17
2ej

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

EL PROBLEMA DE FALSIFICACION DE LAS FIRMAS DIGITALES INTRINSECAMENTE PROTEGIDAS

T E S I S

QUE PARA OBTENER EL TITULO DE:

A C T U A R I O

P R E S E N T A

JOSE ABEL GONZALEZ SILVA

DIRECTOR DE TESIS: DR. ISIDORO GITLER GOLDWAIN

1999

MAT. MARGARITA ELVIRA CHÁVEZ CANO
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo de Tesis:

El Problema de Falsificación en las Firmas Digitales Intrínsicamente Protegidas

realizado por JOSE ABEL GONZALEZ SILVA

con numero de cuenta 9354847-6 , pasante de la carrera de *Actuaría*

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis		
Propietario	Dr. ISIDORO GITLER GOLDWAIN	CINVESTAV-IPN
Propietario	Dr. SERGIO RAJSBAUM GORODESKY	UNAM-IMATE
Propietario	Dr. GILBERTO CALVILLO VIVES	BANCO DE MEXICO
Suplente	Dr. GERARDO VEGA HERNANDEZ	UNAM-DGSCA
Suplente	Dr. GUILLERMO MORALES LUNA	CINVESTAV-IPN

Consejo Departamental de Matemáticas

M. en A.P. MA. DEL PILAR ALONSO REYES

El problema de falsificación en las firmas digitales intrínsecamente protegidas

José Abel González Silva

Director de Tesis: Isidoro Gitler Goldwain

A mis padres, hermanos y a Dios.

Agradecimientos:

Estoy enteramente agradecido con todas las personas que han favorecido, para que me haya sido posible llegar a este punto de instrucción ante la vida.

Doy especialmente las gracias al Doctor Isidoro Gitler Goldwain, por el apoyo y entusiasmo que me brindó para poder llevar a cabo este trabajo de tesis. También agradezco a los doctores Gilberto Calvillo V., Sergio Rajsbaum G., Gerardo Vega Hdz. y Gillermo Morales Luna, por su gran interés y por la revisión de la tesis.

Hago un particular agradecimiento a mis padres y hermanos quienes siempre me han apoyado en todas las facetas de mi vida. Gracias otra vez a ellos.

Agradezco al CINVESTAV, a la UNAM, y al ISA: por el apoyo y amparo que me proporcionaron para poder realizar y concluir mi carrera universitaria.

Quiero agradecer enfáticamente a mis amigos Isaías López M. y Armando F. Mendoza P., por su disposición y atención a mis preguntas.

También agradezco a mis amigos Raquiel R. López Mtz., Josue Ramírez O., Guadalupe Rodríguez y Rigoberto Gabriel, por su invaluable amistad.

Finalmente quiero agradecer a la Sra. Silvia Mercedes Hernandez M. por la revisión ortográfica y metodológica de la tesis.

Contenido

Lista de figuras

Lista de tablas

Introducción

Actualmente la seguridad es un factor definitivo para la realización de múltiples tareas que involucran proceso de información digital, sin demérito de su importancia comprobada, es posible anticipar un despliegue crecientemente significativo a futuro, que proporcionan un confiable nivel de autenticidad y seguridad a los usuarios de la información digital.

Las firmas digitales fueron propuestas por Diffie y Hellman en 1976 y formalmente definidas por Goldwasser, Micali y Rivest en 1988; básicamente son instrumentos que permiten a una entidad "A" diseñar firmas de tal naturaleza que cualquiera que conozca la llave pública de "A" esté en posibilidad comprobar su validez; el procedimiento, sin embargo, sólo es computacionalmente seguro para el firmante, en vista de que las firmas son suceptibles de falsificación para cualquier sujeto que disponga de un alto poder computacional; por ejemplo, una persona capaz de factorizar grandes números enteros, quien puede falsificar fácilmente un plan de firmas RSA.

Por otra parte y en vista de que la seguridad de estos planes se basa exclusivamente en supuestos criptográficos, en caso de una falsificación será difícil para "A" demostrar que no es el autor de la firma.

Las firmas digitales son una herramienta criptográfica que a través de estructuras matemáticas proporcionan el servicio de autentificación en una comunicación digital.

Las **Firmas Intrínsecamente Protegidas (FIP)**[1] se distinguen de las firmas digitales convencionales porque permiten al

[1] Denominaremos como firmas intrínsecamente protegidas (FIP) a lo que otros autores llaman "Signatures Fail-Stop".

firmante demostrar con certeza la eventual falsificación y, una vez publicada la demostración correspondiente, el sistema puede detenerse y garantizar un grado más alto de seguridad que el provisto por las firmas digitales ordinarias.

Un plan de firmas intrínsecamente protegido resuelve el problema al ofrecer un método de prueba basado, precisamente, en el hecho de que una falsificación ha ocurrido, incluso si se enfrenta a un falsificador que disponga de un poder computacional ilimitado y tenga, por tanto, acceso a firmas correspondientes a mensajes realizados previamente.

Un firmante que disponga de un poder computacional polinomial está en plena posibilidad de mostrar que el supuesto subyacente -la factorización de un número, por ejemplo- ha sido roto, en cuanto vea la falsificación contra una posibilidad insignificante de error; por tanto el firmante queda protegido contra un falsificador arbitrariamente potente. Una vez detectada la primera falsificación, todos los participantes, o el operador, saben que el plan de firmas fue roto y pueden detener el sistema.

En el campo de las actividades financieras existe información que se considera estratégica y requiere de autentificación. Tal es el caso, por ejemplo, de las transacciones que se realizan entre un banco y un cliente: ambos requieren de información auténtica: si el cliente aplica un plan de firmas intrínsecamente protegido para autentificar su compromiso con el banco, estará seguro de que todas sus firmas serán aceptadas; estará también capacitado para demostrar falsificaciones y el banco, a su vez, tendrá garantizado que el cliente no podrá construir una demostración de falsificación sobre una firma prefabricada para el efecto.

En síntesis, la aplicación de un plan de firmas intrínsecamente protegido cubre tanto al firmante como al receptor de una firma digital.

Los avances tecnológicos y sus resultados ejercen un fuerte impacto y han revolucionado prácticamente todos los campos de la

vida humana, en particular en lo que atañe a actividades como la política, el comercio y las finanzas, entre muchas otras, que han llegado a depender en buena medida de la rapidez y la seguridad de la transmisión de la información, ya que un retraso o cualquier transtorno en este aspecto podría ocasionar grandes pérdidas y, por el contrario, cuando la información transmitida es oportuna y auténtica optimiza los resultados estratégicos que se persiguen.

Así como ha abierto múltiples oportunidades de desarrollo, el progreso tecnológico también ha dado pie a la inseguridad de la información, que implica la necesidad de autentificar todo contenido que se transmite, y genera dos problemas básicos: el de la **privacidad** y el de la **autenticidad**, que deberán encontrar un punto de equilibrio para garantizar óptimos resultados. Es en este aspecto donde la **criptografía**[2] encuentra su campo de aplicación.

La cripotgrafía proporciona un conjunto de arreglos y técnicas que permiten transmitir con un alto grado de seguridad información que se considera estratégica, debe ser auténtica y mantenerse, además, en secreto. Cuestión de importancia capital para organizaciones gubernamentales o instituciones financieras, por ejemplo.

Los medios de comunicación digital de fácil intercepción, como teléfono, radio, televisión correo electrónico, entre otros, aumentan considerablemente su eficacia cuando disponen de servicios de privacidad y autenticidad, procedimiento desacostumbrado en México, pero usual en otros países donde existe incluso compañías especializadas en la prestación de este servicio.

La comunicación a través del correo usual constituye un modelo que trasparenta la complejidad del fenómeno de la transmisión de mensajes, porque no sólo enmarca la necesidad de comunicarse, sino también la de autentificar tanto los mensajes como al emisor y receptor participantes en el proceso, además de garantizar la transmisión. Del dominio público es el procedimiento básico de

[2]Vid definición 2.1.1

autentificación implementado en este caso, a saber, las autógrafas sobre los documentos.

En el marco de la comunicación digital el procedimiento de autentificación será análogo al arriba descrito, es decir por medio de **firmas digitales**[3], conceptualmente distintas de las firmas manuscritas ya que consisten en un conjunto de dígitos que se adjunta a un mensaje digital por medio de un procedimiento electrónico. A diferencia de la firma manuscrita, invariable, la firma digital dependerá del mensaje, de tal forma que si a éste se le cambia un dígito creando un nuevo mensaje, la firma correspondiente variará también de la del mensaje original.

La importancia del tema y su novedad hacen que esta tesis se limite metodológicamente a tratarlo en exclusiva, lo que dará pie a que futuras investigaciones lo contemplen en pausas más amplias con otros métodos de seguridad.

0.1 Resumen general

En el capítulo 2 se trata la importancia que tiene la seguridad de la información en la comunicación digital. Dentro de la transmisión de la información digital hay dos problemas fundamentales que son, la privacidad y la autenticidad, sobre los cuales se aplican técnicas de criptografía para su solución. También se presentan las firmas digitales que son una herramienta criptográfica que a través de estructuras matemáticas proporcionan el servicio de autentificación en una comunicación digital.

En el capítulo 3 se presentan las firmas digitales intrínsecamente protegidas. Iniciando con un resumen sobre el conjunto de trabajos dedicados a este plan. Posteriormente se dan las características de las FIP, en seguida se enuncia la definición general de un plan de firmas convencional, el cual es extendido para estructurar la definición general de un plan de firmas intrínsecamente protegido. Conti-

[3]Vid sección-definición 3.2

nuando con este tema, se expone la definición formal de un plan de firmas intrínsecamente protegido, junto con las definiciones de seguridad de este plan de firmas. En seguida se detalla la relación de las FIP con las firmas digitales ordinarias. Y finalmente se da la idea sobre la construcción de un plan de firmas intrínsecamente protegido, que es llevado a cabo entre el firmante y los receptores.

En el capítulo 4 se presentan las bases matemáticas y la estructura computacional en que se basan las firmas intrínsecamente protegidas. Primeramente se expone a "grosso modo" la teoría de números, grupos y conjuntos. Enseguida se dan unas funciones en las que las preimágenes son conjuntos grandes. Finalmente se presenta la complejidad sobre las suposiciones criptográficas.

El capítulo 5 comienza con la definición de los homomorfismos fibrados, para posteriormente presentar la construcción general de los planes de firmas intrínsecamente protegidos basados en estos homomorfismos, donde a su vez tienen como base alguna suposición criptográfica. Después se presentan dos planes de FIP, basados en el PLD y en el PFE. El plan basado en este último problema, es construido por parejas de permutaciones que tienen la propiedad de ser libres de pinza. Al final de este capítulo se expone un cuadro que compara la complejidad de estos dos planes de firmas intrínsecamente protegidos.

En el capítulo 6 se expone la generalización del plan de firmas basado sobre el problema del logaritmo discreto. En este plan se logra reducir el tamaño de la llave secreta por un factor de 2 sobre el número de mensajes firmados.

En el capítulo 7 se presenta un protocolo de firmado en donde el cliente de un banco firma las transacciones, aplicando un plan de firmas intrínsecamente protegido, y el banco aplica un plan de firmas digitales ordinario. El propósito de este protocolo es hacer al cliente incondicionalmente seguro, ya que él firma sus compromisos aplicando un plan de FIP, lo cual le permite hacer demostraciones de falsificación y además es receptor de las firmas del banco

Capítulo 1

Preliminares

1.0.1 Clasificación[1]

El conjunto de técnicas criptográficas aplicadas para la protección y seguridad de la información se clasifica según los siguientes casos:

- **Sistemas secretos**: que, como su nombre lo indica, son aquellos que se emplean para reservar el contenido de los mensajes.
- **Sistemas de autentificación**: aquellos utilizados para asegurar la autenticidad de los mensajes y los emisores y receptores.

Estos **sistemas criptográficos** se les denominan **Criptosistemas**; un plan de encriptamiento se denominará sistema secreto o bien sistema criptográfico y puede cumplir simultaneamente las funciones de un sistema secreto y las de uno de autentificación.

Los sistemas criptográficos se clasifican en dos tipos, según la modalidad en que se distribuya la llave:

- **Sistemas simétricos**: son aquellos en que el emisor y el receptor de un mensaje disponen de la misma llave (secreta).

[1]Los siguientes datos fueron tomados en su mayor parte de [DiHe], [Pf90], [PfWa91] y [GMR].

- **Sistemas asimétricos**: aquellos en que uno de los participantes tendrá la **llave secreta** y el otro la **llave pública**.

1.0.2 Origen de las firmas digitales

En 1976 cuando Diffie y Hellman introdujeron el concepto de "criptosistema de llave pública" en [DiHe], también presentaron la idea de firma digital. En su artículo[2] correspondiente proponen que el firmante elija una función "g", públicamente disponible, que sea a la vez fácil de aplicar y computacionalmente difícil de invertir (esta función es la que respalda la seguridad contra falsificación). Así pues, si el firmante quiere firmar un mensaje binario $m = m_1, \cdots m_n$, donde $m_i \in \{0, 1\}$ con una longitud de n dígitos, entonces elegirá en forma aleatoria $2n$ elementos en el dominio de g como **llave secreta**, esto es, $x_1, y_1, x_2, y_2, \ldots, x_n, y_n \in Dom(g)$ serán la **llave de firma**, es decir, la información que sólamente es conocida por el firmante para autentificar el mensaje. La **llave pública** o **llave de prueba** es la información que se da a conocer públicamente para poder probar la autenticidad del mensaje, será las imágenes de la llave secreta bajo g, esto es $f(x_1), f(y_1) \cdots, f(x_n), f(y_n) = X_1, Y_1, \cdots, X_n, Y_n$. Así el firmante construirá la firma sobre m como sigue: si $m_i = 0$, enviará al receptor el valor x_i y si el $m_i = 1$, enviará el valor y_i, para $i = 1, 2, \ldots, n$. El receptor aplicará g a los valores recibidos y comparará los resultados con los respectivos valores publicados para saber si m_i es un dígito 0 ó 1, para $i = 1 \cdots n$, teniendo como resultado un mensaje m, autentificado y privado.

Desde que surgió la idea de las firmas digitales el proceso ha conocido el desarrollo de distintos planes, entre ellos:

Plan de firmas Trap-door [DiHe], plan de firmas Rivest-Shamir-Adleman [RSA], plan de firmas Merkle-Hellman [MH78], plan de firmas Rabin [Ra79], plan de firmas Willimans [Wi80], plan de firmas Lieberherr [Li81], plan de firmas Shamir [Sh78], plan de firmas Goldwasser-Micali-Yao [GMY83], plan de firmas Ong-Schnorr-Sha-

[2] Vid artículo [DiHe].

mir [OSS84a], plan de firmas El-Gamal [EG84], plan de firmas Okamoto-Shiraishi [OS85].

A todos ellos se les denomina usualmente **planes de firmas convencionales** o **planes de firmas ordinarios** y funcionan de la siguiente manera: el emisor-firmante genera la llave secreta "sk" y, bajo ésta, la llave pública "pk" , mantendrá en secreto "sk" y publicará la respectiva "pk", firmará los mensajes con "sk"y el receptor verificará su autenticidad con "pk" correspondiente.

A la función empleada inicialmente por W. Diffie y M. Hellman se le dio el nombre de **un-sentido**[3]. El obstáculo para falsificar estas firmas digitales se basa en la dificultad para invertir tales funciones. La mayoría de los planes de firmas se basan en dos grandes problemas: el **Problema de Factorizar Enteros (PFE)** y en el **Problema del Logaritmo Discreto (PLD)**[4], por tanto la seguridad de estos sistemas se basa en la suposición de que un falsificador no disponga de un algoritmo suficientemente bueno ni del tiempo necesario para realizar la factorización o bien calcular el logaritmo discreto. Cuando se alude a cualquiera de estos dos problemas se hablará exclusivamente en términos de **Suposición Criptográfica**.

1.0.3 Parámetros de seguridad

Es importante considerar que las suposiciones criptográficas se supediten al crecimiento del poder de cómputo, así como a los algoritmos creados para hacer más eficaz su cálculo; lo que implica la necesidad de incrementar los **parámetros de seguridad** en lapsos de tiempo adecuados y a niveles tecnológicos plausibles, que garanticen el nivel de seguridad a los usuarios de un sistema criptográfico.

[3]Vid definición 2.1.6

[4]Vid sección 4.4.

1.0.4 Tipos de ataques a un plan de firmas

Básicamente pueden ser de dos clases:

- *Ataque a través de las llaves*: en el que el intruso conoce únicamente la llave pública del firmante.

- *Ataque a través del mensaje*: donde el intruso es capaz de examinar algunas firmas correspondientes a cualquier mensaje conocido o elegido antes de intentar romper el sistema.

Los ataques a través de los mesajes se caracterizan a partir del método de selección de los mensajes, según se dé antes o después de ver las firmas correspondientes.

- *Ataque con mensajes conocidos.* El intruso tiene acceso a las firmas para un conjunto de mensajes $m_1, \ldots m_t$. Los mensajes son conocidos por el intruso pero no son elegidos por él.

- *Ataque-genérico con mensajes elegidos.* el intruso tiene permitido obtener las firmas válidas del firmante de una lista seleccionada de mensajes $m_1, \ldots, m_t$, antes del intento de romper el plan. Los mensajes son elegidos por el intruso, pero fijados e independientes de la llave pública del firmante -por ejemplo, los mensajes $m_i's$ pueden elegirse aleatoriamente-. Se trata de un ataque no-adaptado: la lista de mensajes se construye antes de ver alguna firma; es génerico porque no depende de la llave pública del firmante y puede usarse contra cualquier firmante.

- *Ataque-dirigido con mensajes escogidos.* es similar al ataque genérico con mensajes escogidos, con la diferencia de que la lista de mensajes puede ser creada después de haber visto la llave pública del firmante, pero antes de ver alguna firma; (por tanto es aún no-adaptado) y está dirigido contra un usuario particular.

- *Ataque-adaptivo en la selección del mensaje.* Este es más general: el intruso considera al firmante como un "oráculo",

ya que *no únicamente* puede requerir las firmas de los mensajes que dependan de la llave pública, sino también puede obtener firmas de mensajes que dependan de firmas previamente obtenidas.

Estos cuatro tipos de ataque están descritos según su grado creciente de severidad, donde el ataque-adaptivo es el más severo.

1.1 Seguridad de las firmas convencionales

Todo plan de firmas existente en la actualidad se basa en alguna suposición criptográfica, dado que la factorización de un número es única, dicha suposición se apoya en el hecho de que un falsificador no dispone del tiempo necesario ni de un algoritmo suficientemente bueno para realizar la factorización; principio válido también para los sistemas basados en el problema del logaritmo discreto, cuya seguridad ha recibido mucha atención recientemente.

Entre los planes de firmas a que se alude y sus mecanismos de seguridad se cuentan los siguientes:

Plan de firmas Rivest-Shamir-Adleman [RSA], **selectivamente falsificable** aplicando un **ataque-dirigido con mensajes escogidos**[5].

Plan de firmas El-Gamal [EG84], basado en la dificultad de calcular el logaritmo discreto, **existencialmente falsificable** por medio de un **ataque-genérico con mensajes escogidos**[6].

En [GMR], se definió la seguridad óptima para los planes de firmas digitales convencionales y se construyó un plan de firmas consecuente: cada firmante da a conocer su llave pública "pk" y una cadena "s" vale como una firma bajo el mensaje "m" si y sólo si

[5] Vid secciones 1.0.4 y 1.1.1
[6] Vid secciones 1.0.4 y 1.1.1

esta firma pasa la "*prueba*(pk, m, s)": un algoritmo con un tiempo de ejecución polinomial, que verifica si "s" es una firma sobre "m", donde "s" se construye con la llave secreta "sk", que corresponde a "pk" que tiene como resultado "aprobar" o "no aprobar" la firma "s" sobre el mensaje "m" para el supuesto firmante.

Esto implica que el problema de decisión cuestionado: ¿es "s" una firma falsificada?, es NP[7], ya que en un tiempo polinomial se tiene aprobación o rechazo.

Análogamente a todos los planes de firmas, la seguridad se basa en suposiciones de la teoría de la complejidad, GMR, es basa en la existencia de parejas de permutaciones que tienen la propiedad de ser **trampa-libre de pinza**[8], tan complejo como al problema de la factorización de enteros[9]. En la práctica deben elegirse funciones especiales bajo la expectativa de que sean un-sentido, y presuponiendo la dificultad para calcular el logaritmo discreto o la factorización de números enteros grandes a partir de la tecnología los conocimientos actuales.

En este trabajo también se demostró que este plan de firmas no es *existencialmente falsificable* usando un **ataque-adaptivo en la selección del mensaje.**

En este plan de firmas se ha probado que si la factorización es difícil, los algoritmos con tiempo de ejecución polinomial tienen una posibilidad insignificante de falsificar alguna firma GMR. Lo que aún no ha sido probado para las firmas RSA.

1.1.1 ¿Que significa "romper" un plan de firmas?

Se puede decir que un intruso ha "roto" el plan de firmas de un firmante si su ataque le permite efectuar cualquiera de los siguientes eventos, con una probabilidad no insignificante:

[7]Vid definición 2.1.18.
[8]Vid definición 2.1.9.
[9]Vid definición 2.1.9

- *Un rompimiento total.* Al calcular la llave secreta del firmante.

- *Falsificación Universal.* Encontrar un algoritmo eficiente que sea funcionalmente equivalente al algoritmo de firma del firmante (posiblemente basado sobre una información secreta muy similar).

- *Falsificación Selectiva.* Falsificar una firma para un mensaje en particular, elegido especialmente por el intruso.

- *Falsificación Existencial.* Falsificar una firma para al menos un mensaje. El intruso no tiene control sobre los mensajes de los cuales obtiene las firmas, puede ser aleatorio o no secuencial. Consecuentemente esta falsificación es la que causa menos ruido para el firmante.

1.1.2 Consecuencias en caso de falsificación

Si un plan de firmas es roto, aun cuando se tenga la esperanza de que ésto no pueda ser, el supuesto firmante de un mensaje de una firma falsificada está indefenso ya que la falsificación se puede aparecer como una firma auténtica. Si esto llegase a ocurrir y se presenta el caso a un juez, éste utilizará únicamente la llave de prueba pk para decidir si la firma es válida o no; por tanto el supuesto firmante será responsable de las implicaciones de dicha firma. El receptor del mensaje firmado está absolutamente asegurado, con la condición de que él haya probado que la firma pasa la prueba, utilizando la llave pública, ya que él sabe, que este proceso será hecho por el juez no importando si la firma es realmente auténtica.

1.2 Firmas Intrínsecamente Protegidas (FIP)

El plan de firmas en el que se centra este trabajo, llamado *plan de firmas intrínsecamente protegidas*[10], resuelve el problema de la responsabilidad jurídica del firmante, dotándolo de un método para que pueda hacer demostraciones de falsificación.

De manera similar a los planes de firmas digitales convencionales, los planes de firmas intrínsecamente protegidos se basan en suposiciones criptográficas. Su funcionamiento es semejante a las firmas digitales convencionales, ya que en este plan el firmante tiene su llave secreta sk, que utiliza para hacer firmas que pueden ser verificadas por cualquiera que conozca la llave pública respectiva pk. Una firma que pasa la prueba se le llama firma **aceptable**.

La idea de la construcción de las firmas digitales intrínsecamente protegidas, es que cada mensaje tiene muchas firmas aceptables que son diferentes, pero el firmante sólamente puede construir una firma, a la cual se le llamará **firma correcta**[11].

La seguridad del emisor se apoya en la pequeña probabilidad de que dos firmas puedan ser iguales, es decir, en la probabilidad de que la firma falsificada sea igual a la firma auténtica de un mensaje específico.

En caso de presentarse una falsificación, el firmante expondrá la firma falsificada y su firma legítima para calcular la demostración de falsificación. La verificación de la demostración de falsificación, se hace utilizando únicamente la llave pública pk, es decir, no se tiene que mostrar la llave secreta del fimante.

[10] Vid definición 3.3.1

[11] Vid sección 1.3.1.

1.2.1 Características de las FIP

Las firmas intrínsecamente protegidas mejoran la seguridad ya que capacitan al firmante incondicionalmente[12] para probar falsificaciones (con una probabilidad alta). Esto se cumple si los adversarios sólo pueden ejecutar algoritmos con tiempo de ejecución polinomial. Las ventajas que proporcionan las FIP son:

- En el momento en que exista una demostración de falsificación, se puede proceder de dos maneras: Detener el sistema o incrementar los parámetros de seguridad.

- El porcentaje de riesgo de una falsificación sobre las firmas puede ser dividido arbitrariamente entre firmantes y receptores. En caso de que se haga una demostración de falsificación, la firma respectiva se invalida y el firmante es incondicionalmente seguro.

Es importante enfatizar que la suposición criptográfica garantiza -supeditada a la tecnología y conocimiento actuales- que las falsificaciones no puedan ocurrir y también que el firmante no pueda reclamar con dolo la falsificación de una firma.

La seguridad para el firmante en un plan de FIP se basa en la teoría de la información que garantiza que él puede hacer demostraciones de falsificación; en tanto que la seguridad del firmante -lo mismo que la del receptor- en un plan de firmas convencionales se basa en una suposición criptográfica: si se le permitiera al firmante detener el plan notificando que una falsificación ha ocurrido o bien que lo secreto de sus llaves se ha perdido, los receptores quedarán en inseguridad total, dado que el firmante podría desconocer incluso sus propias firmas.

[12] El concepto de incondicionalidad quiere decir que cuando el firmante ha hecho una demostración de falsificación, ésta siempre será aceptada en la verificación

1.2.2 Errores de probabilidad Vs. suposiciones criptográficas

La probabilidad de que una demostración de falsificación falle es muy pequeña, está determinada por un parámetro de seguridad, ($2^{-\sigma}$, por ejemplo)[13]. Esto no es grave: un pequeño error de probabilidad es inevitable en los sistemas de autentificación digital, incluyendo los simétricos, en donde el firmante usa un número finito de bits de información secreta. Si el falsificador adivina todos los bits correctamente, nada puede hacerse al respecto. Por tanto se llamarán sistemas de autentificación "incondicionalmente seguros", si éstos han sido probados para un falsificador, es decir:

- Su mejor falsificación sólo la puede hacer adivinando.
- No tiene la capacidad de probar si en efecto ha adivinado correctamente la firma. (Excepto si le pregunta al verdadero firmante).

En este sentido, es incondicionalmente seguro de que una falsificación sobre las firmas intrínsecamente protegidas, pueda ser probada.

La diferencia sustancial entre un plan con un pequeño error de probabilidad y un plan que se basa en una suposición criptográfica, como las firmas digitales convencionales, es que con éstas últimas no se sabe si la probabilidad de que haya una falsificación sea pequeña.

Las firmas intrínsecamente protegidas no están protegidas en absoluto si la llave es robada físicamente: claro que ésto también es inevitable en todos los sistemas de autentificación: una vez que el falsificador tiene la información secreta completa, es imposible distinguir al falsificador de un firmante a través de medios digitales.

[13]Vid sección 3.3

1.3 Descripción sobre la construcción

1.3.1 Idea de contrucción de las FIP

La idea básica para la contrucción de las firmas intrínsecamente protegidas, es que a cada llave pública pk le corresponden muchas llaves secretas y cada mensaje m, que sea firmado con diferentes llaves secretas tendrá como resultado muchas firmas diferentes que pasan la prueba (aún después de un ataque-adaptivo[14] en la selección del mensaje, es decir, cuando el adversario ha recibido varias firmas sobre mensajes elegidos por él). En donde el firmante únicamente tiene una llave secreta sk y por tanto sólo puede calcular una firma s.

Para hacer una demostración de falsificación basta con presentar dos firmas diferentes sobre el mismo mensaje m, que corresponden a la misma llave pública.

Los siguientes puntos nos dan una idea de porqué un plan de firmas intrínsecamente protegido, con estas propiedades, es un plan seguro:

1. Lo que puede hacer un falsificador con un poder computacional ilimitado, es calcular todas las firmas posibles sobre un mismo mensaje, después adivinar cual de todas éstas es la firma correcta. Este plan de firmas garantiza que con una probabilidad grande el adversario elegirá mal la firma correcta. La comparación de un par de firmas diferentes (la falsificada y la correcta), implica una demostración de falsificación.

2. Por otro lado, es computacionalmente difícil para el firmante calcular una demostración de falsificación, si ésta no ha ocurrido previamente. Esta afirmación se basa en la suposición criptográfica. Ya que si la suposición criptográfica es verdadera, este plan trabaja en forma semejante a un plan de firmas convencionales.

[14] Vea la sección 1.0.4

Cuando se dice que hay "muchas" posibles firmas que pasan la prueba sobre un mensaje, quiere decir hay del orden de 2^{τ}, en donde 2^{τ} es la cardinalidad de el círculo de la figura 0.0.1, (τ es el parámetro de seguridad para el firmante)[15].

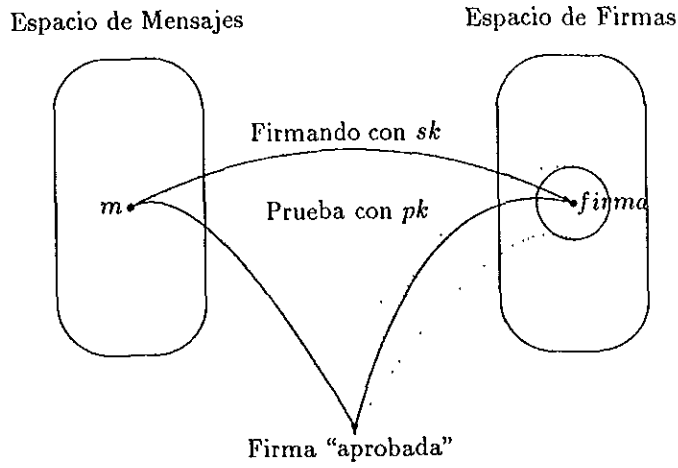


Figura 0.0.1 Esquema de la idea básica de la construcción de las FIP

La construcción garantiza que: si la suposición criptográfica es verdadera, nadie excepto el firmante puede encontrar alguna firma aceptable y exactamente una. Es importante notar que el círculo es pequeño en comparación con el espacio total de firmas.

Si un falsificador encuentra la llave de la suposición criptográfica, podrá ser capaz de calcular los valores del círculo (fig. 0.0.1), ya que con esta información puede invertir la función de prueba, pero no podrá encontrar fácilmente cuál de estos valores es la firma auténtica. Remitimos al lector al ejemplo 5.2.1.1 donde se muestra todo lo expuesto en esta sección.

[15]Vid sección 3.3.

1.3.2 Ejemplo de homomorfismos fibrados

En seguida presentaremos un panorama sobre la seguridad de las FIP. También mostraremos los parámetros de seguridad en la construcción de éstas.

Las firmas digitales intrínsecamente protegidas están construidas en base a los **homomorfismos fibrados** y éstos a su vez tienen como problema subyacente alguna suposición criptográfica[16].

Un homormfismo fibrado h, es un homomorfismo entre dos grupos abelianos $(G, +, 0)$ y $(H, \cdot, 1)$ que tiene las siguientes características:[17].

1. Cada $z \in Im(h)$ tiene al menos 2^τ preimágenes.
2. h, es **resistente a colisiones**. (Es decir, es difícil encontrar dos valores en el dominio que tengan la misma imagen).

Los parámetros de seguridad están caracterizados para cada uno de los participantes de la comunicación. La seguridad del firmante se garantiza por (1); Aquí τ es el parámetro de seguridad para el firmante. La seguridad del receptor de las firmas se garantiza por (2). En este requrimiento está *implícito* otro parámetro de seguridad, el cual garantiza que la suposición criptográfica sea verdadera.

En seguida construiremos un homomorfismo fibrado respaldado por la dificultad del PFE.

Construcción:[18]

La construcción de un homomorfismo fibrado se hará considerando un par de permutaciones que sean **libres de pinza**, esto es, una pareja de permunaciones sobre el mismo dominio para las cuales sea difícil encontrar dos elementos distintos en el dominio tal que ambas tengan la misma imagen sobre estos valores[19].

[16] Vid las secciones 5.2 y 5.3.
[17] Vid definición 5.1.1.
[18] Vid sección 5.3.
[19] Vid definición 2.1.9.

Como se ha dicho, la seguridad del receptor está sustentada por la dificultad de el PFE. Para esto considérese dos primos p y q, con $p \equiv 3 \pmod 8$ y $q \equiv 7 \pmod 8$ y sea $n = pq$. A los enteros de esta forma se les llama enteros Blum[20]. Si $n = pq$ y tiene las características antes dichas, existe una demostración en la que se prueba que realmente $n = pq$, sin mostrar los factores[21].

Considerese un entero Blum y sea,

$$D_n := \{x | x \in \mathbb{Z}_n^* \text{ y } (\tfrac{x}{n}) = 1 \; y \; x \in \{1, 2, \ldots, \tfrac{n-1}{2}\}\}.$$

Donde $(\frac{x}{n})$ es el símbolo de Jacobi[22]. El símbolo de Jacobi toma los valores de 1, -1, ó 0 y es una herramienta útil para saber si un número es o no un residuo cuadrático.

Sean f_0 y f_1 dos permutaciones sobre D_n,

$$\begin{aligned} f_0(x) : &= |x^2| \pmod n \\ f_1(x) : &= |4 \cdot x^2| \pmod n \end{aligned}$$

Este par de permutaciones es libre de pinza[23], bajo la suposición de PFE. La demostración de la afirmación está en el lema 5.3.1.

Considerando estas permutaciones, el homomorfismo fibrado es construido por la aplicación consecutiva alternada de f_0 y f_1, esto es:

$$\{h : \{0,1\}^\tau \times D_n \longrightarrow D_n\}$$

$$\{(a_0, a_1 \cdots a_{\tau-1}, x) \longrightarrow f_{a_0}(f_{a_1} \cdots (f_{a_{\tau-1}}(x)) \cdots)\}$$

donde $a_i \in \{0,1\}$, $i = 1, 2, \ldots, \tau - 1$.

Por tanto podemos escribir h, como:

---

[20] Vid definición 4.1.4.
[21] Vid [Je]
[22] Vid la sección 4.1.3.
[23] Vid definición 2.1.9.

$$h(a,x) = 4^a \cdot x^{2^r} \pmod{n}$$

donde $a = (a_o, a_1 \cdots a_{r-1})$ es interpretado como el entero $a_{r-1} \cdot 2^{r-1} + \ldots + a_1 \cdot 2 + a_0$.

Ahora sólo falta demostrar que h es realmente un morfismo fibrado. La demostración se verá en el teorema 5.3.1.

El homomorfismo fibrado sirve para generar las llaves públicas del plan de firmas y para probar la autenticidad de una firma sobre un mensaje. Además el homomorfismo forma parte de la llave pública.

1.3.3 Distribución de la llave

La distribución de la llave en un plan de firmas digitales no es trivial, pero si es más fácil que la distribución en sistemas simétricos. En la práctica se asume un centro de distribución de confianza. Sin embargo, en sistemas de gran relevancia legal, es mejor no tener a un tercer ente en quien deben de confiar los usuarios, para no tener posibles fugas de información.

Esta situación incluso puede ampliarse: el firmante dirá su *pk* a un centro, éste a su vez debe firmar la *pk* que deberá ser distribuida. Para esto, el centro puede usar su plan de firmas convencionales. La llave pública del centro es el punto donde comienza toda la autenticidad digital y en donde será distribuida confiablemente sobre toda la red de usuarios. Cualquiera que solicite información al centro por determinada *pk*, exigirá una respuesta firmada. En el caso de una disputa las partes afectadas deberán mostrar la firma del centro.

1.4 Aplicación en sistemas de pagos

Los sistemas de pagos digitales pueden considerarse como una de las aplicaciones más importantes de las firmas digitales intrínsecamente protegidas.

En grandes organizaciones como los bancos, se requiere del uso de las firmas digitales para el intercambio de mensajes con los clientes. Por ejemplo, si los clientes usan un plan de firmas intrínsecamente protegido y los bancos un plan de firmas convencionales, los clientes son incondicionalmente seguros, esto quiere decir, que los clientes no necesitan confiar en una suposición criptográfica, ya que ellos son los firmantes en el plan de firmas intrínsecamente protegido y receptores en el plan de firmas convencionales, y en ambos casos ellos son incondicionalmente seguros.

Una pregunta que es importante es la siguiente: ¿Por que al banco no se le dota para que sea incondicionalmente seguro?

En primer término, porque el banco es un socio fuerte en el sentido computacional, así él puede elegir el plan de firmas convencionales y también sus parámetros de seguridad, para suministrar su propia seguridad. También, porque el banco puede informarse de la confianza que se tenga para la suposición criptográfica.

En segundo lugar, el banco no tiene necesidad de una demostración de falsificación si sus propias firmas son falsificadas: ya que él es el operador del sistema, por tanto él mismo puede pararlo.

En tercer lugar, el banco puede tener ventaja al aplicar un plan de firmas convencional en sistema de pagos digitales, ya que le beneficia el tener una publicidad de gran seguridad para los clientes, por el hecho de que el banco no pueda hacer una demostración de falsificación.

El problema que se tiene con las actuales firmas, es que, la longitud crece rápidamente en función de la longitud del mensaje. Como ya se ha dicho las firmas intrínsecamente protegidas tienen su primordial aplicación en sistemas de pagos digitales. El siguiente protocolo asegura que los clientes que utilizan un plan de FIP solamente firmarán mensajes de 1-bit, dejando la mayor parte del trabajo al banco, quien tiene un plan eficiente de firmas convencionales.

El siguiente protocolo de tres fases hace que el sistema sea completamente práctico:

1. *Instrucción*: El cliente le dirá al banco su i-ésima orden, por ejemplo, "Enviar 1000 pesos al cliente C".

2. *Confirmación*: El banco confirmará la instrucción con un mensaje firmado (usando un plan de firmas convencionales), por ejemplo, la i-ésima orden del cliente A es: "Enviar 1000 pesos a C".

3. *Aprobación*: Si la confirmación es correcta, el cliente firmará solamente 1-bit, lo cual quiere decir "*aprobar*", al hacer su i-ésima firma intrínsecamente protegida.

Con respecto al tercer paso del protocolo, se debe notar que las firmas intrínsecamente protegidas actuales son enumeradas automáticamente[24].

En caso de haber disputa por algún posible error imputable al banco: la i-ésima orden fue inadecuadamente confirmada, por ejemplo, un juez puede decidir lo siguiente: primero el banco debe presentar la "aprobación" del cliente, que permitió que el banco llevara a cabo la i-ésima orden completa. Por otro lado, el cliente debe presentar la confirmación firmada por el banco. El contenido de esta confirmación es considerado como la orden correcta.

Los clientes son incondicionalmente seguros: ya que su seguridad solamente depende de sus firmas intrínsecamente protegidas que almacenan la confirmación correcta del banco. El banco es seguro sobre la suposición criptográfica, la cual es elegida por él mismo. La manera de engañar al banco puede presentar dos modalidades: o bien falsificar la confirmación del banco o bien presentar un demostración de falsificación para una firma intrínsecamente protegida.

De manera más general, cualquier organización puede usar este protocolo para intercambiar mensajes firmados con sus clientes. Esto esta limitado para alguien que sea cliente de varias organizaciones ya que necesita diferentes llaves para cada organización. Por tanto, únicamente puede ser cliente de un número limitado

[24] Vid Captítulo 6.

de organizaciones. En un sistema de pagos digitales esto no es problema.

La manera de usar una función fibrada para firmar un bit se presentará en seguida: (Recordando que en el protocolo de 3-fases no se debe hacer distinción entre el bit "0" y el bit "1"). El cliente del banco, elegirá aleatoriamente un elemento sk del dominio de h, este valor será al mismo tiempo la llave secreta y la firma del usuario. El cliente publicará la imagen $pk = g(sk = firma)$ como la llave pública. Posteriormente cuando el cliente de su *aprobación* (en el 3^{er} paso) al revelar la "$firma$", cualquiera puede probar que ésta es realmente legítima, al calcular $h(sk = firma) = pk$. Como estamos trabajando con homomorfismos fibrados, tenemos exactamente las mismas características que hemos dicho de ellos. Por tanto el cliente y el banco son seguros.

Capítulo 2

Complejidad y Criptografía

2.1 Criptografía

A través del desarrollo computacional se han logrado acrecentar las redes de comunicación, permitiendo que con una gran facilidad y bajo costo la gente pueda tener contacto con otras personas, aun cuando se encuentren en lugares muy distantes entre sí.

Este progreso computacional ha ocasionado el reemplazo del correo ordinario por las redes telecomunicaciones. Por otro lado estos avances tecnológicos también benefician a personas que no actúan de buena fe, por tanto es necesario tomar medidas al respecto.

En muchas aplicaciones los contactos o bien la comunicación entre la gente deben ser seguros contra espias y también contra la inyección ilegal de mensajes dentro de los canales de transmisión[1].

Uno de los principales problemas criptográficos es el de la *privacidad*: si se resuelve este problema se previene la extracción ilegal de información en la comunicación sobre canales inseguros, es decir, medios de transmisión donde la información dirigida sea reacomodada, borrada o leída.

[1]Para el lector que esté interesado en estos temas, le sugerimos consultar [MOV] y [GMR].

Si se hace una retrospección en la historia de la información, se puede concluir que la forma de grabarla no ha cambiado dramáticamente con el paso del tiempo. Anteriormente la información se almacenó y transmitió sobre papel, en la actualidad la mayor parte se registra sobre medios magnéticos y se transmite vía sistemas de telecomunicaciones. Cabe aclarar que lo que realmente ha cambiado de una forma radical es la habilidad de almacenar, copiar y alterar la información.

De hecho se pueden hacer copias idénticas a una pieza de información almacenada electrónicamente y cada una de estas réplicas es indistinta de la original. Lo que necesita una sociedad, en donde la mayor parte de la información se almacena y transmite en forma electrónica, es tener un tipo de seguridad sobre la información, de tal manera, que sea independiente del medio físico en el que se ha grabado o trasportado, en el cual los objetivos de la seguridad de la información se enfoquen únicamente en la información digital.

Otro de los principales problemas dentro de la criptografía que se presenta en la transmisión de información, es la *autentificación*, la cual es indispensable dentro del uso de teleprocesos y en las transacciones de negocios. Generalmente dentro de los negocios se han utilizado las firmas para la validación de los contratos, es decir, para su legitimación. Un contrato firmado sirve como evidencia de un acuerdo entre dos partes, en donde el poseedor del documento firmado puede presentarlo ante un juez si es necesario. El uso de las firmas requiere de algún modo, de una transmisión y almacenamiento de los contratos en que se han aplicado.

Para poder reemplazar las firmas manuscritas por *firmas digitales*, es indispensable que cada usuario pueda ser capaz de firmar un mensaje de tal manera que sea posible verificar su autenticidad por cualquier otra parte.

Una de las herramientas fundamentales de la seguridad de la información son las firmas digitales. Estas firmas constituyen un bloque con muchos servicios, tales como: no rechazo, autenticidad

de datos originales, identificación, testigos, etc.

Por lo anterior es claro que el concepto de la firma necesita ser cambiado: ya que la firma no puede ser simplemente algo único para el firmante e independiente de la información firmada.

Lograr la seguridad de la información en una sociedad electrónica, requiere una inmensa cantidad de arreglos de técnicas y leyes. Sin embargo, no se puede garantizar que todos los objetivos de la seguridad de la información que se han considerado tradicionalmente como necesarios, se puedan satisfacer. Estas técnicas son en gran medida proporcionadas por la *Criptografía.*

En seguida se presentan algunos requerimientos relacionados con la seguridad de la información.

- Privacidad o Confidencialidad
- Integridad de los datos
- Autenticidad de la entidad o identificación
- Autenticidad del mensaje
- Firmas
- Autorización
- Control de acceso
- Certificación
- Testigos
- Recepción
- No rechazo

Definición 2.1.1 *La criptografía es el estudio de técnicas matemáticas aplicadas dentro de la comunicación y almacenamiento digital sobre aspectos de seguridad en la información, tales como: confidencialidad, integridad de datos, autenticidad de la entidad, autenticidad de los datos originales.* ◇

Los siguientes conceptos son considerados como metas criptográficas:

1. *Confidencialidad.* Es un servicio usado para mantener el contenido de la información, sólo para quienes estén autorizados para accesar a la misma.

2. *Integridad de los datos.* Es un servicio dirigido contra la alteración no autorizada de los datos. Para asegurar la integridad del los datos se debe tener la habilidad de detectar la manipulación por las partes no autorizadas. Dentro de la manipulación de datos se tiene, inserción, borrado y sustitución.

3. *Autenticidad.* Es un servicio relacionado con la identificación, del emisor, receptor y de la información misma. La información transmitida sobre un canal debe ser autorizada tomando en cuenta lo siguiente: datos originales, contenido de datos, tiempo de envio, etc.

4. *No rechazo.* Es un servicio el cual previene a la entidad receptora que la parte que emisora pueda negar los compromisos hechos previamente. (Ver página 45)

La criptografía es una parte de la criptología.

Definición 2.1.2 *El criptoanálisis es el estudio de técnicas matemáticas que se utilizan para intentar derrotar las técnicas criptográficas, y en general los servicios sobre la seguridad de la información.* ◇

Definición 2.1.3 *La criptología, es el estudio de la criptografía y criptoanálisis.* ◇

Definición 2.1.4 *Criptosistema es un término general referido a el conjunto de herramientas criptográficas usadas para proporcionar servicios de información segura.* ◇

Las herramientas criptográficas son usadas para proporcionar información segura, por ejemplo: los planes de encriptamiento simétricos y asimétricos, las funciones digestivas (funciones hash), planes de firmas, etc.

2.1.1 Funciones

Hasta este momento no se ha hecho mención de las matemáticas que son utilizadas en criptografía. En seguida se expondrán algunos conceptos matemáticos útiles para los fines de esta tesis:

Un concepto aplicado continuamente en el ámbito criptográfico es el de *función*, también llamada *transformación*. En seguida se definirán tres tipos de *funciones* que utilizaremos posteriormente.

Funciones un-sentido, trampa un-sentido y permutaciones

Definición 2.1.5 *Se dice que un proceso es* **computacionalmente difícil** *si su costo promedio por la cantidad de memoria usada o tiempo necesario para llevarlo a cabo es finito, pero muy grande.*

Definición 2.1.6 *Una función f de un conjunto X a un conjunto Y es llamada función* **un-sentido** *, si $f(x)$ es "fácil" de calcular para toda $x \in X$, pero para la "mayoría" de los elementos $y \in Im(f)$ es "computacionalmente difícil" encontrar algún $x \in X$ tal que $f(x) = y$.* ◇

Ejemplo 2.1.6.1 *Sea $X = \{1, 2, 3, \cdots, 6\}$ y sea $f(x) = r_x$, para cada $x \in X$, donde r_x es el residuo al dividir 3^x entre 7, ($f(x) = 3^x \pmod 7$). Entonces,*

x	1	2	3	4	5	6
$f(x)$	3	2	6	4	5	1

Dado un número entre 1 y 6 es relativamente fácil encontrar la imagen bajo f. Pero, dado un número en la imagen, por ejemplo 6 es "difícil" encontrar un x, tal que $f(x) = 6$. Claro que si se toma el número 3 es claro que $x = 1$. La dificultad de invertir esta función un-sentido se incrementa cuando es más grande el conjunto X.

Dos candidatos a funciones un-sentido

Aunque aún no se ha demostrado que existan las funciones un-sentido, se cree que realmente existen. La dificultad de los siguientes 2 candidatos a funciones un-sentido se basan en la dificultad de calcular el logaritmo discreto en $\mathbb{Z}_p^*$ y la dificultad de calcular las raíces cuadradas módulo n, respectivamente.

1) **Función exponencial módulo p:** Sea p un número primo y sea α un generador de $\mathbb{Z}_p^*$. La función $f : \mathbb{Z}_P^* \to \mathbb{Z}_p^*$, es definida como $f(x) = \alpha^x \pmod p$. [Vea la sección 4.4.2].

2) **Función Rabin:** Sea $n = pq$, donde p y q son primos distintos, y $p \equiv 3 \pmod 4$ y $q \equiv 3 \pmod 4$. La función $f : QR_n \to QR_n$, definida como $f(x) = x^2 \pmod n$, $QR_n = \{a \in \mathbb{Z} \mid a \equiv x^2 \pmod n\}$, donde calcular la principal raiz cuadrada es tan difícil como el problema de la factorización. [Vea la sección 4.4.1].

Definición 2.1.7 *Una función* **trampa un-sentido** *es una función un-sentido $f : X \to Y$ con la propiedad adicional de que dando información extra (la información trampa) resulta posible encontrar para valor $y \in Im(f)$, un $x \in X$ tal que $f(x) = y$.*

$\diamondsuit$

Ejemplo 2.1.7.1 *Dados dos primos $p = 1997$ y $q = 53993$, $n = pq = 107824021$, sea $X = \{1, 2, 3 \cdots n-1\}$. La función f definida sobre X es $f(x) = r_x$ para cada $x \in X$, donde r_x es el residuo al dividir x^3 entre n.*

Calcular $f(x)$ es relativamente fácil, pero encontrar la inversa de cualquier elemento de la imagen es difícil. Es claro que el tener información extra, como el valor de los factores de n facilita invertir la función para cualquier elemento. Encontrar estos factores no es fácil ya que si se tomasen primos de 100 dígitos resultaría muy complicado encontrar los factores del número resultante y por tanto la inversa de la función.

También las permutaciones son funciones que son usadas en varias construcciones de criptografía.

Definición 2.1.8 *Sea Ω un conjunto finito de elementos. Sea $S_\Omega := \{f : \Omega \to \Omega \mid f \text{ es biyectiva}\}$. Un elemento de S_Ω se le llama permutación sobre S_Ω.* ◇

Además de tener permutaciones, se requiere de permutaciones con ciertas propiedades. A continuación definimos un concepto importante.

Definición 2.1.9 *Sean $f_0, f_1 \in S_\Omega$. Se dice que f_0 y f_1 es un par de permutaciones que tienen la propiedad* **libre de pinza** *si y sólo si es computacionalmente difícil encontrar x, $y \in \Omega$ tales que $f_0(x) = f_1(y)$. Una tercia (x, y, z) de elementos de Ω, que satisfagan, $f_0(x) = f_1(y) = z$ será llamado un* **f-pinza**. *Si la dupla (f_0, f_1) es libre de pinza y además es posible bajo ciertas propiedades adicionales calcular computacionalmente f_0^{-1} y f_1^{-1}, se dice que la dupla (f_0, f_1) es* **trampa-libre de pinza**. ◇

Ejemplo 2.1.9.1 *Sea $n = pq$ donde $p \equiv 3 \pmod 8$ y $q \equiv 7 \pmod 8$ y sea $D_n := \{x | x \in \mathbb{Z}_n^* \text{ y } (\frac{x}{n}) = 1 \text{ y } x \in \{1, 2, \ldots, \frac{n-1}{2}\}\}$.*

$$f_{0,n}(x) := x^2 \pmod n,$$
$$f_{1,n}(x) := 4x^2 \pmod n.$$

Encontrar una pinza sobre $f_{0,n}$ y $f_{1,n}$ es tan difícil como solucionar el problema de la factorización.

2.1.2 Probabilidad y complejidad

Definición 2.1.10 *Un algoritmo es un procedimiento computacional bien definido que toma un dato como entrada y para con una salida.* ◇

Usualmente el interés que se tiene sobre los algoritmos, es encontrar el mejor (es decir, el más rápido), para solucionar un problema computacional dado. El tiempo que un algoritmo toma desde su inicio hasta parar depende del *tamaño* de la entrada.

Definición 2.1.11 *El* **tamaño** *de la entrada es la cantidad de bits necesarios para representar la entrada.* ◇

El número de bits de la representación binaria de un entero $n > 0$ es, $1 + \lfloor \log_2(n) \rfloor$, donde $log_2(n)$ es el logaritmo de n en base 2.

Definición 2.1.12 *El* **tiempo de ejecución** *de un algoritmo sobre una entrada particular, es el número de operaciones primitivas o pasos ejecutados.* ◇

A menudo un *paso* o *operación primitiva*, es tomado como una operación bit. Cuando se suman dos números en representación binaria con longitudes n y m, $m \leq n$, se tienen exactamente n operaciones bit. Para algúnos algoritmos será mas conveniente tomar un *paso* como una multiplicación modular.

Definición 2.1.13 *El peor caso de tiempo de ejecución de un algoritmo es la cota superior del tiempo de ejecución para cualquier entrada, que es expresado como una función del tamaño de la entrada.* ◇

Notación asintótica

Para calcular el tiempo de ejecución de un algoritmo, es suficiente estimar un tiempo de ejecución *asintótico*. Esto es, el estudio de como se incrementa el tiempo de ejecución de un algoritmo cuando se incrementa el tamaño de la entrada.

En las siguientes definiciones se está considerando que las funciones están definidas para enteros positivos de algún número en adelante y toman un valor real positivo. Sean f y g dos de estas funciones.

Definición 2.1.14 Notación de orden. **Cota superior asintótica.** *Se dice que* $f(n) = O(g(n))$, *si existe una constante positiva* c, *y un entero positivo* n_o, *tal que* $0 \leq f(n) \leq cg(n)$, *para todo* $n \geq n_o$. ◇

Intuitivamente, $f(n) = O(g(n))$ es equivalente a decir que $f(n)$ no crece mas rápido asintóticamente que $g(n)$.

Clases de complejidad

Definición 2.1.15 *Un algoritmo con* **tiempo de ejecución polinomial**, *es un algoritmo que en el peor de los casos, su tiempo de ejecución* $T(n)$ *es,* $T(n) \leq O(n^k)$, *donde* n *es el tamaño de la entrada y* $k > 0$. *Para cualquier algoritmo, tal que el tiempo de ejecución no puede ser acotado polinomialmente, es llamdo* **algoritmo con tiempo de ejecución exponencial.** ◇

Definición 2.1.16 *En la teoría de la complejidad los* **problemas de decisión**, *son aquellos para los cuales se tiene sólo dos respuestas, SI o NO.* ◇

Definición 2.1.17 *La clase de complejidad* P *es el conjunto de todos los problemas de decisión que se pueden solucionar en un tiempo polinomial.* ◇

Definición 2.1.18 *La clase de complejidad* NP *es el conjunto de todos los problemas de decisión para los cuales la respuesta SI, puede ser verificada en un tiempo polinomial, usando alguna información extra, llamada testigo.* ◇

Ejemplo 2.1.18.1 *(problema en NP). Considérese el siguiente problema de composición: Dado un entero n, la pregunta es: ¿n es un entero compuesto? Esto es, ¿hay enteros* a, $b > 1$, *tales que,* $n = ab$?. *Este problema de decisión es NP.*

Generalmente los cálculos que son factibles se expresan por algoritmos probabilísticos con tiempo polinomial. Para ser más concretos, la máquina de Turing se usa como un modelo de cálculo.

En criptología los **algoritmos probabilísticos** se representan generalmente por máquinas Turing determinísticas, con la entrada adicional de una cinta magnética, la cual es llamada cinta magnética aleatoria. La cinta aleatoria contiene una secuencia de bits aleatorios -potencialmente infinitos-. Donde el contenido ésta cinta tiene una distribución uniforme.

Los algoritmos probabilísticos toman decisiones aleatorias en cada uno de los puntos de la ejecución, lo cual depende del contenido de una cinta magnética aleatoria. (Nótese que esta cinta no es la cinta de la máquina de Turing). Así, en cada punto a ejecutar, primero se verá el contenido en una celda de la cinta, donde cada celda sólo puede ser leida una única vez.

Para un espacio de probabilidad S con una distribución probabilidad dada, se denotará a $P(E)$ como la probabilidad del evento E en S y se denotará a $[S]$ como el conjunto de elementos con probabilidad positiva. Para representar una elección de un valor de S y asignarlo a una variable x, se escribirá $x \leftarrow S$.

Si a es un algoritmo probabilístico, se denotará con $a(i)$ a el espacio de probabilidad sobre la salida de a, con la entrada i. (Es decir, si se tiene una entrada i sobre a, donde la entrada i no esta incluida en la cinta aleatoria). Para cubrir los cálculos no terminados, el espacio de salida en un algoritmo es aumentado en un elemento $\uparrow$, es decir el algoritmo probabilístico ha terminado sin arrojar un resultdo.

Si $a_1, \ldots, a_n$ son n algoritmos probabilísticos y p es una afirmación con n entradas, $(n \in \mathbb{N})$, entonces $P(p(x_1, \ldots, x_n) :: x_1 \leftarrow a_1(i_1); \cdots; x_n \leftarrow a_n(i_n))$ denota la probabilidad de que la afirmación $p(x_1, \ldots, x_n)$ sea verdadera después de que los resultados de los algoritmos a_j con entradas i_j han sido asignados a x_j, para $j := 1, 2 \ldots, n$ (en este orden).

La longitud de una cadena de caracteres es denotado por, $|\cdot|$. La longitud binaria de un número es la cantidad de bits necesarios para representarlo, y es denotado por $|\cdot|_2$. El conjunto de todas las cadenas binarias no vacías es denotado por $\{0,1\}^+$.

Una consecuencia de la seguridad computacional, es que los sistemas de firmas usualmente dependen de un **parámetro de seguridad**, el cual determina la longitud de las muestras del problema que se supone que es difícil de resolver, como es el caso del logaritmo discreto. Por tanto la selección de un parámetro de seguridad grande dará como resultado llaves y firmas más grandes, además los procesos de firmado y de prueba de las firmas también

necesitarán más tiempo, al pagar este costo se desearía que las construcciones de las falsificaciones para las firmas también sean difíciles, de manera que la discrepancia entre la complejidad de los algoritmos legales: *gen*, *firma y prueba* y los algoritmos de los falsificadores sea más grande. Tales parámetros serán representados por candenas unarias, es decir por unos $1^k = \underbrace{11\cdots1}_{k-veces}$, para la entrada de cualquier algoritmo.

La razón de que las representaciones sean unarias es la siguiente: por un lado, se tiene que el parámetro de seguridad es usado como la entrada del algoritmo generador de la llave. Por otro lado, se supone que denota la longitud de las llaves generadas, es decir, si se desea una llave de longitud de 512-bit entonces el parámetro de seguridad será $k = 512$.

2.1.3 Terminología y conceptos básicos de encriptamiento

Dominio y codominio de encriptamiento

- D denota un alfabeto finito llamado *alfabeto de definición.* Por ejemplo el conjunto $A = \{0, 1\}$ es llamado el alfabeto binario y frecuentemente es usado como alfabeto de definición. Nótese que cualquier alfabeto de definición se puede poner como alfabeto binario.

- M denota el conjunto de mensajes, llamado *espacio de mensajes* y consiste en cadenas de elementos del alfabeto de definición. Así, $M \in \{0,1\}^n$, $n \in \mathbb{N}$.

- C denota el conjunto de mensajes cifrados, llamado *espacio de textos cifrados.* C consiste de cadenas de símbolos de un alfabeto de definición. Este puede diferir del alfabeto de definición de M.

Transformaciones de encriptamiento y desencriptamiento

- K denota un conjunto de llaves llamado *espacio de llaves.* Un elemento de K es una llave.

- Cada elemento $e \in K$ determina una única biyección $E_e : M \rightarrow C$. A E_e se le llama *función de encriptamiento* o una *transformación de encriptamiento.* Nótese que E_e debe ser una transformación biyectiva si el proceso es invertible y únicamente se puede recuperar un único mensaje por cada mensaje encriptado.

- Cada $d \in$ K determina una biyección $D_d : C \rightarrow M$. D_d es llamada función desencriptamiento o transformación desencriptamiento.

- A el proceso de aplicación de la transformación E_e a un mensaje $m \in$ M, se le conoce como encriptamiento de m.

- El proceso de aplicación de la trasformación D_d a un texto cifrado c, es usualmente referido como desencriptamiento de c.

- Un protocolo de encriptamiento consiste de un par de conjuntos, el primero es un conjunto $\{E_e : e \in K\}$ de funciones de encriptamiento y el otro conjunto corresponde a $\{D_d : d \in$ K $\}$ de funciones de desencriptamiento, con la propiedad de que para cada $e \in$ K existe una única llave $d \in$ K, tal que $D_d = E_e^{-1}$, esto es $D_d(E_e(m)) = m$ para todo $m \in$ M. Algunas veces el plan de encriptamiento es llamdo $cifrador$.

- Las llaves e y d definidas anteriormente son llamadas *par de llaves* y se denotan por (e, d).

- Para construir un plan de encriptamiento se requiere la selección de un espacio de mensajes M, un espacio de mensajes cifrados C, un espacio de llaves K, un conjunto de funciones

de encriptamiento $\{E_e : e \in K\}$ y el correspondiente conjunto de funciones de desencriptamiento $\{D_d : d \in \mathrm{K}\}$.

La figura 1.1.1 muestra un modelo simple de comunicación entre dos participantes transmitiendo mensajes encriptados.

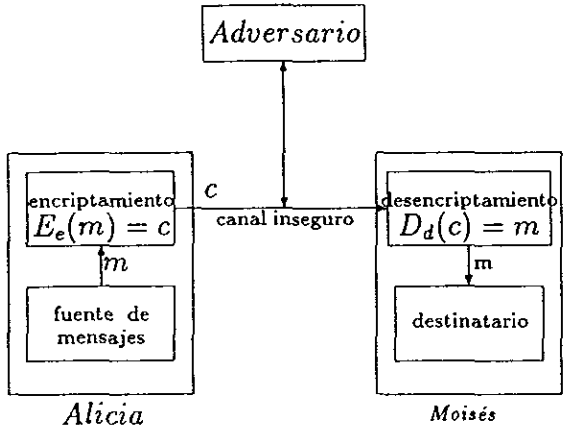


Figura 1.1.1: Esquema de comunicación entre dos partes, usando un método de encriptamiento

Participantes de la comunicación

- Una **entidad** o *parte* es alguien o algo, que envía, recibe, o manipula información. En la Fig. 1.1.1, Alicia y Moisés son entidades.

- Un **emisor**. Es una entidad en la comunicación de dos partes, la cual es el trasmisor legítimo de información. En la Fig. 1.1.1, Alicia es el trasmisor.

- Un **receptor**. Es una entidad en al comunicación entre dos partes, la cual es el receptor de la información. En la Fig. 1.1.1, Moisés es el receptor.

- Un **adversario o intruso**. Es una entidad en la comunicación entre dos partes, el cual no es el emisor ni el receptor y quien trata de derrotar el servicio de seguridad sobre la información entre el emisor y receptor. Un adversario a menudo intentará jugar el papel del receptor o emisor.

Este trabajo está enfocado a las firmas digitales, es decir, se trata de cubrir los aspectos referentes a la autenticidad. En los siguientes capítulos se desarrollará una clase especial de firmas, llamadas FIP.

2.1.4 Firmas digitales

Los sistemas de firmas digitales tienen la misma función sobre los mensajes digitales que la función que han tenido las firmas manuscritas para los documentos sobre papel: Las firmas digitales deben proporcionar autenticidad a los mensajes, con la garantía de que esta autenticidad se pueda probar, de tal forma que una tercera parte pueda intervenir en caso de una disputa entre el firmante y el receptor, ver Figura 1.1.2.

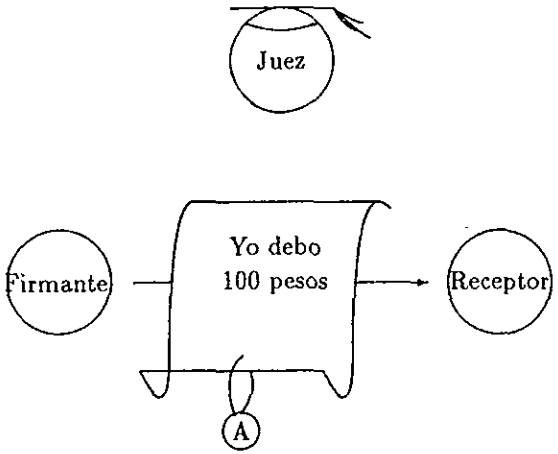


Figura 1.1.2. Ejemplo del uso de las firmas digitales.
El firmante envía una nota de compromiso firmada, en donde la firma está representada por un sello. El receptor se debe convencer de que compromiso proviene del firmante. Es caso de una disputa entre ellos, el receptor debe ser capaz de convencer a un juez de que una falsificación ha ocurrido

Como se ha mencionado anteriormente la información digital puede ser fácilmente copiada, así también las firmas digitales pueden ser copiadas arbitrariamente. Esto implica que las firmas de un firmante sobre mensajes distintos deberán ser diferentes. Por tanto la firma depende directamente del mensaje firmado.

Breve descripción de las aplicaciones de las firmas digitales

El uso que tienen las firmas digitales en la práctica ha sido propuesto principalmente sobre los mismos campos en donde las firmas manuscritas fueron usadas hasta ahora, con la ventaja de que ahora es posible transmitir los mensajes a través de redes de comunicación digital. Por ejemplo, en sistemas de pagos digitales, en la compra de información de bases de datos comerciales, en los envíos de correos usando comunicación digital, etc.

Por tanto las firmas digitales son una herramienta de criptografía fundamental para la *autentificación, autorización y no-rechazo.* El propósito de una firma digital es proporcionar un medio

a una entidad para ligar su identidad con la pieza de información. El proceso de firmado se hace aplicando una transformación sobre el mensaje y alguna información secreta poseída por una entidad.

Estructura

La idea inicial sobre la existencia de las firmas digitales fue publicado en el artículo [DiHe].

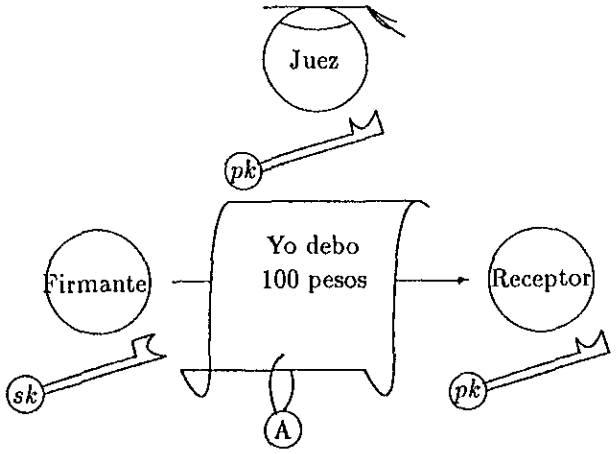


Figura 1.1.3. Componentes de un plan de firmas digitales.
El firmante elige una pareja de llaves (sk, pk), publicando la llave pk. El receptor y el juez conocen pk. Solamente el firmante puede firmar sus mensajes con sk. El receptor y el juez aceptarán el mensaje firmado por el verdadero firmante si y sólo si éste pasa la prueba con pk.

La idea funadamental del artículo dc [DiHe] es la de mostrar que puede haber pares de llaves (sk, pk) con correspondencia mutua (ver Figura 1.1.3): con la llave sk, se pueden firmar los mensajes y con la otra llave pk, se pueden verificar las firmas sobre estos mensajes. Por tanto, una llave es llamada **llave de firma** y la otra **llave verificadora.**

Este plan de firmas se proyecta para que sea usado de la siguiente forma: cualquiera que desee firmar un mensaje generará un par de llaves. El firmante debe de mantener en secreto la llave

de firma y publicará la llave de prueba. Esto nos sugiere renombrar las llaves de firma y de prueba, por *llave secreta* y *llave pública* respectivamente.

Cualquiera puede firmar mensajes con su correspondiente llave secreta, mientras que todos los paticipantes del plan pueden probar estas firmas con la correspondiente llave pública.

Nomenclatura y conjuntos

- M es el un conjunto de mensajes que serán firmados.
- S es el conjuto de elementos llamados *firmas*, posiblemente cadenas binarias de una longitud fija.
- $S_{\mathcal{A}}$ es una trasformación del conjunto de mensajes M a el conjunto de firmas S, y es llamada *transformación de firmas* para la entidad $\mathcal{A}$ (emisor). Las trasformaciones S_A se mantienen en secreto por el emisor, y son usadas para crear firmas sobre los mensajes de M.
- $V_{\mathcal{A}}$ es una transformación del conjunto $M \times S$ al conjunto {aprobar, no-aprobar}. $V_{\mathcal{A}}$ es llamada *transformación de verificación* para las firmas de $\mathcal{A}$, esta transformación es de conocimiento público, y puede ser usada por otras entidades para verificar las firmas creadas por la entidad $\mathcal{A}$.

Definición 2.1.19 *Las transformaciones $S_{\mathcal{A}}$ y $V_{\mathcal{A}}$ proporcionan un plan de firmas digitales para $\mathcal{A}$. Ocacionalmente es usado el término* **sistema de firmas digitales**. ◇

Procedimiento de firmado

El firmante construirá la firma para un mensaje $m \in$ M de siguiente la siguiente forma:

1. Calculará $s = S_{\mathcal{A}}(m)$.

2. Trasmitirá el par (m, s). Donde, s es la *firma* para el mensaje m.

Procedimiento de verificación

Para verificar que un firma s sobre un mensaje m, fué creada por el verdadero firmante el receptor debe hacer lo siguiente:

1. Obtener la función de verificación $V_{\mathcal{A}}$ de $\mathcal{A}$,

2. Calcula $u = V_{\mathcal{A}}(m, s)$ y

3. Aceptará la firma y también aceptará que ha sido creada por $\mathcal{A}$ si $u = aprobar$, y rechazará la firma si $u = no\ aprobar$.

Propiedades requeridas para las funciones de firmado y verificación

Hay algunas propiedades que las transformaciones de firmas y transformaciones de verificación deben satisfacer:

a) s es una firma válida de $\mathcal{A}$ sobre el mensaje m si y sólo si $V_{\mathcal{A}}(m, s) = aprobar$

b) Se desea que sea computacionalmente difícil para cualquier otra entidad, encontrar para algún $m \in M$ y un $s \in S$ tal que $V_{\mathcal{A}}(m, s) = aprobar$.

Los componentes de un plan ordinario de firmas y sus interacciones son descritos en al Figura 1.1.4. Nótese que realmente los componentes de un plan ordinario de firmas son algoritmos. Como se verá mas adelante algunos de estos algoritmos son probabilísticos.

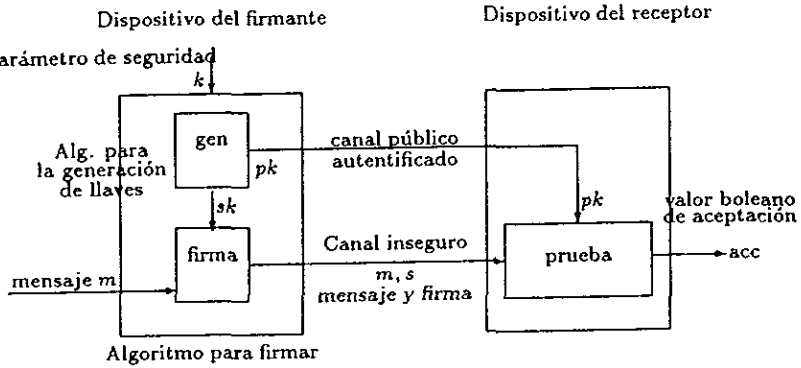


Figura 1.1.4. Componentes de un plan ordinario de firmas. Los componentes son los algoritmos *gen*, *firma* y *prueba*, en donde *gen* y *firma* son probabilísticos. Los valores pk, sk, s y *acc* son los resultados de los algoritmos. Los dispositivos y canales no son parte del plan, sirven solamente de ilustración.

2.1.5 Autenticidad e identificación

Cuando se transmite información digital deseamos garantizar que las entidades sean realmente quienes ellas afirman ser o que la información que se ha transmitido no ha sido manipulada por partes no autorizadas. La autentificación es el objetivo principal de la seguridad de la información que se quiere alcanzar. Como ejemplos de los objetivos específicos que se pueden cubrir son: el control de acceso, la autentificación de la entidad, la autentificación de los mensajes, la integridad de datos, el no rechazo y la autentificación de llaves.

Autentificación es uno de los objetivos mas importantes dentro de la seguridad de la información que se logra con la ayuda de las firmas, y en el caso de transmisión de informacion digital, con las firmas digitales. En los siguientes capítulos, se presentará un tipo especial de firmas.

Identificación

Definición 2.1.20 *Una técnica de identificación o autentificación de la entidad asegura a una de las partes, la identidad de una segunda parte involucrada, y además que la segunda parte fue activa en el momento que se hizo acreedor a ella.* ◇

Generalmente, sólo son nccesarios los datos transmitidos para identificar las partes que se comunicaron.

Ejemplo 2.1.20.1 *Una persona $\mathcal{A}$, proporciona a un cajero automático un número de identificación personal, después de haber introducido una tarjeta, donde esta tarjeta contiene dentro de la cinta magnética información sobre alguien. El cajero automático usa la información de la tarjeta y la información introducida para verificar la identidad del poseedor de la tarjeta. Si la verificación es válida, $\mathcal{A}$ tendrá acceso a varios servicios proporcionados por el cajero.*

Autentificación de datos originales

Definición 2.1.21 *Las técnicas de autentificación de datos originales o autentificación de mensajes proporcionan a la parte receptora que a recibido un mensaje seguro, de la identidad de la parte donde se originó el mensaje.* ◇

A menudo se transmiten los mensajes con información adicional para que la entidad receptora pueda determinar fácilmente la identidad de la entidad donde es originado el mensaje. Esta forma de autentificación no es garantía, pero es útil en situaciones donde una de las partes no es activa en la comunicación.

El siguiente ejemplo deja claro la necesidad que se tiene de autentificar datos originales.

Ejemplo 2.1.21.1 *$\mathcal{A}$ envía a $\mathcal{B}$ un mensaje por medio del correo electrónico. El mensaje puede viajar a través de varios sistemas de redes de comunicación y finalmente se almacenará en un lugar específico en donde $\mathcal{B}$ lo recuperará posteriormente. Generalmente $\mathcal{A}$ y $\mathcal{B}$ no están en contacto directo. Por tanto $\mathcal{B}$ tiene que estar seguro de que el legítimo emisor fue $\mathcal{A}$.*

Firmas digitales en la práctica

Para que las firmas digitales sean útiles en la práctica, además de los conceptos anteriores deberán tener propiedades adicionales:

- Deben ser fáciles de cálcular por el firmante,
- Deben ser fáciles de verificar por cualquier entidad que lo desee y,
- Debe ser computacionalmente seguras contra falsificación.

La propuesta de una firma digital (o cualquier método de firma) es permitir la resolución de disputas. Por ejemplo, una entidad $\mathcal{A}$ (emisor) puede negar en algún momento que ha firmado algún mensaje, o bien otra entidad $\mathcal{B}$ (receptor) puede reclamar falsamente que una firma sobre un mensaje fue producida por $\mathcal{A}$. Una manera de resolver tales problemas es con una *Tercera Parte de Confianza* (TPC) o un *Juez*. La TPC debe ser una entidad que todas las partes involucradas la han acordado por anticipado.

Si $\mathcal{A}$ (firmante) niega que un mensaje m que recibio $\mathcal{B}$ fue firmado por él, entonces $\mathcal{B}$ deberá presentar la firma $s_{\mathcal{A}}$ sobre el mensaje m a la TPC junto con m. Las reglas de la TPC estarán a favor de $\mathcal{B}$ si $V_{\mathcal{A}}(m, s_{\mathcal{A}}) = \textit{aceptar}$ y en favor de $\mathcal{A}$ en caso de una respuesta *contraria*. El receptor aceptará la decisión, si él esta seguro de que la TPC tiene la misma transformación de verificación $V_{\mathcal{A}}$, que emitió la entidad $\mathcal{A}$. $\mathcal{A}$ aceptará la decisión si él está seguro de que la TPC usó $V_{\mathcal{A}}$ y que $S_{\mathcal{A}}$ no ha sido modificado. Por lo tanto la resolución imparcial de las disputas requiere de los siguientes criterios:

1. $S_{\mathcal{A}}$ y $V_{\mathcal{A}}$ deberán tener las siguientes propiedades:
   a) s es una firma válida de $\mathcal{A}$ sobre el mensaje m si y sólo si $V_{\mathcal{A}}(m, s) = \textit{aprobar}$
   b) Se desearía que fuese computacionalmente difícil para cualquier otra entidad encontrar, para algún $m \in M$, un $s \in S$ tal que, $V_{\mathcal{A}}(m, s) = \textit{aprobar}$.

2. La TPC debe tener una copia auténtica de $V_{\mathcal{A}}$.

3. La trasformación de firmado $S_{\mathcal{A}}$ deberá mantenerse en secreto y permanecer segura.

Definición 2.1.22 *Se dice que Una Tercera Parte de Confiaza (TPC) es incondicionalmente segura si es segura para emisores y receptores. Por ejemplo, esta puede tener acceso al secreto y a la llave privada de los usuarios, también como ser encargada de la asociación de llaves públicas para los identificadores.* ◇

Definición 2.1.23 *Una TPC se dice que es funcionalmente segúra si la entidad supone que es honesta e imparcial pero no tiene acceso a la llave privada y llave pública de los usuarios.* ◇

Capítulo 3

Definición general de FIP

En este capítulo comenzaremos con un breve resumen sobre los trabajos enfocados a las firmas intrínsecamente protegidas. Después, se definirán las FIP y se mostrará su relación con las firmas digitales ordinarias.

3.1 Resultados previos

La idea de las firmas intrínsecamente protegidas fue dada en 1989 por Michael Waidner y Birgit Pfitzmann en [WaPf89]. De igual forma que las firmas digitales convencionales, la dificultad de falsificación de las firmas intrínsecamente protegidas también se basa en una suposición criptográfica, pero si una firma es falsificada el supuesto firmante puede probar que tal firma es una falsificación.

En [PfWa90] Birgit Pfitzmann y Michael Waidner definen que las firmas intrínsecamente protegidas son tan difíciles de falsificar como las mejores firmas convencionales, con la propiedad adicional de hacer demostraciones de falsificación. También construyeron las firmas intrínsecamente protegidas sobre parejas de permutaciones libres de pinza. Este trabajo usa la idea de las firmas una-vez de [La]. En estas construcciones las firmas y las llaves son muy grandes y por tanto las firma y la verificación de esta requieren mucho cálculo computacional.

rasgos ella presenta en este extenso y completo volumen lo siguiente: primero discute el objetivo de las firmas digitales y los requisitos resultantes. Posteriormente presenta la historia de las firmas digitales. En seguida presenta la nueva característica de los planes de firmas intrínsecamente protegidos (firmas fail-stop). Después presenta un bosquejo de la definición general de las firmas digitales y da una clasificación de los planes de firmas digitales. Posteriormente se centra en las firmas intrínsecamente protegidas, presentando los beneficios y aplicaciones de éstas. Así como la definición formal. También hace una demostración formal de la relación que tienen las firmas digitales ordinarias y las firmas intrínsecamente protegidas. Continuando con este extenso tema, desarrolla la construcción general de las FIP. Finalmente proporciona los límites inferiores de la eficiencia lograda con las firmas intrínsecamente protegidas.

Finalmente, en 1997 Torben Pryds Pedersen y Birgit Pfitzmann muestran en [PePf] la relación de las FIP con las firmas digitales ordinarias. Además dan una construcción y un par de planes derivados de esta construcción. Estos planes son bastante eficientes para ser usados en la práctica.

Si se desea información mas específica que la presentada en este capítulo puede consultar [PePf],[HePe], [Pf96], [MOV] y [PfWa91].

3.1.1 Propiedades de las FIP

La propiedad fundamental de las firmas intrínsecamente protegidas se puede describir al considerar un juez en una disputa entre el firmante y el receptor de una firma digital. Comunmente, el juez probará si la firma es correcta y dará su veredicto de "*aprobar*" o "*no aprobar*" según sea el caso. Las firmas intrínsecamente protegidas relevarán al juez con una tercera posibilidad: Si el firmante puede probar que la firma es realmente falsa, el juez puede decir "falsificación probada", esto se puede interpretar como que la suposición criptográfica fue rota. Naturalmente, esta posibilidad de distinguir las firmas falsificadas de una firma auténtica puede existir si el falsificador no ha robado la llave secreta del firmante.

Si en un plan de firmas intrínsecamente protegidas se tiene el acuerdo de que todas las firmas para las cuales la falsificación se pueda probar, sean rechazadas, entonces se obtiene un plan de firmas en donde el firmante es incondicionalmente protegido contra falsificaciones, aunque el receptor solo sea computacionalmente protegido, es decir, un adversario con un poder computacional ilimitado ocaciona que una firma aceptada por el receptor sea posteriormente rechazada por un juez. A este plan de firmas se le llama, "*plan de firmas duales*" ya que es similar a las firmas digitales ordinarias con respecto a la seguridad para el firmante y el receptor. Las firmas intrínsecamente protegidas constituyen el primer ejemplo publicado como un plan de firmas dual. Dado que las firmas intrínsecamente protegidas además permiten que el sistema sea parado tan pronto como la suposición criptográfica es rota, se puede decir que estas firmas tienen una noción más precisa que cada uno de los planes de firmas convencionales.

El ejemplo de las firmas intrínsecamente protegidas en el caso especial cuando juegan el papel como firmas duales, puede ser ventajoso si se considera un sistema de pagos electrónicos en donde los usuarios firman digitalmente sus requisitos con el banco. Porque el banco tiene más poder computacional que el usuario y éste puede seleccionar el sistema y elegir los parámetros de seguridad que le convengan, por tanto es razonable proteger al usuario incondicionalmente, aunque el banco puede suponer que el usuario no tiene suficiente poder computacional para rechazar sus firmas. Por tanto el usuario puede firmar en un plan de firmas dual y el banco con firmas digitales ordinarias.

3.1.2 Idea de la construcción

Las firmas intrínsecamente protegidas trabajan de manera muy semejante a las firmas digitales ordinarias. El firmante tiene una llave secreta que usa para hacer firmas y las firmas pueden ser probadas por cualquiera que conozca su correspondiente llave pública. Una firma que pasa esta prueba es llamada **firma aceptable**. Ahora, la idea básica de la construcción de las FIP, es que cada mensaje

tiene muchas firmas diferentes que son aceptables, para las cuales el firmante solamente puede construir una firma (al menos que pueda romper la suposición subyacente, por ejemplo que pueda calcular el logaritmo discreto); esta firma es llamada **firma correcta**. Sin embargo, un falsificador arbitrariamente potente no tiene suficiente información para determinar cual de las muchas firmas aceptables es la correcta sobre un nuevo mensaje. Consecuentemente, con una probabilidad muy grande, un firma falsificada es diferente de la firma correcta. En el momento de que ocurra una falsificación y se obtenga la firma falsificada el firmante expondrá las dos firmas diferentes sobre el mismo mensaje (la firma falsificada y la firma correcta) para hacer la demostración de falsificación.

Note que esta construcción permite a un firmante con un poder computacional no restringido desconocer sus propias firmas y esto sólo puede ocurrir si la suposición computacional es rota, pero esto no es un problema. Ya que una demostración de falsificación no indica por quien fue rota la suposición criptográfica y por tanto pudo haber sido el firmante, pero en cualquier caso el sistema deberá ser parado.

3.2 Definición de firma

Un plan de firmas digitales está definido por los siguientes componentes: [GMR].

1. Un parámetro de seguridad k, el cual es elegido por el usuario cuando crea sus llaves públicas y llaves secretas. El parámetro k determina: la longitud de la firma, la longitud de los mensajes a ser firmados, el tiempo necesario para correr el algoritmo de firma, en general determina la seguridad del plan.

2. Un espacio de mensajes M, que es el conjunto de mensajes para los cuales el algoritmo de firma puede ser aplicado. Sin pérdida de generalidad se asumirá que el espacio de mensajes es representado por cadenas binarias, esto es, $M \in \{0,1\}^+$. Para asegurar que el proceso total de firmado sea polinomial

sobre el parámetro k, se asumirá que la longitud de los mensajes que son firmados estan acotados por k^c, para alguna constante $c > 0$.

3. El número máximo de firmas que pueden ser producidas por el plan de firmas, es generalmente acotado por arriba por un polinomio de bajo grado.

4. Un generador de llave, con el cual el firmante en un tiempo polinomial puede calcular las llaves secretas sk y las llaves públicas pk. Las llaves secretas son algunas veces llamadas información *trampa*.

5. Un método de firmado, el cual produce una firma, $firma(sk, m)$ sobre el mensaje m, usando la llave secreta sk. Este algoritmo puede tener entradas adicionales, tal como, el número de mensajes firmados previamente.

6. Un método para verificar las firmas, el cual verifica si una firma es válida para un mensaje m usando la llave pública pk (esto es, $prueba(pk, m, firma)$ tendrá una salida de *aprobar* si y sólo si la firma es válida).

Si las llaves son generadas correctamente, esto es, usando el generador de llave, estos tres últimos métodos deberán satisfacer los siguientes puntos:

- Si el firmante construye una firma un mensaje correctamente, cualquiera que conozca la llave pública del firmante aceptará la firma.

- Un falsificador con un poder de cómputo polinomial no puede hacer que alguna firma pase la prueba de la firma.

(Nota: Hasta aquí, solo se tiene una definición de "firmas digitales ordinarias")

En un plan de firmas intrínsecamente protegido se agrega un protocolo (con tiempo de ejecución polinomial) que permite al firmante probar a un tercera parte que una firma falsificada es verdaderamente falsificada.

Este protocolo consiste de dos algoritmos más:

7. Un método para construir la demostración de falsificación, y

8. Un método para la verificación de la demostración de falsificación (cualquiera que conozca la llave pública del firmante puede aplicarlo).

Un demostración de falsificación es siempre no-interactiva, es decir, si se ha llegado a la conclusión de que se tiene una demostración de falsificación, ésta será aceptada al hacer la verificación de la prueba, de tal manera que puede ser subsecuentemente mostrado a otras entidades y el sistema puede ser parado como consecuencia. La demostración debe satisfacer dos nuevos requerimientos de seguridad:

- La facilidad para hacer una demostración de falsificación debe ser independiente del poder computacional del falsificador.

- Debe ser difícil para un firmante construir firmas y posteriormente poder demostrar que son firmas falsificadas.

Como se observa estas dos propiedades implican una seguridad contra falsificación (ver la sección 3.4). En un plan de firmas ordinario, el propósito fundamental de las llaves secretas es capacitar al firmante para hacer firmas que nadie puede construir y estas llaves pueden ser seleccionadas por el mismo firmante o por un centro de autentificación de llaves que es elegido por el firmante. Y cómo es igualmente importante que las firmas intrínsecamente protegidas no puedan ser falsificadas, por tanto el firmante también tomará parte en la elección de las llaves. Dado que al firmante se le está permitido rechazar las firmas (falsas), a pesar de que las firmas pasen la prueba pública de firma los receptores de las firmas deben estar seguros de que el firmante no pueda rechazar sus propias firmas. Por tanto es necesario que los receptores o el centro confiado por los receptores también participen en la generación de la llave. En particular, tal centro es necesario si los receptores no se encuentran presentes en el momento de la elección de las llaves. Cuando la llave pública ha sido seleccionada, generalmente es almacenada en un directorio de llaves públicas que tengan una integridad aceptable.

En seguida, se formalizará la defincion para el caso donde el firmante y el centro generan las llaves (sección 3.3). En la sección 3.4 se demostrará que cada plan que satisfaga esta definición es seguro contra las falsificaciones y en la sección 3.5 se discutirá, como los receptores pueden participar en la generación de las llaves.

3.3 Definición de las FIP

La información contenida en lo que resta de este capítulo se obtuvo de [PePf].

En esta sección, se considerará la situación donde el firmante y una entidad de confianza para los receptores generan la llave. A ésta entidad se le llamará **centro** y se denotará por C.

En general se asume que se tiene una complejidad uniforme para este plan. En particular las suposiciones computacionales en la siguiente sección son descritas uniformemente. Esto es, los algoritmos generadores de la llave son probabilísticos, los cuales tienen como entrada adicional una cinta magnética aleatoria que tiene una distribución uniforme.

La seguridad de un plan criptográfico es usualmente determinada por un parámetro de seguridad, el cual especifica el tamaño de las muestras de las llaves de la suposición criptográfica. Ver la página 52, punto 1.

Un plan de firmas intrínsecamente protegido puede ser roto[1] de dos maneras: por un firmante, si éste tiene éxito en construir una firma y posteriormente poder probar que es falsa, o bien, por un falsificador, si éste tiene éxito en construir una firma falsa que el firmante no pueda rechazar. Dado que estos dos ataques tienen diferentes consecuencias para los participantes, es natural definir los parámetros de seguridad para un plan de firmas intrínsecamente protegidas como un par de enteros positivos (k, σ), donde k mide la **seguridad computacional** para los receptores y σ determina la **seguridad incondicional** para el firmante.

Las firmas en estas construcciones dependenderán del número de firmas que se han firmado previamente. Se tratarán todos estos

[1] Vease la sección 1.1.1.

casos tomando a los bits aleatorios como parte de la llave secreta y la firma como una función determinística de la llave secreta y la secuencia de todos los mensajes (previos). La siguiente notación aclara ésto:

El número del mensaje que será firmado es denotado por $i \in \mathrm{IN}$. Además de los parámetros de seguridad σ y k en un plan de firmas intrínsecamente protegidas se agregará un parámetro $N \in \mathrm{IN}$, éste indica el número total de firmas que el firmante puede construir usando la misma llave secreta, por tanto, $1 \leq i \leq N$. Semejante a los parámetros de seguridad, también se asume que N es representado en cadenas unarias (es decir, N es representado por $11\cdots1$). Para hacer más simple la notación, y no estar escribiendo todos los parámetros de seguridad cuando se hace referencia a ellos, se escribirá solamente *par*, es decir $par := (k, \sigma, N)$.

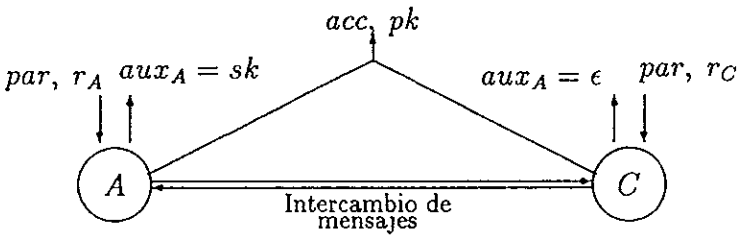


Figura 2.1.1. Descripción de la ejecución correcta del protocolo generador de llaves, G.

Definición 3.3.1 *Un plan de firmas intrínsecamente protegido (se abreviará como: plan de FIP) sobre un espacio de mensaje* $M \subseteq \{0,1\}^+$, *es una quíntupla (G, firma, prueba, dem, ver) donde:*

- *G, es un protocolo para la generación de la llave, el cual es ejecutado en un tiempo polinomial entre dos partes. El protocolo es ejecutado por el firmante con un algoritmo A y el centro con un algoritmo C, quienes darán como entrada*

a par $= (k, \sigma, N)$. *Además cada participante dará un cadena de bits aleatorios*[2] r_A *y* r_C *respectivamente. Cada participante tiene un canal de salida privado y hay un canal para la salida común para A y C, ver la Figura 2.1.1. La salida común es* (acc, pk) *donde* $acc \in \{aceptar, rechazar\}$ *y pk es la llave pública, (se dice que está bien definida si* $acc = aceptar$*). El canal de emisión que tienen en común A y C puede ser individual si uno de los participantes obtiene pk y el otro obtiene acc. Generalmete se denotará a los productos de los canales de salida privada como,* aux_A *y* aux_C *respectivamente, y la salida completa de* $G(k, \sigma, N)$ *es el cuarteto* $(\mathbf{acc}, \mathbf{pk}, \mathbf{aux_A}, \mathbf{aux_C})$. *Si A es ejecutado correctamente,* aux_A *será simplemente la llave secreta, denotado por sk y si C es ejecutado correctamente,* aux_C *es igual una cadena vacia* ϵ.

- *El algorimo de firma, denotado por* **firma**, *es un algoritmo de tiempo polinomial que tiene como entradas la llave secreta, un número i que corresponde al mensaje que será firmado, y una secuencia de mensajes* $m = (m_1, \ldots, m_i)$ *de M, y tiene como salida una firma sobre* m_i. *Así,* $firma(sk, i, m)$ *denota la firma sobre* m_i, *si los mensajes firmados previamente fueron* $m_1, \ldots m_{i-1}$, *y todos los bits aleatorios son considerados como parte de sk (y por tanto originalmente de* r_A*). A ésta se llama* **firma correcta**, *porque está construida con las llaves secretas del firmante.*

- *El algoritmo de* **prueba**, *tiene un tiempo de ejecución polinomial que al tener como entradas la llave pública pk, un mensaje* $m_i \in M$, *y una posible firma* s_i *sobre* m_i, *da como salida "aprobar" o "no aprobar". Si* $prueba(pk, m_i, s_i) = aprobar$, *se dice que* s_i *es una firma aceptable sobre el mensaje* m_i.

- *El algoritmo demostración denotado por* **dem** , *es una algoritmo con tiempo polinomial que al tener como entradas la*

[2]Como A y C son algoritmos probabilisticos, entonces tienen como entrada adicional una cinta mangnética aleatoria que tiene una distribución uniforme. Así, r_A y r_C denotan tales cintas.

llave secreta, un mensaje $m \in M$, *una posible*[3] *firma* s *sobre* m, *y la historia* hist *de mensajes firmados previamente (incluyendo sus firmas) da cualquiera de las siguientes dos salidas: un mensaje de, "no hay falsificación" o una cadena de bits denotada por* demos $\in \{0,1\}^+$.

- *La verificación que se denota por* ver, *es un algoritmo con tiempo de ejecución polinomial que tiene como entradas, la llave pública, un mensaje* $m \in M$, *una posible firma* s *sobre* m *y una cadena demos y tendrá cualquiera de las salidas: "aceptar" o "rechazar". Si el resultado es "aceptar", se dice que se tiene una demostración de falsificación válida.*

Esta quintupla satisface la siguiente propiedad básica de exactitud. Para todo $k, \sigma, N \in \mathbb{N}$, *si* A *y* C *siguen los cálculos preescritos por* G, *cada salida (acc, pk, sk,* ϵ*) de* G *satisface que acc = aceptar y para todo número de mensaje* $i \in \{1, \ldots, N\}$ *y secuencia de mensajes* $m = (m_1, \ldots, m_i)$ *de* M, *se tiene,*

- *Si* s_i = *firma(sk, i, m), entonces prueba(pk, m ,* s_i*) = aprobar, es decir, las firmas correctas son aceptadas.*

$\diamond$

Una salida del algoritmo *dem* tal como, "no hay falsificación" significa que el algoritmo *dem* no es capaz de costruir una demostración de falsificación válida.

Note que *prueba* y *ver* no son tan generales como *firma* y *dem*, porque no dependen de la historia de firmas previas: en general todas las firmas pueden tener diferentes receptores. Por tanto no podemos asumir que un receptor (o verificador) conoce cualquier cosa acerca de la historia, es decir, el conjunto de todos los mensajes y sus respectivas firmas que el firmante ha construido.

Antes de continuar, introduciremos la notación que será usada en el protocolo para la generación de la llave. Un firmante o un

[3]Realmente este algoritmo es dirigido para firmas aceptables. De cualquier modo, es conveniente que esto sea definido mas general. Si S no es una firma aceptable, usualmente se tendrá una salida de "no hay falsificación".

centro que no necesariamente siguen el protocolo G de la definición 3.3.1 serán denotados por $\tilde{A}$ y $\tilde{C}$, respectivamente.

$G_{\tilde{A},C}$ y $G_{A,\tilde{C}}$ denotan los protocolos resultantes y $aux_{\tilde{A}}$ y $aux_{\tilde{C}}$, la salida privada, respectivamente. Un participante fraudulento también puede usar a *par* como entrada. Se denotará la distribución de los resultados de $G_{\tilde{A},C}$ por $G_{\tilde{A},C}(par)$ con entradas de bits aleatorios r_C de C y $r_{\tilde{A}}$ de $\tilde{A}$ si la entrada común es *par*. Similarmente $G_{A,\tilde{C}}(par)$ denota la distribución de los resultados de $G_{A,\tilde{C}}$.

Definición 3.3.2 *Un plan de FIP es seguro para los receptores si y sólo si para todos los algoritmos probabilísticos con tiempo de ejecución polinomial $\tilde{F}$ y $\tilde{F}^*$ (estos algoritmos representan los dos pasos de un firmante fraudulento podría aplicar) se tiene que:*

Si el firmante (fraudulento) usando el algoritmo $\tilde{F}$ ejecuta G con C (para lo cual el firmante con $\tilde{F}$ y C tendrán como salidas sk y ϵ, respectivamente) si el firmante utiliza la llave secreta de $\tilde{F}$, como entrada para $\tilde{F}^*$ y construye una tercia (m, s, dem), entonces la probabilidad de que dem sea una demostración de falsificación válida tiende a cero más rápido que el inverso de cualquier polinomio en k. Esta probabilidad es inducida por la selección aleatoria (uniformemente distribuida) de r_C , $r_{\tilde{A}}$ y la selección aleatoria usada en $\tilde{F}^*$.

Esto es, $\forall, \tilde{F}$ y $\tilde{F}^*$ (algoritmos probabilísticos con tiempo de ejecución polinomial), $\forall \sigma$, N y c, y para k suficientemente grande,

$$P(acc = aceptar,\ y\ ver(pk, m, s, demos) = aceptar :: (acc, pk, aux_{\tilde{F}}, \epsilon) \leftarrow G_{\tilde{F},C}(par);\ (m, s, demos) \leftarrow \tilde{F}^*(aux_{\tilde{F}})) < k^{-c}$$

(recordando que, $par = (k, \sigma, N)$)

$\Diamond$

Hasta este momento no se ha dicho nada sobre la seguridad del firmante. Considerando que no se ha requerido que el firmante confíe en el centro, el firmante debe ser protegido, no importando lo que haga el centro durante el protocolo para la generación de la llave, mientras que el firmante ejecute el protocolo. Además la

propuesta de las firmas intrínsecamente protegidas es que el firmante debe ser seguro contra falsificadores arbitrariamente potentes. Por tanto se considerá un centro arbitrariamente poderoso $\tilde{C}$. Recordando que $[G_{A,\tilde{C}}(par)]$ denota el conjunto de posibles resultados del protocolo si *par* fue la entrada del algoritmo.

Definición 3.3.3 *Consideremos un plan de FIP con un espacio de mensajes* $M \subseteq \{0,1\}^+$, *un conjunto de parámetros par* = (*k*, σ, *N), además un centro arbitrariamente poderoso* $\tilde{C}$. *Si A y* $\tilde{C}$ *han ejecutado G y los resultados fueron (acc, pk, sk,* $aux_{\tilde{C}}$*) con acc = aceptar, se define lo siguiente:*

a) El conjunto de historias posibles es:

$$Hist(sk) := \{(m_1,\ldots,m_j),(s_1,\ldots,s_j) \mid 1 \leq j \leq N \text{ y } (m_i \in M \text{ y } s_i = firma(sk,i,(m_1,\ldots,m_i)) \text{ para } i = 1,\ldots,j\}.$$

Dada una historia hist, se denotará por M(hist) como el conjunto de mensajes firmados en la historia.

b) El conjunto de posibles llaves secretas (desde el punto de vista de un falsificador irrestringido) dados, pk, $aux_{\tilde{C}}$ *y una historia hist es:*

$$SK_{\tilde{C}}(pk,hist,aux_{\tilde{C}}) := \{sk \mid (aceptar,pk,sk,aux_{\tilde{C}}) \in [G_{A,\tilde{C}}(par)] \text{ y } hist \in Hist(sk)\}.$$

$SK_{\tilde{C}}$ *es equipado con una distribución que es inducida por los bits aleatorios del firmante y corresponde a la incertidumbre que tiene un atacante acerca de la llave secreta cuando trata de seleccionar la falsificación "óptima ", esto es, usar la llave secreta del firmante.*

c) El conjunto de falsificaciones con éxito, dada la llave pública pk y una historia hist es:

$$Falso(pk, hist) := \{f = (m, s) \mid m \in M \setminus M(hist) \ \ y$$
$$prueba(pk, m, s) = aprobar\}$$

esto es, el conjunto de firmas aceptables sobre mensajes no contenidos en la historia.

d) Un valor $f \in Falso(pk, hist)$ es una falsificación probable, después de un historia hist, y es denotado por,

$probable(sk, pk, hist, f) \Leftrightarrow$

$$ver(pk,\ m,\ s,\ dem(sk,\ m,\ s,\ hist)) = aceptar$$

esto es, si al aplicar dem se tiene como resultado una demostración de falsificación válida.

$\Diamond$

La siguiente definición nos dice que no importa que es lo que haga un *centro* (arbitrariamente poderoso) durante el protocolo para la generación de la llave, tampoco importa que mensajes ha construido el firmante, y no importa que mensajes y firmas seleccionó el falsificador como una posible falsificación, el firmante puede rechazar la firma falsificada con probabilidad muy grande. La probabilidad esta inducida por las posibles llaves secretas (entre las cuales, el falsificador debe adivinar cuales son las que realmente tiene el firmante).

Hay dos posibles origenes para tener un pequeño error de probabilidad: Una es durante la generación de la llave, en donde A puede ser engañado al aceptar mal un par de llaves, la otra es que el falsificador encuentre exactamente una firma correcta del firmante, es decir su llave secreta.

Definición 3.3.4 *Considerando un plan de FIP, se tiene lo siguiente:*

a) Dada cualquier entrada de parámetros k, σ, $N \in$ IN y cualquier centro $\tilde{C}$, defínase el conjunto de **buenos** *resultados de la generación de la llave como, $Bueno_{\tilde{C}}$, de la siguiente forma:*

$$(sk, pk, aux_{\tilde{C}}) \in Bueno_{\tilde{C}} \Leftrightarrow$$
$$\forall\, hist \in Hist(sk),\ \forall\, f \in Falso(pk, hist):$$

$$P(probable(sk', pk, hist, f) ::$$
$$sk' \leftarrow SK_{\tilde{C}}(pk, hist, aux_{\tilde{C}})) \geq 1 - 2^{-\sigma}$$

Por tanto un resultado es llamado bueno si garantiza que después de cualquier historia, un falsificador irrestringido tiene aún demasiada incertidumbre, acerca de las llaves secretas como para que sus falsificaciones sean probables con gran probabilidad.

b) El plan de FIP es seguro para el firmate $\Leftrightarrow \forall par = (k, \sigma, N)$ *y para cualquier centro* $\tilde{C}$ *arbitrariamente potente,*

$$P((sk, pk, aux_{\tilde{C}}) \notin Bueno_{\tilde{C}}\ y\ acc = aceptar ::$$
$$(acc, pk, sk, aux_{\tilde{C}}) \leftarrow G_{A,\tilde{C}}(par)) \leq 2^{-\sigma}$$

c) Si A y C ejecutan G, cada salida es buena, es decir,

$$(sk, pk, \epsilon) \in Bueno_C.$$

◇

Definición 3.3.5 *Un plan de FIP seguro es un plan de FIP que es seguro para ambos, firmante y receptores.* ◇

3.4 Relación con firmas ordinarias

Seguridad contra falsificación. En la definición anterior de las firmas intrínsecamente protegidas seguras, no se tiene explícitamente cuantificado lo que significa "difícil", para que un falsificador pueda construir firmas aceptables. Es claro que un plan de firmas es ineficiente si es fácil hacer falsificaciones aún cuando puedan ser rechazadas. Se muestra que la definición 3.3.5, realmente supone, que la falsificación es difícil para adversarios con

capacidad de cómputo polinomial. Para ser más precisos, se demostrará que una **falsificación existencial**: es imposible justamente después de un ataque-adaptivo en la selección del mensaje: en este ataque el enemigo puede requerir las firmas de los mensajes que dependen de la llave pública, y también puede obtener las firmas de los mensajes que dependen de firmas previamente obtenidas.

En un ataque-adaptivo en la selección del mensaje que es aplicado contra un plan de firmas, con un parámetro de seguridad l, (el cual se usa como entrada en la generación de la llave) un falsificador con un algoritmo F usando la llave pública como la entrada hace lo siguiente:

1. Repetir una cantidad de veces (dedependiendo de l) lo siguiente: Generar (de alguna manera) un mensaje m y obtener la firma correcta sobre m. (Donde m es un mensaje ya firmado),

2. Hasta obtener como salida la pareja (m', s'), como falsificación exitosa. (Donde s', es una firma sobre un nuevo mensaje m').

Definición 3.4.1 *Un plan de firmas con un parámetro de seguridad l, es seguro contra un ataque-adaptivo en la selección del mensaje si y sólo si para todo $c > 0$ y para todo falsificador probabilístico F, con tiempo polinomial (como el descrito arriba), se tiene que la probabilidad de que F obtenga una pareja (m', s') tal que m' sea diferente*[4] *de todos los mensajes seleccionados en el paso 1 y s' sea una firma aceptable sobre m', es menor que l^{-c} para l suficientemente grande.*

La probabilidad está implicada por los bits aletorios usados en la generación de la llave, los bits aleatorios de F, y la selección de las firmas, si el algoritmo de firma es probabilístico. ◇

El siguiente teorema considerará la seguridad de un plan de firmas intrínsecamente protegido contra un falsificador que no participa en la generación de la llave.

[4]Los mensajes tienen que ser diferentes porque lo que se hace en el paso 1 es generar firmas que pasen la prueba sobre mensajes que ya han sido firmados

Teorema 3.4.1 *Un plan de FIP es seguro contra un ataque-adaptivo en la selección del mensaje, hecho por falsificadores que no participan en G.* ◇

Es decir, solamente tienen como entrada a *par* y el resultado pk de una ejecución correcta de G. El parámetro k de este plan de FIP juega el papel del parámetro l de la Definición 3.4.1.

Demostración: Considerese un plan de FIP con las definiciones 3.3.1 y 3.3.5. Supongase además que existe un falsificador con un algoritmo probabilístico F con tiempo polinomial (como el descrito en la página 62), tal que sobre el par de entradas (par, pk) obtiene un nuevo mensaje y un firma aceptable sobre este mensaje, es decir una pareja (m', s') con una probabilidad de $p(par)$. Esta probabilidad es implicada por todos los bits aleatorios usados en la generación de la llave, y en F. Dado que F se ejecuta con tiempo polinomial, un firmante que tenga como propósito de defraudar a un receptor puede usar el mismo algoritmo.

1. Ejecutar G correctamente con el centro C, para obtener como resultado (sk, pk).

2. Ejecutar F y alternarlo con el algoritmo real de firmado para obtener una historia *hist* y una falsificación $f = (m', s')$.

3. Usar el algoritmo *dem* para calcular una demostración de falsificación sobre (m', s'), dada la llave secreta sk y la historia *hist*.

El argumento es el siguiente: Por un lado, se tiene que la seguridad para el firmante contra un falsificador con un algoritmo F, implica que el paso 3, induce a una demostración de falsificación válida con una probabilidad no insignificante. Por otra parte, esto es lo mismo, que el firmante ejecute el algoritmo completo (es decir, el paso 1 es equivalente a $\tilde{F}$ y los pasos 2 y 3 a $\tilde{F}^*$ de la definición 3.3.2), por lo tanto esto contradice la seguridad para el receptor.

La historia y los mensajes con sus respectivas firmas, para el paso 2, tiene exactamente la misma distribución que las que el

falsificador pudiera haber construido (es decir, la distribución usada en la definición 3.4.1). Esto implica dos cosas: La primera es, que el par (m', s') es una falsificación exitosa con probabilidad de $p(par)$. La segunda, es que cualquiera que sea el caso, el firmante puede rechazar la falsificación, es decir, el paso 3 resulta una demostración de falsificación válida, con una probabilidad de al menos $q := 1 - 2^{-\sigma}$, porque $(sk, pk, \epsilon) \in Bueno$[5]. Donde q es una probabilidad que se tiene sobre las posibles llaves secretas, dadas en cualquier historia. Por tanto un firmante (fraudulento) que pueda hacer una falsificación con éxito, es decir, hacer un firma que pase la prueba y además poder demostrar que ésta es una firma falsa, está dado por la siguiente probabilidad:

$$p(par)(1 - 2^{-\sigma}) \geq p(par)/2.$$

Es decir, un firmante con un algoritmo F que pudo ejecutar los paso 1 2 y 3.

Dado que el plan de FIP es seguro para los receptores, esto implica que $p(par)$ debe ser insignificante, en función de k. ■

La seguridad óptima de un plan de firmas digital fue definida en [GMR]. En base a esta definición, el teorema anterior puede ser reconstruido para que cumpla con todas las características de un plan [GMR]. El siguiente teorema nos muestra esto.

Teorema 3.4.2 *Cada plan de firmas intrínsecamente protegido seguro, puede ser usado para construir un plan de firmas ordinario (igualmente eficiente) que sea seguro contra un ataque-adaptivo la selección del mensaje.*

◇

Demostración. La razón principal por la que un plan de firmas intrínsecamente protegido no pueda satisfacer la definición de un plan de firmas ordinario, que es seguro contra un ataque-adaptivo en la selección de los mensajes, es por la generación de la llave. Esto se soluciona de la siguiente forma: Dar como entradas únicamente a (k, N), donde el firmante y el centro con los algoritmos A y C

[5] ver definición 3.3.4(a,c)

respectivamente, ejecutarán el protocolo para la generación de la llave, con $\sigma := 1$. Los algoritmos *firma* y *prueba* permanecen iguales, y los algoritmos *dem* y *ver* son omitidos. Por el teorema 3.4.1 se concluye que este plan es un plan seguro de firmas contra un ataque-adaptivo en la selección del los mensajes.

■

Una segunda relación que se puede obtener del teorema 3.4.1 se mostrará en la sección 3.5. Aquí se reforzará la seguridad incondicional para el firmante al combinarla con la seguridad que ya se tiene, contra un ataque-adaptivo en la selección del mensaje, el cual puede ser hecho por cualquiera, es decir, aun sin confiar en el centro.

3.5 Plan de FIP con receptores conocidos

En las definiciones de la sección 3.3 la llave es generada por el firmante y por un centro de confianza de los receptores. En esta sección se discutirá brevemente un plan de firmas intrínsecamente protegido donde los receptores mismos participan en la generación de la llave. Para ser más precisos, se debe distinguir los receptores y los portadores del riesgo. Un portador del riesgo es quien tiene desventaja si una demostración de falsificación ha sido aceptada como una firma falsificada. Entonces la compañía aseguradora - y no el receptor- es el portador del riesgo y además la compañía aseguradora debe tener confianza en el proceso de la generación de llave. En seguida se describirá brevemente un plan donde se tiene la participación de un receptor en al generación de la llave y *posteriormente se hará la descripción para varios receptores.*

Un receptor. En este caso, simplemente se reemplaza el centro de confianza por el receptor. Se puede pensar que un plan de firmas intrínsecamente protegido con un sólo receptor pudiera tener desventaja para el receptor, por no haber un centro de confianza, pero no es así, porque cualquiera que conozca la llave pública

puede probar las firmas y también verificar la demostración de falsificación. Este caso puede ser aplicado en sistemas de pagos electrónicos para las firmas de las solicitudes de los usuarios a un banco.

Muchos receptores. La definición de un plan de FIP con muchos receptores que participan en el protocolo de la generación de llave, también se sigue de las definiciones previas. El firmante junto con cada participante i, darán la entrada *par* y una cadena secreta de bits aleatorios r_A y r_i respectivamente, y la salida común de la ejecución de $G_{A,i}$ puede ser, *rechazar* o $(aceptar, pk)$. (Se asume que todos los receptores están de acuerdo con el parámetro de seguridad k). El firmante obtendrá la llave secreta sk, correspondiente a la llave pública pk. La seguridad del firmante se define como un plan de FIP *normal* y la seguridad de los receptores se define esencialmente por los requerimientos de la Definición 3.3.2, para cada uno de ellos. En otras palabras, si un receptor sigue correctamente el protocolo de generación de llave, queda automaticamente asegurado de que un firmante (con un poder de cómputo polinomial) no pueda rechazar sus propias firmas, aún cuando el firmante y los receptores restantes se cooperen entre si, porque esto contradice la seguridad para el receptor según la definición 3.3.2.

En seguida se mostrará como un plan de FIP con muchos receptores puede ser construido a través del plan de FIP en el sentido de la definición 3.3.1. En los planes de FIP el trabajo del centro en el protocolo de la generación de la llave es seleccionar una serie de bits aleatorios. En este plan es totalmente fácil reemplazar el centro por muchos receptores porque ellos sólamente tienen que ejecutar el protocolo.

Construcción 2.1.1 (muchas llaves). Sea R el número de receptores. Cada receptor ejecuta G con el firmante una vez. Se tiene un resultado de R pares de llaves (sk_j, pk_j), $j = 1, \ldots, R$. La llave secreta del firmante es definida como $(sk_1, \ldots, sk_R)$ y la llave pública como $(pk_1, \ldots, pk_R)$. Todas las firmas y además las demostraciones de falsificación consisten de R partes, una por cada

par de llave. Note que cualquiera de todas los partes puede probar cada firma o demostrar la falsificación.

◇

La seguridad del j-ésimo receptor es garantizada, ya que si él, tiene una ejecución correcta de G, será computacionalemente difícil calcular una demostración de falsificación válida para pk_j. La seguridad del firmante es garantizada porque si se tiene una falsificación con éxito, él puede probar la demostración de falsificación para cada par de llaves con gran probabilidad.

Como una consecuencia inmediata del Temorema 3.4.1, este plan es seguro contra un ataque-adaptivo en la selección del mensaje, si al menos uno de los receptores ejecuta G correctamente con el firmante, aunque el falsificador coopere con los $R-1$ receptores restantes. También se puede lograr la seguridad aunque el falsificador coopere con todos los receptores, al permitirle al firmante formar parte en el plan, como un receptor *más*, es decir, el firmante generará adicionalente un par de llaves (sk_{R+1}, pk_{R+1}), que serán completamente propias, el cual será tratado como cualquier otro par de llaves.

Capítulo 4

Fundamentos matemáticos de las FIP

En este capítulo se presentará las bases de las firmas intrínsecamente protegidas. Las secciones 4.1 y 4.2 presentan la parte matemática y las secciones 4.3 y 4.4 la parte computacional.

Un poco de notación: si a y b son enteros tales que a divide a b lo denotaremos por $a \mid b$. Si G es un conjunto denotaremos a la cardinalidad de G, por $|G|$. El máximo común divisor de a y b será denotado por $mcd(a, b)$

Si el lector está interesado con estos temas puede consultar [Pf96, Her].

4.1 Teoría de conjuntos, grupos y números

4.1.1 Un poco sobre conjuntos

Definición 4.1.1 *Si A es un conjunto dotado de una relación de equivalencia $\sim$, entonces la* clase de equivalencia *de $a \in A$, es el conjunto $\{x \in A \mid a \sim x\}$. Este conjunto es denotado por $cl(a)$ o $[a]$.* ◇

Teorema 4.1.1 *Las distintas clases de equivalencia constituyen una partición de A.* ◇

◇

Demostración Se tendrán que probar que se cumplen las tres propiendades de una relacion de equivalencia.

1. $\forall\ a \in G$, $aa^{-1} = e \in H$, ya que H es un subgrupo, $a \equiv a \pmod{H}$.
2. $\forall\ a,\ b \in G$, $ab^{-1} \in G$, y como H es un subgrupo, $(ab^{-1})^{-1} = ba^{-1} \in H$, $a \equiv b \pmod{H} \Rightarrow b \equiv a \pmod{H}$.
3. $\forall\ a,\ b,\ c \in G$, se tiene que $ab^{-1} \in H$ y $bc^{-1} \in H$, ya que es un H subgrupo, entonces $(ab^{-1})(bc^{-1}) = ac^{-1} \in H$. Por lo que si $a \equiv b \pmod{H}$ y $b \equiv c \pmod{H} \Rightarrow a \equiv c \pmod{H}$

1, 2, y 3 muestran que la relación considerada es de equivalencia.

■

Definición 4.1.6 *Si H es un subgrupo de G y $a \in G$, entonces al conjunto $Ha = \{ha \mid h \in H\}$, se le llama* **clase lateral derecha** *de H en G.* ◇

Lema 4.1.2 $\forall\ a \in G$, $Ha = \{x \in G \mid a \equiv x \pmod{H}\}$ ◇

Demostración Sea $[a] = \{x \in G \mid a \equiv x \pmod{H}\}$.

$x \in Ha \Leftrightarrow x = ha$, para alguna $h \in H$, o bien $h = xa^{-1}$, como H es subgrupo $h^{-1} = ax^{-1} \in H \Leftrightarrow a \equiv x \pmod{H} \Leftrightarrow x \in [a]$, por tanto $Ha = [a]$.

■

Comentario. La colección de todas las clases laterales derechas forman una partición.

Lema 4.1.3 *Hay una correspondencia biyectiva entre dos clases laterales derechas cualesquiera de H en G.* ◇

Demostración Dadas cualesquier dos clases Ha y Hb, defínase una relación de tal forma que cada $ha \in Ha$ se le asocia $hb \in Hb$. Es claro que esta relación es sobre. Para probar que es inyectiva tomese $h_1b = h_2b$ con $h_1, h_2 \in H$, entonces $h_1 = h_2$ así, $h_1a = h_2a$, por lo tanto se tiene una relación inyectiva. ■

Teorema 4.1.2 *(Lagrange). Si G es un grupo finito y H es un subgrupo de G, entonces $|H| \mid |G|$.*

Demostración Primero se necesita saber cuantos elementos tiene una clase lateral derecha. Notese que $H = He$, por tanto H, es una clase lateral derecha, así por los lemas 4.1.2 y 4.1.3 el $|H|$ es el número de elementos de cualquier clase lateral derecha, ya que dos clases laterales derechas o son iguales o no tienen elemento en común alguno. Sea $G = \cup_{a \in G} Ha$. Tomemos sólo las clases disjuntas, $G = \cup^{\bullet} Ha$, y como $|Ha| = |H|$, $G = |\cup^{\bullet} Ha|$, por lo que $G = \sum Ha$, por tanto $|G| = k|H|$. ∎

Definición 4.1.7 *Si G es un grupo y $a \in G$ el orden de a es el entero positivo mas pequeño t, tal que $a^t = e$, denotado por $ord(a)$.* ◇

Un resultado inmediato del teorema 4.1.2 es que si el orden de G es primo, entonces G solamente tiene como subgrupos los triviales, G y $\{1\}$.

Definición 4.1.8 *Sea G un grupo y $a \in G$, Sea $< a >= \{a^i \mid i = 0, \pm 1, \pm 2, \cdots\}$, claramente $< a >$ es un subgrupo de G llamado* **subgrupo cíclico** *generado por a. En particular si $< a >= G$, G es cíclico.* ◇

Corolario 4.1.3 *Si G es un grupo finito y $a \in G$, $ord(a) \mid |G|$.*

Demostración Como a proporciona un subgrupo de G, denotado por $< a >$. Por el teorema 4.1.2 (de Lagrange) se tiene que $ord(a) = | < a > | \mid |G|$. ∎

Corolario 4.1.4 *Si G es un grupo finito y $a \in G$, $a^{|G|} = e$.*

Demostración Por el colorario 4.1.3, $|G| = ord(a)k$, asi $a^{|G|} = a^{(ord(a))k} = e^k = e$. ∎

Definición 4.1.9 *Un subgrupo N de G es un subgrupo normal si y sólo si $gNg^{-1} = N \ \forall \ g \in G$.* ◇

En particular si G es abeliano N todo subgrupo de G es normal.

Teorema 4.1.5 *Si G es un grupo y N es un subgrupo normal en G, entonces $G/N = \{Na \mid a \in G\}$ (es decir, la colección de todas las clases laterales derechas) es un grupo con la operación $NaNb = Nab \; \forall \; a, b \in G$. A este grupo se llama* **grupo cociente** *o* **grupo factor** *de G por N.*

Demostración La demostración se sigue de propiedades conocidas de subgrupos normales. ∎

Obs. Con la notación del teorema precedente, se tiene que si G es abeliano entonces G/N también lo es.

Lema 4.1.4 *Si G, es un grupo finito y N es un subgrupo normal de G, entonces $|G/N| = \frac{|G|}{|N|}$* ◇

Demostración Como se vió en la demostración del teorema 4.1.2, todas las clases laterales derechas tienen el mismo número de elementos y esta cantidad es el $|N|$. Si el grupo G, es finito el conjunto de clases laterales derchas Na llenan todo G. Por tanto $|G|/|N|$ es el número de clases laterales derechas de G por N. Por tanto $|G/N| = \frac{|G|}{|N|}$.

∎

Homomorfismos

Definición 4.1.10 *Sean G, $\bar{G}$ dos grupos, un mapeo $\phi : G \to \bar{G}$ es un homomorfismo de grupos si $\forall \; a, b \in G$, $\phi(ab) = \phi(a)\phi(b)$.* ◇

Definición 4.1.11 *Si ϕ es un homomorfismo de G a $\bar{G}$, el núcleo de ϕ, denotado por K_ϕ, se define como el conjunto $K_\phi = \{x \in G \mid \phi(x) = \bar{e}\}$ donde $\bar{e}$ es el elemento identidad de $\bar{G}$.* ◇

Obs. K_ϕ es subgrupo de G.

Lema 4.1.5 *Si ϕ es un homomorfismo, de G en $\bar{G}$ entonces:*

1. *$\phi(e) = \bar{e}$, el elemento identidad de $\bar{G}$.*

2. $\phi(x^{-1}) = \phi(x)^{-1}\ \forall\ x \in G$.

$\Diamond$

Demostración

1. $\phi(x)\bar{e} = \phi(x) = \phi(xe) = \phi(x)\phi(e) \Rightarrow \bar{e} = \phi(e)$
2. $\bar{e} = \phi(e) = \phi(xx^{-1}) = \phi(x)\phi(x^{-1})$. De la misma forma vale $\phi(x^{-1})\phi(x) = \bar{e}$. Por tanto $\phi(x^{-1}) = \phi(x)^{-1}$.

■

Lema 4.1.6 *Si ϕ es un homomorfismo de G en $\bar{G}$, de núcleo K, entonces el conjunto de todos las preimagenes de $\bar{g} \in \bar{G}$ bajo ϕ en G está dado por el conjunto $Kx = \{kx \mid k \in K\}$, donde x es una preimagen particular de $\bar{g}$.* $\Diamond$

Demostración Sea K el núcleo de ϕ y sea x una preimagen de $\bar{g}$, es decir, $\phi(x) = \bar{g}$. Como ϕ es un homomorfismo $\phi(kx) = \bar{e}\bar{g} = \bar{g}$, $\forall\ k \in K$, ya que $\phi(kx) = \phi(k)\phi(x) = \phi(x)\bar{e} = \phi(x)$.

Lo que ahora se necesita probar es que exactamente $|K|$ es el número de preimagenes. Así pues, considerese un elemento $z \in G$ tal que $\phi(z) = \bar{g} = \phi(x)$, entonces $\phi(z)\phi(x)^{-1} = \bar{e}$, entonces $\phi(zx^{-1}) = e$, así $zx^{-1} \in K$, por tanto $z = Kx$.

Por tanto el número de preimagenes está dado por la cardinalidad del kernel. ■

Definición 4.1.12 *Un homomorfismo ϕ de G en $\bar{G}$ se dice que es un isomorfismo si ϕ es binyectivo.* $\Diamond$

Teorema 4.1.6 Teorema de Cauchy para grupos abelianos. *Supongase que G es un grupo abeliano finito y que $p/Ord(G)$ donde p es un número primo. Entonces hay un elemento $a \neq e \in G$ tal que $a^p = e$.*

Demostración Procederemos por inducción sobre el $|G|$. En otras palabras, suponemos que el teorema es válido para todos los grupos abelianos que tengan menos elementos que G. Basandonos en esto queremos probar que el resultado también se verifica para

G. Para comenzar con la inducción notemos que el teorema es cierto para grupos que tienen un dos elementos.

Si G no tiene ningún subgrupo $H \neq (e)$, G, entonces G debe ser cíclico de orden primo. Este primo debe ser p y G, ciertamente, tiene $p-1$ elementos $a \neq e$ que satisfacen $a^p = a^{|G|} = e$.

Supongamos que G tiene un subgrupo N distinto de los triviales e y G. Si $p/|N|$, de acuerdo con la hipótesis de inducción, como $|N| < |G|$ y N es abeliano, entoces hay un elemento $b \in N$, $b \neq e$, que satisface $b^p = e$; como $b \in N \subset G$ habríamos con ello exhibido un elemento del tipo requerido. Podemos, por tanto, suponer que p no divide al $|N|$. Como G es abeliano, N es un subgrupo normal de G, de modo que G/N es un grupo. Además, $|G/N| = \frac{|G|}{|N|}$, y como p no divide $|N|$, $p/\frac{|G|}{|N|} < |G|$. Además, como G es abeliano, G/N es abeliano. Luego, por nuestra hipótesis de inducción, hay un elemento $X \in G/N$ que satisface $X^p = e_1$, (el elemento unidad de G/N), $X \neq e_1$. Los elementos de G/N son de la forma $X = Nx$, $x \in G$, por lo que $X = Nb$ para alguna $b \in G$, así $X^p = (Nb)^p = Nb^p$. Como $e_1 = Ne$ y $X^p = e_1$, $X \neq e_1$ entonces $Nb^p = N$, $Nb \neq N$. Así $b^p \in N$ pero $b \notin N$. Usando uno de los corolarios del teorema de Lagrange, $(b^p)^{|N|} = e$. Es decir $b^{|N|p} = e$. Sea $c = b^{|N|}$.

Ciertamente $c^p = e$. Para demostrar que c, es un elemento que satisface la conclusión del teorema debemos finalmente mostrar que $c \neq e$. Pero si $c = e$, $b^{|N|} = e$, y, por tanto, $(Nb)^{|N|} = N$. Combinando esto con $(Nb)^p = N$, y p no divide al $|N|$, y siendo p un número primo, encontramos que forzosamente habría que cumplirse que, $Nb = N$ entonces $b \in N$, lo cual es una contradicción. Por lo que $c \neq e$ y $c^p = e$, y hemos completado la inducción. ∎

4.1.3 Anillo de los enteros módulo n

En seguida se presentará la estructura de los enteros módulo n. Además se enfatizará la atención a los residuos cuadráticos y a las raíces cuadradas, ya que la función cuadrática juega un papel importante en los planes que se presentarán en los siguientes

capítulos.

Los anillos

Definición 4.1.13 *Un conjunto no vacío A, se dice que es un* **anillo asociativo,** *si en A están definidas dos operaciones, denotadas por "+" y "·" respectivamente, donde $(A,+)$ es un grupo abeliano y además para cualquier $a, b, c \in A$ se tiene,*

1. $a \cdot b \in A$
2. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
3. $a \cdot (b + c) = a \cdot b + a \cdot c$ *y* $(a + b) \cdot c = a \cdot c + b \cdot c$

◇

Definición 4.1.14 *Un campo es un anillo conmutativo, es decir $\forall a, b \in A$ $ab = ba$ tal que, todos los elementos distintos de cero tienen inverso multiplicativo.*

◇

Definición 4.1.15 *Si a y b son enteros, entonces se dice que a es congruente con b módulo n; denotado por $a \equiv b \pmod{n}$, si n divide a (a-b).* ◇

Definición 4.1.16 *El anillo de los enteros módulo n, denotado por $\mathbb{Z}_n$ es el conjunto de (clases de equivalencia) enteros $\{0, 1, \ldots n-1\}$.*

◇

De aquí en adelante para n impar se tomará la representación simétrica de $\mathbb{Z}_n$, es decir $\mathbb{Z}_n = \{-\frac{n-1}{2} \cdots -1, 0, 1, \cdots \frac{n-1}{2}\}$.

Definición 4.1.17 *El grupo multiplicativo de $\mathbb{Z}_n$, es $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n \mid mcd(a, n) = 1\}$. Llamado grupo de unidades. En particular si n es primo $\mathbb{Z}_n^* = \{a \mid 1 \leq a \leq n-1\}$.* ◇

Lema 4.1.7 *Para cualquier primo p el anillo $\mathbb{Z}_p$ es un campo.*

◇

Demostración Como $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\}$ y como $\mathbb{Z}_p^*$ es un grupo conmutativo y todo elemento tiene su inverso. Por tanto se concluye que $\mathbb{Z}_p$ es un campo. ∎

Teorema 4.1.7 Chino del residuo. *Si* $n_1, n_2, n_3, \ldots n_k \in \mathbb{N}$, *tales que* $mcd(n_i, n_j) = 1 \ \forall \ i \neq j$ *y sean* $a_1, a_2, \ldots a_k \in \mathbb{Z}$ *tales que* $mcd(a_i, n_i) = 1$ *entonces existe* $x \in \mathbb{Z}$, *tal que,*

$mcd(x, n_1 \cdots n_k) = 1$ *y* $x \equiv a_i \pmod{n_i}$ *para* $i = 1, \cdots, k$

el entero x *es único si* $1 \leq x \leq n_1 n_2 \cdots n_k - 1$.

Este teorema es equivalente a que la siguiente aplicación,

$$\mathbb{Z}^*_{n_1 n_2 \cdots n_k} \rightarrow \mathbb{Z}^*_{n_1} \times \ \mathbb{Z}^*_{n_2} \times \cdots \times \mathbb{Z}^*_{n_k}$$

$$[a]_{n_1 n_2 \cdots n_k} \rightarrow ([a]_{n_1}, [a]_{n_2}, \cdots, [a]_{n_k})$$

sea un isomorfismo de grupos.

◇

Residuos Cuadráticos y Raíces Cuadradas

Definición 4.1.18 *Sea* $a \in \mathbb{Z}_n^*$. *Se dice que* a *es un* **residuo cuadrático** *módulo* n, *o un cuadrado módulo* n, *si existe* $x \in \mathbb{Z}_n^*$, *tal que,* $x^2 \equiv a \pmod{n}$. ◇

Definición 4.1.19 *Sea* $n \in \mathbb{Z}$, $QR_n := \{a \in \mathbb{Z}_n^* \mid a$ *es un residuo cuadrático módulo* $n\}$, *a este conjunto se le llama, conjunto de residuos cuadráticos módulo* n. *Denotamos por* $-QR_n = \{-x \mid x \in QR_n\}$. $\tilde{QR_n} := \mathbb{Z}_n^* \setminus QR_n$

El símbolo de Legendre es un herramienta útil para saber si un entero a es o no es residuo cuadrático módulo un primo p .

Definición 4.1.20 *Sea* p *un primo impar y* a *un entero. El símbolo de Legendre* $(\frac{a}{p})$ *es,*

$$(\frac{a}{p}) = \begin{cases} 0 & si \ p|a \\ 1 & si \ \ a \in QR_p \\ -1 & si \ \ a \in \tilde{QR_p} \end{cases}$$

◇

Si p es un primo impar y $a, b \in \mathbb{Z}$. Entonces el símbolo de Legendre tiene las siguientes propiedades:

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \pmod{p}$. En particular $(\frac{1}{p}) = 1$ y $(\frac{-1}{p}) = (-1)^{(p-1)/2}$. Por tanto $-1 \in QR_p$, si $p \equiv 1 \pmod 4$ y $-1 \in R\tilde{Q}_p$, si $p \equiv 3 \pmod 4$.

ii) $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$. Por tanto si $a \in \mathbb{Z}_P^*$, entonces $(\frac{a^2}{p}) = 1$.

iii) Si $a \equiv b \pmod p$, entonces $(\frac{a}{p}) = (\frac{b}{p})$.

iv) Ley de los residuos cuadráticos. Si q es un primo impar distinto de p, entonces $(\frac{p}{q}) = (\frac{q}{p})(-1)^{\frac{(p-1)(q-1)}{4}}$

Teorema 4.1.8 *x es un residuo cuadrático módulo $n = n_1 n_2 \cdots n_k$ si y sólo si x es un residuo cuadrático módulo cada n_i para $i = 1, 2, \ldots k$.* ◇

Demostración: Considerando el teorema del residuo chino, es decir, el isomorfismo,

$$\mathbb{Z}^*_{n_1 n_2 \cdots n_k} \to \mathbb{Z}^*_{n_1} \times \mathbb{Z}^*_{n_2} \times \cdots \times \mathbb{Z}^*_{n_k}$$

$$[a]_{n_1 n_2 \cdots n_k} \to ([a]_{n_1}, [a]_{n_2}, \cdots, [a]_{n_k})$$

demostraremos que:

- Sea x un residuo cuadrático módulo n, así pues si
  $mcd(x, n_1 n_2 \cdots n_k) = 1 \Rightarrow$ si $x \in \mathbb{Z}_n^*$ $\exists$ $a \in \mathbb{Z}_n^*$ tal que
  $[x]_n = [a]_n^2 = [a^2]_n$, es decir $x^2 \equiv a \pmod n$,
  como $f([x]_n) = f([a^2]_n)$
  donde $f([x]_n) = ([x]_{n_1}, [x]_{n_2}, \ldots [x]_{n_k})$
  y $f([x]_n) = ([a^2]_{n_1}, [a^2]_{n_2}, \ldots [a^2]_{n_k})$
  entonces $[x]_{n_i} = [a]_{n_i}^2$ $\forall$ $i = 1, 2, \ldots, k$
  por tanto $y \equiv a^2 \pmod{n_i}$ $\forall$ $i = 1, \ldots, k$.

- Supongamos ahora que si x es residuo cuadrático módulo n_i para $i = 1, \ldots, k \Rightarrow \exists a_i$ tal que $x \equiv a_i^2 \pmod{n_i}$ para $i = 1, \ldots k \Rightarrow [x]_{n_i} = [a^2]_{n_i}$ para $i = 1, 2, \ldots, k$.

  donde $([a]_{n_1}, [a]_{n_2}, \cdots, [a]_{n_k}) \in \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^* \times \cdots \times \mathbb{Z}_{n_k}^*$,

  como f es un isomorfismo, $\exists\ [c]_n \in \mathbb{Z}_n^*$, tal que

  $f([c]_n) = ([a_1]_{n_1}, [a_2]_{n_2}, \ldots [a_k]_{n_k})$, de aquí, por propiedeades de isomorfismo

  $f([c^2]_n) = ([c^2]_{n_1}, [c^2]_{n_2}, \ldots [c^2]_{n_k}) = f([c]_n[c]_n) =$

  $([a_1^2]_{n_1}, [a_2^2]_{n_2}, \ldots [a_k^2]_{n_k}) = ([x]_{n_1}, [x]_{n_2}, \ldots [x]_{n_k}) =$

  $f([x]_n)$, siendo f inyectiva se sigue que

  $[x]_n = [c^2]_n$

  es decir, $x \equiv c^2 \pmod{n}$.

■

Símbolo de Jacobi

Si n es un entero compuesto de la forma $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$. Entonces el símbolo de Jacobi de un entero esta dado por:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_1}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k}.$$

El símbolo de Jacobi, tiene las siguientes propiedades:

Sean $m \geq 3$ y $n \geq 3$ enteros impares, y $a, b \in \mathbb{Z}$. Entonces se tiene:

i) $\left(\frac{a}{n}\right) = 0$, 1 ó -1. Además si $\left(\frac{a}{n}\right) = 0$ si y sólo si $mcd(a, n) \neq 1$.

ii) $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$. Por lo tanto si $a \in \mathbb{Z}_n^*$, entonces $\left(\frac{a^2}{n}\right) = 1$.

iii) $\left(\frac{a}{nm}\right) = \left(\frac{a}{n}\right)\left(\frac{a}{m}\right)$.

iv) Si $a \equiv b \pmod{n}$, entonces $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$.

v) $\left(\frac{1}{n}\right) = 1$.

vi) $(\frac{-1}{n}) = (-1)^{(n-1)/2}$. Por tanto $(\frac{-1}{n}) = 1$, si $n \equiv 1 \pmod 4$ y $(\frac{-1}{n}) = -1$, si $n \equiv 3 \pmod 4$.

vii) $(\frac{m}{n}) = (\frac{n}{m})(-1)^{(m-1)(n-1)/4}$, entonces $(\frac{m}{n}) = (\frac{n}{m})$ a menos que n y $m \equiv 3 \pmod 4$.

El subgrupo de residuos con símbolo de Jacobi $+1$ es denotado por $\mathbb{Z}_n^*(+1)$. Así, $QR_n \subseteq \mathbb{Z}_n^*(+1)$, y en general $\mathbb{Z}_n^*(+1)$ es mucho mas grande que QR_n y $\mathbb{Z}_n^*(+1)$ siempre contiene la mitad de los elementos de $\mathbb{Z}_n^*$.

El símbolo de Jacobi, es de gran interés porque este puede ser calculado eficientemente para cualquier par de números.

4.1.4 Enteros Blum generalizados

La mayoría de los planes criptográficos, basados en una suposición criptográfica requieren enteros con propiedades adicionales; en particular enteros que tengan exactamente dos factores primos.

Definición 4.1.21 *Un entero Blum es un número compuesto, de la forma* $n = pq$, *donde* $p \equiv 3 \pmod 4$ *y* $q \equiv 3 \pmod 4$, ◇

estos enteros fueron nombrados en [Blum82].

Definición 4.1.22 *Un entero Blum Generalizado es un número compuesto, de la forma* $n = p^s q^t$, *donde* $p \equiv 3 \pmod 4$, $q \equiv 3 \pmod 4$, s *y* t *son enteros impares.* ◇

Si $k \in \mathbb{N}$, sea $B_k := \{p \cdot q \mid p, q \text{ son primos y } \lfloor log_2(p) \rfloor = \lfloor log_2(q) \rfloor = \frac{k-1}{2} \text{ y } p \equiv 3 \pmod 4 \text{ y } q \equiv 3 \pmod 4\}$.

Símbolo de Jacobi y raíces cuadradas

Si n es un entero Blum generalizado, el símbolo de Jacobi para cualquier $x \pmod n$ es: $(\frac{x}{n}) = (\frac{x}{p})^s (\frac{x}{q})^t = (\frac{x}{p})(\frac{x}{q})$.

Así, $(\frac{-1}{p}) = (\frac{-1}{q}) = -1$, por que $p \equiv q \equiv 3 \pmod 4$. Esto implica los siguientes resultados:

1. $(\frac{-1}{n}) = +1$ pero -1 no es un residuo cuadrático, es decir $-1 \in \mathbb{Z}_n^* \setminus QR_n$.

2. Las dos raíces $\pm w_p$ de un residuo cuadrático módulo p tienen símbolo de Legendre distinto, de manera análoga para q. Por tanto las cuatro raíces, de un resiudo cuadrático módulo n toman los siguientes cuatro valores, donde la entrada de cada par, es el símbolo de Legendre: (1,1), (1,-1), (-1,1) y (-1,-1). Por tanto dos de estas raíces tienen símbolo de Jacobi +1; pero solamente una de ellas es residuo cuadrático.

La importancia contenida en los enteros Blum generalizados es el hecho de que hay un plan de demostración de cero conocimiento eficiente, es decir, se puede probar que un entero n es un entero Blum generalizado sin mostrar los factores de n [GrPe88], mientras que no se sabe si para los enteros Blum haya algo parecido.

4.1.5 Enteros Williams

Una clase especial de números, son los enteros Williams, [Wi80].

Definición 4.1.23 *Los números enteros Williams son denotados por:*

$$will_n := \{n = pq \mid p, q \text{ son primos y } p \equiv 3 \pmod 8 \text{ y } q \equiv 7 \pmod 8\}$$

◇

La propiedad especial que tiene los enteros Williams (como en [Wi80], [GMR]) es que 2 tiene símbolo de Jacobi -1 módulo cualquiera de ellos. Esto se sigue de que $(\frac{2}{n}) = (-1)^{(n^2-1)}/8$.

Si n es un entero Williams generalizado, entonces:

$$n = pq = 3 \cdot 7 = 5 \pmod 8 \Rightarrow n^2 \equiv 25 \pmod{16}$$

entonces $(n^2 - 1)/8$ es impar.

4.1.6 Densidad de primos

La mayoría de los planes criptográficos, necesitan números primos grandes. Por tanto es importante saber cuantos números primos hay de cierto tamaño.

El teorema básico en este tema es el teorema de los números primos.

Así, la cantidad de primos entre 1 y n esta aproximada como sigue:

$$\Pi(n) \sim \frac{n}{ln(n)}.$$

Donde ln denota el logaritmo natural y $\sim$ quiere decir que el cociente de $\Pi(n)$ y $\frac{n}{ln(n)}$ tiende a 1 cuando n tiende a ∞. (Véase [HaWr79, capítulo 1]).

Algunas veces, se necesitan números en cierta clase de congruencia, por ejemplo, primos p, tal que $p \equiv 3 \pmod 4$, para los enteros Blum. Para estos casos el teorema de los números primos de Dirichlet dice, que de cierta manera estos primos están distribuidos igualmente sobre las posibles clases de congruencia: Dado cualquier módulo v, entonces hay regularmente muchos primos congruentes $\gamma \ (mod\ v)\ \forall \gamma \in \mathbb{Z}_v^*$. Si $\Pi_{v,\gamma}(n)$ denota el número de primos en el conjunto $\{0, 1, \cdots n\}$, que son congruentes con $\gamma \ (mod\ v)$, entonces el teorema dice que:

$$\Pi_{v,\gamma}(n) = \frac{1}{\phi(v)} \frac{n}{ln(n)},$$

Donde $\phi(v)$ es la función de Euler, es decir, determinta la cantidad de primos relativos con v, que son menores que v. (Véase [Kran86]).

El teorema de los números primos considera todos los números primos mayores a cierto límite. Si se quieren números primos grandes, por ejemplo, primos con una longitud binaria, k. La siguiente relación es muy importante,

Para toda $n \in \mathbb{N}$

$$\Pi_{v,\gamma}(2n) - \Pi_{v,\gamma}(n) \sim \frac{1}{\phi(n)} \frac{1}{ln(v)}$$

La generación de los primos es una parte a operar en los planes que se presentarán en los siguientes capítulos, por lo que deben tener casos concretos para un k dado, por ejemplo $k = 512$.

En [RoSc62] hay una forma de demostrar la siguiente relación. Para toda $n \geq 21$,

$$\Pi(2n) - \Pi(n) = \frac{3}{5}\frac{n}{ln(n)}.$$

4.2 Preimágenes-conjuntos grandes

En esta sección se presentan funciones, donde cada imagen tiene una cantidad grande de preimágenes. Esta propiedad es llamada **propiedad fibrada** (ver fig. 4.1.1). Se dice que la función tiene **grado fibrado** d, si cada imagen tiene al menos d preimágenes. Las funciones que tienen propiedad fibrada, se pueden aplicar en los propósitos criptológicos, si en éstas es difícil encontrar colisiones.

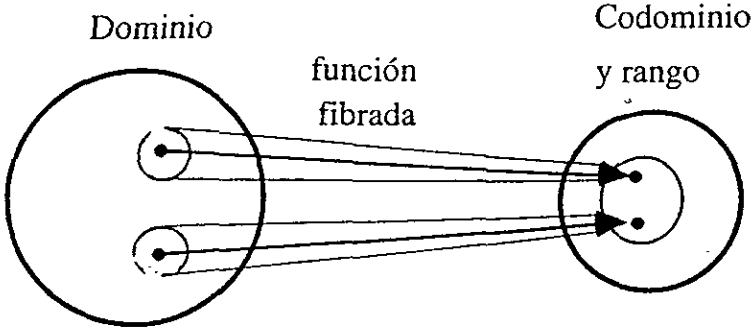


Figura 4.1.1. Función Fibrada. Los círculos pequenos son los conjuntos de preimágenes; la propiedad fibrada garantiza que estos son al menos de tamano d. El rango no debe ser necesariamente todo el codominio.

4.2.1 Caso del logaritmo discreto: Exponentes de vectores

En el caso del logaritmo discreto una función con propiedad fibrada es simplemente una función formada por el producto de varios enteros que están elevados a un exponente. A esta función se le conoce como **exponencial-vector**, y si este vector es de tamaño u, la función es llamada **exponencial u-vector**.

En Grupos Generales

Definición 4.2.1 *Sea H un grupo Abeliano de orden q, (donde q no es necesariamente primo) y $u \in \mathbb{N}$.*

a) Para cualquier vector $\bar{g} = (g_1, \cdots, g_u) \in H^u$ y $\bar{X} = (x_1, \cdots, x_u) \in \mathbb{Z}^u$, sea

$$\bar{g}^{\bar{X}} := g_1^{x_1} g_2^{x_2} \cdots g_u^{x_u} \in H.$$

Nótese que el vector exponente también puede ser denotado por: $\bar{X} = (x_1, \cdots, x_u) \in \mathbb{Z}_q^u$.

b) Para cualquier u y $\bar{g}$, la función exponencial u-vector con base $\bar{g}$ de $\mathbb{Z}_q^u$ a H, denotada por $exp_{\bar{g}}$,

$$exp_{\bar{g}} := \bar{g}^{\bar{X}}$$

c) Para $\bar{X} = (x_1, \cdots, x_u)$, $\bar{Y} = (y_1, \cdots, y_u) \in \mathbb{Z}_q^u$, denotamos el producto interno usual como $\bar{X}\bar{Y} = (x_1 y_1 + \cdots + x_u y_u)$.

d) Un vector con todos las entradas cero es denotado por $\bar{0}$.

$\diamondsuit$

Lema 4.2.1

a) Para cualquier $u \in \mathbb{N}$ y cualquier vector $\bar{g} \in H^u$ de longitud u, es decir, un u-vector, la función $exp_{\bar{g}}$, es un homomorfismo:

$$exp_{\bar{g}}(\bar{X} + \bar{Y}) = \bar{g}^{(\bar{X}+\bar{Y})} = \bar{g}^{\bar{X}} \bar{g}^{\bar{Y}} = exp_{\bar{g}}(\bar{X}) exp_{\bar{g}}(\bar{Y})$$

b) Si H es cíclico, una función exponencial u-vector es un producto interno seguido de una exponencial normal: si $\bar{g} = (g_1, \cdots, g_u)$ y g es un generador de H, cada g_i puede ser escrito como $g_i = g^{c_i}$, y

$$exp_{\bar{g}}(\bar{X}) = g_1^{e_1 x_1} \cdots g_u^{c_u x_u} = g^{\bar{e}\bar{X}}.$$

$\diamondsuit$

Caso en grupos de orden primo

El caso mas importante es cuando el orden q, de un grupo es primo, por lo que $\mathbb{Z}_q$ es un campo y como su grupo multiplicativo es cíclico, una ecuación exponencial u-vector, corresponde a una ecuación lineal de los exponentes (por el lema 4.2.1). Por tanto se puede determinar el número de soluciones de esta ecuación.

Lema 4.2.2 *Sea H un grupo de orden primo q, y $\bar{g} = (g_1, g_2)$ un par de elementos de H que son generadores. Entonces para cada $x_1 \in \mathbb{Z}_q$, la función $exp_{\bar{X}}(x_1, \cdot) : \mathbb{Z}_q \to H$ es biyectiva. En otras palabras, para cada $z \in H$ y cada x_1, hay exactamente un $x_2 \in \mathbb{Z}_q$ tal que $exp_{\bar{g}}(x_1, x_2) = z$.* ◇

Demostración Usando el lema 4.2.1 y tomando a g_1 como generador de H: cualquier $z \in H$, g_2 y z pueden ser representados como, $g_2 = g_1^{e_2}$ y $z = g_1^{e^*}$. Entonces

$$\bar{g}^{\bar{x}} = z \Longleftrightarrow g_1^{x_1 + e_2 x_2} = g_1^{e^*} \Longleftrightarrow x_1 + e_2 x_2 \equiv e^* \pmod{q}$$

Como g_1 es un generador $e_2 \neq 0$. Por tanto esta ecuación tiene una única solución para x_2. ∎

Corolario 4.2.1 Grado Fibrado. *Si H es un grupo de orden primo q y $\bar{g}$ representa un par de generadores de H. Cada imagen z de la función exponencial 2-vector $exp_{\bar{g}}$, tiene exactamente q preimágenes. Es decir $exp_{\bar{g}}$ es de grado fibrado q.*

◇

La demostración puede verse en el teorema 5.2.2b

∎

Otra consecuencia del lema 4.2.2 es que el resultado z no da información (Shannon) [Sh], acerca del primer parámetro x_1, de $exp_{\bar{g}}(x_1, x_2)$, si x_2 es elegido aleatoriamente de manera uniforme. A esto se le llama **propiedad escondida**, porque el resultado z esconde a x_1 prefectamente.

4.2.2 Parejas de permutaciones

Asúmase que se tienen un par de permutaciones (f_0, f_1) con dominio común D. En base a este par se construirán las funciones B y B_σ (para $\sigma \in \mathbb{N}$) con propiedades fibradas. La construcción es debido a [GMR], pero ahí, solamente fue usada por lo difícil que es encontrar colisiones, es decir, no se mostró la propiedad fibrada.

La idea de la construcción es la siguiente: Sean f_0 y f_1 dos funciones tales que cada elemento $z \in D$ tiene una preimágen por cada una de ellas; esto es, se tienen dos preimágenes por cada $z \in D$. Esto se puede interpretar como una única función en la que cada z tiene dos preimagenes, al declarar al índice 0 ó 1 como un parámetro adicional. Una extensión de esta función para obtener muchas preimágenes para cada z, se hace al hacer aplicaciones iteradas, hasta obtener el número de preimágenes deseado.

La aplicación sucesiva de esta función, es realmente una nueva función, a esta función se le llamará B y si el número de iteraciones esta determinado por una cantidad σ, la función se llamará B_σ. Lo anterior se puede apreciar en la siguiente definición.

Definición 4.2.2 Función Iterada o Permutaciones. *Si f_0 y f_1 son funciones sobre un dominio común D. Sea B una función definida como:*

$$B : \{0,1\}^* \times D \longrightarrow D$$
$$B(b_1, b_2, \ldots b_l, y) := f_{b_0}(f_{b_1}(\cdots f_{b_l}(y)\cdots))$$

donde $b_1, b_2, \cdots b_l$ son bits individuales de una cadena de bits, los cuales forman el primer parámetro. (ver figura 4.1.2). La restricción de B a una cadena de longitud fija σ, como primer parámetro es denotada por B_σ, esto es,

$$B_\sigma : \{0,1\}^\sigma \times D \longrightarrow D,$$

con $B_\sigma(b, y) := B(b, y)$. Similarmente, si las funciones tienen un índice k (usualmente alguna llave), es decir, si las funciones originales estan denotadas por; $f_{0,k}$ y $f_{1,k}$, las funciones resultantes serán denotadas por,

B_k y $B_{\sigma,k}$

◊

Por ejemplo, $B(1011, y) = f_1(f_0(f_1(f_1(y))))$. En la figura 4.1.2, se representa una aplicación de f_0 y es como dirigir la flecha hacia arriba a la derecha, y una aplicación de f_1 es como dirigir la flecha hacia arriba a la izquierda. Es decir, una aplicación de B es una secuencia de flechas, que comienzan en y y los bits b_i, son usados para formar la dirección de las flechas.

Lema 4.2.3 *Si f_0 y f_1 son permutaciones con dominio D, las funciones iteradas B y B_σ tienen las siguientes propiedades:*

a) La restricción $B(b, \cdot)$ es una permutación sobre D para cada $b \in \{0,1\}^*$, en otras palabras, para cualquier $z \in D$ y cualquier cadena b, de bits, existe exactamente un y sólo un $y \in D$, tal que $B(b, y) = z$.

b) Para toda $\sigma \in \mathbb{N}$, la función B_σ es de grado 2^σ. Esto es, cada $z \in D$ tiene exactamente 2^σ preimagenes.

◊

Demostración

a) La función $B(b, \cdot)$ es una composición de permutaciones y por tanto también es una permutación.

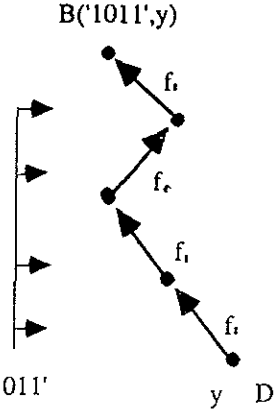


Figura 4.1.2. Ejemplo de la función B.

b) De acuerdo con el inciso a), para cada cadena b de longitud σ, hay exactamente un valor y_b, con $B_\sigma(b, y_b) = z$. Como hay 2^σ cadenas de longitud σ, por tanto hay 2^σ parejas (b, y_b) que son preimágenes de z sobre B_σ. (Nótese que los valores y_b no necesitan ser diferentes).

■

La figura 4.1.3 muestra un conjunto de preimágenes de la función B_3.

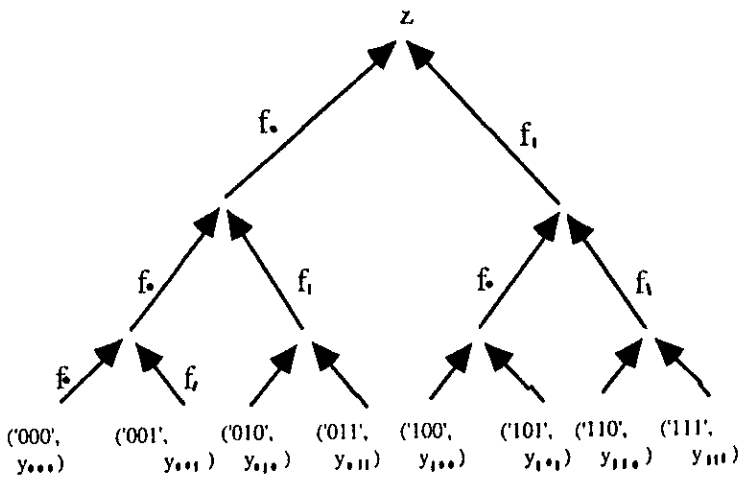


Figura 4.1.3. Propiedades de B_σ.
La figura muestra las 8 preimágenes de (b, y_σ) de un valor z bajo la función B_σ para el caso de b = 3.

La parte a) de este lema corresponde al lema 4.2.2 para el caso del logaritmo discreto y tiene las mismas consecuencias. El resultado z no da información (Shannon) acerca del primer parámetro, b, si el segundo es decir y, es escogido aleatoriamente de manera uniforme de D. Esta es de nuevo una **propiedad escondida** otra vez.

4.2.3 Caso para la factorización: (Construcción del homomorfismo)

Todas las funciones presentadas en esta sección están basadas sobre la función cuadrática módulo n. Las cuales son una permutación

sobre QR_n, donde n es un entero Blum generalizado[1].

Antes de comenzar con estas definiciones es importante tener presente los siguientes puntos:

- A fin de hacer más eficiente las futuras aplicaciones de las permutaciones, es necesario que éstas sean definidas sobre conjuntos, en donde sus miembros se pueden probar eficientemente que son residuos cuadráticos. Por lo que el conjunto QR_n es reemplazado por otro conjunto relacionado a éste, que es llamado RQR_n. (Los aspectos computacionales referente a estos conjuntos serán presentados posteriormente).

- Es importante decir que algunas de las propiedades también se cumplirán si n no es un entero Blum generalizado. Por lo que algunas veces se puede omitir la demostración de cero-conocimiento y usar un algoritmo verificador localmente que sea más eficiente. Por tanto, algunas de las definiciones, son presentadas para enteros n, mas generales.

Dominios y Parejas de Funciones

Lema 4.2.4 *a) Para cualquier* $n \in \mathbb{N}$, $n = 4m+1$, $m \in IN$, *los conjuntos* $\{\pm 1\}$ *y* $\pm QR_n$, *son subgrupos de* $\mathbb{Z}_n^*$. ($\{\pm 1\} := \{1, -1\}$ *y* $\pm QR_n := QR_n \cup -QR_n$).

b) Si n *es un entero Blum generalizado, entonces* $\pm QR_n = \mathbb{Z}_n^*(+1)$. ◇

Demostración

a) (Notese que se toma una representación simétrica de $\mathbb{Z}_n$, es decir $\mathbb{Z}_n = \{-\frac{(n-1)}{2}, \cdots, -1, 0, 1, \cdots \frac{(n-1)}{2}\}$). La demostración es inmediata.

b) Recuerdese que $-1 \in \mathbb{Z}_n^*$, $-1 \in \mathbb{Z}_n^* \setminus QR_n$ (Sección 4.1.4). Entonces $\pm QR_n \subseteq \mathbb{Z}_n^*(+1)$. Ahora como $|\pm QR_n| = 2|QR_n|$. Considerando la sección 4.1.3 se tiene que $|QR_n| = 1/4 \mid \mathbb{Z}_n^*|$

[1] Vid la sección 4.1.4.

y $|\ \mathbb{Z}_n^*(+1)| = 1/2|QR_n|$ para los enteros Blum generalizados. Por tanto $|\ \mathbb{Z}_n^*(+1)| = |\pm QR_n|$.

■

Definición 4.2.3 *Reemplazo de Residuos Cuadráticos.*

a) Para cualquier entero Blum generalizado n, se reemplazará el grupo de residuos cuadráticos módulo n, por el grupo factor o grupo cociente:

$$RQR_n := \mathbb{Z}_n^*(+1)/\{\pm 1\} = \pm QR_n/\{\pm 1\}$$

b) Para cualquier entero $n \in \mathbb{N}$, $n = 4m + 1, m \in \mathbb{N}$, se puede distinguir dos generalizaciones de RQR_n:

$$RQR_n^{<} := \pm QR_n/\{\pm 1\},$$
$$RQR_n^{>} := \mathbb{Z}_n^*(+1)/\{\pm 1\}$$

c) Para el propósito computacional, cada elemento de estos grupos factores, es decir el conjunto $\{\pm y\} = \{y, -y\} := \{y, n - y\}$ con $1 \leq y \leq n$, es representado por el miembro más pequeño que está caracterizado por la condición, $y \leq n/2$. Esto produce lo siguiente:

$$RQR_n^{<} = \{y \in \mathbb{Z} \mid 1 \leq y \leq n/2 \ y \ (y \in QR_n \ \acute{o} \ n - y \in QR_n)\}$$
$$RQR_n^{>} = \{y \in \mathbb{Z} \mid 1 \leq y \leq n/2 \ y \ y \in \mathbb{Z}_n^*(+1)\}$$

d) Los mapeos canónicos, (es decir, cada elemento será mapeado a su valor absoluto) de $\pm QR_n$ a $RQR_n^{<}$ y de $\mathbb{Z}_n^*(+1)$ a $RQR_n^{>}$ en esta representación serán denotados por $|\bullet|$, ya que estos son el valor absoluto usual sobre $\mathbb{Z}$ si $\mathbb{Z}_n$ tiene una representación simétrica.

e) La multiplicación en los grupos factores $RQR_n^{<}$ y $RQR_n^{>}$ es denotada por $\cdot$. Por lo que se escribirá $y \cdot y' = |yy'|$.

$\diamond$

Nótese que:

$$RQR_n^{<} \subseteq RQR_n^{>}$$

y si n es un entero Blum generalizado,

$$RQR_n^{<} = RQR_n^{>} = RQR_n$$

.

Ejemplo 4.2.3.1 *Tomemos* $n = pq = (3)(7) = 21$, *un entero Blum, donde* $\mathbb{Z}_n^* = \{1,2,4,5,8,10,11,13,16,17,19,20\}$ *y su subgrupo de residuos cuadráticos es,*

$$QR_n = \{1,4,16\} \text{ y } -QR_n = \{-1,-4,-16\} = \{20,17,5\}.$$

Por tanto,

$RQR_n = \pm QR_n\{\pm 1\} =$
$\{\{\pm 1\} \cdot 1, \{\pm 1\} \cdot 4, \{\pm 1\} \cdot 16, \{\pm 1\} \cdot 20, \{\pm 1\} \cdot 17, \{\pm 1\} \cdot 5\}.$

La principal razón de representar a QR_n *como* RQR_n, *se debe a razones aritméticas y en especial, es más fácil probar si un número en* RQR_n *es o no residuo cuadrático que en* QR_m. *En realidad se sustituirán todos aquellos números que sean mayores que* $n/2$.

Así en lugar de trabajar directamente con el conjunto $\{1,4,16\}$ *se trabajará con el conjunto* $\{\{\pm 1\} \cdot 1, \{\pm 1\} \cdot 4, \{\pm 1\} \cdot 5\}$, *porque probar que 16 sea un residuo cuadrático módulo n es más difícil que probar que 5 ó -5 sea un residuo cuadrático. Nótese que todo ésto lo estamos pensando en números grandes.*

Definición 4.2.4 función Par. *(Esta definición es adaptada de [GMR].)*

Para cualquier $n \in \mathbb{N}, n = 4m + 1$, $m \in \mathbb{N}$, *las funciones*

$$f_{0,n}, f_{1,n} : RQR_n^{>} \longrightarrow RQR_n^{<}.$$

estarán definidas por,

$$f_{0,n}(x) := x^2,$$
$$f_{1,n}(x) := 4x^2.$$

El índice n, puede ser omitido si esto es claro de contexto. Note que el rango es a lo mas $RQR_n^<$, porque los mapeos cuadrados están dentro de $RQR_n^<$ y $4 = 2^2 \in RQR_n^<$. ◇

De acuerdo a las convenciones de la definición 4.2.2, se definirán las siguientes funciones:

$$B_n : \{0,1\}^* \times RQR_n^> \longrightarrow RQR_n^<$$
$$B_n(b_1, \ldots, b_l, y) = f_{b_1}(f_{b_2}(\cdots(f_{b_l}(y))\cdots)).$$

y la restricción para una cadena de longitud σ, es $B_{\sigma,n}$. La restricción de B_n a $\{0,1\}^* \times RQR_n^<$ será denotada por $B_n^<$.

Propiedades Fuertes de B_n para Enteros Blum Generalizados

Lema 4.2.5 Permutaciones. *(Adaptado de [GMR]).*

a) Si n es un entero Blum generalizado, $f_{0,n}$ y $f_{1,n}$ son permutaciones sobre RQR_n.

b) Si n es un entero Blum generalizado, la restricción $B_n(b,\cdot)$ es una permutación sobre RQR_n para cada $b \in \{0,1\}^*$. En otras palabras, para cualquier imagen $z \in RQR_n$ y cualquier cadena b hay exactamente un único $y \in RQR_n$, tal que $B_n(b,y) = z$.

◇

Demostración

a) El mapeo canónico induce a un homomorfismo de QR_n a RQR_n. Esto es, QR_n es el conjunto de representaciones en RQR_n y las multiplicaciones en RQR_n están representadas por las multiplicaciones en QR_n. Así todas las propiedades de las funciones cuadráticas son inducidas en RQR_n.

En particular dada la permutación $f_{0,n}$. Si se toma a $f_{1,n}$ como una composición de la permutación $f_{0,n}$, y un multiplicación por 4. Una multiplicación por 4 también es una permutación, porque $4 \in RQR_n$.

b) En base a la parte a) la demostración es exactamente el lema 4.2.3a.

■

La parte b) de este lema corresponde a el lema 4.2.2 para el caso del logaritmo discreto, y también se tienen las mismas dos consecuencias.

- a) La restricción de $B_{\sigma,n}$ tiene una propiedad fibrada (vea el lema 4.2.3 y la figura 4.1.3).
- b) El resultado z no da información (Shannon) acerca del primer parámetro b, si el segundo parámetro es elegido de manera uniforme de RQR_n. Esta es otra vez la **propiedad escondida.**

Representación de B_n

La idea de la siguiente representación para B_n es sencilla: En cada iteración el parámetro original y está elevado al cuadrado, y algunas veces se multiplica un 4 a éste. Por tanto, si la longitud de b, es l, se tiene como resultado $4^{\beta} \cdot y^{2^l}$ para alguna β.

Lema 4.2.6 *Para cada cadena* $b = b_1, b_2, \ldots b_l$, *sea* $num(b) := b_1 + 2b_2 + \cdots + 2^{l-1}b$, *es decir, b es interpretado como un número binario escrito hacia atraz. Entonces,*

$$B_n(b, y) = 4^{num(b)} \cdot y^{2^l}$$

$\forall n$ *de la forma* $4n + 1$, $n \in \mathbb{N}$ ◇

Demostración La demostración se hará por inducción sobre la longitud de l.

Para $l = 1$, $B_n(b, y) = f_{b_1}(y)$, donde $f_0(y) = y^2 = 4^0 \cdot y^{2^1}$ y $f_1(y) = 4^1 \cdot y^{2^1}$, como se es requerido.

Supóngase que el lema se cumple para $l - 1$. Y en base a esto se demostrará que también se cumple para l.

Sea $b' = b_1, b_2, \ldots b_{l-1}$, así $num(b) := num(b') + 2^{l-1}b_l$. Por tanto,

$$\begin{aligned} B_n &= f_{b_1}(f_{b_2}(\cdots f_{b_{l-1}}(f_{b_l}(y))\cdots)) \\ &= B_n(b', f_{b_l}(y)) \\ &= B_n(b', 4^{b_l} \cdot y^2) \\ &= 4^{num(b')} \cdot (4^{b_l} \cdot y^2)^{2^{l-1}} \\ &= 4^{num(b')+2^{l-1}b_l} \cdot y^{2\cdot 2^{l-1}} \\ &= 4^{num(b)} \cdot y^{2^l}. \end{aligned}$$

■

Al considerar la restricción $B_{\sigma,n}$ para la cadena b de longitud σ, los números $num(b)$ estan entre 0 y $2^\sigma - 1$. En seguida se definirá una variante de $B_{\sigma,n}$ que depende directamente de los números.

Definición 4.2.5 *Para todo n de la forma $4n+1$, $n \in \mathbb{N}$ y $\sigma \in \mathbb{N}$, sea:*

$$B^*_{\sigma,n} : \{0, \ldots 2^\sigma - 1\} \times RQR^>_n \longrightarrow RQR^<_n$$

definida por:

$$B_{s,n}(a, y) = 4^a \cdot y^{2^s}$$

La restricción de $B^*_{\sigma,n}$ a $\{0, \ldots 2^\sigma - 1\} \times RQR^<_n$ se denotará por $B^{*<}_{\sigma,n}$.

◇

En el lema 4.2.6 se puede escribir a $B_n(b, y) = B^*_{\sigma,n}(num(b), y)$, si la longitud de b es σ.

La función $B^*_{\sigma,n}$ como homomorfismo

Para ver que $B^*_{\sigma,n}$ es un homomorfismo, se derivará la operación del grupo del dominio. Esto fue hecho por [Bleu90].

Notación: Si $a, b \in \mathbb{Z}$, denotaremos al cociente de a entre b, como $a \; div \; b$.

Definición 4.2.6 Operación del Grupo. *Para toda n de la forma $4m+1$, $m \in$ IN y $\sigma \in$ IN, el dominio de $B^*_{\sigma,n}$ es representado como:*

$$G_{\sigma,n} := \{0, 1, \cdots, 2^\sigma - 1\} \times RQR_n^>,$$

y la operación $*$, sobre $G_{\sigma,n}$ es definida por:

$$(a, y) * (b, y') := ((a + a') \pmod{2^\sigma},\ y \cdot y' \cdot 4^{(a+a') \; div 2^\sigma}).$$

◇

Teorema 4.2.2 Grupos y Homomorfismos.
Sea n de la forma $4m+1$, $m \in$ IN y $\sigma \in$ IN.

a) El conjunto $G_{\sigma,n}$ es un grupo Abeliano con la operación binaria $*$; su elemento neutro es $(0, 1)$.

b) La función $B^*_{\sigma,n}$ es un homomorfismo de $G_{\sigma,n}$ a $RQR_n^>$.

◇

Demostración

a) La demostración de que $G_{\sigma,n} := \{0, 1, \cdots, 2^\sigma - 1\} \times RQR_n^>$ es un grupo Abeliano con la operación

$$(a, y) * (b, y') := ((a + a') \pmod{2^\sigma}, y \cdot y' \cdot 4(a + a') \; div \; 2^\sigma,$$

donde $RQR_n^> = \mathbb{Z}_n^*(+1)/\{\pm 1\}$, se hará al probar que realmente se cumplen las propiedades de grupo. Es decir probar que:

i) Para $(a,y),(a',y') \in G_{\sigma,n} \Longrightarrow (a,y)*(a',y') \in G_{\sigma,n}$. Donde,

$$(a,y)*(a',y')= \\ ((a+a') \pmod{2^\sigma},\ y\cdot y'\cdot 4^{(a+a')\ div\ 2^\sigma},$$

es claro que $(a+a') \pmod{2^\sigma} \in \{0,\ldots 2^\sigma-1\}$. Así, solo resta mostrar que $y\cdot y' 4^{(a+a')\ div\ 2^\sigma} \in RQR_n^{\geq}$.

El exponente de 4 es 0 ó 1. Como $y,y',4 \in RQR_n$, por el Teorema 4.1.5 $y\cdot y'\cdot 4^0$ y $y\cdot y'\cdot 4^1 \in RQR_n^{\geq}$. Por tanto, se cumple la propiedad de cerradura.

ii) Si $(a,x),(b,y)\ y\ (c,z) \in G_{\sigma,n} \Longrightarrow ((a,x)*(b,y))*(c,z) = (a,x)*((b,y)*(c,z))$.
$((a,x)*(b,y))*(c,z) =$

$$\begin{aligned}
&= ((a+b) \pmod{2^\sigma},\ x\cdot y\cdot 4^{(a+b)\ div\ 2^\sigma})*(c,z) \\
&= (((a+b)+c) \pmod{2^\sigma},\ (x\cdot y)\cdot z\cdot 4^{((a+b)+c)\ div\ 2^\sigma}) \\
&= ((a+(b+c)) \pmod{2^\sigma},\ x\cdot(y\cdot z)\cdot 4^{(a+b+c)\ div\ 2^\sigma}) \\
&= (a,x)*((b+c) \pmod{2^\sigma},\ y\cdot z\cdot 4^{(b+c)\ div\ 2^\sigma}) \\
&= (a,x)*((b,y)*(c,z))
\end{aligned}$$

iii) Sea $(a,x) \in G_{\sigma,n}$, Y sea $(0,1) \in G_{\sigma,n}$, entonces. $(a,x)*(0,1) = ((a+0) \pmod{2^\sigma},\ x\cdot 1\cdot 4^{(a+0)\ div\ 2^\sigma}) = (a \pmod{2^\sigma}, x\cdot 4^0) = (a,x)$.

iv) Sea $(a,x) \in G_{\sigma,n}$ y sea $a' = 2^\sigma - a$ y $y' = (4y)^{-1}$, Así, $(a,y)*(a',y') = ((a+a') \pmod{2^\sigma}, y\cdot y'\cdot 4^{(a+a')\ div\ 2^\sigma})$ $= ((a+2^\sigma-a) \pmod{2^\sigma}, y\cdot(4y)^{-1}\cdot 4^{(a+2^\sigma-a)\ div\ 2^\sigma}) =$ $(2^\sigma \pmod{2^\sigma}, 4^{-1}\cdot 4^{2^\sigma\ div\ 2^\sigma}) = (0, 4^{-1}4^1) = (0,1)$.

Por tanto $G_{\sigma,n}$ es un grupo Abeliano.

b)

$$\begin{aligned}
|h((a,x)*(b,y))| &= |h(a+b \pmod{2^\tau}, x\cdot y\cdot 4^{(a+b)div2^\tau}|) \\
&= |4^{a+b \pmod{2^\tau}}\cdot(x\cdot y\cdot 4^{(a+b)div2^\tau})^{2^\tau}| \\
&= |4^{a+b \pmod{2^\tau}+\ 2^\tau\cdot((a+b)div2^\tau)}\cdot(x\cdot y)^{2^\tau}|
\end{aligned}$$

$$
\begin{aligned}
&= |4^{a+b} \cdot (x \cdot y)^{2^r}| \\
&= . \mid |4^a \cdot x^{2^r}| \cdot |4^b \cdot y^{2^r}| \mid \\
&= \mid h(a,x) \cdot h(b,y) \mid
\end{aligned}
$$

■

Tamaño del conjunto de preimágenes de $B^*_{\sigma,n}$ para n, en general

Como $B^*_{\sigma,n}$ es un homomorfismo, en base a la sección 4.1.2 se puede deducir fácilmente el *número de preimágenes* para cualquier elemento que tiene alguna preimagen. Sólo que ahora el dominio de $B^*_{\sigma,n}$ es mucho más grande que el codominio, por lo que el conjunto de preimágenes es en promedio mucho más grande. Como todo el conjunto de preimágenes de un homomorfismo es igual para cada imagen, en particular cada conjunto es grande.

Lema 4.2.7 *a) Para toda n de la forma $4m+1$, $m \in \mathbb{N}$ y $\sigma \in \mathbb{N}$, cada elemento z en el rango de $B^*_{\sigma,n}$ tiene al menos 2^σ preimágenes, es decir,*

$$z \in B^*_{\sigma,n}(G_{\sigma,n}) \Longrightarrow |B^{*-1}_{\sigma,n}| \geq 2^\sigma .$$

Por lo tanto el grado del homomorfismo fibrado es 2^σ.

b) El inciso a) también se cumple para $B^{*<}_{\sigma,n}$.

◇

Demostración

a) $|B^{*-1}_{\sigma,n}| = |kernel(B^*_{\sigma,n})| = |G_{\sigma,n}|/|B^*_{\sigma,n}(G_{\sigma,n})| \geq$
$2^\sigma |RQR^>_n|/|RQR^<_n| \geq 2^\sigma .$

b) $|B^{*<-1}_{\sigma,n}| = |kernel(B^{*<}_{\sigma,n})| = |G^<_{\sigma,n}|/|B^{*<}_{\sigma,n}(G^<_{\sigma,n})| \geq$
$2^\sigma |RQR^<_n|/|RQR^<_n| = 2^\sigma .$

■

4.3 Algunos algoritmos eficientes

Para poder visualizar que los algoritmos de los planes que se presentan en los siguientes capítulos, se necesita conocer algunos resultados de la teoría de números computacional.

El libro estandar que se lleva en estos campos es [Knut81], pero existe una gran cantidad de literatura, en particular sobre trucos especiales para los planes criptológicos. Aquí, solamente se mencionan algoritmos secuenciales. En esta sección no se menciona una bibliografía muy extensa. Si se quiere saber las descripciones precisas y comparaciones de algoritmos grandes y rápidos, que están en las implementaciones de softwere, puede consultar [Fox91].

Multiplicación Modular

Las operaciones básicas que se utilizan en la mayoría de los siguientes planes, son las multiplicaciones modulares. Se asume que $\mathbb{Z}_n$ es representado por los números $\{0, 1, \ldots n-1\}$. Una multiplicación modular puede ser calculada por una multiplicación y una división. Una multiplicación de un número de longitud l-bits sobre una máquina de longitud de palabra w, puede ser fácilmente ejecutado con $\lceil l/w \rceil^2$ multiplicaciones de palabras. La división toma un número de multiplicaciones de palabras un poco mayor.

Hay manera de hacer multiplicaciones asintóticamente más rápido. Sin embargo, en general se tiene la confianza de que el algoritmo más rápido de [ScSt71] sól vale la pena para números grandes como los que se usan en criptología.

El algoritmo intermedio de Karatsuba [KaOf], tiene su punto óptimo mas o menos de un tamaño típico, el cual depende de la implementación específica, por ejemplo véase [Guin91]. Por tanto, decir que una multiplicación se realiza en $O(l^2)$, es una estimación conservadora tanto en la teoría como en la práctica.

También con la división se tiene lo mismo, y se puede decir que se ejecuta en $O(l^2)$, no está muy desviado de las implementaciones prácticas, aunque se puede ser más rápidos asintóticamente: La única alternativa para la división estándar es el método de Mont-

gomery, donde la representación de la clase de residuos por los números es cambiada, de tal manera que la multiplicación modular puede ser hecha sin la división [Mont85]. En su lugar hay una operación, llamada operación reducción. Esto tiene dos variantes en las implementaciones, que no pueden ser distinguidas fácilmente. La primera usa dos llamadas a la subrutina de la multiplicación. Para los números de tamaño típico, ésta no se usa si se tiene una buena implementación de la división; sin embargo esto es más rápido para números grandes, donde la multiplicación de Karatsuba no fue usada. La eficiencia de la segunda variante, la cual usa multiplicaciones de palabra directamente, está entre la multiplicación estandar y la división estandar.

Inversión, Teorema Chino del Residuo y Símbolo de Jacobi

Los inversos modulares pueden ser calculados eficientemente con el algoritmo Euclideano extendido. Sin embargo, este es un poco más lento que las multiplicaciones modulares y es asintóticamente $O(l^3)$.

El teorema chino del residuo se puede calcular eficientemente después de hacer un precálculo que puede aplicarse una única vez para todos los cálculos criptológicos del plan específico: para dos congruencias módulo números de longitud $l/2$, se necesita una multiplicación modular y una no modular con longitud $l/2$.

Consecuentemente se pueden llevar a cabo cálculos módulo un número compuesto, donde los factores de n son conocidos, es algunas veces ventajoso calcular los módulos para cada uno de los factores de manera separada y combinar los resultados con el algoritmo chino del residuo, hasta el final [QuCo82].

Como se mencionó, el símbolo de Jacobi se puede evaluar usando la ley de la reciprocidad cuadrática, lo cual produce un algoritmo similar al algoritmo Euclideano, con una complejidad asintótica $O(l^3)$.

Exponenciación

Los exponentes pueden ser calculados eficientemente en cualquier familia de grupos, en donde se tiene un algoritmo eficiente para la multiplicación debido a los algoritmos sobre la multiplicación y división. Serán necesarios l^* cuadrados y un promedio de $l^*/2$ multiplicaciones, para los exponentes con longitud $l^* - bits$.

Si el orden $|G|$, del grupo es conocido, los exponentes grandes deberán reducirse a módulos de tamaño $|G|$.

El número de multiplicaciones, puede ser reducido. Las técnicas para el caso que la base sea una variable, puede ser encontrado en [Knut81, BoCo90]; sin embargo ninguno de ellos logra menos de l^* cuadrados y l^* multiplicaciones. Si la base g, está fija y sólamente varían los exponentes, el tener algunas potencias de g precalculadas, puede ayudar a tener una reducción hasta $l^*/log_2(l^*)$ multiplicaciones, [BGMW93].

Para los exponenciales modulares con una base de tamaño de $l - bit$ y un exponente de un tamaño de $l^* - bits$, se tendrá una complejidad asintótica de $O(l^2 l^*)$. Es útil tener una subrutina para los cuadrados que sea más rápida que la multiplicación general; por lo que se puede reducir un factor de 2. Para una composición modular, en donde la factorización sea conocida, se debe aplicar el teorema chino del residuo, para tener una multiplicación más rápida para números mas pequeños y reducir el exponente.

Exponencial u-vector

El producto de varios exponentes, es decir un exponencial u-vector, $\bar{g}^{\bar{x}}$, como el de la definición 4.2.1, puede ser calculado más eficientemente que el hacer el cálculo de todos los exponentes $\bar{g}_i^{\bar{x}_i}$, separadamente y multiplicar los resultados. Para $u = 3$, se producen l^* cuadrados y un promedio de $7/8l^*$ multiplicaciones si los exponentes tienen longitud l^*. [EG84].

4.4 Suposiciones criptológicas

Los planes de firmas digitales intrínsecamente protegidos, sobre suposiciones criptológicas más comúnes son, la factorización y el logaritmo discreto.

4.4.1 Factorización de enteros

Suposición de la factorización

La primera suposición, es que la factorización de enteros grandes es difícil de llevarse a cabo. Para esto únicamente se necesitan los enteros Williams, es decir, los enteros que tienen exactamente 2 factores primos q y q; donde $p \equiv 3 \pmod 8$ y $q \equiv 7 \pmod 8$. La siguiente suposición fue usada en [GMR].

Definición 4.4.1 *a) Para todo $k \in \mathbb{N}$, sea*

$$Will_k = \{n = pq \in Will \mid |p|_2 = |q|_2 = k\}.$$

b) La suposición de la factorización, es que para todo algoritmo probabilístico con tiempo de ejecución polinomial $\tilde{F}$ (el cual trata de factorizar), para toda constante $c > 0$ y k suficientemente grande:

$$P(p \text{ sea un divisor no trivial de } n \mid n \in Will_k;\ p \leftarrow \tilde{F}(n)) < k^{-c}$$

$\diamondsuit$

Ultimos Adelantos

Los adelantos de los últimos años están resumidos en [LeLe90]. Desde entonces, fue inventado un algoritmo, el algoritmo de Criba en Campos de Números. La complejidad de los mejores algoritmos se puede abreviar como:

$$L_n[1/i, c] := (e^{ln(n)^{1/i} ln(ln(n))^{(i-1)/i}})^{c+O(1)}, \ c > 0 \text{ y } n, i \in \mathbb{N}.$$

Los mejores algoritmos propestos que han sido utilizados últimamente tienen un tiempo de ejecución esperado de $L_n[1/2, c]$, donde c es una constante pequeña (entre 1 y $\sqrt{4/3}$, considerando solamente algoritmos que han sido probados). El algoritmo criba en campos de números tienen un tiempo de ejecución de $L_n[1/3, c]$, donde c esta alrededor de 2.

En la páctica han sido factorizados números generales, es decir, números en los que no se saben sus propiedades, estos números tienen 130 dígitos decimales, esto es 430 bits. El algoritmo de criba en campos de números a lo mas a logrado factorizar números de 120 dígitos [DoLe95].

Si se quiere ver el reto de la computación cuántica puede consultar [Shor94].

Generación de enteros Williams uniformemente distribuidos

Un plan criptográfico necesita de un algoritmo específico gen_{will}, para generar enteros Williams, pero aquí la distribución de la salida no es uniforme. La suposición que realmente se necesita es que la factorización sea difícil, si los números son generados con gen_{will}. Además la mayor parte de los algoritmos que se usan en la páctica tienen un tiempo esperado de ejecución polinomial.

Nótese que se puede lograr una distribución uniforme, si se hace una elección repetidamente de los números p y q de longitud k, (en la clase de congruencia apropiada, aleatoriamente y con su prueba de primalidad), hasta que los primos puedan ser encontrados. De acuerdo a los teoremas de los números primos, la búsqueda termina después de un número esperado de pruebas polinomial [AdPR83, CoLe87], y únicamente los algoritmos completamente probados, que reconocen primos sin error en un tiempo esperado polinomial (es decir, un algoritmo ZPP en la notación de [BaDG88]), el cual es [AdHu87] junto con [Rabi80] menos prácticos. En una suposición con respecto a la densidad de los primos en un intervalo de tamaño medio se puede usar el algoritmo de [GoKi86]. (La suposición es más débil que la llamada conjetura de Crammer, la cual dice que

$\pi(n + \lceil ln(n)^2 \rceil) > \pi(n)$, donde $\pi(n)$ es el número aproximado de primos entre 1 y n, para n suficientemente grande.

En la práctica, normalmente se usan cantidades pequeñas de iteraciones de la prueba de Rabin-Miller [Rabi80], en donde se puede detener con un número compuesto con una probabilidad pequeña, (es decir, esto es un algoritmo co-R). Afortunadamente, la suposición de la factorización para la distribución resultante es una consecuencia de lo que se dijo unos renglones antes: Para n suficientemente grande el error de probabilidad que se tiene después de una sóla iteración de la prueba decrece más rápido que cualquier polinomio sobre k (ver [DaLP93]), y decrece exponencialmente en el número de iteraciones adicionales. Por tanto, si hay un algoritmo rápido que factorice todos los números con más de dos factores primos, se tendrá un impacto total sobre la probabilidad de factorizar todos los números elegidos. Por tanto el algoritmo usado en la prueba de Rabin-Miller, es una elección razonable para el algoritmo de la factorización, con tiempo polinomial gen_{will}, tal que la suposición de la factorización implica que para todos los algoritmos probabilísticos con tiempo polinomial, $\tilde{F}$, para $c > 0$ y k suficientemente grande:

$$P(p \text{ sea un divisor de } n \mid n \leftarrow gen_{will}(k);\ p \leftarrow \tilde{F}(n)) < k^{-c}.$$

Una alternativa para usar la prueba de Rabin-Miller son los algoritmos de [Muar95], que generan números que son primos, pero no se prueba su distribución.

4.4.2 Logaritmo discreto

La segunda suposición usada, es que los grupos H, de orden primo q, existen en donde el cálculo del logaritmo discreto es difícil de hacer.

Resumen de la suposición

El siguiente algoritmo sobre la suposición del logaritmo discreto, está presentada en un sentido no común, ya que se consideran diferentes grupos de interés. Una parte selecciona los grupos en donde

el cálculo del logaritmo discreto se supone que es difícil. La otra parte necesita estar segura de que ha sido generado el grupo completo y de orden primo, y además otras propiedades definidas en seguida.

Definición 4.4.2 *a) Una familia de grupos de orden primo tiene los siguientes componentes:*

- *Generación de claves.*
  - *Un algoritmo probabilístico con tiempo polinomial gen, el cual es el generador del grupo, que tiene como entrada a k, con $k \in$ IN (parámetro de seguridad), y tiene como salida un primo q y un valor desc (que representa la descripción de un grupo $H_{q,desc}$ de orden q).*
  - *Una familia de conjuntos $(All_k)_{k\in\mathrm{IN}}$ de parejas $(q, desc)$ (que representan todas las descripciones de los grupos aceptables, las cuales contiene información sobre la familia para la parte que no ha elegido el grupo). Sea $All := \cup_k All_k$.*
  - *Un algoritmo verificador de grupos, Alg-ver, que con la entrada de una tercia $(k, q, desc)$ decide en un tiempo polinomial si $(q, desc) \in All_k$ en la primer entrada solamente.*
- *Una familia de grupos, $(H_q, desc)_{q,desc\in All}$. Los elementos del grupo pueden ser representados por vectores de cadenas de bits.*
- *Un algoritmo que con una primer entrada de $(q, desc) \in All$ y en un tiempo polinomial hará lo siguiente:*
  - *Elegirá elementos aleatorios en $H_{q,desc}$ con una distribución uniforme.*
  - *Probará la calidad de los miembros de $H_{q,desc}$.*
  - *Calculará las operaciones en $H_{q,desc}$.*

Estas componentes deben tener las siguientes propiedades adicionales:

i) Todos los grupos de descripciones que son generados correctamente son aceptables, es decir $[gen(k)] \subseteq All_k \; \forall \; k \in \mathbb{N}$, *y la longitud de los elementos de* All_k *es polinomial en* k.

ii) Propiedades de todos los grupos aceptables: $\forall \; k \in \mathbb{N}$ *y* $\forall \; (q, desc) \in All_k$, *el valor* q *es un número primo mayor que* 2^k, *el orden del grupo* $H_{q,desc}$ *es* q.

b) Se dice que el **logaritmo discreto es difícil** *en tal familia de grupos si y sólo si, para cada algoritmo probabilístico con tiempo polinomial* $\tilde{A}$, *se tiene una probabilidad insignificantemente pequeña de que el algoritmo* $\tilde{A}$, *con la entrada de un grupo de descripción generado correctamente y dos elementos del grupo (uno de ellos generador), encuentre el logaritmo discreto de estos dos elementos.*

Esto es: $\forall \; c > 0 \; \exists \; k \geq k_0$, *tal que:*

$$P(e = log_g(g^*) \mid (q, desc) \leftarrow gen(k);\; g \in H_{q,desc} \setminus \{1\} :$$
$$g^* \in H_{q,desc} :\; e \leftarrow \tilde{A}(k, q, desc, g, g^*)) < k^{-c}$$

c) El **resumen de la suposición del logaritmo discreto** *es que una familia de grupos de orden primo, en donde el logaritmo discreto es difícil, existe.*

$\Diamond$

Comentario: La definición 4.4.2a, implica que la longitud de los elementos de todos los grupos aceptables son polinomial en $(q, desc)$, porque los elementos distribuidos uniformemente pueden ser seleccionados en tiempo polinomial. Por tanto la longitud también es polinomial sobre k.

Suposición del Logaritmo Discreto: En subgrupos de campos con orden primo

La suposición más conocida e investigada del logaritmo discreto es sobre grupos multiplicativos de campos finitos con orden primo, es

decir, *grupos cíclicos* $\mathbb{Z}_p^*$ de orden $p-1$. Por lo que es llamada suposición estandar del logaritmo discreto. Claro que $p-1$ es par, por lo que no es primo, y por tanto estos grupos no pueden aplicarse directamente aquí.

Si $p-1$ tiene un primo factor q, entonces $\mathbb{Z}_p^*$ tiene un único subgrupo de orden q^2. De acuerdo a la sección 4.1.3, este grupo puede ser descrito como:

$$H_{q,p} := \{g \in \mathbb{Z}_p^* \mid g^q = 1\}.$$

Conforme a la notación de la definición 4.4.2 estos grupos son aplicados como: $desc := p$ es usado como la descripción del grupo, junto con q. Los elementos del grupo son representados como es usual, es decir, enteros módulo p. Por lo que las siguientes suposiciones están hechas tomando al algoritmo generador del grupo como un parámetro libre:

Definición 4.4.3 *a) Un* **sistema generador para subgrupos de campos con orden primo**, *está compuesto por los siguientes puntos:*

- *Un algoritmo probabilístico con tiempo de ejecución polinomial "gen" (algoritmo generador del grupo).*
- *Una función* len_p.
- *Un algoritmo con tiempo polinomial* len_{p^*} *que calcula a* len_p, *en un sistema unario, donde gen, con la entrada* $k \in \mathbb{N}$, *tiene como salida un par de primos* (p,q) *con:*

$$q > 2^k \text{ y } q \text{ divide a } p-1 \text{ y } |p|_2 \leq len_p(k).$$

b) Se dice que el **logaritmo discreto es difícil** *para tal sistema generador o* **la suposición concreta del logaritmo discreto** *se cumple para éste, si y sólo si para cada algoritmo probabilístico con tiempo polinomial* $\tilde{A}, \forall\, c > 0$ *y para* k *suficientemente grande,*

[2]Vid el Teorema 4.1.6

$$P(e = log_g(g^*) \mid (q,p) \leftarrow gen(k);\ g \in H_{q,p} \setminus \{1\};$$
$$g^* \in H_{q,p} \leftarrow \tilde{A}(k,q,p,g,g^*)) < k^{-c}$$

$\Diamond$

Construcción 4.4.1. Dado un sistema generador para subgrupos de campos de orden primos[3]. La familia de grupos de orden primo esta definida por los siguientes componentes:

- Generador de la clave.
  - Se usa el mismo algoritmo *gen*, como en la generación del plan.
  - Los conjuntos All_k están definidos como sigue,

$$All_k := \{(q,p) \in \mathrm{IN} \times \mathrm{IN} \mid q,p \text{ son primos y } q > 2^k \text{ y q divide a p-1 y } |p|_2 \leq len_p(k)\}$$

  - El algoritmo *alg-ver* al tener las entradas (k,q,p), primero verifica que $|q|_2 \leq |p|_2 \leq len_p(k)$; esto es hecho en un tiempo polinomial sobre k. Si se tiene una respuesta afirmativa, se probará si $q > 2^k$, q divida a $p-1$, y que p y q sean primos. Al igual como se discutió con con la factorización, la prueba de Rabin-Miller se aplica en la práctica, aunque ésta presenta un error de probabilidad exponencialmente pequeño en algunas propiedades (pero no en la disponibilidad de servicio). Esta prueba toma un tiempo polinomial sobre la longitud de q y p lo cual implica que puede ser verificado en un tiempo polinomial sobre k.
- El grupo $H_{q,p}$ es el único subgrupo de $\mathbb{Z}_p^*$ de orden primo q.
- El algoritmo adicional trabaja como sigue:
  - Elige elementos uniformemente aleatorios de $H_{q,p}$ al elevar los elementos aleatorios de $\mathbb{Z}_p^*$ a la $\frac{(p-1)}{q}$.

[3]Vid Definición 4.4.3

- Prueba la calidad de los miembros g^* en $H_{q,p}$, al probar primeramente que son miembros de $\mathbb{Z}_p^*$ y después probar que $g^{*q} = 1$.
- Las operaciones del grupo son los mismas de $\mathbb{Z}_p^*$.

◇

Lema 4.4.1 *Si el logaritmo discreto es difícil para el sistema generador dado en subgrupos de campos de orden primo, la construcción 4.4.1, es una familia de grupos de orden primo, en donde el logaritmo discreto es difícil.* ◇

Demostración Como se están tomando todos los algoritmos con tiempo de ejecución polinomial sobre los parámetros correctos. Lo que resta demostrar son las propiedades i) y ii) de la definición 4.4.2a y que el logaritmo discreto es difícil de acuerdo a la definición 4.4.2b.

i) Todos los grupos de descripciones generados correctamente son aceptados por la construcción de *gen* y *All*, y la longitud de los elementos (q, p) de All_k, están acotados por $2len_p(k)$,

ii) Esto es inmediato por las definiciones.

ii) Que el logaritmo discreto sea difícil es la misma formula de la definición 4.4.3b.

■

Ultimos adelantos

Para los grupos multiplicativos de los campos con orden primo, es decir, los que corresponden a la suposición estandar del logaritmo discreto, la situación es bastante similar a la factorización: Los resúmenes de los aldelantos de los últimos años estan en [LeLe90, McCu90, LaOd91]. Desde entonces, el algoritmo Criba en campos de números fue inventado [Gord93a] y su tiempo de ejecución es del orden $L_n[1/3, c]$, mientras que el tiempo de ejecución de los

mejores algoritmos usados en la práctica hasta ahora es del orden de $L_n[1/2, c]$, donde c es del mismo tamaño para ambos.

En la práctica el logaritmo discreto ha sido calculado para primos p, de alrededor de 200 bits. Sin embargo, no se ha hecho más trabajo sobre el logaritmo discreto que en la factorización.

Si se desea saber los retos de las computadoras cuánticas sobre el cálculo del logaritmo discreto consulte [Shor94].

Generación de Algoritmos

Las diferentes propuestas para los algoritmos generadores de grupos tienen en común lo siguiente: Primero, q es elegido como un primo aleatorio de cierta longitud, y entonces a los valores $p = dq + 1$, con d en cierto rango, se les hace la prueba de primalidad. En algunos casos, por ejemplo, si sólamente se toma $d = 2$, la elección de q debe de repetirse si ninguno de los posibles $p's$ es primo.

Parece ser, que generalmente se cree, que cualquiera de los algoritmos *gen* inducen a una suposición del logaritmo discreto. Por ejemplo, en la propuesta estandar DSS [DSS91], junto con [Schn91], p es elegido de acuerdo a la suposición estandar del logaritmo discreto, es decir entre 312 y 1024 bits, de longitud, y q es mucho más pequeño (160 bits), lo cual corresponde a una suposición, que no existe un algoritmo para subgrupos que sea subexponencial sobre el logaritmo discreto.

Si se quiere una relación con la suposición estandar del logaritmo discreto, se debe asegurar que el cociente $d = (p-1)/q$, es decir el índice del subgrupo, es únicamente polinomial sobre k.

En este caso, cualquier método para calcular el logaritmo discreto en subgrupos $H_{q,p}$, induce a un método para calcular logaritmos discretos en $\mathbb{Z}_p^*$, el cuál está acotado por abajo en forma polinomial sobre k; Dados $g, g^* \in \mathbb{Z}_p^*$ primeramente calcule el logaritmo discreto e^* de g^*d con respecto a g^d en $H_{q,p}$. Así $g^{de^*} = g^{*d}$, por tanto $de^* = log_g(g^*) \pmod{p-1}$, y $e := log_g(g^*)$ pueden ser buscados a través de las d soluciones de esta congruencia.

Capítulo 5

Construcción de las FIP una-vez

5.1 Construcción de FIP

En este capítulo se presenta la construcción general de un plan de firmas intrísecamente protegido, la cual es subsecuentemente usada en ejemplificaciones reales. Se presentarán dos ejemplificaciones basadas en las suposiciones criptográficas de la dificultad de calcular el logaritmo discreto y la dificultad de factorizar enteros. Estas son las suposiciones computacionales mejor conocidas que se usan en criptografía.

Como se ha descrito, las construcciones de esta sección permiten únicamente firmar un mensaje, por tanto son llamadas **firmas una-vez**. En el capítulo 6 se presentará la generalización, para poder firmar más de un mensaje.

Para ver con más detalle la información de esta sección consulte [PePf] y [Pf96].

5.1.1 Homomorfismos fibrados

A manera de presentar la construcción general de las firmas intrínsecamente protegidas, primero se definen los homomorfismos fibrados. Estos son un caso especial de las funciones criptográficas digestivas. Un homomorfismo fibrado h, es

un homomorfismo $h : G \to H$ entre dos grupos Abelianos $(G, +, 0)$ y $(H, \cdot, 1)$ que satisface lo siguiente:

- Cada imagen $h(x)$ tiene al menos 2^τ preimagenes. (Se dice que el homomorfismo fibrado es de grado 2^τ).
- Es difícil encontrar colisiones.

A manera de hacer el segundo de estos requerimientos más preciso, se considerará una familia de tales funciones. Cada una de las funciones de esta familia es indexada por una llave K. La selección de la llave depende de dos parámetros: τ , el cual determina el grado del homorfismo fibrado, y k, que mide la seguridad computacional contra el encuentro de colisiones. Los parámetros τ y k son parte del indice para la función, pero solamente se escribirá K en lugar de (K, τ, k).

Considerando esto a manera de introducción, se proseguirá con la siguiente definición:

Definición 5.1.1 *Una familia de homomorfismos fibrados es un cuádruple $(g, h, \mathcal{G}, \mathcal{H})$, donde:*

- *g es un algoritmo probabilístico generador de llaves, el cual es ejecutado en un tiempo polinomial sobre la entrada de los parámetros $k, \tau \in \mathbb{N}$ y tiene como salida un valor K. Sea $\mathcal{K}$ el conjunto de todas las posibles llaves, es decir, la unión de los conjuntos $[g(k, \tau)]$ para todo $k, \tau \in \mathbb{N}$.*
- *$\mathcal{G}$ y $\mathcal{H}$ son familias de grupos Abelianos, y a cada llave le corresponde un par de grupos. Esto es, $G_\mathcal{K} = (G_K, +, 0)_{K \in \mathcal{K}}$ y $H_\mathcal{K} = (H_K, \cdot, 1)_{K \in \mathcal{K}}$.*
- *h, es un algoritmo con un tiempo polinomial que sobre la entrada de $K \in \mathcal{K}$ y $x \in G_K$ tiene como resultado $h(x) = z \in H_K$. La restricción de h para la llave particular K es denotado por h_K.*

  Esta cuádrupla satisface las siguientes propiedades:

  a) Cada h_K es un homomorfismo del grupo $(G_k, +, 0)$ al grupo $(H_K, \cdot, 1)$.

b) Para todo $k, \tau \in \mathbb{N}$, $K \in [g(k,\tau)]$, *cada* $z \in h_K(G_K)$, *tiene al menos* 2^τ *preimagenes bajo* h_K.

c) La familia es resistente a colisiones: Para toda $c > 0$ *y para cada algoritmo probabilístico* $\tilde{A}$, *con tiempo polinomial, se tiene que la probabilidad de que* $\tilde{A}$, *con la entrada* $K \in [g(k,\tau)]$ *de un resultado de* (x,x') *tal que* $x \neq x'$ *y* $h_K(x) = h_K(x')$, *es menor que* k^{-c}, *para* k *suficientemente grande. Esto es,* $\forall\ \tau\ \forall\ c\ \exists\ k_0$, *tal que* $\forall\ k \geq k_0$,

$$P(h_K(x) = h_K(x')\ y\ x \neq x'\ ::$$
$$K \leftarrow g(k,\tau); (x,x') \leftarrow \tilde{A}(K)) < k^{-c}.$$

(Un nombre mas común para "resistencia a colisiones es" "libre de colisiones", pero el nombre de resistente a colisiones enfatiza mejor el aspecto computacional.)

Adicionalmente, debe haber algoritmos con tiempo polinomial que sobre la entrada K*:*

- *Calculen las operaciones en los grupos* $(G,+,0)$ *y* $(H,\cdot,1)$,
- *Seleccionen elementos de* G_K *uniformemente aleatorios, y*
- *Prueben la calidad de los miembros de* H_K *y* G_K.

$\Diamond$

Note que k_0 depende de τ en la definición de resistencia a colisiones. En las contrucciones 3.1.2 y 3.1.4, k_0 es independiente de τ.

5.1.2 Construcción general

En seguida se presenta un plan de firmas intrínsecamente protegido, donde solamente puede firmarse un mensaje. Aquí, la generación de llave (semejante a todas las construcciones de la literatura) es bastante simple.

Definición 5.1.2 *Un plan de FIP una-vez con prellave, se define de manera semejante al plan de FIP de la Definición 3.3.1, la diferencia es que el parámetro N es igual a 1 y el protocolo G, es de una forma especial: G es construido por una tercia $(gen_C, (P,V), gen_A)$, donde:*

- *gen_C, es un algoritmo probabilístico con tiempo polinomial, que sobre la entrada* $\mathbf{par}^* = (k, \sigma)$, *genera los valores:* **prek** *(la prellave) y w (es llamado testigo).*

- *(P, V), es el* **protocolo verificador** *de la prellave, que es llevado a cabo entre dos partes y es ejecutado con un tiempo de ejecución polinomial, donde P tiene como entrada (par*, prek, w) y V únicamente (par*, prek). La sadida del algoritmo V es aceptar o rechazar. La salida correcta de la generación de la prellave debe ser siempre aceptar, es decir, si $(prek, w) \in [gen_C(k, \sigma)]$, la salida de V debe ser aceptar. Los casos especiales más eficientes es donde P , no se aplica, es decir, V decide localmente en aceptar o rechazar la prellave, prek.*

- *gen_A es un algoritmo probabilístico con tiempo de ejecución polinomial que con la entrada de una prellave "prek", tiene como salida un par de llaves (sk, pk). Este algoritmo es llamado el algoritmo principal de la generación de la llave.*

El protocolo G se ejecutará como sigue: Primeramente, el centro C ejecutará gen_C y publicará las prellaves resultantes. En seguida, (P V) es ejecutado para esta prellave por el centro (P) y el firmante (V), donde el centro tiene a w como una entrada adicional. (El propósito de este paso es probar algúnas propiedades deseadas de prek, para la seguridad del firmante.) Finalmente, el firmante A ejecuta gen_A sobre la entrada prek. Se dice que se generó un par de llaves basado en la prellave prek. La llave pública es el par (prek, pk). Sin embargo, con frecuencia se omitirá en la notación de prek. ◇

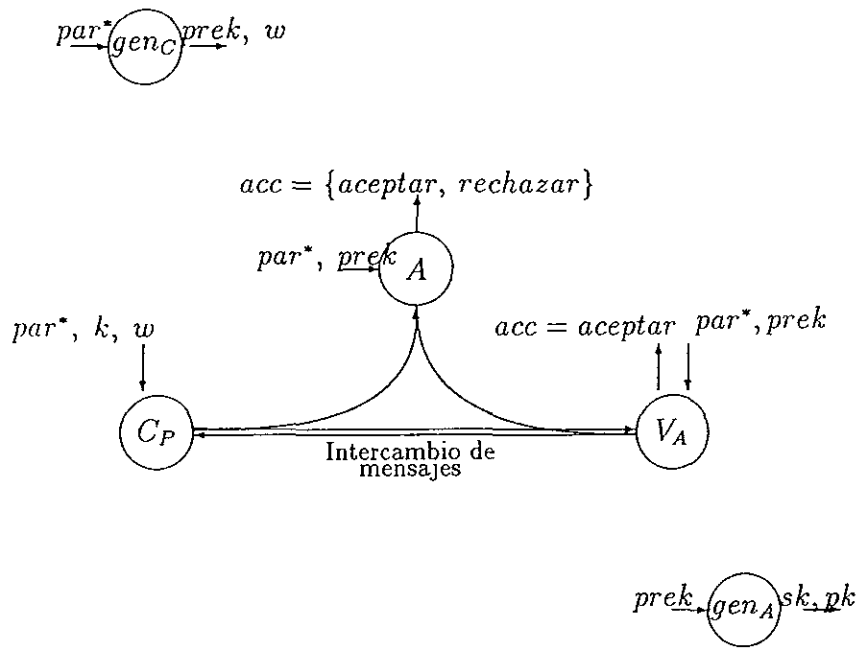


Figura 5.1.1. Descripción de la ejecución correcta del protocolo generador de llaves, G, para un plan de firmas con prellave.

El firmante, también puede aceptar a *prek* (utilizando V) aun cuando *prek* no sea un posible resultado de la generación correcta de la prellave. Una de las razones por la que se usa (P, V) es para probar las propiedades de *prek*, que son necesarias para la seguridad del firmante. A menudo, esto es mucho más eficiente que probar una generación correcta de la llave.

Los planes con prellave tienen la siguiente ventaja si hay varios firmantes. El centro puede publicar la prellave sin conocer cuales son los firmantes que tomarán parte en el sistema. Cada firmante puede llevar acabo la demostración interactiva con el centro una vez y entonces puede generar muchas llaves secretas sucesivas sobre esta prellave sin una interacción posterior con el centro. La seguirdad no

se debilita aunque sean muchos firmantes los que basen sus llaves sobre la misma prellave por lo siguiente:

- Si es dirigido para los firmantes que son capaces de rechazar sus propias firmas, un firmante fraudulento solamente podrá generar muchos pares de llaves basadas sobre la prellave y experimentar con ellas localmente hasta que pueda rechazar una firma, y hasta entonces, él podrá publicar la llave pública correspondiente.

- En caso de que haya falsificadores, que puedan hacer falsificaciones improbables, con una probabilidad no insignificante (sobre la elección de los pares de llaves de todos los firmantes), la probabilidad de falsificación para cada par de llaves, debe ser también no insignificante, porque todos los pares son identicamente distribuidos.

En seguida se describe la estructura general para las construcciones de planes de FIP una-vez con prellave, basados en una familia de homomorfismos. Los parámetros publicados en el plan dependen de la elección de la familia de homomorfismos fibrados. También se presentan dos teoremas que reducen la seguridad del plan de firmas a una propiedad de los homomorfismos fibrados y a sus parámetros. Esta propiedad será probada y los parámentros serán especificados en las ejemplificaciones de las secciones 5.2 y 5.3.

Como ya hemos dicho, ésta construcción es para firmar únicamente un mensaje por tanto $i = 1$, y en lugar de una secuencia de mensajes $\underline{m}$ de longitud 1, se escribirá m.

Construcción 5.1.1 (Construcción General). Considerese una familia de homomorfismos fibrados con un generador de llave g. En base a esto se definen los componentes de un plan de firmas intrínsecamente protegidas una-vez con prellave, como sigue:

- Generación de la llave: Los parámetros de seguirdad son k y σ. El parámetro τ que define el grado del homomorfismo fibrado esta en función de σ.

  - Generación de la prellave: gen_C. El centro calculará $K \leftarrow g(k, \tau)$ y la publicará, es decir, $prek := K$. Esta corresponde a la selección de un homomorfismo h_K de la familia. Sea $h := h_K$, $G := G_K$, y $H := H_K$. En lugar de g se usar un algoritmo denotado por g' que dará como salida a K con la misma distribución de probabilidad que g y además dará la salida testigo, w. La información contenida en w es usada para una demostración de cero-conocimiento[1].

  - Verificación de la prellave (P, V): El firmante se debe asegurar de que K es un posible resultado de $g(k, \tau)$, (es decir, que cumple con las tres condiciones de la definición 5.1.1 o a menos que h sea un homomorfismo qur satisface el inciso b) de la definición 5.1.1. La prueba de una de estas dos opciones se hace dependiendo de la elección de la familia de homomorfismos fibrados. Una prellave K es llamada **buena** si satisface alguna de estas dos opciones, en otro caso es llamada **mala**. Este protocolo es una prueba local para el firmante. De otra forma, esto es una demostración de cero-conocimiento. La probabilidad de que el firmante acepte una demotración sobre una mala prelcave K debe ser a lo más $2^{-\sigma}$ para cada K.

  - Generación principal de de llave: gen_A. El firmante genera su llave secreta $sk := (sk_1, sk_2)$ al seleccionar sk_1 y sk_2 aleatoriamente de G, y calcula su llave pública $pk := (pk_1, pk_2)$, donde $pk_i = h(sk_i)$ para i = 1,2.

- El espacio de mensajes M es un subconjunto de $\mathbb{Z}$ que depende de la elección de la prellave.

[1]Se dice que una demostración es una demostración de cero-conocimiento si no proporciona información extra, que la que se expone en ella. Ver [GMRac]

- El firmado: La firma correcta sobre un mensaje m del espacio de mensaje es,

$$firma(sk, m) := sk_1 + msk_2.$$

(Las multiplicaciones de los elementos de $\mathbb{Z}$ es como de costumbre, definido por adiciones repetidas).

- Prueba: El algoritmo *prueba*, el cual determina si una firma $s \in G$ es aceptable, es decir,

$$prueba(pk, m, s) = aprobar \iff pk_1 \cdot pk_2^m = h(s).$$

- Demostración de Falsificación: Dada una firma aceptable $s' \in G$ sobre m, tal que $s' \neq firma(sk, m)$, entonces el firmante calcula $s := firma(sk, m)$ y la demostración es implicada al mostrar dos firmas distintas, denotada por $dem := (s, s')$.

- Verificación de una demostración de falsificación: Dado un par (x, x'), primero verificar que x y x' son elementos de G, además que $x \neq x'$ y para concluir la verificación de falsificación se calcula $h(x)$ y $h(x')$, y si $h(x) = h(x')$, entonces se dice que es una demostración de falsificación válida.

Esto concluye la construcción general de un plan de firmas intrínsecamente protegido. Los siguientes teoremas muestran que cualquier ejemplificación que trabaje con esta construcción es segura para los receptores.

Teorema 5.1.1 *Para cualquier familia de homomorfismos fibrados y con cualquier elección de los parámetros que han sido públicados, la construcción general tiene las siguientes propiedades:*

a) Las firmas correctas siempre pasan la prueba (es decir, la Definición 3.3.1 es satisfecha).

b) Un firmante con tiempo polinomial no puede construir una firma y también una demostración de falsificación válida sobre ésta. (Definición 3.3.2).

c) Si s^* es una firma aceptable sobre m^* y $s^* \neq firma(sk, m^*)$, entonces el firmante obtendrá una demostración de falsificación válida (Definición 3.3.4).

d) Si todos los valores $(sk, (K, pk), aux_{\tilde{C}})$, donde K es Bueno, están dentro de $Bueno_{\tilde{C}}$, el plan es seguro para el firmante (Definición 3.3.4).

$\diamondsuit$

Demostración.

a) Si el firmante ha ejecutado correctamente el algoritmo generador de las llaves (sk, pk) y él utiliza sus llaves secretas para firmar un mensaje m, entonces su firma será:

$$s = sk_1 + msk_2$$

por ser h, un homomorfismo y $h(sk_i) = pk_i$, $i = 1, 2$. Entonces,

$$\begin{aligned} h(s) &= h(sk_1 + msk_2) \\ &= h(sk_1) \cdot h(sk_2)^m \\ &= pk_1 \cdot pk_2^m \end{aligned}$$

por tanto la firma correcta s del firmante pasa la prueba.

b) Primeramente recordemos, que una demostración de falsificación se hace al mostrar dos firmas $s \neq s'$, sobre un mismo mensaje, tal que $h(s) = h(s')$. También recordemos que K es buena es decir, h_k satisface al menos c) de la definición 5.1.1 para todo algoritmo con tiempo de ejecución polinomial. Por tanto, un firmante con un poder computacional polinomial no puede construir un par de firmas sobre un mismo mensaje, para hacer una demostración de falsificación válida.

c) La demostración de falsificación es inmediata al mostrar una pareja (s, s'), de firmas aceptables.

d) La seguridad del firmante se basa en la propiedad (b) sobre los homomorfismos de la definición 5.1.1. Si esta propiedad se cumple, entonces el firmante puede hacer demostraciones de falsificación.

■

Este teorema demuestra que la construcción general es segura para los receptores y también es segura para el firmante si un falsificador arbitrariamente potente no puede adivinar la firma correcta $firma(sk, m^*)$, excepto por una probabilidad muy pequeña, aunque la prellave sea buena. A manera de estimar la probabilidad con la cual un falsificador puede encontrar tal firma, primero nótese que a las llaves públicas les corresponde $2^{2\tau}$ llaves secretas. Después de que se ha firmado un mensaje m correctamente, el falsificador tiene más información sobre las llaves secretas, por tanto, se le facilitará más tratar de hacer una falsificación. El Teorema 5.1.2 da la condición que asegura que esta información, no es suficiente para adivinar la firma correcta sobre $m^* \neq m$, con una probabilidad grande.

Teorema 5.1.2 *Considérese la construcción 5.1.1, además los parámetros k y σ, una prellave buena K, y dos mensajes $m \neq m^*$ del espacio de mensajes. Sea,*

$$T := \{d \in G \mid h(d) = 1 \ y \ (m - m^*)d = 0\}.$$

Entonces para todo par de llaves $(sk, pk) \in [gen_A(K)]$ y todos los valores $s^* \in G$ (falsos) que satisfacen la, $prueba(pk, m^*, s^*) =$ aprobar, se tiene que la probabilidad de obtener $s^* = firma(sk, m^*)$, dado que se conoce $s := firma(sk, m)$, es a lo más $|T|/2^\tau$, es decir, para cualquier centro$\tilde{C}$;

$$P(s^* = firma(sk', m^*) :: sk' \leftarrow SK_{\tilde{C}}(pk, (m, s), aux_{\tilde{C}})) \leq |T|/2^\tau.$$

◇

Note que la probabilidad en este teorema es semejante a la Definición 3.3.4 (a).

Demostración.

Dado que K es buena, h es al menos un homomorfismo y satisface el inciso b) de la Definición 5.1.1. Notese que $aux_{\tilde{C}}$ puede contener información adicional acerca de K pero la única información que el firmante divulga acerca de sk es pk y una firma correcta s sobre un mensaje m. El conjunto de las posibles llaves secretas que aportan información son:

$$SK_{\tilde{C}} = \{(sk'_1, sk'_2) \in G \times G \mid h(sk'_1) = pk_1 \ y \ h(sk'_2) = pk_2 \ y \ sk'_1 + msk'_2 = s\} = \{(s \cdot msk'_2, sk'_2) \mid h(sk'_2) = pk_2\}$$

porque h es un homomorfismo y $s = sk_1 + msk_2$. El tamaño de $SK_{\tilde{C}}$ es por tanto al menos 2^τ. Se debe ahora encontrar cuantas de estas llaves satisfacen $s^* = firma(sk', m^*)$, es decir,

$$sk'_1 + m^* sk'_2 = s^* \tag{1.1.1}$$

Dado que solamente se consideran las llaves en $SK_{\tilde{C}}$ se puede reemplazar a sk'_1 por $s - msk'_2$. Por tanto 1.1.1 es equivalente a

$$(m^* - m)sk'_2 = s^* - s.$$

Esta ecuación puede no tener solución, pero si hay alguna solución sk''_2, el conjunto de todas las soluciones en $SK_{\tilde{C}}$ es:

$$\{(s - msk'_2, sk'_2) \mid h(sk'_2) = h(sk''_2) \ y \ (m^* - m)(sk'_2 - sk''_2) = 0\}.$$

Por tanto el número de soluciones es $|T|$ (donde d corresponde a la diferencia $sk'_2 - sk''_2$). Dado que sk es uniformemente distribuida en al Construcción 5.1.1. todos los elementos de $SK_{\tilde{C}}$ son igualmente probables y el atacante es exitoso con una probabilidad de a lo más $|T|/|SK_{\tilde{C}}| \leq |T|/2^\tau$.

■

Corolario 5.1.3 *El Teorema 5.1.2 junto con los inciso (c-d) del Teorema 5.1.1. demuestran que la construcción general es segura para el firmante (como en la Definición 3.3.4) si τ es elegido tal que $|T|/2^\tau \leq 2^{-\sigma}$, es decir, $\tau \geq \sigma + log_2(|T|)$.*

Consecuentemente, se necesita encontrar el máximo tamaño de

$$T_{m'} := \{d \in G \mid h(d) = 1 \ y \ ord_G(d) | m'\}$$

sobre todas las posibles diferencias m', de dos mensajes. El tamaño de este conjunto depende de la familia de los homomorfismos fibrados.

◇

5.2 Plan de FIP basado en PLD

En esta sección se presenta una ejemplificación de firmas intrínsecamente protegidas, que se basa en la construcción de una familia de homomorfismos fibrados, en donde encontrar una colisión del homomorfismo, equivale a calcular el logaritmo discreto. [Vease la sección 4.4]

Para comenzar con la construcción es importante tener presente el Teorema de Cauchy para grupos Abelianos.

Teorema 5.2.1 *Supóngase que G es un grupo cíclico finito y que hay un primo p, tal que, $p||G|$. Entonces existe un elemento $a \neq e \in G$, tal que, $a^p = e$.* ◇

Considerando este teorema, se elegirán dos primos de tamaño grande, p y q tal que q divida a $p-1$, esto implica que hay un único subgrupo H_q de $\mathbb{Z}_p^*$, de orden q. Recordando que todos los grupos $\mathbb{Z}_p^*$ son cíclicos y de orden $p-1$ [2]. Además todos los elementos de H_q son generadores, excepto la identidad.

Se asumirá que es difícil calcular el Logaritmo Discreto (LD) en H_q. El Problema del Logaritmo Discreto en subgrupos de $\mathbb{Z}_p^*$ está dado como sigue:

Definición 5.2.1 *Dados p y q, y dos elementos $a, b \in H_q$, donde $a \neq 1$, entonces el LD, $log_a(b)$ es definido como el número x, $0 \leq x \leq q-1$, tal que $a^x \equiv b \pmod{p}$.* ◇

[2] Vease [MOV]

Suposición de intractabilidad del LD.

Para todo algoritmo probabilístico D con tiempo polinomial, para $c \in \mathbb{N}$ y para k suficientemente grande, la probabilidad de que D, con entradas de los primos p y q y dos generadores a y b de H_q, obtenga el resultado $\log_a(b)$ es menor que k^{-c}. Donde q es un primo con una longitud de k-bit (esto es $|q|_2 = k$), divisor de $p-1$ y $q > (p-1)/q$.

La probabilidad es sobre los bits aleatorios usados en D y la selección aleatoria de p, q, a, b.

$\Diamond$

Construcción 5.1.2 Homomorfismos fibrados para el plan basado en el logaritmo discreto[3].

- Generador de llave, g: El generador g, sobre la entrada k, τ seleccionará los primos p y q con las características mencionadas, tal que $|q|_2 = max(k, \tau)$ y también dos generadores aleatorios a y b de H_q. La llave resultante es $K := (p, q, a, b)$.

- Las familias de grupos están definidas como sigue:

$$G_q := \mathbb{Z}_q \times \mathbb{Z}_q$$

  donde las operaciones en G_q son entrada a entrada con la operación adición (módulo q) y la operación en H_q es la multiplicación (módulo p). En lugar de escribir $G_{(a,b,p,q)}$ escribiremos G_q y en lugar de $H_{(a,b,p,q)}, H_q$.

- El homomorfismo es definido como:

$$h_{(a,b,p,q)} : G_q \longrightarrow H_q;$$
$$h_{(a,b,p,q)}(x, y) := a^x b^y \pmod{p}.$$

[3]Vid la sección 4.2.1

El siguiente teorema prueba que la construcción 5.1.2 es realmente una familia de homomorfismos fibrados.

Teorema 5.2.2 *Bajo la suposición de la dificultad para calcular el LD, la Construcción 5.1.2 es una familia de homomorfismos fibrados.*

◊

Demostración. Es claro que todas las operaciones necesarias se pueden calcular eficientemente. En particular, g genera a q y entonces busca un primo p a través de los números de la forma $tq+1$. Ahora la Definición 5.1.1 (a-c) debe ser verificada.

(a) Se probará que cadà $h_{(a,b,p,q)}$ es un homomorfismo de G_q a H_q.

Sean (x,y) y $(x',y') \in G_q$, entonces

$h((x,y)+(x',y')) = h(x+x', y+y') \pmod p$

$= a^{x+x'}b^{y+y'} \pmod p \;=\; (a^x b^y)\cdot(a^{x'}b^{y'}) \pmod p$

$= h(x,y)\cdot h(x',y')$

(b) Se probará ahora que para cada $z \in H_q$, hay q elementos (x,y) de G_q, tal que h mapea a z: para cada x hay exactamente un y, tal que $b^y \equiv za^{-x} \pmod p$, porque a es generador.

Ahora sólo se necesita saber la cardinalidad del kernel. Esto es el conjunto de pares (x,y), tal que $h(x,y) = a^x b^y = 1 \pmod p$. Como a es un generador de H_q, entonces existe un número $l \neq 1$ tal que $b = a^l \pmod p$. Entonces $h(x,y) = a^x a^{ly} \pmod p = 1$, esto implica que $a^x = a^{-ly} \pmod p$. Por tanto el conjunto de elementos en el kernel son de la forma $(-ly, y)$, y como H_q es de cardinalidad q, por lo tanto hay exactamente q elementos en el kernel. Por tanto para cada $z \in H_q$ existen exactamente q preimagenes.

(c) Se necesita probar que h_q es resistente a colisiones: La prueba se hará por contradicción. Supóngase que un algoritmo probalístico $\tilde{A}$, con tiempo polinomial puede calcular colisiones sobre h_q, entonces se puede construir un algoritmo D que con la entrada de las llaves públicas (p,q,a,b) calcula el LD de a

con respecto a b de la siguiente manera: Primero D ejecutará el algoritmo $\tilde{A}$ y si $\tilde{A}$ tiene como salida una colisión es decir, $(x, y) \neq (x', y')$, tal que,

$a^x b^y = a^{x'} b^{y'} \pmod{p}$,

$a^{x-x'} = b^{y'-y} \pmod{p}$

pero como b es un generador de H_q, existe un número e tal que $a = b^e \pmod{p}$, por tanto

$b^{e(x-x')} = b^{(y'-y)} \pmod{p}$, $b^{e(x-x')-(y'-y)} = 1 \pmod{p}$,

y como el orden de b es q por tanto q divide a $e(x-x')-(y'-y)$.

Ahora tenemos,

$e(x - x') - (y' - y) = kq \pmod{q}$,

$e(x - x') = (y' - y). \pmod{q}$, por tanto

$e = (y' - y)(x - x')^{-1} \pmod{q}$,

entonces D calculará el $\log_b(a)$ como: $(y' - y)(x - x')^{-1} \pmod{q}$. Por tanto, construir un algoritmo $\tilde{A}$ que calcule colisiones bajo h_q es tan difícil como construir un algoritmo D que calcule el logaritmo discreto. Por tanto ésto sería una contradicción a la suposición de la dificultad del problema del logaritmo discreto. ■

Este teorema implica que se puede construir un plan de FIP que sea seguro para los receptores bajo la suposición de que es difícil calcular el logaritmo discreto. Antes de seguir con los detalles para la seguridad del firmante se describirá el plan resultante y los parámetros que serán publicados.

Construcción 5.1.3 Plan de FIP basado en el PLD, bajo la construcción 5.1.2.

Notese que ésta construcción es un plan de FIP con prellave[4].

[4]Vid definición 5.1.2.

- Generación de la llave: Sobre la entrada de k y σ, donde el parámetro τ que determina el grado del homomorfismo esta en función de σ.

  - Generación de la prellave: El centro selecionará los primos p y q y los generadores a y b (con el algoritmo g), y los publicará. Así $prellave = (p, q, a, b)$.
  - Verificación de la prellave: El firmante puede verificar que p y q son primos y que a y b son generadores de H_q, al verificar que su orden es q. (Como se observa la generación de la llave es muy simple para este plan).
  - Generación principal de la llave: El firmante seleccionará la llave secreta que consiste de cuatro números x_1, x_2, y_1 y y_2 entre 0 y $q-1$, esto es, $sk = \{(x_1, y_1), (x_2, y_2)\} \in G_q$, y calculará la correspondiente llave pública (pk_1, pk_2) como,

$$pk_1 := a^{x_1} b^{x_2} \pmod{p} \quad \text{y} \quad pk_2 := a^{y_1} b^{y_2} \pmod{p}$$

- El espacio de mensajes esta definido en el conjunto $\{0, \ldots, q-1\}$.

- El firmado: La firma correcta sobre un mensaje m de este espacio es:

$$\begin{aligned}(s_1, s_2) &:= (x_1, yx_2) + m(y_1, y_2) \\ &= ((x_1 + my_1) \pmod{q}, (x_2 + my_2) \pmod{q})\end{aligned}$$

- Prueba de la firma: Un par (s_1, s_2) es una firma aceptable sobre el mensaje m si y sólo si

$$a^{s_1} b^{s_2} \equiv pk_1 pk_2^m \pmod{p}$$

- Demostración de falsificación y su verificación: De acuerdo a la construcción general, una demostración de falsificación es una colisión bajo h_k. Por lo tanto tal demostración consiste de cuatro números. Sin embargo, el firmante puede justificar la demostración de falsificación al mostrar $e := \log_a(b)$, esto es equivalente a la otra demostración pero más corta y fácil de verificar. Porque se puede exibir una colisión,

$$h(e,0) \equiv h(0,1) \pmod{p}, \text{ donde}$$
$$a^e b^0 \equiv a^0 b^1 \pmod{p}.$$

Hasta este momento únicamente se ha considerado la seguridad del receptor. El siguiente teorema muestra que este plan también es seguro para el firmante.

Teorema 5.2.3 *El Plan de FIP basado en el Logaritmo Discreto es seguro para el firmante:* ◇

Demostración. De acuerdo con el Corolario 5.1.3, se necesita encontrar el tamaño máximo del conjunto

$$T_{m'} = \{d \in G_q \mid h(d) = 1 \ y \ ord(d)/m'\}$$

para todos los valores m' entre 0 y $q-1$, (m' es la diferencia de dos mensajes diferentes). Como q es primo, y el orden de todos los elmentos distintos de 0 de G_q es q. Por tanto $(0,0)$ es el único elemento de $T_{m'}$. Esto implica que es suficiente elegir $\tau := \sigma$ en la generación de la llave.

Por tanto este plan de FIP con prellave es seguro para el firmante.

■

Al elegir $\tau := \sigma$ es semejante a que $|q|_2$ sea elegida como el $max(k,\sigma)$. Para dos parámetros razonables k y σ, es lo mismo a que $|q|_2 = k$.

La evaluación de eficiencia sobre este plan es la siguiente:

- La firma requiere de dos multiplicaciones módulo q.

- La prueba de la firma requiere menos dos exponenciaciones módulo p. Esto es porque los receptores pueden probar la firma (s_1, s_2) por el cálculo de $a^{s_1} b^{s_2} pk_2^{-m}$ y verificar que este resultado sea igual a pk_1. Esto equivale a un valor entre k y $2k$ multiplicaciones modulares, dependiendo de los mensajes.
- La longitud de las llaves secretas es $4k$.
- La longitud de la llave pública es $2|p|_2$ bits.
- La longitud de una firma sobre un mensaje de tamaño k-bits es $2k$.

En la siguiente sección se presentará una construcción muy semejante a ésta. Una de las diferencias es que ahora se basa en otro problema, la dificultad de factorizar números enteros grandes.

El siguiente ejemplo está basado en el plan sobre el logaritmo discreto.

Ejemplo 5.2.1.1 *Sea $p = 31$ y $q = 5$ (es claro que $q/p-1$). Sea $a = 16$ un generador del subgrupo H_q de $\mathbb{Z}_p^*$. Tomese $b = a^3 \pmod{31} = 4$. Supongase que las llaves secretas del emisor son: $\bar{x} = (x_1, x_2, y_1, y_2) = (3,0,2,3)$; Por tanto la llave pública del emisor es:*

$$(pk_1, pk_2) = (a^3 b^0 \pmod{31},\ a^2 b^3 \pmod{31}) = (4,\ 16)$$

La siguiente tabla muestra los $q^2 = 5^2$ cuádruples de llaves secretas que le corresponden a la misma llave pública (4, 16).

0102	1402	2202	3002	4302
0110	1410	2210	3010	4310
0123	1423	2223	3023	4323
0131	1431	2231	3031	4331
0144	1444	2244	3044	4344

Si consideramos que el mensaje a firmar es $m = 1$, entonces la firma para éste mensaje bajo la clave secreta $sk = (3,0,2,3)$ es:

$$(s_{1,m}, s_{2,m}) = (s_{1,1}, s_{2,1}) = (5 \pmod 5, 3 \pmod 5) = (0, 3)$$

La siguiente tabla muestra las $q = 5$ firmas que se obtienen de las q^2 llaves secretas correspondientes a la clave pública $pk = (4, 6)$ y también muestra las $q = 5$ claves secretas que producen la misma firma para cada una de las $q = 5$ firmas diferentes.

Pares de firmas	*(1,1)*	*(2,4)*	*(3,2)*	*(4,0)*	*(0,3)*
Cuadruplas	*0110*	*0123*	*0131*	*0144*	*0102*
	1402	*1410*	*1423*	*1431*	*1444*
	2244	*2202*	*2210*	*2223*	*2231*
	3031	*3044*	*3002*	*3010*	*3023*
	4323	*4331*	*4344*	*4302*	*4310*

La probabilidad que se tiene para que un adversario encuentre la firma del firmante para un mismo mensaje es, $q/q^2 = 1/q$; esta probabilidad es independiente de los recursos computacionales del adversario.

5.3 Plan de FIP basado en la factorización

En esta sección se muestra una segunda forma de construir un plan de FIP que se basa en la construcción de familias de homomorfismos fibrados, donde la seguridad de los receptores se apoya en el problema de la factorización[5]. Encontrar una colisión en el homorfismo es equivalente a calcular la factorización de un número entero, de tamaño grande, como se hará ver a continuación.

Este plan es un poco mas laborioso que el que se basa en el problema del logaritmo discreto.

Construcción de la familia de homorfismos

Para esto considerese la sección 4.1.3

[5]Vease sección 4.4

Sea $k \in N$ y $B_k := \{p \cdot q \mid p,q \text{ son primos y } \lfloor log_2(p) \rfloor = \lfloor log_2(q) \rfloor = \frac{k-1}{2} \text{ y } p \equiv 3 \pmod 8 \text{ y } q \equiv 7 \pmod 8\}$. Sea $B = \cup B_k$.

Una vez, que se han considerado los números compuestos, necesitamos asegurar que es difícil factorizarlos, por tanto consideraremos la siguiente suposición.

Suposición de Dificultad para la Factorización (SDF). Sea F un algoritmo probabilístico factorizador con tiempo de ejecución polinomial. Entonces para toda constante $c > 0$ y k suficientemente grande se tiene:

$$P(\text{sea } x \text{ un divisor no trivial de } n :: n \leftarrow B_k; x \leftarrow F(n)) < 1/k^c.$$

◇

La probabilidad está implicada por los bits aleatorios usados por F y la selección aleatoria de n.

Como ya se ha dicho, las firmas FIP se construyen sobre parejas de permutaciones libres de colisiones, y que estas a su vez se basan sobre el PFE. En esta construcción se considerarán las permutaciones f_0 y f_1, con dominio D_n, $n \in B_k$. [Vea la sección 4.2.2]

$$D_n := \{x | x \in \mathbb{Z}_n^* \text{ y } (\tfrac{x}{n}) = 1 \text{ y } x \in \{1, 2, \ldots \tfrac{n-1}{2}\}\}.$$

y

$$\begin{aligned} f_0(x) : &= |x^2| \pmod n \\ f_1(x) : &= |4 \cdot x^2| \pmod n \end{aligned}$$

Lema 5.3.1 *Ambas funciones son permutaciones sobre D_n y encontrar una colisión, es decir, (x_0, x_1) tal que $f_0(x_0) = f_1(x_1)$ es tan difícil como factorizar n.* ◇

Demostración. Supongase que existe un algoritmo eficiente para encontrar $x, y \in D_n$ tal que $f_0(x) = f_1(y)$.

Entonces $x_2 = 4y_2 \pmod n$ (nótese que es imposible que $x_2 = -4y_2 \pmod n$ porque $4y^2$ es un residuo cuadrático.

Por tanto $x_2 - 4y_2 \equiv 0 \pmod{n}$. $(x-2y)(x+2y) = 0 \pmod{n}$, donde $x \not\equiv \pm 2y \pmod{n}$, ya que $(\frac{x}{n}) = 1$ y $(\frac{2y}{n}) = -1$, Por tanto $mcd(x \pm 2y, n)$ dará un factor no trivial de n, ya que si:

$$\begin{aligned} x \pm y &= p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k} pq \\ n &= p_1^{d_1} p_2^{d_2} \cdots p_k^{d_k} pq \end{aligned}$$

donde los $e_i \geq 0$, $d_i \geq 0$ y p, q, p_i son primos. Entonces,

$$mcd(x \pm y, n) =$$

$$\begin{aligned} &= p_1^{min(e_1,d_1)} \cdots p_k^{min(e_k,d_k)} p^{min(e_{k+1},d_{k+1})} q^{min(e_{k+2},d_{k+2})} \\ &= p^{min(e_{k+1},1)} q^{min(e_{k+2},1)} \end{aligned}$$

se tienen cuatro casos de los cuales se descartan dos de ellos: uno es cuando $e_{e+1} = e_{e+2} = 0$ y el otro es cuando $e_{e+1} \geq 1$ y $e_{e+2} \geq 1$. Y solo restan los casos cuando $e_{e+1} = 1$ y $e_{e+2} = 0$ ó $e_{e+1} = 0$ y $e_{e+2} = 1$, en donde cualquiera de estos casos tenemos como resultado que $mcd(x \pm y, n) = p$ ó q.

Obtener este resultado implica una contradicción de la suposición criptográfica.

■

Lo que ahora se necesita es un homomorfismo fibrado. Para poder tener un homomorfismo con las características de la definición 5.1.1, se herá lo siguiente: tomense las permutaciones f_0 y f_1 y hagase una composición sucesiva de estas.

Por tanto se tendra una función h definida sobre los grupos D_n y $G = \mathbb{Z}_{2^r} \times D_n$.

Esto es,

$$\{h : \{0,1\}^r \times D_n \longrightarrow D_n\}$$

o bien,

$$\{(a_0, a_1 \cdots a_{r-1}, x) \longrightarrow f_{a_0}(f_{a_1} \cdots (f_{a_{r-1}}(x)) \cdots)\}$$

y puede escribirse como:

$$h(a,x) = |4^a \cdot x^{2^r}| \pmod{n},$$

donde $a = (a_o, a_1 \cdots a_{r-1})$ es interpretado como el entero $a_{r-1} \cdot 2^{r-1} + \ldots + a_1 \cdot 2 + a_0$.

La función inversa $h^{-1}(a,\cdot)$ se leerá como $(h(a,\cdot))^{-1}$, tal que $h^{-1}(h(a,x)) = x$.

La cadena binaria a, se codificará, usando un mapeo **libre de prefijo** $< \cdot >$.

Definición 5.3.1 *Un mapeo es libre de prefijo, en el sentido que* $< a_1, a_2, \cdots, a_n >$ *nunca es un prefijo de* $< b_1, b_2, \cdots, b_n >$ *a menos que* $n = m$ *y* $a_1 = b_1, \cdots, a_n = b_n$.

El mapeo $< \cdot >$, debe ser un mapeo uno a uno, que va de cadenas binarias de bits a cadenas binarias.

Ejemplo 5.3.1.1 *Considerese siguiente plan de codificación, para la n-ada de cadenas binarias* $a_1, a_2, \cdots, a_n$. *Cada* a_i *es codificado al cambiar cada 0 por 00 y cada 1 por 11, agregando al final 01. La codificacion de* $a_1, a_2, \cdots a_n$ *será concatenada y seguida por 10. Así,*

Si $a_1 = 111001$, $a_2 = 101001$, $a_3 = 100001$,
$a_1a_2a_3 = 111001101001100001$,
por tanto,
$a_1 = 111001 \longrightarrow 11111100001101$
$a_2 = 101001 \longrightarrow 11001100001101$
$a_3 = 100001 \longrightarrow 11000000001101$
Por tanto,

$< a_1a_2a_3 >= 1111110000110111001100001101110000000110110$

◊

Encontrar una colisión sobre la función h , es decir un par $((a,x),(b,y))$ con $h(a,x) = h(a,y)$ y $(a,x) \neq (b,y)$, es tan difícil

como encontrar un f-pinza[6].

El siguiente lema demuesta escencialmente que si f_0 y f_1, son un par de permutaciones, entonces es difícil encontrar dos diferentes cadenas binarias a y b y dos elementos x y y tal que $h(a,x) = h(b,y)$, es decir, una colisión sobre h.

Lema 5.3.2 *Sean f_0 y f_1 un par de permutaciones con la propiedad libre de pinza[7], sean x, $y \in D_n$ y a, b dos diferentes cadenas binarias tales que existe un $z \in D_n$, tal que $z = h(a,x) = h(b,y)$. Entonces existe una f-pinza, es decir, (x_1, x_2, x_3) donde $x_3 = h^{-1}(c,z)$ para algún prefijo c de $< a >$.*

◇

Demostración. Sea $c \in \{0,1\}^*$ el prefijo común más grande de $< a >$ y $< b >$. Tal c debe de existir dado que $< \cdot >$ es una codificación libre de prefijo. Por tanto, asignando $x_3 \leftarrow h^{-1}(c,z)$, $x_1 \leftarrow h^{-1}(c0,z)$, y $x_2 \leftarrow h^{-1}(c1,z)$, si obtiene una f-pinza, porque h es una permutación biyectiva. (si c es una cadena vacía entonces $h^{-1}(c,\cdot)$ denota la función identidad, tal que $x_3 = z$, $x_1 = h^{-1}(0,z)$ y $x_2 = h^{-1}(1,z)$). Por tanto, esto es una contradicción de las propiedades de las permutaciones libres de pinza. Y también es una contradicción a la dificultad del problema de la factorización, ya que encontrar una f-pinza es tan difícil como resolver el problema de la factorización.

■

Considerando esta función h, construiremos una familia de homomorfismos fibrados.

Construcción 5.1.4

Construcción de homomorfismo fibrado para el plan basado en el PFE[8].

[6] Vid definición 2.1.9.
[7] Vid definición 2.1.9.
[8] Vid sección 4.2.3.

- Generación de llaves: g. Con la entrada de k y τ, selecciona un entero de la forma $n = pq$ de longitud de k-bits, donde p y q son primos tal que, $p \equiv 3 \pmod 8$ y $q \equiv 7 \pmod 8$, tendrá una salida de $K := (\tau, n)$.

- Familias de grupos: Los grupos para la llave, $K = (\tau, n)$ son:

$$H_K := (\pm QRn)/\{1, -1\} \quad \text{y} \quad G_K := \mathbb{Z}_{2^\tau} \times H_K.$$

La razón de usar el grupo factor H_n en lugar de QR_n es porque los miembros de H_n pueden ser probados eficientemente: Un número entre 0 y $n/2$ pertenece H_n si y sólo si el símbolo del Jacobi es $+1$. (Por tanto es también fácil probar los miembros en G_K).

Las operaciones en $G_{\tau,n}$ estan definidas por

$$(a, x) \cdot (b, y) := ((a + b) \pmod{2^\tau}, xy4^{(a+b)\ div\ 2^\tau}),$$

(la operación $(a + b)\ div\ 2^\tau$, indica el cociente al dividir $a + b$ entre 2^τ) y el elemento unidad de G_K es $(0, 1)$.

- El homomorfismo, $h_K : G_{\tau,n} \to H_n$, está definido por,

$$h_K(a, x) := \pm(4^a x^{2^\tau}) \pmod n$$

Se tomará, cualquiera de los dos valores, $4^a x^{2^\tau}$ o $-4^a x^{2^\tau}$, dependiendo de cual de ellos es más pequeño que $n/2$.

El siguiente teorema muestra que esta construcción es una familia de homomorfismos.

Teorema 5.3.1 *Bajo la suposición del problema de la factorización, la construcción 3.1.4, es una familia de homomorfimos.*

◇

Demostración.

Lo que se necesita probar son los incisos (a) (b) y (c) de la definición 5.1.1.

a) Se probará que la función h_K es un morfismo de G_K a H_K. Para todos los números $a, b \in \{0,1\}^+$ y $x, y \in E$ se tiene:

$$\begin{aligned}
|h((a,x)*(b,y))| &= |h(a+b \pmod{2^\tau}, x \cdot y \cdot 4^{(a+b)div2^\tau}|) \\
&= |4^{a+b \pmod{2^\tau}} \cdot (x \cdot y \cdot 4^{(a+b)div2^\tau})^{2^\tau}| \\
&= |4^{a+b \pmod{2^\tau}+\, 2^\tau \cdot ((a+b)div2^\tau)} \cdot (x \cdot y)^{2^\tau}| \\
&= |4^{a+b} \cdot (x \cdot y)^{2^\tau}| \\
&= |\,|4^a \cdot x^{2^\tau}| \cdot |4^b \cdot y^{2^\tau}|\,| \\
&= |\,h(a,x) \cdot h(b,y)\,|
\end{aligned}$$

b) Se probará cada elemento $z \in im(h)$ tiene al menos 2^τ preimagenes. Dado que h es un homomorfismo, cada $z \in im(h)$ tienen el mismo número de preimagenes. Esto es

$$|h^{-1}(z)| = \frac{|\{0,1\}^\tau \times G|}{|im(h)|} = 2^\tau \cdot \frac{G}{|im(h)|} \geq 2^\tau.$$

c) La demostración a la resistencia a colisiones es el Lema 5.3.2

Por tanto esta familia es realmente una familia de homomorfismos fibrados.

■

Bajo esta familia de homomorfismos fibrados, se construye un plan de firmas intrínsecamente protegidas , llamado Plan sobre Factorización.

Construcción 5.1.5

Plan de FIP basado en el PFE bajo la construcción 5.1.4

- Generación de la llave: Sobre los parámetros k y σ. Aquí, el parámetro τ que determina el grado del homomorfismo está en función de σ.

- Generación de la Prellave: El centro genera n con g, al elegir p y q. Así $prellave = (k, \sigma, n)$.
- Verificación de la prellave: El firmante probará que n es un entero Blum generalizado.
- Generación principal de la llave: El firmante selecciona $sk := (sk_1, sk_2) = ((a, x), (b, y)) \in G_{\tau,n}$, como llave secreta y calcula la llave pública $pk := (pk_1,\ pk_2) = (\pm 4^a x^{2^\tau} \pmod n,\ \pm 4^b y^{2^\tau} \pmod n)$.

- El espacio de mensajes es: $\{0, \ldots, 2^\rho - 1\}$, para cualquier $\rho \in \mathbb{N}$.
- El parámetro τ que define el grado del homomorfismo, es calculado por $\tau := \sigma + \rho$.
- El firmado: La firma, $firma(sk, m) = sk_1 \cdot sk^m = (a, x) \cdot (b, y)^m$ para mensajes m del subconjunto de $\{0, \ldots, 2^\rho - 1\}$.
- Prueba: $Prueba(pk, m, s) = aceptar \Leftrightarrow pk_1 \cdot pk_2^m = h(s)$.
- Prueba de falsificación: Dada una firma falsa sf sobre un mensaje m^* , el firmante calculará $s = firma(sk, m^*)$ y si $s \neq sf$, entonces presentará el par (s, sf) como prueba de la falsificación.
- Verificación de la demostración de falsificación: La verificación se hace al probar que dos elementos en G_K son mapeados a un mismo valor de H_K.

Teorema 5.3.2 *El Plan sobre la Factorización es seguro para el firmante.* ◇

Demostración.

De acuerdo al Corolario del Teorema 5.1.2, lo único que resta por demostrar es $|T_{m'}| \leq 2^\rho$, como la prellave es buena, entonces

$$|T_{m'}|/2^\tau \leq 2^{-\sigma}$$

Para esto notese que:

$$(a,x)^{m'} = (0,1) \Rightarrow m'a \pmod{2^\tau} = 0 \Rightarrow ord(a)|m'$$

Por tanto

$$T_{m'} \subseteq \{(a,x) \in G_{\tau,n} | h_{\tau,n}((a,x)) = 1 \ y \ ord(a)|m'\}.$$

Como h es un homomorfismo, para cada a existe exactamente un x tal que $h_{\tau,n}((a,x)) = 1$. Por tanto.

$$|T_{m'}| \leq |\{a \in \ \mathbb{Z}_{2^\tau} \mid ord(a)/m'\} = mcd(2^\tau, m').$$

Por la elección de el espacio de mensajes, m' está entre 1 y $2^\rho - 1$ (m' es la diferencia entre dos mensajes diferentes) y por tanto $mcd(2^\tau, m') < 2^\rho$.

■

Una operación del grupo en G_K es realmente una multiplicación modular porque el exponente de 4 es 0 o 1.

Por tanto, los dos planes requieren al menos de la misma cantidad de operaciones y almacenamiento para las llaves y firmas. La principal diferencia es que la generación de la llave en el primer plan es mas simple. La siguiente tabla, hace una comparación de los dos planes. Para $k = \rho = |q|_2 = |p|_2$ acordando que la longitud del mensaje es $|k|_2$.

Complejidad de los dos planes basados en la construcción general

Complejidad	Logaritmo Discreto	Factorización
firma	2 multiplicaciones	≈k multiplicaciones
prueba	< 2k multiplicaciones	< 2k + σ multiplicaciones
longitud de PK	2k	2k
longitud de SK	4k	4k + 2σ
longitud de la firma	2k	2k + σ

Capítulo 6

Plan FIP-n mensajes

6.1 Generalización del plan basado en el PLD

En el capítulo 5 se presentó un plan de firmas intrínsecamente protegido en el cual se puede firmar únicamente un mensaje. En este capítulo se generalizará la construcción 5.1.3 basada en el PLD para firmar un número $N \in \mathbb{N}$ de mensajes.

Si se desea ver con más detalle la información presentada en este capítulo consulte [PePf].

La manera más fácil que se podría pensar para firmar más de un mensaje, es usar un plan como el que se describió en el Capítulo 5 y preparar tantas llaves como mensajes se quieran firmar.

Sin embargo ésto no es muy práctico. En particular, se asume que cada firmante tiene acceso a un canal de transmisión confiable para la distribución de las llaves públicas (esto puede ser realizado por una supervisión de alguien de confianza o bien por una agenda telefónica). El uso de este canal debe ser minimizado. De hecho, algunos autores de planes ordinarios de firmas digitales requieren que después de la generación principal de la llaves que tiene longitud fija, el proceso de firmado sea ejecutado polinomialmente.

En seguida se presentará una variante de la construcción, basada

en el logaritmo discreto, esta construcción acorta la llave secreta aproximadamente por un factor de 2.

En la construcción de FIP para firmar N mensajes la cantidad total de llaves secretas que se necesitan almacenar es $(2N+2)k$.

La siguiente construcción se basará en familias de homomorfismos fibrados (construcción (5.1.2) sobre el logaritmo discreto, tomando todos sus parámetros y características. Por tanto también estará basada en la construcción 5.1.3 de planes de FIP sobre el PLD.

Construcción 6.1.3 Plan de FIP Generalizado basado en el PLD.

- Generación de la llave: (idéntica a la construcción 5.1.3)
  - Se elegirá una prellave (a, b, p, q) y será verificada, tal que p y q son primos, y $p/(q-1)$, donde a y b son generadores de el grupo H_q, que es el único subgrupo de $\mathbb{Z}_p^*$ de orden q.
  - El firmante seleccionará 2N+2 llaves secretas,

    $$sk := ((x_1, y_1), \ldots, (x_{N+1}, y_{N+1}))$$

    que son números entre 0 y $q-1$. La llave pública, correspondiente es,

    $$pk := (pk_1, \ldots pk_{N+1}) = (a^{x_1} b^{y_1}, \ldots, a^{x_{N+1}} b^{y_{N+1}})$$

- El espacio de mensajes está dado en $\mathbb{Z}_q^*$
- El firmado: Una firma correcta sobre el j-esimo menaje, m_j, es una tercia,

$$\begin{aligned} s &= (j, s_{1,j}, s_{2,j}) \\ &:= (j, x_j + m_j x_{j+1}, y_j + m_j y_{j+1}) \end{aligned}$$

- La prueba: Una tercia $(j, s_{1,j}, s_{2,j})$ es una firma aceptable sobre el j-esimo mensaje, m_j si y sólo si

$$a^{s_{1,j}} b^{s_{2,j}} = pk_j pk_{j+1}^{m_j}$$

- La demostración y verificación de falsificación está dada en la Construcción 5.1.3.

Teorema 6.1.1 *Bajo la suposición del LD, la construcción 6.1.3, describe un plan de FIP seguro.*

Demostración

Si un mensaje es firmado correctamente, la firma debe pasar la prueba, ya que las firmas individuales son identicas a las firmas de la Construcción 5.1.3. Similarmente, la seguridad para los *receptores* se satisface, porque las demostraciones de falsificación son idénticas a la Construcción 5.1.3. Además, cada falsificación exitosa, f, que no es una firma correcta, es decir, $f = (m_j^*, (j, s_{1,j}, s_{2,j}))$ con $(j, s_{1,j}, s_{2,j}) \neq firma(sk, j, m_j^*)$, puede ser probada como antes. Esto demuestra que el uso de la mayoría de las llaves secretas no facilitan al falsificador poder adivinar una firma correcta.

Basta con demostrar el peor caso, en donde el firmante ya ha hecho N firmas sobre los mensajes $m_1, \ldots m_N$, considerando que el falsificador puede tener acceso a todas las firmas junto con los mensajes, y por tanto el falsificador a maximizado la información que le puede ayudar. Para esto, primero se determinará el tamaño de el conjunto $SK_{\bar{C}}$ de llaves secretas posibles, dadas por estas firmas y llaves públicas. $SK_{\bar{C}}$, es el conjunto de soluciones para la ecuaciones definidas por la llave pública y las firmas. Si tomamos a $e := \log_a(b)$, todas estas ecuaciones pueden ser escritas como ecuaciones lineales sobre el campo $GF(q)$. Estas ecuaciones son descritas por una matriz A, de $(3N + 1) \times (2N + 2)$. Las columnas corresponden a los elementos de sk. Los primeros $2N$ renglones corresponden a las firmas, y los siguientes $N + 1$ corresponden a la llave pública.

$$A = \begin{pmatrix}
1 & 0 & m_1 & & & & & & & \\
 & 1 & 0 & m_1 & & & & & & \\
 & & 1 & 0 & m_2 & & & & & \\
 & & & 1 & 0 & m_2 & & & & \\
 & & & & \cdot & & & & & \\
 & & & & & \cdot & & & & \\
 & & & & & & \cdot & & & \\
 & & & & & & & 1 & 0 & m_k \\
 & & & & & & & & 1 & 0 \\
1 & e & & & & & & & & \\
 & & 1 & e & & & & & & \\
 & & & & \cdot & & & & & \\
 & & & & & \cdot & & & & \\
 & & & & & & \cdot & & & \\
 & & & & & & & & 1 & e
\end{pmatrix}$$

El rango de A es $2N + 1$. Para esto, considerese la siguiente submatriz de A,

$$\begin{pmatrix}
1 & 0 & m_j & 0 \\
0 & 1 & 0 & m_j \\
1 & e & 0 & 0 \\
0 & 0 & 1 & e
\end{pmatrix}$$

Esta submatriz tiene rango 3, donde el tercer renglón puede ser borrado. Repitiendo este proceso para $j = 1, \ldots N$, se borrarán todos los renglones de la forma $(0, \ldots, 1, e, o, \ldots, 0)$ excepto el último. Por tanto, se tiene como resultado una matriz $\overline{A}$ de $2N + 1$ renglones,

$$\overline{A} = \begin{pmatrix} 1 & 0 & m_1 & & & & & & & \\ & 1 & 0 & m_1 & & & & & & \\ & & 1 & 0 & m_2 & & & & & \\ & & & 1 & 0 & m_2 & & & & \\ & & & & \cdot & & & & & \\ & & & & & \cdot & & & & \\ & & & & & & \cdot & & & \\ & & & & & & & 1 & 0 & m_N \\ & & & & & & & & 1 & 0 \\ & & & & & & & & & 1 \end{pmatrix} \quad \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ m_N \\ e \end{matrix}$$

Por tanto $|SK_{\bar{C}}| = q$. Se demostrará ahora que cada una de estas q posibles llaves secretas, tiene como resultado una firma correcta diferente sobre un nuevo mensaje m_j^*. Para probar esto, basta con demostrar que cualquier firma adicional determinará solamente una llave secreta. La firma adicional introducirá dos nuevas ecuaciones que corresponden a los renglones,

$$0 \ldots 0\ 1\ 0\ m_j^*\ 0\ 0 \ldots 0$$
$$0 \ldots 0\ 0\ 1\ 0\ m_j^*\ 0 \ldots 0$$

Haciendo una cuantas operaciones con los renglones de esta matriz se puede ver que el rango de esta nueva matriz es de $2N + 2$. (Recuerdese que $m_j \neq 0\ \forall j$).

Por lo tanto una firma falsificada es correcta solamente para una de las q posibles llaves secretas.

■

Capítulo 7

Aplicación de las FIP

7.1 Firmas intrínsecamente protegidas en sistemas de pagos

En este Capítulo se presentará una aplicación de las firmas intrínsecamente protegidas en un sistema de pagos digitales y se describirán los pasos para una transmisión confiable.

La información presentada en este capítulo puede verse en [DiHe], [Pf91] y [PfWa].

A las firmas intrínsecamente protegidas se les ha dotado de ventajas adicionales en aplicaciones públicas sobre sistemas de pagos: Es realmente dudoso que una persona pueda falsificar firmas, aun cuando tome parte en el sistema. Recordando que este sistema se basa en suposiciones criptográficas y además los parámetros de seguridad no son elegidos por el firmante.

Utilizando un plan de firmas intrínsecamente protegido y un plan de firmas convencionales juntos[1], se puede hacer que los clientes de tal sistema sean incondicionalmente seguros. El hecho de que los clientes tengan un plan de firmas intrínsecamente protegido y considerando que la suposición criptográfica es cierta, hace que los clientes pueden reclamar irrefutablemente de que sus firmas han sido falsificadas en un caso dado, lo cual puede causar una considerable incertidumble pública.

[1] Vid sección 7.2.

La siguiente construcción de firmas está basada en la idea de [DiHe]. Estas firmas son llamadas una-vez.

Considérese un homomorfismo fibrado h_K, (disponible públicamente) que cumpla los tres requsitos de la definición 5.1.1. Este homomorfismo puede tener como problema subyacente al PLD o el PFE. Por tanto la llave K dependerá de la suposición criptográfica determinada.

Cosntrucción 6.1.1.

- El firmante elegirá $2m, m \in \mathbb{N}$ elementos del dominio de h_k, como llave secreta.

$$SK = ((r_{1,0}, r_{1,1}), (r_{2,0}, r_{2,1}), \ldots, (r_{m,0}, r_{m,1})).$$

- La llave pública correspondiente será calculada como las imágenes de la llave secreta, considerando tambien a la llave K. Esto es,

$$PK = (K, (pk_{1,0}, pk_{1,1}), (pk_{2,0}, pk_{2,1}), \ldots, (pk_{m,0}, pk_{m,1})).$$

  Donde $pk_{i,b} := h_K(r_{i,b})$, $b \in \{0,1\}$ y $i = 1, 2, \ldots, m$.

- El espacio de mensajes es una cadena binaria de longitud m.

- La firma para el *i-esimo* bit, $b \in \{0,1\}$, será $s := r_{i,b}$, esto es, si $b = 0$, $s := r_{i,0}$ y si $b = 1$, $s := r_{i,1}$.

- El receptor de una tercia (i, b, s) aceptará la firma s si $h_K(s) = pk_{i,b}$.

- La demostración de falsificación es inmediata al mostrar un par de firmas aceptables (s, s') sobre un mismo mensaje.

- La verficación de una demostración se acepta si $h_K(s) = h_K(s')$.

Este plan de firma es seguro, si las propiedades del homomorfismo fibrado se cumplen.

En esta construcción la llave pública PK no permitirá a un adversario obtener información acerca de las al menos 2^{σ} preimagenes que el firmante a elegido originalmente, es decir los $r_{i,b}$'s, que son utilizados para el cálculo de los $pk_{i,b}$'s. Para falsificar una firma el adversario deberá calcular una preimagen para un $pk_{i,b}$, considerando que la firma no a sido publicada todavía, es decir, el valor $r_{i,b}$ (ya que el firmante tiene cualquiera de las dos entradas para *i-esimo* bit). Por lo tanto, la probabilidad de que un falsificador elija una $r^{*}_{i,b} \neq r_{i,b}$, es al menos $1 - 2^{-\sigma}$.

7.2 Aplicaciones

7.2.1 Protocolo en un sistemas de pagos

Una de las aplicaciones importantes que pueden tener las firmas intrínsecamente protegidas es en los sistemas de pagos digitales, tales sistemas se pueden construir de manera que los clientes sean incondicionalmente seguros. Esto también se puede aplicar en organizaciones que tienen muchos clientes y en donde las firmas son únicamente intercambiadas entre la organización y el cliente, más no de cliente a cliente. Otra vez un ejemplo muy práctico para esta aplicación es en un banco.

Uso de las FIP

Supongase un sistema digital de pagos simple, en el cual se hacen depósitos, retiros y envíos de una cuenta a otra. Para esto se necesita una implementación de tal forma que los clientes sean incondicionalmente seguros.

Para tener la seguridad de que se satisfagan estos requerimientos, se puede usar la siguiente estrategia: para los clientes se utiliza un plan de firmas intrínsecamente protegido y el banco puede usar un plan de firmas convencionales, (por ejemplo GMR). Para evitar que alguien que no es cliente trate de pasarse por un cliente, cada

envio del cliente al banco será confirmado a éste, por una firma del banco.

Esto implica que el banco es el único portador del riesgo sobre las firmas intrínsecamente protegidas de los clientes, por tanto el banco debe escoger la prellave K, para las funciones fibradas. Una vez que el banco a publicado la prellave K, cada cliente puede escoger su llave secreta SK (es decir, las llaves secretas son un conjunto del dominio de h), y en seguida calculará la llave pública PK con estas llaves secretas al aplicar la función fibrada. De donde tenemos que cualquiera puede probar la validez de las firmas, en especial el juez y también cualquiera puede verificar una demostración de falsificación. Con todo esto se a dotado a los clientes una seguridad incondicional contra falsificaciones y el banco es criptográficamente seguro de que los clientes no puedan construir demostraciones de falsificación sobre sus propias firmas, porque él mismo es quien elige la prellave. Se puede pensar que el banco es capaz de calcular demostraciones de falsificación correspondientes a K, pero esto no le es conveniente pues el único dañado es el mismo, dado que las órdenes de envios y retiros pueden resultar inválidas, aunque las confirmaciones de envios y depósitos puedan ser válidas.

Los clientes son incondicionalmente seguros de las firmas que reciben del banco, que confirma una orden de retiro o depósito, ya que cada firma que recibe el cliente es válida para siempre y el banco es criptográficamente seguro contra falsificaciones.

7.2.2 Ventajas

Se puede pensar en distribuir el riesgo entre los clientes y el banco. Antes de seguir amortiguando esta idea es claro que el banco es quien selecciona su plan de firmas y sus parámetros de seguridad, y el banco tiene la capacidad de hacerse tan seguro como quiera, es decir, no se tiene la necesidad de distribuir el riesgo. Así que tan pronto como una falsificación ocurra, el banco es seguro, ya que él mismo se notificará de que sus firmas han sido falsificadas y para las falsificaciones de las firmas de los clientes se debe presentar una demostración de falsificación. Además tememos que en un sistema

de pagos puede ser ventajoso no únicamente para los clientes, sino también para el banco, ya que si el banco tiene argumentos de que hay una seguridad incondicional, ésto le puede atraer más clientes.

Una razón por la que un sistema de pagos digitales pueda tener un inconveniente, es que el banco debe publicar su prellave K y ésta sea sometida ante adversarios para tratar de resolver el secreto que en ella existe, pero esto no es un problema ya que se puede usar un plan como el de la sección 5.1.4, que se base en la dificultad de la factorización, en donde es necesaria una prueba que verifique que la prellave K cumple con los requisito necesarios: Así, esta publicación puede ser fácilmente incorporada a los protocolos anteriores ya que en el mismo momento, el banco puede publicar otros datos, por ejemplo, el plan de firmas usado, los parámetros de seguridad y su llave pública PK, etc. Por tanto todos los clientes pueden probar si K cumple los requisítos antes de que ellos seleccionen su llave secreta SK y su correspondiente llave pública PK.

7.2.3 Protocolos para firmar mensajes de 1-bit

Sistema normal de pagos:

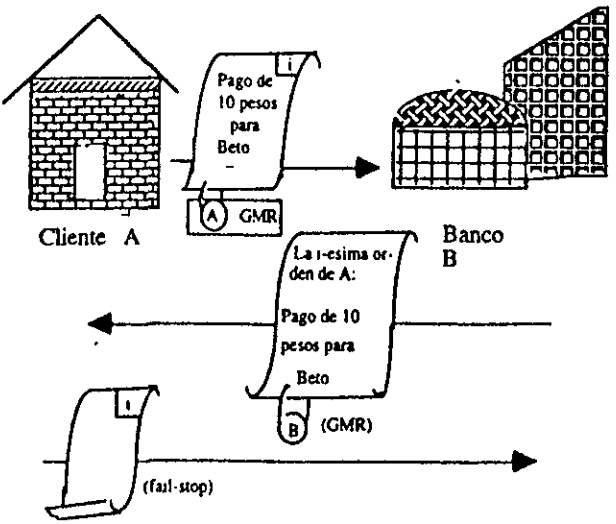
Se puede lograr que los clientes firmen mensajes de solamente 1-bit. De esta forma se obtendrán sistemas de pagos que serán definitivamente prácticos.

1. En primer lugar el cliente le dirá al banco su *i-ésima* orden. En realidad este mensaje no necesita ser firmado. De cualquier forma se requerirá una firma para resguardarse contra falsificaciones (ya que puede haber malicia de que alguien envie un mensaje en nombre de otro).

2. El banco firmará de recibido, para esta orden, junto con el nombre o número de cuenta del cliente, así como el número i usado en el plan de firmas convencionales.

3. Entonces el cliente firmará un bit usando su *i-ésima* firma, enviando la preimagen r_i al banco. Esto es definido que la *i-ésima* orden es aceptada por el banco, es correcta.

El siguiente esquema da una visión más clara de esto.



Esquema que representa una orden en el sitema de pagos con mensajes de longitud de 1-bit; Las firmas representan la garantía.

Si el banco recibe esta firma, entonces ejecutará la orden. Si posteriormente el cliente reclama que el banco a ejecutado la *i-ésima* orden incorrectamente, un juez decidirá lo siguiente: En primera instancia el banco debe probar que fue permitida la ejecución de la *i-ésima* orden, mostrando la firma del cliente de el paso 3. Si el banco puede hacer esto y el cliente no puede presentar una demostración de falsificación, el cliente debe mostrará la *i-ésima* orden firmada por el banco (paso 2). Esto cuenta como una orden válida.

Como en la sección 7.2.2, los clientes son incondicionalmente seguros y el banco es criptograficamente seguro: ya que los clientes únicamente puede firmar un bit (paso 3) después de que el banco haya firmado la orden correcta en el paso 2. El firmante es incondicionalmente seguro contra falsificaciones sobre su propia firma y la firma recibida. Un cliente puede engañar al banco de dos formas, por el cálculo de la demostración sobre una firma propia o por falsificar la firma del banco en una orden incorrecta. Ninguna de estas dos cosas es posible bajo una suposición criptográfica.

Capítulo 8

Conclusiones

El tema central de este trabajo son a las firmas digitales intrínsecamente protegidas. Se presentó sus características y aplicaciones.; así como su semejanza con los planes de firmas digitales convencionales.

Un plan de firmas intrínsecamente protegido comprende todas las características de los planes de firmas convencionales; de hecho un plan de FIP puede ser modificado con facilidad, para que opere como un plan convencional.

Un plan intrínsecamente protegido hace que el firmante tenga la capacidad de construir demostraciones de falsificación de manera incondicional y permite también cuantificar la probabilidad de que falle una demostración de falsificación. La implementación de un sistema con dichas propiedades satisface adecuadamente las necesidades típicas de transmisión de información segura en el ámbito tecnológico actual.

Los ejemplos presentados en esta tesis están basados en las suposiciones criptográficas más comunes, esto es, tanto el problema del logaritmo discreto como el problema de factorización, que han sido exaustivamente estudiados.

Como en el caso de las firmas digitales convencionales, la suposición de estos problemas criptográficos sustenta la seguridad para el receptor de las firmas; por lo que los receptores son tan seguros como los de un plan de firmas convencional.

En general se asume que tanto la seguridad del firmante como la seguridad del receptor, se basa en el hecho de que los falsificadores tienen un poder computacional polinomial, esto es, solamente pueden ejecutar algoritmos con tiempos de ejecución polinomial.

La definición general de las firmas intrínsecamente protegidas, se basa en familias de homomorfismos fibrados. Realmente toda la estructura de un plan de firmas intrínsecamente protegido, se soporta en los homomorfismos fibrados. Así las características de seguridad en general son aportadas por ellos. El hecho de que un homomorfismo fibrado tenga la propiedad de que a cada imagen le corresponde una buena cantidad de preimagenes, hace que las llaves públicas no den información suficiente sobre las llaves secretas, que permita calcular una falsificación sobre algun mensaje. La otra propiedad que tiene los homomorfismos fibrados, es que son resistentes a colisiones. Esta propiedad tiene como problema subyacente una suposición criptográfica. Si un falsificador llega a construir una falsificación, el supuesto firmante podrá demostrarla con una gran probabilidad, ya que presentará necesariamente una colisión sobre el homomorfismo, lo que implica que la suposición criptográfica sea resuelta. Así, encontrar una colisión sobre el homomorfismo implica resolver particularmente el problema del logaritmo discreto o bien el de la factorización.

El plan basado en el problema de la factorización, es algo mas laborioso a el plan basado en el problema del logaritmo discreto, ya que la construcción de los homomorfismo fibrados se basa en parejas de permutaciones, que tienen la propiedad de ser libres de pinza. En este plan la demostración de falsificación está dada al presentar una colisión sobre el homomorfismo y esta última a su vez, implica calcular una f-pinza, y sucesivamente encontrar los factores primos de nl lo cual contradice la suposición original a cerca del problema de la factorización.

Una pregunta que es conveniente hacer, es: ¿cuántos mensajes

se pueden firmar con la misma llave secreta?.

Algo que no es factible, de aplicación práctica es generar tantas llaves como mensajes se quiera firmar.

El plan de firmas FIP-n mensajes está diseñado para firmar n mensajes. Este plan tiene la característica de reducir el tamaño de la llave secreta a la mitad, además es una generalización del plan presentado en el capítulo anterior, por tanto, también está estructurado bajo las propiedades de los homomorfismos fibrados.

Los sistemas de pagos digitales son una de las aplicaciones de los planes de firmas digitales intrínsecamente protegidos.

El protocolo presentado para la aplicación de las FIP, es llevado a cabo entre un banco y su cliente: el cliente firmará los mensajes con un plan de firmas intrínsecamente protegido y el banco firmará con un plan de firmas convencional, por lo que se hace al cliente incondicionalmente seguro, ya que su plan de firmas le permite hacer demostraciones de falsificación y además es receptor de un plan de firmas convencional.

La razón de que no se le dote al banco incondicionalidad es porque se considera computacionalmente fuerte, por tanto la seguridad proporcionada en este protocolo, es suficientemente fuerte como para que las implementaciones satisfagan las necesidades de seguridad para los usuarios.

Bibliografía

[AdHu87] Leonard M. Adleman, Mind-Deh A. Huang: Recognizing Primes In Random Polynomial Time; *19th Symposium on Theory of Computing (STOC) 1987, ACM New York 1987, 462-469.*

[AdPR83] Leonard M. Adleman, Carl Pomerance, R.S. Rumely: On distinguishing prime numbers from composite numbers; *Annals of Mathematic 117 (1983) 173-206.*

[AGLL95] Derek Atkins, Michael Graff, Arjen K. Lenstra, Paul C. Leyland: The Magic Words are Squeamish Ossifrage; *Asiacrypt '94, LNCS 917, Springer-Verlag, Berlin 1995, 263-277.*

[BaDG88] José Luis Balcázar, Josep Díaz, Joaquim Gabarró: Structural Complexity I; *EATCS Monographs on Theoretical Computer Science 11, Springer-Verlag, Berlin 1988.*

[BeVa93] Ethan Bernstein, Umesh Vazirani: Quantum Complexity Theory; *25 Symposium on Theory of Computing (STOC) 1993, ACM, New York 1993, 11-20*

[BGMW93] Ernest Brickell, Daniel M. Gordon, Kevin S. McCurley, David B. Wilson: Fast expnentiation with precomputation; *Eurocrypt '92, LNCS 658, Springer-Verlag, Berlin 1994, 56-60.*

[BlePfWa] Bleumer, Gerrit, Pfitzmann B. Waidner Michael: A remark on a signature scheme where forgery can be proved,

in Proc. 1996 Eurocrypt, Lecture Notes in Comput. Sci. 473, Springer-Verlag, Berlin, 441-445.

[Bleu90] Gerrit Bleumer: *Vertrauenswürdige Schlüssel für ein Signatursystem, dessen Brechen beweisbar ist; Studienarbeit, Institut Für Rechnerentwurf und Fehlertoleranz, Universität Karlsruhe 1990.*

[Blum82] Manuel Blum: Coin Flipping by Telephone, A Protocol for Solving Impossible Problems*; compcon spring 1982, 133-137.*

[BoCo90] Jurjen Bos, Mathijs Coster: Addition chain heuristics*; Crypto '89, LNCS 435, Springer-Verlag, Heidelberg 1990, 400-407.*

[Brow94] Julian Brown: A Quantum Revolution For Computing*; New Scientist 24/1944 (Sept. 1994) 21-24.*

[CoLe87] Henri Cohen, Arjen K. Lenstra: Implementation of a New Primality Test*; Mathematics of Computation 48/177 (1987) 103-121.*

[DaLP93] Ivan Damgård, Peter Landrock, Carl Pomerance: Average Case Error Estimates for the Strong Probable Prime Test*; Mathematics of Computation 61/203 (1993) 177-194.*

[DiHe] Diffie Whitfield y Hellman Martin E: New Directions in Cryptography*; IEEE Transaction on Information Theory, Vol. No. 6, Nov. 1976.*

[DoLe95] Bruece Dodson, Arjen K. Lenstra: NFS with Four Large Primes: An Explosive Experiment*; Crypto'95, LNCS 963, Springer-Verlag, Berlin 1995 372-385.*

[DSS91] *Announcing a Digital Signature Standard; Federal Information Processing Standards Publication (FIPS PUB XX), Draft, 19 de Agosto de 1991.*

[Fox91] Dirk Fox: Effiziente Softwareiplementierung asymmetrischer Kryptosysteme und der zugrundeliegenden modularen Langzahlarithmetik; *Diplomarbeit, Institut für Rechnerentwurf und Fehlertoleranz, Universität Carlsruhe, Abril de 1991.*

[EG84] T. El-Gamal: A public key cryptosystem and a Signature Scheme based on discrete logarithms, *Proc, Crypto 84, Springer-Verlag, New York, Heidelberg, Berlin, 1985, 10-18.*

[GoKi86] Shafi Goldwasser, Joe Kilian: Almost All Primes Can be Quickly Certified; *18th Symposium on Theory of Computing (STOC) 1986, ACM, New York 1986, 316-329.*

[Gord93a] Daniel M. Gordon: Discrete logarithms in GF(p) using the number field sieve; *SIAM Journal on Discrete Mathematics 6/1 (1993) 124-138.*

[GMR] Goldwasser, S., Micali, S., Rivest R: A Digital Signature Scheme Secure against adaptive chosen-message attacks, *SIAM J. Comput. 17, (1988), 281-308.*

[GMRac] Goldwasser, S. Micali, C. Rackoff: The knowledge Complexity of Interactive Proof Systems, *SIAM J. Comput., 18 (1989), 186-207.*

[GMY83] S.Goldwasser, S. Micali y Andy Yao: Strong Signature Schemes; *15th Symposium on Theory of Computing (STOC) 1983, ACM, New York, 1983, 431-439.*

[Guin91] Daniel Guinier: The Multiplication of Very Large Integers Using the Discrete Fast Fourier Transform; *ACM SIGSAC Review 9/3 (1991) 26-36.*

[HaWr79] G. H. Hardy, E. M. Wright: *An Introduction to the Theory of Number;* (5th, ed.). Oxford at the Clarendon Press, Oxford 1978.

[Her] Herstein, I. N.: *Algebra Moderna,* Trillas, (1990).

[HePe] Van Heyst, E. y Pedersen, T.P.: How to make efficient fail-stop signatures, *in Proc. 1992. Eurocrypt, Lectures notes in comput. Sci, 658, Springer-Verlag, Berlin, (1993) 366-377.*

[HePePf] Van Heyst, E., Pedersen, T.P. y Pfitzmann B.: New constructions of fail-stop sigantures and lower bounds, *in Proc. 1992 Crypto. Lecture Notes in Comput. Sci. 740, Springer-Verlag, Berlin (1993) 15-30.*

[Je] Jeroen van de Graaf: A Simple and Secure Way to show the Validity of Your Public Key, *in Proc. 1987, Crypto Lecture Notes in Comput. Sci. 293, Springer-Verlag, Berlin, 1988, 128-134.*

[Knut81] Donald E. Knuth: *The art of Computer Programming, Vol. 2: Seminumerical Algorithms;* (2nd ed.) Addison-Wesley, Reading 1981.

[Kram86] Evangelos Kranakis: *Primality and Cryptography,* Wiley-Teubner Series in Computer Science, B.G. Teubner, Stuttgart 1986.

[La] L. Lamport: Constructing Digital Signatures from a One-Way Function, *SRI Int. CSL-98 (Octubre 1979).*

[LaOd91] Brian A., Adrew M. Odlyzko: Computation of Discrete Logarithms in Prime fields*; Designs, Codes and Cryptography 1/1 (1991) 47-62.*

[LeLe90] A. K. Lenstra, H. W. Lenstra, Jr: *Algorithms in Number Theory; in: J. van Leeuwen (ed.):* Handbood of Theorical Computer Science; Elsevier, 1990, 673-715.

[Li81] K. Lieberherr: Uniform complexity and digital signatures, *Theoret. Computer. Sci., 16 (1981), 99-110.*

[Maur92] Ueli M. Maurer: Some Number-theoretic Conjectures and Their Relation to the Generation of Cryptographic Primes*; 2nd IMA Symposium on Cryptography and Coding, 1989; Oxford University Press, 1992, 173-1991.*

[Maur95] Ueli M. Maurer: Fast Generation of Prime Numbers and Secure Public-Key Cryptographic Parameters: *Journal of Cryptoloy 8/3 (1995) 123-155.*

[McCu90] Kevin S. McCurley: The Discrete Logarithm Problem; *in: Carl Pomerance (ed.); Cryptology and Computational Number Theory; Proceedings of Sympsia in Applied Mathematics, Vol. 42, American Mathematical Society, Providence 1990, 49-74.*

[MH78] R. Merkle y M. Hellman: Hiding information and signatures in trap-door knapsacks, *IEEE Trans. Inform. Theory, IT-24(1978), 525-530.*

[Mont85] Peter L. Montgomery: Modular Multiplication without trial division; *Mathematics of Computation 44 (1985) 519-512.*

[MOV] Menezes, Alfred J., van Oorschot, Paul C., y Vanstone, Scott A.: *Handbook of Applied Cryptography,* CRC Press, Inc (1996).

[OSS84a] H. Ong, C. Schnorr y A. Shamir: An efficient signature scheme based on quadratic equations, *Proc. 16th Anneal ACM Symposium on the Theory of Computing, Washington, D.C. Abril 1984, 208-217.*

[OS85] T Okamoto y S. Matyas: *Cryptography: A new dimension in data security,* John Wiley, New York, 1982.

[Pf91] Pfitzmann, B.: Fail-Stop signatures; Principles and Aplications, *Proc. 8th World Conference on Computer Security, Audit, and Control. Elsevier. Oxford, UK, (1991), 125-134.*

[Pf94] Pfitzmann, B.: Fail-Stop signatures Without Trees, *Hildesheimer Informatik-Berichte 16/94, ISSSN 0941-3014, Institut Für Informatik, Universität Hildesheim, Junio 1994.*

[Pf96] Pfitzmann, B.: Digital Signature Schemes: General Framework and Fail-stop Signature, *Lectures Notes in Comput. Sci. 1100, Springer-Verlag, Berlin, (1996).*

[PfWa90] Pfitzmann, B., Waidner, M.: Formal aspects of fail-stop signatures, *Technical Report 22/90, Facultät für Informatik, Universität Karlsruhe, Germany, 1990.*

[PfWa91] Pfitzmann, B., Waidner, M.: Fail-Stop Signatures and Aplications, *(Securicom 1991), Paris 1991, 145-160.*

[PePf] Pryds Pedersen, Torben y Pfitzman B.: Fail-stop Signatures, *SIAM J. COMPUT, Vol. 26, No. 2 (1997), 291-330.*

[QuCo82] Jean-Jacques Quisquater, C. Couvreur: Fast Decipherment Algorithms for RSA Public Key Cryptosystem; *Electronics Letters 18/21 (1982) 905-907.*

[Ra79] Michael O. Rabin: Digitalized signatures as intractable as factorization, *MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology Cambridge, MA, Enero 1979.*

[Rabin80] Michael O. Rabin: Probabilistic algorithm for primality testing; *J. Number Theory 12/(1980) 128-138.*

[RoSc62] J. B. Rosser, L. Schoenfield: Approximate Formulas for Some Functions of Prime Numbers; *Illinois J. Math. 6 (1962) 64-94.*

[RSA] R. Rivest, A. Shamir y L. Adleman: A method for obtaining digital signatures and public-key cryptosystems, *comm. ACM 21 (1978), pp. 120-126.*

[Schn91] Claus P. Schnorr: Efficient Signature Generation by Smart Cards; *Journal of Cryptology 4/3 (1991) 161-174.*

[ScSt71] Arnold Schönhage, Volker Strassen: Schnelle Multiplikation großer Zahlen; *Computing 7/(1971) 281-292.*

[sh] C. E. Shannon; Communication theory of secrecy systems, *Bell Sistem Tecnical J. 28 (1149) pags 656-715.*

[Sh78] A. Shamir: A fast signature scheme, *MIT, Laboratory for computer Science Technical Memo, MIT/LCS/TM-107, Massachusetts Institute of Technology, Cambridge, MA, Julio 1978.*

[Shor94] Peter W. Shor: Algcrithms for Quantum Computation: Discrete Logarithms and Factoring; *35th Symposium on Fundations of Computer Science (FOCS) 1994, IEEE Computer Society, 1994 124-134.*

[WaPf89] W. Michael y P. Birgit: Dining, Cryptographers in the Disco: Unconditional sender and recipient untraceability with computationaly secure serviceability, *Advances in Cryptology-EUROCRYPT'89, LNCS 434, Springer-Verlag, p. 690.*

[Wi80] H.C. Williams: A modification of the RSA public-key Cryptosystem, *IEEE Transaction on Information Theory, IT-26(1980), 726-729.*

Indice