74
2 Ejem.

Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

Sistemas de Criptografía de Clave Pública

Tesis Profesional

Que para obtener el título de

INGENIERO MECANICO ELECTRICISTA

Presenta

JIANG YING

Director de Tesis:
DR. FEDERICO KUHLMANN RODRIGUEZ

México, D. F. 1985

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

R E S U M E N

En este trabajo se presenta un desarrollo de sistema de criptografía modernos llamados sistemas criptográficos de clave pública. También se presenta una breve descripción de la criptografía convencional, la cual sirvió como una base para la criptografía contemporanea.

Las amplias aplicaciones de teleprocesamiento hace que la necesidad de nuevos tipos de sistemas criptográficos aumenta, ya que los sistemas convencionales no puede proveer suficiente seguridad y/o no son aplicables en muchos de los sistemas de comunicación electrônicos, sobre todo que tiene una gran dificultad para sususuarios que es el problema de distribución de claves por canales seguros. En los nuevos sistemas de clave pública, se evita el uso de canales seguros, se distribuyen las claves a través de un canal público (e.g., un archivo público) sin comprometer la seguridad del sistema. Estos sistemas son computacionalmente seguros, esto es que para un criptoanalista, su esfuerzo de romper la seguridad del sistema es finito, pero demasiado grande hasta que llega ser imposible aún usando la computadora.

Se describen dos sistemas diferentes de clave pública y un sistema auxiliar para uno de los sistemas con motivo de disminuir la desventaja que tiene ésto. Se dan los antecedentes matemáticos requeridos, para sus desarrollos. Se discute la seguridad de cada sistema, y al final, como conclusión se hace una comparación y una interrelación entre ellos.

C O N T E N I D O

I. INTRODUCCION

Cuando información valiosa o secreta necesita ser transmitida o almacenada, es recomendable protegerla contra uso por personas no autorizadas. Por ejemplo, en el correo la correspondencia contiene información personal que puede ser considerada como información secreta. Esta está protegida por los sobres y también por las leyes que prohiben violación de las mismas.

Sin embargo, esta protección a veces no es suficiente. En este ejemplo, el sobre se puede destruir fácilmente y la gente puede tener acceso illegal sobre el contenido del mismo (a esa gente le llamamos oponente). Para incrementar la protección, la gente empezó a modificar su información siguiendo alguna regla específica llamada clave, y enviar o almacenar esa modificación de la información en vez de la información misma; esa clave se transmite por otro medio considerado seguro llamado canal seguro (e.g. un mensajero privado) al receptor deseado o legítimo. Así forma una técnica llamada genéricamente criptografía.

Una etapa muy importante en el desarrollo de la criptografía fue durante la Segunda Guerra Mundial. En este período, debido a las necesidades militares y diplomáticas, se desarrollaron muchos métodos y máquinas para resolver ese problema de la insuficiencia de protección física para la seguridad de la información. Después de esa época, se abandonó un poco el estudio de esa técnica criptográfica hasta hoy en día, en que las telecomunicaciones ya no se realizan por sistemas de lápiz y papel, sino por sistemas electrónicos o de microondas, y la información ya no se almacena en libretas sino en las memorias de las computadoras o en las memorias secundarias para las computadoras que son cintas y discos; en ese caso la protección física ya no solamente es insuficiente, sino también inaplicable. Por lo tanto es cuando hay que utilizar y desarrollar más las técnicas criptográficas. Por otra parte, afortunadamente la proliferación de las comunicaciones electrónicas y las computadoras también produce un decremento marcado en el costo de las técnicas criptográficas, lo cual había sido una de las grandes limitaciones en el uso de las mismas.

El estudio de las técnicas criptográficas es la criptografía. En la siguiente sección hablamos de los conceptos fundamentales de la criptogra

fía.

1.1 Fundamentos de la criptografía.

La criptografía es el estudio de esquemas matemáticos para resolver dos tipos de problemas de seguridad. El primer problema criptográfico es la privacía, la cual evita la extracción no autorizada de información de los medios de comunicación a través de un canal público en donde la seguridad no es adecuada para las necesidades de los usuarios. El segundo problema criptográfico es la autenticación que evita la inyección no autorizada de mensajes a un canal público, asegurando al receptor del mensaje la autenticidad de su remitente.

La figura 1 ilustra el flujo de información en un sistema criptográfico convencional de privacía.

El transmisor genera un texto simple o mensaje no cifrado P, el cual es comunicado a un receptor legítimo a través de un canal inseguro o canal público al cual tienen acceso las personas no autorizadas u oponentes.

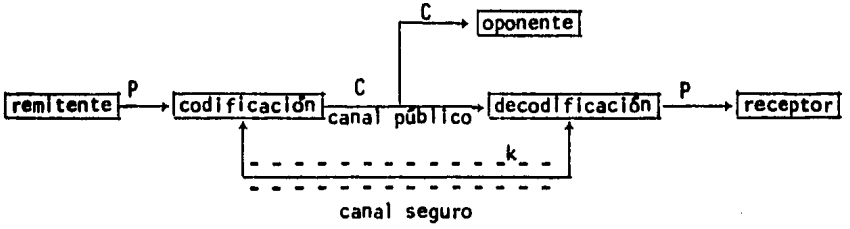

Fig. 1. Flujo de información en un sistema criptográfico convencional de privacía.

Para prevenir que el oponente entienda el contenido de P, el remitente cifra o codifica P por medio de una transformación invertible S_k para producir el criptograma o el texto en cifras $C=S_k(P)$, y lo transmite al receptor legítimo a través de un canal inseguro.

La clave K se transmite a través de un canal seguro (e.g., por un correo certificado, o con un mensajero privado). Cuando el receptor legí

timo obtiene C, como él conoce la clave K, lo puede descifrar o decodificar con la transformación inversa S_k^{-1} para obtener $S_k^{-1}(C)=S_k^{-1}(S_k(P))=P$ que es el mensaje original. No se utiliza el canal seguro para transmitir P por razones de capacidad y/o retardo.

La transformación S_k se selecciona de una familia de transformaciones conocida como un sistema criptográfico. El parámetro que se selecciona para la transformación individual es llamado la clave. Más formalmente, un sistema criptográfico es una familia de parámetros $\{S_k\}_{k\in\{K\}}$ de transformaciones invertibles:

$$S_k : \{P\} \rightarrow \{C\}$$

de un espacio de mensajes de texto simple P, a un espacio de mensajes de texto en cifras C. El parámetro o clave K es seleccionado de un conjunto finito {K} llamado espacio de claves.

La figura 2 ilustra por qué un sistema criptográfico se puede usar también para resolver problemas de autenticación.

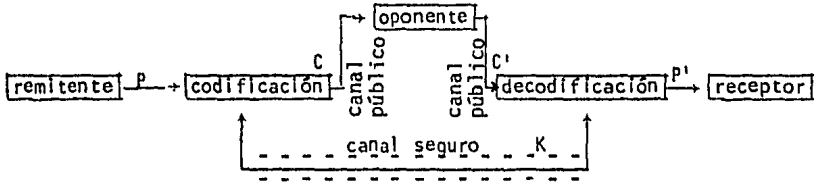

Fig. 2 Flujo de información en un sistema criptográfico convencional de autenticación.

En este caso, el oponente no solamente ve todo el criptograma transmitido por el canal, sino, tal vez, también pueda alterarlo. El receptor legítimo se protege descifrando todos los mensajes que recibe y aceptando solamente los mensajes cifrados con la clave correcta.

Un criptoanálisis es cualquier intento de un oponente para descifrar un criptograma C, para así obtener texto simple P, o cifrar un texto simple no auténtico P' para obtener un criptograma aceptable C' sin sa

ber la clave K. Si el criptoanálisis es imposible, esto es, que un criptoanalista no puede deducir P de C, o C' de P' sin conocer la clave K de antemano, entonces el sistema criptográfico es seguro. Esto es a lo que se pretende llegar.

1.2 Ataques criptoanalíticos.

El primer paso para el acceso de un sistema criptográfico es clasificar los tipos de ataques. A continuación se presenta una clasificación.

- Ataque con solo texto en cifras.

Si en un ataque criptoanalítico el criptoanalista posee solamente el texto en cifras, éste es un ataque con solo texto en cifras; esto ocurre frecuentemente en la práctica. En este caso, el criptoanalista utiliza solamente el conocimiento de las propiedades estadísticas del lenguaje usado (e.g., en inglés la letra E ocurre 13%), y conocimiento de ciertas palabras probables (e.g., una carta empieza probablemente con "Estimado señor:"). Es la amenaza más débil a la que un sistema puede estar sujeto, y cualquier sistema que sucumba a esto se considera totalmente inseguro.

- Ataque de texto simple conocido.

Si en un ataque criptoanalítico, el criptoanalista domina una cantidad substancial del texto simple y texto en cifras correspondientes, el ataque se llama ataque de texto simple conocido. Un sistema que es seguro contra ese tipo de ataques libera a los usuarios de las necesidades de mantener los mensajes pasados en secreto o de parafrasearlo antes de la desclasificación. Aunque un ataque de texto simple conocido no es siempre posible, su ocurrencia es suficientemente frecuente, y un sistema que no puede resistir esto, no es considerado seguro.

- Ataque de texto simple elegido.

Si en un ataque criptoanalítico, el criptoanalista puede seleccionar de un número ilimitado de mensajes de texto simple y examinar los criptogramas resultantes, éste es un ataque de texto simple elegido, y es diff

cil llevar a cabo en la práctica, pero puede ser aproximado. Un sistema que está seguro del ataque de texto simple elegido, puede evitar que sus usuarios se preocupen porque sus oponentes puedan colocar mensajes en su sistema.

1.3 Ejemplos de criptografía convencional.

1.3.1 La técnica más sencilla para codificar es la simple sustitución cuya clave es un alfabeto permutado, y las letras del alfabeto normal son sustituidas por las del alfabeto permutado. Por ejemplo, si la clave es: CYWAFQHOULPMZSINJGBKVDRETX, entonces, A está reemplazada por C, B está reemplazada por Y, etc. Si el mensaje del texto simple es:

ESTE MENSAJE ESTA CIFRADO,

el mensaje transformado será:

FBKF ZFSBCLF FBKC WUQGCAI.

Es fácil ver que este sistema no es seguro ni siquiera contra el ataque con solo texto en cifras.

1.3.2 Otra técnica es la de la transposición, en donde se transponen las posiciones de las letras de texto simple en vez de permutar el alfabeto. Como un ejemplo, si el mensaje anterior se divide en grupos de cinco caracteres (incluye los espacios), y las letras en cada grupo se vuelven a permutar de acuerdo a la regla (1↦5, 2↦3, 3↦1, 4↦2, 5↦4), el tercer caracter de cada grupo está escrito como el primero, el primer caracter está escrito como el quinto, etc. El criptograma será:

TES ENSEAM EESJ CAITADROF

En este caso, cualquier persona puede reconstruir un texto simple descubriendo el rompimiento de las palabras cortas de uso común, así como E y S de ESTE y juntarlos nuevamente.

Aunque estas técnicas no son usadas con frecuencia, son componentes

importantes en los sistemas criptográficos más complicados.

1.3.3 Se desarrolló también otra técnica que consiste en un codificador polialfabético en donde varios alfabetos de sustitución se utilizan periódicamente para codificar un mensaje. Por ejemplo, si se usan 5 alfabetos, las letras número $5n+i$ son cifradas en el i-ésimo alfabeto (e.g., las letras número 1, 6, 11, 16, ... son cifradas en el primer alfabeto).

Como el uso periódico de alfabetos consiste precisamente en que cada n-ésima letra está en el mismo alfabeto en el texto en cifras, el criptoanalista puede descifrar verificando el período n con un conteo de frecuencia en cada n-ésima letra de texto en cifras, y esto lo hace en cada una de las n posibles fases. Si cada uno de los n conteos de frecuencia obtenido en esta forma tiene una distribución muy distinta de la uniforme, quiere decir que tiene la característica de una sustitución simple monoalfabética. Entonces, el período supuesto es correcto. Una vez que el período ha sido determinado, el problema se puede resolver como un conjunto de n simples sustituciones diferentes.

1.3.4 En un esfuerzo por remover el defecto de los codificadores polialfabéticos periódicos, los criptógrafos cambiaron al uso de codificadores de clave corrida, los cuales son polialfabéticos aperiódicos. La clave es típicamente el nombre de un libro accesible junto con número de página, de renglón y columna (e.g., "La Santa Biblia", 1960, Sociedad Bíblica en América Latina, Pág. 436, renglón 11, columna 6). Para cifrar el mensaje: "TENEMOS UN GRAN PROBLEMA", escribimos el mensaje y el texto del libro "TRAJO A LA CASA DE DIOS LO QUE" uno sobre el otro como se muestra en lo siguiente, ordenando el alfabeto de 0 a 25, y a la suma módulo 26.

texto simple:	T	E	N	E	M	O	S	U	N	G	R	A	N
	19	4	13	4	12	14	18	20	13	6	17	0	13
	P	R	O	B	L	E	M	A.					
	15	17	14	1	11	4	12	0					
clave:	T	R	A	J	O	A	L	A	C	A	S	A	D
	19	17	0	9	14	0	11	0	2	0	18	0	3
	E	D	I	O	S	L	O	Q.					
	4	3	8	14	18	11	14	16					

suma (mód. 26): 12 21 13 13 0 14 3 20 15 19 20 22 15
3 15 0 16

texto en cifras: M V N N A O D U P G J A Q T U W P D P A Q.

Para descifrar el texto en cifras, se restan los números del texto de la clave a los números del texto en cifras, y la resta módulo 26, ya se obtiene el texto simple:

resta (mód. 26): 19 4 13 4 12 14 18 20 13 6 17 0 13 15 17
14 1 11 4 12 0.

texto simple: T E N E M O S U N G R A N P R O B L E M A .

Aunque en este método se usó la selección aperiódica de 26 alfabetos (uno por cada uno de los 26 valores posibles de la clave corrida), Bazeries [4] resolvió codificaciones de clave corrida en los años 1890. La manera más efectiva de resolver el problema depende de que el criptoanalista sepa una palabra probable que posiblemente ocurre en el texto simple. En comunicaciones militares, las palabras probables son: BATALLON, ATAQUE, COMPAÑIA, etc. Aún sin saber qué tipo de comunicación es, el criptoanalista puede usar palabras comunes o grupos de letras tal como: ESTE, CUAL, CION, etc. para probar la existencia de una palabra probable, criptoanalista la substrae del texto simple y la descifra, o sea resta al criptograma los números del texto simple supuesto. La resta módulo 26, en todas las posiciones posibles, si se presenta la palabra, se presenta la clave cuando se prueba en la localización correcta. Cuando se prueba en una ubicación incorrecta (y todas las ubicaciones son incorrectas si la palabra no se presenta), el criptoanalista encuentra un resultado de aspecto aleatorio.

Las técnicas criptográficas tales como sustitución y transposición que operan sobre el texto simple sin considerar su escritura lingüística son llamadas cifradores, y el texto criptográfico que producen es texto en cifras.

1.3.5 Un sistema criptográfico que opera sobre grandes unidades lingüísticas de texto simple, tal como palabras o frases, es llamado un códi

go. Un código usualmente consiste en una lista de palabras o frases llamados grupos de código. Como los grupos de códigos son generalmente más cortos que las expresiones que ellos presentan, los códigos ofrecen la ventaja de compresión de datos. Para uso propio, los códigos son mucho más difíciles de romper que los sistemas clásicos descritos anteriormente, por la razón de que involucran una gran cantidad de claves. Por ejemplo, la clave para la sustitución simple en un alfabeto permutado se representa por menos de 90 bits, mientras que un libro de código de un buen tamaño tal vez contiene unos cientos de miles o aún millones de bits. Además, se remueve la redundancia del mensaje, y se opera sobre bloques relativamente grandes de texto simple (palabras o frases) y con esto se encubre información local que tal vez de otro modo podría proveer indicios criptoanalíticos valorables.

A pesar de sus éxitos en algunas circunstancias, los códigos son vulnerables al ataque de texto simple conocido, y no son fáciles de cambiar si están comprometidos y esto viola el principio básico de la seguridad.

A medida que las técnicas tan elementales como éstas eran utilizadas frecuentemente, se vió que ninguna de ellas ofrecía una garantía absoluta de seguridad. Actualmente estas técnicas ya no son de mucha utilidad debido a la disponibilidad de las técnicas electrónicas más fáciles de usar y menos costosas. Sin embargo, estas técnicas simples sirvieron como base para lograr los desarrollos posteriores.

1.4 Seguridad incondicional y computacional.

Hay dos caminos diferentes fundamentales en los cuales los sistemas criptográficos pueden ser seguros.

En algunos sistemas la cantidad de información disponible para el criptoanalista es realmente insuficiente para determinar las transformaciones para cifrar y descifrar, sin importar cuánta capacidad de cómputo tenga disponible el criptoanalista. Un sistema de este tipo es un sistema incondicionalmente seguro.

En el otro caso el material interceptor contiene suficiente informa

ción que permite que el problema criptoanalítico tenga una solución, pero debido a que los recursos de cómputo son limitados no hay garantía de que el criptoanalista pueda encontrarla. Entonces, el objetivo del diseño de un sistema criptográfico es hacer operaciones no costosas para cifrar y descifrar tales que cualquier operación criptoanalítica sea lo suficientemente complicada para que pueda ser costeable. Esto requiere que la tarea del criptoanalista, a pesar de que sepa que es factible encontrar la clave en un tiempo finito, es prohibitivamente grande tanto la cantidad de memoria que usa como el tiempo que se tarda en hacerla. La tarea de esa magnitud no es computacionalmente ejecutable y el sistema criptográfico asociado es computacionalmente seguro.

El sistema incondicionalmente seguro más comúnmente usado es un cifrador que usa sólo una vez, en el cual el texto simple es combinado con una clave de la misma longitud seleccionada aleatoriamente la gran cantidad de claves para que tal sistema sea descifrado hace que ésto no sea práctico en muchas aplicaciones.

En este trabajo se habla más en detalle de los sistemas computacionalmente seguros, puesto que son los de mayor aplicabilidad en general.

1.5 Sistema de Clave Pública.

La dificultad de la distribución de clave ha sido una de las mayores limitaciones en el uso de la tecnología criptográfica convencional. En un sistema como se muestra en la fig. 1, el remitente y el receptor, para tener una comunicación criptográfica, necesitan usar un canal físicamente seguro para la distribución de la clave. Ambos tienen que esperar mientras las claves sean mandadas o las mandan a priori. Si en tal sistema hay un número grande de usuarios, habrá un número aún más grande $(n^2-n)/2$ de parejas virtuales, esto es, requiere casi 500 mil millones de claves para un sistema con solo un millón de usuarios. Por tanto, para desarrollar sistemas grandes y seguros de telecomunicaciones, se necesita de otras soluciones para la distribución de claves.

Una solución es la codificación en cadena. Sin embargo, se puede realizar sólo en las redes de comunicación militar donde cada nodo es una

organización físicamente segura, provista por determinada persona. Un sistema de este tipo consiste en permitir que la vulnerabilidad se concentre en un lugar, en vez de distribuirse a través de la red. Cada usuario, en lugar de compartir una clave en un nodo local comparte una clave "maestra" con una central de redes especial llamada central de distribución de clave. Cuando un usuario A desea comunicarse con un usuario B, el primero hace contacto con la central de distribución de clave, la cual genera una clave para esa comunicación específica. La central manda esta clave a A, cifrada con la clave maestra de A, y a B, cifrada con la clave maestra de B. Entonces, se retira para permitir a los usuarios A y B comunicarse directamente con la clave específica de esa conversación. La seguridad de esta conversación depende solamente de la seguridad de la central de distribución de la clave en lugar de depender de la seguridad de cada nodo a través de los cuales tienen que pasar los mensajes entre A y B.

Para las redes comerciales con un presupuesto menor comparado con las militares, esta solución ya no es tan atractiva, porque sus nodos no son manejables. Algunas redes comerciales aún ponen nodos en las propiedades de sus clientes en las ciudades donde no tienen sus propias instalaciones. En este caso, la codificación en cadena podría proveer poco o nada de protección.

Una solución más satisfactoria al problema de distribución de clave es la criptografía de clave pública. Como se indica en la figura 3, un sistema criptográfico de dos caminos permite la comunicación entre el transmisor y el receptor, pero el oponente es pasivo y sólo escucha.

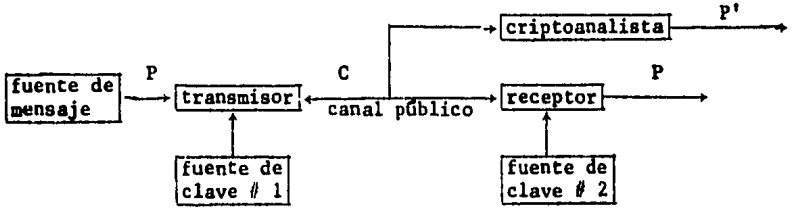

Fig. 3 Flujo de información en sistema de clave pública.

En este sistema, dos personas se comunican sólo a través de un canal público usando técnicas conocidas públicamente. Un sistema de este tipo es un sistema de clave pública. En caso contrario, es un sistema convencional.

Dos formas básicas de solución han sido sugeridos para este problema: 1ª. sistema de distribución de clave pública; 2ª. sistema criptográfico de clave pública.

En un sistema de distribución de clave pública, los usuarios pueden ponerse de acuerdo en una clave a través de un canal público. Esto quiere decir que ellos pueden obtener la misma clave (la llamamos la clave común) a través de los cálculos en donde utilizan las claves secretas de uno mismo y las claves públicas del otro usuario. Se usa esta clave común para cifrar el mensaje, y el inverso de esa clave para descifrar. Una tercera persona, o sea el oponente, no puede calcular esa clave común de estos dos usuarios a partir de sus claves públicas, debido a que a partir de esas claves públicas no se puede calcular las claves secretas aún usando la computadora. Por lo tanto no puede entender conversaciones que se efectúan entre estos dos usuarios. En el capítulo 3 se describirá un sistema de distribución de clave pública.

En un sistema criptográfico de clave pública se pueden separar las claves para cifrar y descifrar, o sea que a partir de la clave para cifrar se calcula la clave para descifrar es computacionalmente imposible, y esto permite que la clave para cifrar pueda estar publicada, mientras que la clave para descifrar se mantiene en secreto. Se presentará una descripción de éste sistema en el capítulo 2.

Un sistema de clave pública simplifica mucho el problema de la distribución de claves. Desde la primera vez que el sistema de clave pública fue propuesto, varios métodos ya se han desarrollado. En este trabajo, se presentarán unos de ellos para proteger las comunicaciones en privacía.

II. SISTEMA CRIPTOGRAFICO DE CLAVE PUBLICA

En este capítulo se presenta un sistema criptográfico de clave pública que evita el uso de un canal seguro para distribuir la clave en las comunicaciones privadas.

En los sistemas criptográficos convencionales, las claves tienen que ser protegidas cuidadosamente en su proceso de distribución, debido a que las funciones para cifrar y descifrar son inseparables, en el sentido que, cualquiera que tenga acceso a la clave para cifrar los mensajes también puede descifrarlos. En cambio, los sistemas criptográficos de clave pública tratan de separar las capacidades para cifrar y descifrar para que la privacia se pueda mantener sin necesidad de guardar en secreto la clave para cifrar, porque esta clave puede no ser utilizada una vez más para descifrar.

Los sistemas criptográficos de clave pública están diseñados de tal forma que es fácil generar un par de claves aleatorias invertibles: E para cifrar y, D para descifrar, y que es fácil operar con E y D, pero que es imposible calcular D a partir de E.

Un sistema criptográfico de clave pública consiste en un par de familias $\{E_k\}$, $k\in\{K\}$ y $\{D_k\}$, $k\in\{K\}$ de algoritmos que representan transformaciones invertibles:

$$E_k : \{M\} \to \{M\}$$

$$D_k : \{M\} \to \{M\}$$

en un espacio de mensaje M, tal que:

1). Para todo $k\in\{K\}$, D_k es el inverso de E_k, esto es, para cualquier k y cualquier M, $D_k(E_k(M))=M$.

2). Para todo $k\in\{K\}$ y $M\in\{M\}$, los valores de $E_k(M)$ y $D_k(M)$ son fáciles de calcular.

3). Para todo $k \in \{K\}$, calcular D_k o algún otro alternativo equivalente a D_k a partir de E_k es computacionalmente imposible.

4). Para todo $k \in \{K\}$, es fácil generar el par de inversas E_k y D_k utilizando k.

La tercera propiedad permite que la clave del usuario para cifrar sea pública sin comprometer la seguridad de su clave secreta D_k para descifrar. El sistema criptográfico puede entonces ser separado en dos partes: una familia de transformaciones para cifrar, y otra familia de transformaciones para descifrar, de tal manera que dado un miembro de una familia es imposible encontrar el miembro correspondiente de la otra.

La cuarta propiedad garantiza que hay un método eficiente para calcular las correspondientes parejas de transformaciones inversas. En la práctica, el equipo criptográfico tiene que contener un generador de números aleatorios para generar la clave k junto con un algoritmo que genere el par (E_k, D_k) a partir de la clave k.

Un sistema de este tipo simplifica mucho el problema de la distribución de la clave. Cada usuario genera un par de transformaciones inversas E y D, mantiene la transformación para descifrar D en secreto, nunca transmite esta clave D en ningún canal y pública la transformación para cifrar E en un directorio público junto con su nombre y dirección. Cualquier persona que quiere comunicarse con este usuario puede cifrar el mensaje con la clave pública de él y se lo manda; sin embargo, ninguna persona lo puede descifrar excepto el mismo usuario, debido a que solamente él conoce D.

Un sistema criptográfico de clave pública empleado por Rivest, Shamir y Adleman (6) se presenta en este capítulo. En la siguiente sección se darán los conceptos matemáticos necesarios para el desarrollo de este sistema.

2.1 Conceptos matemáticos requeridos.

2.1.1 Teorema de Euler

1) Definición. Sea p un entero, diferente de 1 y -1. Si p=ab, donde a y b son enteros, entonces p es llamado número primo si y sólo si uno de los dos, a ó b es igual a 1 ó -1.

2) Definición. Si a y b son enteros, entonces b es divisible por a, o a divide b, si existe un entero x, tal que ax=b.

3) Definición. Si a y b son enteros, entonces un entero c es un común divisor de a y b, si c divide a, y c divide b. El máximo común divisor de a y b es el mayor de sus comunes divisores, y se denota como: (a,b) o gcd(a,b).

4) Definición. Dos números a y b, son relativamente primos si gcd(a,b)=1, es decir, el máximo común divisor de estos dos números es igual a 1.

5) Definición. Si n es un número entero positivo, y a y b son enteros con la propiedad de que a-b es divisible por n, decimos que a y b son congruentes módulo n, y lo denotamos como $a \equiv b \pmod{n}$.

6) Propiedades. La congruencia módulo n es una relación de quivalencia, o sea, satisface las siguientes propiedades:

a) Reflexividad: $a \equiv a \pmod{n}$.
b) Simetría: si $a \equiv b \pmod{n}$; entonces, $b \equiv a \pmod{n}$.
c) Transitividad: si $a \equiv b \pmod{n}$, y $b \equiv c \pmod{n}$, entonces, $a \equiv c \pmod{n}$.

7) Definición. Si a es un entero y n es un entero positivo, entonces de acuerdo con el algoritmo de la división, existen enteros q y r, tales que $a \equiv qn+r$, donde $0 \leq r < n$. El número r (que es único) es llamado residuo, cuando a es dividido por n. Por ejemplo:

si: a=7 y n=5
entonces q=1 y r=2
es decir, 7=1*5+2

8) Teorema. Dos enteros a y b son congruentes módulo n, si y sólo si sus residuos al dividirlos por n son iguales. (Demostración ver [1, pág. 20]). Por ejemplo: 7 y 2 son congruentes módoulo 5.

9) Teorema. Supóngase que $a\equiv a'(\text{mod } n)$, y $b\equiv b'(\text{mod } n)$; entonces,

$a+b\equiv a'+b'(\text{mod } n)$,
$a-b\equiv a'-b'(\text{mod } n)$,
$ab\equiv a'b'(\text{mod } n)$. (Demostración ver [1, pág. 21])

Ejemplo:

$1\equiv 6(\text{mod } 5)$ $2\equiv 7(\text{mod } 5)$

$2+1\equiv 3\equiv 7+6(\text{mod } 5)$,
$2-1\equiv 1\equiv 7-6(\text{mod } 5)$,
$2*1\equiv 2\equiv 7*6(\text{mod } 5)$.

10) Definición. Si n es cualquier entero mayor que 1, se define la función de Euler $\phi(n)$ como el número de los números positivos menores y relativamente primos a n. Por ejemplo:

$\phi(2)=1$, $\phi(3)=2$, $\phi(4)=2$, $\phi(5)=4$, $\phi(6)=2$, $\phi(7)=6$.

Es fácil ver que para cualquier número primo p, $\phi(p)=p-1$.

11) Teorema. Teorema de Euler. Si n es un entero mayor que 1, y a es un entero que es relativamente primo a n, entonces,

$a^{\phi(n)}=1(\text{mod } n)$. (Demostración ver [1, pág. 35])

Por ejemplo: si: a=2 y n=5
entonces $\phi(5)=4$
$2^4=1(\text{mod } 5)$

Los teoremas y definiciones anteriores se aplicarán en la sección 2.2

2.1.2 Símbolo de Jacobi.

12) Definición. Para todo a, tal que (a,n)=1, a recibe el nombre de residuo cuadrático módulo n, si la congruencia $x^2 \equiv a \pmod{n}$ tiene solución. Por ejemplo:

si: a=4 y n=7 decimos que 4 es residuo cuadrático módulo 7 porque $x^2 \equiv 4 \pmod{7}$ sí tiene solución, es decir, x=2.

13) Teorema. Si p es primo, y (a,p)=1 entonces, la congruencia $x^n \equiv a \pmod{p}$ tiene una solución, si

$$a^{(p-1)/(n,p-1)} \equiv 1 \pmod{p}$$

o ninguna solución, si

$$a^{(p-1)/(n,p-1)} \not\equiv 1 \pmod{p}$$

(Demostración, ver [5, pág. 59]).

Ejemplo: si: p=7, a=4 y n=2

$4^{6/2} \equiv 1 \pmod{7}$

entonces $x^2 \equiv 4 \pmod{7}$ sí tiene solución

y si: p=7, a=3 y n=2

$3^{6/2} \equiv 27 \equiv 6 \pmod{7} \not\equiv 1 \pmod{7}$

entonces $x^2 \equiv 3 \pmod{7}$ no tiene solución.

14) Definición. Si p es primo diferente de 2, y (a,p)=1, el símbolo de Legendre $\left(\frac{a}{p}\right)$ se define como 1, si a es residuo cuadrático, y -1, si a no es residuo cuadrático módulo p. Por ejemplo:

si: p=7 y a=4

$\left(\frac{4}{7}\right) = 1$

ahora bien, si: p=7 y a=3

$$\left(\frac{3}{7}\right)=-1$$

15) Teorema. Sea p un número primo impar y supóngase que a y b denotan enteros relativamente primos. a p, entonces:

a) $\left(\frac{a}{p}\right)\equiv a^{(p-1)/2}(\text{mod } p)$, e.g., $\left(\frac{4}{7}\right)\equiv 4^{6/2}(\text{mod } 7)\equiv 1(\text{mod } 7)$

b) $\left(\frac{a}{p}\right)\left(\frac{b}{p}\right)\equiv\left(\frac{ab}{p}\right)$, e.g., $\left(\frac{4}{7}\right)\left(\frac{3}{7}\right)=\left(\frac{12}{7}\right)\equiv -1$

c) $a\equiv b(\text{mod } p)$ implica que $\left(\frac{a}{p}\right)=\left(\frac{b}{p}\right)$, e.g.,

$5\equiv 12(\text{mod } 7)$ $\left(\frac{5}{7}\right)\equiv\left(\frac{12}{7}\right)=-1$

d) $\left(\frac{a^2}{p}\right)=1$, $\left(\frac{1}{p}\right)=1$, $\left(\frac{-1}{p}\right)=(-1)^{(p-1)/2}$, e.g.,

$\left(\frac{2^2}{7}\right)=1$, $\left(\frac{1}{7}\right)=1$, $\left(\frac{-1}{7}\right)=(-1)^{(7-1)/2}=-1$

(Demostración, ver [5, pág. 73-74]).

16) Definición. Sean P y Q dos números tales que (P,Q)=1, Q>0, Q es impar. Se puede expresar $Q=q_1q_2\ldots q_s$, donde q_i son primos diferentes de 2, no necesariamente distintos. Entonces, el símbolo de Jacobi $\left(\frac{P}{Q}\right)$ está definido por

$$\left(\frac{P}{Q}\right)=\prod_{i=1}^{s}\left(\frac{P}{q_i}\right)$$ donde $\left(\frac{P}{q_i}\right)$ es el símbolo de Legendre.

e.g., Q=9 y P=2

Q=3x3

$$\left(\frac{2}{9}\right)=\left(\frac{2}{3}\right)\left(\frac{2}{3}\right)=\left(\frac{4}{3}\right)=1$$

17) Teorema. Si Q es impar y Q>0, entonces

$$\left(\frac{-1}{Q}\right)=(-1)^{(Q-1)/2} \quad y$$

$$\left(\frac{2}{Q}\right)=(-1)^{(Q^2-1)/8} \qquad \text{(Demostración, ver [5, pág. 81]).}$$

e.g., si Q=9

$$\left(\frac{-1}{9}\right)=(-1)^4=1$$

$$\left(\frac{2}{9}\right)=(-1)^{10}=1$$

18) Teorema. Si P y Q son impares y positivos, y si (P,Q)=1, entonces

$$\left(\frac{P}{Q}\right)\left(\frac{Q}{P}\right)=(-1)^{((p-1)/2)\cdot((Q-1)/2)}$$

(Demostración, ver [5, pág. 82]).

e.g., si P=7 y Q=9

$$\left(\frac{7}{9}\right)\left(\frac{9}{7}\right)=(-1)^{3x4}=1$$

2.1.3 Teorema de los Números Primos.

19) Teorema. Existen arbitrariamente grandes vacíos en la serie de los primos. Dicho de otra manera, dado cualquier entero positivo k, existen k enteros compuestos consecutivos.

Por ejemplo: Entre 3 y 5, k=1; entre 7 y 11, k=3; entre 19 y 23, k=3; etc.

Los primos están irregularmente espaciados, tal y como lo sugiere éste teorema. Si se denota el número de primos que no exceden a x por $\Pi(x)$, podría preguntarse acerca de la naturaleza de esta función. Debido a lo irregular de la ocurrencia de los primos, no puede esperarse una fór

mula sencilla para $\Pi(x)$. Sin embargo, uno de los resultados más impresionantes de la teoría avanzada de los números, el teorema de los números primos, proporciona una aproximación asintótica para $\Pi(x)$. Establece que

$$\lim_{k\to\infty} \Pi(x) \frac{\log x}{x} = 1,$$

es decir, que la razón de $\Pi(x)$ a $x/\log x$ tiende hacia 1 conforme x se hace indefinidamente grande.

Los teoremas y definiciones de los dos incisos anteriores se aplicarán en la sección 2.3.

2.2 Desarrollo del Sistema RSA [6].

El sistema de Rivest, Shamir y Adleman (RSA) resuelve el problema de distribuir la clave para cifrar sin comprometer la seguridad de la clave para descifrar, a través de un canal público. Este sistema está basado en el hecho de que es computacionalmente fácil encontrar números primos grandes (e.g., 100 dígitos), pero computacionalmente ineficiente factorizar un producto de dos tales números en sus factores primos. Entonces, un usuario A puede usar el producto de dos números primos grandes como la clave para cifrar y la pública, además, calcula la clave para descifrar a partir de esos dos números primos grandes que están mantenidos en secreto.

Cualquier otra persona no puede calcular la clave para descifrar a partir de este producto, debido a que no puede factorizar ese producto en sus factores primos, fácilmente.

El usuario A selecciona dos números primos muy grandes P y Q en forma aleatoria y los multiplica para obtener un número N. El número N se hace público, pero sus factores P y Q se mantienen en secreto. El usuario A, entonces calcula la función de Euler $\emptyset(N)$ (ver definición 10 de la sección 2.1.1) por medio de:

$$\emptyset(N)=(P-1)(Q-1) \qquad [6]$$

Luego, el usuario A elige aleatoriamente otro número E en el intervalo $(2,\emptyset(n)-1)$, y este número E también se publica.

El usuario B representa el mensaje que quiere comunicar al usuario A como una sucesión de enteros $M_1, M_2, \ldots,$ siendo cada M_i un entero entre 0 y N-1 (aproximadamente 10 dígitos decimales para que sea muy difícil una búsqueda directa de M_i) y los cifra usando la información pública E y N por medio de:

$$C \equiv M^E \pmod N$$

donde C representa el bloque cifrado.

Usando el número secreto $\emptyset(N)$, el usuario A puede, fácilmente, calcular el número D, debido a que

$$ED \equiv 1 \pmod{\emptyset(N)}$$

Equivalentemente $ED \equiv k\emptyset(N)+1$, aplicando el teorema de Euler, (ver el teorema 11 de la sección 2.1.1), $X^{\emptyset(N)} \equiv 1 \pmod N$, deducimos que $X^{ED} \equiv X^{k\emptyset(N)+1} \equiv X \pmod N$, para todos los enteros X que están entre 0 y N-1, y para todos los enteros k. Por lo tanto:

$$C^D \equiv M^{ED} \equiv M^{k\emptyset(N)+1} \equiv M \pmod N.$$

Para calcular D, el usuario A puede usar el algoritmo extendido de Euclides 2, pág. 315, ex. 15 el cual se explicará a continuación. Este algoritmo determina el máximo común divisor de los números enteros positivos U y V. Se usa un vector de números enteros (U_1, U_2, U_3), tal que $UU_1+VU_2=U_3=\gcd(U,V)$, es decir que U_3 es el máximo común divisor de U y V. En nuestro caso, como

$$ED=k\emptyset(N)+1,$$

tenemos

$$ED-k\emptyset(N)=1=\gcd(E,\emptyset(N)),$$

es fácil, ver que U es E, V es $\emptyset(N)$, y $(U_1, U_2, U_3)=(D, -k, 1)$. Si

el máximo común divisor de E y $\emptyset(N)$, U_3, resulta diferente de 1, U_1 no es la clave para descifrar, D, que estamos buscando. Entonces hay que seleccionar otro valor de E y volver a calcular siguiendo el mismo proceso. Resumimos este algoritmo en la siguiente forma:

1a. Definir $U=E$, $V=\emptyset(N)$;
1b. Asignar $(U_1, U_2, U_3)=(1, 0, U)$;
1c. Asignar $(V_1, V_2, V_3)=(0, 1, V)$.

2. Si $V_3\neq 0$, entonces, $q=\lfloor U_3/V_3 \rfloor$ (la parte entera del cociente U_3/V_3).

$$(U_1', U_2', U_3')=(U_1, U_2, U_3),$$
$$(V_1', V_2', V_3')=(V_1, V_2, V_3);$$

Si $V_3=0$, si U_1 es positivo, el algoritmo termina.
si U_1 es negativo, $q=q-1$,

$$(U_1, U_2, U_3)=(U_1', U_2', U_3')$$
$$(V_1, V_2, V_3)=(V_1', V_2', V_3');$$

3a. Asignar $(t_1, t_2, t_3)=(U_1, U_2, U_3)-(V_1, V_2, V_3)$;

3b. Asignar $(U_1, U_2, U_3)=(V_1, V_2, V_3)$;

3c. Asignar $(V_1, V_2, V_3)=(t_1, t_2, t_3)$;

3d. Regresar al paso 2.

Ejemplo. Sean $U=E=7$, $V=\emptyset(N)=480$, $q=\lfloor U_3/V_3 \rfloor$.

La siguiente tabla muestra el procedimiento para calcular la clave para descifrar, D, a partir de E y $\emptyset(N)$, donde $\emptyset(N)$ es clave secreta.

q	U_1	U_2	U_3	V_1	V_2	V_3
/	1	0	7	0	1	480
0	0	1	480	1	0	7
68	1	0	7	-68	1	4
1	-68	1	4	69	-1	3
1	69	-1	3	-137	2	1
3	-137	2	1	480	-7	0
2	-137	2	1	343	-5	1
1	343	-5	1	-580	7	0

De aquí vemos que D=343, y k=5.

Ahora siguiendo este ejemplo veremos el caso del procedimiento de codificación y decodificación. Si se supone que se seleccionaron P=17, C=13, entonces, N=PQ=527 y $\emptyset(N)=(P-1)(Q-1)=480$. Si se elige E=7, entonces usando el algoritmo de Euclides se puede calcular D=343 (se calculó en el ejemplo anterior).

Para cifrar un mensaje M=2, se usa la clave pública (E,N) por medio de las relaciones:

$$C \equiv M^E \pmod N$$
$$\equiv 2^7 \pmod{527}$$
$$\equiv 128$$

Para descifrar, se usa la clave privada D:

$$M \equiv C^D \pmod N$$
$$\equiv 128^{343} \pmod{527}$$
$$\equiv 128^{256} \cdot 128^{64} \cdot 128^{16} \cdot 128^4 \cdot 128^2 \cdot 128^1 \pmod{527}$$
$$\equiv 35x256x35x101x47x128 \pmod{527}$$
$$\equiv 2 \pmod{527}$$

Nótese que de acuerdo al teorema 9 de la sección 2.1.1., se tiene que

$128 \equiv 128 \pmod{527}$, $128^2 \equiv 47 \pmod{527}$, $128^4 \equiv 47^2 \equiv 101 \pmod{527}$, etc.

Los ejemplos anteriores ilustra únicamente la metodología del procedimiento para cifrar y descifrar. En casos prácticos, los números primos P y Q tienen que ser mucho muy grandes (100 dígitos), para que la factorización del producto de ellos sea muy difícil. Los métodos usuales para buscar números primos sólo puede encontrar primos pequeños (menos que 10 dígitos), no sirve para nuestro caso. Por lo tanto, exponemos un método para encontrar números primos muy grandes.

2.3 Búsqueda de Números Primos Grandes.

Para encontrar un número primo aleatorio de 100 dígitos, se generan aleatoriamente números impares de 100 dígitos hasta que se encuentra un primo. Por el teorema de números primos (ver el Teorema 19 en 2.1.3), se necesita buscar aproximadamente $(\ln 10^{100})/2 = 115$ números para encontrar un número primo.

Para buscar un número primo grande b, existe el algoritmo probabilístico de Solovay y Strassen [6]. Este selecciona un número aleatorio a de una distribución uniforme en [1, ..., b-1] y prueba:

$$\gcd(a,b)=1, \quad y \quad J(a,b)=\left(\frac{a}{b}\right)=a^{(b-1)/2} \pmod{b} : \ldots\ldots\ldots (1)$$

donde $J(a,b)$, o bien $\left(\frac{a}{b}\right)$ es el símbolo de Jacobi (ver la definición 16 en la sección 2.1.2). La expresión (1) es cierta, siempre y cuando b es primo. Si b es compuesto (producto de primos), la expresión (1) es falsa con una probabilidad de por lo menos de 1/2. Si (1) se mantiene cierto para cien valores de a seleccionados aleatoriamente, entonces es casi seguro que b es un primo, y la probabilidad de que b sea compuesto es igual a 2^{-100}. Aún si un compuesto fuera utilizado accidentalmente en el sistema, el receptor probablemente podría detectarlo por notar que la decodificación no ha sido correcta. Cuando b es impar, $a \leq b$, y $\gcd(a,b)=1$, el símbolo de Jacobi tiene un valor en -1, 1 y se puede calcular eficientemente con el siguiente programa recursivo de computadora:

```
J(a,b) = IF a=1 THEN 1 ELSE
```

$$\text{IF a es par THEN } J(a/2,b)*(-1)^{(b^2-1)/8}$$

$$\text{ELSE } J(b(\text{mod } a),a)*(-1)^{(a-1)*(b-1)/4}$$

Nótese que este método para encontrar números primos no analiza la primalidad de un número tratando de factorizarlo, sino que la analiza probabilísticamente.

Para una ganancia adicional en protección contra algoritmos sofisticados de factorización, P y Q deben ser diferentes en longitud, ambos (P-1) y (Q-1) deben contener factores primos grandes, y gcd(P-1, Q-1) debe ser pequeño. Con la primera condición se aumenta el dominio donde están P y Q. La segunda y la tercera condición sirven para que la factorización de (P-1)Q, o de P(Q-1), o de (P-1)(Q-1), sea muy difícil también. Esto debido a que como P y Q son números muy grandes P-1 y Q-1 se aproximan mucho a P y Q respectivamente (sólo que P-1, y Q-1 son números pares), por lo tanto tal vez existe algún algoritmo sofisticado para factorizar N por el camino de factorizar alguno de los productos (P-1)Q, P(Q-1), o (P-1)(Q-1). Para verificar la última condición, se usa el algoritmo de Euclides, el cual está descrito en la sección 2.2. Y para encontrar un número primo P, tal que P-1 tenga un factor primo grande, se genera un primo grande aleatorio U usando el método probabilístico, y entonces se calcula P=i*U+1, para i=2, 4, ..., (i no debe ser muy grande), probando con el mismo método hasta encontrar un número P que es primo. Si además se puede asegurar que U-1 también tiene factor primo grande, aún tiene otra seguridad adicional.

Una computadora de alta velocidad puede determinar si un número de 100 dígitos es primo en unos cuantos segundos, y puede encontrar el primer primo mayor que un número dado en unos minutos. De aquí vemos que el esfuerzo del usuario legítimo es pequeño.

Otro método para encontrar números primos grandes, es tomar un número de factorización conocida, sumar uno a éste, y analizar la primalidad del resultado. Si se encuentra un número primo P, es posible probar que éste es realmente un primo usando la factorización de P-1. Omitimos una discusión de ésto porque el método probabilístico es adecuado.

2.4 Prueba de Seguridad

Aunque no existen técnicas para probar que un sistema de codificación es seguro, con las pruebas disponibles se puede estimar si alguna persona no autorizada podría ingeniar alguna manera para romperlo.

Se muestra en esta sección que todos los métodos obvios para romper este sistema son por lo menos tan complejos como factorizar N en sus factores primos. Aunque factorizar números grandes probablemente no es difícil, es un problema bien conocido, en cuya solución han trabajado durante los últimos trescientos años muchos matemáticos famosos. Fermat (1601-1665) [2, pág. 342] y Legendre (1752-1833) [2, pág. 351] desarrollaron algoritmos de factorización, y algunos algoritmos modernos más eficientes están basados en el trabajo de Legrendre. Sin embargo, nadie ha encontrado todavía un algoritmo que puede factorizar un número de 200 dígitos en un tiempo razonable (unos minutos o cuando mucho unas horas). Se puede concluir que el sistema RSA ha sido ya prácticamente "certificado" por estos esfuerzos previos para encontrar algoritmos eficientes de factorización.

A continuación vemos la dificultad que tiene un criptoanalista para calcular la clave secreta a partir de la clave pública. No consideramos aspectos de protección de la clave secreta para evitar que sea robada, porque es suficiente con los métodos usuales de seguridad física.

Primero consideramos que un criptoanalista trata de factorizar la clave pública N en sus factores primos P y Q que son factores secretos utilizados en la determinación de la clave para descifrar, vemos que su esfuerzo es inmensamente grande.

2.4.1 Factorización de N.

Factorizar N puede permitir a un criptoanalista romper la secrecía del sistema. Los factores de N le permiten calcular $\emptyset(N)$, y así puede obtener la clave para descifrar D. Sin embargo, factorizar un número es mucho más difícil que determinar si un número es primo o compuesto. Ya hemos visto que para determinar si un número es primo sólo se necesitan unos

segundos o cuando mucho unos minutos. En cambio, veremos en seguida que para factorizar un número grande se necesitan días, o hasta años.

Hay varios algoritmos que factorizan un número grande. Knuth [2, sección 4.5.4] ha dado una excelente descripción de muchos de ellos. El algoritmo más rápido es el de Richard Schroeppel [6, (el algoritmo no está publicado)]. Este algoritmo puede factorizar un número grande N en aproximadamente

$$e^{\sqrt{\ln(N)\cdot\ln\ln(N)}}$$

$$= N^{\sqrt{\ln(\ln(N))/\ln(N)}}$$

$$= (\ln(N))^{(\sqrt{\ln(N)/\ln(\ln(N))})}$$

pasos (ln denota la función logaritmo natural).

La tabla 1 muestra el número aproximado de operaciones necesarias para factorizar N con el método de Schroeppel, y el tiempo que se requiere si cada operación tarda un microsegundo para varias longitudes del número N (en dígitos decimales).

Tabla 1

Dígitos	Número de operaciones	Tiempo
50	1.4×10^{10}	3.9 horas
75	9.0×10^{12}	104 días
100	2.3×10^{15}	74 años
200	1.2×10^{23}	3.8×10^{9} años
300	1.5×10^{29}	4.9×10^{15} años
500	1.3×10^{39}	4.2×10^{25} años

Cada usuario puede especificar la longitud del número N dependiendo de la seguridad y la rápidez de codificación que requiere en cada uso individual.

Ya vimos la dificultad de factorizar N para un criptoanalista, pero

éste quizás trate de determinar $\emptyset(N)$ sin factorizar N para calcular la clave para descifrar, D. En seguida mostramos que ésto no es más fácil que factorizar N.

2.4.2 Cálculo de $\emptyset(N)$ sin factorizar N.

Si un criptoanalista puede calcular $\emptyset(N)$, entonces puede romper la seguridad del sistema calculando la clave para descrifrar, D, como la inversa multiplicativa de E (la clave para cifrar que está publicada) módulo $\emptyset(N)$ utilizando el procedimiento mencionado anteriormente [ver en la sección 2.2]. Sin embargo, calcular $\emptyset(N)$ sin factorizar N es por lo menos igual de complicado que factorizar N. En seguida mostramos la razón de esto.

Primero vemos que una vez que se conoce $\emptyset(N)$, se puede factorizar fácilmente N;

$$\emptyset(N)=(P-1)(Q-1)=PQ-P-Q+1$$

$$\therefore \quad P+Q=N-\emptyset(N)+1$$

además P-Q es la raíz cuadrada de $(P-Q)^2$, y $(P-Q)^2$ se puede calcular de manera:

$$(P-Q)^2=P^2+Q^2+2PQ-4PQ=(P+Q)^2-4N$$

$$Q = \frac{(P+Q)-(P-Q)}{2}$$

Como este camino para factorizar N no ha resultado práctico, podemos decir que calcular $\emptyset(N)$ sin factorizar N no puede ser menos complicado que factorizar N.

Ahora consideramos que el criptoanalista trata de calcular la clave para descifrar, D, directamente, sin factorizar N. En el siguiente inciso mostramos que calcular D sin factorizar N tampoco es más fácil que factorizar N.

2.4.3. Determinar D sin factorizar N.

Desde luego, D debe ser seleccionado de un conjunto suficientemente grande para que una búsqueda directa sea ineficiente.

Demostraremos que calcular D no es más fácil para un criptoanalista que factorizar N, puesto que una vez conocida D, N podría ser factorizada fácilmente. Sin embargo, este método para factorizar N tampoco ha resultado fructuoso.

Una D conocida permite que N sea factorizada de la siguiente manera: primero se clacula ED-1, lo cual es un múltiplo de $\emptyset(N)$, recordando que $ED=k\emptyset(N)+1$, donde k es entero. Luego, utilizando el múltiplo de $\emptyset(N)$ ya se puede factorizar N. Por lo tanto, si N es grande, un criptoanalista no debe poder determinar D más fácilmente que factorizar N.

2.4.4. Prueba directa del texto simple.

Como en un sistema de clave pública RSA no hay un canal seguro para que se ponga de acuerdo cada pareja de usuarios en la asignación numérica a cada letra o caracter, esta asignación se tiene que conocer públicamente, y existe un número finito de asignaciones. Si tomamos cada letra del mensaje de texto simple como una unidad, y aplicamos el método mencionado anteriormente para cifrarlo, el texto en cifras tendrá la misma codificación para letras iguales del texto simple (por ejemplo, si la codificación 3792 representa la letra E del texto simple, entonces cada vez que usa la letra E en el texto simple, aparece 3792 en el texto en cifras). Como ya hemos visto, un sistema así no es seguro ni siquiera contra el ataque con solo texto en cifras. Para solucionar este problema, podemos agrupar cada dos letras (incluyendo el espacio) como una unidad de mensaje. Sin embargo, el criptoanalista puede experimentar con las claves públicas, cifrando las combinaciones de dos números que asignan a los caracteres, y perificando con el texto en cifras. Si se necesita 80 caracteres (26 letras mayúsculas, 26 letras minúsculas, 10 números, y símbolos ortográficos, etc.) para comunicarse, hay 80 asignaciones numéricas. El criptoana

lista sólo necesita probar 80^2 veces (hay 80^2 formas de permutación, es la misma razón que el caso de que dos dígitos decimales forman 10^2 permutaciones distintas) a lo máximo para descifrar una unidad de mensaje. Si una prueba tarda 10 microsegundos, 80^2 pruebas tardarán 64 milisegundos. Si un mensaje tiene 2000 letras, tardaría un criptoanalista 64 segundos en descifrar el mensaje.

Para garantizar que el sistema sea seguro contra la prueba directa del texto simple (ataque de texto simple elegido), se necesita tomar más letras como una unidad de mensaje. Por ejemplo, si se agrupan 8 letras, habrá 80^8 formas distintas de permutación para una unidad de mensaje, para esta cantidad de pruebas, un criptoanalista tardaría $80^8 x 10^{-6}/(60x60x24x365)=532$ años (aunque ese tiempo es lo máximo que se tardaría, de todas maneras se tarda bastante tiempo).

De aquí podemos decir que este sistema criptográfico de clave pública es prácticamente seguro.

2.5 EJEMPLO.

Debido a la limitación de cómputo, a continuación mostraremos solamente un ejemplo muy sencillo para explicar el procedimiento de codificación en un sistema criptográfico de clave pública de RSA.

Un usuario U_1 elige dos números primos P=10267, y Q=1571 [19, pág. 870-873]de forma aleatoria, calcula:

$$N=P*Q=10267x1571=16129457$$

$$\phi(N)=(P-1)Q-1)=10266x1570=16117620$$

Escoge otro número aleatorio E=6991847, en donde g.c.d.$(E,\phi(N))=1$, y $2\leq E\leq\phi(N)-1$. Con los números E y $\phi(N)$ calcula D utilizando el algoritmo extendido de Euclides (ver en la sección 2.2), D=1803023. Ahora el usuario U_1 publica sus claves para cifrar E y N, y guardar su clave para descifrar D y la clave para calcular D, $\phi(N)$ en secreto.

Cuando un usuario U_2 quiere mandar un mensaje "EN ESTE TRABAJO, SE

PRESENTA UN PROGRAMA PARA CIFRAR." al usuario U_1 privadamente, él asigna primero a estos caracteres a números: A→65, B→66, ..., Z→90, (espacio) →32, ,→44, .→46, etc. el mensaje queda como:

69 78 32 69 83 84 69 32 84 82 65 66 65 74 79 44 32 83 69 32 80 82 69 83 69 78
84 65 32 85 78 32 80 82 79 71 82 65 77 65 32 80 65 82 65 32 67 73 70 82 65 82 46

Entonces, el usuario U_2 busca en el archivo público las claves para cifrar del usuario U_1 que son: E=6991847, y N=16129457, y con estas claves cifra el mensaje elevando cada número a la potencia E módulo N.

El Texto en cifra es:

11564013 14759908 12899003 11564013 8387544 5933507 11564013 12899603
5933507 15264593 12255266 9930010 12255266 3619211 10653481 7133179
12899603 8387544 11564013 12899603 13666144 15264593 11564013 8387544
11564013 14759908 5933507 12255266 12899603 4035428 14759908 12899603
13666144 15264593 10653481 1061692 15264593 12255266 3647573 12255266
12899603 13666144 12255266 15264593 12255266 12899603 12096351 11448205
7211157 15264593 12255266 15264593 7197176

Ahora si él junta dos números como uno:

6978 3269 8384 6932 8482 6566 6574 7944 3283 6932 8082 6903
6978 8465 3285 7832 8082 7971 8265 7765 3280 6582 6532 6773
7082 6582 4600

El texto en cifra es:

6440961 10836656 6465196 7027870 13263543 4010882 9241976 1426144 14426757 7027870
11292655 4456295 6440961 12615339 12366384 15049686 11292655 14275902 8437915
15528100 2775035 12902545 4039549 11906578 3795392 12902545 14161493

(Nótese que el agrupamiento de caracteres del texto simple, no sólo sirve para la seguridad, sino también para la compresión de datos que transmiten).

El usuario U_2 manda este texto en cifrar al usuario U_1 por un canal público.

Cuando U_1 recibe este mensaje cifrado, la descifrar elevando a cada número a la potencia D módulo N, y obtiene los números agrupados, desagrupándolos y asignando estos números a caracteres, ya obtiene el mensaje

de texto simple:

EN ESTE TRABAJO, SE PRESENTA UN PROGRAMA PARA CIFRAR.

El trabajo de un criptoanalista es factorizar el producto de los números primos. En caso del ejemplo, los números primos tienen tamaño muy pequeño (de 4-5 dígitos), su producto es muy fácil de factorizar, incluso se puede factorizarlo por la división. Pero en el uso real de este sistema, se usan números primos mucho más grandes, así ya es muy difícil factorizar su producto.

III. SISTEMA DE DISTRIBUCION DE CLAVE PUBLICA

El sistema que se trata en este capítulo resuelve el problema de la distribución de la clave de distinta manera que el sistema criptográfico de clave pública que se describió en el capítulo anterior. La meta de este sistema no es separar las claves para cifrar y descifrar, de manera tal que la clave para cifrar puede estar publicada sin comprometer la seguridad de la clave para descifrar, sino que dos usuario A y B se pongan de acuerdo en una clave que se llama la clave común a través de un canal público. O sea, que el usuario A calcula la clave común utilizando la clave pública del usuario B y su propia clave secreta; mientras que el usuario B también usa la clave pública del usuario A y su clave secreta propia. Haciendo operaciones con esas claves cada usuario, llegan al mismo resultado, esto es la clave común, los dos usuarios. Ambos utilizan esa clave común para cifrar y su inversa para descifrar. Cualquier otra persona que no conozca alguna de las claves secretas de A o de B no puede clacular la clave común utilizando únicamente las claves públicas de ambos.

M. Hellman y W. Diffie [3] sugirieron un algoritmo para el sistema de distribución de clave pública, el cual tiene las siguientes ventajas:

1). Requiere solamente una clave pública del otro usuario para poder calcular la clave común.
2). El esfuerzo criptoanalítico crece exponencialmente con respecto al esfuerzo del usuario legítimo.
3). Su uso puede ser ligado con el archivo público de usuarios.

Para poder entender y aplicar este algoritmo se necesitan los siguientes antecedentes.

3.1 Antecedentes Matemáticos.

3.1.1. Definición. Si a es un número entero, la clase de residuo de a módulo n es el conjunto de todos los enteros que tienen el mismo residuo que a cuando se dividen entre n, o sea que son congruentes con

a módulo n. (Esto es que todos los elementos que están en la clase de residuo de a módulo n son congruentes con a módulo n).

3.1.2 Notación. Si n>1 es entero, usamos Z_n para denotar el conjunto de clases de residuo módulo n.

3.1.3 Corolario. Si p es un número positivo primo, Z_p es un campo. (Demostración, ver en [1, pág. 281]).

Por ejemplo, Z_3 es el campo ternario [0, 1, 2], en donde $1+2\equiv 3\equiv 0 \pmod 3$. $2x2\equiv 4\equiv 1 \pmod 3$, $1-2\equiv -1\equiv 2 \pmod 3$, etc.

Este campo también se llama campo de Galois de orden p, y se puede denotar también como GF(p).

3.1.4 Observación. Como GF(p) ó Z_p es un grupo multiplicativo cíclico de orden p-1, $a^p=a$ para cualquier a en z_p. (Esto es justamente el teorema de Fermant: $a^p\equiv a \pmod p$ para cualquier entero a) 14, pág. 457 . Y los elementos de GF(p) son {0, 1, ..., p-1}. Las operaciones suma, resta, multiplicación y división sobre el campo GF(p) se realizan módulo p. Por ejemplo, GF(2) es el campo binario {0,1} $1+1\equiv 2\equiv 0 \pmod 2$, $0-1\equiv -1\equiv 1 \pmod 2$, etc. [8].

3.2 Desarrollo del método de Hellman y Diffie.

Esta técnica aprovecha la dificultad de calcular logaritmos sobre un campo finito (campo de Galois) GF(p) con un número p de elementos. (Los números {0,1,...,p-1}, bajo aritmética módulo p).

Cada usuario puede generar una clave y_i para publicarla, usando una clave secreta x_i generada en forma aleatoria de la siguiente manera:

$$y_i=a^{x_i} \pmod p, \quad \text{para} \quad 1\leq x_i \leq p-1,$$

donde a es un elemento primitivo fijo de GF(p), (esto es que el rango de las potencias de a está entre los elementos diferentes de cero de GF(p), o sea {1,2,...,p-1}).

El cálculo de y a partir de x es computacionalmente fácil siguiendo los siguientes pasos:

Paso 1: Representar x en su forma binaria $x_k x_{k-1} \ldots x_1 x_0$ (donde x_k es el bit más significativo, y x_0 es el bit menos significativo).

Paso 2: Asignar la variable y igual a 1.

Paso 3: Repetir los pasos 3a y 3b para $i=k, k-1, \ldots, 0$;

Paso 3a. Asignar y igual al residuo de y^2 cuando se divide por p.

Paso 3b. Si $x_i=1$, entonces asignar y igual al residuo de $y*a$ cuando se divide por p.

Paso 4: Alto. (Ahora y es igual a $a^x (\text{mod } p)$).

Por ejemplo, si queremos calcular $2^{19} (\text{mod } 7)$ hacemos lo siguiente:

19 en su forma binaria es 10011.

$i=4$: $1^2 (\text{mod } 7) \equiv 1$

$1 \times 2 (\text{mod } 7) \equiv 2$

$i=3$: $2^2 (\text{mod } 7) \equiv 4$

$i=2$: $4^2 (\text{mod } 7) = 2$

$i=1$: $2^2 (\text{mod } 7) = 4$

$4 \times 2 (\text{mod } 7) = 6$

$i=0$: $6^2 (\text{mod } 7) = 1$

$1 \times 2 (\text{mod } 7) = 2$

por lo tanto, $2^{19} (\text{mod } 7) = 2$.

Así el esfuerzo para calcular y a partir de x (que es el esfuerzo del usuario legítimo) a lo más es de $2\lfloor \log_2 p \rfloor$ multiplicaciones (acordándose que $1 \leq x \leq p-1$), donde $\lfloor \log_2 p \rfloor = \lceil \log_2 p \rceil + 1$, o sea la parte entera de $(\log_2 p)$ más 1.

Pero calcular x a partir de y (que es el esfuerzo del criptoanalista) es equivalente a calcular un logaritmo sobre el campo de Galois, o sea

$$x=\log_a y, \text{ sobre } GF(p),$$

y esto puede ser mucho más difícil. Para ciertos valores de p cuidadosamente seleccionado se puede requerir hasta del orden de $p^{1/2}$ operaciones aun usando el mejor algoritmo que se conoce [7]. El origen de su dificultad lo veremos en la sección de Prueba de Seguridad más adelante).

Entonces, cuando dos usuarios i y j se quieren comunicar privadamente, ellos pueden calcular la clave común en forma independiente usando las claves secretas x_i, x_j, y las claves públicas y_j, y_i respectivamente. Esto es equivalente a que el usuario i busque la clave pública del usuario j, y_j, y usando su propia clave secreta x_i calcule la clave común k_{ij} por medio de:

$$k_{ij} \equiv y_j^{x_i} \pmod p.$$

El usuario j calcula k_{ji} del mismo modo, o sea:

$$k_{ji} \equiv y_i^{x_j} \pmod p.$$

Se puede ver que k_{ij} es igual a k_{ji}, esto es,

$$k_{ij} \equiv y_j^{x_i} \pmod p \equiv (a^{x_j})^{x_i} \pmod p$$
$$\equiv a^{x_j x_i} \pmod p \equiv (a^{x_i})^{x_j} \pmod p$$
$$\equiv y_i^{x_j} \pmod p \equiv k_{ji}$$

Una vez que encuentren la clave común k_{ij} o k_{ji}, la pueden utilizar como la clave para cifrar, y la inversa de k_{ij}, como la clave para descifrar, esto es, la pueden utilizar como una clave en un sistema criptográfico convencional.

Sin embargo, ¿cuál es la mejor utilización de esta clave para cifrar un mensaje del texto simple?. En la siguiente sección daremos una solución de esta pregunta.

3.3 Realización de la clave común por elevación exponencial.

Una vez que obtiene los usuarios i y j su clave común k_{ij}, la pueden utilizar para cifrar mensajes de texto simple de cualquier manera. Esto es que pueden hacer cualquier operación sobre su texto simple usando su clave. Por ejemplo, pueden sumar o multiplicar el texto simple con la clave. Sin embargo, observamos que el texto en cifras así obtenido no es seguro contra los ataques criptoanalíticos (e.g., en el caso de multiplicación, el criptoanalista puede buscar un factor común de las unidades del mensaje de texto en cifras para romper la seguridad del sistema). Sobre todo no está seguro contra un ataque de texto simple conocido. Para resolver este problema, se cifra un mensaje de texto simple elevándolo a la potencia k_{ij}.

Sean el mensaje de texto simple, la clave, y el texto en cifras denotados por M, k_{ij}, y C respectivamente, con las siguientes restricciones:

$$1 \leq M \leq p-1,$$
$$1 \leq C \leq p-1,$$
$$1 \leq k_{ij} \leq p-2,$$
y $$\gcd(k_{ij}, p-1) = 1,$$

donde p es un número primo grande (acordándose que $\phi(p)=p-1$, si p es un primo). La primera restricción sirve para que se puede aplicar el teorema de Euler, en donde M tiene que ser relativamente primo con respecto p. La segunda restricción se cumple al realizar módulo p. La tercera restricción sirve para que la inversa de k_{ij} exista, donde k_{ij} tiene que ser relativamente primo que $\phi(p)$, que es la última restricción, obviamente que k_{ij} tiene que ser no igual a y menor que p-1, esto es que k_{ij} tiene que ser menor o igual a p-2.

Entonces, se cifra el mensaje de texto simple de la siguiente manera:

$$C \equiv M^{k_{ij}} \pmod{p}.$$

El receptor del mensaje, al recibir C, lo puede descifrar con la inversa

de la clave común, k_{ij}^{-1} aplicando el teorema de Euler (ver en el teorema 11 de la sección 2.1.1):

$$M \equiv C^{k_{ij}^{-1}} (\text{mod } p) \equiv (M^{k_{ij}})^{k_{ij}^{-1}} (\text{mod } p) \equiv M (\text{mod } p),$$

donde $k_{ij} k_{ij}^{-1} \equiv 1 (\text{mod } \phi(p))$.

Para calcular k_{ij}^{-1} a partir de k_{ij} y $\phi(p)$ se sigue el mismo proceso que en el cálculo de D (la clave para descifrar) utilizando la clave para cifrar E y la clave secreta $\phi(N)$ en el sistema criptográfico de clave pública descrito en el capítulo 2.

A continuación mostramos que la realización de la clave común por la elevación exponencial forma un sistema criptográfico que es seguro contra el ataque de texto simple conocido.

Si un criptoanalista tiene la ventaja de conocer la pareja texto simple - texto en cifras, o sea una pareja correspondiente de M y C, y quiere calcular la clave común k_{ij}, tiene que hacer la siguiente operación:

$$k_{ij} \equiv \log_M C, \text{ sobre } GF(p),$$

lo cual es un cálculo de logaritmos sobre el campo de Galois GF(p). Ya se había mencionado que es muy difícil su cálculo, aún usando el mejor algoritmo para ello.

Sin embargo, nadie ha sido capaz de demostrar que el sistema anterior puede resistir un ataque de texto simple elegido en donde el criptoanalista escoge M y k_{ij}, y verifica con su correspondiente C; pero tampoco se encuentra mayor ventaja de que él use esta opción a que use un ataque de texto simple conocido.

Después de haber estudiado este método, podemos observar que éste tiene una gran desventaja. Esto es que los valores de k_{ij} y p-1 tienen que ser relativamente primos para que la inversa de la clave común, k_{ij}^{-1} (que es la clave para descifrar) exista. Esta condición quizás no se cumpla para $k_{ij} \equiv y_j^{x_i} (\text{mod } p)$ calculada con una y_j dada en el archivo público y una

x_i fija (nótese que se había escogido x_i aleatoriamente para calcular y_i, pero una vez que y_i está publicada, x_i ya está fija). Una forma de proceder es que el usuario i selecciona una x_i de tal manera que $\gcd(k_{ij},p-1)=1$, para un valor dado de y_j, y después publica la y_i calculada con x_i así elegida. Pero ésto causa retardo en la comunicación, y además, para comunicarse con otro usuario k, es necesario que el usuario i determine una nueva x_i, para que $\gcd(k_{ik},p-1)=1$ se cumpla, donde $k_{ik}\equiv y_k^{x_i}(\text{mod } p)$. Esto significa que el usuario i tiene que publicar una y_i nueva. Esto implica que para distintos usuarios con quien desea comunicarse el usuario i, tiene que elegir diferentes x_i, calcular y_i correspondiente y publicarla.

Una regla muy importante en la criptografía es que en un sistema criptográfico debe ser fácil modificar la clave, una vez que está comprometida. Aunque este sistema cumple esta regla, o sea que es fácil modificar las claves en un archivo público, poner distintas y_i para distintos usuarios con quien quiere comunicarse, no puede ser un acceso práctico. Para resolver este problema, comentaremos en la siguiente sección otra forma de realizar la clave común, la cual fue desarrollada por los autores de este trabajo.

3.4 Realización en forma matricial.

Se define una matriz A de orden nxn y una matriz X del mismo orden que A, en donde los elementos de A son elementos primitivos de $GF(p)$, y los elementos de X se eligen aleatoriamente. (La matriz A se conoce públicamente).

Entonces, cada usuario genera su clave pública Y, donde Y es una matriz de orden nxn que tiene los elementos definidos por:

$$y_{ij} \equiv a_{ij}^{x_{ij}}(\text{mod } p), \quad i,j=1,2,\ldots,n.$$

y se denota como:

$$Y \equiv A^X(\text{mod } p).$$

Este usuario publica su matriz Y, y mantiene la matriz X en secreto.

Cualquier otra persona no puede calcular X a partir de Y por la dificultad de calcular logaritmos sobre $GF(p)$.

Cuando dos usuarios U_1 y U_2 se quieren comunicar privadamente, ellos calculan su clave común de la siguiente manera:

$$K_{12} \equiv Y_2^{X_1} (\text{mod } p) \equiv Y_1^{X_2} (\text{mod } p),$$

donde Y_1, Y_2 son las matrices públicas, X_1, X_2 las matrices secretas de los usuarios U_1 y U_2 respectivamente. K_{12} tiene la dimensión igual que la matriz Y que es de dimensión nxn. Si k_{12} es no singular, se puede cifrar un mensaje de texto simple M que tiene forma matricial de orden nxr, para obtener la matriz de texto en cifras de orden nxr:

$$C = K_{12}M.$$

Para que se puedan multiplicar las matrices K_{12} y M, la matriz M tiene que tener dimensión adecuada, es decir, M tiene que ser de dimensión nxr, donde n es el orden de la matriz cuadrada K_{12}. Esto es que M tiene nxr elementos. Si el número de elementos de la información es menor que nxr, o sea que este número es igual a $nxr-s$ (donde $s<n$), entonces se asignan números aleatorios a los s elementos que no tenían información.

Para descifrar el mensaje cifrado C, se premultiplica a la matriz C, la matriz inversa de K_{12}, K_{12}^{-1}, y así se obitne M:

$$M = K_{12}^{-1}C = K_{12}^{-1}(K_{12}M) = M.$$

En este caso, la matriz de clave común también tiene la restricción de que K_{12} tiene que ser no singular para que K_{12}^{-1} exista. Sin embargo, esta restricción es mucho más fácil de cumplir comparada con la restricción de que $\gcd(k_{ij},p-1)=1$ para el caso de la realización de la clave común por elevación exponencial descrita en la sección anterior. Esto es debido a que la posibilidad de que el determinante de K_{12} sea igual a cero es demasiado pequeña (casi nula) comparada con la posibilidad de que el determinante de K_{12} sea cualquier número diferente de cero, para una matriz K_{12}

que tiene elementos calculados a partir de otros elementos seleccionados aleatoriamente [16, pág. 28].

Ahora mostramos que este sistema es seguro contra el ataque de texto simple conocido, cuando n es mayor que r, donde n es número de renglones, y r, número de columnas de M.

Cuando un criptoanalista conoce los correspondientes M y C, los cuales están relacionados por la expresión:

$$C = K_{12}M,$$

él puede calcular la clave común de los usuarios U_1 y U_2, K_{12}, solamente cuando la matriz M tiene rango n. Esto es que $r \geq n$. Cuando r=n, lo más seguro es que M tiene rango n, o sea M es una matriz no singular, porque M tiene elementos cualesquiera (se puede decir aleatorios) [16, pág. 28]. Entonces, el criptoanalista puede calcular K_{12} de forma:

$$K_{12} = CM^{-1}.$$

Cuando $r > n$, M tiene inversas derechas [9, pág. 765], o sea que existe M^R (M^R no es única), tal que

$$\underset{nxr}{M} \times \underset{rxn}{M^R} = \underset{nxn}{I_n}$$

El criptoanalista puede calcular una inversa derecha M^R (cualquiera de ellas), y postmultiplica a la matriz C, esa M^R para obtener K_{12}:

$$\underset{nxr}{C} \times \underset{rxn}{M^R} = \underset{nxn}{K_{12}MM^R} = K_{12}I_n = K_{12}.$$

Sin embargo, cuando $r < n$, el criptoanalista ya no puede calcular K_{12} tan fácilmente. Esto es debido a que: en este caso, M tiene inversas izquierdas (existe un número infinito de esas inversas), o sea que existe M^L, tal que:

$$\underset{rxn}{M^{L}} \underset{nxr}{xM} = \underset{rxr}{I_r}$$

(Nótese que $MxM^{L} \neq I$, ni siquiera es no singular, porque tiene a lo máximo rango r, y r es menor que n).

Si el criptoanalista calcula una M^{L} cualquiera, y postmultiplica a la matriz C, esa M^{L}, le da:

$$\underset{nxr}{Cx} \underset{rxn}{M^{L}} = \underset{nxr}{K_{12}} \underset{rxn}{MM^{L}} = \underset{nxn}{K'_{12}}.$$

Pero como M^{L} no es única, K'_{12} tampoco es única (son diferentes K'_{12} para diferentes M^{L}). La posibilidad de que $K'_{12}=K_{12}$ es muy pequeña. Si él premultiplica a la matriz C, la M^{L} que calculó:

$$\underset{nxn}{M^{L}} \underset{nxr}{xC} = \underset{rxn}{M^{L}} \underset{nxr}{K_{12}M} = \underset{rxr}{K''_{12}}$$

K''_{12} nunca es igual a K_{12}, porque no tienen la misma dimensión.

Por lo tanto, cuando $r < n$, aunque el criptoanalista conozca la pareja texto simple-texto en cifras, no puede calcular fácilmente la clave común de los usuarios U_1 y U_2, K_{12}.

3.5 Prueba de Seguridad

La seguridad de este sistema depende principalmente de la dificultad del cálculo de logaritmos sobre el campo de Galois. Si este cálculo es computacionalmente fácil, entonces la seguridad del sistema se puede romper fácilmente. Sin embargo, si se elige cuidadosamente el número primo p, ni son el mejor algoritmo que se conoce se puede hacer este cálculo fácilmente [7].

Para comparar un algoritmo con otro algoritmo que resuelve el mismo problema, se usa un criterio llamado complejidad, el cual se clasifica en:

complejidad de tiempo y complejidad de espacio [18, pág. 2]. La complejidad de tiempo de un algoritmo es el tiempo que se necesita el algoritmo para resolver el problema, expresado como una función del tamaño de este problema. De igual manera se define la complejidad de espacio de un algoritmo, que es el espacio que se necesita el algoritmo para las memorias, expresado como una función del tamaño del problema. Si n es el tamaño del problema, y el algoritmo procesa este problema en tiempo cn^2, para alguna constante c, entonces, se dice que la complejidad de tiempo de este algoritmo es de orden n^2, y se denota como $O(n^2)$. Más precisamente, una función $g(n)$ se dice que es de orden $f(n)$, o sea, $O(f(n))$, si existe una constante c, tal que $g(n) \leq cf(n)$, para toda n no negativa. Y para la complejidad de espacio se tiene la misma notación.

En este caso, el algoritmo óptimo para el cálculo de logaritmos sobre el campo de Galois se desarrolló por S.C. Pohlig y M.E. Hellman [7]. (Su desarrollo en detalle se presenta en [7, pág. 108-110]). Ahora presentamos un teorema (para su demostración, ver en [7, pág. 110]) que habla de la complejidad de ese cálculo usando este algoritmo.

3.5.1 Teorema. Considérese que

$$p-1=p_1^{n_1}p_2^{n_2}\cdots p_k^{n_k}, \qquad p_1 < p_{i+1}$$

sea la factorización prima de $p-1$, donde p es primo, los p_i son diferentes primos, y $n_i \geq 1$. Entonces, para cualquier $\{r_i,\ i=1,2,\ldots,k\}$, con $0 \leq r_i \leq 1$, logaritmos sobre $GF(p)$ se puede calcular en

$$O\left(\sum_{i=1}^{k} n_i\left(\log_2 p + p_i^{1-r_i}\left(1+\log_2 p_i^{r_i}\right)\right)\right)$$

operaciones y con

$$O\left(\log_2 p \sum_{i=1}^{k}\left(1+p_i^{r_i}\right)\right)$$

bits de memoria.

De este teorema podemos observar que si p es tal que todos los p_i

son pequeños entonces, el esfuerzo de computación requerido para calcular logaritmos sobre el campo de Galois es aproximadamente $\log_2 p$, y este esfuerzo es el del criptoanalista. Acordándose que el esfuerzo para descifrar del usuario legítimo es $2^{\lceil \log_2 p \rceil}$. Como el esfuerzo del criptoanalista debe ser mucho mayor (e.g., 10^9 veces mas grande) que el esfuerzo del usuario legítimo para descifrar el mensaje cifrado, el sistema criptográfico de distribución de clave pública de Hellman y Diffie tiene que evitar el uso de tales valores de p. Sin embargo, cuando p_k, el mayor factor primo de $p-1$, es comparable con el tamaño de p (e.g., $p-1=2p_k$), entonces, este algoritmo ya no es más eficiente que los otros previamente conocidos.

De aquí podemos concluir que cuando $p-1$ tiene factores primos grandes, ni el mejor algoritmo puede ser eficiente para calcular logaritmos sobre $GF(p)$.

Sin embargo, nadie ha demostrado que si logaritmos sobre $GF(p)$ son difíciles de calcular, el sistema es seguro, pero tampoco se han encontrado un camino de calcular la clave común de un par de usuarios i y j, k_{ij}, a partir de las claves públicas y_i y y_j sin obtener primero una de las claves secretas x_i o x_j. Por lo tanto se puede aprovechar su seguridad aparente por el momento.

3.6 EJEMPLOS

Primero vemos un ejemplo sencillo del proceso de codificación utilizando el método de realización de la clave común por la elevación exponencial.

Se supone que en un sistema de distribución de clave pública, dos números enteros públicamente conocidos del sistema α y q tienen los siguientes valores:

$$\alpha = 19 \quad , \quad Q = 19423$$

donde q es un número primo [19, pág. 870-875] y $1<\alpha<q$.

Un usuario U_1 elige un número aleatorio x_1=541, y lo guarda en secreto.

El calcula su clave pública Y_1:

$$y_1 \equiv \alpha^{x_1} (\text{mod } q) \equiv 19^{541} (\text{mod } 19423) \equiv 6074,$$

y publica esa y_1.

Un usuario U_2 quiere mandar al usuario U_1 un mensaje "EN ESTE TRABAJO, SE PRESENTA UN PROGRAMA PARA CIFRAR." El busca en el archivo público la clave y_1=6074 U_2 calcula la clave común k_{12}:

$$k_{12} = y_1^{x_2} (\text{mod } q).$$

con una x_2 elegida aleatoriamente opera x_2 tiene que ser de tal manera que $\gcd(k_{12}, q-1)=1$ (nótese que ésto es una desventaja de esta realización de clave común, ver en la sección 3.3). Si una x_2=3923 fue elegida, (x_2 cumple esa condición, i.e., gcd(3923, 19 a 22) entonces

$$k_{12} = 6074^{3923} (\text{mod } 19422) = 15451$$

Después calcula su clave pública y_2,

$$y_2 = \alpha^{x_2} (\text{mod } q) = 19^{3923} (\text{mod } 19423) = 11913.$$

y la publica.

Ahora U_2 puede cifrar su mensaje asignando los caracteres a números juntando 2 números en un grupo, elevando cada grupo a la potencia k_{12} módulo q:

E	N		E	S	T	E		T	R	A	B	A	J	O	,	.	S	E		P	R	E	S	E	N
69	78	32	69	83	84	69	32	84	82	65	66	65	74	79	44	32	83	69	32	80	82	69	83	69	78

T	A		U	N		P	R	O	G	R	A	M	A		P	A	R	A		C	I	F	R	A	R	.	
84	65	32	85	78	32	80	82	79	71	82	65	77	65	32	80	65	82	65	32	67	73	70	82	65	82	46	00

El texto en cifras es:

18328 17841 4166 14876 14730 10145 17001 2106 7581 14876 4531 3954 18328 17189
7695 3936 4531 19408 11593 9769 12512 12252 8415 880 14269 12252 9153

Cuando el usuario U_1 recibe el texto en cifras del U_2, él también busca y_2 en el archivo público, $y_2 = 11913$, y calcula k_{12}:

$$k_{12} \equiv y_2^{x_1} (\text{mod } q) \equiv 11913^{541} (\text{mod } 19423) \equiv 15451$$

Con k_{12} y q-1 calcula k_{12}^{-1} usando el algoritmo de Euclides (ver en 2.2).

$$k_{12}^{-1} = 7087 \text{ (mod } 19423)$$

Ahora descifra este mensaje de texto en cifras elevando cada número a la potencia k_{12}^{-1}, obtiene los números agrupados, desagrupándolos y asignándolos a los caracteres y ya detiene el mensaje de texto simple:

EN ESTE TRABAJO, SE PRESENTA UN PROGRAMA PARA CIFRAR.

Ahora vemos un ejemplo muy pequeño de procedimiento de cifrar un mensaje de texto simple realizando la clave común en forma matricial.

Con dos números públicamente conocidos α=19 y q=10247, los usuarios U_1 y U_2 eligen sus matrices secretas en forma aleatoria:

$$X_1 = \begin{bmatrix} 193 & 127 \\ 71 & 199 \end{bmatrix}, \qquad X_2 = \begin{bmatrix} 9306 & 3923 \\ 9306 & 9408 \end{bmatrix}$$

y calculan sus matrices públicas, con la fórmula $y_{ij} \equiv \alpha^{x_{ij}} (\text{mod } q)$, i,j = 1, 2

$$Y_1 = \begin{bmatrix} 1755 & 7681 \\ 8236 & 998 \end{bmatrix}, \qquad Y_2 = \begin{bmatrix} 5046 & 751 \\ 5046 & 864 \end{bmatrix}$$

respectivamente. Cuando U_1 y U_2 se quieren comunicar privadamente,

ellos calculan la matriz de clave común k_{12} independientemente.

$$k_{ij}(1,2) = y_{ij}(1)^{x_{ij}(2)} \pmod q$$

$$= y_{ij}(2)^{x_{ij}(1)} \pmod q$$

$$k_{12} = \begin{bmatrix} 9123 & 2003 \\ 5305 & 2213 \end{bmatrix}$$

Ahora el usuario U_1 manda un mensaje 'CODIGO' al usuario U_2, él hace lo siguiente:

C O D I G O
67 79 68 73 71 79

$$M = \begin{bmatrix} 677968 \\ 737179 \end{bmatrix}$$

$$C = k_{12} M = \begin{bmatrix} 9123 & 2003 \\ 5305 & 2213 \end{bmatrix} \begin{bmatrix} 677968 \\ 737179 \end{bmatrix} = \begin{bmatrix} 7661671601 \\ 5227997367 \end{bmatrix}$$

Cuando el usuario U_2 recibe el mensaje cifrado C, lo descifra de modo:

$$M = k_{12}^{-1} C = \begin{bmatrix} 0.23145\text{x}10^{-3} & -0.209449\text{x}10^{-3} \\ 0.554725\text{x}10^{-3} & 0.953961\text{x}10^{-3} \end{bmatrix} \begin{bmatrix} 7661671601 \\ 5227997367 \end{bmatrix}$$

$$M = \begin{bmatrix} 677968 \\ 737179 \end{bmatrix}$$

lo desagrupa y obtiene el mensaje de texto simple: CODIGO.

En uso real de este sistema la matriz k es de orden mucho más grande (e.g., de orden 100), entonces, el mensaje también puede tener un tamaño mucho más grande que el del ejemplo anterior, siempre y cuando tenga menos columnas k, para que el sistema sea seguro contra el ataque de texto simple conocido. Y para que haya menos errores en el procedimiento de de

codificación, es conveniente utilizar 'DOUBLE PRECISSION' en la realización de convertir la matriz k.

IV. SISTEMA AUXILIAR PARA EL SISTEMA CRIPTOGRAFICO DE CLAVE PUBLICA

El sistema criptográfico de clave pública que se desarrolló en el capítulo 2 tiene como desventaja principal el manejo de números muy grandes. En ese sistema, un mensaje se representa como una sucesión de enteros M_1, M_2, ..., con cada M_i un entero entre 0 y N-1 (aproximadamente 10 dígitos), donde N es el producto de dos números primos muy grandes (100 dígitos). La codificación de este mensaje consiste en elevar cada bloque M a la potencia E módulo N, siendo E mayor o igual que 2, pero menor o igual que $\emptyset(N)-1$. Si N es de 200 dígitos (debido a que es un producto de dos números de 100 dígitos). $\emptyset(N)$ también es de 200 dígitos, entonces E puede llegar a ser también de este orden. Elevar número a la potencia E es un trabajo muy laborioso aunque se puede aligerarlo aplicando el teorema 9 del capítulo 2, o usando el algoritmo que se describió en la sección 3.2. Usando este algoritmo para elevar un número a la potencia de un número de 200 dígitos, el esfuerzo (del usuario legítimo) es aproximadamente 665 (que es $\log_2 10^{200}$) multiplicaciones y 665 divisiones [6, pág.]. Por esta razón, los autores de este trabajo desarrollaron un sistema auxiliar para aligerar aún mas este trabajo. Con este sistema auxiliar se puede cifrar la mayoría del texto simple, y la minoría de este texto se sigue cifrando con el mismo sistema (el sistema criptográfico de clave pública RSA), pero ya se logra una gran disminución del número de las veces que tenían que elevar los números a la E módulo N. En el presente capítulo describimos la consistencia de este sistema auxiliar, su seguridad así como las ventajas y las desventajas que tiene.

4.1 Descripción del Sistema.

En base a que una matriz no cuadrada tiene infinidad de inversas, o sea que es imposible resolver un conjunto de ecuaciones simultáneas en forma única cuando el número de incógnitas excede al número de ecuaciones. Así podemos cifrar la información, multiplicando una tal matriz de orden adecuada por el vector que representa la información.

Cada usuario genera una matriz rectangular A con números enteros aleatorios de orden m*n, donde m es menor que n, y podemos usar

m=n-1 por conveniencia. Publica su matriz A junto con sus claves públicas E y N, las cuales son utilizadas en el sistema criptográfico de clave pública RSA. Si un usuario U_1 quiere mandar un mensaje a un usuario U_2, él busca en el archivo público y obtiene la matriz A y las claves E y N del usuario U_2. Ahora asigna los caracteres de su mensaje (letras, números y símbolos de ortografía, etc.) por ciertos números. Por ejemplo, si usa la función ORD del lenguaje PASCAL, las letras de A a Z estarían asignadas por los números de 65 a 90. Y esta asignación debe ser conocida públicamente debido a que no hay ningún canal seguro para que dos usuarios se pongan de acuerdo en alguna asignación númerica especial. Posteriormente junta estos números en grupos de un tamaño adecuado (e.g., 10 números en un grupo). Considérese el conjunto de estos grupos $\{G_1, G_2, \ldots, G_r\}$ como vectores x_i, i=1,2,...,k+1.

$$X_1 = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix}, \quad X_2 = \begin{bmatrix} G_{n+1} \\ G_{n+2} \\ \vdots \\ G_{2n} \end{bmatrix}, \ldots\ldots, \quad X_{k+1} = \begin{bmatrix} G_{kn+1} \\ G_{kn+2} \\ \vdots \\ G_{(k+1)n} \end{bmatrix}$$

donde k es un número entero positivo, y n es el número de columnas de la matríz A. En caso de que r no sea múltiplo de n, se asignan números aleatorios a los elementos $\{G_{r+1}, \ldots G_{(k+1)n}\}$. Se cifra estos vectores premultiplicando la matriz A, y asi ya se obtiene el texto en cifras y_i, i=1,2,...,k+1.

$$Y_1 = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix}, \quad Y_2 = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_{2n} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} G_{n+1} \\ G_{n+2} \\ \vdots \\ G_{2n} \end{bmatrix},$$

$$, \ldots, Y_{k+1} = \begin{bmatrix} y_{kn+1} \\ y_{kn+2} \\ \cdot \\ y_r \\ \cdot \\ y_{(k+1)n} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} G_{kn+1} \\ G_{kn+2} \\ \cdot \\ G_r \\ \cdot \\ G_{(k+1)n} \end{bmatrix}$$

El usuario U_1 ya puede mandar este texto en cifras y_i, $i=1,2,\ldots,k+1$, al usuario U_2 a través de un canal público. Pero para que el usuario U_2 pueda descifrar este tanto, el usuario U_1 le tiene que mandar además la codificación de los primeros elementos de cada vector x_i usando las claves públicas E y N, o sea cifrando estos elementos de la siguiente manera:

$$C_1 = G_1^E \pmod N, \qquad C_2 = G_{n+1}^E \pmod N, \ \ldots, \ C_{k+1} = G_{kn+1}^E \pmod N$$

y mandar estos valores de C_i al U_2 también.

De aquí podemos observar que en lugar de elevar cada G_i a la E módulo N, o sea, en lugar de hacer r veces la exponenciación módulo N, nada más tiene que hacer ésto k+1 veces, donde $k*n+1 < r < (k+1)n$. Es decir, que k es del orden de r/n. Como n es el número de columnas de la matriz A, n es un entero positivo, y obviamente k es mucho menor que r, y mientras que m sea mayor, k es menor. Podemos concluir que si la dimensión de la matriz A. (n-1)*n es suficientemente grande (e.g., n=100) se reduce el número de veces que se tiene que elevar números grandes $\{G_i, i=1,n+1,\ldots,kn+1\}$ a un número mucho más grande todavía E módulo N, lo cual es mucho más laborioso que hacer las multiplicaciones.

Cuando el usuario U_2 recibe el texto en cifras y_i y C_i, $i=1,2,\ldots,k+1$, donde los y_i son vectores de números enteros y los C_i son números enteros, descifra primero los C_{ij} con su clave secreta D, de la siguiente manera:

$$C_i^D \pmod N = G_{(i-1)n+1}^{ED} \pmod N = G_{(i-1)n+1}^{k\phi(N)+1} = G_{(i-1)n+1} \pmod N, \ i=1,2,\ldots, k+1.$$

Como N es mucho mayor que $G_{(i-1)n+1}$ y además N y G_i son relativamente primos entonces:

$$G_{(i-1)n+1} \pmod N = G_{(i-1)n+1}, \quad i=1,2,\ldots,k+1.$$

Y luego, con estos valores obtenidos $\{G_1, G_{n+1}, \ldots, G_{kn+1}\}$ sustituyendo en los sitemas:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ \cdot \\ G_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ y_n \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} G_{n+1} \\ G_{n+2} \\ \cdot \\ G_{2n} \end{bmatrix} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \cdot \\ y_{2n+1} \end{bmatrix}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} G_{kn+1} \\ G_{kn+2} \\ \cdot \\ G_{(k+1)n} \end{bmatrix} = \begin{bmatrix} y_{kn+1} \\ y_{kn+2} \\ \cdot \\ y_{(k+1)n} \end{bmatrix}$$

respectivamente. Vemos que ahora en cada sistema hay $n-1$ ecuaciones y $n-1$ incógnitas, lo cual tiene una solución unica, y esto se puede resolver fácilmente. Existen muchos métodos para resolver estos sistemas. Uno de ellos es el método de la eliminación de Gauss [12, pág.]. Asi el usuario U_2 puede descifrar resolviendo estos sistemas, obteniendo los G_i, desagrupándolos y asignándolos por las letras u otros caracteres.

4.2 Prueba de Seguridad.

La seguridad del sistema criptográfico de clave pública (como ya se había visto en el capítulo 2), ahora solamente vamos a discutir la seguridad del sistema auxiliar.

El sistema auxiliar que es de la forma:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} G_{(i-1)n+1} \\ G_{(i-1)n+2} \\ \vdots \\ G_{(i-1)n+n} \end{bmatrix} = \begin{bmatrix} Y_{(i-1)n+1} \\ Y_{(i-1)n+2} \\ \vdots \\ Y_{(i-1)n+n} \end{bmatrix}, \quad i=1,2,\ldots,k+1$$

con cada subsistema de n-1 ecuaciones con n incógnitas. Para resolver estos sistemas, se puede asignar arbitrariamente un valor a la variable libre y obtener una solución del sistema. Como se puede asignar una infinidad de valores diferentes a la variable libre, el sistema tiene una infinidad de soluciones. El criptoanalista puede asignar valores supuestamente correctos a la variable libre, obtener soluciones diferentes del sistema, asignar los números por las letras y ver si el mensaje así obtenido tiene algún significado. Pero para que encuentre la solución correcta con pocas pruebas, la probabilidad es muy pequeña (casi cero). Aun con 10^9 pruebas, es posible que no encuentre la solución correcta.

Como las asignaciones numéricas son conocidas públicamente, y existe un número finito de asignaciones numéricas (por ejemplo: las 26 letras se asignan por los números de 65 a 90), si ciframos el texto simple letra por letra, o sea tomando cada G_i igual a la asignación numérica de una letra, entonces el criptoanalista puede usar un ataque de texto simple elegido para descifrar el mensaje, y su esfuerzo es solamente hacer 26 pruebas. Para evitar un ataque como este, agrupamos muchas letras como una unidad para cifrarla. Esto quiere decir que cada G_i corresponde a la asignación de muchas letras. Así la dificultad del criptoanalista para romper el código es mayor. Por ejemplo, si agrupamos 10 letras o caracteres para una G_i el criptoana

lista tendrá que probar por lo menos 26^{10} veces para descifrar el mensaje (decimos "por lo menos" por que ahora solamente consideramos 26 letras, nos falta considerar los números, los caracteres ortográficos, las letras mayúsculas y minúsculas, etc.). Este problema de seguridad lo hemos visto también en el sistema criptográfico de clave pública en el capítulo 2. Podemos observar que es un problema común para los dos sistemas. Como en un sistema criptográfico de clave pública se cifra el mensaje con la clave pública, o sea que en ese caso, el criptoanalista conoce la clave para cifrar y el texto en cifras, entonces puede probar con un mensaje supuesto cifrándolo con la clave pública para cifrar y comparando con el texto en cifras. Por lo tanto, en el sistema criptográfico de clave pública que se describió en el capítulo 2, en donde cada usuario genera un par de claves, una de ellas se publica, y la otra se mantiene en secreto, y no se puede calcular la clave secreta a partir de la clave pública, la clave pública sirve para cifrar, y la secreta para descifrar. Entonces siempre se tendrá que agrupar los caracteres o letras para que el sistema sea seguro contra ese tipo de ataques.

Sin embargo, este sistema tiene la desventaja de que hay que cifrar dos veces una parte del texto simple, aunque esta parte es muy pequeña. Y tal vez por esto causa alguna inseguridad para el sistema, pero hasta el momento todavía no la vemos.

4.3 Conclusión.

Con este sistema auxiliar aplicado en un sistema criptográfico de clave pública de RSA, se puede disminuir los trabajos laboriosos de elevación de numeros grandes. Considérese que un mensaje de texto simple tiene r números que cifrar. Entonces, si se usa solamente el sistema de clave pública de RSA para cifrar este mensaje, se tiene que elevar los r números a la potencia E ($E<\emptyset(N)-1$. $\emptyset(N)$ es de 200 dígitos), y su esfuerzo es de $665*r$ multiplicaciones y $665*r$ divisiones (aunque este esfuerzo es a lo máximo, porque E tiene número de dígitos menor o gual que 200). En cambio, si se aplica el sistema auxiliar en el sistema de RSA, se tiene que elevar solamente $k+1<<r$ números a la potencia E (donde $k+1=r/n$, siendo n el número de columnas de la matriz A), y hacer $n(n-1)k+1)=(n-1)r$ multiplicaciones. Esto es que, el total esfuerzo del último es $2*665(r/n)+r(n-1)=$

[2*665/n+(n-1)] r multiplicaciones y divisiones.

Si elegimos n cuidadosamente, tal que

[2*665/n+(n-1)]r < 2*665*r.

esto es que

2*665 > 2*665/n+(n-1)

2*665*n > 2*665+n*n-n

n*n-(2*665+1)*n+2*665 < 0

1 < n < 1330

Así es que si la dimensión de la matriz A que es (n-1)xn es adecuada, o sea que si n es mucho menor que 1330, digamos que n es alrededor de 100, entonces, el esfuerzo de codificación aplicando el sistema auxiliar en el sistema RSA es aproximadamente de [2*665/100+(100-1)]r=112r multiplicaciones y divisiones, lo cual es casi 12 veces menor que el esfuerzo de codificación sin aplicar este sistema auxiliar en el sistema RSA.

V. CONCLUSION

Los sistemas de clave pública para la criptografía que hemos desarrollado en los capítulos anteriores resuelven el problema de la distribución de claves que siempre ha sido una gran limitación en el uso de la tecnología criptográfica convencional. Estos sistemas son computacionalmente seguros que sirven para aumentar la privacía. Tiene muchas ventajas comparándolos con los sistemas convencionales. En primer lugar, en un sistema criptográfico convencional, el transmisor necesita mandar la clave por un canal seguro, como por ejemplo; por un correo certificado, o un mensajero de mucha confianza. Y para poder usar un canal seguro (por ejemplo: contratar un mensajero hábil), podría resultar muy costoso, en caso de guerra o en otros casos militares muy importantes, tal vez le puede costar hasta la vida el mensajero. En cambio, en un sistema de clave pública, mandar a publicar la clave en el archivo público podría resultar mucho menos costoso. En segundo lugar, en un sistema convencional, la seguridad del sistema depende mucho de la seguridad del canal seguro, y ese canal puede ser supuestamente seguro, o sea que existe la inseguridad. Por ejemplo, en el caso de mensajero, se le puede robar la libreta de claves al mensajero, o lo puede amenazar de alguna manera para que le diga las claves al oponente, y en el caso del correo certificado, el oponente podría leerlo ilegalmente. Pero en un sistema de clave pública, como no se necesita usar ningún canal seguro, se evita el riesgo que se puede tener en el canal seguro, aunque la seguridad del sistema no es incondicional, es computacional, nadie puede encontrar la clave secreta a partir de la clave pública, aún usando las computadoras. Con que se logre esto, ya es suficiente para proteger las comunicaciones en cuanto a su privacía. En tercer lugar, mandar la clave por un canal seguro causa retardos para las comunicaciones. Esto es, mientras que la clave sea distribuida, el transmisor de mensajes tiene que esperar o mandar la clave de antemano, y en los sistemas de clave pública no existe este problema. Las claves ya están publicadas, será cuestión de buscarlas en el archivo público, lo cual no causa tanto retardo. En cuarto lugar, en un sistema convencional, el transmisor tiene que mandar la clave a los receptores uno por uno, y en un sistema de clave pública, con solo publicar la clave, la obtienen todos, lo cual es mas eficiente y requiere menos esfuerzo. Por último, cuando las claves están comprometidas, deben ser cambiadas fácilmente (recordando una regla muy importante de la criptografía),

y cambiar la clave pública en un archivo público es más fácil que cambiar la clave y mandarla de nuevo por un canal seguro.

De aquí vemos la importancia de desarrollar los sistemas nuevos de criptografía de clave pública. Los sistemas que se han presentado en este trabajo tienen todas las ventajas mencionadas anteriormente, y podemos decir que son caminos que llevan a cabo los requerimientos de un sistema criptográfico de clave pública. Sin embargo, estos sistemas también tienen sus desventajas. En seguida, vamos a mencionar las desventajas y ventajas de los sistemas que hemos desarrollado y hacer una comparación entre ellos.

5.1 Comparación de los sistemas.

En el capítulo 2 vimos un sistema criptográfico de clave pública que se desarrolló por Rivest, Shamir, y Aldleman. En este sistema se aprovecho el hecho de que encontrar números primos muy grandes es computacionalmente fácil, pero que factorizar el producto de dos números tales es computacionalmente imposible. Se ha probado la seguridad de este sistema por varios caminos 6 , aunque sería mejor examinarla en más detalle, o sea examinar más estrictamente la dificultad de factorizar números grandes. Andres de que se encuentre un algoritmo mucho más eficiente que los que se conoce actualmente para factorizar un producto de números primos muy grandes, podemos considerar que este sistema es computacionalmente seguro. Mientras que en el sistema de distribución de clave pública que vimos en el capítulo 3 (donde se hace uso de la dificultad de calcular logaritmos sobre campo Galois GF(p) con un número primo p, o sea si

$$Y = \alpha^{*} \mod p. \quad \text{para} \quad 1 \leq X \leq p-1),$$

el criptoanalista tiene que calcular

$$X = \log_{\alpha} r \quad \text{sobre} \quad GF(p), \quad \text{para} \quad 1 \leq Y \leq p-1.$$

Para poder romper la seguridad del sistema. Si logaritmos sobre GF(p) son fáciles de calcular, la seguridad del sistema se puede romper, sin embargo, nadie ha probado que el sistema es seguro si logaritmos sobre GF(p) son difíciles de calcular, aunque tampoco hubo alguien que pudo calcular la clave

común de dos otros usuarios a partir de sus claves públicas sin obtener primero las claves secretas X, o sea sin calcular los logaritmos sobre GF(p). Como no hay una tal prueba, siempre se puede dudar de su seguridad, aunque en el momento podemos aprovechar de su seguridad aparente. Desde este punto de vista, podemos decir que el sistema criptográfico de clave pública (capítulo 2) es muy seguro que el sistema de distribución de clave pública (capítulo 3).

Sin embargo, en el sistema criptográfico de clave pública que hemos visto en el capítulo 2 existe una gran incomodidad de manejo, o sea tenemos que hacer operaciones con números muy grandes (e.g., los dígitos), y estas operaciones son difíciles de realizar en las microcomputadoras. Por ejemplo, en un Apple II, usando el lenguaje PASCAL, solamente se pudo lograr hacer operaciones de suma, resta, multiplicación y división con números que tienen máximo 36 dígitos, utilizando un arreglo llamado "Long Integer", pero ya no se puede aplicar muchas funciones sobre estos números enteros grandes, entre ellas, la función "Módulo N" (que se necesita mucho en este caso), no se puede utilizar, se tuvo que resolver este problema de la siguiente manera:

Si quiere calcular Y=X(mod N), donde [X y N] son enteros grandes, hacemos Y=X-N*([X/N]), donde X/N es la parte entera del cociente de la división de X entre N. Aunque este problema se resolvió fácilmente, de todas maneras no es cómodo manejar números muy grandes, y aún no se ha logrado usar números de 100 dígitos. Una manera de manejar números tan grandes es encadenar los números. Por ejemplo, para sumar dos tales números, primero se dividen estos dos números en cadenas, y se empieza a sumar los números por la última cadena, o sea la cadena de los dígitos menos significativos y acarreando a la cadena de los dígitos más significativos, así sucesivamente. Para la resta y la multiplicación se sigue un procedimiento parecido. Pero la división de estos números tan grandes ya no se puede hacer tan fácilmente. Esto quiere decir que el manejo de números grandes sigue siendo un gran problema.

En el sistema de distribución de clave pública para que el trabajo del criptoanalista (cálculo de logaritmos sobre el campo Galois) sea muy grande, tendríamos que elegir un número primo p, tal que p-1 tenga factor primo

muy grande (e.g., de orden de (p-1)/2), o sea que otra vez hay que hacer operaciones de los números muy grandes, a pesar de que buscar un tal número primo p es relativamente fácil, o sea que podemos encontrar primero un número primo grande q, usando el método probabilístico que se mencionó en el capítulo 2, multiplizando ese número q por otro número grande de su orden (aunque sea un número compuesto), a este producto la suma 1, obtenemos el número p, probando este número p con el mismo método a ver si es un número primo, si no lo es, podemos escoger otro q, o multiplicar otro número grande de su orden.

Ahora podemos concluír que los algoritmos que se han utilizado, tanto en el sistema criptográfico de clave pública como en el sistema de distribución de clave pública tienen la imperfección de que hay que usar números muy grandes para que el sistema sea seguro.

Una idea sugerida por los autores de este trabajo para resolver este problema, se resume en lo siguiente.

5.2 Sugerencia de un Nuevo Algoritmo.

En el capítulo 4 habíamos concluído que todo sistema criptográfico de clave pública en donde se publica la clave para cifrar y se mantiene la clave para descifrar en secreto, tiene que agrupar los números de tal manera que sea imposible romper la seguridad de este sistema con un ataque de texto simple elegido [capítulo 2]. Por ese agrupamiento: tenemos que manjar números muy grandes. En cambio, en un sistema de distribución de clave pública, como las claves para cifrar y descifrar no son públicas, o sea que dos usuarios se ponen de acuerdo en las claves para cifrar y descifrar a través de las claves públicas, es difícil realizar un ataque de texto simple elegido. Por lo tanto, no es necesario agrupar tantos números (pero, de todos modos, hay que agrupar por lo menos dos números, para que el sistema sea seguro contra el ataque con solo texto simple, es decir, evitar que los textos en cifras sean los mismos para las mismas letras que aparecen en el texto simple). Por esa gran ventaja que tiene el sistema de distribución de clave pública, sugerimos una idea de un nuevo algoritmo para este sistema.

Cada usuario 1 genera dos matrices A_i y B_i de orden m*n y n*m res

pectivamente, hace la multiplicación:

$$G = A * B$$

donde C es de orden m*m. Publica la matriz del producto C, y mantiene en secreto las matrices A_i y B_i.

Si existe una función, tal que:

$$f(A_i, B_i, C_j) = f(A_j, B_j, C_i) \neq f(C_i, C_j)$$

donde A_i, B_i son las matrices secretas del usuario i, y A_j, B_j, son las matrices secretas del usuario j.

Así, cuando dos usuarios i y j quieren comunicarse en privacía, el usuario i busca la clave pública C_j del usuario j y viceversa, calcu lan independientemente la clave para cifrar k_{ij}, usando las claves secre tas del uno mismo, y la clave pública del otro de la siguiente manera:

$$k_{ij} = f(A_i, B_i, C_j) = f(A_j, B_j, C_i),$$

y usan k_{ij} como la clave para descifrar.

Si para calcular la clave común k_{ij}, se necesita conocer forzosamente las matrices A_i, B_i, o las matrices A_j, B_j, podemos escoger m y n, tal que para calcular A y B a partir de C, necesitan resolver un sistema no lineal en donde hay mas incógnitas que ecuaciones lo cual tiene infinidad de soluciones y es muy difícil de encontrar la solución correcta. Es obvio que si se conoce la matriz C_i de m*m, y se quiere calcular las matrices A_i de m*n, y B_i de n*m, se tendrá que resolver un sistema no lineal de m*m ecuaciones con 2m*n incógnitas. Si escogemos $m*m < 2m*n$, o sea, $m < 2n$, ya logramos nuestra meta. Sin embargo, todavía no hemos en contrado una función que satisface esas condiciones, lo ponemos como sugeren cia para dar una opción a las personas interesadas en este problema.

5.3 Interrelaciones con problemas de autenticación.

Los sistemas criptográficos propuestos para la privacía se puede utilizar

para proveer la autenticación contra la falsificación de la tercera persona. Un tal sistema puede utilizar para crear otros objetos criptográficos.

Un sistema criptográfico que es seguro contra el ataque de texto simple conocido se puede utilizar para producir una función de un solo sentido (que es computacionalmente fácil calcular esta función , y computacionalmente imposible calcular la inversa de ella).

Como se indica en la figura 4, tomando un sistema criptográfico $(S_k: \{P\} \rightarrow \{C\})_{k \in \{k\}}$ que es seguro contra un ataque de texto simple conocido, y considera el mapeo:

$$f : \{K\} \text{ -- } \{C\}$$

definido por:

$$f(x) = S_k(P).$$

Esta función es de un solo sentido, porque resolver X dada f(X) es equivalente al problema criptoanalítico de encontrar la clave a partir de la pareja texto simple-texto en cifras. Sin embargo lo converso de este resultado no es necesariamente cierto.

Las funciones de un solo sentido son básicas para los generadores de claves, debido a que para que el sistema sea seguro, el cálculo de la clave a partir del torrente de claves que es la salida de un generador aleatorio de bits tiene que ser computacionalmente imposible, y para que el sistema sea utilizable, el cálculo del torrente de clave a partir de la clave tiene que ser computacionalmente simple. Así que un buen generador de clave tiene que ser por definición, una función de un solo sentido.

Una otra relación es: Un sistema criptográfico de clave pública se puede utilizar para generar un sistema de autenticación de un solo sentido. El converso no parece cierto, porque construir un sistema criptográfico de clave pública es un problema mucho más difícil que el de la autenticación de un solo sentido. Similarmente un sistema criptográfico se puede utilizar como un sistema de distribución de clave pública, pero no viceversa.

para proveer la autenticación contra la falsificación de la tercera persona. Un tal sistema puede utilizar para crear otros objetos criptográficos.

Un sistema criptográfico que es seguro contra el ataque de texto simple conocido se puede utilizar para producir una función de un solo sentido (que es computacionalmente fácil calcular esta función , y computacionalmente imposible calcular la inversa de ella).

Como se indica en la figura 4, tomando un sistema criptográfico $(S_k: \{P\} \rightarrow \{C\})_{k \in \{k\}}$ que es seguro contra un ataque de texto simple conocido, y considera el mapeo:

$$f : \{K\} \text{ -- } \{C\}$$

definido por:

$$f(x) = S_k(P).$$

Esta función es de un solo sentido, porque resolver X dada f(X) es equivalente al problema criptoanalítico de encontrar la clave a partir de la pareja texto simple-texto en cifras. Sin embargo lo converso de este resultado no es necesariamente cierto.

Las funciones de un solo sentido son básicas para los generadores de claves, debido a que para que el sistema sea seguro, el cálculo de la clave a partir del torrente de claves que es la salida de un generador aleatorio de bits tiene que ser computacionalmente imposible, y para que el sistema sea utilizable, el cálculo del torrente de clave a partir de la clave tiene que ser computacionalmente simple. Así que un buen generador de clave tiene que ser por definición, una función de un solo sentido.

Una otra relación es: Un sistema criptográfico de clave pública se puede utilizar para generar un sistema de autenticación de un solo sentido. El converso no parece cierto, porque construir un sistema criptográfico de clave pública es un problema mucho más difícil que el de la autenticación de un solo sentido. Similarmente un sistema criptográfico se puede utilizar como un sistema de distribución de clave pública, pero no viceversa.

5.4 Aplicaciones de la Criptografía

En los sistemas de tiempo compatido, o sea en las computadoras de multiusuario, la criptografía puede proporcionar una protección en areas del almacenamiento de datos y la autenticación.

La aplicación en el proceso de autenticación es tratar de usar una función de un solo sentido f. Las claves (passwords) están cifradas, y las imágenes de estas claves bajo la función f están almacenadas en la tabla de clave, en vez de que las mismas claves se almacena en esa table. El sistema ahora puede juzgar la validez de un ruego de uso operando con la función f sobre la clave dada por el usuario antes de compararla con la table. Un oponente que roba el directorio de clave no puede usar esa información para personificar a otro usuario, porque la imagen de una clave no es la clave, e invertir f para encontrar la clave es computacionalmente imposible. Para que una función sea de un solo sentido realmente, ésta tiene que tener un dominio grande, asi que esta técnica es segura solamente con claves grandes. Esta es la aplicación más general en la seguridad de la computadora hasta la fecha.

La aplicación de la criptografía a la computadora en la protección de los archivos de datos es obvio. La protección de archivos por la criptografía elimina la desconfianza de los usuarios a los administradores del sistema, también evita las responsabilidades de los administradores del sistema.

Otra aplicación de la criptografía a la seguridad de las computadoras es la protección de los almacenamientos secundarios, tal como cintas y paquetes de disco. Cifrar los contenidos de tales expedientes portátiles de memoria además de servir para la prevenir robos de datos, también sirve para disminuir el costo de la precaución contra la destrucción física, porque las cintas con contenidos cifrados se pueden almacenar en locaciones de remoto, mienbras que las copias no cifradas tienen que estar bien guardadas físicamente, y esto puede ser considerablemente costoso.

La aplicación de la criptografía en las comunicaciones ya es una aplicación clásica. En todos los sistemas de comunicación digital se puede

aplicar la técnica criptográfica descrita en este trabajo. Ahora no solamente en las comunicaciones militares y diplomáticas se necesita usar la tecnología criptográfica, sino también en las comunicaciones comerciales se necesita usar; porque los sistemas de comunicación comercial están cambiando de contactos personales o correspondencias escritas a vías telefónicas, telegramas, telex, facsimil, microondas, etc., en donde una protección física para privacía o autenticación es inaplicable o insuficiente. Las comunicaciones de dato son mas vulnerables que la telefonía, debido a que para escuchar la voz por teléfono, el oponente necesita más habilidad para entender el contenido del mensaje hablado, en cambio, si el material interceptado está en una forma de computadora que se puede leer (e.g., telex), esta limitación ya no existe. Por lo tanto, la técnica criptográfica no solamente se puede aplicar en las comunicaciones humanas, sino también en las comunicaciones entre los humanos y las computadoras. Se necesita más y más técnicas de criptografía para proteger la privacía e identificar la autenticidad de sus usuarios.

REFERENCIAS

[1] Richman, F. Number Theory: An Introduction to Algebra Betmon Calif.: Brooks, 1971

[2] Knuth, D.E. The Art of Computer Programming, Vol. 2: Seminamerical Algorithms. Addison-Wesley, Reading, Mass., 1969.

[3] Diffie, W., y Hellman, M. New Directions in Cryptography. IEEE Trans. Inform. Theory IT-22, 6 (Nov. 1976), 644-654.

[4] Diffie, W., y Hellman, M. Privacy and Authentication: An Introduction to Cryptography. IEEE Proceedings, Vol. 67, No. 3 March 1979. 397-427

[5] Niven, I., y Zuckerman, H.S. An Introduction to the Theory of Numbers. Wiley, New York, 1972.

[6] Rivest, R.L., Shamir, A., y Adleman, L. A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Commication ACM, Vol. 21, No. 2, pag. 120-126, Feb. 1978.

[7] Pohling, S.C. y Hellman, M.E. An Improved Algorithm for Computing Logarithms in G-E (p) and its Cryptographic Significance, IEEE Trans. Inform. Theory, Vol. IT-24, pag. 106-III, Jan. 1978.

[8] Macwilliams, F.J., Sloane, N.J.A., The Theory of Error-Correcting Codes, North-Holland, 1978.

[9] Takahashi, Yasundo Control and Dynamic Systems, Addison-Wesley. 1974.

[10] Director, S.W. y Rohrer, R. A., Introduction to Systems Theory McGraw-Hill, 1975

[11] Eves, H., Elementary Matrix Theory, Dever, 1966.

[12] Sey Moar Lipschutz, Algebra Lineal, Schaum's McGraw-Hill, 1979

[13] Kehn David, Cryptology Goes Public, IEEE Communication, March, 1980 pag. 19-28.

[14] Birkoff, Garretl. Algebra, MacLane, Segunda Edición.

[15] Birkhoff, Garretl. A Survey of Modern Algebra. MacMillan, 1977.

[16] Wonham, W.M., Linear Multivariable Control: a Geometric Approach segunda Edición Springer-Verlag. 1979.

[17] Shannon, C.E., Communication Theory of Secrecy Systems, Reporte Confidencial "A mathematical Theory of Cryptography", sept. 1, 1945.

[18] Aho, A.V.,Hoperoft, J.E., y Ullman, J.D., The Design and Analysis of Computer Algorithms. Reading, MA.: Addison Wesley, 1974.

[19] Hand book of Mathematical Functions with formulas, Graphs and Mathematical Tables, Milton Abramowitz and Irene A. Stegun Dever Publications. Inc., 1972.