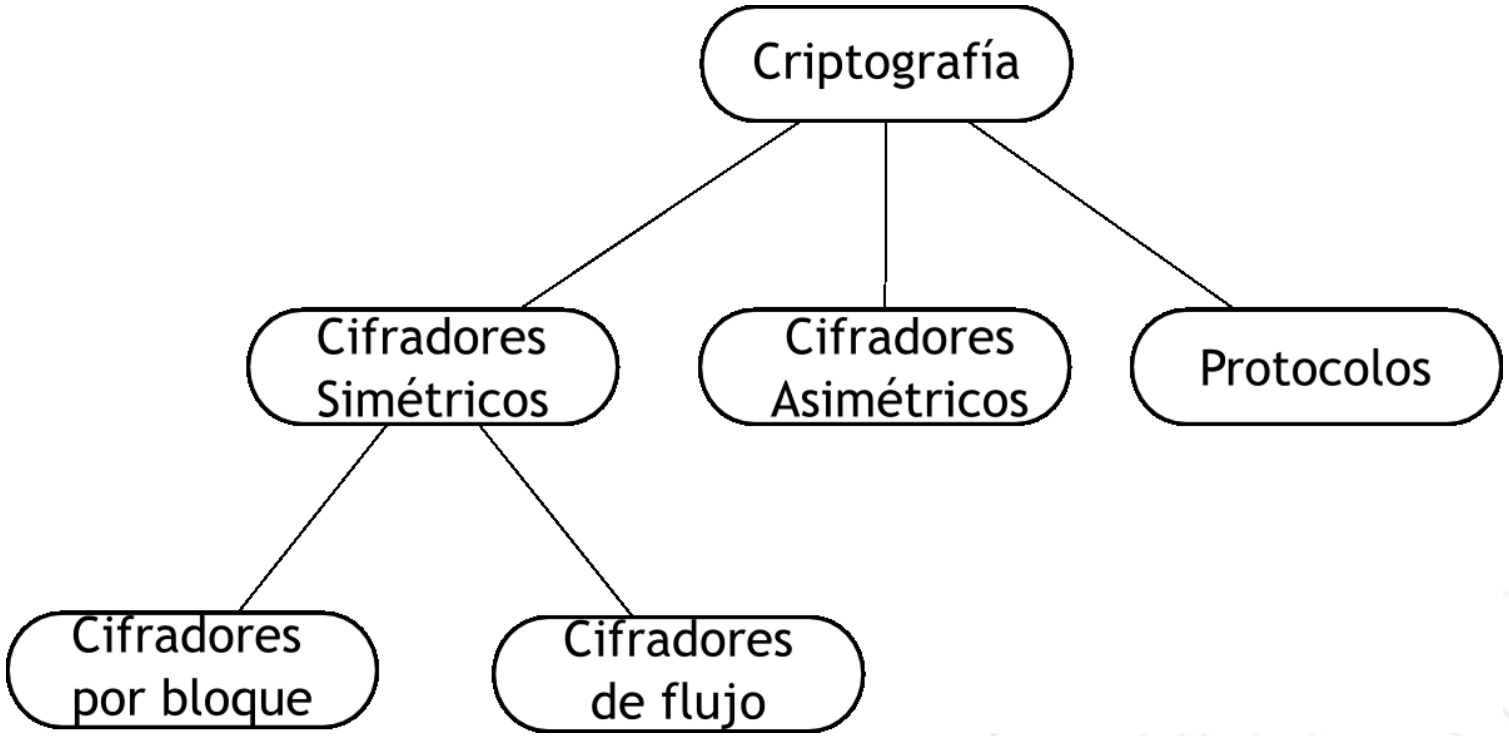
Entonces, la probabilidad de una colisión es: $p(n) = 1 - \bar{p}$.

Valores



Ejercicio: encontrar una colisión entre los participantes, y 2 de sus parientes.

Áreas de la criptografía



Criptografía, 1/3

La criptografía se divide en 3 partes principales:

- **Algoritmos simétricos.** Son lo que la mayoría de la gente piensa de la criptografía: dos partes tienen un método de cifrado y descifrado, y comparten una clave secreta.

La criptografía desde tiempos ancestrales hasta 1976 era exclusivamente simétrica. La criptografía simétrica sigue siendo fundamental hoy en día.

Criptografía, 2/3

- **Algoritmos asimétricos.** En 1976, un tipo diferente de cifrado fue introducido por Whitfield Diffie, Martin Hellman, y Ralph Merkle (aunque este último es generalmente menos reconocido).

  En la criptografía de clave pública, un usuario posee una clave secreta como en los algoritmos simétricos, pero también una clave pública.

  Podemos utilizar este tipo de criptografía para firmas digitales y el establecimiento de llaves, así como para el cifrado de datos tradicional.

Criptografía 3/3

- **Protocolos.** Hablando en términos concretos, los protocolos criptográficos es la aplicación de los algoritmos criptográficos.

Los algoritmos simétricos y asimétricos se pueden ver como los bloques de construcción, o cajas negras con las que las aplicaciones como la comunicación segura en internet se realiza.

Criptografía 3b/3

El esquema de Seguridad de Capa de Transporte (TLS), el cual es utilizado por todos los navegadores Web, es un ejemplo de un protocolo criptográfico.

Criptografía 3b/3

El esquema de Seguridad de Capa de Transporte (TLS), el cual es utilizado por todos los navegadores Web, es un ejemplo de un protocolo criptográfico.

Hablaremos de este protocolo más adelante.

Sistemas híbridos

En la práctica, la mayoría de las aplicaciones criptográficas utilizan tanto algoritmos simétricos como asimétricos (así como funciones picadillo). Esto es conocido, en algunas ocasiones, como esquemas híbridos.

La razón para utilizar ambos esquemas, es que cada uno tiene sus fortalezas y debilidades...

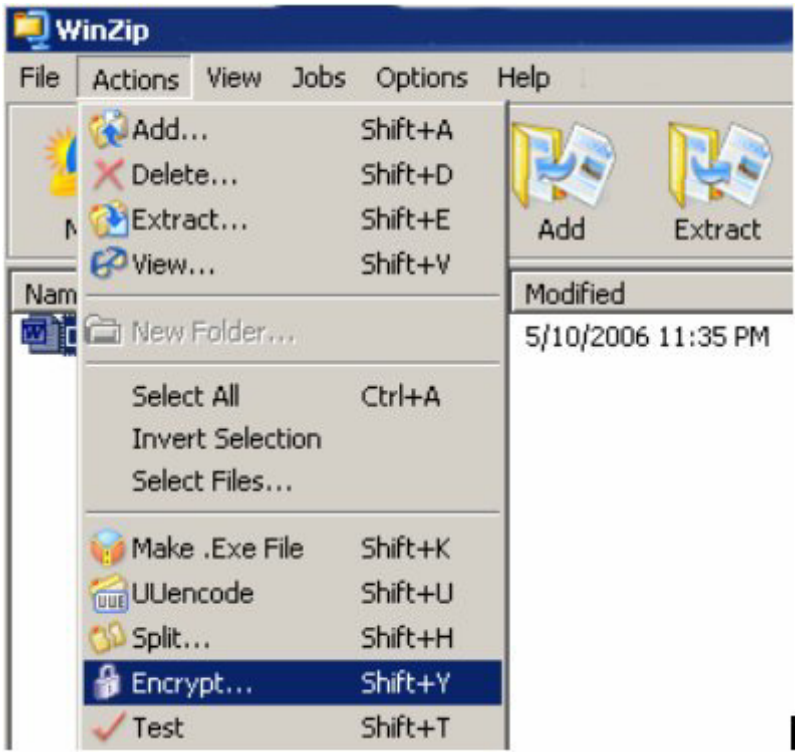
Sistemas híbridos

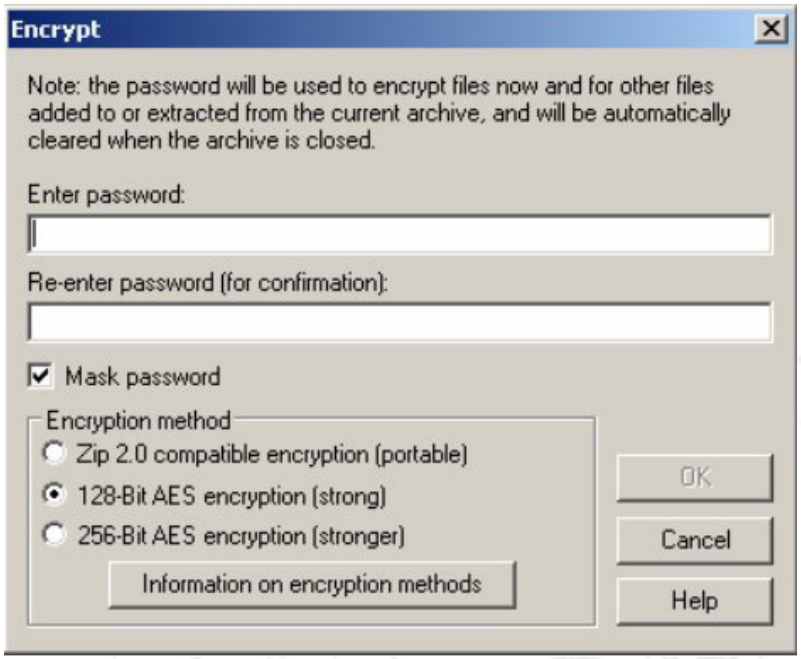
En la práctica, la mayoría de las aplicaciones criptográficas utilizan tanto algoritmos simétricos como asimétricos (así como funciones picadillo). Esto es conocido, en algunas ocasiones, como esquemas híbridos.

La razón para utilizar ambos esquemas, es que cada uno tiene sus fortalezas y debilidades... las cuales veremos a continuación.

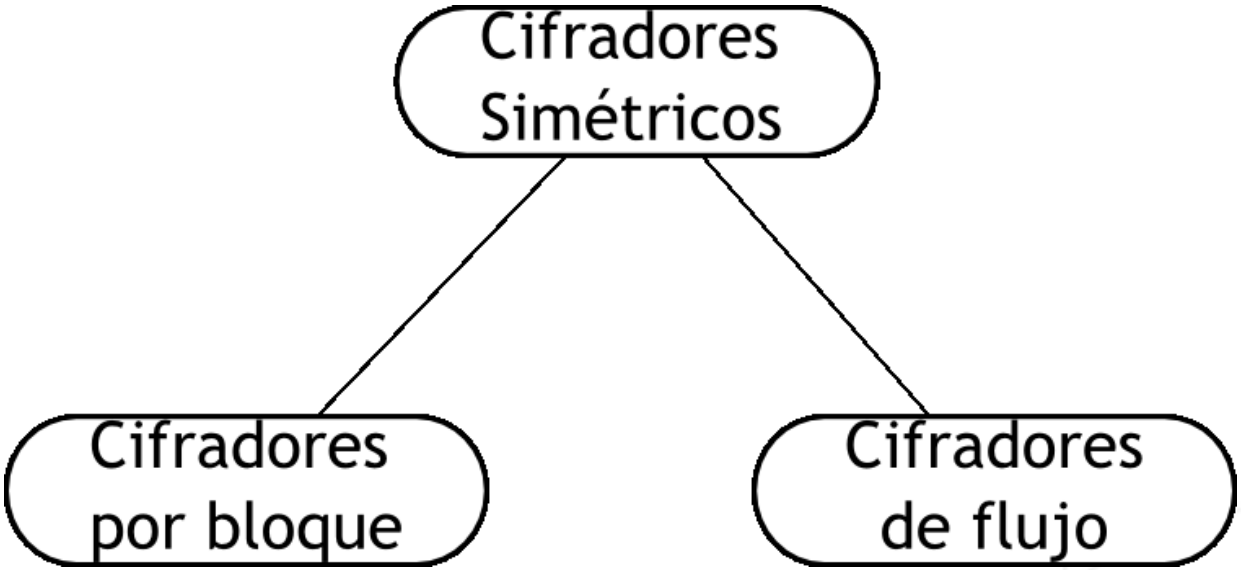
Criptografía simétrica

Los esquemas de criptografía simétrica también son referidos como esquemas o algoritmos de clave-simétrica, clave-secreta, o clave-única.





Tipos de cifradores simétricos



Ejemplo

Dados Alice, y Bob, que se quieren comunicar a través de un canal inseguro (como internet, el aire, etc.)...

Ejemplo

Dados Alice, y Bob, que se quieren comunicar a través de un canal inseguro (como internet, el aire, etc.)...

El problema comienza cuando un tercero: Eve, Charlie u Oscar; tienen acceso al canal: metiéndose a un enruteador, escuchando las señales del radio del WiFi, etc.

Este tipo de escuchas no autorizados se le conoce como *eavesdropping*. Existen muchos casos en los que Alice y Bob preferirían comunicarse sin ser escuchados.

Diagrama inseguro

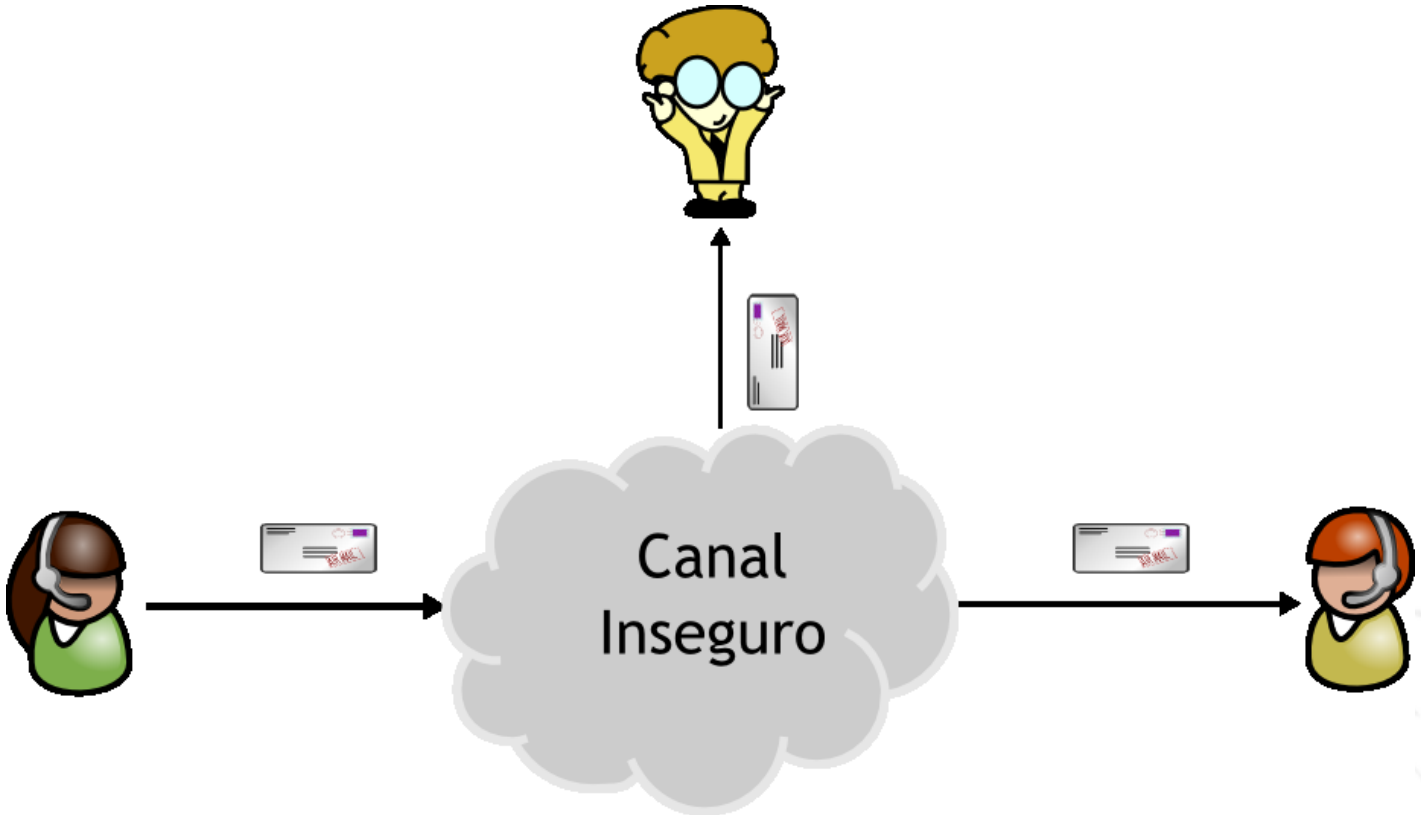
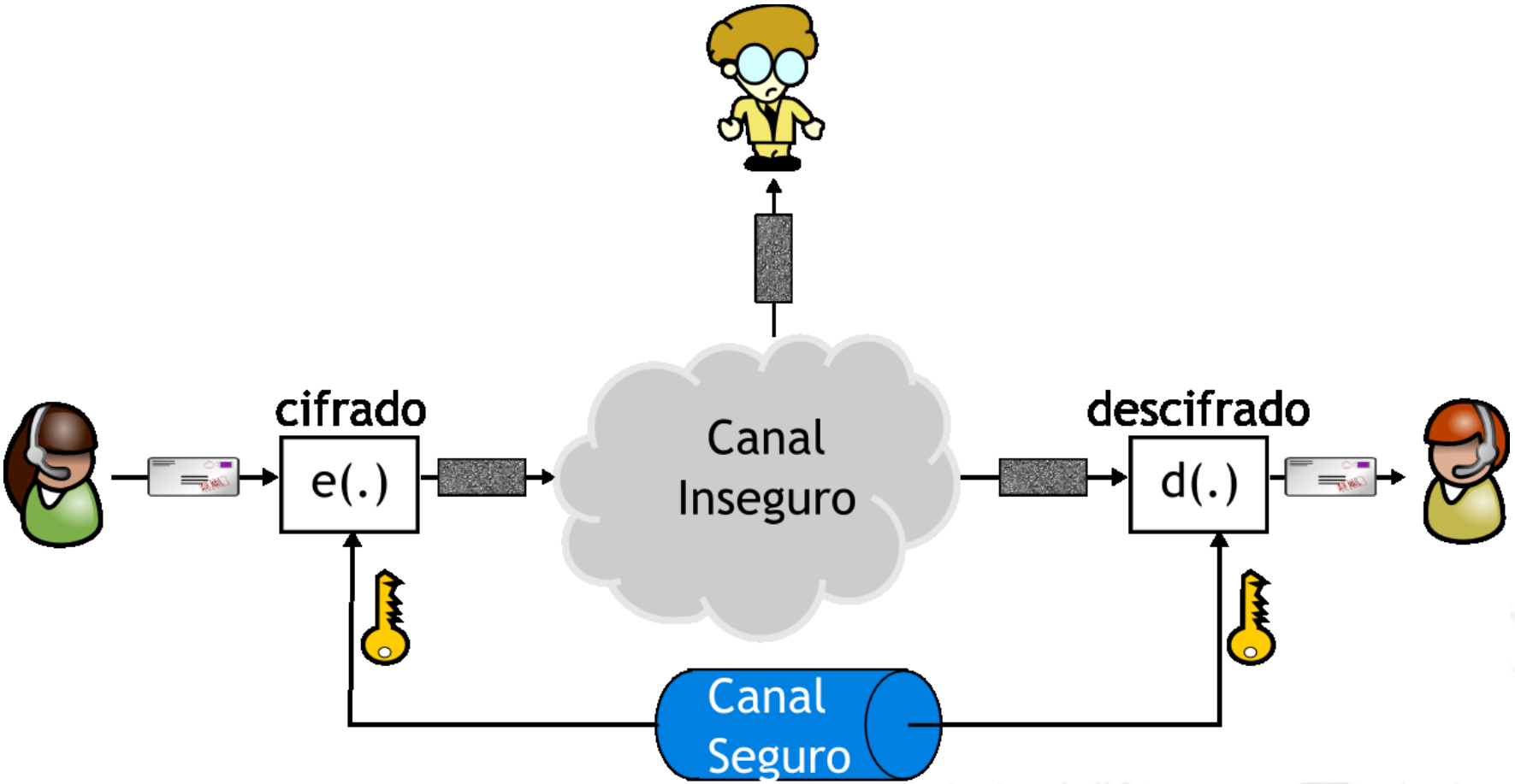


Diagrama con protección



Principio de Kerckhoff

En 1883 Augusto Kerckhoff escribió dos artículos en la revista La Cryptographie Militaire, en donde establecía seis principios de diseño para los cifradores militares:

- El sistema debe de ser prácticamente, si no es que matemáticamente, indescifrable.
- No debe de ser un requerimiento el que sea secreto, y debe de ser posible que caiga en manos del enemigo sin resultar ser un inconveniente.
- Sus claves deben de ser comunicables y retenibles sin la ayuda de notas escritas, y canjeables o modificables a discresión de los corresponsales.
- . . .

Principio de Kerckhoff 2

- . . .
- Debe de ser aplicable a la correspondencia telegráfica.
- Debe de ser portable, y su uso y funcionamiento no debe de requerir la presencia de varias personas.
- Finalmente, es necesario que, dadas las circunstancias que requieran su aplicación, que el sistema sea fácil de usar, no requiriendo esfuerzo mental ni el conocimiento previo de una larga serie de reglas que seguir.

Dado el poder computacional actual, algunos ya no son relevantes, sin embargo, el segundo axioma es vital y se conoce como el Principio de Kerchkhoff

Contenido, sección 2

Cifrador por sustitución

También conocido como cifrador por reemplazo, es uno de los métodos más simples para cifrar un texto.

La idea es sustituir cada letra del alfabeta por otra (o la misma), de tal manera que el texto no pueda entenderse a simple vista:

Ejemplo:

$$A \to L$$
$$B \to C$$
$$C \to J$$
$$\vdots$$

BABA = CLCL

Ataques al cifrador por sustitución

- Fuerza bruta, búsqueda exhaustiva.

El atacante tiene el texto cifrado gracias a que escuchó la conversación; además tiene una parte del texto original, por ejemplo: la cabecera del mensaje (i.e. *%PDF-1.4*, *PK*, *GIF87a*, *0xFFD8*)

Ahora solo tiene que probar atacar el inicio del texto con todas claves posibles hasta que coincida.

Ataque por fuerza bruta

Formalmente,

Búsqueda exhaustiva básica de clave o ataque de fuerza bruta

Dada una pareja (x, y), el texto en claro y el texto cifrado, y sea $K = \{k_0, \ldots, k_{n-1}\}$ sea el espacio de todas las posibles claves. Un ataque de fuerza bruta verifica a todo $k_i \in K$ si:

$$d_{k_i}(y) \stackrel{?}{=} x,$$

Si la relación lógica se mantiene, entonces se ha encontrado la clave y se detiene el proceso, de otro modo, se continua.

$d(.)$ es la función de descifrado. En la práctica depende del protocolo.

Ataques por fuerza bruta

En principio, todos los cifradores *simétricos* son susceptibles a ataques por fuerza bruta. Que sea factible o no, depende del espacio de la clave (el número de posibles claves).

Por ejemplo, el NIP de las tarjetas es de 4 dígitos, existen 10^4 posibles NIPs. En este caso, robar dinero de un cajero automático tardaría nada si no fuera porque los bancos bloquean las tarjetas ante ataques.

Ataques por fuerza bruta

En cambio, si al realizar un ataque utilizando alguna computadora moderna toma mucho tiempo (i.e. décadas), se dice que el cifrador es *computacionalmente seguro* ante ataques de fuerza bruta.

En el caso del cifrador por sustitución, la letra A se sustituyó por la letra L, pero teníamos 26 opciones. La letra B se sustituyó por la letra C, de las 25 opciones restantes. Y así sucesivamente.

El número de posibles sustituciones en un ataque por fuerza bruta es:

Ataques por fuerza bruta

En cambio, si al realizar un ataque utilizando alguna computadora moderna toma mucho tiempo (i.e. décadas), se dice que el cifrador es *computacionalmente seguro* ante ataques de fuerza bruta.

En el caso del cifrador por sustitución, la letra A se sustituyó por la letra L, pero teníamos 26 opciones. La letra B se sustituyó por la letra C, de las 25 opciones restantes. Y así sucesivamente.

El número de posibles sustituciones en un ataque por fuerza bruta es:

$$26 \cdot 25 \cdots 1 = 26! \approx 2^{88}.$$

¿Cuánto es 2^{88}?

- Un procesador Intel Core i7 @3.4 GHz realiza alrededor de 2^{31} ciclos de reloj por segundo, aunque tiene 4 cores...

Ciclos de reloj

- Para realizar 2^{32} ciclos de reloj, se requieren el doble de cores que el nivel anterior (2^{31}), esto es, el procesador realiza 2^{33} ciclos de reloj en total.
- tenemos $88 - 32 = 55$, hay que duplicar la cantidad de cores 55 veces.
- Es un crecimiento geométrico, tal como la leyenda del Ambalappuzha Paal Payasam (granos de arroz y el ajedrez)

Ciclos de reloj

- Para realizar 2^{32} ciclos de reloj, se requieren el doble de cores que el nivel anterior (2^{31}), esto es, el procesador realiza 2^{33} ciclos de reloj en total.
- tenemos $88 - 32 = 55$, hay que duplicar la cantidad de cores 55 veces.
- Es un crecimiento geométrico, tal como la leyenda del Ambalappuzha Paal Payasam (granos de arroz y el ajedrez)

- Sin embargo, eso es sólo 1 segundo. . .

- Sin embargo, eso es sólo 1 segundo. . . además de que en algunos casos hay que analizar la información.

Para fines de benchmarking, normalmente no se utiliza el TurboBoost, además que en un ataque aumentaría la radiación térmica.

Entonces se dice que este esquema es seguro ante ataques de fuerza bruta...

- Sin embargo, eso es sólo 1 segundo. . . además de que en algunos casos hay que analizar la información.

Para fines de benchmarking, normalmente no se utiliza el TurboBoost, además que en un ataque aumentaría la radiación térmica.

Entonces se dice que este esquema es seguro ante ataques de fuerza bruta... (a menos que el *bruto* haya sido el que seleccionó la clave)

Un ataque *diferente*

En el ataque por fuerza bruta, tomamos al cifrador como una caja negra, sin analizarla internamente.

El cifrador por sustitución puede romperse mediante un ataque analítico.

La principal debilidad del cifrador, es que cada símbolo del texto en claro tiene una única representación en el texto cifrado. Esto es, que las propiedades estadísticas del texto en claro se preservan en el texto cifrado.

Letras en los idiomas

La letra que más se repite en el idioma inglés es la letra “e” (alrededor del 13% de los textos), después la “t” con un 9%, y la “a” con un 8%.

En español, la frecuencia es similar (la “e” también es la más utilizada). Se puede construir una tabla para el idioma español tomando cualquier libro y contando la ocurrencia de cada una de las letras.

Letras en los idiomas

La letra que más se repite en el idioma inglés es la letra “e” (alrededor del 13% de los textos), después la “t” con un 9%, y la “a” con un 8%.

En español, la frecuencia es similar (la “e” también es la más utilizada). Se puede construir una tabla para el idioma español tomando cualquier libro y contando la ocurrencia de cada una de las letras.

Pero aquí están ordenadas de mayor a menor frecuencia: E A O S R N I D L C T U M P B G V Y Q H F Z J Ñ X W K

Notas sobre el conteo de letras

- En un diccionario, la letra que más se repite tiende a ser la “a”
- En un libro, como en el Quijote, se mantiene el orden antes mencionado.
- Aunque hay excepciones.
- además, existen muchas frases cortas en el español que van con la “e”: qué, le, sé, etc.

Finalmente, con la estadística es muy fácil descifrar un texto por sustitución.

Cifrado de Cæsar

El cifrador de César es un tipo especial del cifrador por sustitución en el cual los valores del alfabeto se rotaban una cantidad de letras en particular.

Por ejemplo, si la clave era 13, entonces la tabla de sustitución es:

$$
\begin{aligned}
A &\rightarrow N \\
B &\rightarrow \tilde{N} \\
C &\rightarrow O \\
&\vdots
\end{aligned}
$$

Para el idioma español.

Cifrado de Cæsar

El cifrador de César es un tipo especial del cifrador por sustitución en el cual los valores del alfabeto se rotaban una cantidad de letras en particular.

Por ejemplo, si la clave era 13, entonces la tabla de sustitución es:

$$
\begin{aligned}
A &\rightarrow N \\
B &\rightarrow \tilde{N} \\
C &\rightarrow O \\
&\vdots
\end{aligned}
$$

Para el idioma español.

¿Cuál es el espacio de claves?

Máquina enigma

La máquina enigma era una especie de máquina de escribir con un determinado número de rotores en serie que giraban de diferente forma a cada pulsación de una tecla, de tal manera que la salida de un rotor era la entrada para el siguiente, y así sucesivamente.

Estos rotores podían cambiar su posición inicial (que era la clave), de tal manera que cada vez que se enviaba un mensaje, se utilizaba una configuración diferente.

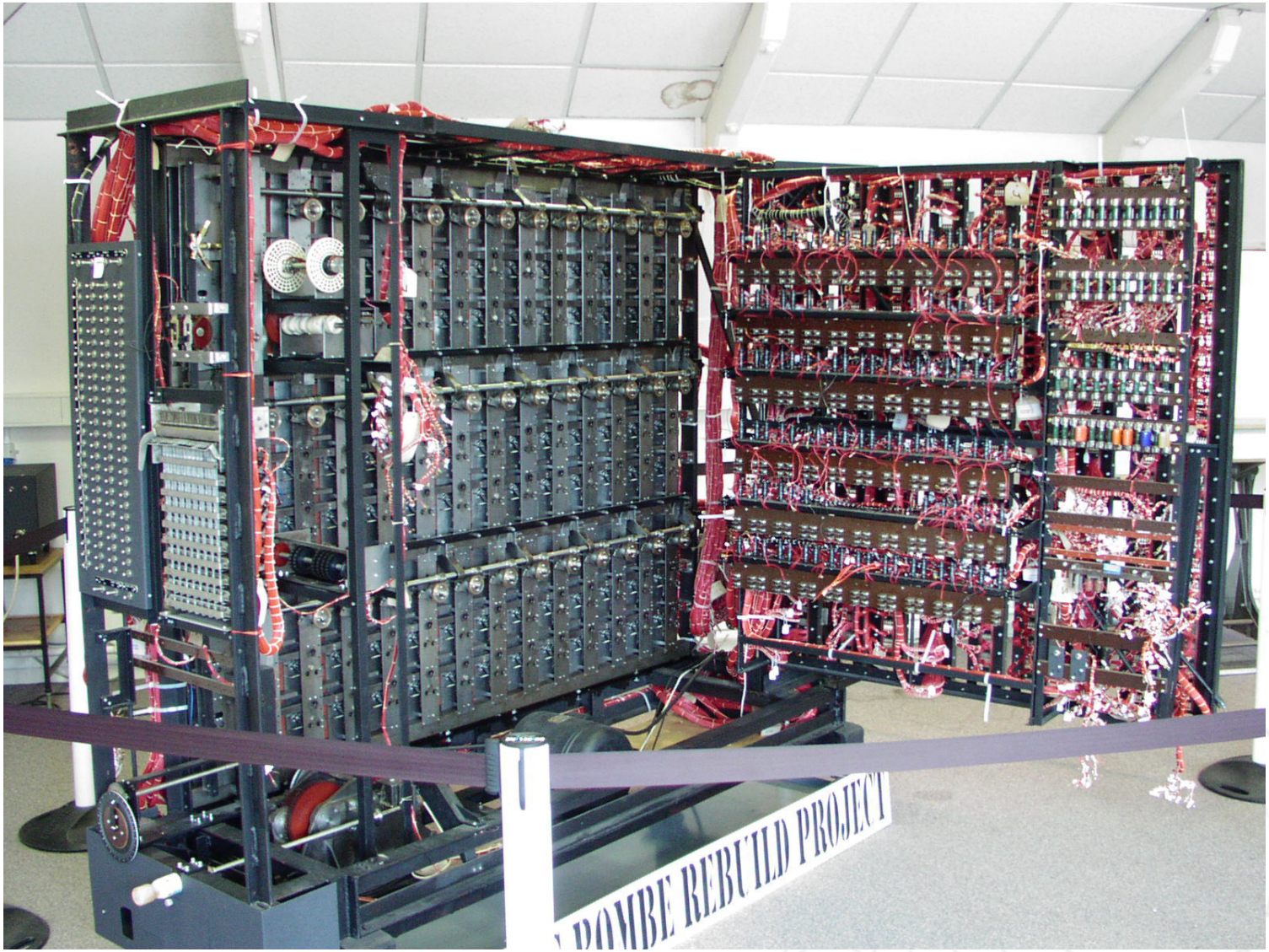
Segunda Guerra Mundial

Durante el Franquismo, los nazis proveyeron a Francisco Franco de unas máquinas limitadas para comunicarse, y probarlas.

Ya probadas, se utilizaron extensamente durante la Segunda Guerra mundial para enviar instrucciones a las líneas de batalla.

Todos los ataques alemanes eran sorpresa y 100% efectivos. Fue hasta que los británicos con su laboratorio en Bletchley Park (liderado por Alan Turín), y las máquinas polacas *bomb*e, que pudieron descifrar los mensajes, y salvar millones de vidas.

Bombes polacas



Simulador máquina enigma

En línea

Contenido, sección 3

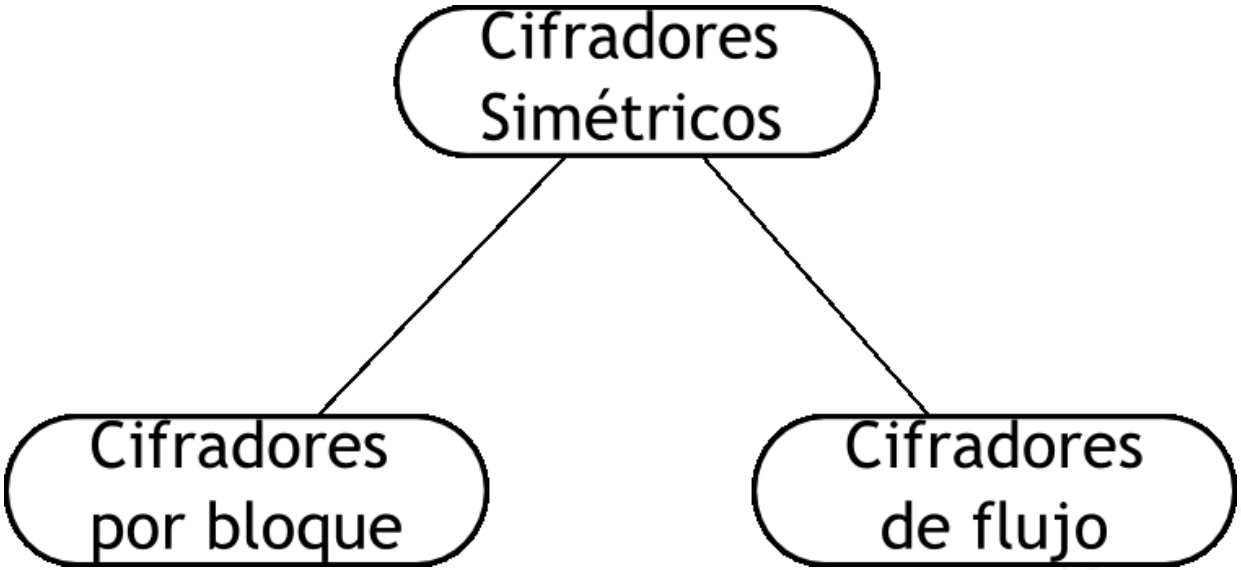
Criptosistema

Un *criptosistema* es una 5-tupla $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, con las siguientes condiciones:

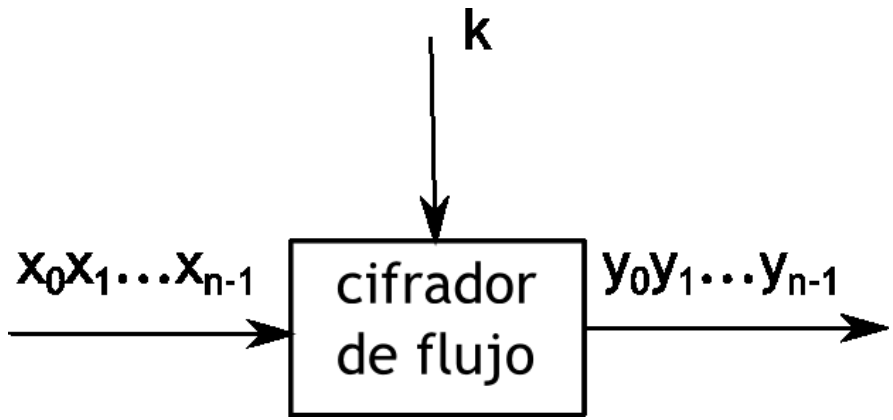
- $\mathcal{P}$ es el conjunto finito de todos los textos en claro posibles
- $\mathcal{C}$ es el conjunto finito de todos los textos cifrados posibles
- $\mathcal{K}$, el *espacio de claves*, es el conjunto finito de todas las *claves* posibles
- $\forall K \in \mathcal{K}$, $\exists E_K \in \mathcal{E}$ (regla de cifrado), $\exists D_K \in \mathcal{D}$ (regla de descifrado)

Cada $E_K : \mathcal{P} \to \mathcal{C}$, $D_K : \mathcal{C} \to \mathcal{P}$, son funciones tal que $\forall x \in \mathcal{P}$, $D_K(E_K(x)) = x$.

Tipos de cifradores simétricos



Cifradores simétricos



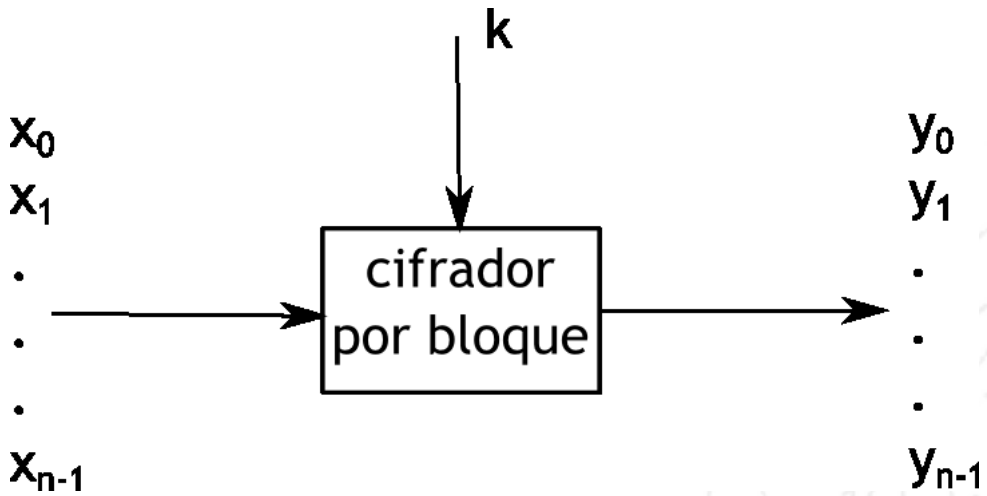
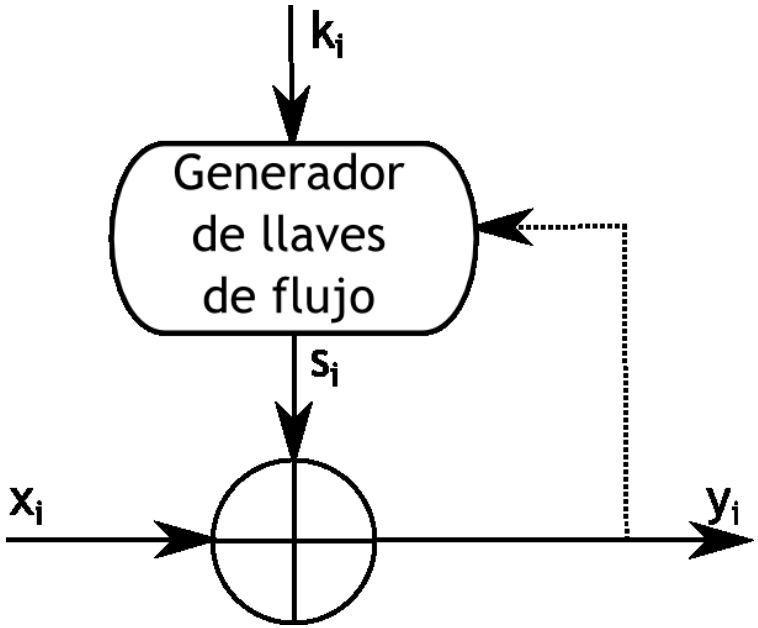


Diagrama cifradores de flujo



Cifradores de flujo.
Cifran bits individualmente. Esto se hace al añadir un bit de un flujo de la llave a un bit del texto en claro.

Los esquemas *síncronos* son en los que la línea punteada en el diagrama no está presente (el cifrado depende exclusivamente de la llave. Son *asíncronos*, cuando existe la dependencia del bit cifrado con los bits cifrados anteriormente (la línea punteada está activa).

Operación

Cifrado y Descifrado de flujo

El texto en claro, el texto cifrado, y el flujo de la llave consiste en bits individuales: $x_i, y_i, s_i \in \{0, 1\}$

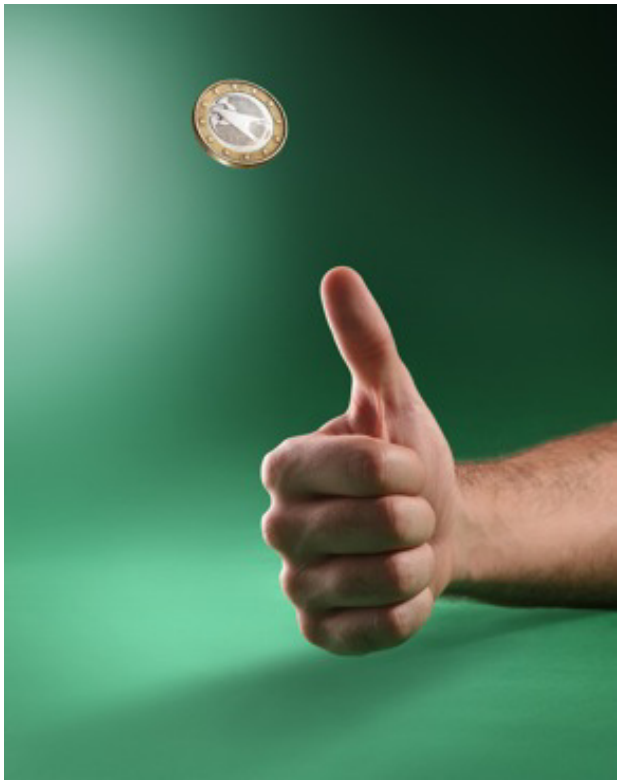
- Cifrado: $y_i = e_{s_i}(x_i) \equiv x_i + s_i \bmod 2$
- Descifrado: $x_i = d_{s_i}(y_i) \equiv y_i + s_i \bmod 2$

Nota: La suma módulo 2 equivale a la operación xor.

Cifradores de Bloque

Cifradores de bloque.
Cifran un bloque completo de bits del texto en claro a la vez con la misma llave. Esto significa que el cifrado de cualquier bit del texto en claro en un bloque dado depende de los otros bits del bloque. En la práctica, la mayoría de los cifradores de bloque esperan bloques de 128 bits (AES), o 64 bits (DES).

Números aleatorios



Generadores de números verdaderamente aleatorios.

- Los generadores de números realmente aleatorios (TRNGs) se caracterizan por el hecho de que su salida no puede ser reproducida. Por ejemplo, si echamos 100 volados y registramos los resultados como una secuencia de bits, dicha secuencia es virtualmente irrepetible (la probabilidad de repetirla es de $1/2^{100}$).
- Los TRNGs están basados en procesos físicos.

Número pseudo-aleatorios

Generadores de números pseudo-aleatorios.

- Los generadores de números pseudo-aleatorios (PRNGs) generan secuencias que pueden ser calculadas a partir de un valor inicial llamado semilla (seed).

Por ejemplo, la función rand(.) del ANSI C es algo así:

$$s_0 = 12345$$
$$s_{i+1} \equiv 1103515245 \cdot s_i + 12345 \bmod 2^{31}, i = 0, 1, \ldots$$

Generadores de números pseudo-aleatorios criptográficamente seguros

Un **generador de números pseudo-aleatorios criptográficamente seguros** (CSPRNGs) es un tipo especial de generador que es impredecible. Dada una secuencia de bits, no existe un algoritmo polinomial que determine el siguiente bit con una probabilidad mayor al 50%. Igualmente, dada una secuencia de bits, es imposible determinar el anterior.

La impredecibilidad de los CSPRNGs es única para la criptografía, por lo que si se toma un generador no diseñado específicamente para criptografía, probablemente no sirva para un producto comercial.

Incondicionalmente seguro

Incondicionalmente seguro.
Un criptosistema es incondicionalmente seguro (o seguro en términos de la teoría de la información) si no puede ser roto aún con recursos informáticos infinitos.

Suponga un criptosistema simétrico con una llave de 10,000 bits que solo pueda ser roto mediante búsqueda exhaustiva (fuerza bruta). Se necesitarían $2^{10,000}$ computadoras. El sistema no es incondicionalmente seguro, pero es computacionalmente seguro (se estima que existen entre 2^{239} y 2^{289} átomos en el universo)

One-time pad

Aquí está un criptosistema incondicionalmente seguro:

One-time pad

Es un cifrador de flujo en el cual:

- el flujo de la llave $s_0, s_1, \ldots$ es generador por un TRNGs
- el flujo de la llave es solamente conocido por los extremos de la comunicación
- cada bit del flujo de la llave s_i es utilizado una única vez.

se le conoce como one-time pad. El one-time pad es incondicionalmente seguro.

Más sobre el one-time pad.

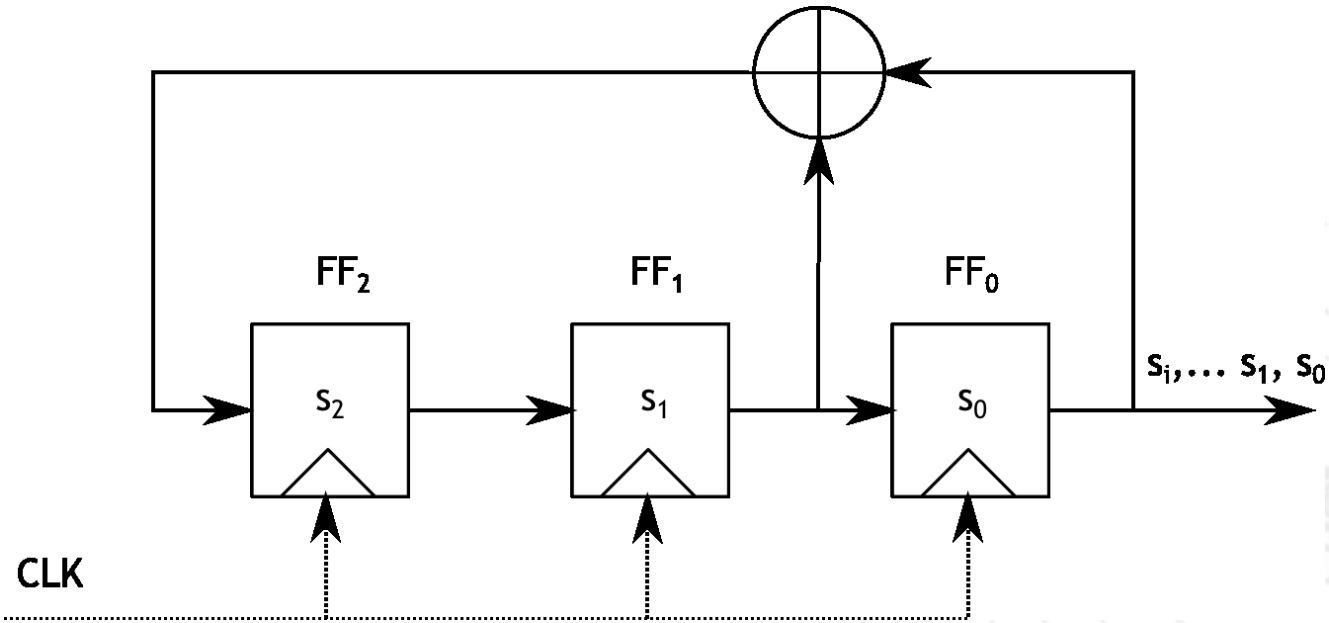
Cada bit del texto cifrado se forma de la siguiente manera:

$$\begin{aligned} y_0 &\equiv x_0 + s_0 \bmod 2 \\ y_1 &\equiv x_1 + s_1 \bmod 2 \\ &\vdots \\ y_{n-1} &\equiv x_{n-1} + s_{n-1} \bmod 2 \end{aligned}$$

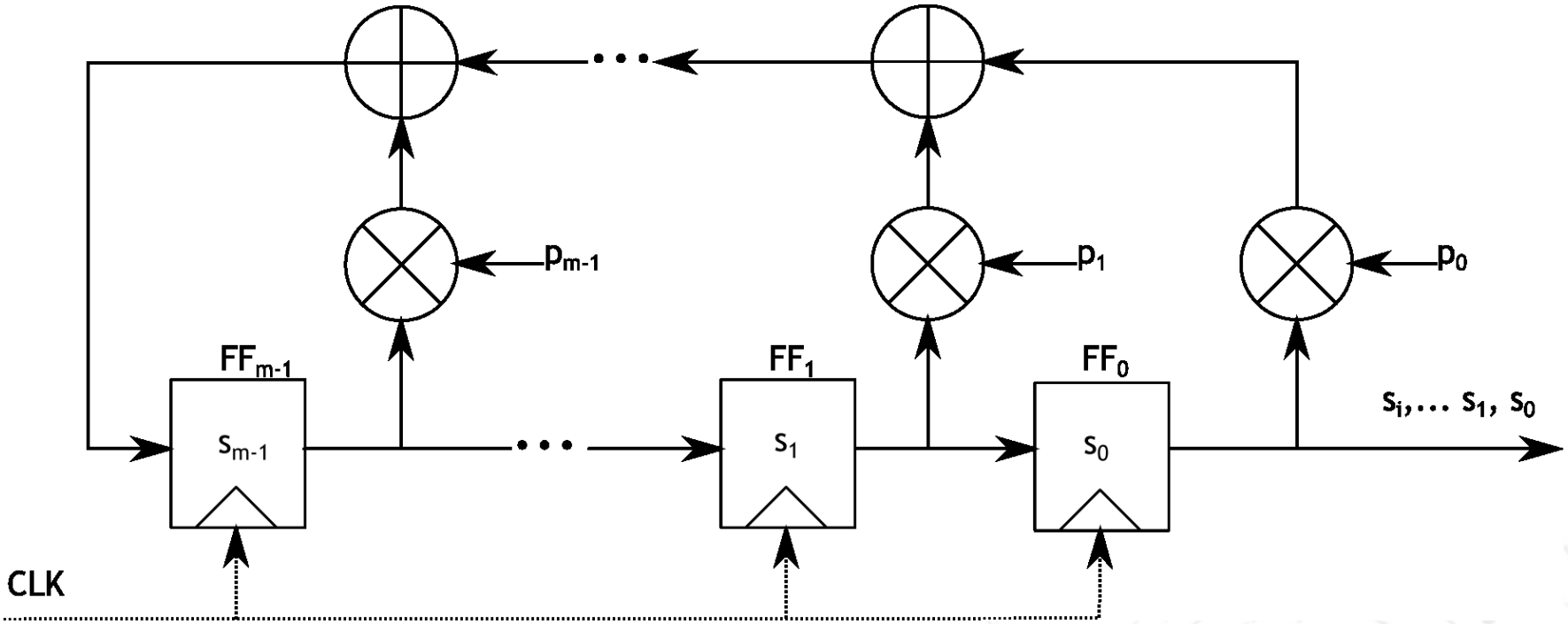
es una ecuación con dos incógnitas por cada bit. Aún si se conoce y_i, los valores de $x_i \in \{0, 1\}$ tienen exactamente la misma probabilidad si se utilizó un TRNG. Sin embargo, los bits aleatorios no se pueden reutilizar, lo que nos lleva al problema de distribución de llaves.

Linear Feedback Shift Register

Un **LFSR** consiste en elementos de almacenamiento sincronizados (flip-flops) y una ruta de retroalimentación. El número de elementos de almacenamiento establece el grado del LFSR. La red de retroalimentación calcula la entrada del último flip-flop como una suma módulo 2 (XOR) de ciertos flip-flops en el registro.



LFSR



DES - Data Encryption Standard

Historia:

- Las compañías financieras necesitaban de un mecanismo de protección que tuviese el visto bueno del gobierno norteamericano.
- El primer llamado para el concurso fue en mayo de 1973, seguido de un segundo llamado en 1974
- No hubo muchas propuestas. IBM presentó Lucifer.
- El gobierno trabajó con la IBM para rediseñar el algoritmo

DES - Data Encryption Standard

Cronograma:

- 1973 - el NBS (El Buró nacional de estándares de los EEUU, ahora NIST) solicita propuestas de criptosistemas para documentos "no-clasificados"
- 1974 - el NBS repite el requerimiento.
  - IBM responde con una modificación de LUCIFER.
    LUCIFER está basado en una familia de cifradores creados por Horst Feistel a finales de los 1960s.
  - NBS le pide a la NSA (National Security Agency) que lo evalúe.
    Por esas fechas la NSA negaba su propia existencia.
  - La NSA convence a la IBM que reduzca el tamaño estándar de la clave del LUCIFER de 128-bits a 56-bits, esto posibilitaría los ataques por fuerza bruta.
  - . . .

DES

- . . .
  - LA NSA escogió algunos de los parámetros del sistema del cifrador.
    En particular pidió que se agregara soporte a un tipo de ataques en particular. Estos ataques se descubrirían en los 1990s, y se llamarían ataques diferenciales. Se desconoce si la NSA sabía de su existencia, o si sólo sospechaban.
  - IBM obtiene la patente del DES (Data Encryption Standar).
- 1975 se publican los detalles del algoritmo, excepto algunos criterios de funciones internas. La discusión pública comienza. La gente tenía la incertidumbre si la NSA le había metido mano para su beneficio (introducción de trap-doors).
- . . .

DES

- 1976 se adopta como un estándar para todos los doumentos gubernamentales “no-clasificados”.
  Data Encryption Standard - FIPS PUB 46.
- Se hace estándar para hardware en 1977
- ANSI X3.92-1981 (hardware + software)
- ANSI X3.106-1983 (modes of operation)
- Australia AS2805.5-1985
- Se reafirmó como estándar hasta 1999 cuando se substituyó por el AES - Advanced Encryption Standard.

Confusión y difusión

De acuerdo con el teorista de la información Claude Shannon, hay dos operaciones primitivas con las cuales los algoritmos de cifrado fuerte pueden ser construídos:

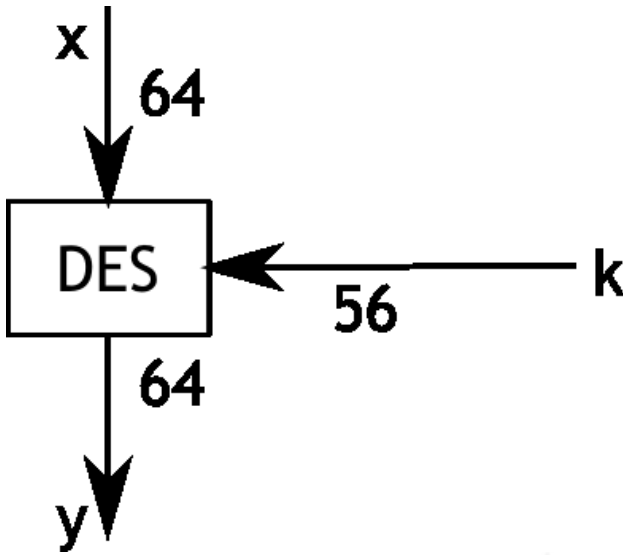
- **Confusión**. Operación de cifrado en la que la relación entre la llave y el texto cifrado se ocultan.
- **Difusión**. Operación de cifrado en la que la influencia de un símbolo del texto en claro se reparte sobre muchos del texto cifrado para ocultar las propiedades estadísticas del texto en claro; i.e., permutaciones y mezcla columnas.

Confusión y difusión

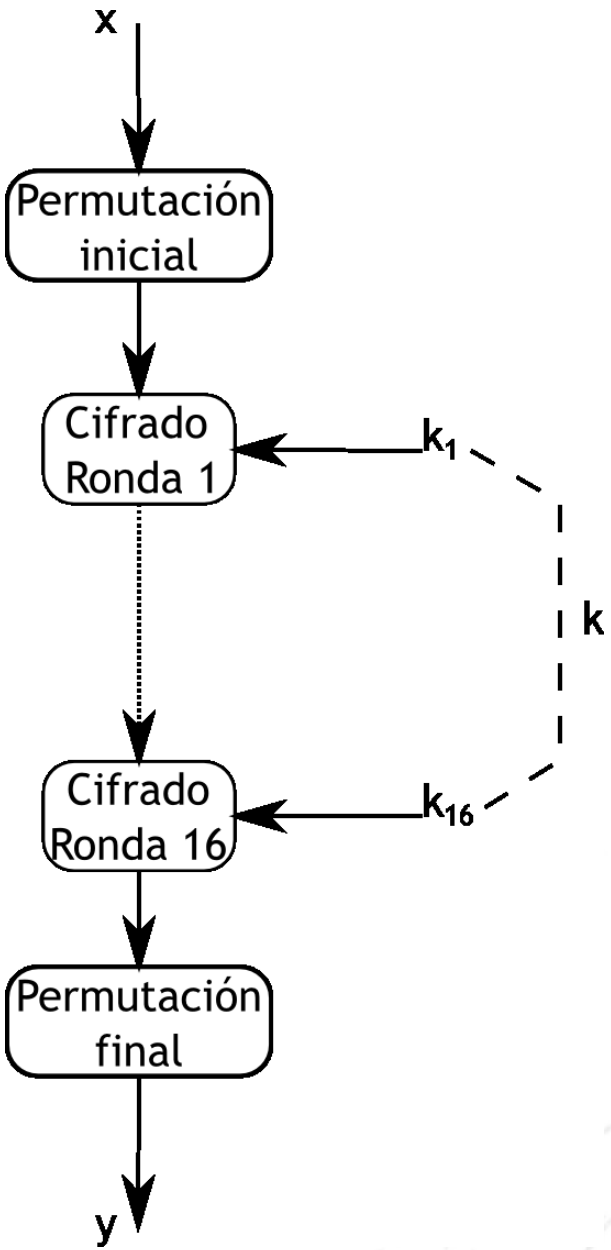
- Utilizar sólo confusión o sólo difusión resulta en cifradores que no son seguros.
- A través de la concatenación de dichas primitivas se construyen cifradores fuertes; estos se conocen como cifradores de producto.
- Todos los cifradores modernos y seguros son cifradores de producto.
- Una secuencia de operaciones de confusión y difusión se le conoce como ronda.
- Las rondas se repiten n veces sobre el texto en claro o el resultante de la ronda anterior. Esto provoca un efecto avalancha.

DES caja negra

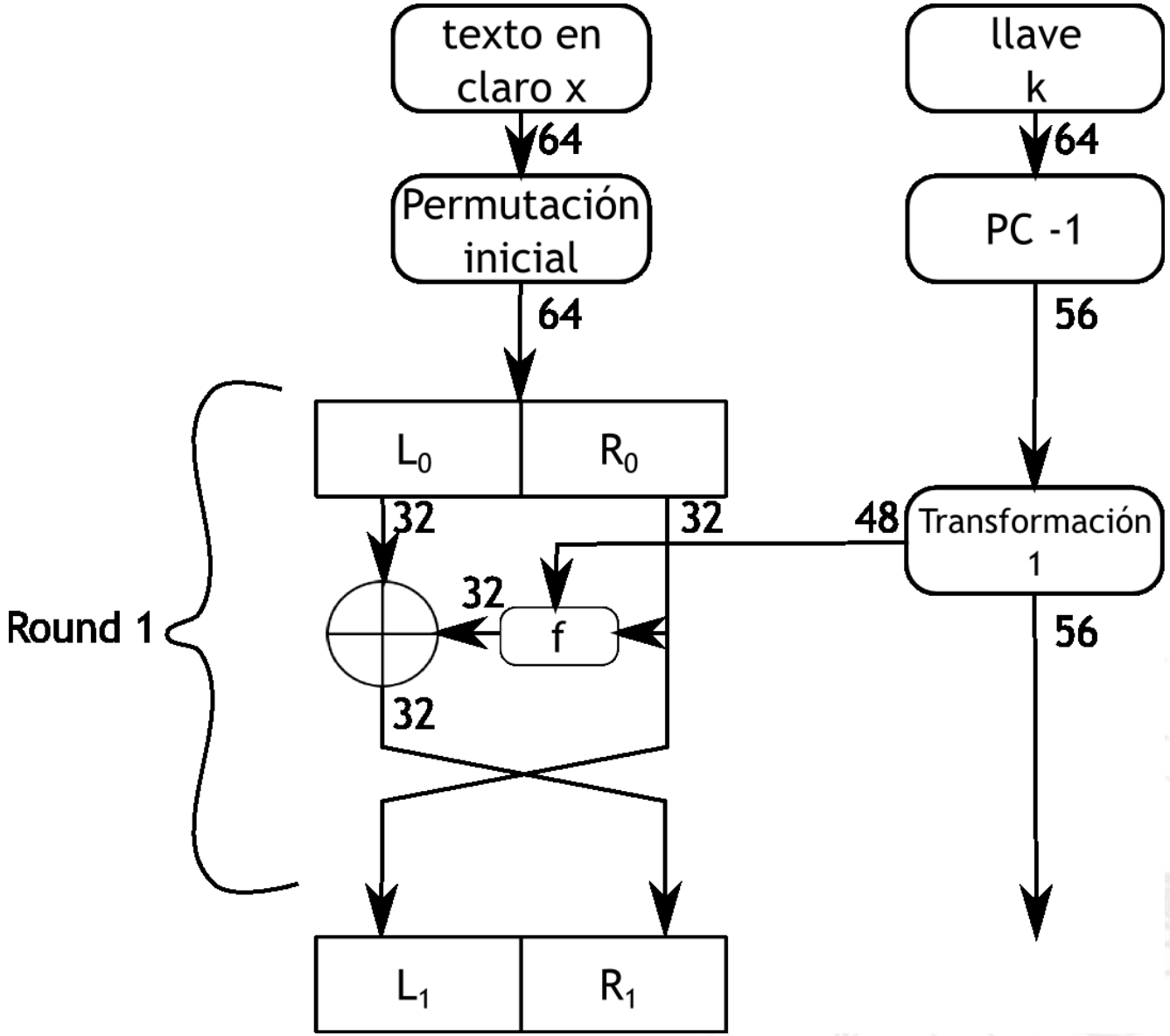
El DES es un cifrador que toma bloques de 64-bits de longitud con una llave de 56-bits.



DES estructura general



DES estructura Feistel



Ataques

- 1977 - Diffie y Hellman sugirieron un diseño de un chip VLSI que puede probar 10^6 llaves/seg. Un equipo con 10^6 de estos circuitos podría romper la clave en cuestión de 10 horas. Costo: USD $20'000,000.00
- 1990 - Eli Biham y Adi Shamir sugirieron un criptoanálisis diferencial
- 1993 - Mitsuru Masui sugirió un criptoanálisis linear

Ataque de la paradoja del cumpleaños en el logaritmo discreto

Dado un primo p, y α y β enteros que no son cero módulo p, es imposible encontrar una x tal que $\alpha^x \equiv \beta \bmod p$, si p es lo suficientemente grande.

Sin embargo, con el ataque del cumpleaños:

- Haga dos listas de longitud $\approx p^{1/2}$
- La primera lista contendrá todos los valores de $\alpha^k \bmod p$, para $\approx p^{1/2}$ valores aleatorios de k.
- La segunda lista contendrá los números $\beta\alpha^{-\ell} \bmod p$, para $\approx p^{1/2}$ valores aleatorios de ℓ.

Cuando haya alguna coincidencia, tenemos que $\alpha^k \equiv \beta\alpha^{-\ell} \bmod p$, por lo que $\alpha^{k+\ell} \equiv \beta \bmod p$. El valor buscado es $x \equiv k + 1 \bmod p - 1 \ldots$

Ataque Meet-in-the-middle

Asuma que Eva capturó un mensaje m y un texto doblemente cifrado $c = E_{k_2}(E_{k_1}(m))$. Calcue y almacene $E_k(m)$ y $D_k(c)$ para todos los valores posibles de la llave k. Compare ambas listas. Debe de existir una coincidencia.

Dado el reducido tamaño de la clave (56-bits):

- En 1998, la Electronic Frontier Foundation rompió una clave en 56 horas: 1536 chips probando 88×10^9 llaves/segundo, con un costo menos a los USD $250,000.
- En 1999, Distribute.Net, en conjunto con la EFF, consiguieron 100,000 equipos voluntarios en internet, y rompieron una clave en 22 horas y 15 minutos.

Suponiendo que usara DES (que no lo usa), cada cuándo cambian su contraseña de Gmail?

Modos de operación

Hemos dicho que DES trabaja con bloques de 64-bits, ¿qué pasa si requerimos cifrar más de 64 bits? Agarramos de 64 bits en 64 bits?

Modos de operación

Hemos dicho que DES trabaja con bloques de 64-bits, ¿qué pasa si requerimos cifrar más de 64 bits? Agarramos de 64 bits en 64 bits? (no)

Modos de bloques

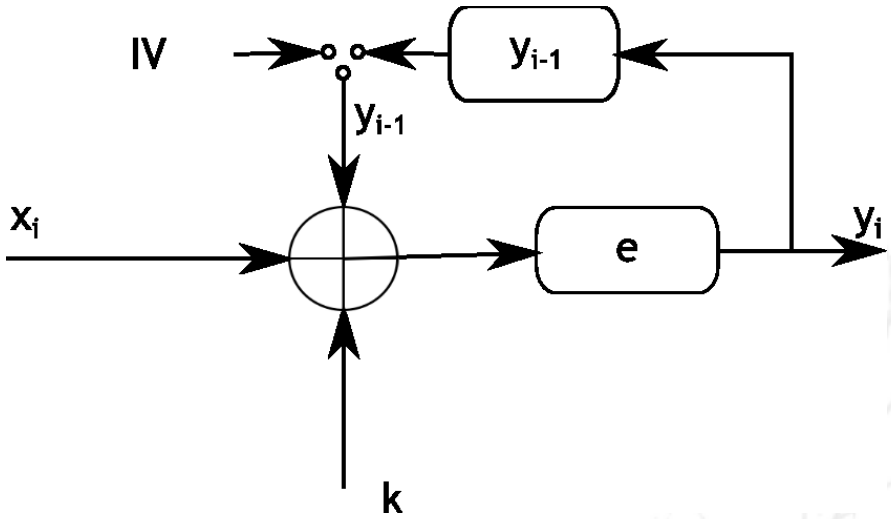
- ECB - Electronic Codebook Block
- CBC - Cipher Block Chaining

Modos de flujo

- CFB - Cipher Feedback
- OFB - Output Feedback

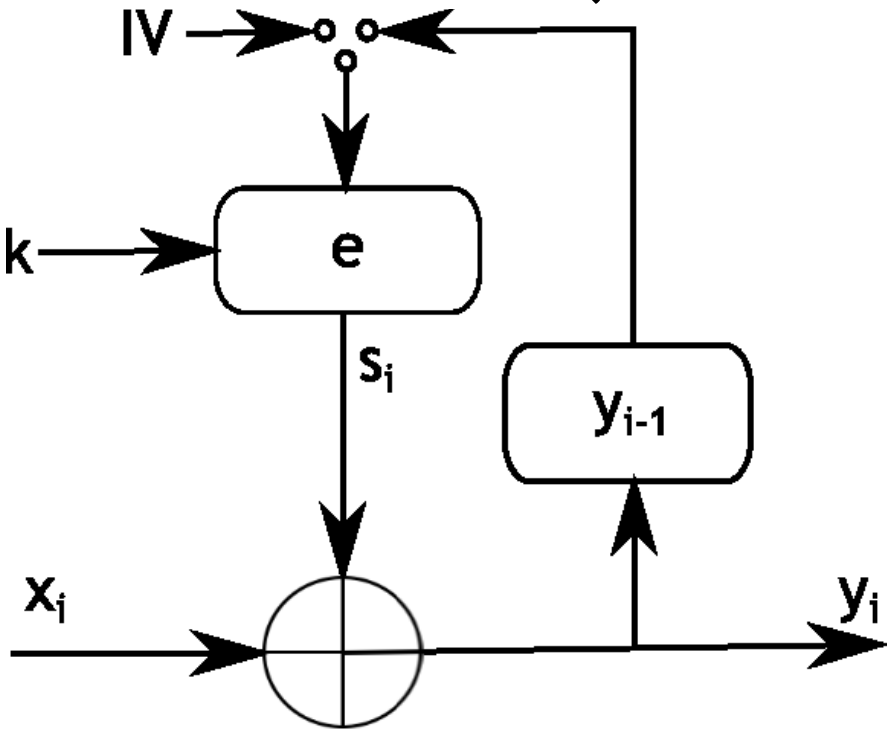
Por bloques

- ECB - El mensaje se rompe en bloques de 64-bits (se rellena con ceros).
- CBC - Hace un xor de la salida anterior con el bloque a cifrar (requiere vector de inicialización).

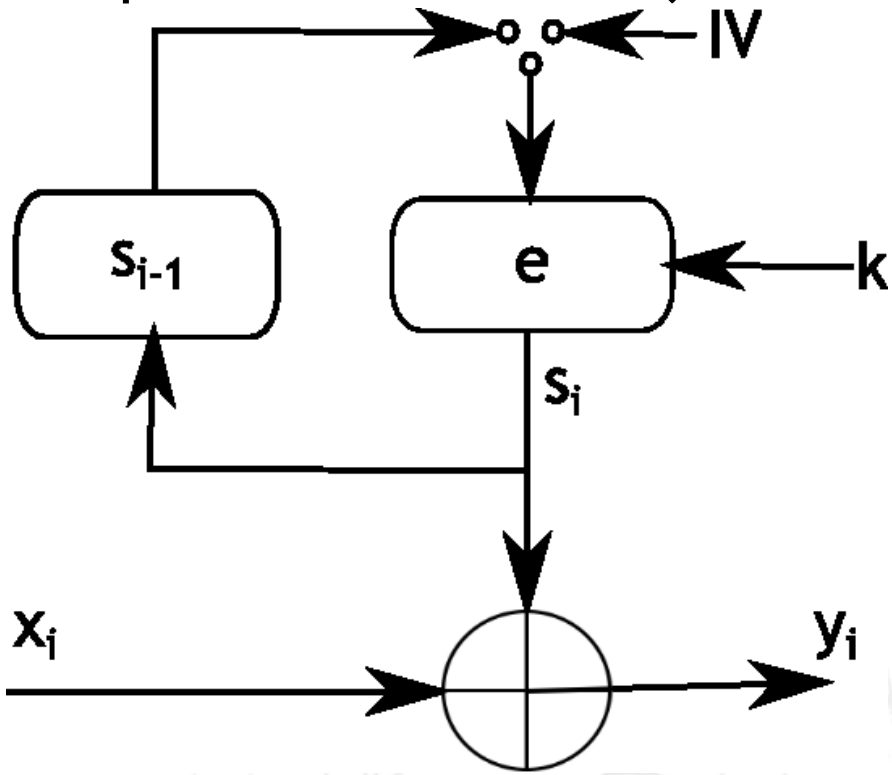


Por flujo

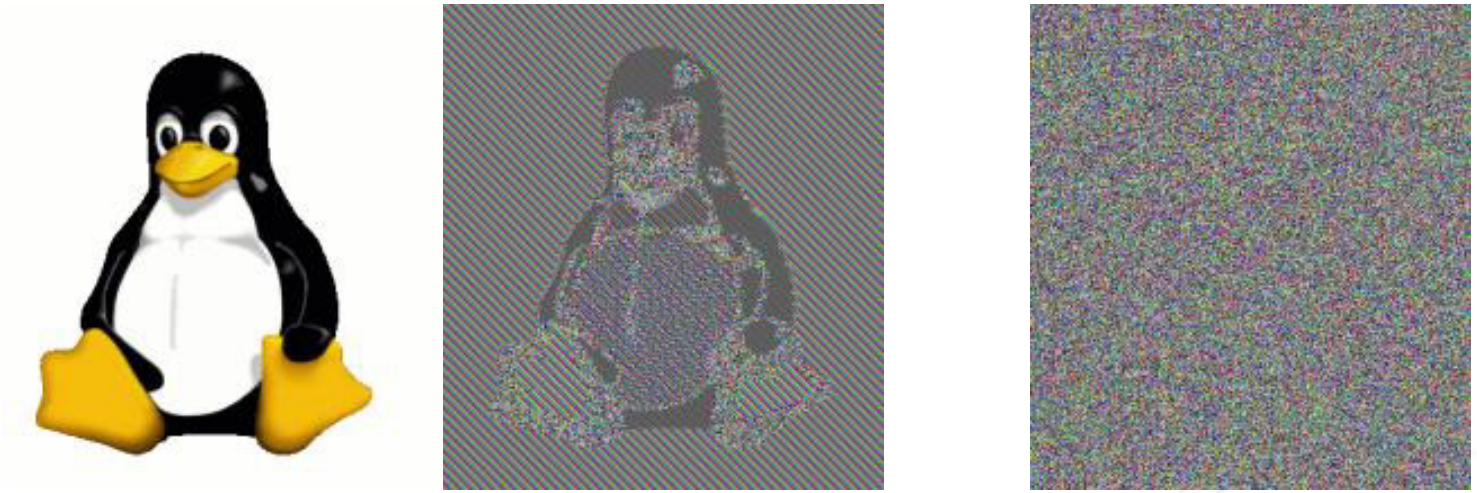
CFB - Se hace un xor de la salida anterior con el mensaje



OFB - El feedback es independiente del mensaje actual



Comparativa

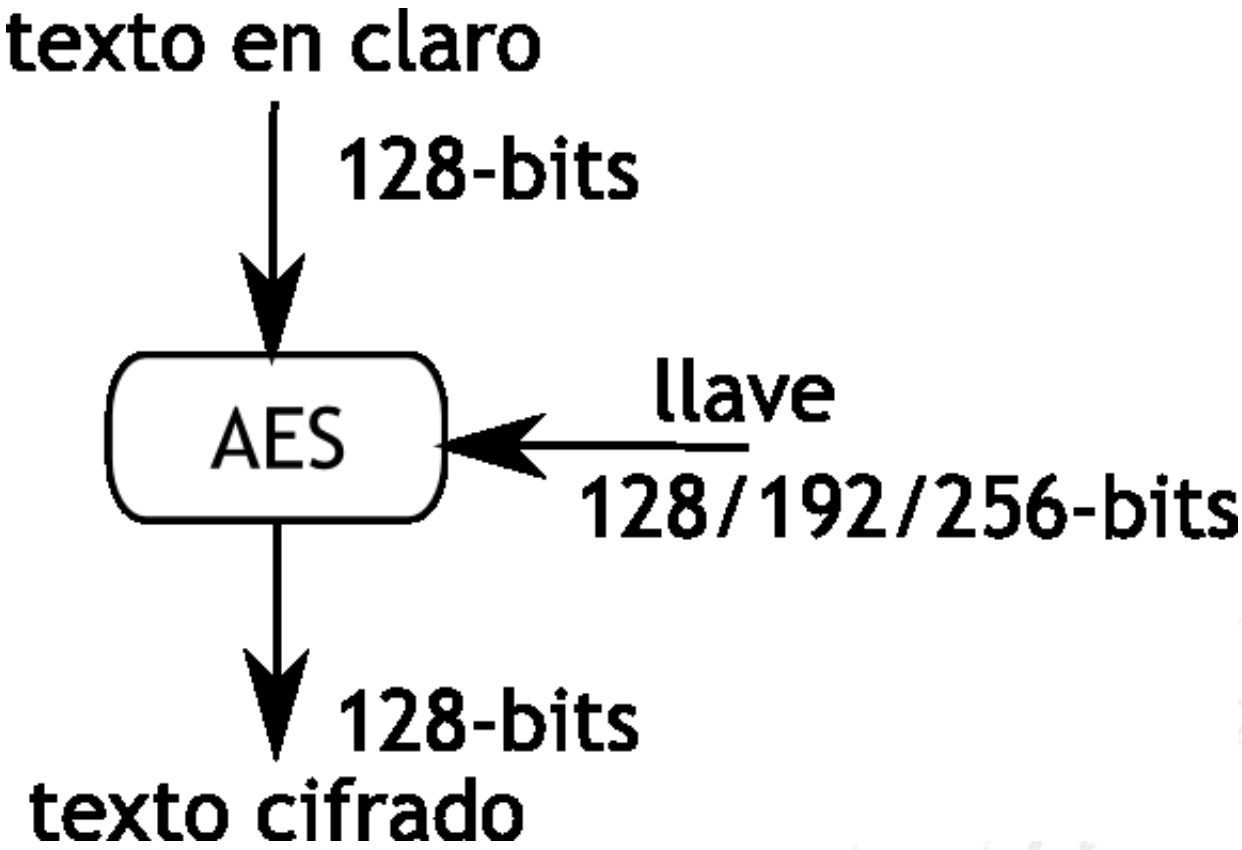


AES - Advanced Encryption Standard

¿Porqué un estándar nuevo?

- Ataques de fuerza bruta
- La solución (Triple DES) lo hace el triple de lento
- DES es eficiente solo para hardware
- Nuevos tipos de ataques
- Utilizar bloques de 64-bits no es útil para todos los escenarios

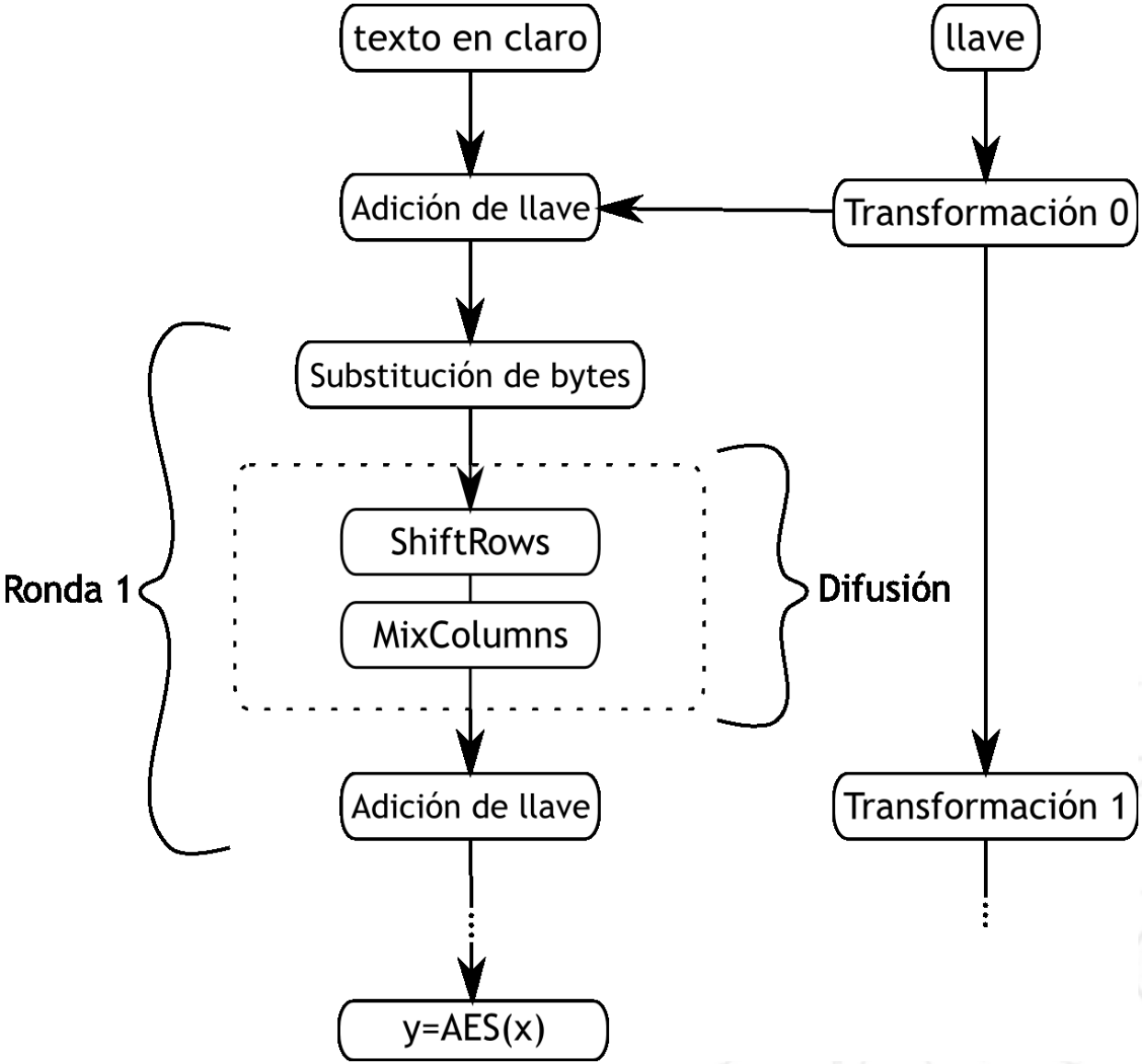
AES - Caja negra



AES - descripción

- AES no utiliza una función Feistel, se desea cifrar todo un bloque por ronda
- Se necesitan 10, 12 o 14 rondas para cifrar con claves de 128, 192 o 256 bits
- En cada ronda hay 3 capas: Adición de llave, de Sustitución de bytes, y de difusión
- la capa de difusión se subdivide en: ShiftRow, que permuta datos a nivel de byte; y MixColumn, que mezcla bloques de 4-bytes dentro de una matriz

AES - rondas


texto en claro
llave
Adición de llave
Transformación 0
Substitución de bytes
ShiftRows
MixColumns
Ronda 1
Difusión
Adición de llave
Transformación 1
y=AES(x)

El nacimiento de la criptografía de llave pública

- En 1976, Whitfield Diffie y Martin Hellman publicaron su famoso artículo: "Nuevas direcciones en criptografía" (New Directions in Cryptography)
- Poco antes, Ralph Merkle inventó una construcción de llave pública para sus clases. Su trabajo se tituló: "Comunicación segura sobre canales inseguros" en 1982 (Secure communication over insecure channels)

El nacimiento de la criptografía de llave pública

- El concepto fue originalmente descubierto por James Ellis, aunque se mantuvo en secreto ya que era infomación clasificada de la GCHQ de 1969 a 1997.
- Adicionalmente, Malcolm Williamson y Clifford Cocks de la GCHQ, descubrieron el intercambio de llave Diffie-Hellman, y el cifrado RSA.

El nacimiento

- Antes de la publicación de “New Directions. . .”, la investigación sobre cifrado en los E.E.U.U. era dominio de la Agencia Nacional de Seguridad (NSA).
- Hasta mediados de 1990’s, la exportación de algoritmos criptográficos era penada con traición.
- Después, solamente prohibía la exportación de algoritmos de seguridad alta si eran leíbles por máquinas (código fuente, ejecutables, etc.).

Curiosidades

- Los algoritmos critográficos se conviertieron en “municiones” para el gobierno
- La gente se hacía playeras con código RSA amenzando con salir del país
- Hubo quien incluso se hizo tatuajes

Curiosidades

- Los algoritmos critográficos se conviertieron en “municiones” para el gobierno
- La gente se hacía playeras con código RSA amenzando con salir del país
- Hubo quien incluso se hizo tatuajes

- Las penas por viajar de los E.E.U.U. a Europa de esta manera llegaban a los 10 años de prisión.
- Ahora es posible comprar implementaciones seguras

Criptografía Pública

- El término Criptografía de Clave Pública (o criptografía pública) es intercambiable con el de Criptografía Asimétrica, denotan lo mismo, y son sinónimos.
- Aunque fue hecha pública en 1976 por Diffie, Hellman, and Merkel, en 1997 se desclasificaron documentos británicos de 1972, en donde James Ellis, Clifford Cocks and Graham Williamson del Government Communications Headquarters (GCHQ, Reino Unido), descubrieron la criptografía pública.

Criptografía simétrica vs. asimétrica

Simétrica

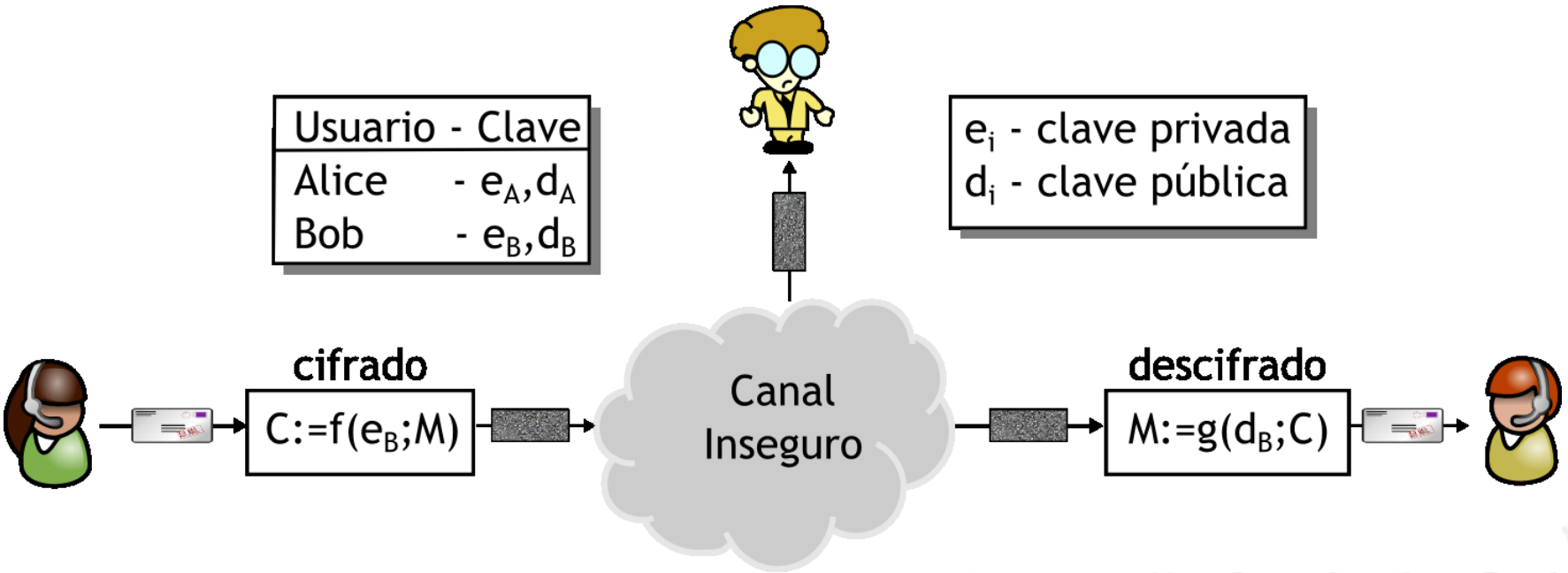
- La misma llave secreta se utiliza para cifrar y descifrar
- La función de cifrado y descifrado son muy parecidas, la misma en algunos casos
- Distribución de llaves
- Número de llaves
- Colusión

Criptografía simétrica vs. asimétrica

Asimétrica

- Sólo se publica la parte pública de la llave, cada quien la baja sobredemanda
- Que haya muchas claves es irrelevante
- Se puede proteger de la colusión
- Ahora se tiene un par de llaves: pública y privada
- La función de cifrado y descifrado son diferentes
- En algunos protocolos hay que intercambiar llaves

Esquema General de la Criptografía Pública



Usos de la criptografía de clave pública

Además de cifrar datos, la criptografía de clave pública puede utilizarse en lo siguiente:

- **Establecimiento de llaves**. Existen protocolos para la compartición de claves secretas sobre medios inseguros (para utilizar criptografía simétrica).
- **No repudiación**. Proveen no repudiación e integridad de mensajes mediante firmas digitales: RSA, DSA, ECDSA
- **Identificación**. Mediante mecanismos de desafío-respuesta junto con firmas digitales: para smart cards y teléfonos móviles.
- **Cifrado**.

El único pendiente es la autenticación de las claves públicas; para esto, se utilizan los certificados digitales.

Algoritmos relevantes

Los tres tipos relevantes en la criptografía de clave pública son:

- **Esquemas de factorización de enteros**. Basados en la dificultad de factorizar números enteros grandes
- **Esquemas de Logaritmos Discretos**. Basados en la dificultad del problema del logaritmo discreto sobre campos finitos
- **Esquemas de Curvas Elípticas**. Generalización del problema anterior a esquemas basados en curvas elípticas

Nivel de seguridad

Dado que los ataques a los sistemas criptográficos son más eficientes en los esquemas asimétricos, se define como nivel de seguridad equivalente: el número de bits en un criptosistema asimétrico que equivalen en resistencia a uno simétrico.

Esto es: esto es, si rompemos una clave simétrica de 80 bits en 1 segundo con 2^{80} computadoras, ¿cuántos bits debe de tener una clave asimétrica para que son 2^{80} computadoras también nos tardemos 1 segundo?

Niveles de seguridad

Familia	Criptosistema	Nivel de seguridad		
		128	192	256
Factorización entera	RSA	3072 bit	7680 bit	15360 bit
Logaritmo discreto	DH, DSA, Elgamal	3072 bit	7680 bit	15360 bit
Curvas elípticas	ECDH, ECDSA	256 bit	384 bit	512 bit
Clave simétrica		128 bit	192 bit	256 bit

Contenido, sección 4

Función de un solo sentido

Definición

Una función $f()$ es una función de un solo sentido si:

- $y = f(x)$ es computacionalmente sencilla, y
- $x = f^{-1}(y)$ es computacionalmente impráctica.

Ejemplos de funciones de un solo sentido

- Logaritmo discreto
  - Dados x, a, y n, es fácil calcular $y = x^a \bmod n$; sin embargo, dados y, x, y n, encontrar a es muy difícil
- Factorización
  - Dados x y y, es fácil calcular $n = xy$; sin embargo, dada n, encontrar los factores x y y es muy difícil
- Raíz cuadrada discreta
  - Dados x y n, es fácil calcular $a = x^2 \bmod n$; sin embargo, dados a y n, encontrar x es muy difícil.

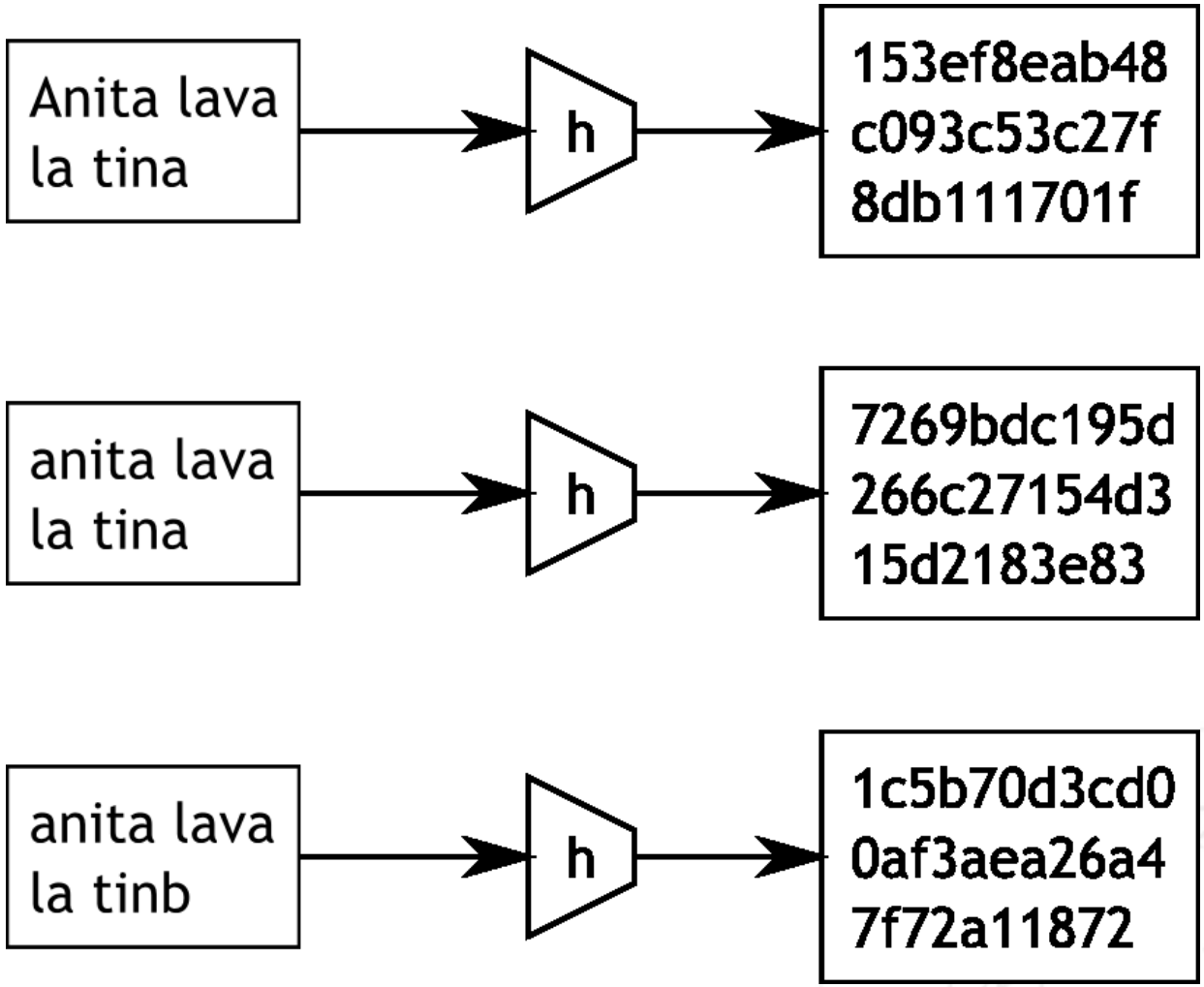
Funciones Picadillo

- Producen huellas digitales de longitud fija a partir de documentos de longitud arbitraria.
- Una pequeña variación en el texto original se ve reflejada en una huella digital totalmente diferente
  - Convierten contraseñas a salidas de longitud fija
  - Se utilizan para generar números aleatorios
  - Proveen autenticación básica a través de los MAC (Message autentication code)
  - Bloques básicos para firmas digitales.

Función picadillo - caja negra

La función picadillo recibe un texto de tamaño indefinido (normalmente se redondea el tamaño con ceros), y la salida es de un tamaño fijo.

Función picadillo - caja negra 2



Una función picadillo útil en criptografía da resúmenes totalmente diferentes incluso ante cambios menores.

Requerimientos de seguridad de las funciones picadillo

Los requerimientos de seguridad para el uso de las funciones en criptografía son los siguientes:

- Resistencia de preimagen (un solo sentido)
- Resistencia de segunda preimagen (débil resistencia a colisiones)
- Resistencia a colisiones (fuerte resistencia a colisiones)

Requerimientos de seguridad de las funciones picadillo - 2

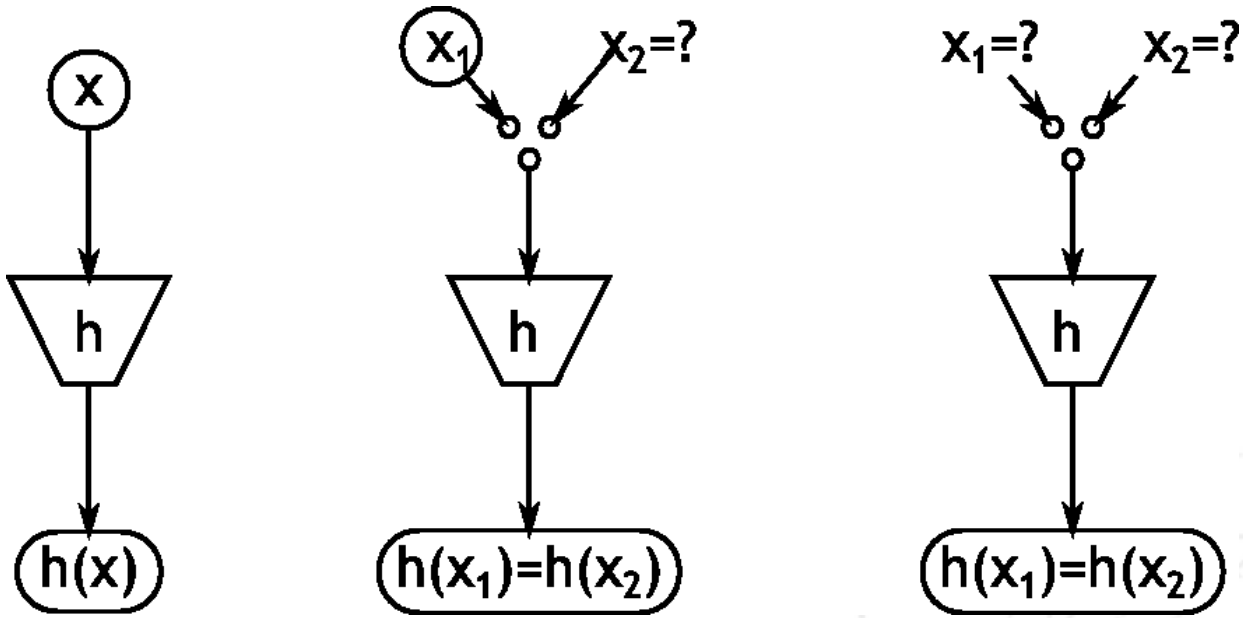
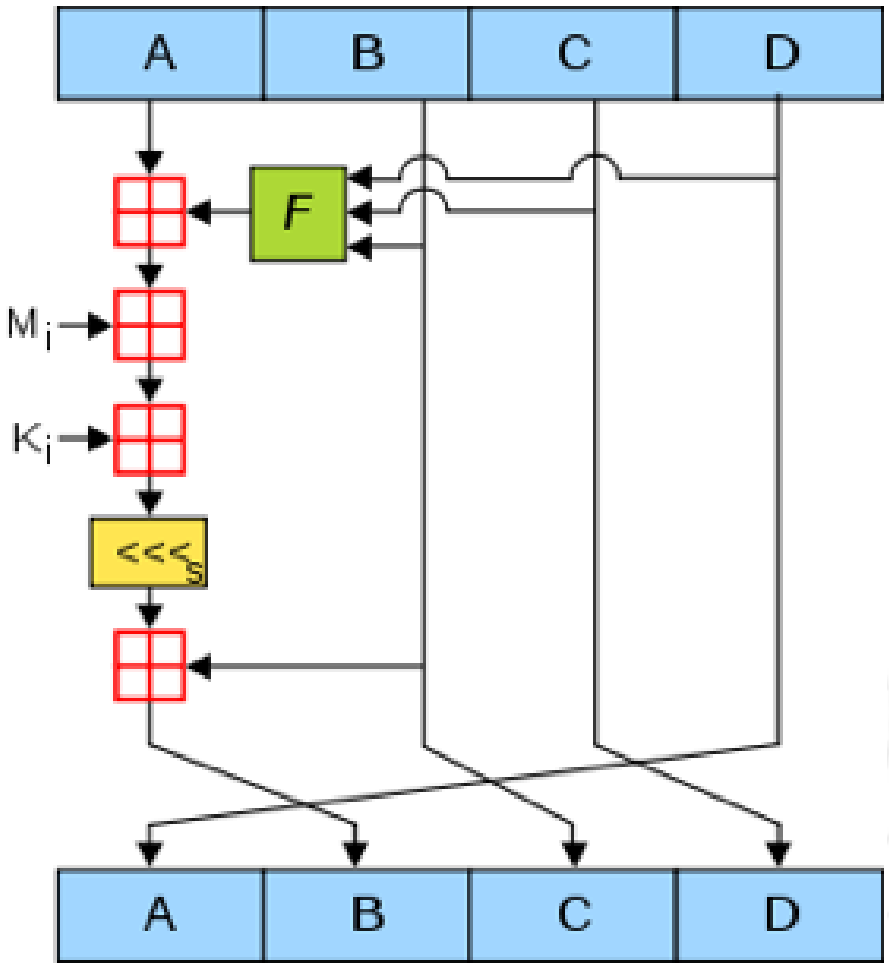


Diagrama MD5



Familias de funciones picadillo

Algoritmo		Salida	Entrada	Rondas	Colisiones
MD5		128	512	64	Sí
SHA-1		160	512	80	Aún no
SHA-2	**SHA-224**	224	512	64	No
	SHA-256	256	512	64	No
	SHA-384	384	1024	80	No
	SHA-512	512	1024	80	No

Recientemente, la función picadillo Keccak ganó el concurso para ser **SHA-3**; es una función *esponja*. Esta es la nueva recomendación del NIST.

Contenido, sección 5

Servicios Básicos de seguridad

Los servicios más importantes son:

- Confidencialidad
- Integridad
- Autenticación de mensajes
- No repudiación

Otros servicios de seguridad

Existen otros servicios opcionales que dependen de la aplicación:

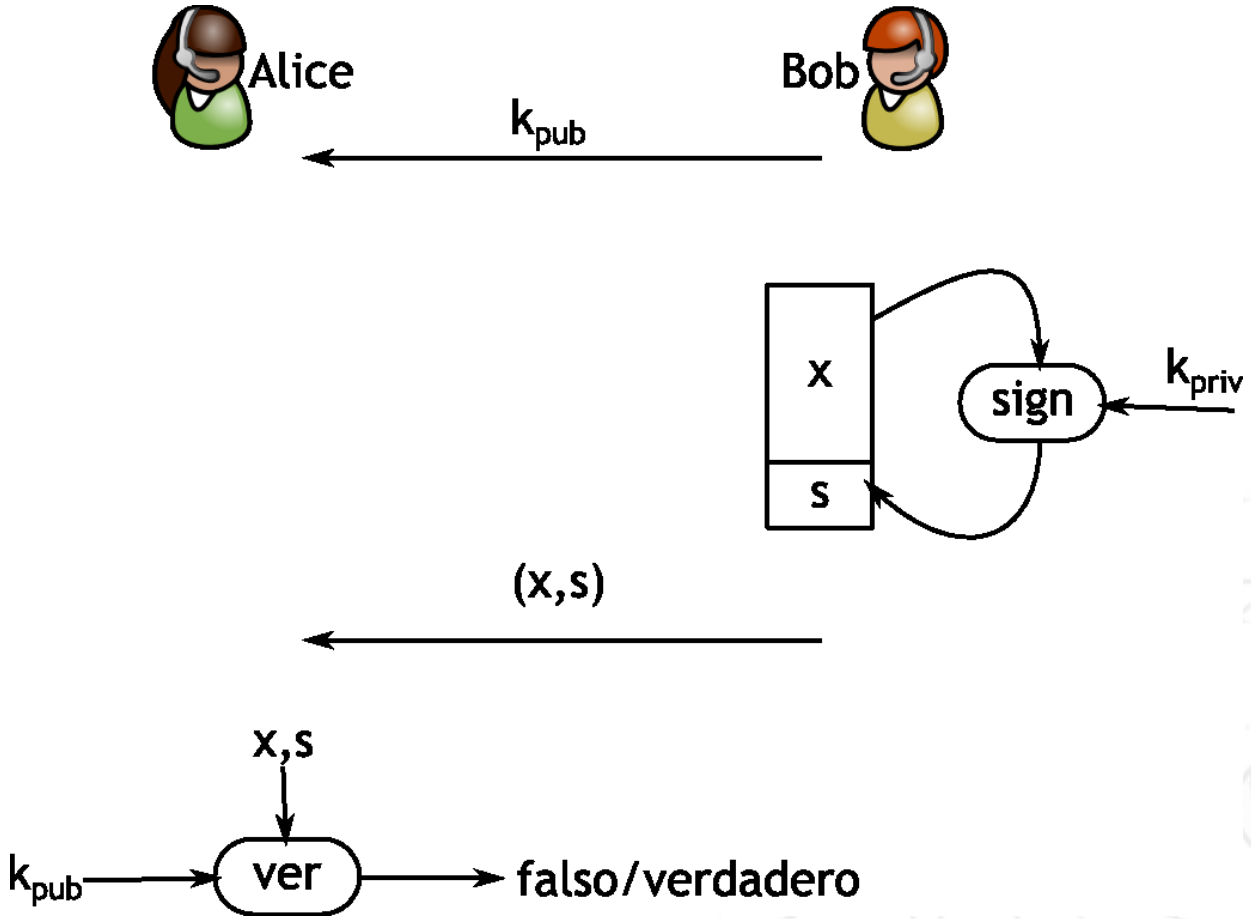
- Identificación o autenticación de entidades
- Control de acceso
- Disponibilidad
- Auditoría
- Seguridad física
- Anonimato

Firmas digitales

- La propiedad de demostrar que cierta persona generó un mensaje es crítica en muchas aplicaciones.
- En el mundo "analógico", se utilizan firmas a mano sobre papel.
- Sólo la persona que crea la firma, puede reproducirla.

Esquema

Esto es posible mediante criptografía de clave pública. El signatario firma utilizando su clave privada, el receptor utiliza la clave pública para verificar.

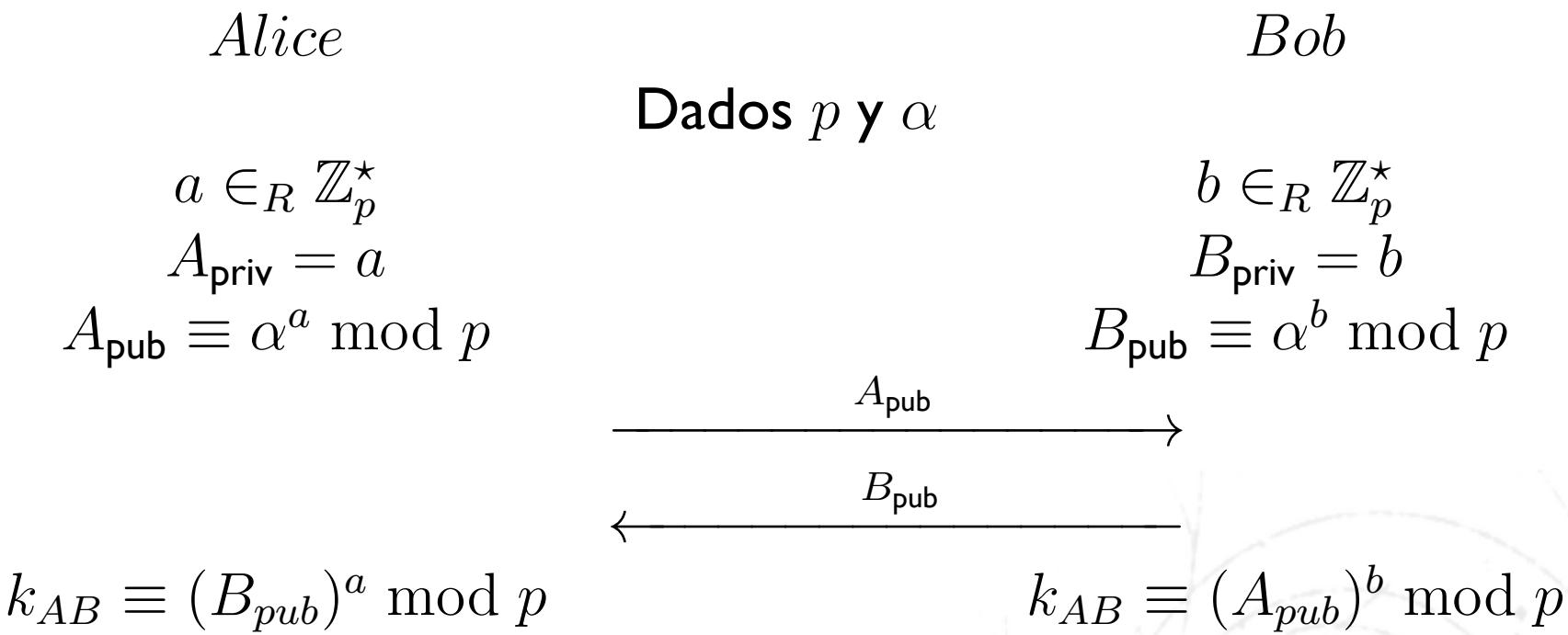


Diffie–Hellman key exchange (DHKE)

- La idea básica detrás del DHKE es que la exponenciación en $\mathbb{Z}_p^\star$, con p primo, es una función de un solo sentido, y que la exponenciación es conmutativa:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

Diagrama DHKE



RSA

Diseñado en 1977 por Ron Rivest, Adi Shamir, y Leonar Adleman.

- Sean p y q dos diferentes grandes números primos aleatorios
- El módulo n es el producto de p y q
- La función $\Phi(n) = (p-1)(q-1)$
- Seleccionamos $1 < e < \Phi(n)$, tal que el MCD$(e, \Phi(n)) = 1$; $e = 2^{16} + 1$ típicamente
- Se calcula $d \equiv e^{-1} \bmod \Phi(n)$

La clave pública es e, y n. La clave privada es d, y los primos p y q.

Cifrado y descifrado RSA

Dado un mensaje $M < n$

- **Cifrado**. $C = M^e \bmod n$
- **Descifrado**. $M = C^d \bmod n$

Para mensajes más largos, se utiliza un modo de operación, como los vistos anteriormente.

Ejemplo

- $p = 11, q = 13$
- $n = p \cdot q = 11 \cdot 13 = 143$
- $\Phi(n) = (p-1)(q-1) = 10 \cdot 12 = 120$
- $\mathsf{MCD}(e, \Phi(n)) = \mathsf{MCD}(e, 120) = 1$; $e = 17$
- $d = e^{-1} \bmod \Phi(n) = 17^{-1} \bmod 120 = 113$

- Clave pública $= (e, n) = (17, 143)$
- Clave privada $= (d, p, q) = (113, 11, 23)$

... ejemplo

- Mensaje $M = 50$
- **Cifrado:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$
- **Descifrado:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

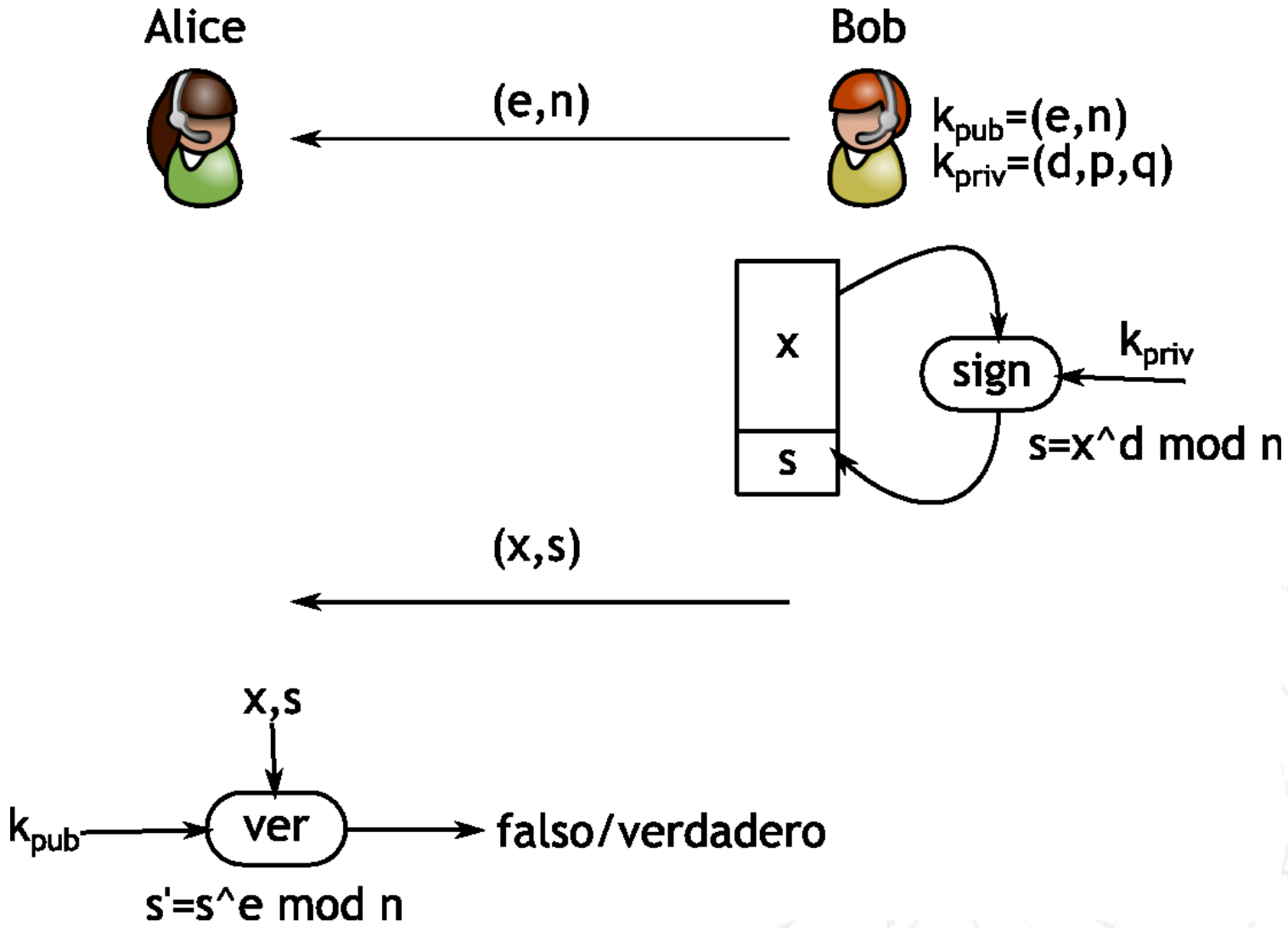
... ejemplo

- Mensaje $M = 50$
- **Cifrado:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$
- **Descifrado:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

parece sencillo, sin embargo, observe el 85^{113}, ¿qué pasaría con números grandes? Recuerde el tamaño en la tabla de Niveles de Seguridad.

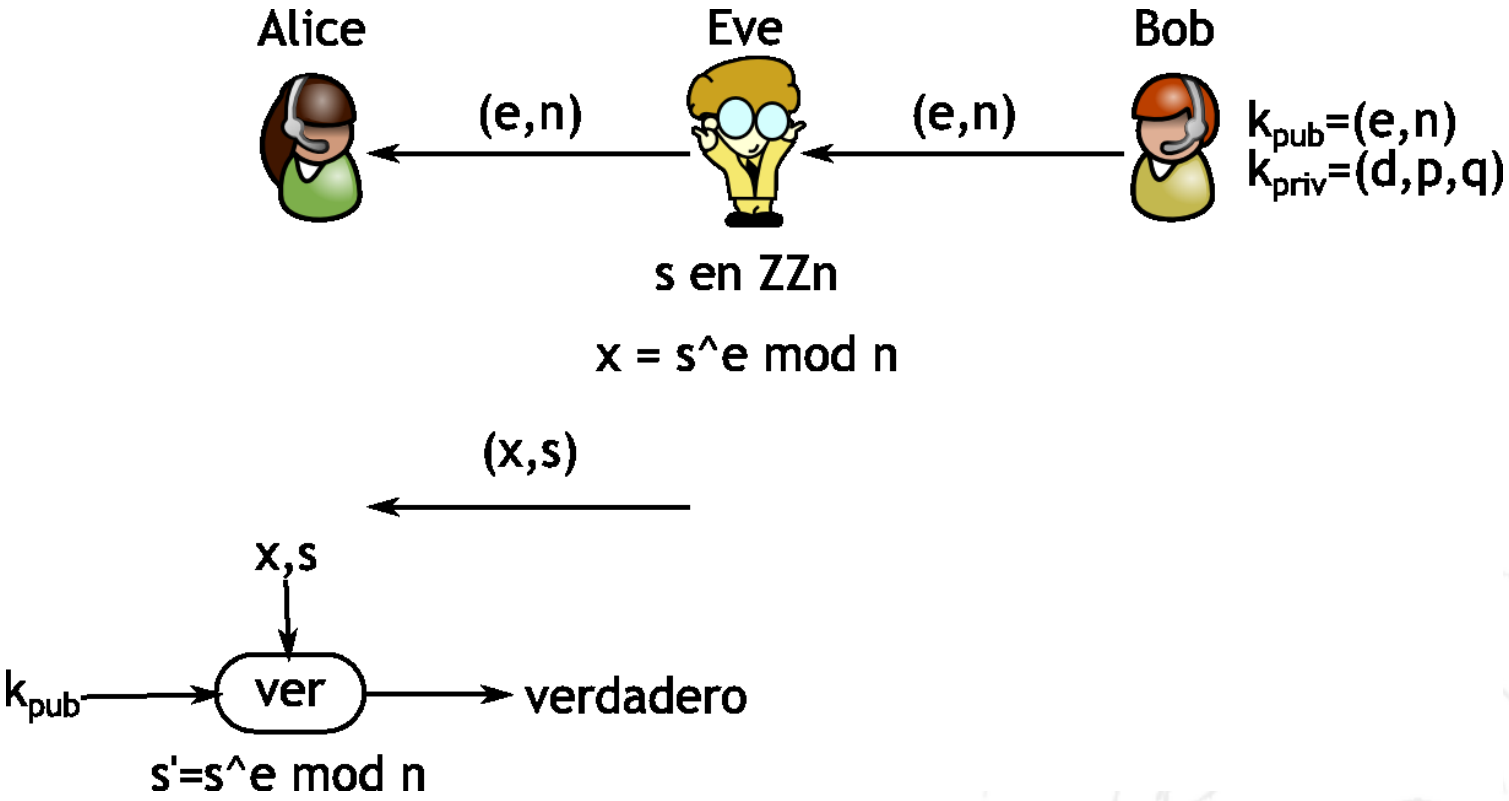
Firma RSA

Firma RSA básica



Ataque a Firma RSA

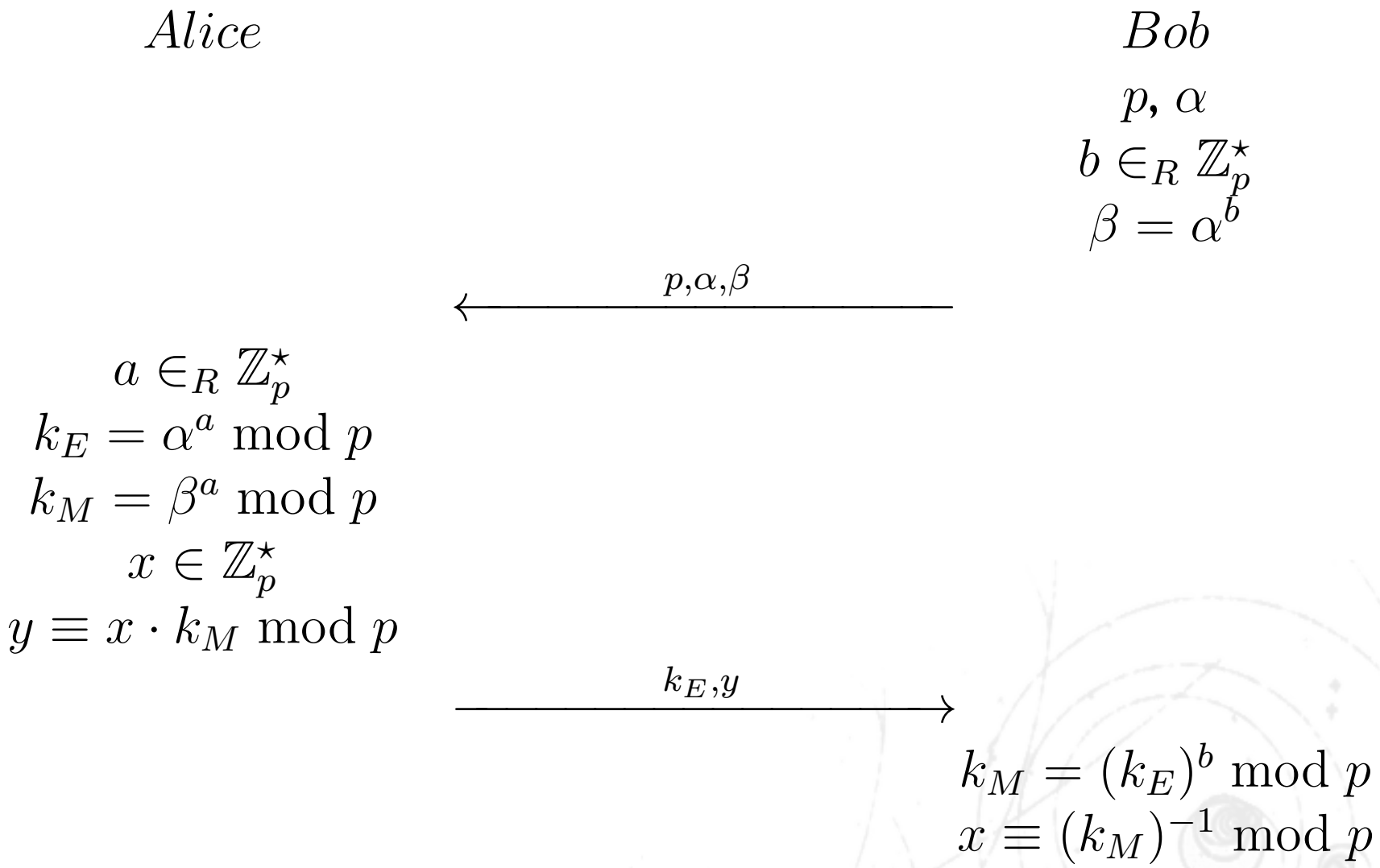
Sobre validación de firmas ficticias



ElGamal

- El cifrado Elgamal fue propuesto por Taher Elgamal en 1985.
- Es una extensión del intercambio de llaves de Diffie-Hellman (DHKE)

Cifrado Elgamal



La llave es **efímera**, se tiene que generar en cada transmisión.

Firma Elgamal

- Generación de llaves:
  - Generar un primo p
  - Encontrar un elemento $\alpha \in \mathbb{Z}_p^\star$
  - Seleccionar un elemento aleatorio d, con $2 < d < p - 2$
  - Calcular $\beta = \alpha^d \bmod p$

Firma Elgamal de mensaje

- Firma de mensaje:
  - Dado un mensaje M
  - Seleccione una llave efímera k_E, con $0 < k_E < p - 2$, con MCD$(k_E, p - 1) = 1$
  - Calcule $r \equiv a^{k_E} \bmod p$
  - Calcule $s \equiv (M - d \cdot r)k_E^{-1} \bmod p - 1$

- La firma de M es (r, s)

Verificación de Firma Elgamal

- Verificación de firma:
  - Calcule $t \equiv \beta^r \cdot r^s \bmod p$

- Si $t \equiv \alpha^x \bmod pq$, la firma verificó.

Ejemplo, M a firmar

$$p = 29,\ \alpha = 2$$
$$d = 12$$
$$\beta = \alpha^d \equiv 7 \bmod 29$$

$$\xleftarrow{k_{\text{pub}}(p,\alpha,\beta)=(29,2,7)}$$

$$k_E = 5\ (5, 28) = 1,\ x = 26$$
$$r = 2^5 \equiv 3 \bmod 29$$
$$s = -10 \cdot 7 \equiv 26 \bmod 29$$

$$\xleftarrow{(26,(3,26))}$$

$$t = 7^3 \cdot 3^{26} \equiv 22 \bmod 29$$
$$\alpha^x \equiv 2^26 \equiv 22 \bmod 29$$
$$t \equiv \alpha^x \bmod 29 \Rightarrow \text{OK}$$

Firma DSA

La firma estándar DSA contiene los siguientes pasos:

- Generación de claves:
  - Generar un primo p, con $2^{1023} < p < 2^{1024}$
  - Encontrar un primo q divisor de p, con $2^{159} < q < 2^{160}$
  - Encontrar un elementos α, cuyo orden sea igual a q
  - Seleccionar un elemento aleatorio d, con $1 < d < q$
  - Calcular $\beta = \alpha^d \bmod p$

- Las claves son:
  - Pública: (p, q, α, β)
  - Privada: d

Firma DSA de mensaje

- Firma de mensaje:
  - Dado un mensaje M
  - Seleccione una llave efímera k_E, con $0 < k_E < q$
  - Calcule $r \equiv (a^{k_E} \bmod p) \bmod q$
  - Calcule $s \equiv (SHA(M) + d \cdot r)k_E^{-1} \bmod q$

- La firma de M es (r, s)

Verificación de Firma DSA

- Verificación de firma:
  - Calcule $w \equiv s^{-1} \bmod q$
  - Calcule $u_1 \equiv w \cdot SHA(M) \bmod q$
  - Calcule $u_2 \equiv w \cdot r \bmod q$
  - Calcule $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$

- Si $v \equiv r \bmod q$, la firma verificó.

Ejemplo, M a firmar

$$p = 59,\ q = 29$$
$$\alpha = 3,\ d = 7$$
$$\beta = \alpha^d \equiv 4 \bmod 59$$

$$\xleftarrow{k_{\text{pub}}(p,q,\alpha,\beta)=(59,29,3,4)}$$

$$k_E = 10$$
$$r = (3^{10} \bmod 59) \equiv 20 \text{ mo}$$
$$s = (26 + 7 \cdot 20) \cdot 3 \equiv 5 \text{ mo}$$

$$\xleftarrow{(M,(r,s))}$$

$$w = 5^{-1} \equiv 6 \bmod 29$$
$$u_1 = 6 \cdot 26 \equiv 11 \bmod 29$$
$$u_2 = 6 \cdot 20 \equiv 4 \bmod 29$$
$$v = 20 \equiv (3^{11} \cdot 4^4 \bmod 59) \bmod 29$$
$$v \equiv r \bmod 29 \Rightarrow \text{OK}$$

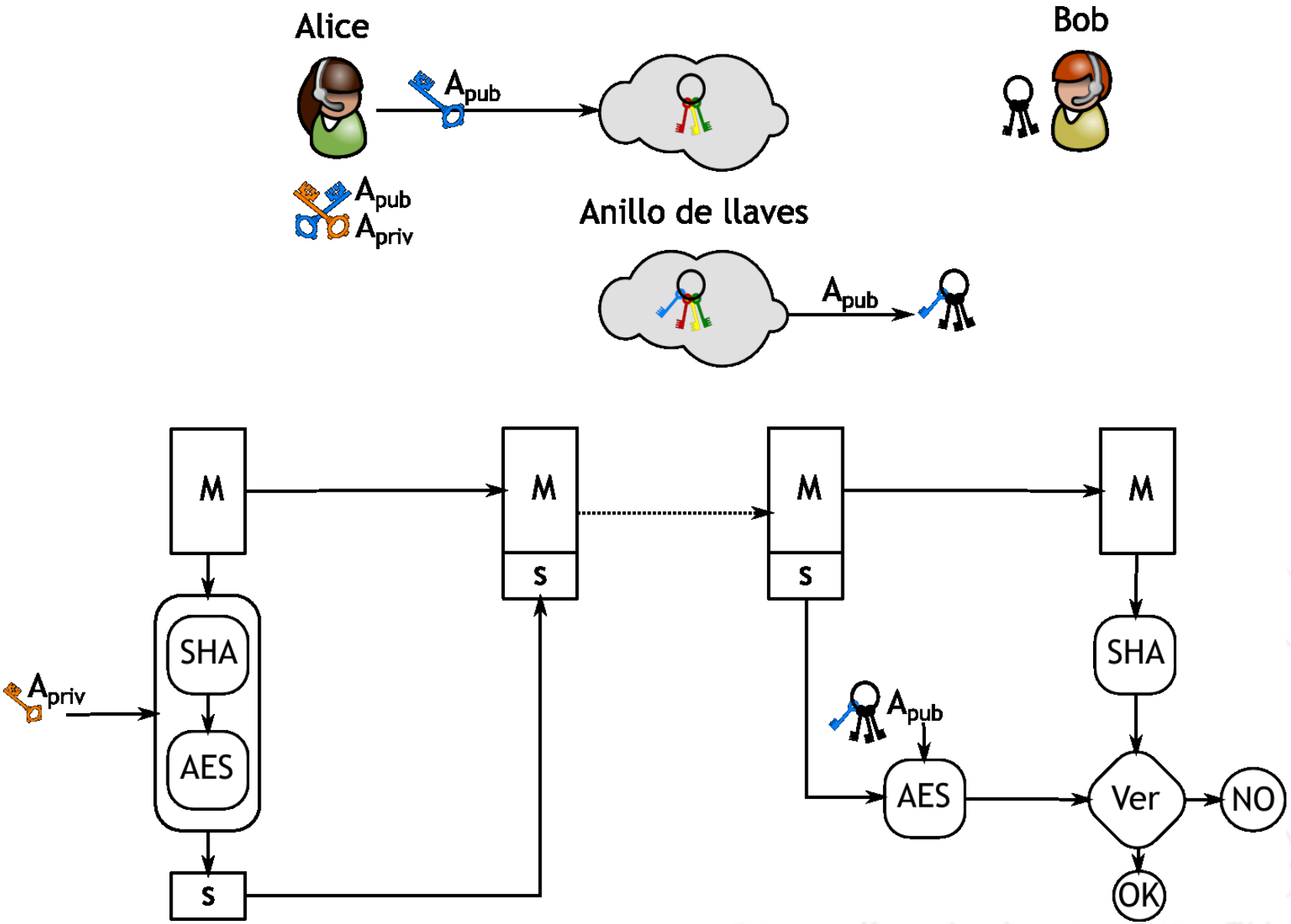
Valores de p y q, Firma DSA

Nivel de Seguridad	p	q	Tamaño de firma
80	1024	160	320
112	2048	224	448
128	3072	256	512
192	7680	384	768
256	15535	512	1024

Otras firmas

- Un método particularmente eficiente en aplicaciones en donde el consumo eléctrico o el espacio es restringido son las firmas cortas basadas en curvas elípticas.
- La criptografía basada en curvas elípticas es un tipo de criptografía de clave pública que requiere menos espacio de la clave que sus contrapartes.
- La criptografía de curvas elípticas es mucho más elaborada, pero permite la implementación eficiente de protocolos de seguridad más interesantes, o a un menor costo.

Firma Digital



Formalizando un protocolo

- Los protocolos de seguridad contienen generalmente varios pasos para definirlos *formalmente*.

- Los pasos incluyen:
  - Setup
  - Key generation
  - Encryption
  - Decryption
  - Key delegation
  - Key revocation
  - . . .

Los pasos dependen del protocolo en particular

Certificados digitales

Es un documento que mediante una firma digital de una entidad de confianza, previamente almacenada en el equipo solicitante, asocia una clave pública con una identidad: nombre de la persona, organización, dirección, etc.

El certificado sirve para garantizar que una clave pública en particular pertenece al que dice ser el poseedor de la contraparte privada.

Los certificados son emitidos por una entidad de confianza, una Autoridad Certificadora.

Certificados digitales

Es un documento que mediante una firma digital de una entidad de confianza, previamente almacenada en el equipo solicitante, asocia una clave pública con una identidad: nombre de la persona, organización, dirección, etc.

El certificado sirve para garantizar que una clave pública en particular pertenece al que dice ser el poseedor de la contraparte privada.

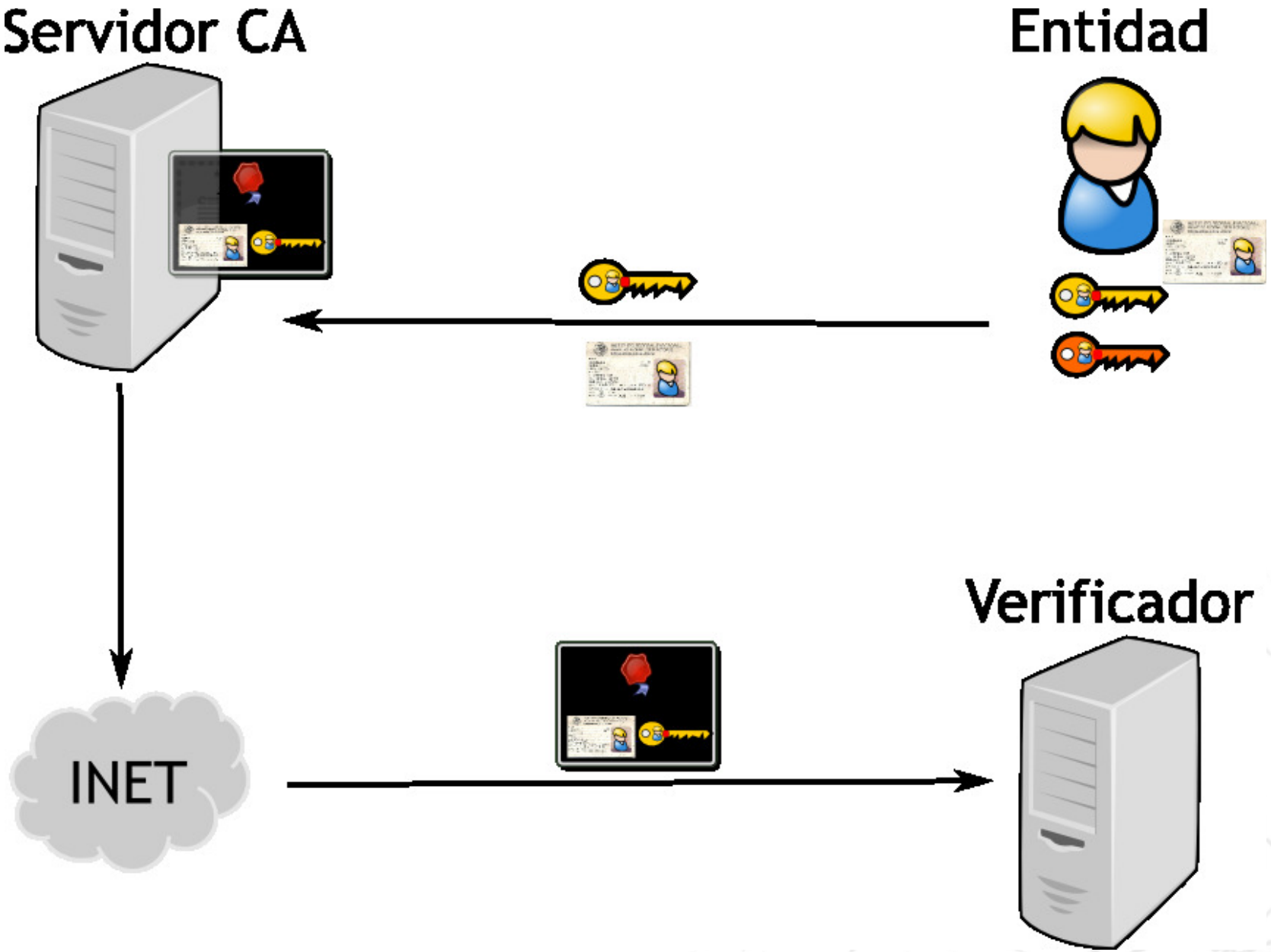
Los certificados son emitidos por una entidad de confianza, una Autoridad Certificadora... aunque en la práctica la relación de confianza se delega a Mozilla, Microsoft, Apple.

Responsabilidades de una CA

Las responsabilidades básicas son:

- Generación de llaves (Intercambio seguro)
- Emisión de Certificados (¿Qué son?)
- Emisión de CRL's (¿Para qué sirven?)

Certificados



Contenido de un certificado

El estándar X.509 establece el formato ASN1 para los certificados digitales, que contienen:

- Número serial
- Sujeto: Persona o entidad identificada
- Algoritmo de firma digital
- Firma digital
- Emisor
- Inicio validez
- Fin validez
- Propósito de la llave: cifrado, firma digital, firma de certificados
- LLave pública
- Algoritmo de huella digital
- Huella digital

Ejemplo

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert High Assurance CA-3
        Validity
            Not Before: Oct  8 00:00:00 2012 GMT
            Not After : Dec 16 12:00:00 2015 GMT
        Subject: C=MX, ST=Distrito Federal, L=Mexico, O=Centro de Investigacion... CN=*.cinvestav.mx
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:57:6c:
                    05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:bd:6c:
                    c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:82:d9:
                    84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:73:5c:
                    be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:74:84:
                    46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:ab:59:
                    8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a6:78:
                    55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8c:06:
                    1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:89:b0:
                    88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0d:f8:
                    c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d3:12:
                    0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:01:c9:
                    65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:fd:33:
                    90:12:5c:1c:46:2b:44:24:04:77:44:82:98:26:93:
                    d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:94:a9:
                    2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6a:89:
                    2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7b:89:
                    b6:a7
```

```
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:F7

    X509v3 Subject Key Identifier:
        37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
    X509v3 Subject Alternative Name:
        DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:www.tamps.cinvestav.mx,
         DNS:webmail.tamps.cinvestav.mx, DNS:noc.tamps.cinvestav.mx
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl3.digicert.com/ca3-g15.crl

        Full Name:
          URI:http://crl4.digicert.com/ca3-g15.crl

    X509v3 Certificate Policies:
        Policy: 2.16.840.1.114412.1.1
          CPS: http://www.digicert.com/ssl-cps-repository.htm
          User Notice:
            Explicit Text:

    Authority Information Access:
        OCSP - URI:http://ocsp.digicert.com
        CA Issuers - URI:http://cacerts.digicert.com/DigiCertHighAssuranceCA-3.crt

    X509v3 Basic Constraints: critical
        CA:FALSE
```

```
Signature Algorithm: sha1WithRSAEncryption
     89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:e4:
     a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:8f:
     58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:35:
     57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:60:
     0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:c4:
     d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:e8:
     e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:0e:
     af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:2b:
     75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:d9:
     f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:65:
     0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:44:
     cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:50:
     bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:4b:
     43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:9a:
     d7:3d:5f:c6
```

Demo

Ejercicio de creación de certificados.

- Generación de Parámetros DSA

```
openssl dsaparam 2048 -out dsaparams.pem
```

- Generación de Llaves

```
openssl gendsa -out dsarootkey.pem dsaparams.pem
```

- Generación de certificado raíz auto-firmado

```
openssl req -newkey dsa:dsaparams.pem -keyout
  dsarootkey.pem -new -x509 -days 365 -out
  rootcert.pem
```

- Examinando el certificado

```
openssl x509 -text -in rootcert.pem | more
openssl asn1parse -in rootcert.pem | more
```

- Generando certificado para el cliente

```
openssl req -newkey dsa:dsaparams.pem -keyout
  dsakey.pem -new -days 365 -out dsareq.pem
```

- Expedición del Certificado

```
openssl x509 -days 180 -CA rootcert.pem -CAkey
  dsarootkey.pem -req -CAcreateserial -CAserial
  ca.srl -in dsareq.pem -out newcert.pem
```

- Examinando el certificado emitido

```
openssl x509 -text -in newcert.pem | more
openssl asn1parse -in newcert.pem | more
```

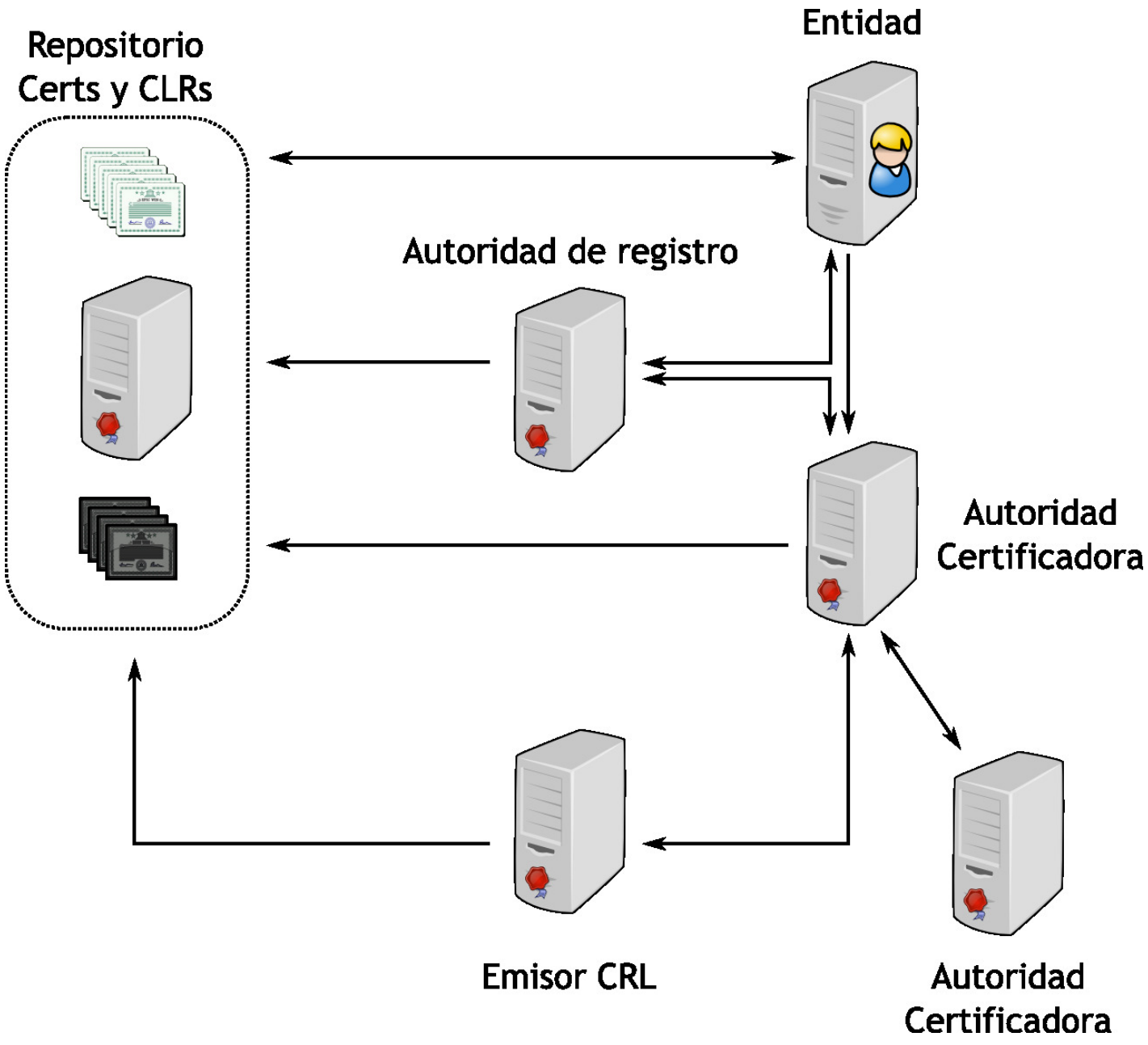
- Verificación del Certificado

```
openssl verify -CAfile rootcert.pem newcert.pem
```

PKI

- La *Infraestructura de Llave Pública (PKI)* es una combinación de software, tecnologías de cifrado, y servicios que permiten proteger la seguridad de las transacciones de información en un sistema distribuido.
- PKI integra (mas bien lo intenta) certificados digitales, criptografía de llave pública y autoridades de certificación en una arquitectura de seguridad unificada.

Diagrama PKI



Elementos en la PKI

- **Entidad final**. Término genérico para denotar a los usuarios finales o cualquier entidad que pueda ser identificada (personas, servidores, compañías, etc.) mediante un certificado digital expedido por una Autoridad Certificadora.

- **Autoridad Certificadora (AC)**. La AC es la entidad que expide los certificados digitales, así como la lista de revocación (CRL). Adicionalmente puede soportar funciones administrativas, aunque generalmente éstas son delegadas a una o varias Autoridades de Registro.

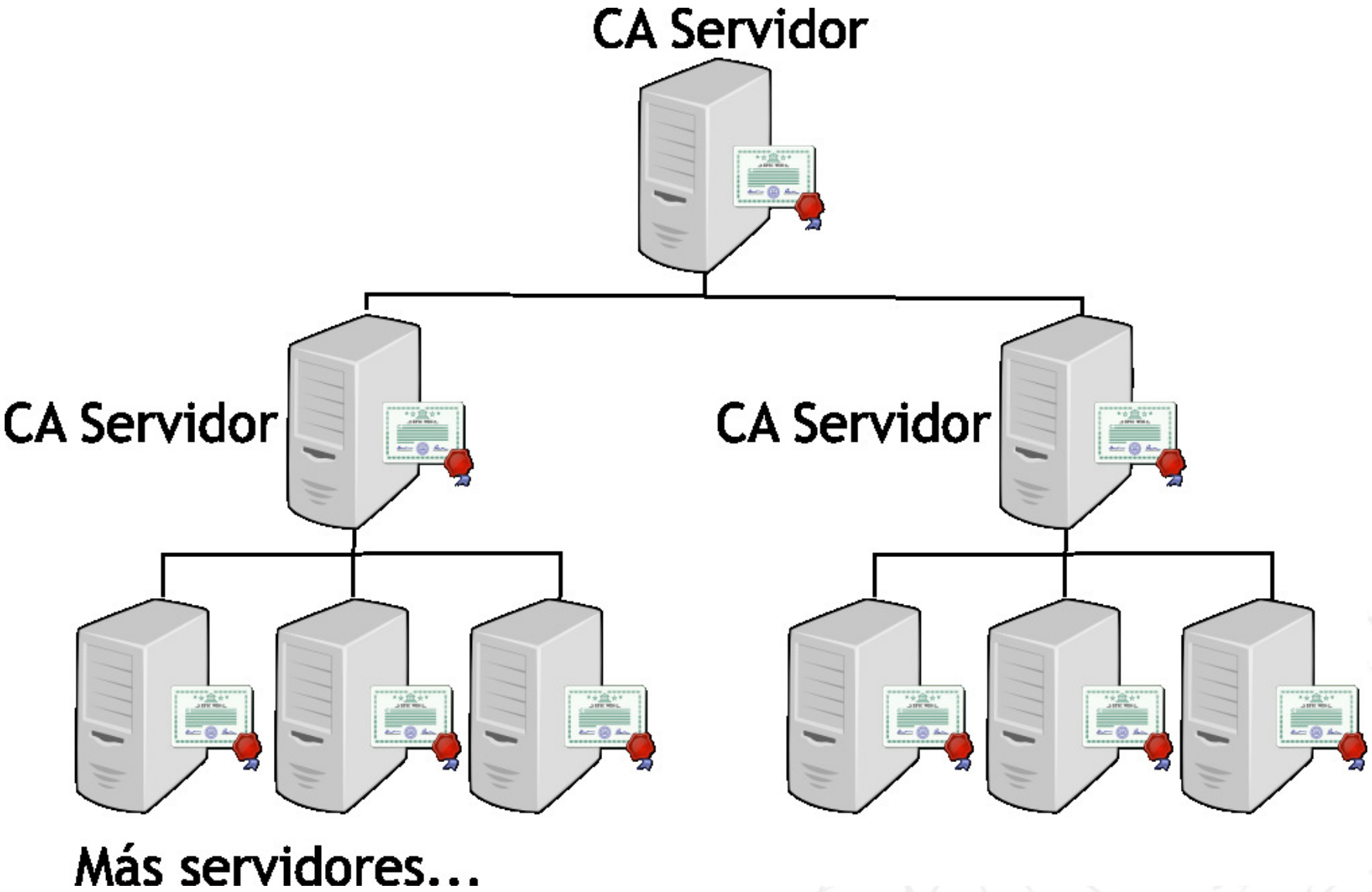
Elementos en la PKI

- **Autoridad de Registro (AR)**. Una AR es componente opcional que puede asumir funciones administrativas de la CA.

- **Repositorio**. El repositorio es el término genérico utilizado para denotar cualquier método para almacenamiento de certificados y listas de revocación (CRLs) que permita el acceso por parte de las entidades finales a dichos documentos.

- **Emisor CRL**. El emisor CRL es un componente opcional el cual puede ser utilizado por una AC para delegar las tareas de publicación de las listas de revocación.

Delegación de Autoridades Certificadoras



Contenido, sección 6

MACs

- Un Código de Autenticación de Mensajes (MAC), también conocido como un suma de verificación criptográfica o función hash con llave, es un instrumento muy utilizado.

- En términos de funcionalidad de seguridad, los MACs comparten algunas propiedades con las firmas digitales, ya que también proveen integridad y autenticación de mensajes. Sin embargo, a diferencia de las firmas digitales, los MACs son esquemas de llave simétrica, por lo que no proveen no-repudiación.

Propiedades

Las propiedades de los Códigos de Autenticación de Mensajes son:

- **Suma de verificación criptográfica**. Genera una etiqueta de autenticación criptográficamente segura para un mensaje dado.
- **Simetría**. Están basadas en llaves simétricas. Para la firma y verificación de las partes debe de compartirse una clave secreta.
- **Tamaño de mensaje arbitrario**. Aceptan mensajes de tamaño arbitrario.
- **Tamaño de salida fijo**. Generan etiquetas de tamaño fijo.
- . . .

. . .propiedades

- . . .
- **Integridad de mensaje**. Proveen integridad de mensaje: cualquier manipilación del mensaje durante la transmisión será detectado por el receptor.
- **Autenticación del mensaje**. La parte receptora se asegura del origen del mensaje.
- **No no-repudiación**. Dado que los MACs están basados en principios de criptografía simétrica, no proveen no-repudiación.

HMAC

- Una manera de hacer MACs es utilizando una función picadillo como SHA-1
- Este tipo de construcciones se les conoce como HMAC

Una manera vulnerable de construirlas es la siguiente:

- MAC de prefijo secreto: $m = MAC_k(x) = h(k||x)$
- MAC de sufijo secreto: $m = MAC_k(x) = h(x||k)$.

HMAC

- Una manera de hacer MACs es utilizando una función picadillo como SHA-1
- Este tipo de construcciones se les conoce como HMAC

Una manera vulnerable de construirlas es la siguiente:

- MAC de prefijo secreto: $m = MAC_k(x) = h(k||x)$
- MAC de sufijo secreto: $m = MAC_k(x) = h(x||k)$.

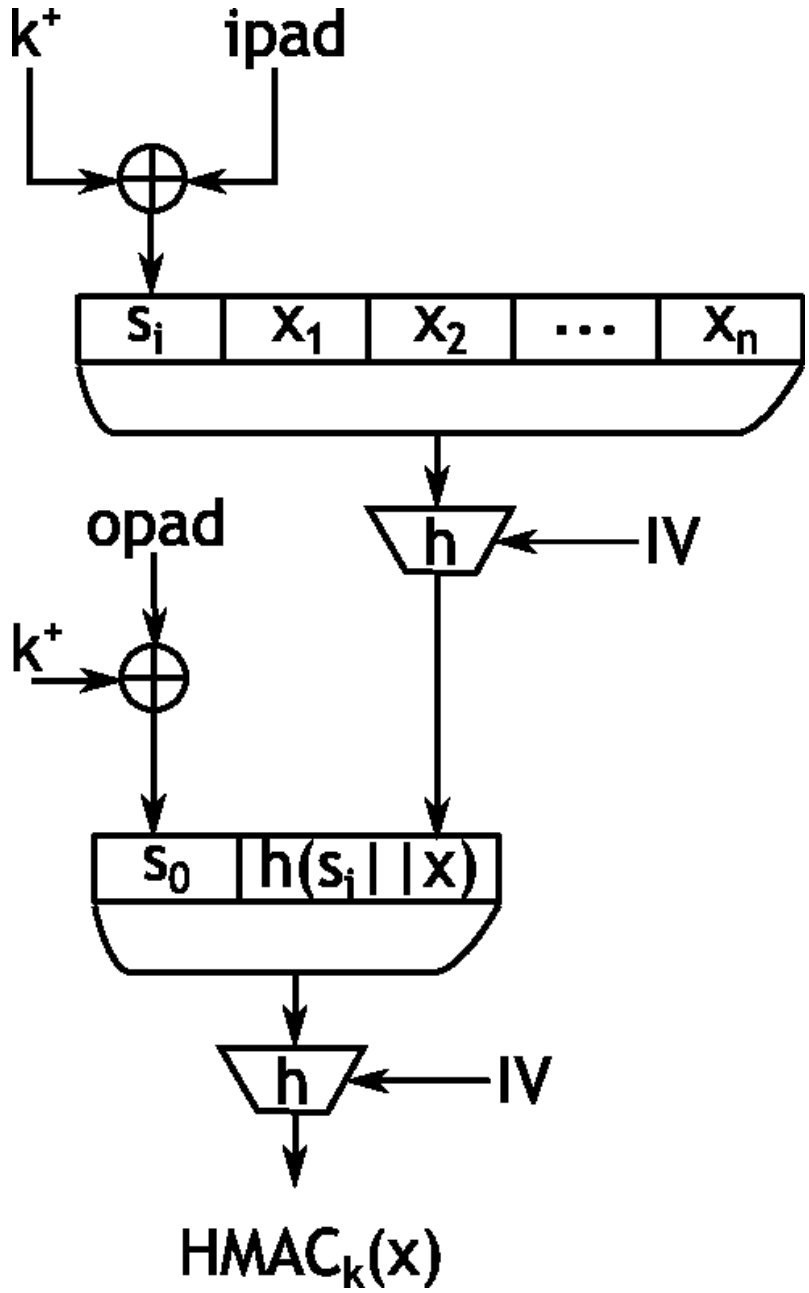
Ambas construcciones son vulnerables.

Ataques

- Prefijo secreto: dada la iteratividad del MAC, se pueden agregar mensajes al final de la secuencia

- Sufijo secreto: dada una coalisión en la función hash, es posible producir otro mensaje diferente que sea válido.

Diagrama HMAC



Contenido, sección 7

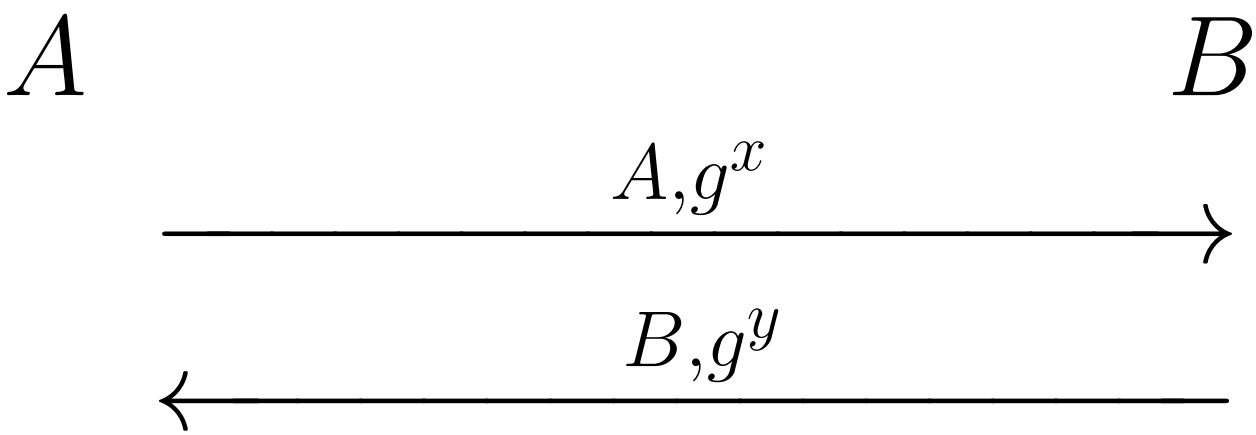
IPSec

- Seguridad de transporte en la capa IP
- Provee tráfico seguro entre dos sistemas IP
- Ofrece servicios de seguridad para los paquetes IP
- Generación y mantenimiento de la Asociación de Seguridad
- Independiente de la aplicación (software)

Para establecer un canal, primero hay que intercambiar claves (simétricas). . .

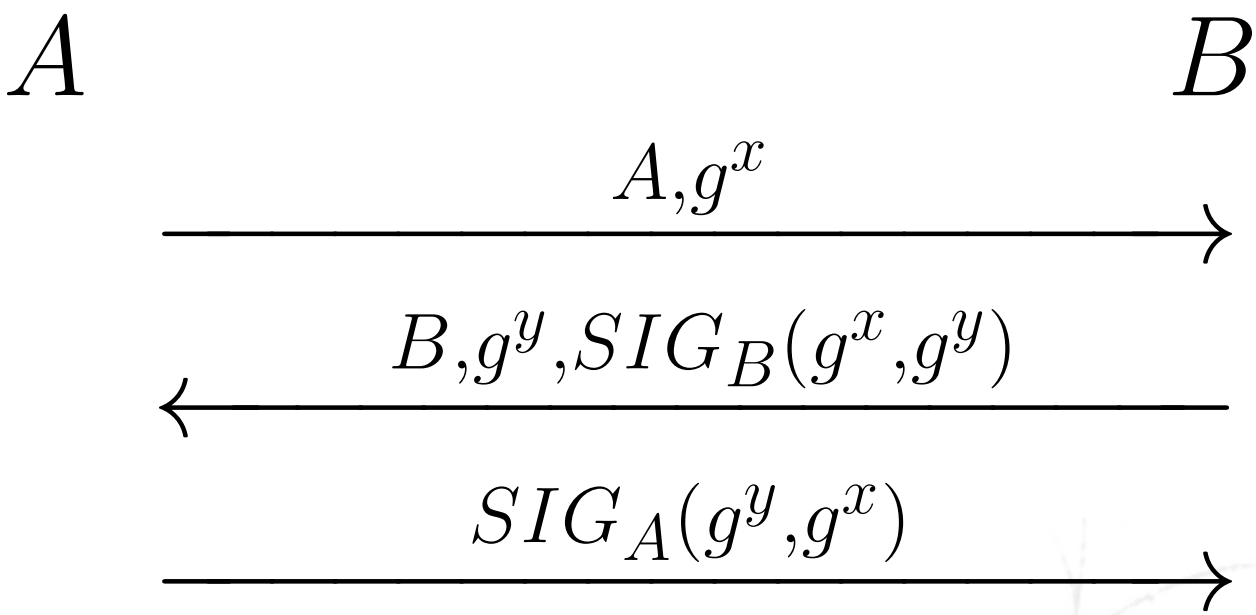
Diffie-Hellman

DH'76 - Diffie-Hellman Exchange

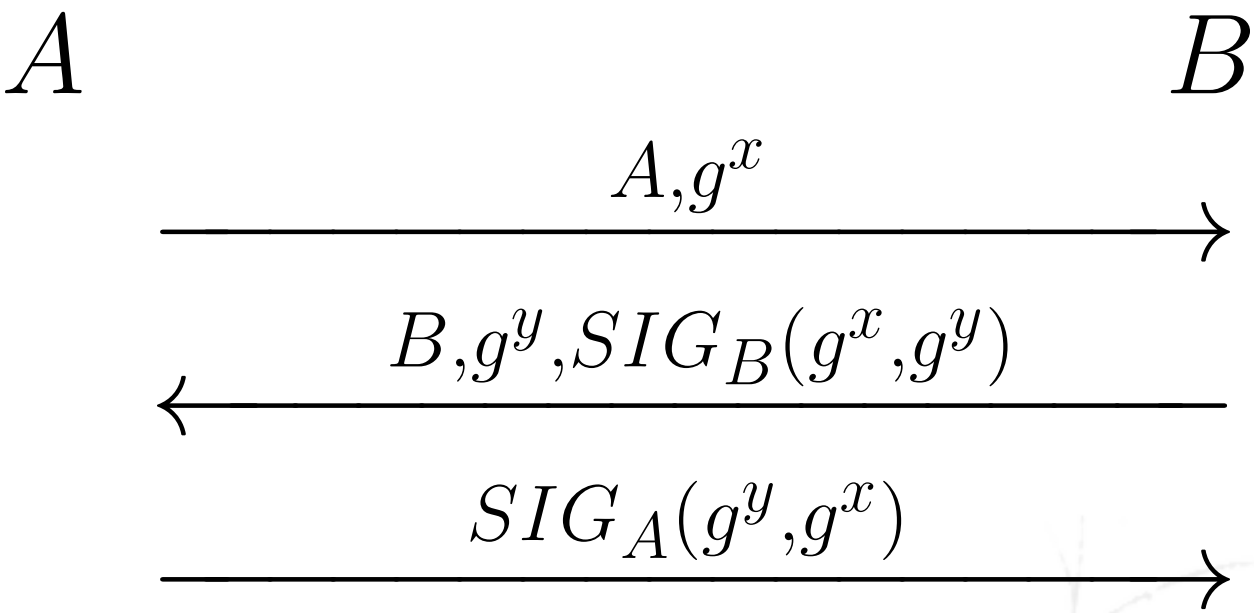


- Ambas partes pueden calcular la llave secreta $K = g^{xy}$
- Dados g^x, y g^y, g^{xy} parece un elemento aleatorio
- Abierto a un ataque M-I-T-M en un ambiente sin autenticación

DH Autenticado básico (BADH)

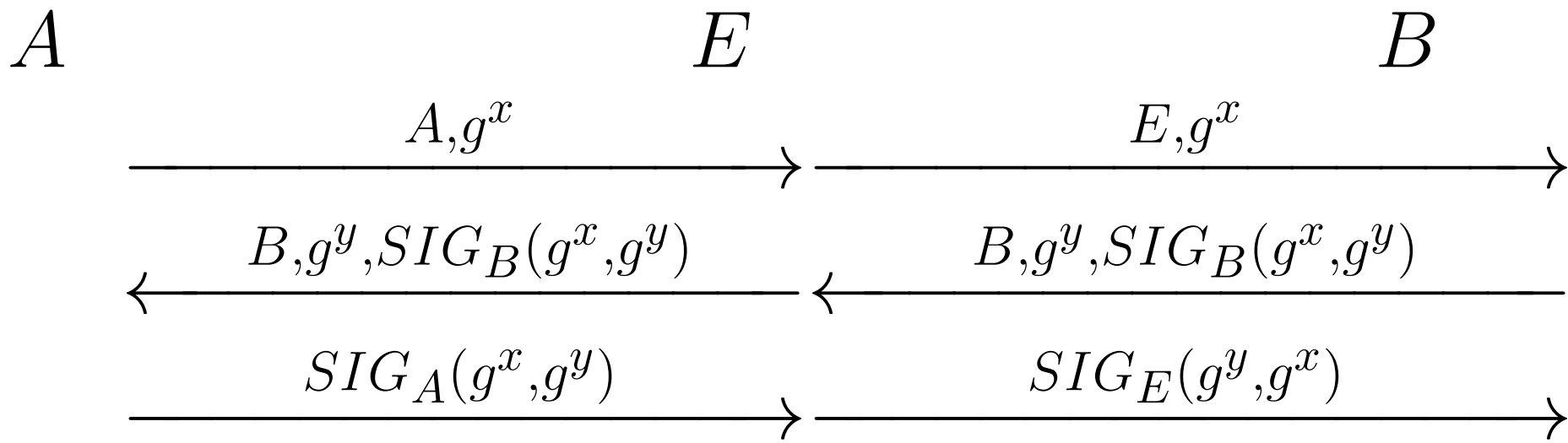


DH Autenticado básico (BADH)

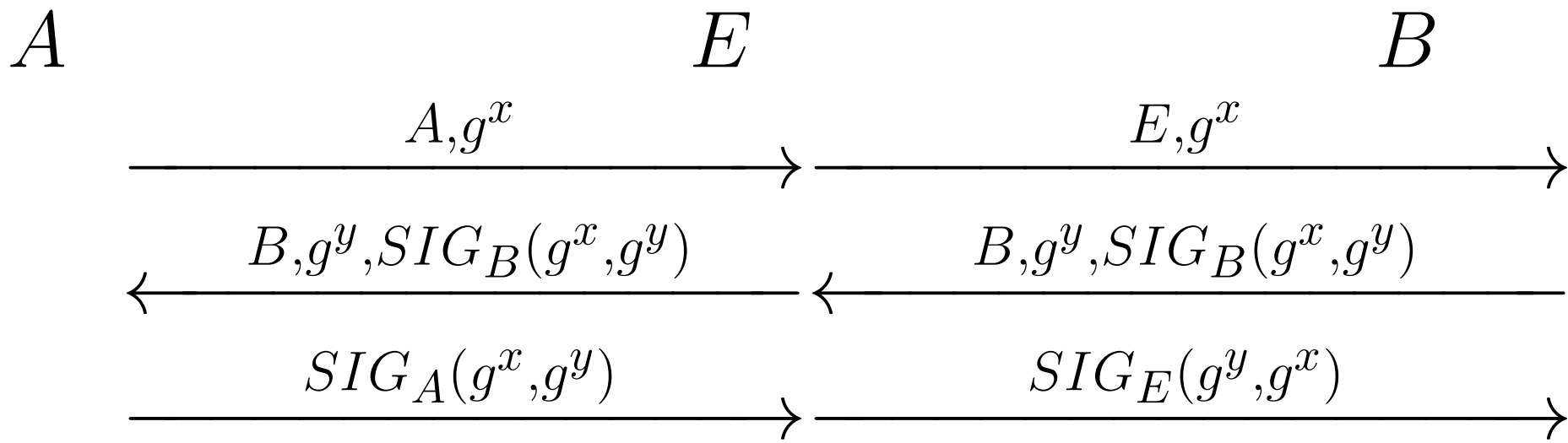


sin embargo. . .

Ataque a BADH



Ataque a BADH

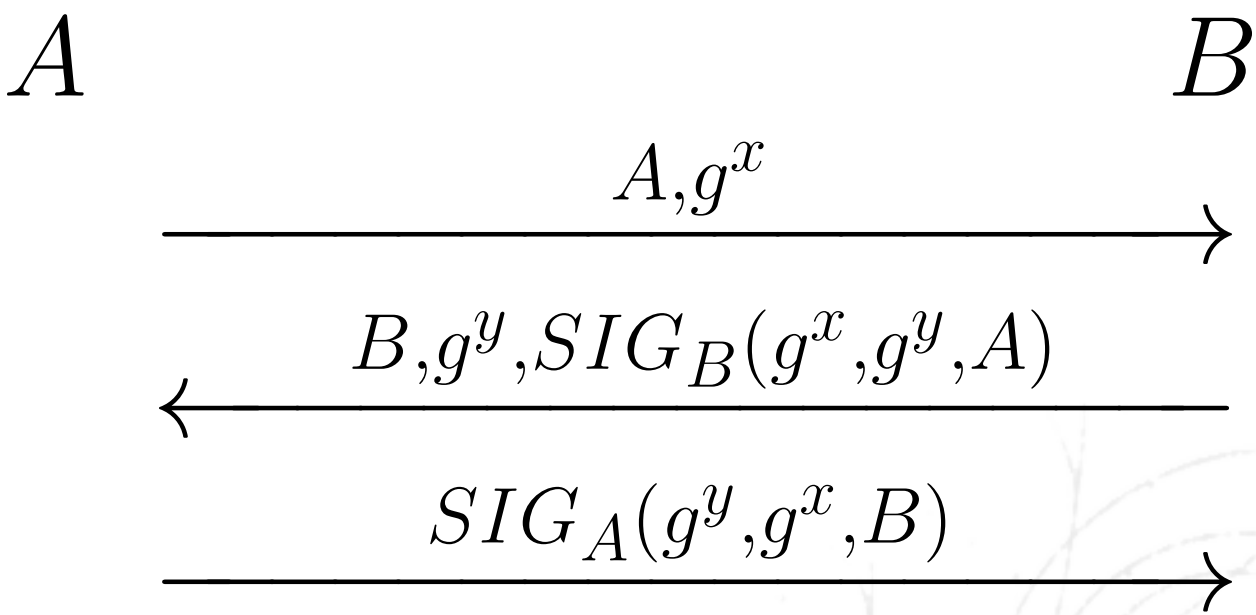


Aunque E no conoce $K = g^{xy}$, B recibe lo que le envíe E pensando que es A.

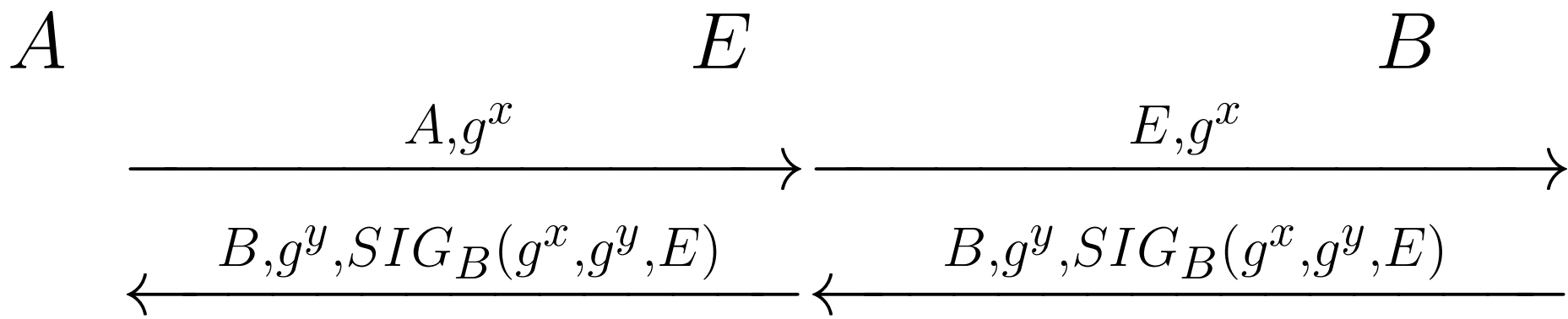
- A recibe mensajes de B correctos, E no puede hacer nada.
- B recibe mensajes de A con identificador de E.

Solución ISO-9796. . .

Incluir la identidad del receptor dentro de la firma:



Ataque. . .



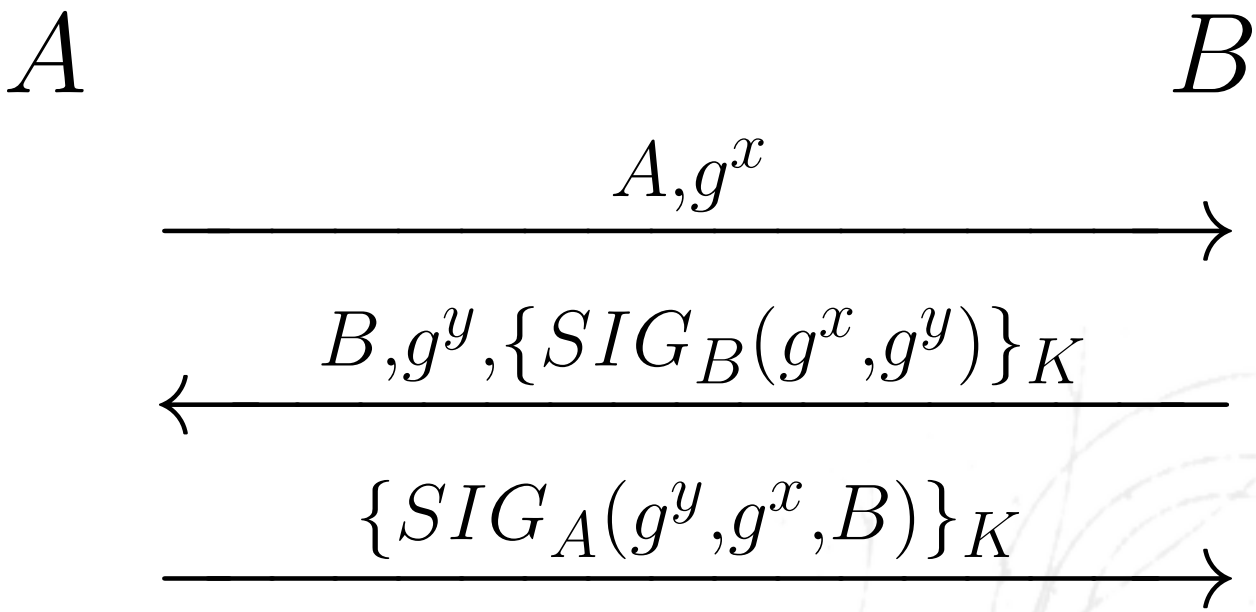
- A: ¡Ajá! B se está comunicando con E, no conmigo
- E no puede producir $SIG_B(g^x, g^y, A)$

¿Es seguro?

- Es técnicamente seguro
- No sirve para protección de identidades:
  - B necesita saber la identidad de A antes de autenticarse, lo mismo para A
  - Privacidad: hay evidencia firmada de la comunicación
  - Hacer que cada parte firme su identidad resuelve el problema de la privacidad, pero volvería el protocolo inseguro (ataque M-I-T-M).

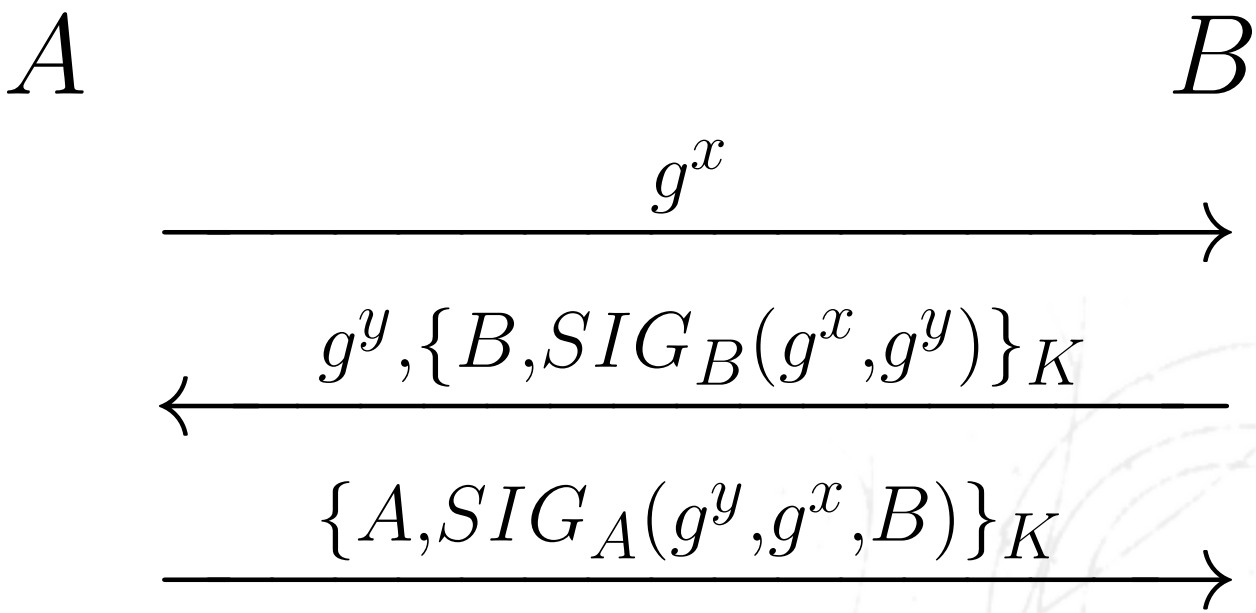
Otra solución: STS

Probar conocimiento de la $K = g^{xy}$ utilizando un cifrado simétrico:



¿Es seguro?

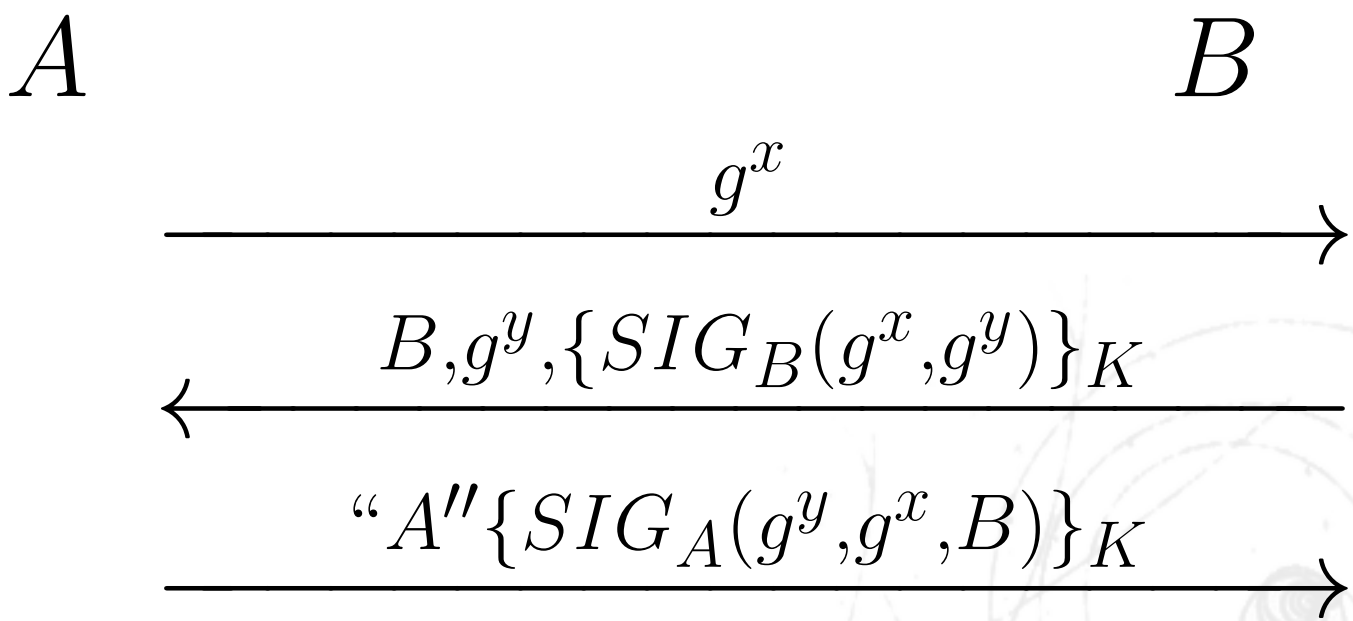
- Es seguro
- se puede extender para proteger a las identidades de la transacción



. . .¿es seguro?

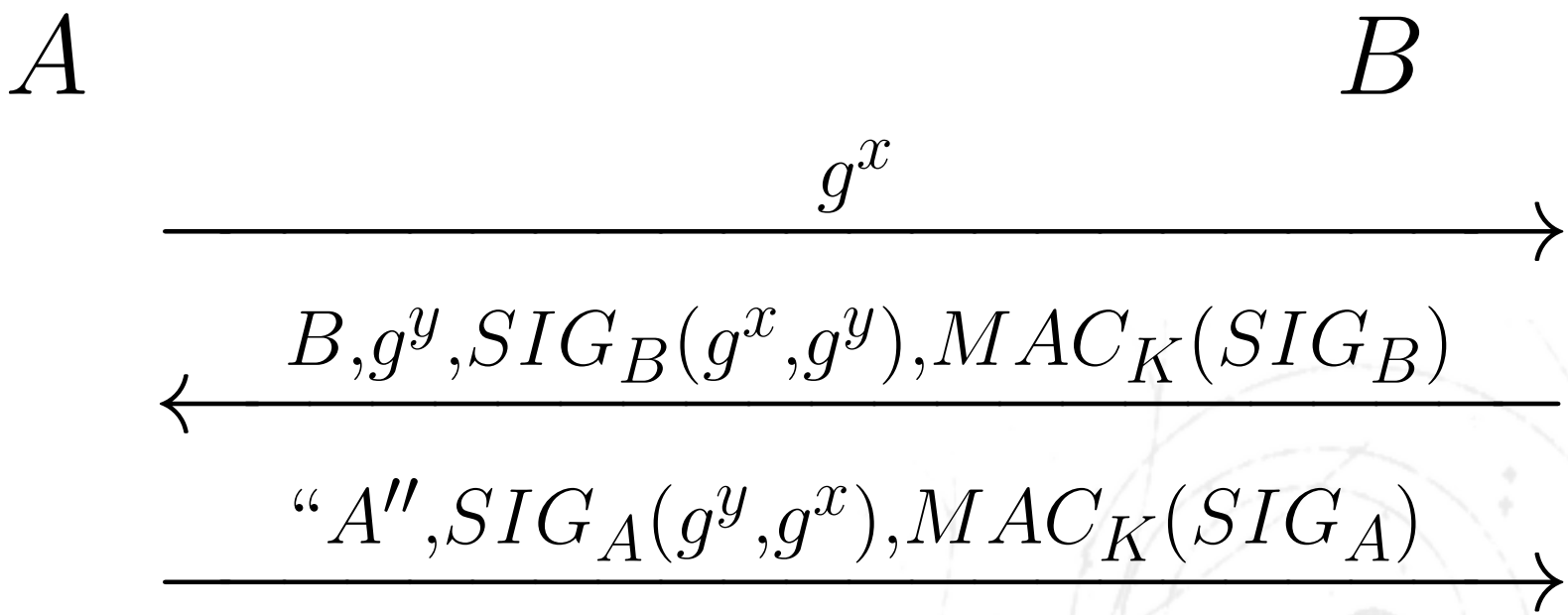
El cifrado, no es la función correcta para provar conocimiento de una llave

- alguien podría registrar como suya la llave pública de otra persona, y montar un ataque I-M (y sin conocer $K = g^{xy}$)



STS con MAC

Algunos esquemas verifican la identididad, pero no la posesión de la llave privada (PoP)

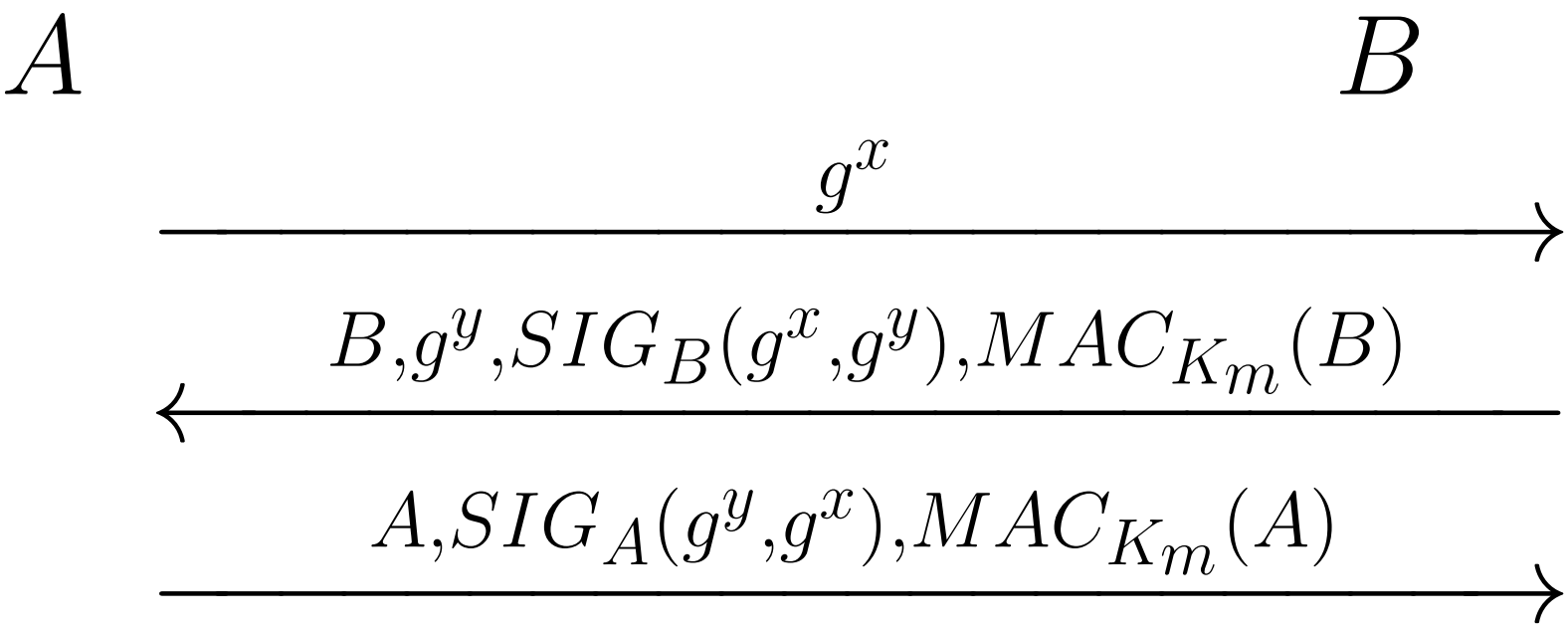


¿Qué pasa?

- El punto es que probar que se conozca la $K = g^{xy}$ no es el problema.
- Lo que se requiere es:
  - asociar la llave K con las identidades

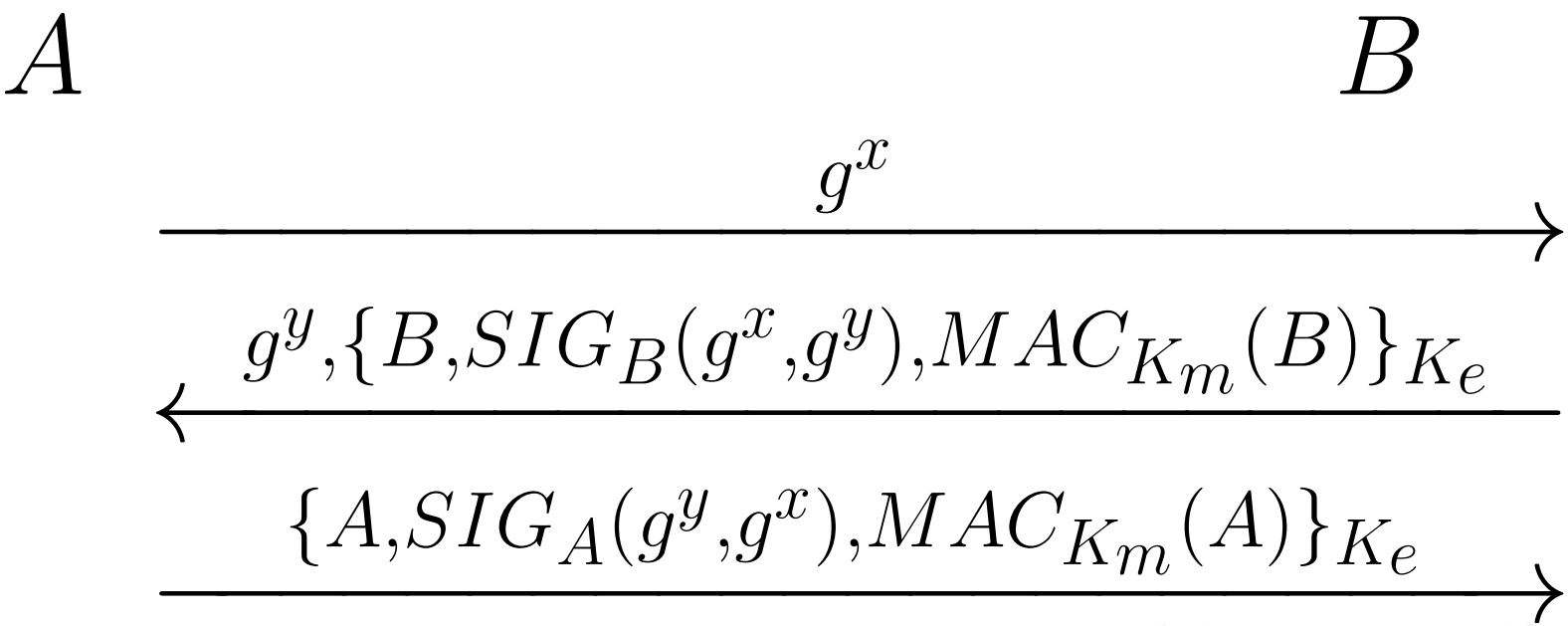
- La solución sería:
  - SIGn-and-MAC su identidad

SIGMA básico



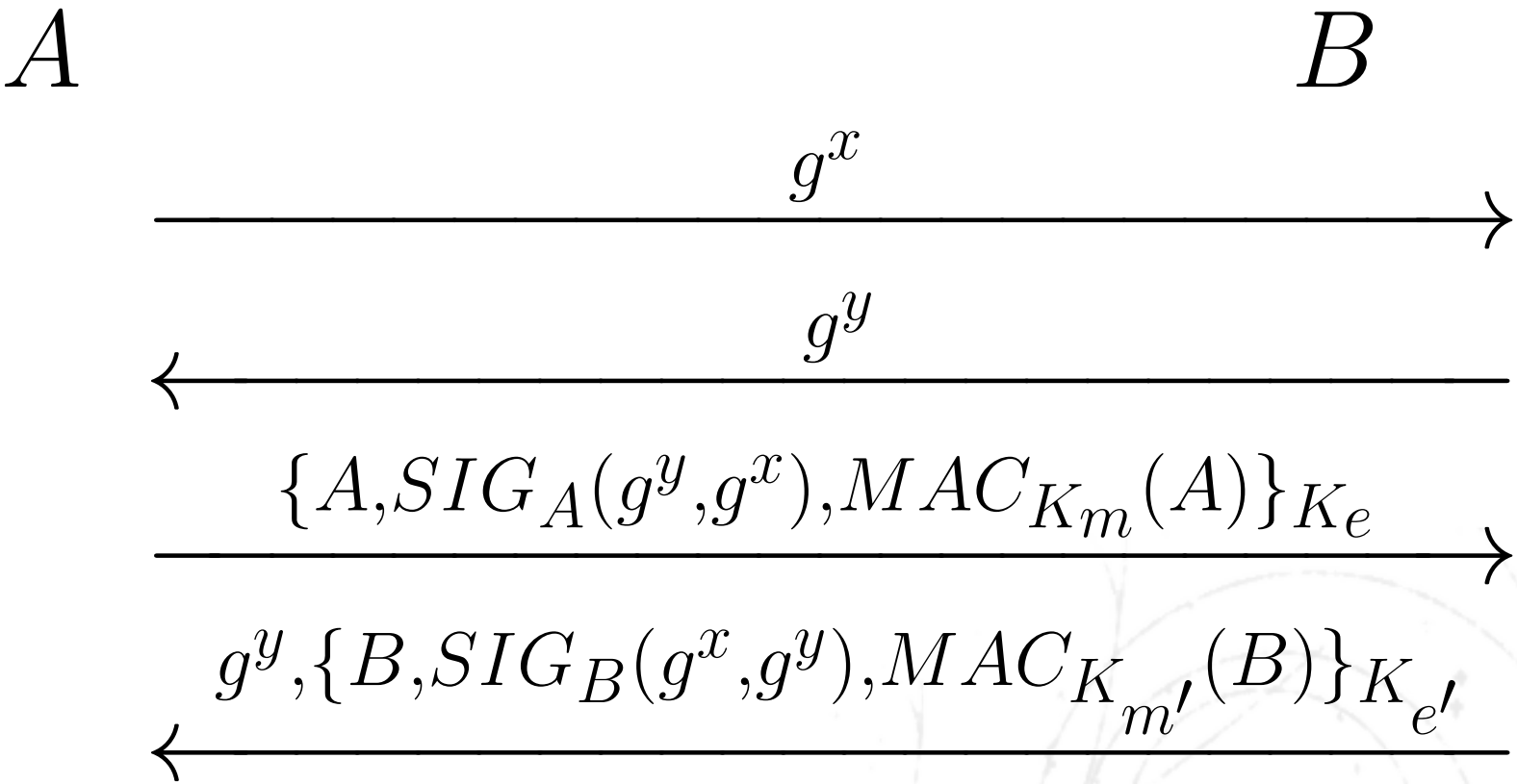
- Se genera una llave K_m a partir de la llave K
- SIG y MAC tienen roles complementarios un ataque M-I-T-M y para asociar la identidad
- No requiere conocer el ID de las partes para su identificación.

SIGMA-I: protección activa del ID del Iniciador



- Se genera una llave K_m, y una K_e a partir de la llave K
- B es el primero que se identifica
- La identidad del Iniciador (A) está protegida.

SIGMA-R: protección activa del ID del Destinatario (R)



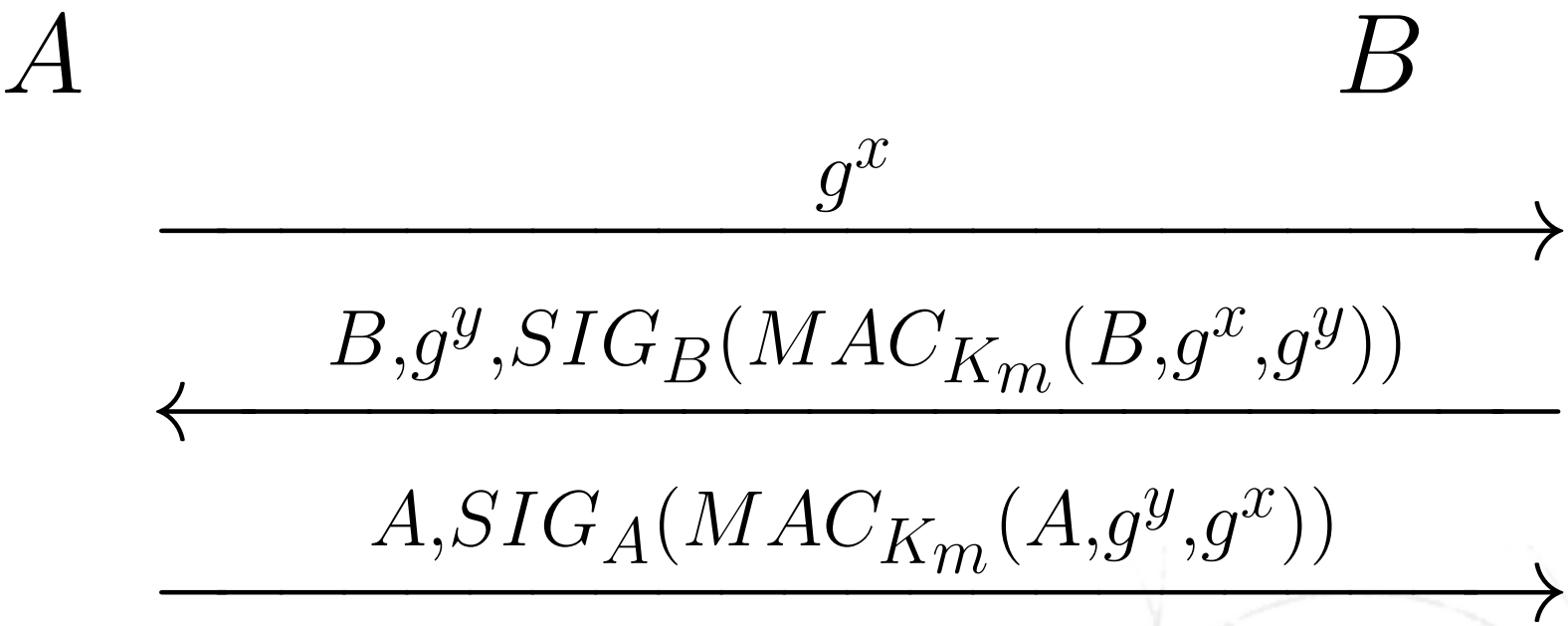
- Se generan llaves K_m, $K_{m'}$, K_e, y K'_e a partir de la llave K

IKE: Internet Key Exchange

- Crea SAs (Asociaciones de Seguridad) para utilizarse en un enlace IPSec
- Administra las SAs

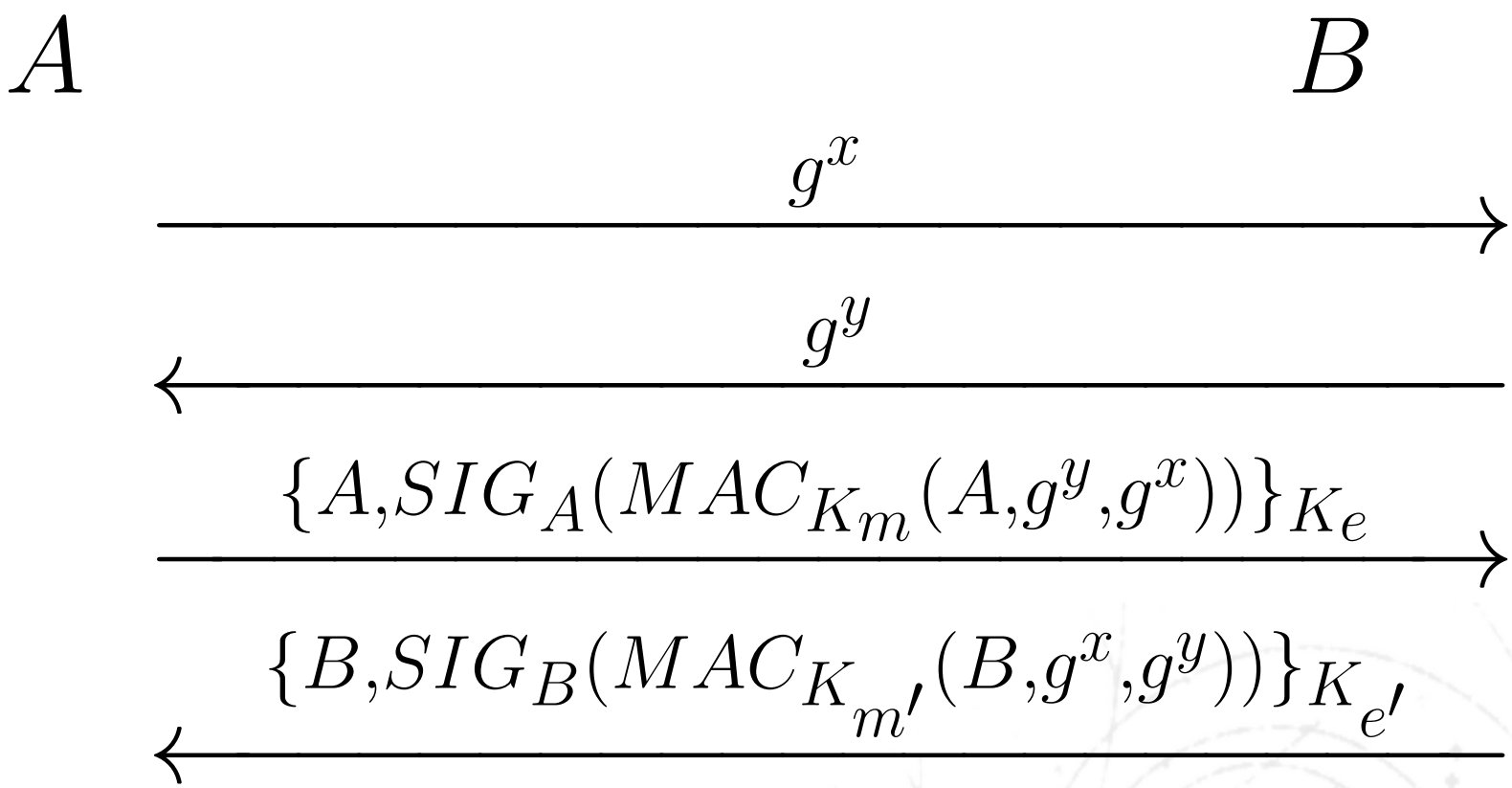
- Cuando un nodo A quiere comunicarse con B bajo la protección de un enlace IPSec, es necesario un intercambio de llave simétrica.
- La derivamos de SIGMA, visto recientemente.

Variante de IKEv1: MAC bajo SIG



- IKE modo "agresivo" (sin protección de ID).

IKE modo principal



- Se generan llaves K_m, $K_{m'}$, K_e, y K'_e a partir de la llave K