UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Ciencias e Ingeniería – El Politécnico

¿Son seguros los accesos a través de Internet a los bancos del Ecuador?

Juan Alejandro Almendáriz Palacios

Tesis de Grado presentada previa la obtención

del título de Ingeniero en Electrónica

Quito

Diciembre de 2009

UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Ciencias e Ingeniería – El Politécnico

HOJA DE APROBACIÓN DE TESIS

¿Son seguros los accesos a través de Internet a los bancos del Ecuador?

Juan Alejandro Almendáriz Palacios

Julio Aráuz, Ph.D.	_____
Director de Tesis	Firma
Miembro del Comité de tesis	
René Játiva, Ph.D.	_____
Miembro del Comité de Tesis	Firma
Vinicio Carrera, Ph.D.	_____
Miembro del Comité de Tesis	Firma
Santiago Navarro, Ph.D.	_____
Director de Ingeniería Electrónica	Firma
Ing. Fernando Romo, MSc.	_____
Decano del Colegio Politécnico	Firma

Quito, diciembre de 2009

Dedicatoria

A mis padres y a mi hermana, pilares fundamentales de mi vida, quienes con su ejemplo, consejos diarios y compañía supieron guiarme por el camino del éxito teniendo siempre presentes la responsabilidad y la honestidad bajo los preceptos de Dios.

Agradecimiento

A todos mis maestros de la especialización, Julio Aráuz, Javier Dávila, Santiago Navarro, René Játiva, Bruce Honeisen, Laurent Sass, por su valioso aporte a mi formación académica y profesional a lo largo de estos años;
y especialmente a Julio por su inestimable contribución en el desarrollo de este proyecto de grado, dirigido desde Alemania y luego desde Estados Unidos.

A mis amigos y compañeros, por las clases, trabajos y momentos agradables compartidos, y por ser siempre un apoyo y un constante estímulo para mi auto superación.

Resumen

Estudios realizados a nivel mundial indican que la banca en línea se ha incrementado significativamente los últimos años; sin embargo, los robos electrónicos también han aumentado. Este trabajo estudia la seguridad de la banca en línea en el Ecuador. Para ello, se han construido modelos matemáticos de los bancos nacionales y se ha realizado una simulación de ataque de fuerza bruta a sus sistemas de acceso. Los resultados del experimento han mostrado las debilidades de los mecanismos de autenticación empleados actualmente y cuán vulnerables son a un eventual ataque. Importantes mejoras deben ser introducidas en la manera en que los bancos definen los nombres de usuario y crean las contraseñas para los clientes.

Abstract

Worldwide studies show online banking has increased significantly in recent years, however, electronic theft have also increased. This thesis studies online banking security in Ecuador. For that, mathematical models of national banks have been built and a simulation of brute force attack to their access systems has been constructed. Experimental results have shown the weaknesses of the authentication mechanisms currently used by some financial institutions in Ecuador and how vulnerable they are to potential attack. Significant improvements should be made in the way banks define usernames and create passwords for customers.

Tabla de Contenido

Lista de Figuras

Lista de Tablas

CAPÍTULO 1

I. INTRODUCCIÓN

El siglo XX fue el siglo de la revolución tecnológica[1], la misma que se mantiene hoy en día con pasos agigantados y cada vez más acelerados llegando incluso a cambiar y crear nuevos sistemas y formas de trabajo y modificando por completo nuestra forma de vida. Tal es la transformación que hemos sufrido en las últimas décadas que ya no hace falta esperar un par de meses hasta que nuestra carta llegue a su destino, no es necesario comprar el periódico para leerlo, ni hay que pagar altos costos por una llamada transatlántica, peor aún esperar llegar a casa para ocupar el teléfono. En pleno siglo XXI, prácticamente todo lo podemos hacer a la hora que queramos y desde cualquier lugar en el que nos encontremos, gracias a la magia del Internet y las comunicaciones inalámbricas.

Este cambio tecnológico ha alcanzado también a las instituciones bancarias desde hace algunos años. En la actualidad ya no es necesario esperar hasta las 8:00h del día siguiente para que las oficinas de nuestro banco estén abiertas y poder realizar una consulta, revisar nuestro saldo, transferir dinero a otra cuenta o solicitar un crédito. Tampoco debemos estar haciendo largas filas para lograr pagar los servicios básicos, cancelar la pensión del colegio o universidad, o esperar encontrar un lugar de venta de tarjetas prepago para acreditar saldo a nuestro celular. Todo esto y mucho más lo podemos hacer desde la comodidad de nuestra casa u oficina sentados frente a una *PC*; o en la calle, en el club, en el mall desde nuestro celular, *smartphone* o *PDA*.

El presente proyecto involucra, precisamente, un estudio de cuán seguros pueden ser los sistemas de acceso a las operaciones bancarias vía Internet. El estudio se enfoca

[1] Langer N. 2001. Revolución Tecnológica. Alfa-Redi: Revista de Derecho Informático. [Internet]. [Citado 2009, Octubre 19]. Disponible en http://www.alfa-redi.org/rdi-articulo.shtml? x=661.

particularmente al ámbito nacional y realiza un análisis de los mecanismos de autentificación vía *web* utilizados por algunos bancos en el Ecuador.

I.1 ANTECEDENTES

En el mundo entero cada día son más los bancos que implementan servicios en línea para sus clientes con la finalidad de ofrecer transacciones electrónicas rápidas y "seguras" desde cualquier lugar del mundo. La banca por Internet se vuelve más popular a medida que pasa el tiempo, ahora con unas cuantas operaciones se puede transferir fondos de una cuenta a otra, realizar pagos de matrículas universitarias, cancelar las cuentas de energía eléctrica, del consumo telefónico y de otros tantos servicios.

Por otra parte, a la par del crecimiento del número de instituciones financieras y/o bancarias que ofrecen servicios en línea para sus socios se encuentra el aumento de los robos y estafas por Internet. Cada año aparecen noticias de miles de cuentas a las que acceden extraños, quienes obtienen información confidencial[2]: seguridades violadas por expertos *hackers*. Presumiblemente cientos de miles de dólares en pérdidas para los bancos y sus clientes que creen que la seguridad de un sistema en línea está plenamente asegurada.

Las plataformas de banca electrónica son consideradas atractivas debido al sinnúmero de ventajas y funcionalidades que éstas ofrecen. Permiten al cliente acceder a sus cuentas, en tiempo real, y a un conjunto de productos y servicios a través de Internet. Proporcionan eficiencia, flexibilidad y control en las transacciones bancarias que se desee

[2] McMillan R. 2005. After theft, bank tightens online security. Network World. [Internet]. [Citado 2009, Octubre 19]. Disponible en http://www.networkworld.com/news/2005/053005bankofamerica.html

realizar. Un usuario puede realizar operaciones bancarias durante las 24 horas del día, 7 días a la semana, los 365 días del año, desde cualquier parte del mundo. Todo esto lo debemos a que el Internet se ha consolidado como el nuevo canal comercial y de prestación de servicios en el mundo entero.

I.2 BANCA 'EN LÍNEA' EN PLENO AUGE

La banca electrónica puede definirse como un sinnúmero de servicios bancarios que permiten que los socios o clientes de un banco tengan acceso a sus cuentas y puedan administrarlas electrónicamente vía *web*. La empresa española de seguridad informática S21sec, en su Informe sobre el Fraude Online 2005[3], la define como "un conjunto de herramientas que una entidad bancaria pone a disposición de sus clientes y que les permiten realizar operaciones bancarias a través de Internet".

A nivel nacional son muchos los bancos – principalmente los de mayor crecimiento – que han incorporado ya desde hace algunos años estos servicios en línea para sus clientes. Desafortunadamente, no existe un estudio estadístico del incremento de usuarios de banca electrónica en los últimos años, es más, la página *web* de la Superintendencia de Bancos ni siquiera incluye entre sus varios enlaces uno relacionado con el tema.

Cabe destacar, sin embargo, que de acuerdo a indicadores internacionales, la banca electrónica ha tenido un crecimiento exorbitante especialmente con el cambio de siglo. De acuerdo a Rob McGann 2005, en un estudio realizado en Estados Unidos a finales de 2004 y patrocinado por Pew Internet & American Life, la banca en línea ha sido la actividad de

[3] Informe sobre fraude online 2005. [Internet]. S21sec. [Citado en 2009, Octubre 19]. Disponible en https://cert.s21sec.com/index.php/es/documentos/func-finishdown/2/

más rápido crecimiento en Internet desde inicios de la década, con 53 millones de estadounidenses – el 44% de los usuarios de Internet en Estados Unidos – usando alguna forma de banca electrónica. Esto representa 3,5 veces más usuarios que los que utilizaban este servicio en el año 2000, período para el cual se tenía unos 14 millones de usuarios.

Este estudio también precisó que en un día típico, alrededor de 13 millones de americanos realizan operaciones bancarias en línea, lo que significa un 58 por ciento de incremento desde octubre de 2002. Todos estos datos, que tienen un margen de error de apenas el cinco por ciento, fueron obtenidos en noviembre de 2004 mediante encuestas telefónicas a usuarios frecuentes de Internet.

Por otra parte, si nos trasladamos a la banca española, los datos afirman que sólo en el año 2003, el número de usuarios *online* se incrementó en un 30% de acuerdo con García y Romero 2004 en su artículo "La expansión de la banca *online* en España". Estas cifras, junto con las estadounidenses, brindan un panorama del comportamiento global de estos indicadores.

Se podría, entonces, seguir detallando datos estadísticos de varias latitudes del mundo que nos muestren que el panorama es el mismo: el incremento de usuarios globales de banca electrónica es cada vez más acelerado. La banca *online* está en pleno auge y su crecimiento continuará aceleradamente en los próximo años y décadas.

I.3 JUSTIFICACIÓN E IMPORTANCIA DEL PROYECTO

Afirmar que un estudio sobre la seguridad de acceso a través de Internet a los bancos del Ecuador no tiene especial relevancia sonaría ilógico. Hasta ahora en el país no se conoce un estudio con las características del realizado y detallado en esta tesis. Ésta es la principal motivación del proyecto.

En la presente tesis se analizará los sistemas de acceso a través de la *web* a los portales de los bancos del Ecuador. Se realizará este análisis desde el punto de vista cuantitativo, se desarrollarán modelos matemáticos particulares para cada banco estudiado y se validará frente a ellos una simulación de un ataque a un sistema de acceso. A través del análisis cuantitativo se quiere determinar cuán seguros son y pueden ser los bancos del Ecuador que tienen acceso a sus portales vía Internet y además se procura identificar posibles fallas de seguridad y riesgos potenciales que existirían en sus sistemas.

También es importante la retroalimentación que se brinda en cuanto a los resultados obtenidos a partir del estudio. En este caso se desea plantear sugerencias y recomendaciones dirigidas a los administradores de los sistemas, autoridades bancarias, y a los usuarios de la banca en línea del país para que se tomen las precauciones debidas en el manejo y utilización de estos sistemas electrónicos. La información proporcionada por el estudio realizado sin duda será de mayor importancia para los administradores de sistemas de los bancos que son los encargados de implementar las seguridades en los mismos.

Todas las recomendaciones que se detallen tienen como propósito primordial el que se pueda mejorar los sistemas de acceso bancario, o en su defecto, mantener los niveles de seguridad verificados por el estudio. Muchos usuarios de banca electrónica en el país, por

no decir todos, desconocen cuáles son los verdaderos niveles de seguridad que le ofrece su banco el momento de llevar a cabo una transacción. Sin duda que los resultados que arroje este estudio permitirán que los clientes de los bancos conozcan qué tan factible o improbable es la realización de un fraude electrónico con sus cuentas.

I.4 OBJETIVOS Y METAS

El objetivo principal que se persigue con este proyecto es estudiar la seguridad de los sistemas de acceso a través de Internet de los principales bancos del Ecuador, el método que utilizan para autentificar un usuario y analizar la vulnerabilidad de sus seguridades. No se realizarán ataques a ningún sistema bancario, sino que se hará un estudio cuantitativo de la probabilidad de acceso a las cuentas de los usuarios electrónicos y se validarán los resultados de una simulación de los sistemas de autentificación e ingreso a las cuentas y servicios bancarios.

Finalmente se quiere generar resultados cuantitativos y recomendaciones sobre los sistemas de acceso a través de Internet de los bancos nacionales. Los resultados se mantendrán anónimos y no se nombrará a ningún banco en particular. Todo esto para alcanzar la meta principal que consiste en realizar una evaluación detallada de los sistemas en línea de ciertos bancos ecuatorianos.

I.5 CONTENIDO

Después de este capítulo introductorio, en el segundo capítulo se revisarán las bases teóricas de la seguridad en transacciones electrónicas, una breve descripción del tipo de certificados y mecanismos de seguridad empleados en ellas y un análisis de los ataques

más comunes realizados por los *hackers*. El Capítulo 3 aborda ampliamente una descripción de los sistemas de acceso a través de Internet a los bancos nacionales seleccionados, sus principales características en cuanto a los sistemas de autentificación. Luego, se presenta el estudio de la seguridad de los sistemas analizados y los modelos matemáticos que se han planteado para realizarlo. El Capítulo 4 se enfoca en la simulación realizada para la validación de los modelos matemáticos construidos para los bancos estudiados; el diseño del experimento y la descripción de las variables, niveles y factores utilizados en la implementación de simulación también se describen en este capítulo.

El Capítulo 5 resume los resultados obtenidos luego de la validación de los modelos matemáticos mediante la simulación. En el Capítulo 6 se muestran las conclusiones a las que se llega después de la realización del estudio y las recomendaciones planteadas a partir de los resultados. Finalmente, se detallan las diferentes fuentes de consulta utilizadas para la realización del proyecto y se adjuntan los programas utilizados en la simulación y otros documentos de interés.

CAPÍTULO 2

II. BASES TEÓRICAS

Actualmente los clientes de instituciones bancarias tienen un nuevo escenario en el que administran sus cuentas y tarjetas de crédito y en el cual efectúan un sinnúmero de operaciones financieras, ese escenario es el Internet. Las transacciones electrónicas que estos usuarios realizan involucran no sólo información secreta sino también dinero, por tanto, es indispensable que se lleven a cabo con mecanismos que garanticen la seguridad durante su ejecución.

II.1 UNA COMUNICACIÓN SEGURA

Cuando dos entidades – por ejemplo un cliente y su banco – quieren comunicarse de manera 'segura' es necesario que se cumplan varios requerimientos. Primero se requiere que el mensaje que envía el usuario pueda ser entendido sin dificultad por el sistema del banco y viceversa. Además el banco desea estar completamente seguro de que el mensaje enviado por el cliente es, efectivamente, remitido por él y no por otra persona; del otro lado, el cliente debe tener la seguridad de que la entidad con quien está estableciendo comunicación es, en efecto, su banco. Ambas entidades también deben estar seguras de que los contenidos de sus mensajes no sean alterados en la etapa de tránsito de su origen a su destino. Adicionalmente las dos partes deben asegurarse de que nadie les niegue el acceso a los recursos necesarios para establecer la comunicación. (Kurose y Ross 2005).

Cabe mencionar además que es deseable que la información no sea interceptada por extraños para así evitar posibles ataques pasivos. En la Figura 2.1 se muestra el

mecanismo utilizado en un ataque de este tipo, donde el intruso se coloca entre el cliente y el banco para espiar la comunicación, leer el contenido de los mensajes enviados y realizar – posiblemente – un análisis de tráfico para identificar patrones en los datos.

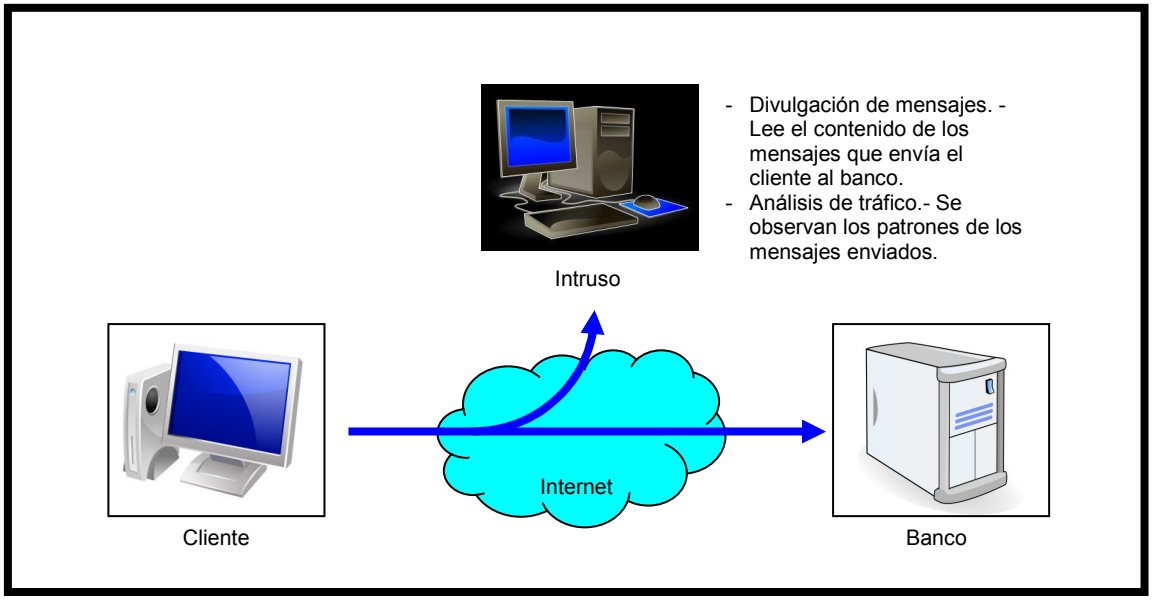


Figura 2.1 Ataques Pasivos

Por otro lado, un intruso podría no sólo escuchar y almacenar los mensajes de datos y de control que circulan por el canal sino utilizarlos para su posterior modificación, inserción u omisión de su contenido, en cuyo caso el ataque pasaría a ser activo. La Figura 2.2 detalla los posibles modos de operación de los ataques activos. En el primero (Figura 2.2.a) el intruso envía mensajes haciéndose pasar por el cliente e incluso puede llegar a interrumpir el servicio brindado por el servidor.

La figura siguiente (2.2.b) muestra el reenvío de mensajes, es decir, el intruso captura el mensaje del cliente, y sin modificarlo lo reenvía más tarde para producir un efecto no autorizado. Finalmente, la Figura 2.2.c muestra la modificación de mensajes donde el intruso captura el mensaje enviado por el cliente, lo modifica de acuerdo a su conveniencia y luego lo envía al servidor del banco.

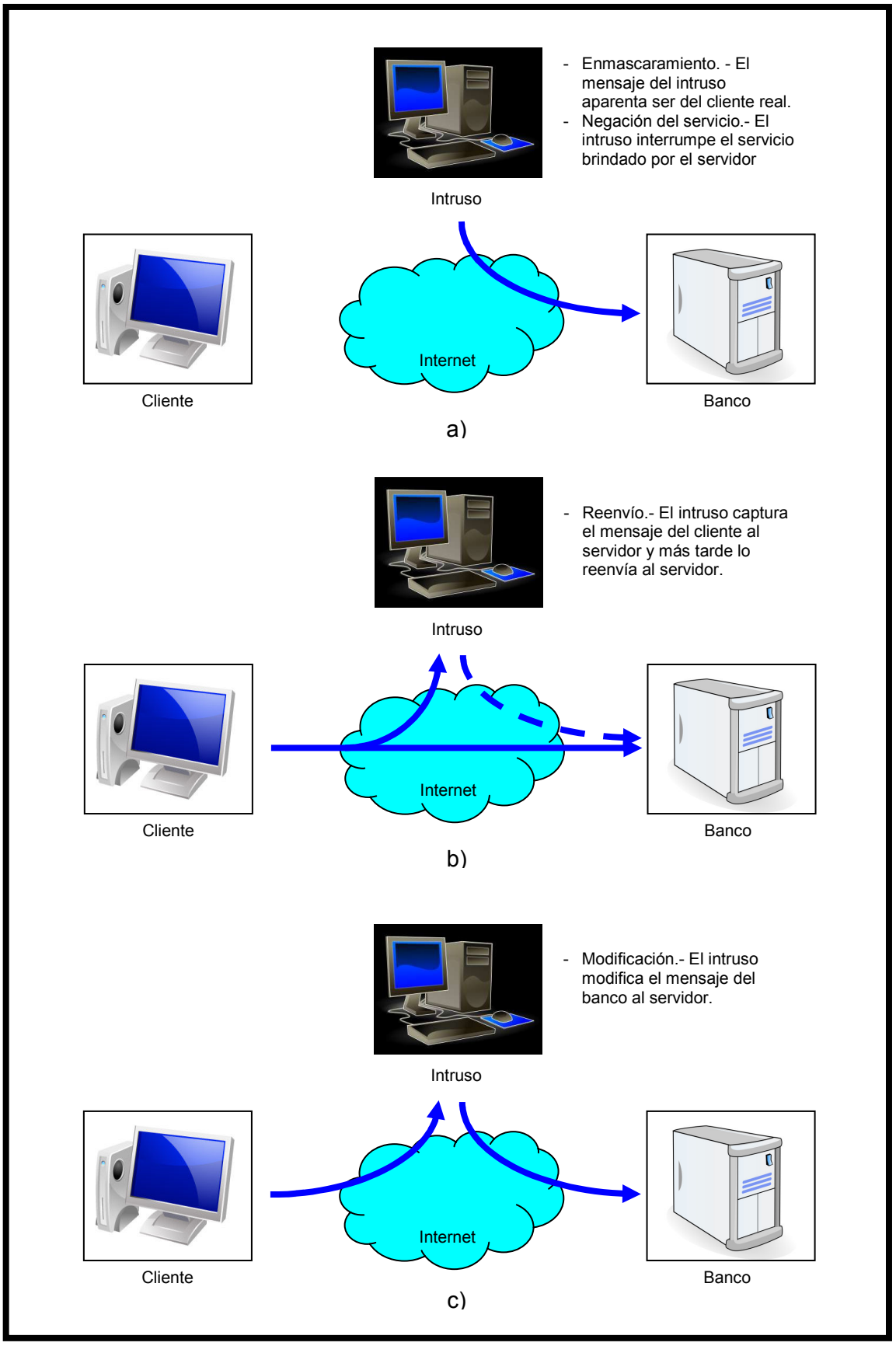


Figura 2.2 Ataques activos

Si no se toman las medidas adecuadas para prevenir estos ataques pasivos y activos descritos, los agujeros en el sistema bancario podrían permitir que el intruso explote una serie de fallas en la seguridad realizando actividades indeseables ya mencionadas, por ejemplo: espiar la comunicación para robar claves de usuario y datos importantes, suplantar al banco o apoderarse de una conexión ya establecida entre el cliente y la entidad.

Formalmente las propiedades deseadas para el establecimiento de una comunicación segura son: confidencialidad, autentificación, integridad de los datos, no repudiación, control de acceso y disponibilidad. Todas estas deben ser garantizadas durante el intercambio de información personal a través de la red.

La *confidencialidad* permite que el contenido del mensaje transmitido pueda ser entendido únicamente por el emisor y el receptor de la información y esté oculto a cualquier atacante que lo intercepte. Entonces se hace necesaria alguna forma de cifrado de la información que haga que, aunque el mensaje se intercepte, éste no pueda ser descubierto. Una comunicación segura, necesita obligatoriamente, poseer una adecuada confidencialidad para su establecimiento.

La *autenticación* garantiza que tanto el remitente como el receptor tengan la capacidad de confirmar la identidad de la otra parte involucrada en la comunicación, es decir, que el receptor pueda comprobar que el transmisor es quien dice ser y viceversa.

La *integridad de los datos* permite que ambas partes involucradas en la conexión se aseguren de que el contenido de su comunicación no sea alterado ni maliciosamente ni por accidente durante la transmisión. En este caso se requiere verificar que no haya habido modificaciones, inserciones, omisiones, repeticiones o reordenación de la información. Para lograr este cometido se utilizan técnicas de control como *checksums* o funciones *hash*

que garanticen la integridad de cada mensaje enviado durante la comunicación, esas técnicas son implementadas por protocolos de transporte confiable.

La *no repudiación* consiste en evitar que la persona o entidad que genera el mensaje o envía los datos pueda negar que fue ella realmente quien remitió tal información. Todas las transmisiones deben ser reconocidas por su emisor.

El *control de acceso* y *disponibilidad* permiten asegurar que las entidades que están intentando lograr acceder a los recursos disponibles en la red sean permitidas únicamente si éstas tienen los derechos de acceso apropiados y llevan a cabo sus ingresos de una manera adecuada. El control de acceso hace que sólo los usuarios legítimos puedan utilizar los recursos disponibles del sistema.

De esta manera, de acuerdo a lo mencionado, se podría intuir que una comunicación segura se enfoca principalmente a proteger la comunicación y los recursos de la red que está siendo utilizada, pero no es tan simple, la seguridad va mucho más allá. La seguridad en una red, en este caso en un sistema bancario en línea, incluye la detección de fallas y de los posibles ataques que se podrían realizar en contra del sistema y la respuesta que da la red a dichos ataques.

Entonces, la seguridad del sistema se convierte en un ciclo de protección, detección y respuesta a los intrusos. Particularmente, la protección se realiza mediante un conjunto de técnicas, una de ellas es el cifrado de la información. A continuación pasamos a explicar brevemente sus nociones básicas.

II.2 SISTEMAS DE CIFRADO SIMÉTRICO Y NO SIMÉTRICO

Como ya se mencionó, la confidencialidad es una de las características más importantes que debe poseer una comunicación entre dos entidades para que ésta pueda considerarse como segura. Esta confidencialidad se la implementa por medio del cifrado o encriptación de los datos a ser intercambiados, en nuestro caso, entre el cliente y el servidor del banco.

Para la implementación del cifrado se hace necesario que ambas partes compartan un secreto común, es decir, la forma en que los datos van a ser descifrados y la clave – o claves – que se utilizará para llevar a cabo el descifrado. Para esto podrían acordar personalmente la clave que van a utilizar, pero en una red tan grande como Internet es obvio que no se puede conocer a todos los individuos y entidades con las que se intercambiará la información sino es a través de la misma red.

Entonces, surge uno de los principales inconvenientes en el establecimiento de una comunicación segura: cómo estar seguros de la identidad de las personas y entidades con las que se comparte datos y se realiza transacciones en Internet.

Para estos fines existen dos tipos de cifrado de información, el simétrico y el no simétrico – también conocido como sistema de llave pública –. En el sistema simétrico las llaves utilizadas por ambas partes – por ejemplo cliente y banco – son idénticas y secretas, de allí la propiedad de simetría. Por otra parte en un sistema no simétrico se utilizan dos claves, una conocida públicamente y una privada. En el siguiente apartado se describe en detalle este sistema.

II.2.1 Claves públicas y privadas

El problema del establecimiento de una comunicación segura se resuelve, actualmente, a través de un sistema de firma electrónica que permite identificar a la otra parte de manera inequívoca y tener la certeza de que el mensaje no ha sido modificado en el tránsito a través de la red.

Estos sistemas de firma electrónica o de llave pública comenzaron a desarrollarse ampliamente alrededor de 1976, cuando W. Diffie y M. E. Hellman publicaron en su artículo "New Directions in Criptography" un algoritmo, que se conocería como Intercambio de Llaves Diffie–Hellman. El uso de los sistemas de llave pública es relativamente simple.

En este tipo de sistemas criptográficos cada persona – o entidad – posee dos claves o llaves: una privada y una pública. La clave privada sólo su propietario la conoce y debe ser usada exclusivamente por él, de esta manera una firma digital identifica de manera única a su firmante. Por otra parte, la clave pública, como su nombre lo indica, está disponible para las demás personas que establezcan intercambio de información con el propietario y permite que el destinatario verifique la identidad del emisor y la integridad del mensaje.

En los sistemas criptográficos de clave pública, ambas claves – la pública y la privada – actúan de forma complementaria. Esto quiere decir que la información cifrada con la clave pública de una persona – o entidad – únicamente puede ser descifrada por la correspondiente clave privada de la misma persona, y viceversa; todo esto utilizando un algoritmo previamente determinado para el cifrado y descifrado del mensaje. De esta manera no se tiene dudas sobre la persona que cifró el mensaje y acerca de la integridad de los datos.

En la Figura 2.3 se expone en detalle la criptografía de llave pública. El mensaje que el usuario A quiere enviar a B es cifrado con la llave pública de B. Una vez que el mensaje ha sido recibido, B aplica el algoritmo de descifrado con su llave privada, de tal manera que el texto o mensaje recuperado sea el original. Como ya se ha mencionado, lo cifrado con una llave – la pública o la privada – puede ser descifrado sólo con la otra – la privada o la pública -.

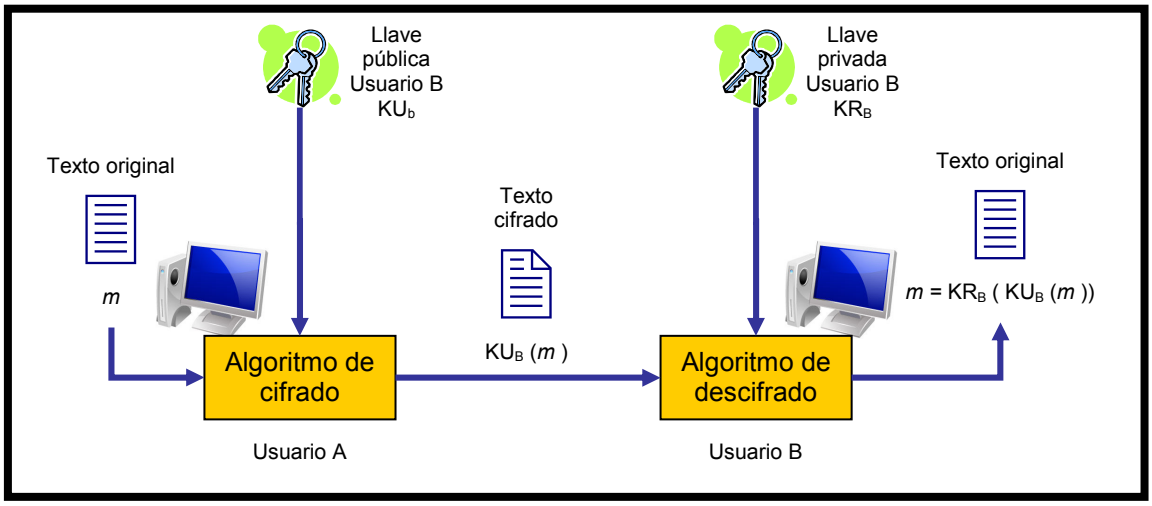


Figura 2.3 Criptografía de llave pública

II.2.1.1 Un ejemplo de algoritmo de cifrado de llave pública: RSA

RSA es uno de los dos algoritmos de llave pública más ampliamente usados. Fue desarrollado en 1977 por Ron Rivest, Adi Shamir y Len Adleman en el MIT (Massachusetts Institute of Technology) y publicado un año después con el nombre de RSA producto de las iniciales de los apellidos de sus desarrolladores.

RSA es un cifrador de bloque en donde el texto original (*M*) y el texto cifrado (*C*) son enteros entre 0 y *n*-1 para un *n* dado. Entonces para el cifrado y descifrado se utilizan las siguientes ecuaciones:

$$C = M^{e} \bmod n$$
$$M = C^{d} \bmod n = \left(M^{e}\right)^{d} \bmod n = M^{ed} \bmod n$$

Ambas partes involucradas en la comunicación deben conocer los valores de *n* y *e*; el valor de *d* es conocido únicamente por el receptor. De allí que la llave pública es KU={*e,n*} y la llave privada KR={*d,n*}. Además, para un cifrado satisfactorio, debe encontrarse valores para *e,d* y *n* tales que $M^{ed} = M \bmod n$ para todo $M < n$. Por otro lado debe ser relativamente fácil calcular M^e y C^d para todos los valores de $M < n$ y se tiene que cumplir que sea improbable hallar *d* dados *e* y *n;* este último requerimiento se lo consigue con valores relativamente grandes de *e* y *n.*

Entonces, lo primero que debe llevarse a cabo para la generación de las llaves es escoger dos números primos *p* y *q*. Para ejemplificar el algoritmo tomaremos *p*=3 y *q*=11. Luego se calcula *n* = *pq,* entonces $n = 3 \times 11 = 33$. A continuación se calcula $\phi(n) = (p-1)(q-1) = 2 \times 10 = 20$. Se selecciona *e* tal que éste sea relativamente primo a $\phi(n)$ y menor que $\phi(n)$; en el ejemplo se escoge *e*=7. Determinamos *d* tal que $de \bmod \phi(n) = 1$ y $d < \phi(n)$; el valor correcto para *d* es 3 ya que $3 \times 7 = 1 \times 20 + 1$. Por tanto las llaves pública y privada obtenidas son KU={*e,n*}={7,33} y KR={*d,n*}={3,33}.

Ahora, por ejemplo, si se quiere cifrar el mensaje *M*=5 se tiene que:

$$C = M^{e} \bmod n = 5^{7} \bmod 33 = 78125 \bmod 33 = 14$$

Para descifrar el mensaje original se aplica:

$$M = C^{d} \bmod n = 14^{3} \bmod 33 = 2744 \bmod 33 = 5$$

La estructura y la matemática que emplea el algoritmo RSA hace que las claves usadas puedan ser descubiertas únicamente usando un ataque de fuerza bruta que tardaría cierto tiempo en llevarse a cabo. Por tanto, si se quiere hacer que el algoritmo sea más seguro se deben utilizar valores altos (de alrededor de 1024 bits, ó 300 dígitos decimales) para *e* y *n,* aunque se debe tomar en cuenta que podría presentarse el inconveniente de tener un sistema lento al intentar factorizar un *n* extremadamente grande para hallar sus números primos.

II.2.1.2 Certificados

Los sistemas de clave pública pueden hacer uso de certificados digitales. Éstos garantizan que la clave contenida en ellos pertenezca efectivamente a la persona o institución indicada. Un certificado es un documento electrónico, emitido por una Entidad Certificadora, que identifica de forma segura al poseedor del mismo, evitando la suplantación de identidad por terceros.[4] Estos certificados asocian una clave pública con su propietario.

Un certificado es una herramienta que permite avalar la identidad de los participantes en una transacción electrónica que obligatoriamente requiere extrema seguridad para su realización. Consiste en una identificación única en Internet que permite demostrar a la otra máquina, por ejemplo el servidor del banco, que uno es quien dice ser. Igualmente, el servidor del banco – o el banco en general – posee su respectivo certificado digital y con ello prueba que el Banco A es efectivamente el Banco A y que el cliente no está enviando sus datos personales a un impostor.

[4] Almagro M. 2000. La seguridad en la banca a distancia. Instituto para la Seguridad en Internet. [Internet]. [Citado 2009, Octubre 19]. Disponible en http://www.instisec.com/publico/verarticulo.asp?id=19

Entonces, es importante también estar seguros de que los datos contenidos en el certificado, por ejemplo la llave pública, sean datos válidos. Para ello se debe recurrir a instituciones confiables que los avalan conocidas como prestadores de servicios de certificación o entidades certificadoras. Estas entidades garantizan la validez del certificado mediante su firma digital incluida en el mismo. Entre los varios prestadores de servicios de certificación que existen actualmente se puede citar a VeriSign.

Pero, si bien es cierto, un certificado puede ser validado por una entidad prestadora de servicios de certificación, tal validación tiene un tiempo para el cual estará vigente. Por tanto, se hace necesario también la verificación del estado del certificado pues éste puede ser reconocido por la entidad como válido, expirado o desconocido, lo cual pondría en duda también la validez de dicho certificado. Para obtener esta información del estado del certificado actualmente se utiliza OCSP (Online Certificate Status Protocol), tema que se trata en el apartado que sigue.

En la Tabla 2.1 de la página siguiente, se muestra un ejemplo de un certificado con sus componentes más importantes y un pequeño comentario de los mismos, los no relevantes se han obviado. El certificado corresponde al BBVA[5], uno de los mayores bancos españoles.

En este certificado son importantes campos como la versión que diferencia el certificado de los distintos formatos existentes. Además se tienen otros campos primordiales como el que define el algoritmo usado para firmar el certificado y los que determinan la validez del mismo con su fecha de expedición y fecha de expiración. Sin duda la parte esencial de un certificado se encuentra en el campo que contiene la llave pública pues ésta permite efectuar la comunicación segura entre el cliente y el servidor. En

[5] Certificado obtenido de www.bbva.es.

este campo se incluye además información sobre el algoritmo con el cual debe usarse la clave proporcionada.

CAMPO	VALOR	FUNCIÓN[6]
Versión	V3	Diferencia entre las sucesivas versiones del formato de certificado.
Número de serie	07 a9 69 15 49 51 75 13 2b 5f 09 97 3d d9 8f 41	Un valor entero único que identifica inequívocamente al certificado.
Algoritmo de firma	sha1RSA	Detalla el algoritmo usado para firmar el certificado con sus correspondientes parámetros.
Emisor	OU = www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign OU = VeriSign International Server CA - Class 3 OU = VeriSign, Inc. O = VeriSign Trust Network	El nombre de la entidad certificadora que ha creado y firmado el certificado.
Válido desde	Miércoles, 12 de Agosto de 2009 19:00:00	Fecha de expedición del certificado.
Válido hasta	Miércoles, 03 de Octubre de 2007 18:59:59	Fecha de caducidad.
Asunto	CN = www.bbva.es OU = Departamento De Informatica O = BBVA L = Madrid C = ES	El nombre del usuario a quien se refiere el certificado.
Clave pública	RSA (1024 bits) 30 81 88 02 81 80 7d 1e d9 67 8d e0 53 c3 66 c7 b1 47 41 46 3f c8 b7 95 bb 20 2e b3 bb 6c df f0…[7]	La llave pública del usuario a quien se expide el certificado. Además se incluye un identificador del algoritmo con el cual la clave debe ser usada.
Restricciones básicas	Tipo de asunto=Entidad final Restricción de longitud de ruta=Ninguno	Campos para extensiones que fueron añadidos a partir de la versión 3.
Uso de la clave	Firma digital, Cifrado de clave (a0)	
Puntos de distribución CRL	[1]Punto de distribución CRL Nombre del punto de distribución: Nombre completo: Dirección URL=http://SVRIntl-crl.verisign.com/SVRIntl.crl	
Algoritmo de identificación	sha1	Estos dos campos corresponden a la firma y contienen el código *hash* del resto de campos cifrado con la llave privada de la entidad certificadora. Se incluye además el identificador del algoritmo de la firma.
Huella digital	5e 3e ea 9b 5a db b3 9e af db c4 1d 19 f7 86 f6 1f 0a bb 27	

Tabla 2.1 Ejemplo de certificado

[6] La tercera columna no forma parte del certificado, se la ha incluido con fines ilustrativos.
[7] Para que la tabla pueda mostrarse en una página se ha suprimido el resto de la clave pública.

II.2.1.3 OCSP

OCSP (Online Certificate Status Protocol) es un protocolo estándar que proporciona a los usuarios y las aplicaciones un método rápido para obtener el estado de un certificado. Permite que las aplicaciones determinen el estado de revocación de los certificados digitales X.509[8] identificados durante una conexión. La validación se realiza en base al número de serie y al firmante del certificado. OCSP se encuentra descrito en el RFC 2560 y está en el registro de estándares de Internet.

Además, OCSP es parte de la familia de estándares técnicos para la Infraestructura de Clave Pública (PKI) X.509 para Internet y es uno de los dos esquemas comunes (el otro es CRL) para mantener la seguridad de un servidor y otros recursos de red. Este protocolo proporciona un sistema fiable para verificar la validez de los certificados utilizados en transacciones electrónicas, tanto en el momento de su realización como a posteriori. Con esta tecnología la validación es en tiempo real siempre con un óptimo nivel de respuesta; OCSP es, en pocas palabras, un servicio de validación online.

Con este método, cuando un usuario intenta acceder a un servidor, por cada certificado involucrado en dicha transacción electrónica, la aplicación correspondiente envía al servidor de validación de certificados una petición de verificación recibiendo instantáneamente una respuesta firmada que indica su estado actual. El estado que es devuelto en la respuesta puede ser 'válido', 'expirado' o 'desconocido'. También se puede devolver un código de error en cuyo caso la respuesta no estaría firmada.

[8] Se trata de un estándar para infraestructura de llave pública (PKI). Especifica formatos estándar para certificados de llave pública. X.509 es una recomendación del ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) parte de la serie X.500 que define un servicio de directorio. El directorio es un conjunto de servidores que mantienen una base de datos de información sobre usuarios. Este directorio puede servir como un depósito de certificados de llave pública. (Housley, et. al. 1999).

Este protocolo especifica la sintaxis para la comunicación entre el servidor que contiene el estado del certificado y la aplicación cliente que necesita la información de validación. Además cabe destacar que OCSP permite a los usuarios con certificados expirados un período de gracia, de tal manera que puedan acceder a los servidores requeridos por un tiempo limitado antes de que la validación tenga que renovarse.

Los mensajes OCSP se codifican en ASN.1[9] y generalmente se transmiten sobre HTTP. La naturaleza de las peticiones y respuestas de OCSP hace que a los servidores OCSP se les conozca como "OCSP *responders*".

El método utilizado con anterioridad a OSCP para verificación de certificados es CRL (*Certificate Revocation List*) que en muchos escenarios ha sido suprimido y reemplazado por esta alternativa más reciente. OCSP presenta múltiples ventajas sobre la validación basada en listas de revocación (CRL); de hecho, fue creado para solventar ciertas deficiencias de las listas de revocación de certificados. La principal limitación de CRL es el hecho de que las actualizaciones deben ser frecuentemente descargadas para mantener la lista actual correcta en el cliente.

[9] ASN.1 (Abstract Syntax Notation One) es una notación estándar que se usa para la definición de tipos de datos, valores y restricciones de tipos de datos. Esta notación puede ser aplicada donde quiera que sea necesario definir sintaxis abstracta de información. Se trata de un estándar utilizado en protocolos de Internet, particularmente en gestión de redes. (ITU-T X.680 Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation. [Internet]. [Actualizado 2002]).

Cuando se despliega una infraestructura de clave pública (PKI) se prefiere la validación de certificados usando OCSP en lugar de CLRs ya que, OCSP brinda ciertas facilidades, entre las que se pueden mencionar:

- No requiere distribuir periódicamente listas de revocación de certificados voluminosas, ahorrando así tráfico de red y procesamiento por parte del cliente.
- Proporciona información más adecuada y reciente del estado de revocación de un certificado.
- Permite la verificación posterior de la validez de las transacciones si se almacenan las respuestas firmadas. Para esto no hay necesidad de archivar todas las listas.
- Se pueden implementar mecanismos de tarifación para que el coste de validación lo asuma el vendedor en lugar del cliente.
- Soporta el encadenamiento de confianza de las peticiones OCSP entre los "*responders*". Esto permite que los clientes se comuniquen con un "*responder*" de confianza para lanzar una petición a una autoridad de certificación alternativa dentro de la misma PKI.

Se debe mencionar también que el uso de OCSP no excluye la utilización de CRLs sino que los dos sistemas se pueden complementar. Dado que la validación con OCSP requiere que el servidor de validación siempre esté disponible, es posible mantener la validación de los certificados contra CRLs como un mecanismo alternativo que puede usarse en el caso de que la validación online no esté disponible de forma temporal.

II.3 SECURE SOCKETS LAYER (SSL)

SSL es un protocolo ampliamente utilizado en el comercio y transacciones electrónicas y ha sido implementado por gran cantidad los servidores *web* existentes. Secure Sockets Layer está diseñado para proveer cifrado de datos y autenticación entre un cliente y un servidor *web*; de esta manera se brinda privacidad a la información entre los extremos de la comunicación mediante la utilización de criptografía.

Fue originalmente desarrollado por Netscape Communications Corporation y al ser un protocolo criptográfico para comunicaciones seguras sobre Internet utiliza un algoritmo de cifrado simétrico como RC4 (*Route Coloniale 4*), DES (*Data Encryption Standard*), o IDEA (*International Data Encryption Algorithm*). También pueden usarse RC2, Triple DES o AES (*Advanced Encryption Standard*). El cifrado de la clave de sesión se realiza mediante un algoritmo de cifrado de clave pública, generalmente RSA, Diffie-Hellman o DSA (*Digital Signature Algorithm*). Esta clave de sesión se utiliza para cifrar la información proveniente y dirigida hacia el servidor; para cada transacción realizada se genera una nueva clave de sesión. El algoritmo utilizado para las funciones *hash* es MD5 (*Message Digest algorithm 5*) o alguno de la familia SHA (*Secure Hash Algorithm*).

Como se puede observar en la Figura 2.4, luego de establecer una conexión TCP, el protocolo SSL inicia con una fase de *handshake* que negocia entre las partes el algoritmo de cifrado que se utilizará en la comunicación. Luego de esto se intercambian las llaves públicas y el servidor autentica al cliente basándose en los certificados digitales. También puede darse que, además, el cliente autentique al servidor. Cuando esta fase termina empieza la transmisión de los datos, entonces, toda la información es cifrada usando las claves de sesión ya negociadas en el *handshake*.

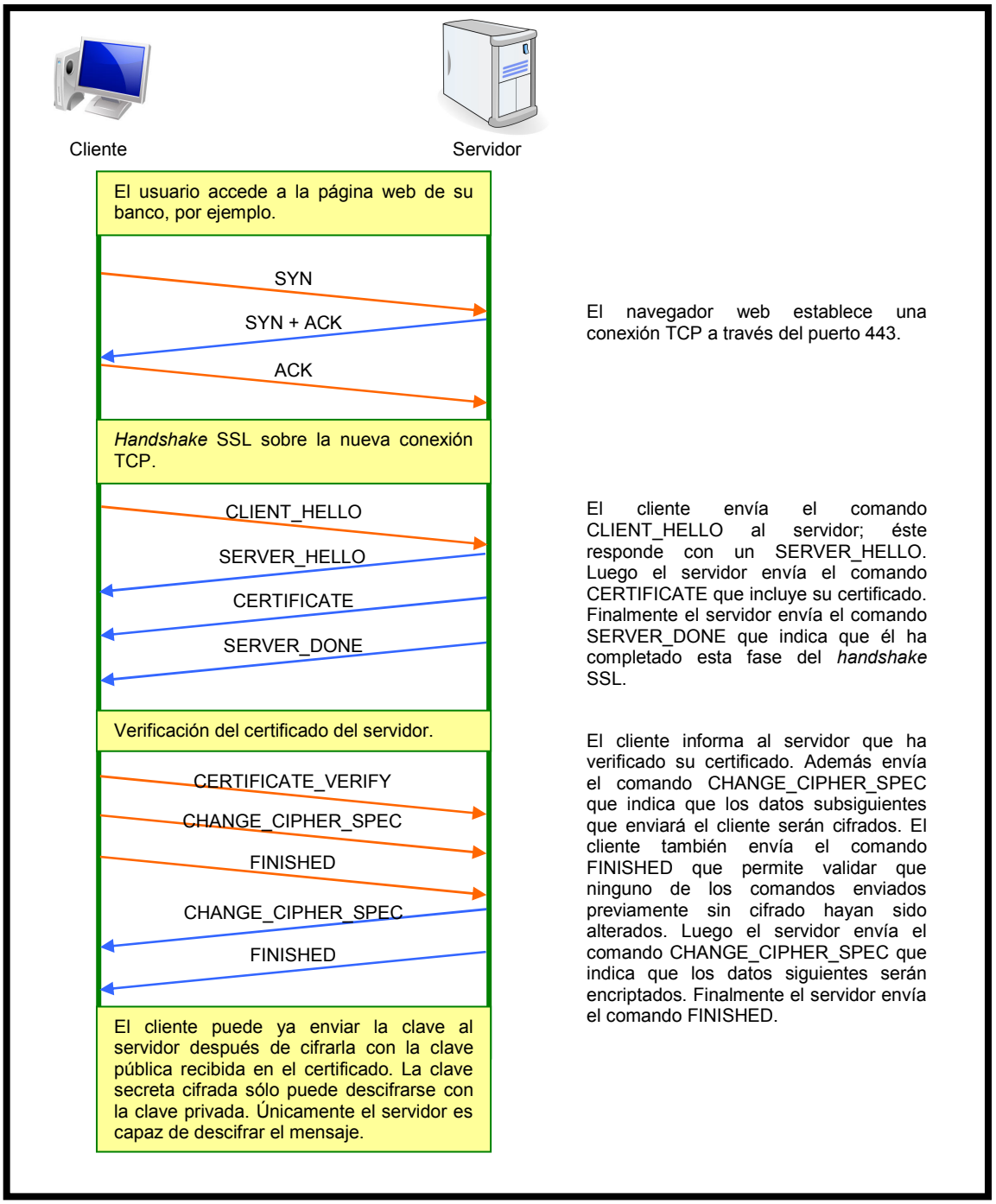


Figura 2.4 Diagrama de tiempo SSL

Entre las varias utilidades que proporciona SSL se pueden destacar: cifrado de datos, autenticación de servidores, integridad de la información y autenticación de cliente para conexiones TCP/IP, ésta última es opcional. Este protocolo permite que las aplicaciones cliente-servidor puedan comunicarse de tal manera que se previenen posibles

intrusos dentro de la conexión, asimismo se evita la falsificación de la identidad del emisor del mensaje.

SSL constituye la base de TLS (*Transport Layer Security*); la última versión de SSL es la 3.0 que fue publicada en 1996 y que luego serviría para desarrollar TSL 1.0[10], protocolo estándar de IETF (*Internet Engineering Task Force*) definido en el RFC 2246. SSL fue diseñado de manera extensible con soporte para compatibilidad y opera de manera modular.

[10] La versión más reciente de TLS es la 1.2, definida en el RFC 5246 en Agosto de 2008.

CAPÍTULO 3

III. DESCRIPCIÓN DE LOS SISTEMAS Y MODELOS MATEMÁTICOS DE LOS BANCOS ANALIZADOS

Con la finalidad de evaluar los sistemas de acceso a través de Internet de algunos bancos del Ecuador es necesario efectuar una descripción de las características que presenta cada uno de ellos. Para ello la identidad de los bancos analizados se mantiene anónima y simplemente se los clasifica en dos tipos (Tipo I y II) de acuerdo a las propiedades de su sistema de autentificación; en nuestro caso de estudio, se han tomado en cuenta las características del nombre de usuario y de la contraseña que emplean para permitir al cliente acceder a los servicios que presta su banco.

Es preciso señalar que para realizar esta clasificación se han analizado los sistemas de cinco bancos ecuatorianos. Las características de cada uno de los tipos mostrados son generalizaciones de las propiedades de los bancos observados; en cada tipo de banco se resumen las principales características que presentan los bancos clasificados.

Asimismo, los modelos matemáticos que se detallan en la segunda parte de este capítulo toman en cuenta la categorización de las instituciones bancarias ya mencionada. De esta manera, tanto el Tipo I como el Tipo II tendrán su propio modelo matemático para determinar las probabilidades de acceso a una cuenta, probabilidades de realizar una transferencia y el número de cuentas a las que se accedería en un ataque al mecanismo de acceso a la banca electrónica.

III.1 PRINCIPALES CARACTERÍSTICAS: NOMBRE DE USUARIO Y CONTRASEÑA

Entre las características principales de los sistemas de autenticación se pueden destacar al nombre de usuario y contraseña. La Tabla 3.1 muestra la información que se usa como nombre de usuario en cada uno de los tipos de banco. El Tipo I emplea la información del número de cédula del cliente como nombre de usuario mientras que para autenticar en el banco Tipo II se requiere ingresar el número de tarjeta de cajero automático.

La longitud del nombre de usuario del banco Tipo I es de 10 caracteres, todos números entre 0 y 9. El cliente introduce estos datos a través del teclado convencional. Igualmente, en el banco Tipo II, la longitud del número de tarjeta es diez caracteres de tipo numérico. El usuario ingresa su identificación mediante teclado.

CARACTERÍSTICA	BANCO TIPO I	BANCO TIPO II
NOMBRE DE USUARIO	*Cédula de Identidad*	*Número de tarjeta de cajero automático*
Longitud	10 caracteres	10 caracteres
Tipo	Numérico	Numérico
Método de ingreso	Teclado	Teclado

Tabla 3.1 Nombres de Usuario

La Tabla 3.2 muestra las características de los tipos de contraseñas que manejan los bancos estudiados. El usuario del banco Tipo I para ingresar debe poseer una clave de 4 dígitos escogida por él; la ingresa utilizando un teclado virtual aleatorio que se presenta en el portal *web* de la institución bancaria. Para el banco Tipo II, la clave presenta las mismas características, con la única diferencia en el modo de ingreso, el

cual se realiza de manera convencional mediante teclado. En ambos casos la cuenta se bloquea luego de ingresar una contraseña incorrecta tres veces consecutivas.

CARACTERÍSTICA	BANCO TIPO I	BANCO TIPO II
CONTRASEÑA	*Clave escogida por el usuario (puede cambiarse)*	*Clave escogida por el usuario (puede cambiarse)*
Longitud	4 caracteres	4 caracteres
Tipo	Numérica	Numérica
Método de ingreso	Teclado virtual aleatorio	Teclado

Tabla 3.2 Contraseñas

III.2 CLAVES ADICIONALES

Además de las dos características detalladas anteriormente, cada tipo de banco posee características adicionales que se indican en la Tabla 3.3.

CARACTERÍSTICA	BANCO TIPO I	BANCO TIPO II
CLAVES ADICIONALES		
Propósito	Realizar transferencias y pagos.	Realizar transferencias.
Forma de obtenerla	Tarjeta adicional con matriz de claves identificadas por la coordenada de la celda.	La clave de confirmación se envía al correo electrónico del usuario.
Longitud	3 caracteres	10 caracteres
Tipo	Numérica	Numérica

Tabla 3.3 Claves adicionales

El banco Tipo I requiere de una clave adicional para realizar transferencias de dinero, pagos de servicios y compras electrónicas. Esta clave es numérica y tiene una longitud de 3 caracteres. El cliente tiene acceso a ella a través de una tarjeta adicional proporcionada por la institución; en dicha tarjeta existe un conjunto de claves dispuestas en forma matricial de tal manera que cada clave se identifica por la ubicación de su celda mediante una letra y un número (coordenada), por ejemplo: A1, H8. Luego de cinco intentos de ingreso fallidos, la cuenta se bloquea.

La clave adicional para el banco Tipo II tiene una longitud de 10 caracteres numéricos. Se la emplea para realizar transferencias entre cuentas del mismo banco así como a cuentas de otras entidades. Esta clave de confirmación de transferencias se envía al correo electrónico del cliente para que luego de ser introducida se complete la transacción. El usuario puede ingresar contraseñas incorrectas sin límite alguno; no existe un control de un número de intentos fallidos para bloquear la cuenta.

III.3 CERTIFICADOS

En la Tabla 3.4 de la página siguiente se presenta información relevante sobre los certificados emitidos[11] para cada tipo de banco. Se puede verificar que ambos tipos manejan certificados versión 3 y que su algoritmo de firma es SHA 1 para RSA. Además se incluyen los datos del emisor y el período de validez del certificado. La parte primordial del certificado se encuentra en el campo de la clave pública, aquí se observan diferencias pues el banco de Tipo I maneja una clave pública de 2 048 bits mientras que el de Tipo II tiene una clave cuya longitud es de 1 024 bits, ambas para

[11] Certificados electrónicos obtenidos el 29 de octubre de 2009. Se reserva la identidad de las instituciones bancarias.

usarse con RSA. Finalmente se incluyen campos que detallan las restricciones básicas, el uso de la clave, el identificador del algoritmo de la firma y la huella digital.

CARACTERÍSTICA	BANCO TIPO I	BANCO TIPO II
Versión	V3	V3
Número de serie	47 8f 39 fd 17 9b a1 fb 3d 2e fe 4b df b9 a1 8f	1c 57 0a 88 5b 49 b5 c3 de a8 7a 32 9c ab 98 d8
Algoritmo de firma	sha1RSA	sha1RSA
Emisor	OU = www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign OU = VeriSign International Server CA - Class 3 OU = VeriSign, Inc. O = VeriSign Trust Network	OU = www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign OU = VeriSign International Server CA - Class 3 OU = VeriSign, Inc. O = VeriSign Trust Network
Válido desde	Jueves, 03 de Abril de 2008 19:00:00	Miércoles, 13 de Junio de 2007 19:00:00
Válido hasta	Domingo, 04 de Abril de 2010 18:59:59	Sábado, 10 de Julio de 2010 18:59:59
Asunto	CN = www02.bancoTipoI.com OU = Member, VeriSign Trust Network OU = Authenticated by E-Sign S.A. OU = Terms of use at www.e-sign.cl/rpa (c) 04 OU = SEGURIDAD INFORMATICA O = BANCO TIPO I L = QUITO S = PICHINCHA C = EC	CN = www.bancoTipoII.com OU = Terms of use at www.verisign.com/rpa (c)05 O = Banco Tipo II S.A. L = Quito S = Pichincha C = EC
Clave pública	RSA (2048 bits) 30 82 01 0a 02 82 01 01 00 b4 0e 26 c3 d2 0b 75 cc fc 7b 1c…[12]	RSA (1024 bits) 30 81 89 02 81 81 00 af df b0 e7 87 38 4b 42 30 1c f5 eb 64…
Restricciones básicas	Tipo de asunto=Entidad final Restricción de longitud de ruta=Ninguno	Tipo de asunto=Entidad final Restricción de longitud de ruta=Ninguno
Uso de la clave	Firma digital, Cifrado de clave (a0)	Firma digital, Cifrado de clave (a0)
Algoritmo de identificación	sha1	sha1
Huella digital	32 36 36 e6 1a df 19 ad b8 92 3e 6b 44 ad ad ac 1f 4b e7 b0	0b 10 3b 58 4d 67 12 02 c8 ed 78 f0 e0 81 0f 2c d5 41 35 a7

Tabla 3.4 Certificados de los bancos de Tipo I y II

[12] Para que la tabla pueda mostrarse en una página se ha suprimido el resto de la clave pública.

III.4 MECANISMOS DE ACCESO

La Figura 3.1 muestra, a breves rasgos, los distintos pasos que debe seguir el cliente del banco de Tipo I para lograr acceso a su cuenta y a los distintos servicios que se encuentran disponibles en su portal electrónico. El diagrama toma en cuenta las propiedades descritas anteriormente con los requisitos necesarios para cada operación.

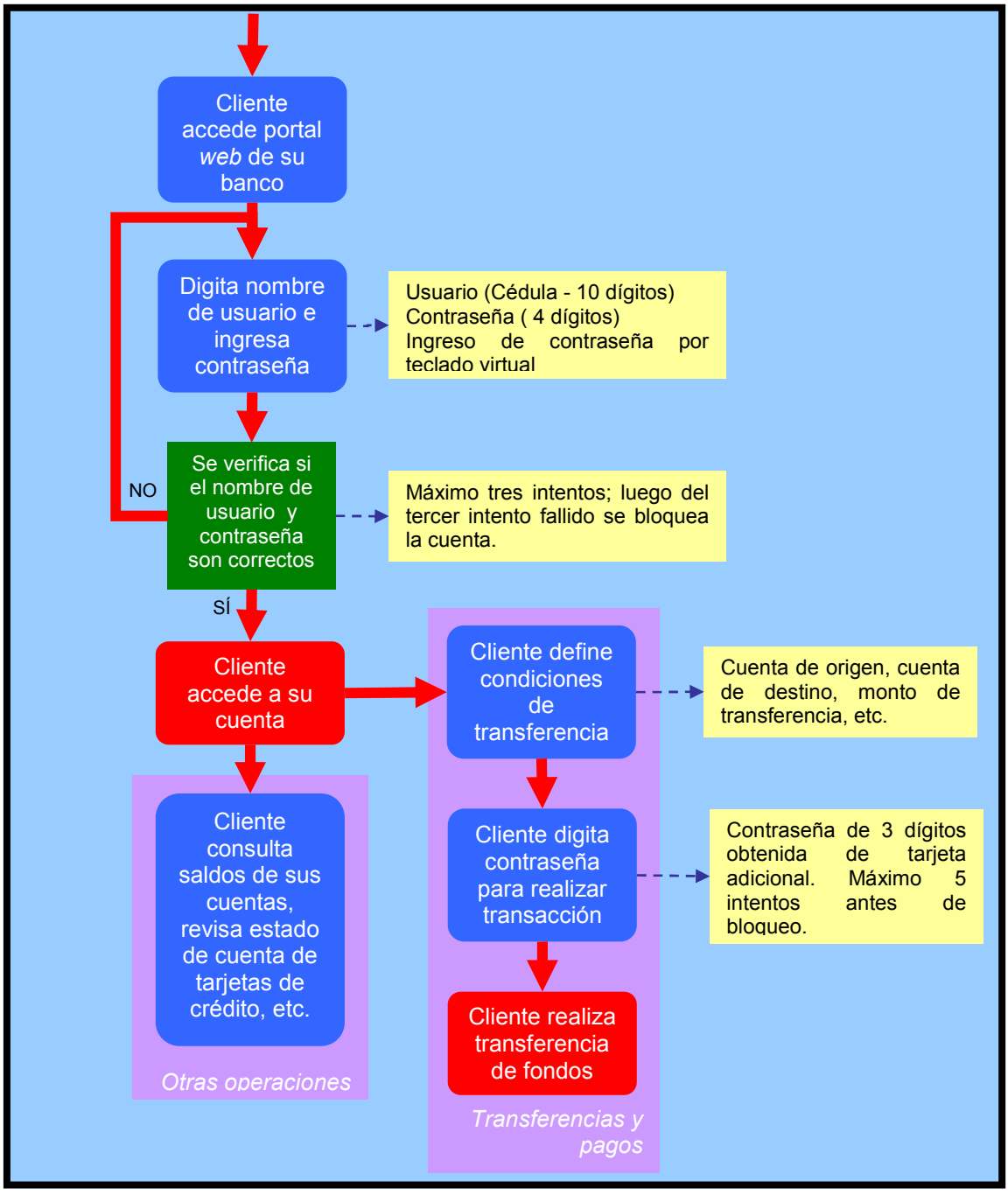


Figura 3.1 Mecanismos de acceso para el banco Tipo I

De forma similar, en la figura siguiente (Figura 3.2) se muestra un flujograma que detalla las operaciones que puede realizar el usuario de un banco de Tipo II y los requisitos que debe cumplir para efectuar las mismas. Si comparamos esta figura con la 3.1 podemos observar con claridad las diferencias existentes entre los dos tipos de banco.

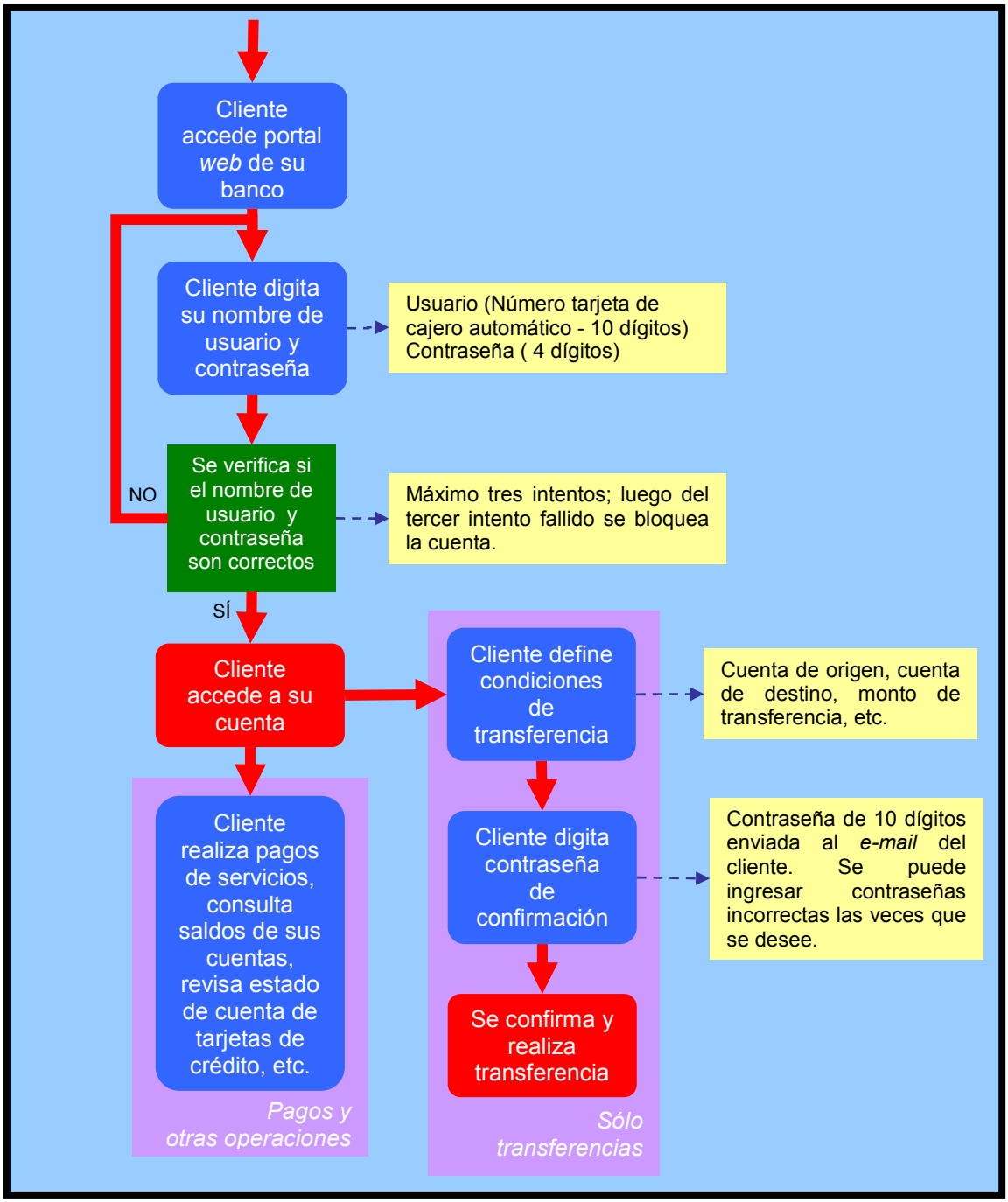


Figura 3.2 Mecanismos de acceso para el banco Tipo II

III.5 MODELOS MATEMÁTICOS

A continuación se detallan los modelos matemáticos de los dos tipos de bancos ya mencionados. Además se incluye un modelo alternativo que se caracteriza por la necesidad de una clave no únicamente numérica, sino alfanumérica (62 caracteres posibles), para lograr el acceso a la cuenta.

En todos los modelos se plantean las ecuaciones correspondientes acompañadas de ejemplos numéricos asignando ciertos niveles a los factores que influyen en las variables analizadas. Los factores que se toman en cuenta para la realización de los modelos descritos son la longitud de la contraseña de acceso, el tipo de la misma – si es sólo numérica o alfanumérica –, el número de clientes que tendría el banco y el número de intentos luego de los cuales se bloquea la cuenta. Para la realización de transferencias se requieren además la longitud de la clave adicional y el número de intentos de transferencia antes de llegar al bloqueo. Ahora se emplean sólo ciertos factores, en el siguiente capítulo se presenta el diseño del experimento donde se expondrá una descripción completa de las variables, factores y niveles que se involucrarán en la simulación realizada.

III.5.1 Banco Tipo I

Las características de este tipo de banco se han descrito ya anteriormente. En base a éstas, podemos considerar los siguientes niveles para la descripción de su modelo matemático particular. Para facilitar el desarrollo consideraremos valores literales y luego se explicará el modelo con un ejemplo numérico; las variables y sus niveles literales se muestran en la Tabla 3.5.

FACTOR	NIVEL
Longitud de la clave (en dígitos)	*n*
Tipo de clave	Numérica
Número de clientes del banco	*N*
Número de intentos de acceso luego de los cuales se bloquea la cuenta	*M*
Longitud de la clave dinámica para transferencias (en dígitos)	*k*
Número de intentos de transferencia antes del bloqueo de la cuenta	*i*

Tabla 3.5 Factores y niveles literales para un banco Tipo I

Como se ha mencionado en las propiedades de este tipo de banco, se necesita de una clave adicional, en este caso de *k* dígitos, para realizar compras, pagos y transferencias de fondos. Por lo tanto, desarrollaremos varios cálculos para analizarlo.

Primeramente, se evaluará la probabilidad de acceso a una cuenta. Luego, se detallará la probabilidad de realizar una transferencia o pago de servicios, una vez que ya se ha logrado acceder a una de ellas. Todo esto tomando en cuenta que esta segunda probabilidad no tiene relación con la primera pues al atacante le interesará primero lograr el acceso a la cuenta, luego, analizando las características de la misma, intentará o no realizar un pago o una transacción electrónica.

Finalmente se desarrolla un cálculo en el cual se toma como supuesto que al atacante le interesa lograr el acceso a la cuenta y, al mismo tiempo, quebrar la clave para realizar transferencias. Entonces, aquí podríamos hablar de un acceso 'total' a la cuenta, en el caso de que las dos claves puedan ser adivinadas en el mismo ataque.

No está por demás acotar que todos estos cálculos son realizados utilizando un modelo que implementa un *ataque de fuerza bruta*[13] para lograr el acceso a las cuentas.

III.5.1.1 Probabilidad de acceso a una cuenta

Entonces, según los niveles literales seleccionados se tiene que, al tratarse de una clave estrictamente numérica (decimal), para cada cliente existirían 10^n posibles claves de acceso (variaciones con repetición entre 0 y 10^n-1, ambas incluidas), asumiendo que uno de los nombres de usuario corresponde efectivamente a un cliente del banco y que se escoge aleatoriamente la clave de acceso, la probabilidad de éxito del ataque para una cuenta es:

$$P_{acc1} = \frac{1}{\#\,\text{claves posibles}} = \frac{1}{10^n} \qquad [3.1]$$

Como la cuenta se bloquea luego de *m* intentos, entonces, la probabilidad de éxito se incrementa a:

$$P_{acc2} = \text{número de intentos de acceso} \times \frac{1}{\#\,\text{claves posibles}} = m \times \frac{1}{10^n} = \frac{m}{10^n} \qquad [3.2]$$

Por tanto, la probabilidad de no acceder a una cuenta será la diferencia entre esta última y 1 (uno):

$$P_{noacceso} = 1 - P_{acc2} = 1 - \frac{m}{10^n} \qquad [3.3]$$

Consecuentemente, dado un conjunto de *N* clientes, para no lograr acceso a ninguna cuenta debería cumplirse que, no se accede a la primera cuenta, y no se accede a la segunda, ni a la tercera; y así hasta completar las *N* cuentas.

[13] Intento de ganar acceso no autorizado a un sistema computacional generando e intentando todas las contraseñas posibles. (McGraw-Hill Dictionary of Scientific and Technical Terms. [Internet]. [Actualizado 2003].)

Por tanto, tal probabilidad se expresa de la siguiente manera:

$$P_{noaccesoNcuentas} = P_{noacceso1} \cdot P_{noacceso2} \cdot P_{noacceso3} \cdot \ldots \cdot P_{noaccesoN} = \left(P_{noacceso}\right)^N \quad [3.4]$$

$$P_{noaccesoNcuentas} = \left(1 - P_{acc2}\right)^N = \left(1 - \frac{m}{10^n}\right)^N \quad [3.5]$$

Entonces, la probabilidad de acceder al menos a una cuenta es 1 (uno) menos esta probabilidad de no acceder a ninguna cuenta del conjunto de *N* clientes; se muestra a continuación:

$$P(x \geq 1) = 1 - P_{noaccesoNcuentas} = 1 - \left(1 - P_{acc2}\right)^N = 1 - \left(1 - \frac{m}{10^n}\right)^N \quad [3.6]$$

Por otra parte, el número esperado de cuentas accedidas durante un ataque viene dada por la probabilidad de éxito de acceso a una cuenta y el número global de cuentas; sería entonces:

$$\#\text{cuentas accedidas} = P_{acc2} \times N \quad [3.7]$$

$$\#\text{cuentas accedidas} = \frac{m}{10^n} \times N = \frac{m \cdot N}{10^n} \quad [3.8]$$

A manera de ejemplo, en la tabla siguiente se detallan los niveles (valores característicos de los bancos analizados) para los distintos factores mencionados.

FACTOR	NIVEL
Longitud de la clave (en dígitos)	*n* = 4
Tipo de clave	Numérica
Número de clientes del banco	*N* = 100 000
Número de intentos de acceso luego de los cuales se bloquea la cuenta	*m* = 3

Tabla 3.6 Factores y niveles para ejemplo de probabilidad de acceso a un banco Tipo I

Entonces, a partir de la ecuación [3.2] se tiene que la probabilidad de lograr acceso a una cuenta es:

$$P_{acc2} = \frac{3}{10\,000} = 0.0003 \qquad [3.9]$$

De donde, se tiene la probabilidad de acceder al menos a una cuenta para estas características particulares:

$$P(x \geq 1) = 1 - \left(1 - \frac{3}{10000}\right)^{100\,000} = 1 - 9{,}3156 \times 10^{-14} \approx 1 \qquad [3.10]$$

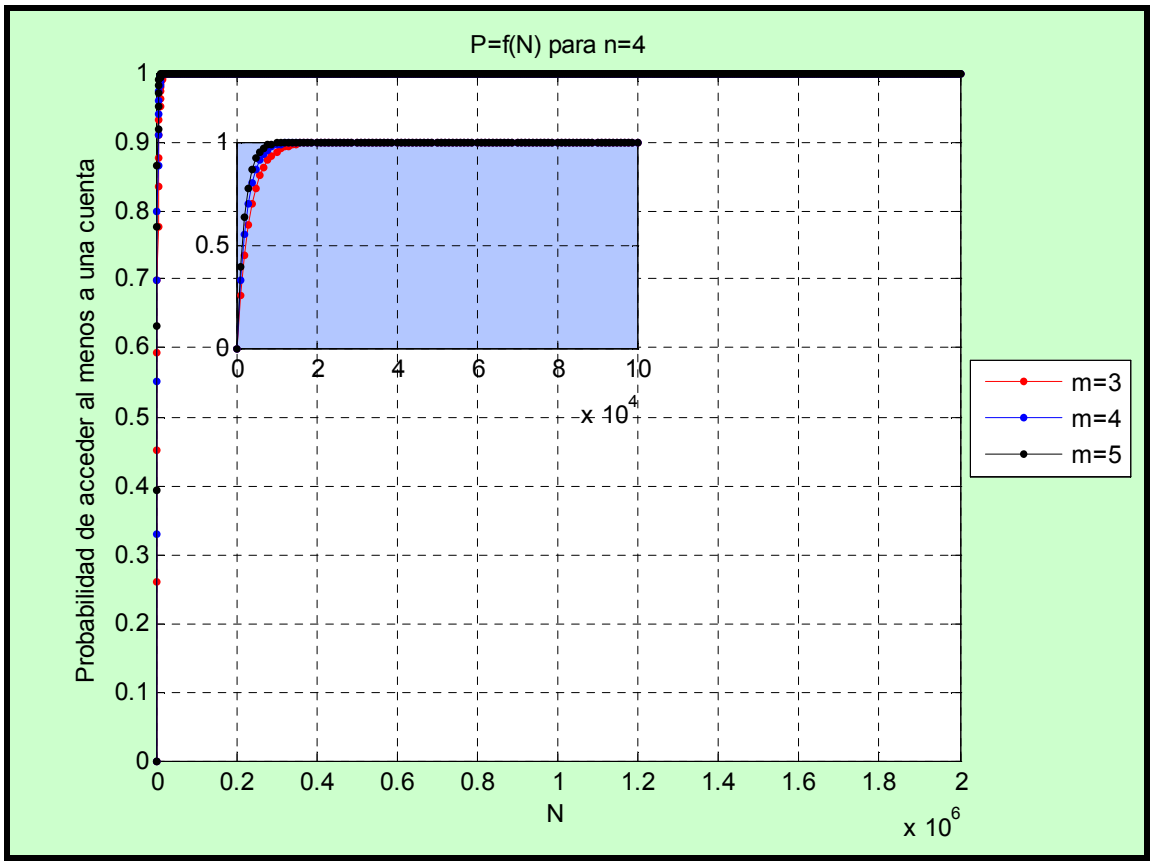


Figura 3.3 Valores teóricos de la probabilidad de acceder al menos a una cuenta para un banco Tipo I

En la Figura 3.3 se grafica la probabilidad de acceder al menos a una cuenta versus el número de usuarios del banco, con una longitud de contraseña *n=4* y para varios *m*. Como se puede observar, para estos niveles, a partir de 30 000 usuarios del

banco existe una probabilidad muy cercana a 1 (uno) de poder lograr acceso al menos a una cuenta. En este caso, si el análisis a realizarse se centra en bancos de más de 100 000 usuarios, siempre se podrá acceder al menos a una cuenta de los mismos.

Se concluye por tanto, que para este sencillo ejemplo existe una gran probabilidad de acceder al menos a una cuenta en la realización de un ataque. Además, siguiendo con el cálculo y usando la ecuación [3.8], el número de cuentas a las que se lograría el acceso durante un ataque sería:

$$\#\text{cuentas accedidas} = \frac{3}{10000} \times 100\,000 = 30 \qquad [3.11]$$

Todo esto tomando en cuenta una distribución uniforme de las claves. Cabe señalar que al ser los ataques aleatorios, en cada nuevo intento – teóricamente – se podría tener acceso a 30 nuevas cuentas para el caso de los niveles escogidos en este ejemplo.

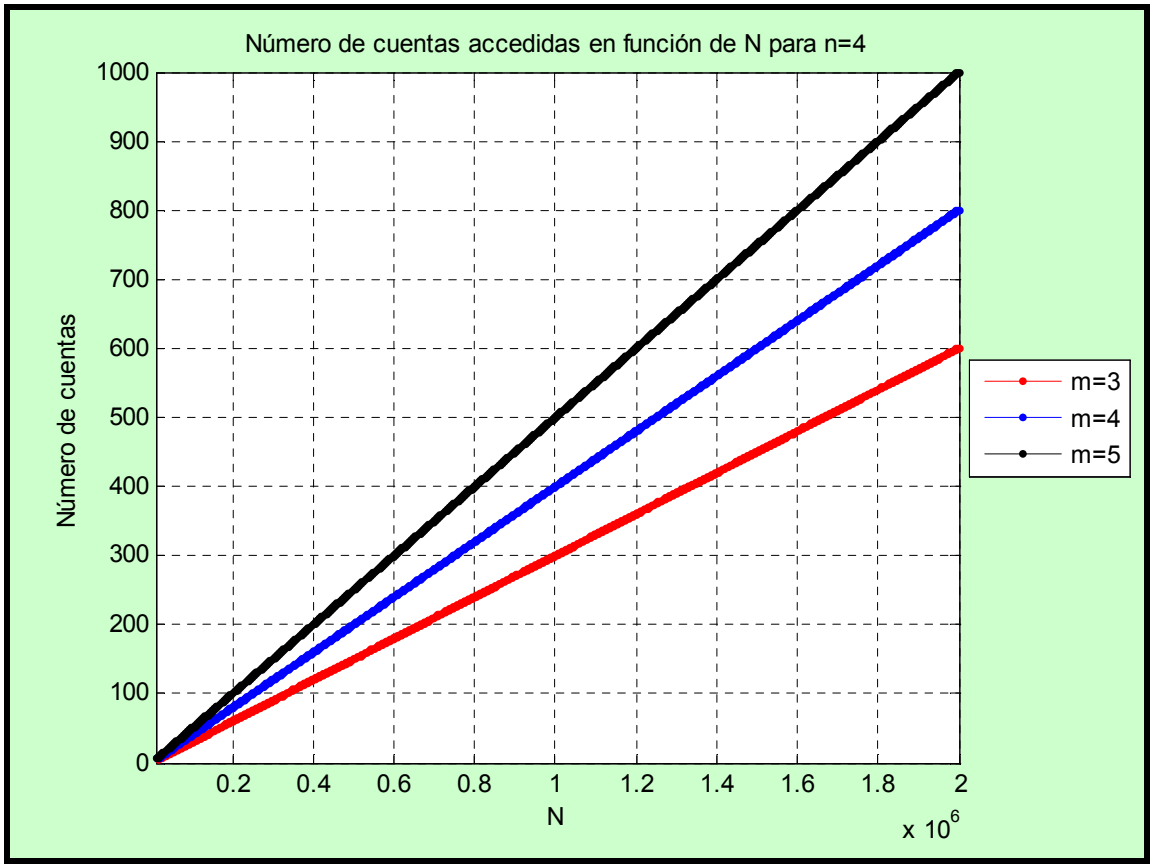


Figura 3.4 Valores teóricos del número de cuentas accedidas para un banco Tipo I

La figura precedente grafica el número de cuentas a las que se logra acceso en cada ataque al sistema bancario, todo esto para una longitud de clave de 4 dígitos y varios niveles de intentos de acceso antes de bloqueo de la cuenta. Tal como muestran las curvas en cada intento de ataque se lograría acceso a un considerable número de cuentas; si colocamos nuestro escenario en bancos que posean entre 100 000 y 1 000 000 de usuarios podemos notar que para *m=3,* se podría tener acceso a un intervalo entre 30 y 300 cuentas, para *m=4,* estas variables se incrementarían a un rango entre 40 y 400 cuentas. Si analizamos bancos con mayor número de usuarios, estas variables sufrirían un incremento de tipo directamente proporcional: con *m=5,* para un banco de 2 000 000 de usuarios, se lograría acceder a 1000 cuentas.

III.5.1.2 Probabilidad de realizar una transferencia, compra o pago de servicios

De acuerdo a las características de este banco, cada vez que el usuario quiera realizar una transacción electrónica deberá ingresar una clave dinámica de *k* dígitos (totalmente numérica). Además, luego de *i* intentos fallidos la cuenta se bloqueará; por lo tanto, la probabilidad de lograr este cometido será:

$$P_{transf} = \text{número intentos de transferencia} \times \frac{1}{\#\,\text{claves posibles}} = i \times \frac{1}{10^k} \qquad [3.12]$$

A manera de ejemplo tomaremos los valores de la Tabla 3.7:

FACTOR	NIVEL
Longitud de la clave dinámica para transferencias (en dígitos)	$k = 3$
Número de intentos de transferencia antes del bloqueo de la cuenta	$i = 5$

Tabla 3.7 Factores y niveles para ejemplo de probabilidad de transferencia para un banco Tipo I

Entonces, si utilizamos la ecuación anterior, tendríamos que la probabilidad de realizar una transferencia para este caso es:

$$P_{transf} = 5 \times \frac{1}{10^3} = 0.005 \quad [3.13]$$

III.5.1.3 Probabilidad de lograr acceso 'total' a la cuenta

Para esto se deberá tomar en cuenta conjuntamente la probabilidad de acceder a la cuenta y la probabilidad de adivinar la clave para transferencias. Entonces, la probabilidad de lograr acceso 'total' a la cuenta viene dada en la ecuación siguiente:

$$P_{accesotot} = P_{acc2} P_{transf} \quad [3.14]$$

Por otro lado, la probabilidad de no lograr tal acceso a la cuenta se obtiene de:

$$P_{noaccesotot} = 1 - P_{acc2} P_{transf} \quad [3.15]$$

Reemplazando [3.2] y [3.12] en [3.15] resulta:

$$P_{noaccesotot} = 1 - \frac{m}{10^n} \frac{i}{10^k} \quad [3.16]$$

Consecuentemente, la probabilidad de acceder 'totalmente' al menos a una cuenta, teniendo un conjunto de *N* clientes del banco, es:

$$P(x \geq 1) = 1 - P_{noaccesotot}{}^N \quad [3.17]$$

$$P(x \geq 1) = 1 - \left(1 - \frac{m}{10^n} \frac{i}{10^k}\right)^N \quad [3.18]$$

Y, el número esperado de cuentas a las que se logre el acceso 'total' durante un ataque es:

$$\#\text{cuentas accedidas 'totalmente'} = P_{acc2} P_{transf} \times N \quad [3.19]$$

$$\#\text{cuentas accedidas 'totalmente'} = \frac{m}{10^n}\frac{i}{10^k} \times N \qquad [3.20]$$

Ahora bien, continuando con el ejemplo tomamos en cuenta los mismos niveles utilizados en los apartados anteriores y agregamos valores para el resto de variables, todos estos se tabulan a continuación:

FACTOR	NIVEL
Longitud de la clave (en dígitos)	$n = 4$
Tipo de clave	Numérica
Número de clientes del banco	$N = 100\,000$
Número de intentos de acceso luego de los cuales se bloquea la cuenta	$m = 3$
Longitud de la clave dinámica para transferencias (en dígitos)	$k = 3$
Número de intentos de transferencia antes del bloqueo de la cuenta	$i = 5$

Tabla 3.8 Factores y niveles para ejemplo de probabilidad de acceso total a un banco Tipo I

Con estos niveles se tiene que la probabilidad de tener acceso total a una cuenta (a partir de [3.14], [3.2] y [3.12]) es:

$$P_{accesotot} = \frac{3}{10^4}\frac{5}{10^3} = (0.0003)\cdot(0.005) = 0.0000015 \qquad [3.21]$$

De [3.16] la probabilidad de no tener acceso total a la cuenta es:

$$P_{noaccesotot} = 1 - \frac{3}{10^4}\frac{5}{10^3} = 1 - (0.0003)\cdot(0.005) = 0.9999985 \qquad [3.22]$$

Seguidamente, la probabilidad de acceder ‘totalmente’ al menos a una cuenta (ecuación [3.18]), teniendo el conjunto de 100 000 clientes del banco, es la siguiente:

$$P(x \geq 1) = 1 - (1 - 0.0003 \cdot 0.005)^{100\,000} = 1 - 0.86071 = 0.13929 \qquad [3.23]$$

Por tanto, en este caso, es mucho más difícil lograr el acceso a una cuenta y en el mismo intento lograr quebrar la clave para transacciones. Como se puede notar la probabilidad es muy pequeña, esto debido a que ahora se necesita adivinar no una sino dos claves en el mismo acceso.

Ahora, tomando en cuenta [3.20], el número esperado de cuentas a las que se logre el acceso 'total' durante un ataque es:

$$\#\text{cuentas accedidas } \text{'totalmente'} = 0.0003 \times 0.005 \times 100\,000 = 0.15 \approx 0 \qquad [3.24]$$

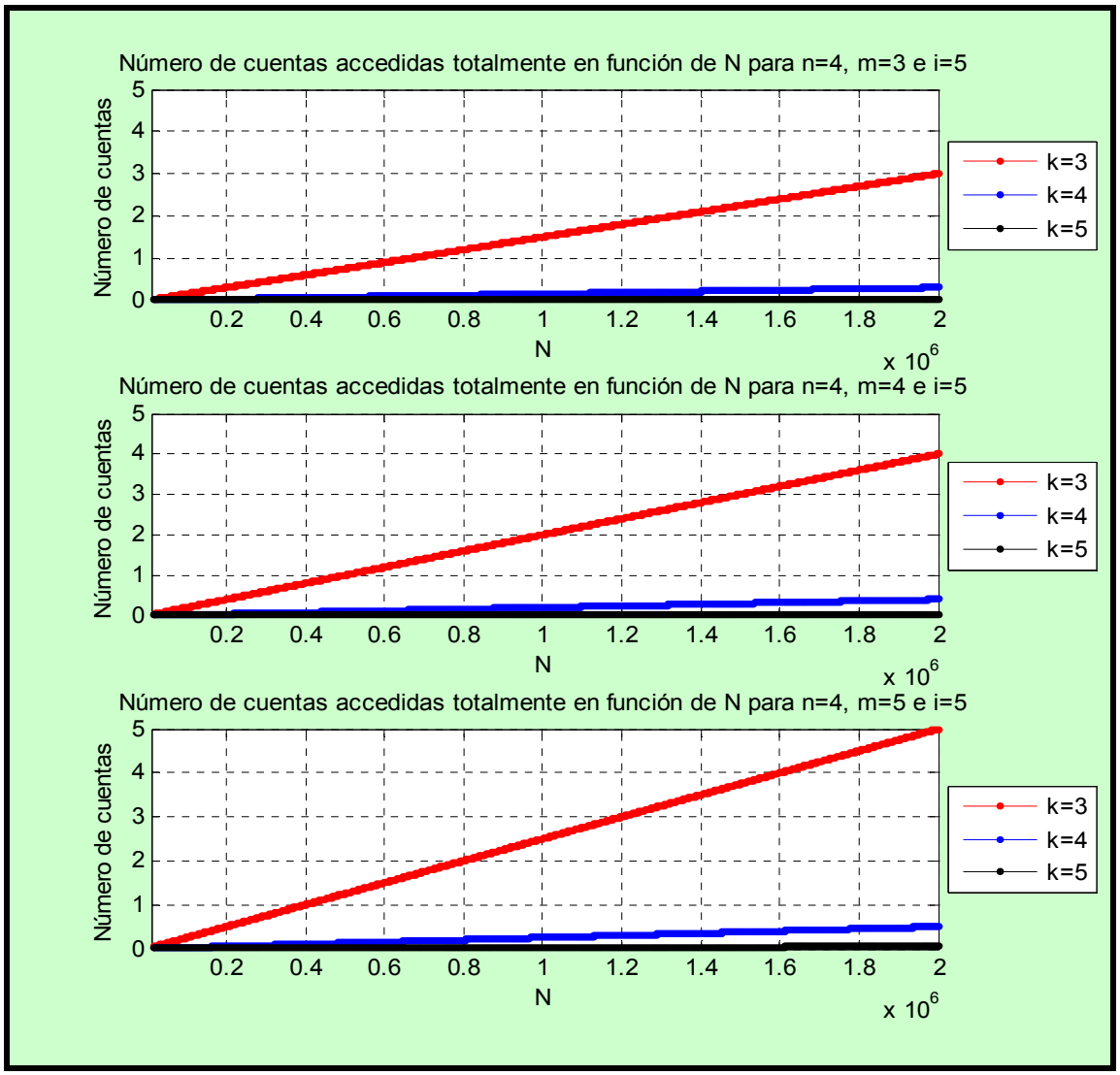


Figura 3.5 Valores teóricos del número de cuentas accedidas totalmente para un banco Tipo I

La Figura 3.5 grafica el número de cuentas accedidas totalmente con niveles de *n=4* e *i=5*, para varios *m* y *k.* Se puede notar que la cantidad de cuentas a las que se logra acceso total en cada intento es mínima; mientras menor es *m* y/o mayor es *k*, menor es el número de cuentas accedidas. Es así que para un banco de 1 000 000 de usuarios, con *m=5* y *k=3* se podría acceder totalmente máximo a 2 cuentas (exactamente 2,5 cuentas, pero tomamos en cuenta únicamente el valor entero).

Entonces es fácil notar que es teóricamente muy difícil lograr el acceso simultáneo tanto a la cuenta como a la realización de una operación bancaria que requiera una clave adicional. No está por demás mencionar que si la clave adicional fuese de menos dígitos y el número de cuentas a las que se intente acceder fuese mayor, entonces existiría mayor probabilidad de acceder 'totalmente' a una cuenta, como ya se puede notar en los gráficos.

También se debe precisar que el que se haya logrado el acceso total a una cuenta no garantiza que se pueda hacer cualquier cantidad de transacciones en la misma pues, como se ha dicho, la segunda clave es una clave dinámica y la misma clave generada puede repetirse únicamente luego de un cierto número de transacciones realizadas.

A continuación, en la página siguiente, se tabulan los resultados obtenidos en el ejemplo desarrollado para un banco Tipo I.

FACTORES Y NIVELES SELECCIONADOS	
Longitud de la clave	4 dígitos
Tipo de clave	Numérica
Número de clientes	100 000
Número de intentos de acceso luego de los cuales se bloquea la cuenta	3
Longitud de la clave dinámica para transferencias	3 dígitos
Número de intentos de transferencia antes del bloqueo de la cuenta	5
VALORES OBTENIDOS PARA LAS VARIABLES ANALIZADAS	
Probabilidad de acceso a una cuenta	0.0003
Probabilidad de acceder al menos a una cuenta	≈1
Número de cuentas accedidas en cada intento	30
Probabilidad de realizar una transferencia una vez logrado acceso	0.005
Probabilidad de lograr acceso 'total' a una cuenta	1.5×10^{-6}
Probabilidad de acceder 'totalmente' al menos a una cuenta	0.13929
Número de cuentas accedidas 'totalmente' en cada intento	≈0

Tabla 3.9 Factores, niveles y valores obtenidos de variables para el ejemplo del banco Tipo I

III.5.2 Banco Tipo II

De acuerdo a los requerimientos preliminares y tomando en cuenta las características de los tipos de bancos citados en las tablas comparativas (Tabla 3.1, 3.2 y 3.3), el modelo matemático del banco II es prácticamente el mismo que los de Tipo I, las diferencias son minúsculas (radican en las propiedades de las claves para realizar transferencias). Sin embargo, detallaremos su modelo matemático y para ello nos valdremos de los siguientes niveles literales para los factores involucrados (Tabla 3.10).

FACTOR	NIVEL
Longitud de la clave (en dígitos)	*n*
Tipo de clave	Numérica
Número de clientes del banco	*N*
Número de intentos de acceso luego de los cuales se bloquea la cuenta	*M*
Longitud de la clave dinámica para transferencias (en dígitos)	*K*
Número de intentos de transferencia antes del bloqueo de la cuenta	*I*

Tabla 3.10 Factores y niveles literales para un banco Tipo II

En este caso, como en el de los bancos Tipo I, se necesita también de una clave adicional para realizar transferencias a cuentas ajenas dentro del mismo banco o a cuentas en otros bancos. Esta clave es de *k* dígitos y – una vez que se han introducido los detalles de la transferencia – es enviada vía correo electrónico al cliente quien debe ingresar dicha contraseña para confirmar y llevar a cabo finalmente la transacción. La particularidad de esta clave es que existe un número indefinido de oportunidades de acceso, es decir la cuenta no se bloquea al ingresar repetitivamente claves erróneas.

Por otro lado, los pagos de servicios y compras electrónicas – de tarjetas prepago para celular, por ejemplo – no requieren la clave adicional.

III.5.2.1 Probabilidad de acceso a una cuenta

Al ser la clave de *n* dígitos, un usuario podría tener 10^n combinaciones distintas, asumiendo que el nombre de usuario corresponde efectivamente a un cliente del banco y que se escoge aleatoriamente la clave de acceso, la probabilidad acceder a la cuenta es:

$$P_{acc2} = \text{número de intentos de acceso} \times \frac{1}{\#\,\text{claves posibles}} = m \times \frac{1}{10^n} = \frac{m}{10^n}, \qquad [3.25]$$

esto considerando que la cuenta se bloquea luego de *m* intentos erróneos.

Entonces, la probabilidad de no acceder a una cuenta será:

$$P_{noacceso} = 1 - P_{acc2} = 1 - \frac{m}{10^n} \qquad [3.26]$$

Ahora, si se tiene un número *N* de clientes, la probabilidad de no acceder a ninguna cuenta se expresa mediante la siguiente ecuación:

$$P_{noaccesoNcuentas} = P_{noacceso1} \cdot P_{noacceso2} \cdot P_{noacceso3} \cdot \ldots \cdot P_{noaccesoN} = \left(P_{noacceso}\right)^N \qquad [3.27]$$

$$P_{noaccesoNcuentas} = \left(1 - P_{acc2}\right)^N = \left(1 - \frac{m}{10^n}\right)^N \qquad [3.28]$$

De esto, la probabilidad de acceder al menos a una cuenta, para dicho conjunto de *N* clientes, es:

$$P(x \geq 1) = 1 - P_{noaccesoNcuentas} = 1 - \left(1 - \frac{m}{10^n}\right)^N \qquad [3.29]$$

El número esperado de cuentas a las que se logra acceso durante un ataque es:

$$\#\,\text{cuentas accedidas} = P_{acc2} \times N = \frac{m}{10^n} \times N \qquad [3.30]$$

Entonces, para ejemplificar este modelo tomaremos en cuenta los siguientes niveles para los factores seleccionados:

FACTOR	NIVEL
Longitud de la clave (en dígitos)	$n = 4$
Tipo de clave	Numérica
Número de clientes del banco	$N = 1\ 000\ 000$
Número de intentos de acceso luego de los cuales se bloquea la cuenta	$m = 5$

Tabla 3.11 Factores y niveles para ejemplo de probabilidad de acceso a un banco Tipo II

Como la clave tiene una longitud de 4 dígitos y la cuenta se bloquea luego de 5 intentos fallidos, y si además se asume que el nombre de usuario corresponde efectivamente a un cliente de la institución y que se escoge al azar dicha clave, la probabilidad de acceso – a partir de [3.25] – es:

$$P_{acc2} = 5 \times \frac{1}{10\,000} = 0.0005 \qquad [3.31]$$

De esto, la probabilidad de no acceder a una cuenta será:

$$P_{noacceso} = 1 - P_{acc2} = 1 - 0.0005 = 0.9995 \qquad [3.32]$$

Por tanto, la probabilidad de ganar acceso al menos a una cuenta, si se tiene un conjunto de 1 000 000 de clientes, tomando en cuenta [3.29] es:

$$P(x \geq 1) = 1 - (0.9995)^{1\,000\,000} = 1 - 6.28715^{-218} \approx 1 \qquad [3.33]$$

De allí se puede afirmar que existe una gran probabilidad – prácticamente igual a uno – de acceder al menos a una cuenta.

Continuando con el ejemplo, se debe calcular el número esperado de cuentas accedidas durante un ataque; de [3.30]:

$$\#\text{cuentas accedidas} = 0.0005 \times 1\ 000\ 000 = 500 \qquad [3.34]$$

En cada nuevo ataque aleatorio – teóricamente – se lograría acceder a 500 nuevas cuentas.

III.5.2.2 Probabilidad de realizar una transferencia

La clave adicional para transferencias consta de k dígitos pero no existe un número límite de veces que se pueda ingresar tal clave, por tanto:

$$P_{transf} = \text{número intentos de transferencia} \times \frac{1}{\#\,\text{claves posibles}} = 10^k \times \frac{1}{10^k} = 1 \qquad [3.35]$$

En este caso se descubre, sorprendentemente, que la probabilidad de realizar una transferencia de fondos es altísima, pues el atacante intenta todas las combinaciones posibles hasta lograr dar con la clave asignada para la transacción.

Por lo tanto, la longitud de esta clave lo único que haría es influir en el tiempo en el que se lograría adivinar la clave para transferencias. Como ejemplo, si para este tipo de banco tomamos k = 10, significaría que se tendría 10 000 000 000 posibles claves y el atacante podría realizar el mismo número de intentos hasta adivinar la clave. Ya se mencionó que la probabilidad de lograr realizar la transferencia sería 1 (uno).

III.5.2.3 Probabilidad de lograr acceso 'total' a la cuenta

Por la particularidad de este tipo de banco, la probabilidad de tener acceso 'total' a la cuenta es la misma que se tenía para lograr únicamente acceso a la cuenta sin intentar quebrar la llave para transferencias. Esto se da porque la probabilidad de

realizar una transferencia es prácticamente uno, por tanto no afecta a los cálculos. Si desarrollamos lo anteriormente mencionado tenemos que:

$$P_{accesotot} = P_{acc2} P_{transf} \quad [3.36]$$

La probabilidad de no acceder 'totalmente' a una cuenta es:

$$P_{noaccesotot} = 1 - P_{acc2} P_{transf} \quad [3.37]$$

Si se reemplaza [3.25] y [3.35] en [3.37] se tiene:

$$P_{noaccesotot} = 1 - \frac{m}{10^n} \quad [3.38]$$

Entonces, la probabilidad de acceder al menos a una cuenta, dado el conjunto de N clientes, es:

$$P(x \geq 1) = 1 - P_{noaccesotot}{}^N \quad [3.39]$$

$$P(x \geq 1) = 1 - \left(1 - \frac{m}{10^n}\right)^N \quad [3.40]$$

Por otro lado, el número esperado de cuentas accedidas 'totalmente' durante un ataque de fuerza bruta es:

$$\#\text{cuentas accedidas 'totalmente'} = P_{acc2} P_{transf} \times N \quad [3.41]$$

A partir de [3.25] y [3.35] se obtiene:

$$\#\text{cuentas accedidas 'totalmente'} = \frac{m}{10^n} \times N \quad [3.42]$$

Para ilustrar este modelo a través de un ejemplo tomamos los valores que se muestran en la Tabla 3.12 (los mismos de los apartados 3.5.2.1 y 3.5.2.2):

FACTOR	NIVEL
Longitud de la clave (en dígitos)	$n = 4$
Tipo de clave	Numérica
Número de clientes del banco	$N = 1\ 000\ 000$
Número de intentos de acceso luego de los cuales se bloquea la cuenta	$m = 5$
Longitud de la clave dinámica para transferencias (en dígitos)	$k = 10$
Número de intentos de transferencia antes del bloqueo de la cuenta	$i = \infty$

Tabla 3.12 Factores y niveles para ejemplo de probabilidad de acceso total a un banco Tipo II

Tomando en cuenta [3.36], [3.25] y [3.35] se obtiene que la probabilidad de lograr acceso 'total' a una cuenta es:

$$P_{accesotot} = \frac{5}{10000} \times 1 = 0.0005 \qquad [3.43]$$

Y de [3.33] la probabilidad de no tener acceso a la cuenta es:

$$P_{noaccesotot} = 1 - 0.0005 = 0.9995 \qquad [3.44]$$

Luego si se calcula la probabilidad de acceder al menos a una cuenta, dado el conjunto de 1 000 000 de clientes, se tiene:

$$P(x \geq 1) = 1 - (0.9995)^{1\,000\,000} = 1 - 6.28715^{-218} \approx 1 \qquad [3.45]$$

Por tanto, existe una gran probabilidad de acceder 'totalmente' a una cuenta al menos. Esto debido a que la segunda clave se la puede adivinar prácticamente siempre.

A partir de [3.42] se obtiene el número esperado de cuentas accedidas 'totalmente' durante un ataque de fuerza bruta:

$$\#\text{cuentas accedidas 'totalmente'} = 0.0005 \times 1\,000\,000 = 500 \qquad [3.46]$$

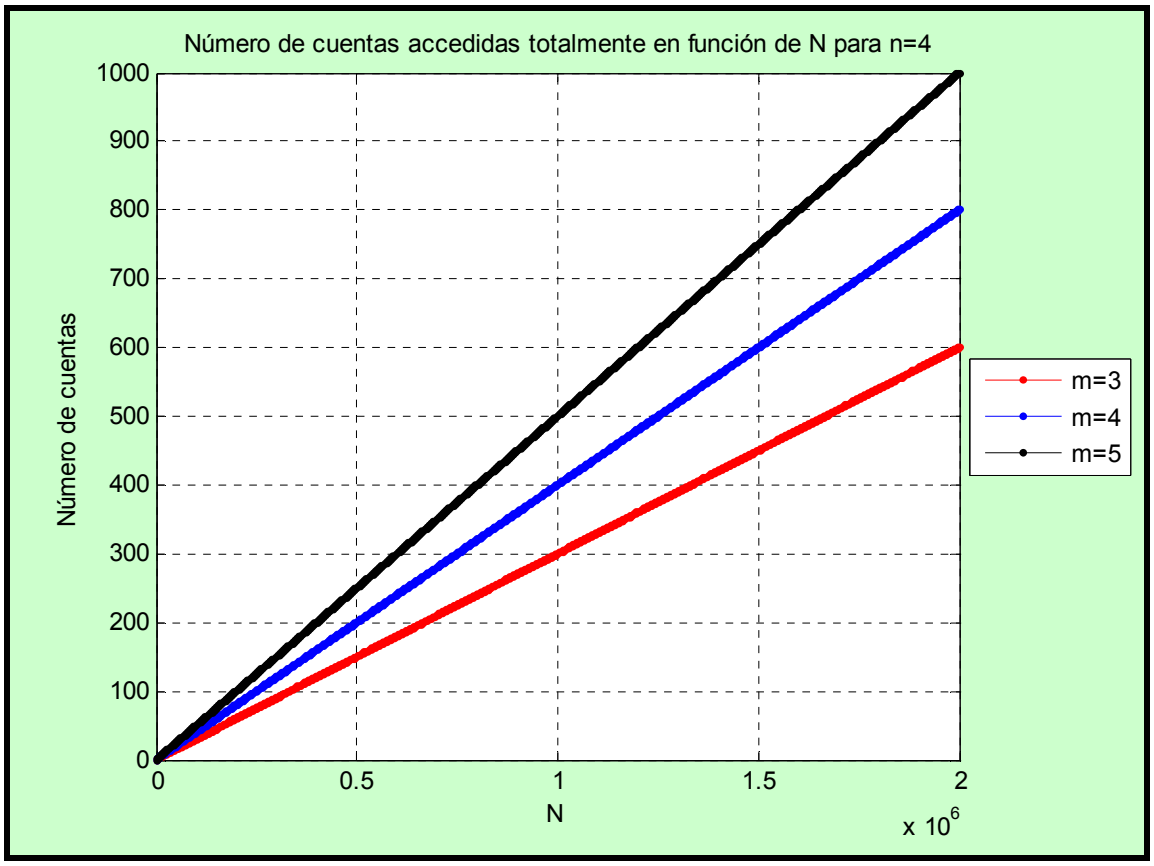


Figura 3.6 Valores teóricos del número de cuentas accedidas totalmente para un banco Tipo II

Nuevamente, como se realizó para los bancos Tipo I, se grafica el número de cuentas accedidas totalmente versus el número de usuarios del banco para varios *m*. En este caso, el valor de *k* no se toma en cuenta pues, por las características de este tipo de banco, éste no influye en esta variable. Las curvas muestran que se tendría un número significativo de cuentas a las que se logra acceso; para 1 000 000 de usuarios se accederían a 300, 400 ó 500 cuentas con *m=3,4 ó 5,* respectivamente.

Adicionalmente, en la Tabla 3.13 se detallan los factores, niveles y resultados obtenidos para el modelo matemático del banco Tipo II.

FACTORES Y NIVELES SELECCIONADOS	
Longitud de la clave	4 dígitos
Tipo de clave	Numérica
Número de clientes	1 000 000
Número de intentos de acceso luego de los cuales se bloquea la cuenta	5
Longitud de la clave dinámica para transferencias	10 dígitos
Número de intentos de transferencia antes del bloqueo de la cuenta	∞
VALORES OBTENIDOS PARA LAS VARIABLES ANALIZADAS	
Probabilidad de acceso a una cuenta	0.0005
Probabilidad de acceder al menos a una cuenta	≈1
Número de cuentas accedidas en cada intento	500
Probabilidad de realizar una transferencia una vez logrado acceso	1
Probabilidad de lograr acceso 'total' a una cuenta	0.0005
Probabilidad de acceder 'totalmente' al menos a una cuenta	1
Número de cuentas accedidas 'totalmente' en cada intento	500

Tabla 3.13 Factores, niveles y valores obtenidos de variables para el ejemplo del banco Tipo II

III.5.3 Caso alternativo

Para este modelo alternativo, como ya se mencionó al inicio de la descripción de los modelos matemáticos, se toma en cuenta que la clave de acceso al banco es alfanumérica, con distinción entre caracteres minúsculas y mayúsculas, lo que da un total de 62 caracteres posibles (de acuerdo al alfabeto inglés: 26 letras mayúsculas, 26 letras minúsculas y 10 dígitos numéricos).

Este incremento en el número de caracteres que podrían integrar la clave sin duda hace que ésta sea mucho más difícil de quebrar mediante un ataque de fuerza bruta, y una alternativa podría ser plantear un *ataque de diccionario*[14]. Pero para caracterizar este modelo, y en concordancia con los anteriormente descritos, se realizarán los cálculos tomando en cuenta un *ataque de fuerza bruta*. Se calculará la probabilidad de acceso al menos a una cuenta y el número de cuentas accedidas.

Entonces, para el efecto, primero se realizará el desarrollo literal del modelo (tomando en cuenta la misma nomenclatura ya utilizada) y luego se lo ilustrará con un ejemplo. Por tanto, los niveles son los mostrados en la Tabla 3.14.

FACTOR	NIVEL
Longitud de la clave (en dígitos)	*N*
Tipo de clave	Alfanumérica
Número de clientes del banco	*N*
Número de intentos de acceso luego de los cuales se bloquea la cuenta	*M*

Tabla 3.14 Factores y niveles literales para el caso alternativo

[14] Ataque de fuerza bruta que usa palabras comunes como posibles contraseñas y que puede proporcionar una manera más eficiente para obtener la clave del usuario. (Computer Desktop Encyclopedia [Internet] [Actualizado 2007]).

Para empezar, se debe calcular el número de claves posibles de *n* caracteres que se pueden formar con un 'alfabeto' de 62 (variaciones posibles de 62 elementos tomados de *n* en *n*). El número de claves que se podría tener es:

$$\#\,\text{claves posibles} = \left(\#\,\text{de caracteres del alfabeto}\right)^n = 62^n \qquad [3.47]$$

Entonces, asumiendo que el nombre de usuario corresponde a un cliente del banco, que se escoge aleatoriamente la clave alfanumérica de acceso y que se dispone de *m* intentos, la probabilidad acceder a la cuenta es:

$$P_{acc2} = m \times \frac{1}{62^n} = \frac{m}{62^n} \qquad [3.48]$$

De donde, la probabilidad de no acceder a una cuenta viene dada por:

$$P_{noacceso} = 1 - P_{acc2} = 1 - \frac{m}{62^n} \qquad [3.49]$$

La probabilidad de no acceder a ninguna cuenta de un conjunto de *N* clientes es:

$$P_{noaccesoNcuentas} = \left(P_{noacceso}\right)^N = \left(1 - \frac{m}{62^n}\right)^N \qquad [3.50]$$

Entonces, teniendo dicho conjunto de *N* clientes, la probabilidad de acceder al menos a una cuenta es:

$$P(x \geq 1) = 1 - P_{noaccesoNcuentas} = 1 - \left(P_{noacceso}\right)^N = 1 - \left(1 - \frac{m}{62^n}\right)^N \qquad [3.51]$$

El número esperado de cuentas accedidas durante este ataque sería:

$$\#\,\text{cuentas accedidas} = P_{acc2} \times N = \frac{m \cdot N}{62^n} \qquad [3.52]$$

Ahora, para la realización del ejemplo de cálculo se utilizarán los siguientes niveles de los factores:

FACTOR	NIVEL
Longitud de la clave (en dígitos)	n = 4
Tipo de clave	Alfanumérica
Número de clientes del banco	N = 1 000 000
Número de intentos de acceso luego de los cuales se bloquea la cuenta	m = 5

Tabla 3.15 Factores y niveles para ejemplo de probabilidad de acceso en el caso alternativo

Se calcula el número de claves posibles de longitud n=4 que se pueden formar con un conjunto de 62 caracteres disponibles:

$$\#\,\text{claves posibles} = 62^4 = 14\,776\,336 \qquad [3.53]$$

La probabilidad acceder a una cuenta, disponiendo de m=5 intentos es:

$$P_{acc2} = 5 \times \frac{1}{14\,776\,336} = 3.38379 \times 10^{-7} \qquad [3.54]$$

Por tanto, la probabilidad de no acceder a una cuenta:

$$P_{noacceso} = 1 - P_{acc2} = 1 - 3.38379 \times 10^{-7} = 0.99999966162 \qquad [3.55]$$

Dado un conjunto de 1 000 000 de clientes, la probabilidad de lograr acceso al menos a una cuenta – tomando en cuenta [3.51] – es:

$$P(x \geq 1) = 1 - (0.99999966162)^{1\,000\,000} = 1 - 0.71292 = 0.28708 \qquad [3.56]$$

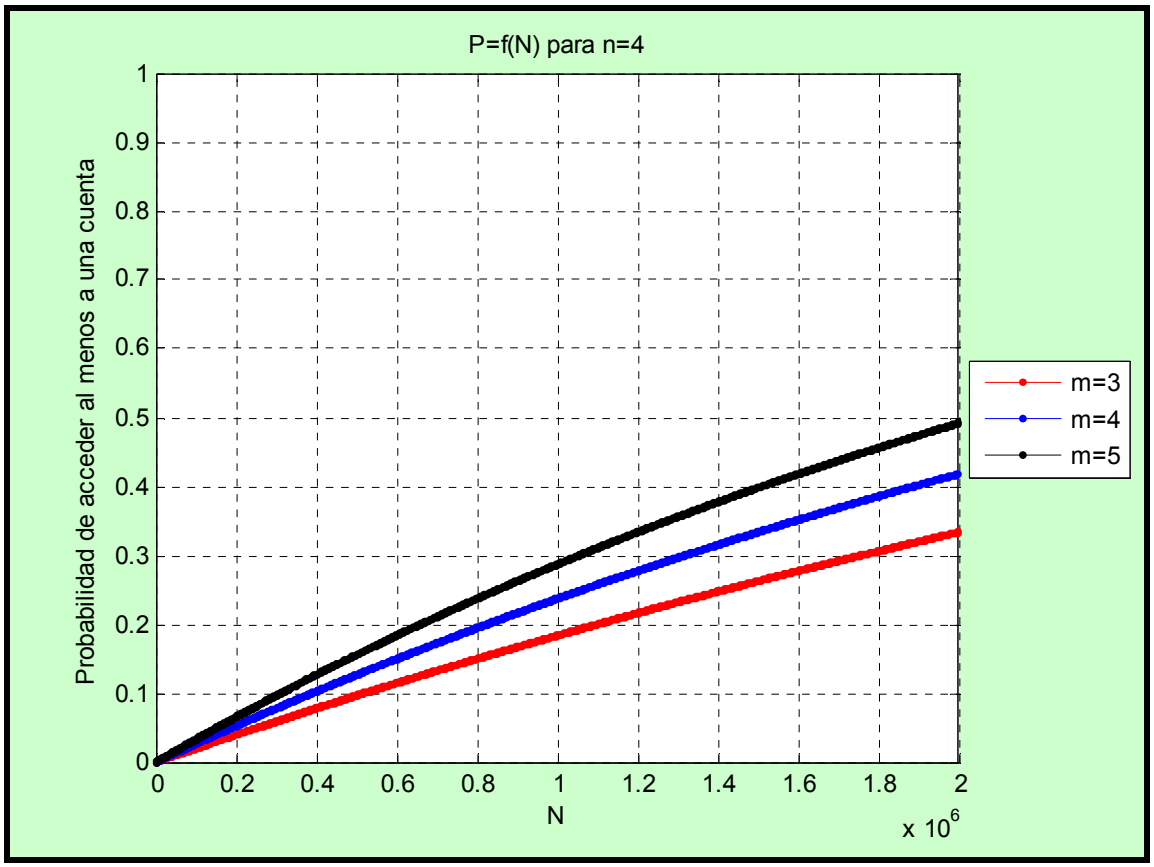


Figura 3.7 Valores teóricos de la probabilidad de acceder al menos a una cuenta para el caso alternativo

Se grafica en la Figura 3.7 la probabilidad de acceder al menos a una cuenta versus el número de usuarios del banco para distinto número de intentos de acceso antes del bloqueo de la cuenta. Como se observa, en ningún caso se obtiene una probabilidad siquiera cercana a 1 (uno); para un banco de 2 000 000 de usuarios, con $m=5$, la probabilidad no llega a 0,5.

Por tanto, para el ejemplo presentado, existe muy poca probabilidad de acceder al menos a una cuenta del conjunto de un millón de cuentas disponibles.

Para terminar el ejemplo, el número esperado de cuentas a las que se accedería durante este ataque es:

$$\#\text{cuentas accedidas} = 3.38379 \times 10^{-7} \times 1\,000\,000 = 0.33 \approx 0 \qquad [3.57]$$

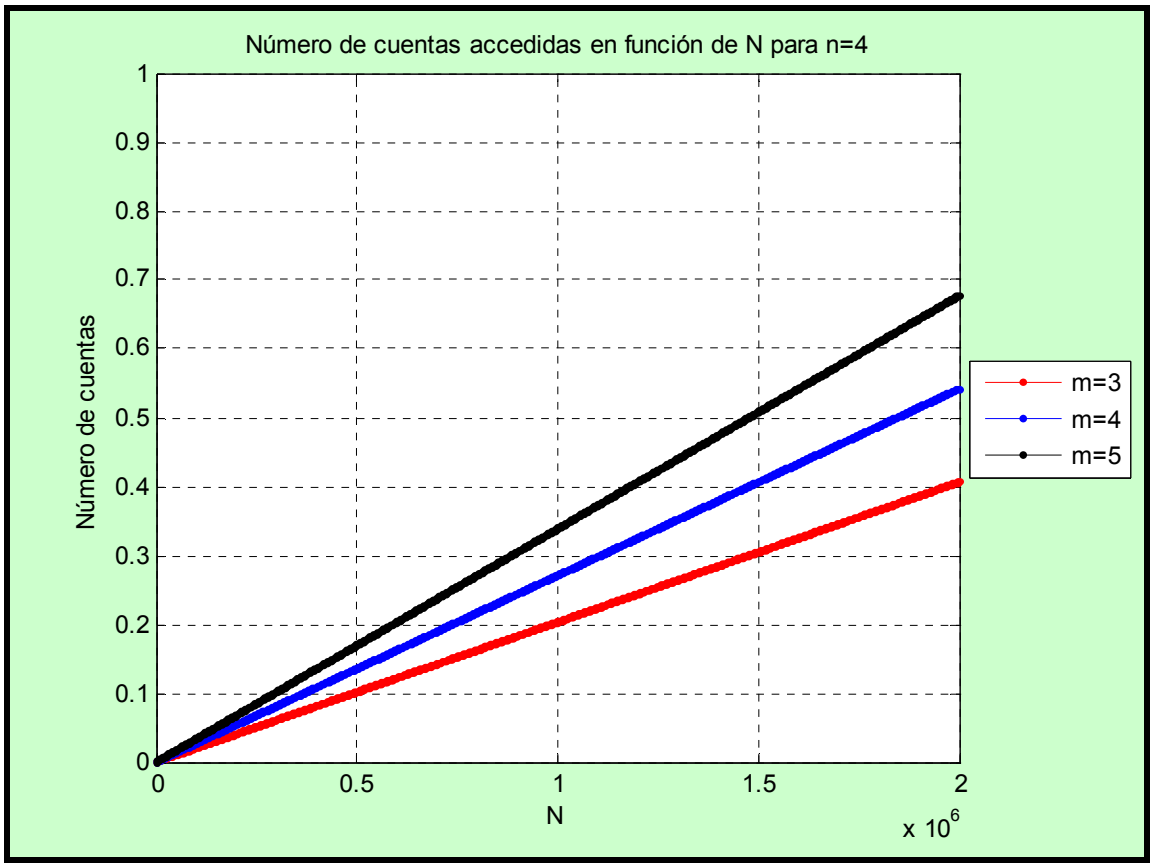


Figura 3.8 Valores teóricos del número de cuentas accedidas para el caso alternativo

La figura precedente muestra que para cualquiera de los niveles de *m* no se logra el acceso ni siquiera a una cuenta de los cientos de miles o millones disponibles. Para un banco de 2 000 000 de usuarios, con *m=5*, el número de cuentas accedidas es inferior a 0,7.

Con la finalidad de presentar de una manera resumida y detallada este caso alternativo, con sus factores y niveles, en la tabla de la página siguiente se resumen los valores obtenidos para las variables analizadas.

FACTORES Y NIVELES SELECCIONADOS	
Longitud de la clave	4 caracteres
Tipo de clave	Alfanumérica
Número de clientes	1 000 000
Número de intentos de acceso luego de los cuales se bloquea la cuenta	5
VALORES OBTENIDOS PARA LAS VARIABLES ANALIZADAS	
Probabilidad de acceso a una cuenta	3.38379×10^{-7}
Probabilidad de acceder al menos a una cuenta	0.28708
Número de cuentas accedidas en cada intento	≈0

Tabla 3.16 Factores, niveles y valores de variables obtenidos para el ejemplo del caso alternativo

Se demuestra entonces que es realmente imposible tener acceso a una cuenta cuando se tiene contraseñas de cuatro caracteres alfanuméricos utilizando un *ataque de fuerza bruta.* Esta particularidad sin duda se podría tomar en cuenta para fortalecer los sistemas de autenticación de la banca electrónica.

III.5.4 Comparación de los modelos

Finalmente, en este apartado, se muestra un resumen comparativo tabulado de los análisis realizados para cada uno de los tipos de banco y el caso alternativo:

FACTORES Y NIVELES SELECCIONADOS	TIPO I	TIPO II	ALTERNATIVO
Longitud de la clave	4 dígitos	4 dígitos	4 caracteres
Tipo de clave	Numérica	Numérica	Alfanumérica
Número de clientes	100 000	1 000 000	1 000 000
Número de intentos de acceso luego de los cuales se bloquea la cuenta	3	5	5
Longitud de la clave dinámica para transferencias	3 dígitos	10 dígitos	-
Número de intentos de transferencia antes del bloqueo de la cuenta	5	∞	-
VALORES OBTENIDOS PARA LAS VARIABLES ANALIZADAS	**TIPO I**	**TIPO II**	**ALTERNATIVO**
Probabilidad de acceso a una cuenta	0.0003	0.0005	3.38379×10^{-7}
Probabilidad de acceder al menos a una cuenta	≈ 1	≈ 1	0.28708
Número de cuentas accedidas en cada intento	30	500	≈ 0
Probabilidad de realizar una transferencia una vez logrado acceso	0.005	1	-
Probabilidad de lograr acceso 'total' a una cuenta	1.5×10^{-6}	0.0005	-
Probabilidad de acceder 'totalmente' al menos a una cuenta	0.13929	1	-
Número de cuentas accedidas 'totalmente' en cada intento	≈ 0	500	-

Tabla 3.17 Factores, niveles y valores de variables obtenidos para los tres casos presentados

En la tabla de la página anterior se puede notar que para contraseñas de igual longitud, mientras mayor número de intentos se permitan antes de bloquear la cuenta se tiene mayor probabilidad de acceder a ella. Además, cuantos más clientes se tenga, mayor es la cantidad de cuentas que pueden ser accedidas en un ataque aleatorio. Por otro lado, considerando el caso alternativo, se nota que el tener claves de tipo alfanumérico disminuye significativamente la probabilidad de acceder a una cuenta, pues se tiene 10 000 contraseñas posibles ($n = 4$) con una clave de tipo numérico versus más de 14 millones de claves posibles para las de tipo alfanumérico. Todas estas comparaciones basadas en la teoría se podrán verificar, de manera práctica, con la simulación que se detalla en el siguiente capítulo.

CAPÍTULO 4

IV. DESCRIPCIÓN DEL EXPERIMENTO Y SIMULACIÓN

El experimento – en formato de simulación – constituye la parte más importante del presente trabajo. La validación de la simulación se la realizará con los resultados obtenidos con el modelo matemático planteado en el capítulo anterior. La realización del mismo y las consideraciones con las que se lo desarrolle serán vitales para las valoraciones finales.

Desarrollaremos a continuación la descripción del experimento, su diseño, así como la mención de las variables, niveles y factores que intervienen en él. Luego de detallar la parte 'teórica' del experimento se pasará a explicar los objetivos de la simulación, ciertas consideraciones, una descripción general, sus flujogramas para finalmente presentar el código empleado para su realización. Es preciso mencionar que los resultados obtenidos se muestran y analizan en el próximo capítulo.

IV.1 EL EXPERIMENTO

El experimento a realizarse consiste en un *ataque de fuerza bruta* a todas las posibles cuentas de un sistema bancario ficticio, tomando en cuenta las consideraciones y limitaciones detalladas en los modelos matemáticos del Capítulo 3 – entiéndase longitud de contraseñas, número de intentos de acceso antes de bloqueo de cuenta, etc.–. Cabe destacar que para fines prácticos, el experimento se centra únicamente en los bancos clasificados como de Tipo I (véase apartados 3.1 y 3.2 para detalle de características) y se recrea dicho escenario en un ambiente sencillo. Se ha elegido este modelo debido a que es el más común entre las instituciones bancarias del país y

también ya que es posible obtener datos reales de nombres de usuario pues éstos corresponden a los números de cédula del cliente del banco. Se explicará más adelante cómo se puede obtener dicha información de una manera totalmente confiable y asequible.

IV.1.1 Diseño del experimento

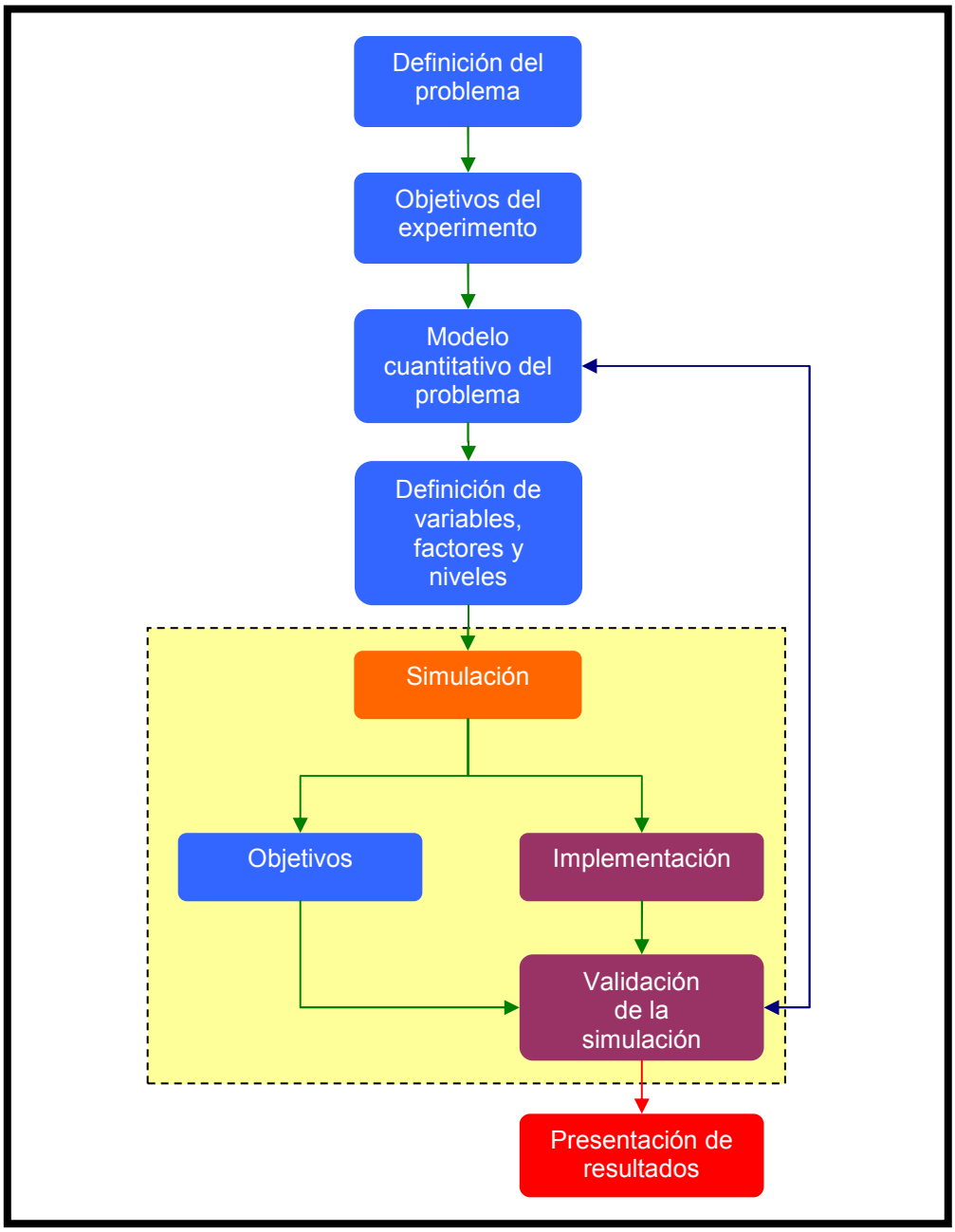


Figura 4.1 Diseño del experimento

Previa la realización de cualquier experimento, es necesario pasar por la etapa de su diseño el mismo que nos dará los lineamientos generales a los cuales debe regirse. En la Figura 4.1 se presenta el diseño del presente experimento en forma de diagrama de flujo.

El diseño realizado toma en cuenta el tipo de experimento que se va a realizar: una simulación de un sistema; por ello son necesarios algunos pasos previos, sobre todo en el ámbito de definición de alcance y objetivos del mismo. En el diagrama expuesto se muestran todas las etapas por las que se ha creído conveniente se debe cursar para la exitosa ejecución del experimento. A continuación se profundizará en cada una de las etapas, con excepción de la tercera, la cual ya ha sido motivo de un amplio estudio en el capítulo anterior. Las etapas mostradas dentro del recuadro amarillo correspondientes a la simulación se estudian y describen en el apartado 4.2; mientras que la etapa final corresponde al siguiente capítulo, como ya se había acotado.

IV.1.1.1 Descripción del problema

El problema consiste en evaluar cuán fácil o difícil resulta lograr el acceso a una cuenta de un sistema bancario. Con el experimento se pretende recrear un sistema cliente – servidor, el cual se asemeje en magnitud a un banco real; el sistema deberá implementar un ataque de fuerza bruta para intentar lograr el acceso a la mayor cantidad de cuentas. Para esto se necesita crear un sistema sencillo que simule varias máquinas atacantes que se autentiquen en el servidor del banco y luego evaluar la cantidad de cuentas a las cuales se accedió.

Es preciso mencionar que únicamente se simulará el acceso a la cuenta bancaria sin intentar lograr quebrar la clave para transferencias pues la posible y primera vulnerabilidad que el atacante intentaría explotar es precisamente el acceder a la cuenta.

Una vez que se ha logrado el acceso no haría falta siquiera que el intruso realice una transferencia: la seguridad del sistema ya se habría vulnerado.

IV.1.1.2 Objetivos del experimento

A través del experimento se pretende satisfacer los objetivos citados a continuación:

- Recrear un sistema bancario cliente – servidor.
- Validar el experimento con los modelos matemáticos expuestos en el Capítulo 3, dando especial importancia al del banco Tipo I.
- Estudiar cómo los distintos niveles de los factores afectan a las variables del experimento.
- Obtener resultados que permitan realizar una valoración cuantitativa de cuán seguros resultaron los sistemas analizados.

IV.1.1.3 Definición de variables, factores y niveles

Otra etapa, previa la realización del experimento, consiste en definir las variables, factores y niveles que se tomarán en cuenta en el mismo para su desarrollo y posterior evaluación de resultados. El experimento se centra en verificar cuán fácil es quebrar el acceso a las cuentas del banco y cuánto tardaría tal intrusión o intrusiones; por lo tanto, las variables a analizarse son (1) el número de cuentas a las que se logra el acceso y (2) el tiempo en el que se logra el mismo.

En la Tabla 4.1 se detallan todas las variables que intervienen en el experimento, los factores que las afectan y los niveles de los mismos.

La primera variable citada es el 'número de cuentas a las que se intentará el acceso', la misma que depende de la cobertura con la que cuente la supuesta entidad

bancaria; los niveles de dicho factor serían centenas de miles o millones de cuentas, particularmente se han elegido 100 000, 500 000 ó 1 000 000 de usuarios.

VARIABLES	FACTORES	NIVELES
Número de cuentas a las que se intenta el acceso	Alcance con que cuente la entidad bancaria (local, regional, nacional e incluso internacional)	100 000, 500 000 ó 1 000 000 de usuarios
Número de cuentas a las que se logra acceso	Longitud y tipo de nombre de usuario	10 caracteres numéricos
	Longitud de clave	4, 5 ó 6 dígitos
	Número máximo de intentos de acceso	3, 4 ó 5 intentos
Tiempo de acceso	Cantidad de máquinas atacantes	10, 20, 30 ó 40 máquinas
	Tiempo de respuesta del servidor del banco	Décimas de segundo o segundos
	Distribución de tiempo entre intentos	Unidades o decenas de segundos

Tabla 4.1 Variables, factores y niveles para la simulación

Las dos variables siguientes hacen referencia las que anteriormente habíamos etiquetado como ‘variables a analizarse’. El ‘número de cuentas a las que se logra el acceso’ es otra variable que interviene en el experimento, – correspondiente a los resultados de éste –; un factor para esta variable es la longitud y tipo de nombre de usuario, y su nivel será: 10 dígitos, de tipo numérico; otro factor es la longitud de la clave con niveles de 4, 5 ó 6 dígitos; y, el número máximo de intentos de acceso cuyos niveles serán 3, 4 ó 5 intentos.

Y, finalmente, la variable ‘tiempo de acceso’ depende de tres factores, la cantidad de máquinas atacantes, el tiempo de respuesta del servidor bancario y la

distribución del tiempo entre intentos. La cantidad de máquinas atacantes tiene como niveles 10, 20, 30 ó 40 máquinas, dado por las restricciones del paquete CSIM en su versión estudiantil. Los niveles para el segundo factor son décimas de segundos o segundos. Y finalmente, el tiempo entre intentos tendrá como niveles unidades o decenas de segundos.

IV.2 LA SIMULACIÓN

La simulación constituye una representación de un sistema bancario cliente – servidor; en ella se emula un ataque de fuerza bruta, varias máquinas atacantes intentan lograr acceso a un servidor, el cual contiene los nombres de usuario y contraseñas de los clientes del banco. Para esto tanto las máquinas como el servidor serán simuladas en un ambiente simple, a través de programación en lenguaje C e implementando pseudoparalelismo[15] para representar el sistema de una manera lo más aproximada a la realidad, con varias máquinas intentando el acceso a la vez.

IV.2.1 Objetivos de la simulación

Los objetivos que se persiguen en la simulación difieren de los objetivos del experimento en que los aquí mencionados involucran la parte práctica del experimento, sus herramientas y mecanismos de ejecución. De acuerdo a lo mencionado, la simulación está orientada a:

- Crear una base de datos sencilla en archivos de texto plano que contenga los usuarios y las contraseñas a utilizarse en el sistema.

[15] Ejecución secuencial de varios procesos o tareas en un mismo procesador con rápida conmutación lo que causa apariencia de que se ejecutan simultáneamente.

- Implementar mediante programación estructurada – en lenguaje C – un sistema simple que emule el mecanismo de acceso a un banco genérico del Ecuador.
- Recrear un ataque de fuerza bruta al sistema con varias máquinas atacantes y un servidor en el cual aquellas se autentiquen para lograr acceso.
- Emplear pseudoparalelismo, mediante el paquete CSIM, para la recreación del acceso simultáneo de varias máquinas al servidor.
- Modelar el tiempo de respuesta del servidor bancario y el tiempo entre intentos de acceso como tiempo real y tener control de una variable de tiempo de simulación; todo esto con el fin de determinar cuánto tardaría lograr acceso a una cuenta.
- Efectuar el control del número de intentos de acceso y la cantidad de éxitos alcanzados para su posterior evaluación y presentación de resultados tabulados y gráficos de los mismos.

IV.2.2 Herramientas empleadas

En los objetivos se ha mencionado vagamente algunas de las herramientas que se utilizarán para la simulación. Previo la realización de ésta, se hace necesaria la obtención de información para la creación de la base de datos; una vez creada se procede a simular el ataque al sistema bancario ficticio. A continuación se detallan las herramientas y procedimientos empleados tanto en la creación de las bases de datos cuanto en la propia simulación.

IV.2.2.1 Generación de las bases de datos

Como es lógico, se necesita contar con dos bases de datos para la realización de la simulación. La primera que corresponda a todos los posibles usuarios – que será

propiedad de las máquinas atacantes – y la segunda que emulará a la base de datos del banco, con sus clientes y claves de acceso.

Un método bastante sencillo de obtención de nombres de usuario, léase números de cédula válidos, es la generación de una especie de robot *web*. Se trata de un pequeño programa que obtiene, mediante consultas repetitivas, números de cédula reales del portal del Consejo Nacional Electoral. Este *script* se ha programado en PHP.

Dicho programa **servirá únicamente como una muestra** de lo sencillo que puede resultar, para alguien conocedor de la materia, el obtener números de cédula válidos de un sitio *web* accesible a cualquier persona y utilizar dichos datos como nombres de usuario para acceder a cuentas bancarias. Se trata, en realidad, de una debilidad del sistema. **En ningún momento se ha intentado ni se intentará obtener ni utilizar una gran cantidad de cédulas válidas del portal del CNE.** Se hace esta aclaración tomando en cuenta el Artículo 58[16] de la "Ley de comercio electrónico, firmas electrónicas y mensajes de datos" del Ecuador.

En la práctica, para generar dichas bases de datos se ha utilizado MATLAB; un archivo de formato '.m' permitirá crear los arreglos que contengan los datos necesarios para la simulación.

IV.2.2.2 Programa de simulación

La programación de la simulación se realiza en lenguaje C utilizando el aplicativo Microsoft Visual Studio 6.0 que permite la compilación, y ejecución

[16] Art. 58.- A continuación del Art. 202 (del Código Penal), inclúyanse los siguientes artículos innumerados: "… Art. ….- *Obtención y utilización no autorizada de información*.- La persona o personas que obtuvieren información sobre datos personales para después cederla, publicarla, utilizarla o transferirla a cualquier título, sin la autorización de su titular o titulares, serán sancionadas con pena de prisión de dos meses a dos años y multa de mil a dos mil dólares de los Estados Unidos de Norteamérica." (Ley de comercio electrónico, firmas electrónicas y mensajes de datos del Ecuador [Internet] [Actualizado 2002, Abril]).

requeridos. Adicionalmente, dado el escenario que se tiene (recrear un ataque real) se ha visto necesaria la adición del paquete CSIM en su versión 19 para C, de Mesquite Software, que permite construir modelos o sistemas complejos y emular procesos paralelos (implementa pseudoparalelismo).

IV.2.3 Consideraciones generales

La presente sección describe los lineamientos que deberá seguir la simulación, tanto en la generación de las bases de datos cuanto en su programación y posterior ejecución. Se ha tratado de explicar en detalle todas las condiciones que deben presentarse para la realización del experimento.

IV.2.3.1 Bases de datos

Para tener un escenario bastante cercano a la realidad y sabiendo que en los bancos que hemos categorizado como de Tipo I el nombre de usuario es el número de cédula, debemos detallar previamente cómo se forma el número de identificación de cada persona.

IV.2.3.1.1 ¿Cómo verificar la validez de una cédula de identidad?

Las cédulas de identidad en nuestro país constan de diez dígitos, los dos primeros hacen referencia a la provincia de origen (e.g. ‘01’ para Azuay, ‘09’ para Guayas, ‘10’ para Imbabura, ‘17’ para Pichincha, etc.), los 7 siguientes son asignados en orden – bien podría llamarse cronológico – y el último dígito, el décimo, es un dígito verificador calculado utilizando el método de verificación módulo 10.

En este método, se toma todo el número a verificar (es decir las 9 cifras asignadas al ciudadano) y a cada dígito se multiplica de derecha a izquierda, desde el

menos significativo al más significativo, al primero por 2 y al segundo por 1, el siguiente por 2 y el próximo por uno y así sucesivamente. Luego se suman todos los resultados obtenidos de cada multiplicación, tomando en cuenta que si el resultado de un producto en particular resulta un número de dos cifras, se suman ambos dígitos (e.g. si un parcial es 12, se agregará 1 + 2 = 3 a la sumatoria total). A continuación, se calcula el valor módulo 10 del resultado. Finalmente se sustrae este valor de 10, y se vuelve a sacar módulo diez de él; aquel será el dígito verificador correspondiente. Este proceso de cálculo – y verificación – del décimo dígito de una cédula vamos a detallarlo con un ejemplo para el número de cédula 1703386431; el algoritmo debería dar como resultado 1 (uno) que es el décimo dígito, los cálculos realizados se muestran en la Figura 4.2.

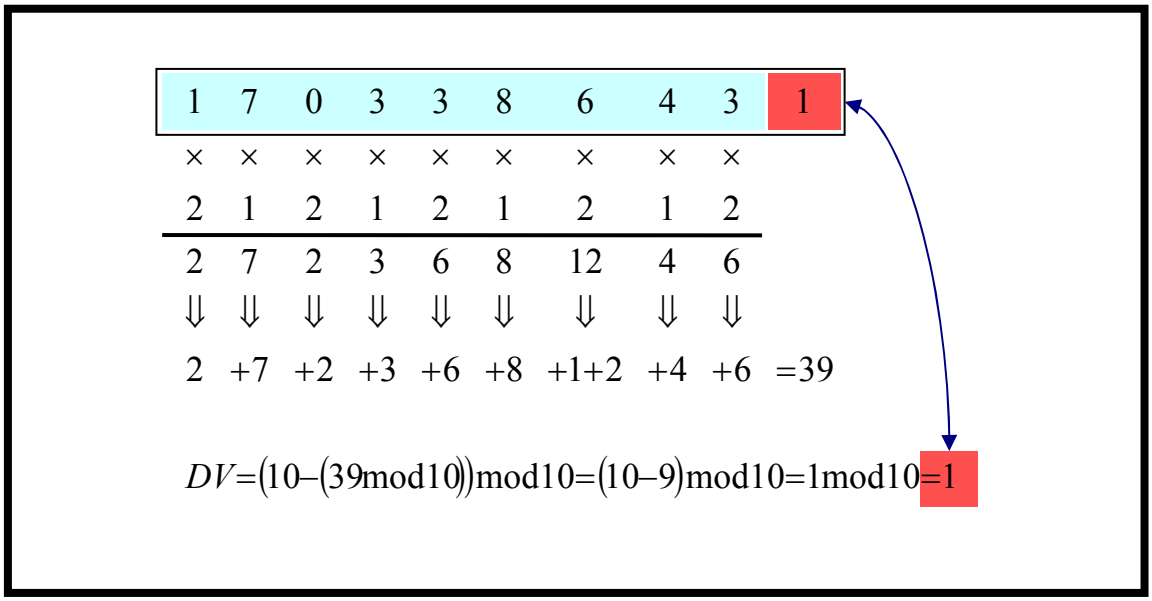


Figura 4.2 Ejemplo de método de verificación módulo 10

Una vez detallado cuál es el procedimiento para comprobar la validez de una cédula de ciudadanía del Ecuador, podemos pasar a explicar las consideraciones generales tanto para el programa que obtiene cédulas reales cuanto para aquel que las genera.

IV.2.3.1.2 *Script* que obtiene cédulas válidas.-

Este pequeño programa genera recurrentemente posibles cédulas válidas (utilizando el método ya expuesto) y lanza una consulta al portal del Consejo Nacional Electoral. Una vez recibido el código fuente de la página obtenida luego de la consulta, una función discrimina una cadena de caracteres para determinar si la cédula consultada existe o no. Si la cédula existe se almacena en un archivo de texto (base de datos), si no existe pasa a la siguiente. El proceso se repite hasta hallar el número de cédulas válidas que el usuario desee. A manera de ejemplo se realizan consultas para cédulas válidas de la Provincia de Pichincha.

No está por demás recalcar que **este *script* no se utilizará para obtener cédulas válidas que puedan ser empleadas posteriormente en la simulación. Únicamente se lo ha desarrollado para demostrar las deficiencias del sistema.**

IV.2.3.1.3 Programa que genera las bases de datos

Por otro lado, el programa que genera las bases de datos crea un vector con posibles números de cédula con el método de verificación módulo 10. De este vector se ha tomado aleatoriamente un subconjunto (una de cada diez cédulas) para crear otro vector que sea la base de datos del banco. Además se ha generado un arreglo de claves aleatorias que correspondan a cada uno de los clientes de la institución bancaria. Cada uno de estos vectores es escrito luego en archivos separados para su lectura desde el programa de simulación.

IV.2.3.2 Simulación del ataque

Para la simulación de se tienen dos tipos de entidades: el servidor de autenticación y las máquinas que realizan el ataque para lograr acceso. Tanto el servidor

como las máquinas clientes serán considerados como procesos[17] de CSIM. El intercambio de información se realizará mediante mensajes que viajan entre *mailboxes*[18].

Entonces, el servidor, tendrá un sistema de autenticación, acceso a una base de datos con los nombres de usuario y claves de acceso de sus clientes y un *mailbox* para envío/recepción de mensajes. Por otro lado, cada máquina tiene un sistema de generación aleatoria de contraseñas, una base de datos con posibles nombres de usuario de los clientes del banco y su propio *mailbox*. A continuación, en la página siguiente, se muestra en la Figura 4.3 el esquema bajo el cual se implementará la simulación:

[17] Un *proceso de CSIM* hace referencia a un procedimiento de C que ejecuta una sentencia '*create*'; un proceso puede ser invocado con argumentos de entrada pero no puede retornar un valor. Pueden haber varias instancias 'activas' del mismo proceso, cada una de estas instancias aparenta estar siendo ejecutada en paralelo aunque de hecho se ejecutan secuencialmente en un procesador simple. (Traducción de CSIM User's Guide (C Version) 2007).

[18] Un *mailbox* permite el intercambio de información entre procesos de CSIM. Cualquier proceso puede enviar un mensaje a un *mailbox* así como intentar recibir un mensaje de cualquier *mailbox*. La cola de un *mailbox* es de tipo FIFO. (Traducción de CSIM User's Guide (C Version) 2007).

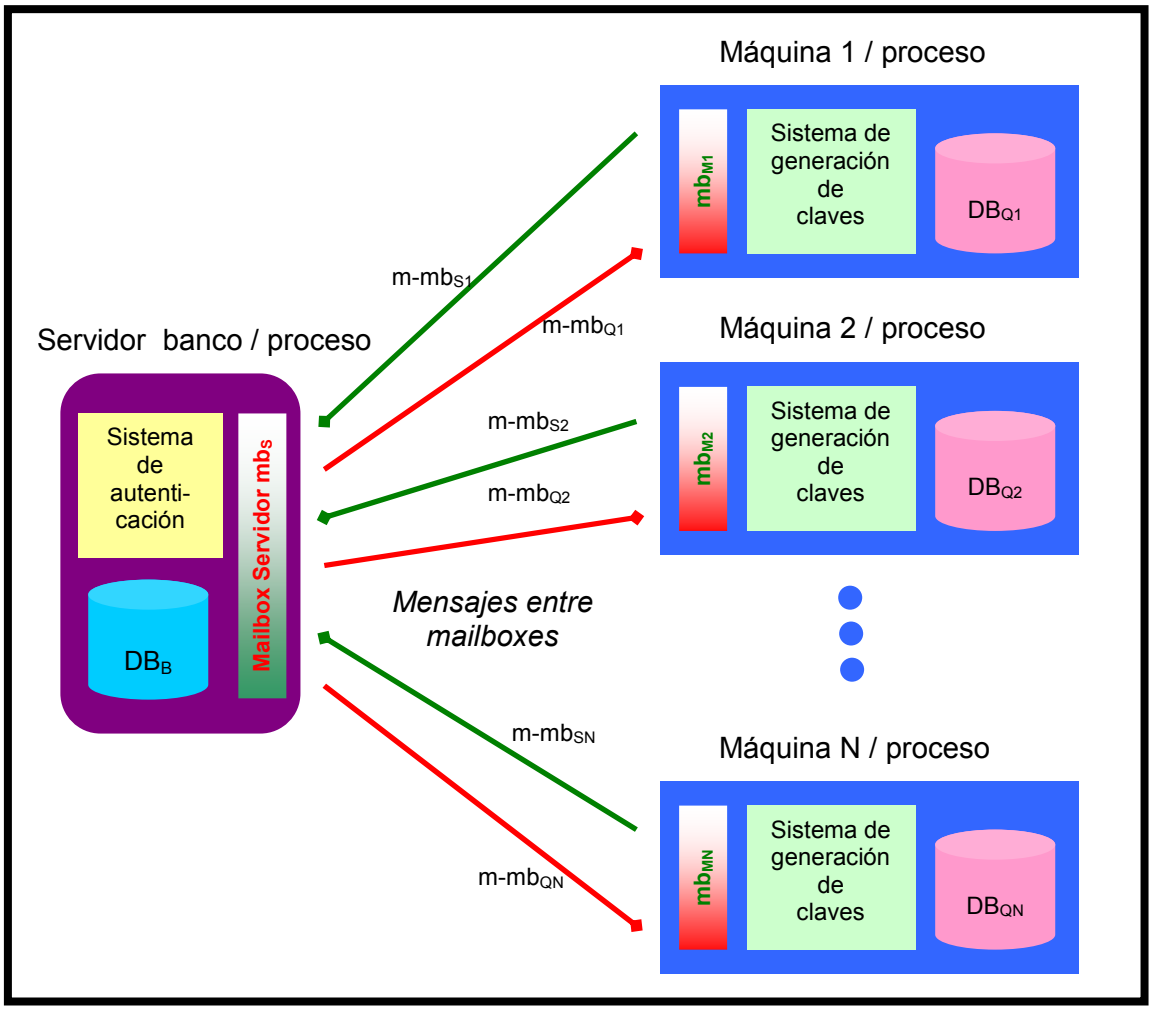


Figura 4.3 Esquema de la simulación

En este esquema se puede explicar con mayor claridad lo expresado anteriormente en la forma de obtención de las bases de datos. Se destaca que la base de datos de nombres de usuario del banco (DB_B) es un subconjunto de una base de datos global (DB_G) con mayor número de usuarios que no necesariamente pertenecen al banco en cuestión. Esto es:

$$DB_B \subset DB_G$$

Además, se debe tomar en cuenta que cada máquina tomará un bloque distinto de la base de datos global DB_G. Así se tendría que:

$$DB_{Q1} \cup DB_{Q2} \cup \ldots \cup DB_{Qn} = DB_G$$

Entonces, el procedimiento que sigue la simulación involucra a las máquinas Q_k, (donde k = 1,2,3,...,n), que toman un nombre de usuario de su base de datos, generan una clave aleatoria para dicho usuario e intentan ganar acceso a su cuenta. Del otro lado, el servidor – que escucha posibles intentos de acceso – verifica si el usuario y la clave corresponden efectivamente a una cuenta de su base de datos y acepta o niega el acceso. Toda esta comunicación se da mediante *mailboxes* que transportan mensajes entre los procesos (máquinas y servidor).

Dado que la comunicación debe ser en un doble sentido – esto es desde las máquinas hacia el servidor y desde el servidor hacia las máquinas – se tiene dos tipos de *mailboxes*. El primero, notado con mb_S es el *mailbox* o buzón de entrada del servidor; el segundo tipo es el mb_{Qk} (donde k = 1,2,3,...,n) que corresponde a los *mailboxes* en los que cada máquina atacante recibe respuesta.

Entonces se deberán enviar también dos tipos de mensajes; uno asociado a cada tipo de buzón. El mensaje que envía toda máquina $m\text{-}mb_{Sk}$ (donde k = 1,2,3,...,n) contiene el identificador de la máquina, el usuario escogido por la máquina Q_k con su respectiva clave generada aleatoriamente; estos datos están integrados en una estructura de C. Todos estos mensajes son idénticos, el único parámetro que cambia es el identificador de la máquina que envió la petición de autenticación; todo esto para que el servidor conozca a dónde debe ir dirigida su respuesta. Por otro lado el mensaje $m\text{-}mb_{Qk}$ (donde k = 1,2,3,...,n) contiene la respuesta del servidor: el nombre de usuario y una variable que indique si se logró o se negó el acceso (organizados en una estructura).

Es importante señalar que este proceso de envió de información hacia el servidor y recepción de respuesta por parte de las máquinas se repite hasta m veces (donde m = 3, 4 ó 5) si en los $m\text{-}1$ intentos anteriores no se ha logrado el acceso. Una vez realizados los m intentos, el servidor envía un mensaje de bloqueo de cuenta, la máquina deja de

intentar acceso con ese usuario y escoge otro de su base de datos. Esta comunicación entre procesos (máquinas hacia el servidor y viceversa) será simultánea, es decir varias máquinas intentan acceder a una cuenta distinta al mismo tiempo. En la práctica se simulará con pseudoparalelismo, como ya se ha indicado.

IV.2.4 Diagramas de flujo de la simulación

Las figuras que se muestran en este apartado clarifican cuál es el proceso que siguen ambas entidades (máquinas y servidor) durante la ejecución de la simulación, su comportamiento se ilustra con flujogramas implementados en programación estructurada.

IV.2.4.1 El servidor de autenticación

La Figura 4.4 muestra el procedimiento que sigue el servidor bancario. En ella se puede notar que una vez iniciado el proceso, el servidor escucha si existen mensajes encolados en su *mailbox*, si no existen permanece esperando hasta que llegue un mensaje. Una vez recibido el mensaje extrae sus datos y trata de hallar dicho usuario en su base de datos, si el usuario existe pasa a verificar su contraseña, de otro modo, envía mensaje con 'usuario incorrecto' (En la realidad enviaría un mensaje del tipo 'El cliente no tiene habilitado el servicio'). Si la clave es exitosa envía el mensaje de respuesta indicando que se logró la autenticación. Caso contrario, cuando la clave es incorrecta se incrementa el número de intentos de acceso para esa cuenta y se comprueba si éste es menor al número máximo de intentos permitido; si se cumple con aquella condición se envía un mensaje indicando que la clave es incorrecta; si se llegó al número máximo de intentos se bloquea el usuario y se envía mensaje al *mailbox* de la máquina indicando 'usuario bloqueado'.

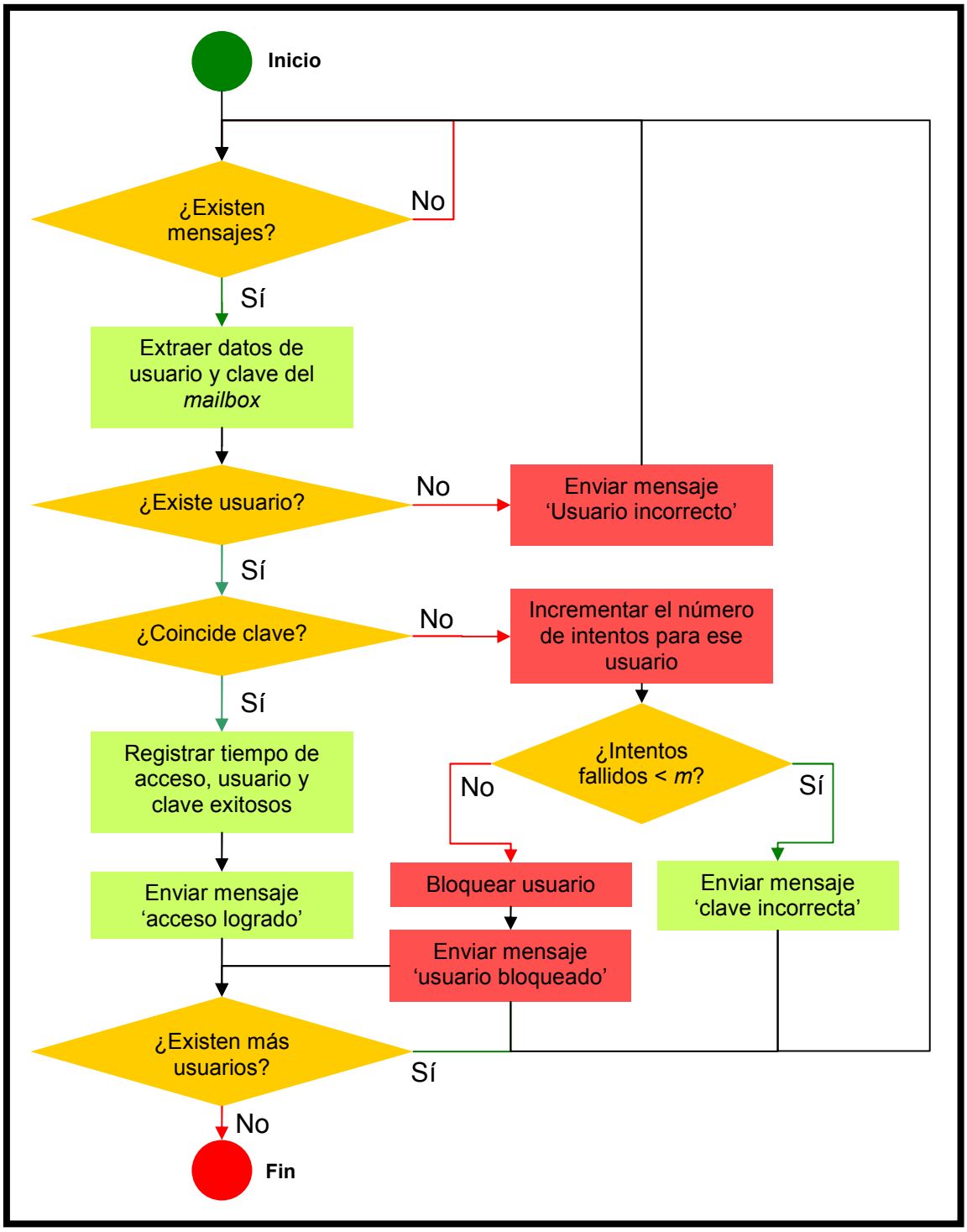


Figura 4.4 Diagrama de flujo del servidor de autenticación

En todos los casos, una vez enviado el mensaje de respuesta vuelve a tomar el próximo mensaje encolado en el *mailbox*, o en su defecto, espera que llegue uno para atenderlo. El proceso servidor terminará su ejecución una vez que todas las máquinas atacantes hayan completado su labor. Esto se logrará observando que todos los

mensajes enviados por las máquinas hayan sido atendidos y se haya enviado una respuesta, la misma que debe haber sido recibida por la máquina.

IV.2.4.2 Las máquinas que intentan lograr acceso

En el escenario planteado, del otro lado se encuentran las máquinas atacantes que intentan autenticarse en el servidor. En este caso, el diagrama que detalla su funcionamiento se muestra en la Figura 4.5.

En ella se puede observar que una vez iniciada, la máquina Q_k toma un usuario de su base de datos, incrementa un contador (*num_us*) para que en la próxima ejecución tome el siguiente usuario e iguala a cero el parámetro *x* que indica el número de veces que se ha intentado el acceso para dicho usuario. Seguidamente genera una clave aleatoria, incrementa el número de intentos, envía el mensaje al *mailbox* del servidor y espera por su respuesta; la máquina se queda en un lazo hasta que en su *mailbox* de entrada tenga un mensaje. Una vez que ha recibido el mensaje, lo extrae del *mailbox* y comprueba el tipo de respuesta obtenida. Si el usuario logró acceder, se registra el tiempo de acceso, se muestran el nombre de usuario y la clave exitosas y se pasa a intentar acceso con el siguiente usuario. Si el usuario es incorrecto o si se obtiene una respuesta en la que el usuario ya fue bloqueado por el servidor, se termina el proceso para éste y se continúa con el siguiente. Y, si el usuario es correcto pero la clave es incorrecta se vuelve a generar una contraseña, se envía un nuevo mensaje al servidor y se espera por nueva respuesta para procesarla.

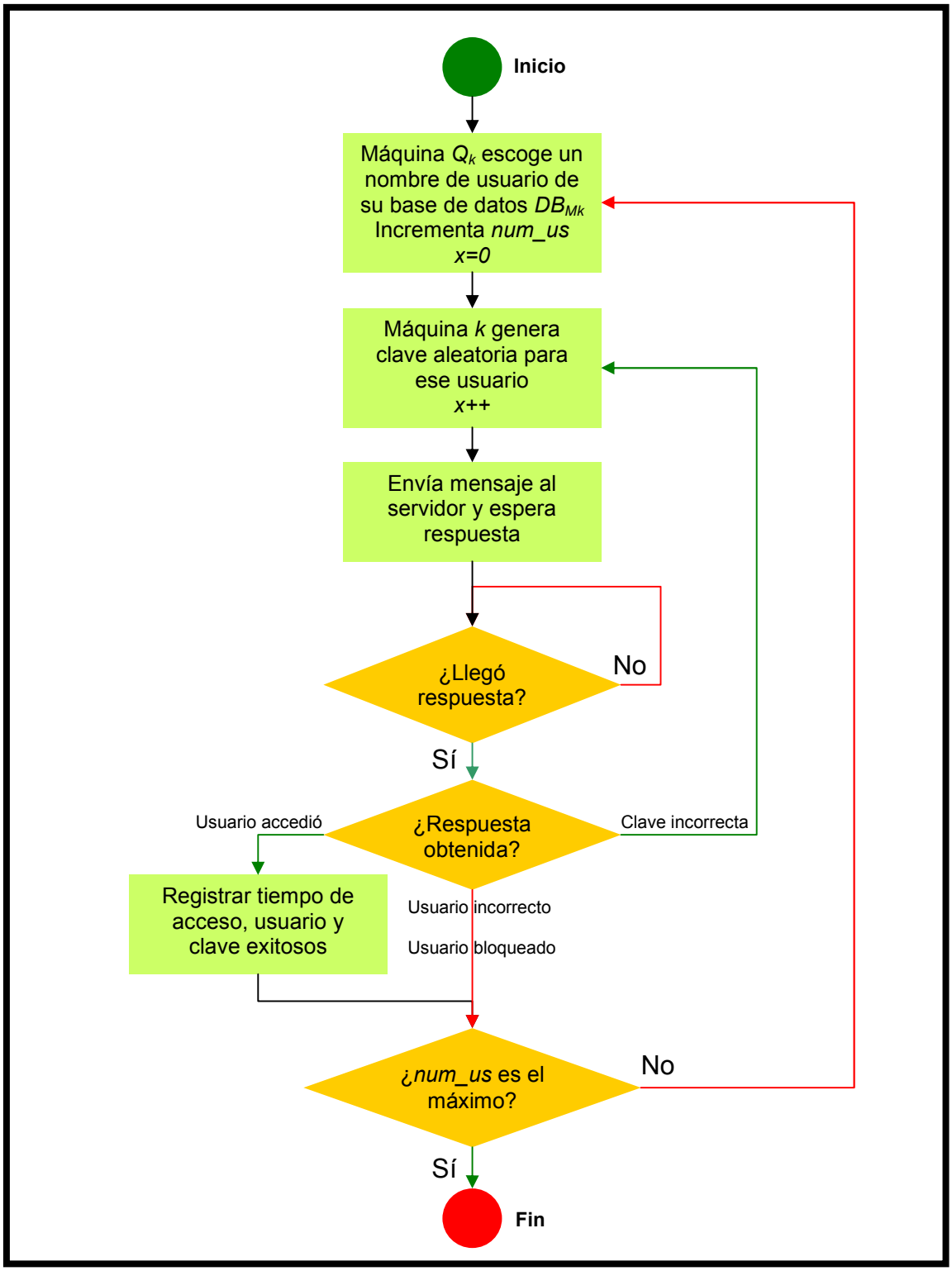


Figura 4.5 Diagrama de flujo de las máquinas atacantes

Cabe destacar que, de acuerdo a lo descrito anteriormente – y también en la realidad –, es el servidor quien tiene el control del número máximo de intentos de acceso; en el caso de las máquinas, una vez que el usuario se bloquea simplemente

continúan con el próximo registrando que ese usuario fue bloqueado. El incremento del número de intentos para cada usuario que se realiza en las máquinas se lo hace simplemente para tener una variable que permita verificar que su valor coincida con los datos en el servidor.

En todos los casos, antes de intentar acceso con el siguiente usuario se verifica que no se haya llegado al final de la base de datos para esa máquina, si tal condición se cumple, la máquina terminará su ejecución.

Capítulo 5

V. ANÁLISIS DE RESULTADOS DE LA SIMULACIÓN

En el presente capítulo se presentan los valores obtenidos en la simulación que se detalló en el capítulo precedente. Además de ello, se realiza un análisis de dichos resultados del simulacro de ataque de fuerza bruta a un sistema de acceso al portal de una entidad bancaria. Cada uno de los análisis que se realice contará con un gráfico que permita describir y observar más claramente los valores y tendencias que se obtuvieron.

Los resultados que se expongan permitirán validar la simulación, frente a los modelos matemáticos planteados en el Capítulo 3.

V.1 NIVELES UTILIZADOS EN LA SIMULACIÓN

Tomando en cuenta la información ya detallada en el apartado IV.1.1.3, los niveles utilizados para los distintos factores que intervienen en la simulación se muestran en la Tabla 5.1.

Para abordar un escenario similar al que se posee en el Ecuador, los niveles se han escogido de acuerdo a los datos presentados en los capítulos anteriores. En particular para el primer factor 'número de usuarios del banco', – que interviene en la variable del 'número de cuentas a las cuales se intentará el acceso' – se ha elegido tres niveles:

100 000, 500 000 ó 1 000 000 de usuarios. En el Ecuador, los bancos más grandes no llegan a los 2 000 000 de clientes[19].

FACTORES	NIVELES
1. Número de usuarios del banco (*N*)	100 000, 500 000 ó 1 000 000 de usuarios
2. Longitud y tipo de nombre de usuario	10 caracteres numéricos
3. Longitud de clave (*n*)	4, 5 ó 6 dígitos
4. Número máximo de intentos de acceso (*m*)	3, 4 ó 5 intentos
5. Cantidad de máquinas atacantes (*Q*)	10, 20, 30 ó 40 máquinas
6. Tiempo de respuesta del servidor del banco	Tiempos aleatorios entre 2.0 y 3.0 segundos con una distribución uniforme
7. Distribución de tiempo entre intentos	4.0 segundos

Tabla 5.1 Factores y niveles utilizados en la simulación

Los factores 2, 3 y 4 influyen en la variable de 'número de cuentas a las que se logra el acceso' porque determinan cuán complicado es 'adivinar' una contraseña de acuerdo, principalmente, a su longitud y número de intentos. De esta manera se supone será más difícil adivinar una contraseña con mayor cantidad de dígitos y en menor número de intentos. Para la 'longitud y tipo de nombre de usuario' se han elegido 10 caracteres numéricos, ya que la mayoría de bancos del país – al menos los más populares – utilizan ese tipo de nombre de usuario. En este caso, como ya se mencionó en el Capítulo 4, ese nombre de usuario corresponde a la cédula de identidad del cliente.

[19] Banco Pichincha recibe medalla en aniversario. [Internet]. [Actualizado 2006, Abril 11]. Diario Hoy. [Citado 2009, Noviembre 4]. Disponible en: http://www.hoy.com.ec/noticias-ecuador/banco-pichincha-recibe-medalla-en-aniversario-231735-231735.html

Los niveles para la ‘longitud de la contraseña de acceso’ son 4, 5 ó 6 dígitos. No está por demás acotar que la mayoría de bancos nacionales incorporan contraseñas de 4 dígitos, pero se ha realizado el análisis también para claves de mayor longitud.

Para los niveles del ‘número máximo de intentos de acceso’ antes de que se bloquee la cuenta, se han escogido 3, 4 ó 5 intentos. Generalmente los bancos nacionales bloquean la cuenta luego del tercer intento fallido.

Finalmente, se detallan los tres factores que intervienen en el ‘tiempo de acceso’ a la primera cuenta. Los niveles para la ‘cantidad de máquinas atacantes’ se han fijado en 10, 20, 30 ó 40 máquinas, tomando en cuenta sobre todo las restricciones de la versión estudiantil de CSIM – que es el simulador que corre sobre lenguaje C utilizado en la implementación – el cuál no permite utilizar más de 40 *mailboxes* de comunicación entre procesos. Además, sería prácticamente irreal que un atacante común pueda contar con un número mayor de equipos para destinarlos a una intrusión de este tipo, ya que requeriría una alta inversión, a menos que pueda contar con máquinas adecuadamente virtualizadas o acceso a máquinas cuya seguridad esté comprometida.

Los ‘tiempos de respuesta del servidor’ son valores aleatorios uniformemente distribuidos entre 2.0 y 3.0 segundos. Estos valores se emplean en todas las variantes de la simulación. Según mediciones realizadas con Wireshark para una conexión a Internet de 1 Mbps, éste es un tiempo que está por encima del que se emplea en obtener acceso u obtener un mensaje de usuario o clave incorrecta.

Para la ‘distribución del tiempo entre intentos’ se ha escogido, para todas las corridas de la simulación, un tiempo fijo de cuatro segundos. De tal manera que, mientras la máquina atacante espera ese tiempo, el resto siga intentando ganar acceso a las cuentas contenidas en su base de datos.

V.2 VARIABLES A ANALIZARSE

Como ya se mencionó en el Capítulo 4, son dos las variables a analizarse: el 'número de cuentas a las que se logró el acceso' y el 'tiempo de acceso a la primera cuenta'.

Estas variables cobran especial importancia debido a que mediante ellas será posible validar la simulación frente a los modelos matemáticos desarrollados. Ya se habló que la simulación se centraría en los bancos de Tipo I, por lo tanto, frente a sus resultados teóricos confrontaremos los resultados obtenidos en el experimento. El número de cuentas a las que se logró el acceso nos permitirá validar la simulación respecto a la teoría propuesta en el modelo. Asimismo, los tiempos de acceso nos permitirán conocer con qué facilidad y prontitud podría un atacante ganar acceso a cuentas bancarias vía los accesos en línea de las instituciones financieras con las características mencionadas.

Para todas las variantes de la simulación, con sus niveles para los distintos factores, se han realizado tres corridas, de tal manera que en los análisis de las variables se muestran las medias calculadas entre las tres observaciones realizadas.

V.3 ANÁLISIS DE NÚMERO DE CUENTAS ACCEDIDAS

V.3.1 Análisis para 100 000 usuarios

En la Figura 5.1 se grafica el número de cuentas accedidas versus el número de máquinas atacantes para 100 000 usuarios del banco y varios *n* (longitud de contraseña) y *m* (número máximo de intentos antes del bloqueo de la cuenta). Este gráfico nos permitirá

observar cómo varía el comportamiento del número de cuentas a las que se gana acceso conforme cambian la longitud de la clave y el número de intentos.

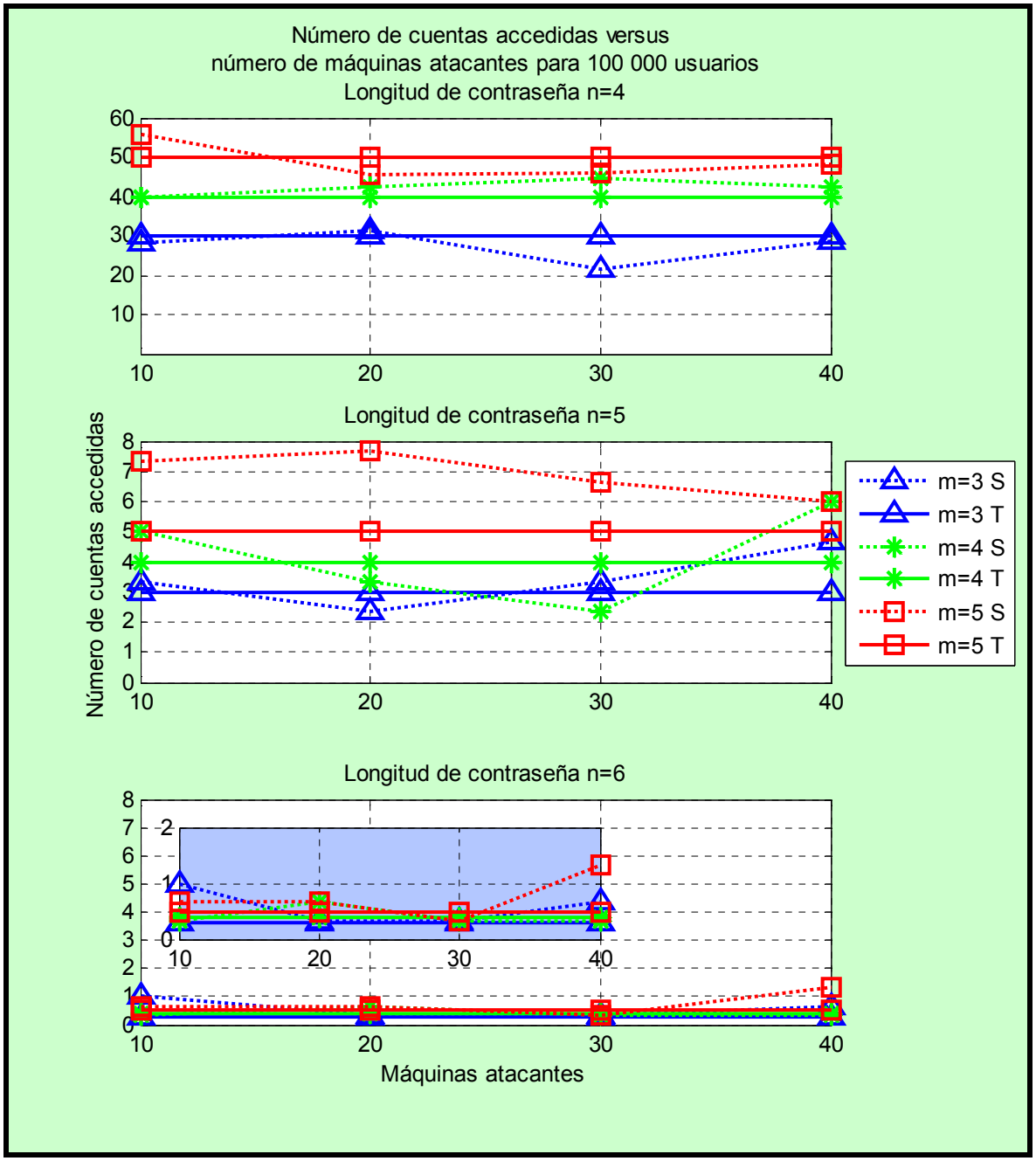


Figura 5.1 Cuentas accedidas versus máquinas atacantes para 100 000 usuarios

Los datos representados en esta figura se muestran tabulados en las páginas siguientes, divididos en cuatro tablas por el número de máquinas atacantes utilizadas en cada simulación.

Número de máquinas atacantes: 10

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	30.00	28.33	3.00	3.33	0.30	1.00
4	40.00	39.67	4.00	5.00	0.40	0.33
5	50.00	55.67	5.00	7.33	0.50	0.67

Tabla 5.2 Número de cuentas accedidas para 10 máquinas atacantes y 100 000 usuarios

Número de máquinas atacantes: 20

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	30.00	31.33	3.00	2.33	0.30	0.33
4	40.00	42.67	4.00	3.33	0.40	0.67
5	50.00	45.67	5.00	7.67	0.50	0.67

Tabla 5.3 Número de cuentas accedidas para 20 máquinas atacantes y 100 000 usuarios

Número de máquinas atacantes: 30

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	30.00	21.67	3.00	3.33	0.30	0.33
4	40.00	44.67	4.00	2.33	0.40	0.33
5	50.00	46.00	5.00	6.67	0.50	0.33

Tabla 5.4 Número de cuentas accedidas para 30 máquinas atacantes y 100 000 usuarios

Número de máquinas atacantes: 40

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	30.00	28.67	3.00	4.67	0.30	0.67
4	40.00	42.33	4.00	6.00	0.40	0.33
5	50.00	48.33	5.00	6.00	0.50	1.33

Tabla 5.5 Número de cuentas accedidas para 40 máquinas atacantes y 100 000 usuarios

De acuerdo a estos datos, se observa que con n=4 los valores obtenidos son muy cercanos a los teóricos (en su mayoría tienen un error relativo del 5%); sin embargo, para n=5 y n=6 existen diferencias mayores entre los datos teóricos y simulados. Por ejemplo para m=5 y Q=20 (número de máquinas) el error absoluto es 2,67 y el relativo asciende a 53,33%; en este caso valor teórico manifiesta que se debe acceder a 5 cuentas del banco, la media de la simulación es 7,67 cuentas (promedio entre 9, 7 y 7 en cada una de las tres corridas). Es decir son dos cuentas más de las que se esperaba acceder. Estos errores podemos atribuirlos a las pocas observaciones que se han realizado – debido al excesivo tiempo que tomaría realizar muchas simulaciones –, sin embargo resulta coherente el acceder a 7 cuentas cuando en realidad se debía haber accedido sólo a 5. Con n=5 el error relativo es del 32,22%, mientras que para n=6 asciende al 63,43%.

Además se observa que para n=5 y n=6, las líneas correspondientes a valores de m distintos se cruzan. Esto se da debido a que los valores teóricos de cuentas accedidas son muy cercanos, apenas difieren en décimas o en unidades, especialmente para n=5 (3, 4 y 5 cuentas accedidas para m=3, 4 y 5 respectivamente) y n=6 (0.3, 0.4 y 0.5 cuentas accedidas para m=3, 4 y 5 respectivamente). Por ello los márgenes entre niveles son muy reducidos y se tienen prácticamente resultados similares en las observaciones.

V.3.2 Análisis para 500 000 usuarios

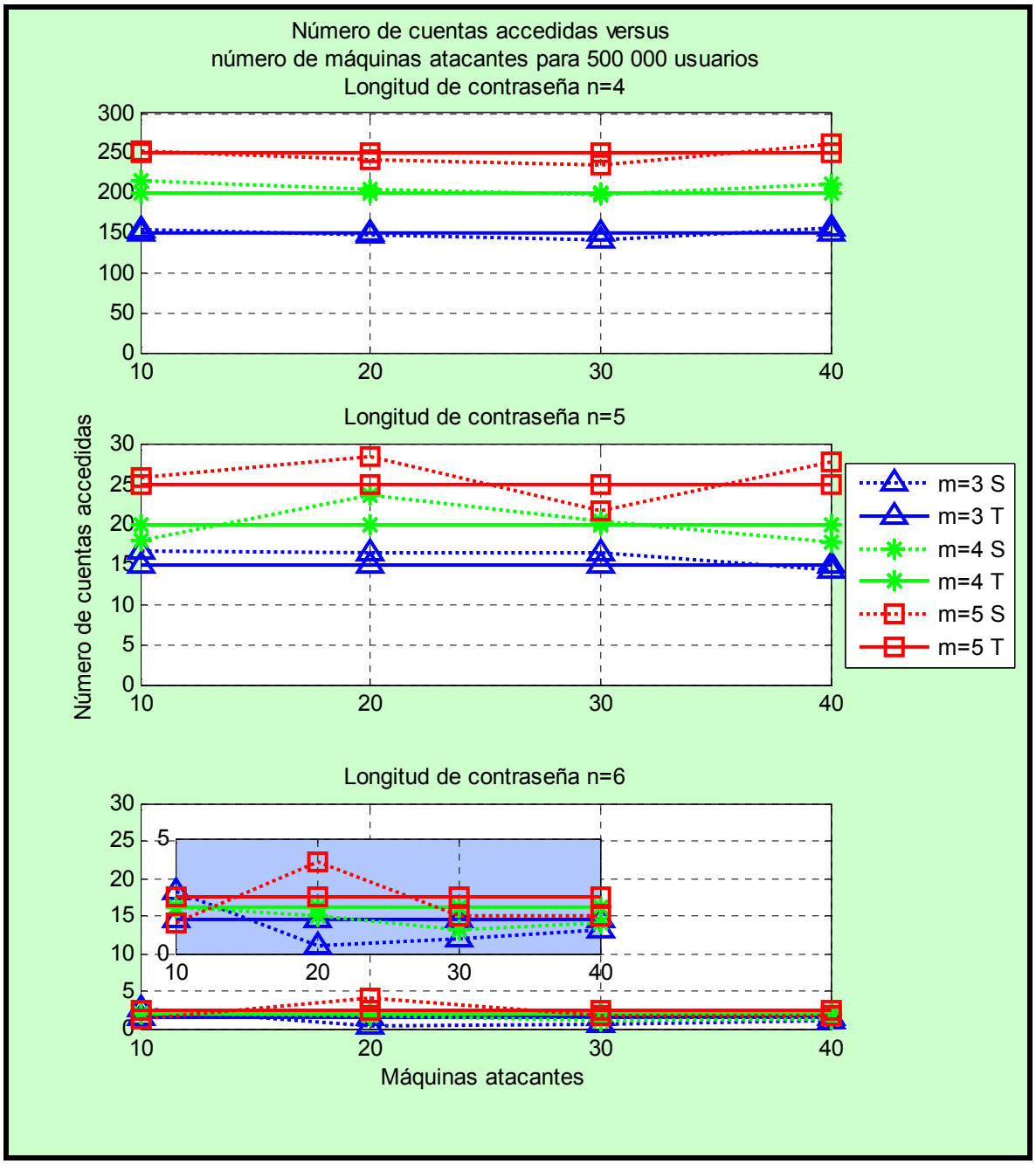


Figura 5.2 Cuentas accedidas versus máquinas atacantes para 500 000 usuarios

Ahora se presentan en la Figura 5.2 los resultados medios obtenidos en las observaciones para 500 000 usuarios – tal como se hizo en la sección anterior –. Se ha

graficado el número de cuentas a las que se logró el acceso versus la cantidad de máquinas atacantes para todos los niveles de los factores *n* y *m*.

A continuación se muestran los datos tabulados, y finalmente un análisis de los valores resultantes de la simulación.

Número de máquinas atacantes: 10

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	150.00	155.33	15.00	16.67	1.50	2.67
4	200.00	215.67	20.00	18.00	2.00	2.00
5	250.00	252.67	25.00	25.67	2.50	1.33

Tabla 5.6 Número de cuentas accedidas para 10 máquinas atacantes y 500 000 usuarios

Número de máquinas atacantes: 20

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	150.00	147.00	15.00	16.33	1.50	0.33
4	200.00	204.33	20.00	23.67	2.00	1.67
5	250.00	243.00	25.00	28.33	2.50	4.00

Tabla 5.7 Número de cuentas accedidas para 20 máquinas atacantes y 500 000 usuarios

Número de máquinas atacantes: 30

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	150.00	141.33	15.00	16.33	1.50	0.67
4	200.00	197.33	20.00	20.33	2.00	1.00
5	250.00	235.67	25.00	21.67	2.50	1.67

Tabla 5.8 Número de cuentas accedidas para 30 máquinas atacantes y 500 000 usuarios

Número de máquinas atacantes: 40

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	150.00	157.00	15.00	14.33	1.50	1.00
4	200.00	211.33	20.00	17.67	2.00	1.33
5	250.00	261.67	25.00	27.67	2.50	1.67

Tabla 5.9 Número de cuentas accedidas para 40 máquinas atacantes y 500 000 usuarios

Para este número de usuarios del banco se observan mejores comportamientos, pues para n=4 las líneas de los resultados están prácticamente sobre las líneas de los valores teóricos y se tiene un error relativo promedio del 3,94%. Con n=5, el valor más alejado del teórico se da para m=5 y Q=20, con 28,33 cuentas accedidas para 25,00 cuentas teóricas; el error relativo promedio para todas las observaciones es del 9,58%. Se puede destacar que a pesar de tener sólo 3 observaciones se obtienen resultados cercanos a los propuestos por los modelos.

Para n=6 se mantiene un escenario algo parecido al visto para 100 000 usuarios, pues se observa que las líneas para varios m se entrecruzan. Nuevamente esto se debe a que la diferencia entre los valores para cada número máximo de intentos es mínima, apenas de 0,50 cuentas (1,50, 2,00 y 2,50 cuentas para m=3, 4 y 5, respectivamente).

V.3.3 Análisis para 1 000 000 de usuarios

Siguiendo el mismo formato, para 1 000 000 de usuarios en la siguiente página se presenta el gráfico (Figura 5.3) del número medio de cuentas accedidas versus las máquinas atacantes para todas las longitudes de contraseña y número máximo de intentos de acceso antes del bloqueo.

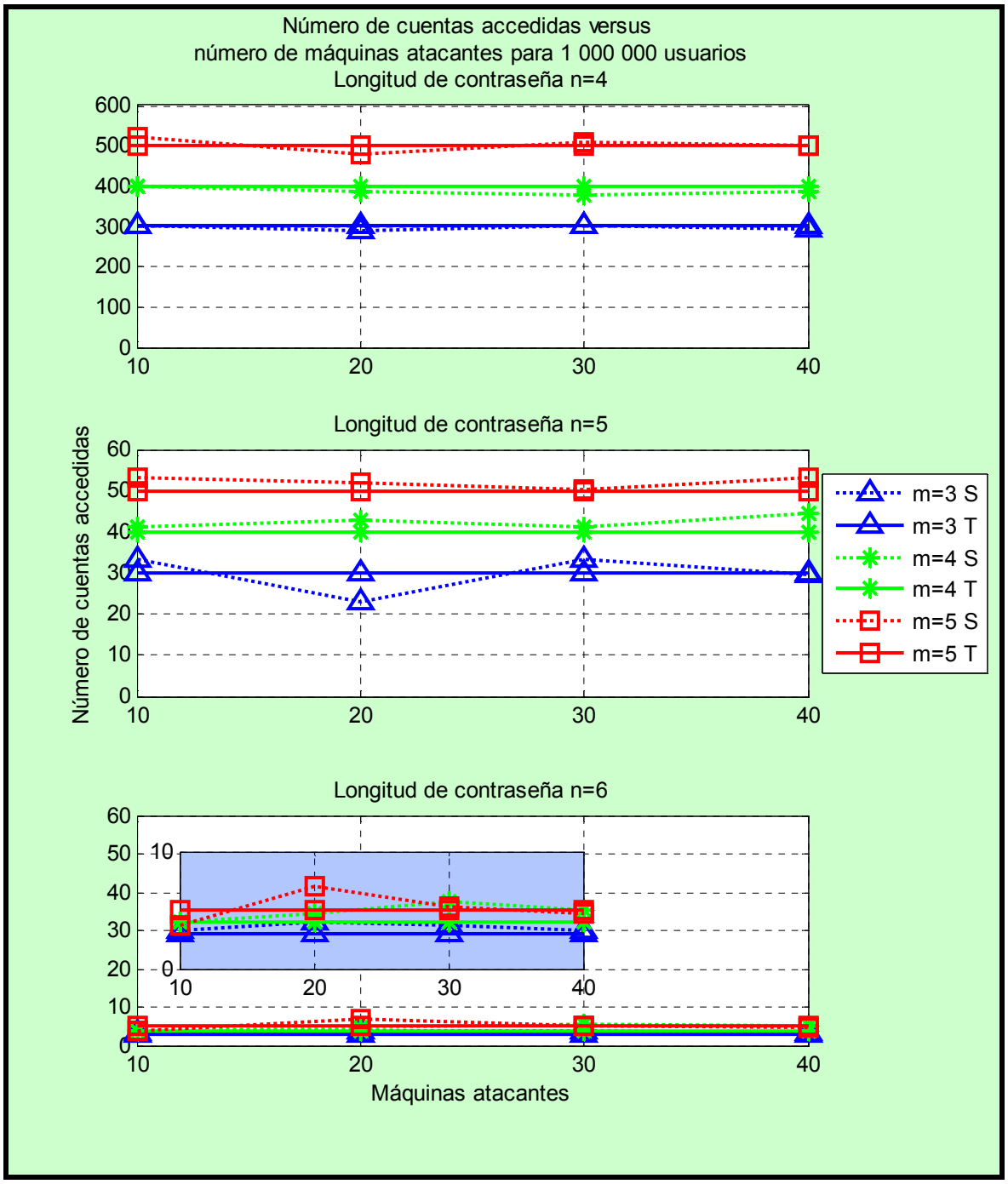


Figura 5.3 Cuentas accedidas versus máquinas atacantes para 1 000 000 de usuarios

Asimismo, a continuación se detallan también los valores tabulados y, finalmente, en unas pocas palabras se resume el análisis de los datos obtenidos.

Número de máquinas atacantes: 10

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	300.00	300.33	30.00	33.33	3.00	3.33
4	400.00	398.33	40.00	41.33	4.00	4.00
5	500.00	523.00	50.00	53.00	5.00	3.66

Tabla 5.10 Número de cuentas accedidas para 10 máquinas atacantes y 1 000 000 usuarios

Número de máquinas atacantes: 20

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	300.00	287.67	30.00	23.00	3.00	4.00
4	400.00	385.67	40.00	42.67	4.00	4.67
5	500.00	480.67	50.00	52.00	5.00	7.00

Tabla 5.11 Número de cuentas accedidas para 20 máquinas atacantes y 1 000 000 usuarios

Número de máquinas atacantes: 30

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	300.00	303.33	30.00	33.33	3.00	3.67
4	400.00	378.33	40.00	41.33	4.00	5.67
5	500.00	508.00	50.00	50.33	5.00	5.33

Tabla 5.12 Número de cuentas accedidas para 30 máquinas atacantes y 1 000 000 usuarios

Número de máquinas atacantes: 40

LONGITUD / INTENTOS	4		5		6	
	Teórico	*Simulado*	*Teórico*	*Simulado*	*Teórico*	*Simulado*
3	300.00	294.00	30.00	29.67	3.00	3.33
4	400.00	387.67	40.00	44.67	4.00	5.00
5	500.00	501.67	50.00	53.00	5.00	4.67

Tabla 5.13 Número de cuentas accedidas para 40 máquinas atacantes y 1 000 000 usuarios

En este caso, con un millón de usuarios del banco, las tendencias son aún mejores, es decir los resultados son más cercanos a los teóricos de lo ya observado en los apartados precedentes. Para n=4 el mayor error relativo se sitúa en 5,42% y el error relativo promedio en apenas 2,52%. Con una longitud de contraseña igual a 5, el error relativo promedio se sitúa en 7,36% y, con n=6 en 20,09%, todo esto tomando en cuenta que únicamente se han hecho tres observaciones por cada nivel de los factores que intervienen en esta variable. Para este el último valor de n, aún se observa que se tiene valores entrecruzados para distintas m ya que la diferencia entre valores para cada simulación es de apenas una unidad (3,00, 4,00 y 5,00 cuentas para todos los valores de m).

V.3.4 Resumen del análisis

En la siguiente tabla se muestra de una manera descriptiva y resumida los resultados del análisis realizado para el número de cuentas accedidas del que hemos tratado en este subcapítulo. En general se presentan los errores relativos de los resultados y ciertas tendencias observadas en los datos, ya que otras explicaciones se han detallado en cada apartado.

También se añaden los valores de cuentas accedidas simulados más y menos cercanos a los planteados teóricamente, para tener una mejor apreciación del tipo de resultados obtenidos en el experimento y ver cuánto se aproximaron a los deseados.

n	100 000 usuarios			
	Error relativo promedio	Tendencia de los datos	Datos más y menos cercanos Simulado vs. Teórico	
4	8,21%	Cercanos a los teóricos	39.67 / 40.00	21.67 / 30.00
5	32,22%	Menos cercanos y entrecruzados	3.33 / 3.00	7.67 / 5.00
6	63,43%	Menos cercanos y entrecruzados	0.67 / 0.50	1.00 / 0.30
n	500 000 usuarios			
	Error relativo promedio	Tendencia de los datos	Datos más y menos cercanos Simulado vs. Teórico	
4	3,94%	Muy cercanos a los teóricos	252.67 / 250.00	215.67 / 200.00
5	9,58%	Cercanos a los teóricos	20.33 / 20.00	21.67 / 25.00
6	43,15%	Menos cercanos y entrecruzados	2.00 / 2.00	0.33 / 1.50
n	1 000 000 usuarios			
	Error relativo promedio	Tendencia de los datos	Datos más y menos cercanos Simulado vs. Teórico	
4	2,52%	Muy cercanos a los teóricos	523.00 / 500.00	300.33 / 300.00
5	7,36%	Cercanos a los teóricos	50.33 / 50.00	23.00 / 30.00
6	20,09%	Menos cercanos y entrecruzados	4.00 / 4.00	7.00 / 5.00

Tabla 5.14 Resumen del análisis del número de cuentas accedidas

A través de la tabla también podemos observar que a medida que la cantidad de usuarios se incrementa, la tendencia se dirige a la obtención de datos más cercanos a los teóricos y a la disminución del error relativo en cada una de las observaciones –obviamente, la cantidad de cuentas accedidas en cada simulación también se incrementará con el aumento del número de usuarios del banco –.

V.4 ANÁLISIS DEL TIEMPO DE ACCESO A LA PRIMERA CUENTA

Al igual que en la sección V.3, en ésta se grafican y tabulan los resultados medios obtenidos para analizarlos de acuerdo a varios criterios. Inicialmente se analizarán los tiempos versus el número de máquinas atacantes para 100 000, 500 000 y 1 000 000 de usuarios con varias longitudes de contraseña. Seguidamente, se estudiarán los datos de los tiempos de acceso versus el número de usuarios del banco para varios valores de máquinas atacantes. Y, finalmente, se analizan los valores resultantes versus la longitud de la clave para los distintos números de usuarios del banco.

V.4.1 Análisis de los tiempos de acceso versus número de máquinas atacantes para varias longitudes de contraseña

V.4.1.1 Análisis para 100 000 usuarios

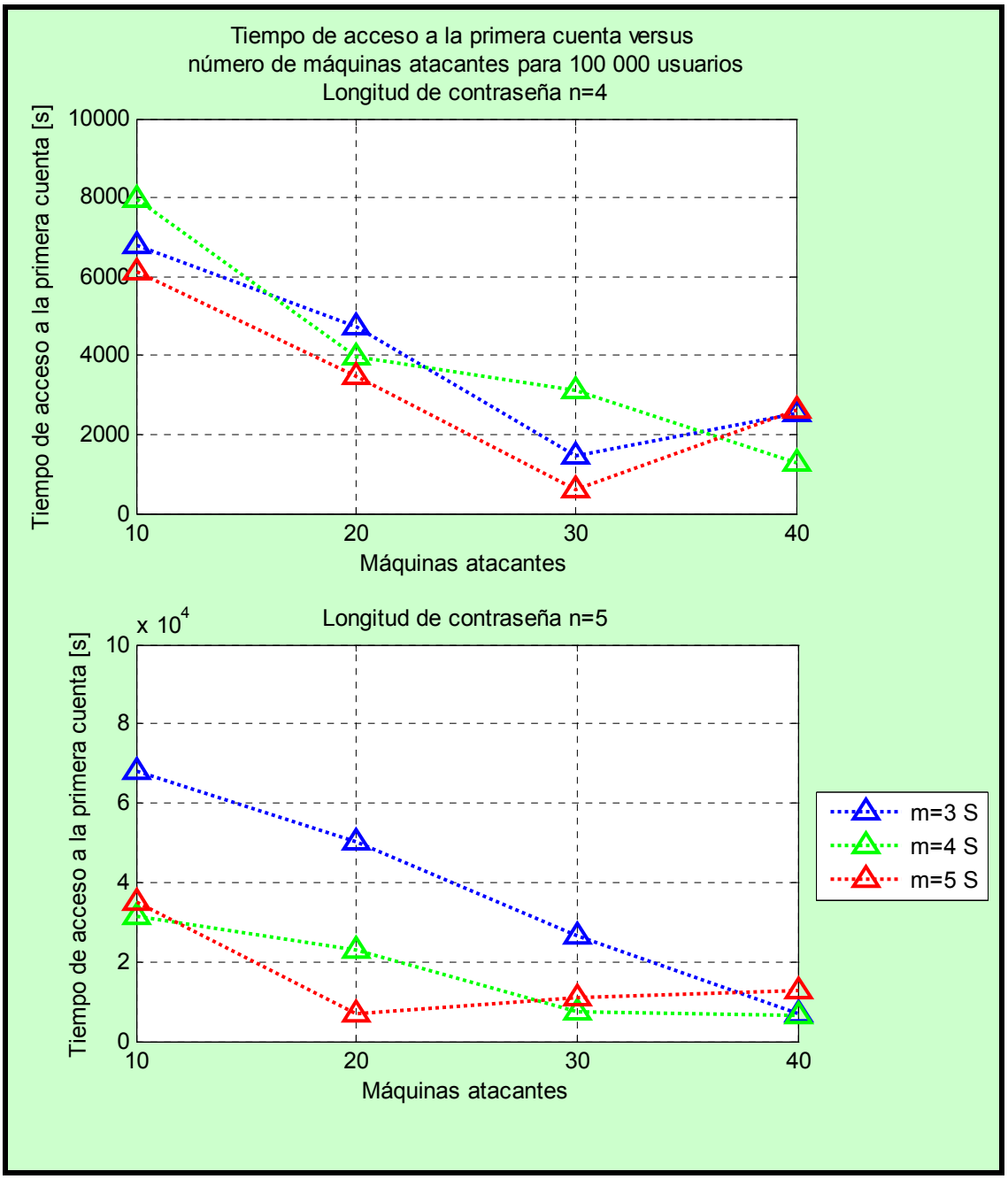


Figura 5.4 Tiempos de acceso versus máquinas atacantes para 100 000 usuarios

La Figura 5.4 y las tablas 5.15, 5.16, 5.17 y 5.18 muestran el comportamiento de los tiempos de acceso simulados con respecto al número de máquinas atacantes para 100 000 usuarios del banco y varias longitudes de contraseña.

Número de máquinas atacantes: 10

LONGITUD / INTENTOS	4	5	6
3	6789.77 s	68286.01 s	n/d
4	7978.94 s	31478.67 s	n/d
5	6136.94 s	34878.47 s	n/d

Tabla 5.15 Tiempos de acceso para 10 máquinas atacantes y 100 000 usuarios

Número de máquinas atacantes: 20

LONGITUD / INTENTOS	4	5	6
3	4753.20 s	50043.24 s	n/d
4	3971.32 s	23201.38 s	n/d
5	3473.62 s	6809.51 s	n/d

Tabla 5.16 Tiempos de acceso para 20 máquinas atacantes y 100 000 usuarios

Número de máquinas atacantes: 30

LONGITUD / INTENTOS	4	5	6
3	1462.22 s	26738.60 s	n/d
4	3137.07 s	7189.27 s	n/d
5	620.90 s	11091.31 s	n/d

Tabla 5.17 Tiempos de acceso para 30 máquinas atacantes y 100 000 usuarios

Número de máquinas atacantes: 40

LONGITUD / INTENTOS	4	5	6
3	2555.05 s	6939.03 s	n/d
4	1272.38 s	6251.40 s	n/d
5	2639.61 s	12587.84 s	n/d

Tabla 5.18 Tiempos de acceso para 40 máquinas atacantes y 100 000 usuarios

De acuerdo a los datos graficados y tabulados se puede observar que, para 100 000 usuarios, los tiempos de acceso disminuyen conforme el número de máquinas atacantes aumenta; esto es totalmente lógico pues para un cierto intervalo de tiempo será mayor la probabilidad de lograr acceso cuanto mayor sea la cantidad de intentos en ese mismo intervalo, eso se logra intentando acceder desde mayor cantidad de atacantes.

Otra observación importante se da al comparar los resultados para distinta longitud de contraseña pues se observa que para n=5 los tiempos de acceso son mucho mayores – la escala se incrementa diez veces – que para n=4. Es así que los tiempos de acceso se incrementan conforme la longitud de la contraseña lo hace.

Se nota también que las líneas para varios valores de m se entrecruzan. Aquello se debe a que se trata de accesos totalmente aleatorios que, si bien es cierto siguen una tendencia, no obedecen a una regla estricta. Es así que para la misma simulación, la primera cuenta accedida podría corresponder al primer usuario intentado o, en el 'peor' de los casos, ésta podría ser del último usuario del banco.

V.4.1.2 Análisis para 500 000 usuarios

A continuación, la Figura 5.5 y las tablas 5.19, 5.20, 5.21 y 5.22 ilustran los datos para 500 000 usuarios del banco versus *Q* para distintas longitudes de contraseña.

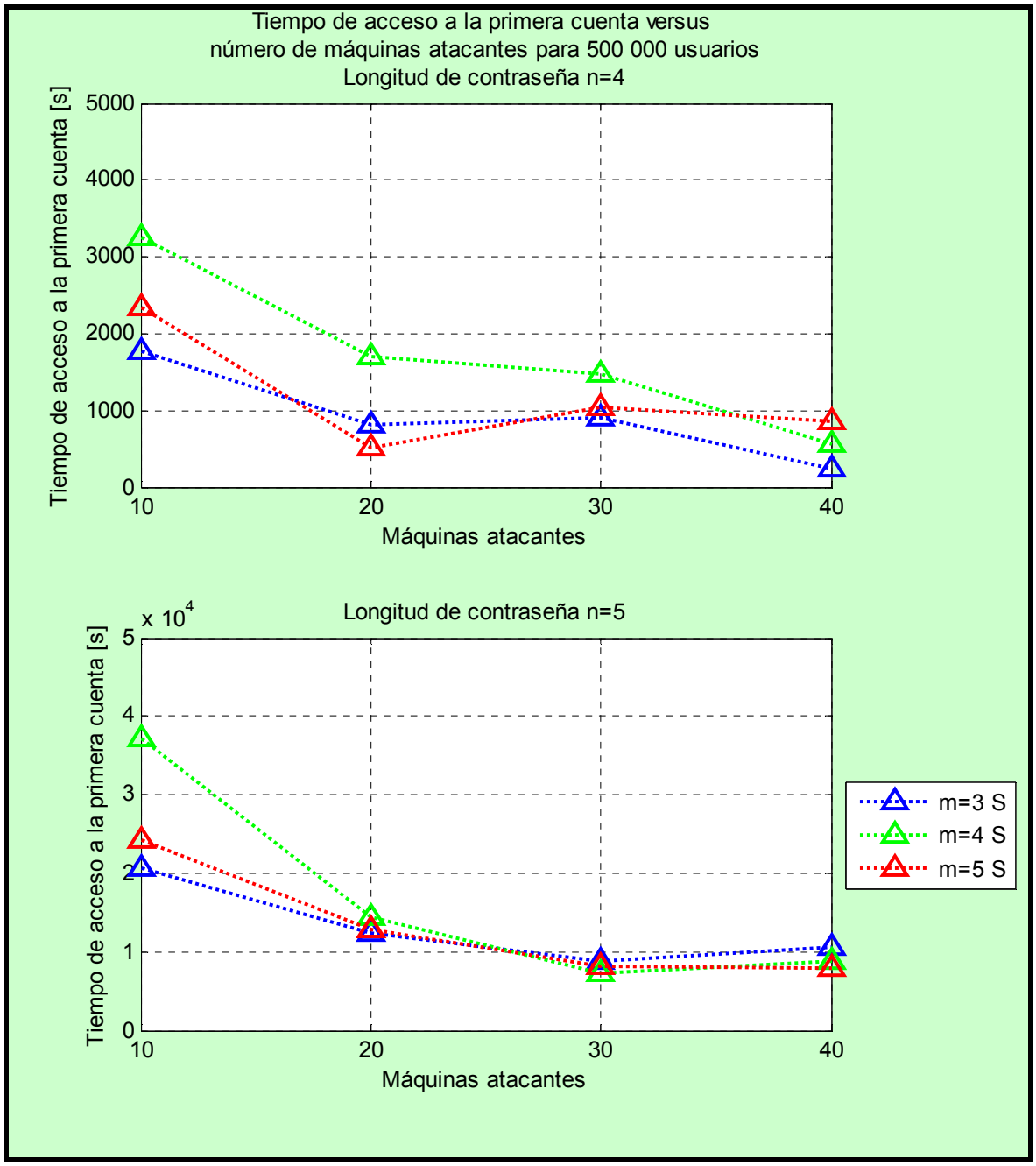


Figura 5.5 Tiempos de acceso versus máquinas atacantes para 500 000 usuarios

Número de máquinas atacantes: 10

LONGITUD / INTENTOS	4	5	6
3	1769.52 s	20672.59 s	277095.34 s
4	3264.67 s	37127.70 s	264432.09 s
5	2343.27 s	24314.25 s	n/d

Tabla 5.19 Tiempos de acceso para 10 máquinas atacantes y 500 000 usuarios

Número de máquinas atacantes: 20

LONGITUD / INTENTOS	4	5	6
3	816.50 s	12331.22 s	n/d
4	1689.63 s	14392.61 s	39824.61 s
5	515.69 s	12831.35 s	87342.00 s

Tabla 5.20 Tiempos de acceso para 20 máquinas atacantes y 500 000 usuarios

Número de máquinas atacantes: 30

LONGITUD / INTENTOS	4	5	6
3	911.92 s	8837.83 s	n/d
4	1471.88 s	7356.30 s	n/d
5	1049.48 s	8056.67 s	n/d

Tabla 5.21 Tiempos de acceso para 30 máquinas atacantes y 500 000 usuarios

Número de máquinas atacantes: 40

LONGITUD / INTENTOS	4	5	6
3	248.73 s	10575.04 s	n/d
4	557.72 s	8854.28 s	62294.64 s
5	858.60 s	8013.36 s	99441.03 s

Tabla 5.22 Tiempos de acceso para 40 máquinas atacantes y 500 000 usuarios

Para 500 000 usuarios se observa un comportamiento similar que para 100 000, pues los tiempos de acceso a la primera cuenta van en decremento a medida que la cantidad de máquinas atacantes se incrementa. Es decir, a mayor número de máquinas atacantes, menor tiempo de acceso; con lo cual se confirma la tendencia observada anteriormente.

También se verifica que para una longitud de contraseña mayor, el tiempo de acceso se verá también incrementado. Para este número de usuarios la escala de tiempos de acceso de n=5 se incrementa diez veces – al igual que para 100 000 usuarios – con respecto a la mostrada para n=4. Igualmente, se vuelve a observar que las líneas para un mismo valor de n y distinto m se entrecruzan dada la aleatoriedad de la simulación.

V.4.1.3 Análisis para 1 000 000 de usuarios

Siguiendo el formato de los apartados precedentes, se muestran a continuación los datos de tiempos de acceso a la primera cuenta obtenidos para 1 000 000 de usuarios del banco.

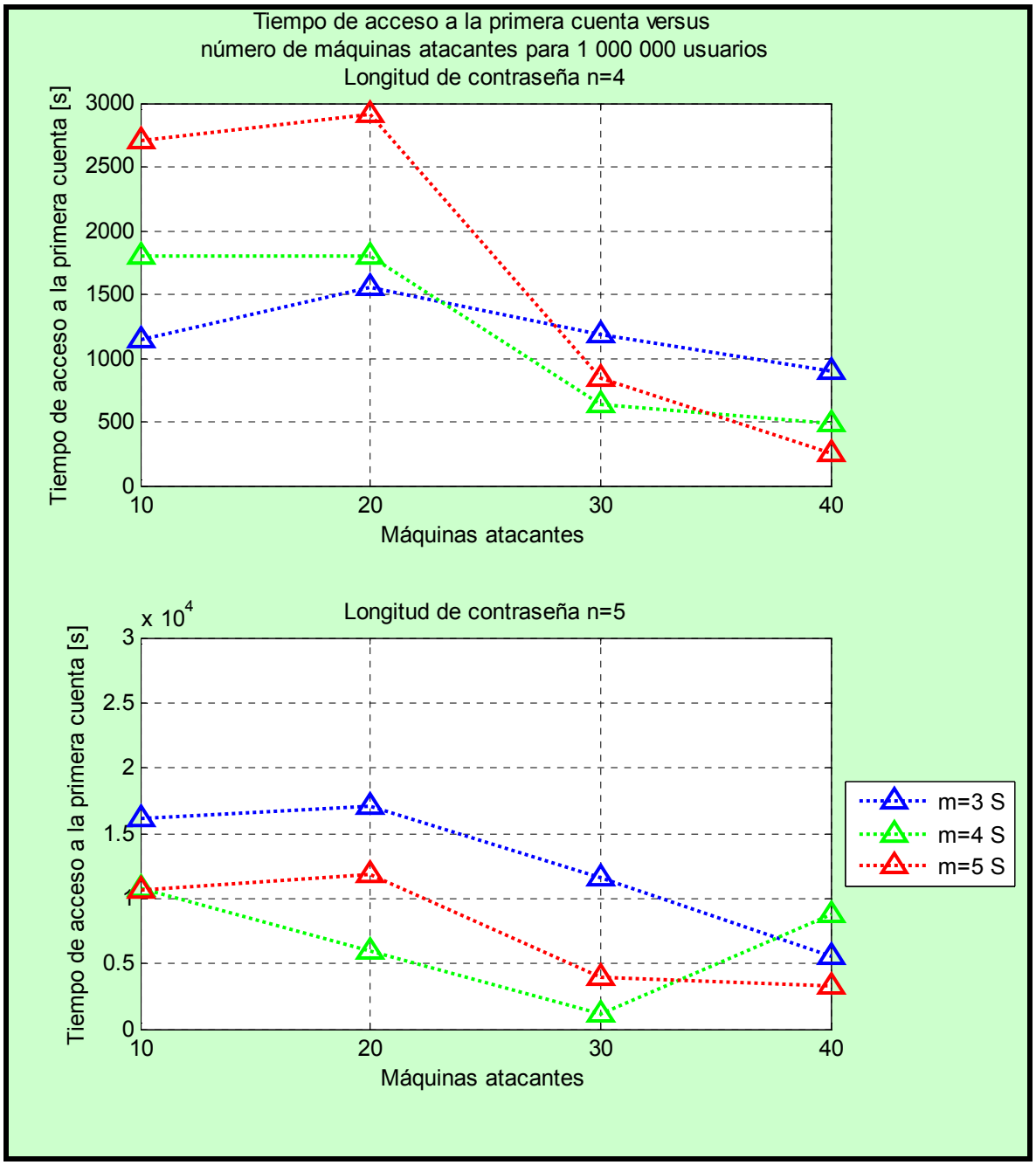


Figura 5.6 Tiempos de acceso versus máquinas atacantes para 1 000 000 usuarios

Número de máquinas atacantes: 10

LONGITUD / INTENTOS	4	5	6
3	1143.21 s	16142.62 s	257170.10 s
4	1801.29 s	10803.32 s	246053.97 s
5	2710.92 s	10645.79 s	202152.83 s

Tabla 5.23 Tiempos de acceso para 10 máquinas atacantes y 1 000 000 usuarios

Número de máquinas atacantes: 20

LONGITUD / INTENTOS	4	5	6
3	1550.60 s	17136.72 s	127109.98 s
4	1804.46 s	5967.88 s	35536.97 s
5	2907.33 s	11801.07 s	79201.45 s

Tabla 5.24 Tiempos de acceso para 20 máquinas atacantes y 1 000 000 usuarios

Número de máquinas atacantes: 30

LONGITUD / INTENTOS	4	5	6
3	1190.22 s	11547.63 s	38612.85 s
4	634.99 s	1110.25 s	66647.74 s
5	836.55 s	3971.96 s	86971.70 s

Tabla 5.25 Tiempos de acceso para 30 máquinas atacantes y 1 000 000 usuarios

Número de máquinas atacantes: 40

LONGITUD / INTENTOS	4	5	6
3	894.91 s	5497.15 s	35702.32 s
4	481.79 s	8720.93 s	52467.29 s
5	259.47 s	3283.84 s	33181.51 s

Tabla 5.26 Tiempos de acceso para 40 máquinas atacantes y 1 000 000 usuarios

Con la Figura 5.6 y las tablas anteriores se confirman las tendencias observadas para los tiempos de acceso con respecto al número de máquinas atacantes: los tiempos disminuyen conforme el número de máquinas atacantes aumenta. Por lo tanto mientras mayor cantidad de atacantes posea, menor será el tiempo que emplee en 'adivinar' la contraseña de acceso. Además, se vuelve a observar que para una longitud de contraseña de cinco dígitos los tiempos de acceso son mayores que para n=4. Entonces, será más difícil – entiéndase más demoroso – 'adivinar' una contraseña de 5 dígitos que una de 4.

Se repite también el entrecruzamiento de líneas para valores distintos m, ya que los accesos son aleatorios. Este comportamiento obedece a que en la simulación, para cada generación de clave, se ha modificado la semilla que la origina.

V.4.1.4 Resumen del análisis

La Tabla 5.27 muestra las tendencias generales observadas para los tres niveles de usuarios del banco: 100 000, 500 000 y 1 000 000 de usuarios y varias longitudes de contraseña. Además se muestran los valores máximos y mínimos de tiempos medios de acceso con el valor de Q para el cual se obtuvieron dichos tiempos para verificar las tendencias.

n	100 000 usuarios		
	Tendencia de los tiempos	Tiempos medios de acceso máximos y mínimos	
4	*t* disminuye a medida que *Q* aumenta. Tiempos entrecruzados para distintos *m* por aleatoriedad de la simulación.	*t* = 7 978.94 s *Q* = 10	*t* = 620.90 s *Q* = 30
5	*t* disminuye a medida que *Q* aumenta. Tiempos entrecruzados para distintos *m* por aleatoriedad de la simulación. Valores mucho mayores que para *n*=4.	*t* = 68 286.01 s *Q* = 10	*t* = 6 251.40 s *Q* = 40
n	500 000 usuarios		
	Tendencia de los tiempos	Tiempos medios de acceso máximos y mínimos	
4	*t* disminuye a medida que *Q* aumenta. Tiempos entrecruzados para distintos *m* por aleatoriedad de la simulación.	*t* = 3 264.67 s *Q* = 10	*t* = 248.73 s *Q* = 40
5	*t* disminuye a medida que *Q* aumenta. Tiempos entrecruzados para distintos *m* por aleatoriedad de la simulación. Valores mucho mayores que para *n*=4.	*t* = 37 127.70 s *Q* = 10	*t* = 7 356.30 s *Q* = 30
n	1 000 000 usuarios		
	Tendencia de los tiempos	Tiempos medios de acceso máximos y mínimos	
4	*t* disminuye a medida que *Q* aumenta. Tiempos entrecruzados para distintos *m* por aleatoriedad de la simulación.	*t* = 2 907.33 s *Q* = 20	*t* = 259.47 s *Q* = 40
5	*t* disminuye a medida que *Q* aumenta. Tiempos entrecruzados para distintos *m* por aleatoriedad de la simulación. Valores mayores que para *n*=4.	*t* = 17 136.72 s *Q* = 20	*t* = 1 110.25 s *Q* = 30
Comparación general entre *n*	*t* aumenta a medida que *n* aumenta para un mismo *N* y *Q*.		

Tabla 5.27 Resumen del análisis de los tiempos de acceso versus máquinas atacantes para varias longitudes de contraseña

En la última fila de la tabla se han evaluado la tendencia de los tiempos para varias longitudes de contraseña, notando que los tiempos son mayores conforme *n* también crece.

Además, es importante mencionar que en todas las figuras de este apartado se han suprimido los gráficos para *n*=6 debido a que, como se puede notar en los datos tabulados, en la mayoría de los casos – especialmente para 100 000 y 500 000 usuarios – se han obtenido tiempos inválidos pues el valor teórico de cuentas a acceder ha sido mínimo (en muchos niveles no llega a una cuenta). Por esto es que, en al menos una de las tres corridas para cada simulación, no se ha obtenido un tiempo de acceso y ha sido imposible promediar un tiempo inválido con otros válidos.

V.4.2 Análisis de los tiempos de acceso versus número de usuarios del banco para varios números de máquinas atacantes

Tomando en cuenta los datos de tiempos de acceso ya tabulados anteriormente, en este apartado se analiza los tiempos de acceso a la primera cuenta versus el número de usuarios del banco para varios números de máquinas atacantes, los cuales se grafican en la Figura 5.6. En general, se observa una tendencia adecuada, los tiempos decrecen conforme el número de usuarios del banco aumenta – esto particularmente para *n*=4 y *n*=5 –.

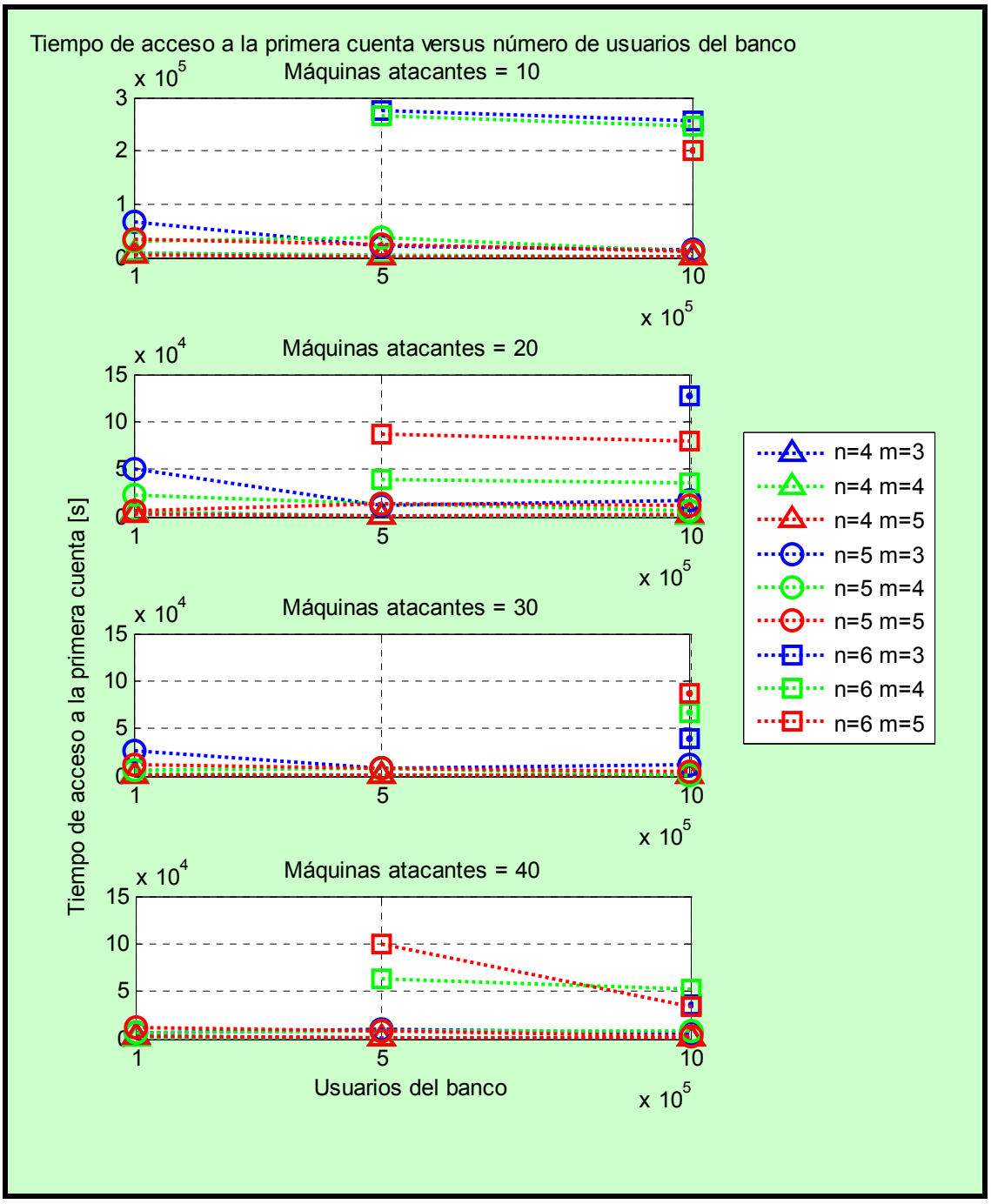


Figura 5.7 Tiempos de acceso versus número de usuarios del banco

También se observa que la tendencia es que los tiempos disminuyan conforme aumente la cantidad de máquinas atacantes – comparando las subfiguras –, como ya se había mencionado en el análisis anterior.

Pero, para una longitud de contraseña de 6 dígitos, aparentemente existe un inconveniente porque los tiempos para 40 máquinas son mayores que los de 30 máquinas. Esta 'inconsistencia' surge por los tiempos inválidos, en este caso para 30 máquinas atacantes con 500 000 usuarios y n=6 no se tienen tiempos válidos, pero por la aleatoriedad de la simulación para 40 máquinas si se tienen dos observaciones válidas. Una buena apreciación del descenso de los tiempos se observa en 1 000 000 de usuarios, donde ya se tienen todos los datos válidos y para n=6 se nota entre las subfiguras que los tiempos decrecen conforme se incrementa el número de máquinas.

V.4.2.1 Resumen del análisis

Q	Tendencia de los tiempos	Valores máximos y mínimos para n=5	
10	*t* disminuye a medida que *N* aumenta. Ciertos valores inválidos para *n*=6.	*t* = 68 286.01 s *N* = 100 000	*t* = 10 645.79 s *N* = 1 000 000
20	*t* disminuye a medida que *N* aumenta. Ciertos valores inválidos para *n*=6.	*t* = 50 043.24 s *N* = 100 000	*t* = 5 967.88 s *N* = 1 000 000
30	*t* disminuye a medida que *N* aumenta. Mayor cantidad valores inválidos para *n*=6.	*t* = 26 738.59 s *N* = 100 000	*t* = 1 110.25 s *N* = 1 000 000
40	*t* disminuye a medida que *N* aumenta. Ciertos valores inválidos para *n*=6. Valores de *t* mayores que para *Q*=30, originado por la aleatoriedad de la simulación.	*t* = 12 587.84 s *N* = 100 000	*t* = 3 283.84 s *N* = 1 000 000
Comparación entre *Q*	*t* disminuye a medida que *Q* aumenta para un mismo *n* y *N*.		

Tabla 5.28 Resumen del análisis de los tiempos de acceso versus número de usuarios del banco para varios números de máquinas atacantes

La tabla anterior resume descriptivamente el comportamiento observado para los tiempos de acceso versus el número de usuarios del banco. Además se detallan, a manera de ejemplo, los tiempos máximos y mínimos para n=5 de modo que se pueda reflejar su comportamiento.

V.4.3 Análisis de los tiempos de acceso versus longitud de contraseña para varios números de usuarios del banco

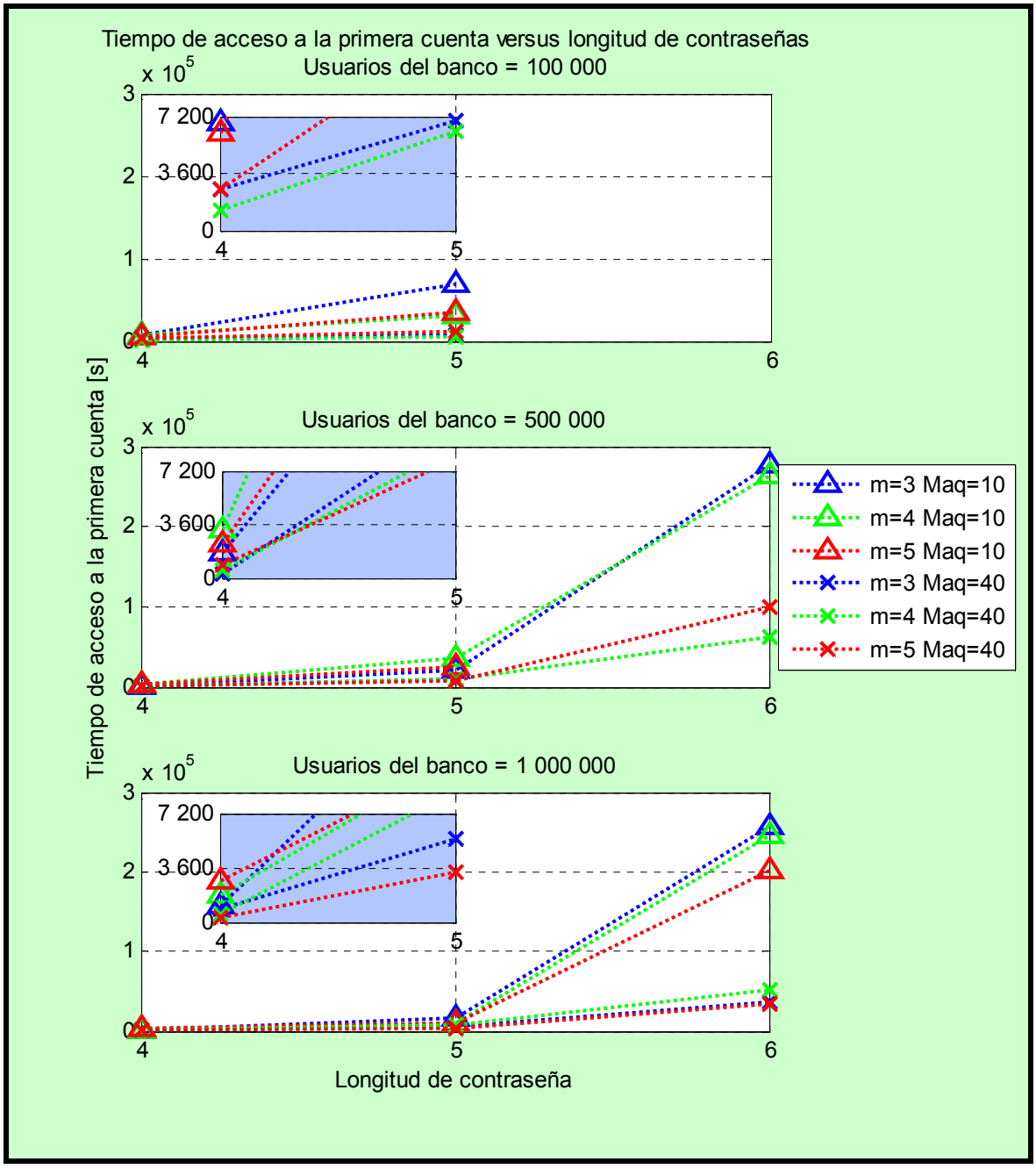


Figura 5.8 Tiempos de acceso versus longitud de contraseña

Para finalizar el análisis, en esta figura se presentan los tiempos de acceso a la primera cuenta versus la longitud de contraseña para todos los niveles de usuarios del

banco; se han tomado en cuenta los valores obtenidos únicamente con 10 y 40 máquinas atacantes para que los gráficos puedan ser más claros.

Es así que se confirma una vez más la tendencia de los tiempos de acceso con respecto a la longitud de la clave; esta es, que el tiempo de acceso se incremente a medida que aumenta la longitud de contraseña. Para 100 000 usuarios no se muestran datos para n=6 debido a que se han obtenido resultados inválidos como se había explicado ya anteriormente.

Al observar cada subfigura parecería que con n=4 el acceso es inmediato, esto realmente no es así – véanse los datos tabulados –. Para darle mayor claridad a los gráficos se han creado en cada subfigura otra figura que amplía la información entre n=4 y n=5. De esta manera se nota que algunos tiempos de acceso están dentro de las dos primeras horas de simulación.

V.4.3.1 Resumen del análisis

N	Tendencia de los tiempos	Valores mínimos y máximos para Q=10	
100 000	*t* aumenta a medida que *n* aumenta. Valores inválidos para *n*=6.	*t* = 6 136.94 s *n* = 4	*t* = 68 286.01 s *n* = 5
500 000	*t* aumenta a medida que *n* aumenta. Ciertos valores inválidos para *n*=6.	*t* = 1 769.52 s *n* = 4	*t* = 37 127.70 s *n* = 5
1 000 000	*t* aumenta a medida que *n* aumenta.	*t* = 1 143.21 s *n* = 4	*t* = 257 170.10 s *n* = 6
Comparación entre *N*	*t* disminuye a medida que *N* aumenta para un mismo *Q* y *n*.		

Tabla 5.29 Resumen del análisis de los tiempos de acceso longitud de contraseña para varios números de usuarios del banco

La tabla anterior resume las tendencias observadas en el análisis realizado. Al igual que en los análisis anteriores, se muestran ciertos valores de tiempos que permiten demostrar las tendencias mencionadas.

V.4.4 Resumen global del análisis de tiempos de acceso a la primera cuenta

A continuación, una vez finalizado el análisis, se presenta una tabla que resume las tendencias generales halladas en los distintos casos estudiados, quedándonos con los comportamientos observados que confirman la validez de la simulación frente a los modelos matemáticos planteados.

Relación	Tendencia de los tiempos	Muestras de la tendencia	
t vs. *n*	*t* aumenta a medida que *n* aumenta para un mismo *N* y *Q*.	*t* = 7 978.94 s *n* = 4 *N* = 100 000 *Q* = 10	*t* = 68 286.01 s *n* = 5 *N* = 100 000 *Q* = 10
t vs. *Q*	*t* disminuye a medida que *Q* aumenta para un mismo *n* y *N*.	*t* = 68 286.01 s *Q* = 10 *n* = 5 *N* = 100 000	*t* = 50 043.24 s *Q* = 20 *n* = 5 *N* = 100 000
t vs. *N*	*t* disminuye a medida que *N* aumenta para un mismo *Q* y *n*.	*t* = 6 136.94 s *N* = 100 000 *Q* = 10 *n* = 4	*t* = 1 769.52 s *N* = 500 000 *Q* = 10 *n* = 4

Tabla 5.30 Resumen global del análisis de los tiempos de acceso

En la tabla también se incluyen muestras de tiempos de acceso a la primera cuenta que permiten verificar las tendencias encontradas a lo largo de todo el análisis realizado.

Capítulo 6

VI. CONCLUSIONES Y RECOMENDACIONES

En este capítulo final se plantean las conclusiones del estudio realizado. Se discutirá los hallazgos en el campo de la seguridad de los mecanismos de acceso estudiados. También se plantearán recomendaciones que pueden adoptarse para mejorar la seguridad de la banca en línea en el país.

VI.1 NOMBRES DE USUARIO

Los bancos del Ecuador que tienen el número de cédula como nombre de usuario para el acceso a su portal electrónico son vulnerables. En nuestro país es relativamente fácil obtener números de cédula válidos. En esta tesis demostramos cómo, realizando consultas recurrentes a portales como el del Consejo Nacional Electoral[20], un individuo con conocimiento básico o medio de programación *web* podría recolectar – en relativamente poco tiempo[21] – suficiente cantidad de números de cédula que le sirvan como base de datos para ejecutar un potencial ataque y acceder a cuentas de usuarios de banca electrónica.

Es más, se podría crear un programa sencillo que además del número de cédula en cada petición discrimine y almacene los nombres y apellidos del ciudadano, información que obviamente es proporcionada al realizar una consulta del lugar de votación. De esta

[20] www.cne.gov.ec
[21] Experimentalmente se han obtenido 1 000 cédulas válidas en menos de 600 segundos, con una conexión de 1 Mbps.

manera habrá recolectado no sólo números de cédula sino también el nombre de sus titulares.

Existen también otros mecanismos para obtener nombres de usuario válidos. Si se quiere obtener el número de cédula basado en los apellidos del individuo, páginas *web* como la del Consejo Nacional de Educación Superior[22] (CONESUP) lo ofrecen sin problema. Mediante una consulta para verificar si un título universitario está registrado en el CONESUP, introduciendo uno o los dos apellidos, se puede obtener el número de cédula del individuo. Y, obviamente, los ciudadanos titulados serían un objetivo mucho más atrayente para un atacante – considerando que éstos supuestamente tendrían mayor capital en los bancos que los ciudadanos no titulados –. Otra fuente para obtener nombres de usuario reales es el Servicio de Rentas Internas[23] (SRI) igualmente a través de su portal, consultando el RUC.

Con todos estos precedentes, es recomendable que los nombres de usuario para acceder a la banca en línea no guarden ninguna relación con documentos de identificación sino que el mismo usuario sea quien personalice su nombre de usuario de tal manera que se vuelva desconocido e inaccesible para un extraño. Igualmente las políticas de seguridad de la institución bancaria no deben permitir a sus clientes seleccionar nombres de usuario basados en números de cédulas o en palabras del diccionario. Seguridad de este tipo la implementa el Bank of America[24] de los Estados Unidos el cual permite al usuario crear una identificación – llamada *Online ID* – de entre 6 y 32 caracteres o letras, haciendo que obtenerla sea prácticamente imposible pues no sólo que se tienen un nombre de usuario de tipo alfanumérico sino que también éste es de una longitud variable, lo que le da un nivel

[22] http://www.conesup.net
[23] http://www.sri.gov.ec
[24] https://www.bankofamerica.com/

más de seguridad. El nombre de usuario es totalmente personalizado y nada tiene que ver con números de cuenta, tarjetas de crédito, débito o credenciales.

Otro ejemplo de un banco que implementa un nombre de usuario con mayores seguridades es Barclays[25] del Reino Unido. El nombre de usuario está compuesto por dos campos, el apellido del cliente y un número de membresía – de 12 dígitos – asignado aleatoriamente por el banco cuando el usuario se registra. Mucho más difícil de adivinar que un número de cédula.

VI.2 CONTRASEÑAS

Los resultados nos han permitido observar las debilidades de los sistemas de acceso que utilizan actualmente los bancos del Ecuador. En un banco con 1 000 000 de usuarios, cuyo mecanismo de acceso requiere un nombre de usuario de 10 dígitos (cédula de identidad), con una contraseña de tipo numérico de 4 dígitos, y que se bloquea luego del tercer intento fallido se pueden ingresar en un ataque de fuerza bruta, teóricamente, a 300 cuentas. La simulación ha demostrado que es posible acceder a la primera cuenta en 1 143.21, 1 550.60, 1 190.22 y 894.91 segundos para 10, 20, 30 y 40 máquinas atacantes, respectivamente (accediendo a 300.33, 287.67, 303.33 y 294.00 cuentas en cada experimento).

Las observaciones son alarmantes pues, en cada nuevo ataque – en el mejor de los casos – se podría acceder a 300 nuevas cuentas sin que se repitan las accedidas en el ataque anterior. Al atacante le bastaría ganar acceso a la cuenta, aún sin realizar una transferencia de fondos, para tener varias maneras de atemorizar al dueño de la misma. Sin duda, el mecanismo de autenticación debe ser modificado para mejorar su seguridad.

[25] http://www.barclays.co.uk/

Una debilidad del sistema radica en que algunos bancos ecuatorianos la contraseña coincide forzosamente con el de la tarjeta de débito o al menos con su tipo, sin que el usuario pueda personalizar su contraseña de banca electrónica. Lo ideal sería que la clave de acceso sea totalmente configurable por el cliente.

Gran parte de la seguridad del mecanismo de acceso recae en el tipo y la longitud de la contraseña, por lo tanto, el sólo mejorar la clave – sin cambiar el tipo y longitud de nombre de usuario – ya incrementaría la seguridad del sistema. En los resultados de la simulación se ha visto que con contraseñas numéricas de 6 dígitos, la cantidad de cuentas a las que se podría acceder en un ataque es prácticamente nula. Si consideramos ahora, contraseñas de 6 caracteres alfanuméricos, es casi imposible ganar acceso: ya se mostró teóricamente este particular en el Caso Alternativo de los modelos matemáticos del Capítulo 3 para n=4. Imaginemos ahora cuán seguro se volvería el mecanismo si además de incrementar la longitud de la contraseña a 6 caracteres e incorporarle letras, subimos su longitud a más de 8 caracteres e incluimos símbolos especiales.

Tomando en cuenta los mecanismos utilizados por bancos extranjeros podremos notar grandes diferencias. Por ejemplo, Bank of America, utiliza contraseñas de entre 8 y 20 caracteres, permitiendo la inclusión de símbolos como @ # * ( ) ={ }/ ? ~ . , ; - entre otros, sin aceptar espacios. Además, la contraseña debe cumplir ciertas características adicionales: debe tener al menos un dígito y una letra y no puede ser igual al nombre de usuario.

Otros bancos, como Citibank[26], utilizan contraseñas de al menos 6 caracteres alfanuméricos permitiendo al cliente su personalización y recomendándole no usar su nombre, el nombre de su esposa, el de su mascota, su fecha de cumpleaños, su comida

[26] http://www.citi.com

favorita, o cualquier otro tipo de información que otros puedan obtener fácilmente. Tampoco se recomienda usar claves que contengan parte del nombre de usuario o de la dirección de correo electrónico del cliente.

A continuación, en la Tabla 6.1 se presentan las características actuales de los mecanismos de acceso a través de Internet a los bancos del Ecuador y las posibles mejoras en las que se podría incursionar para incrementar su seguridad y hacerlos menos vulnerables. Todo esto recogiendo lo ya concluido anteriormente.

CARACTERÍSTICA	ACTUAL	POSIBLE MEJORA
NOMBRE DE USUARIO	Cédula de identidad	Personalizado por el usuario
Longitud	10 caracteres	10 – 32 caracteres
Tipo	Numérico	Alfabético o alfanumérico.
CONTRASEÑA	Clave de tarjeta de débito	Personalizada por el usuario
Longitud	4 dígitos	Mínimo 6 caracteres
Tipo	Numérico	Alfanumérico

Tabla 6.1 Características actuales y posibles mejoras para la banca electrónica en el país

Obviamente, la seguridad de la banca electrónica no sólo se basa en tener nombres de usuario y claves más robustos, también pasa por temas como de verificar la identidad del cliente y evitar intrusiones fraudulentas a sus cuentas: políticas anti-*phising*. Estos mecanismos permitirían al cliente reconocer que está realmente en el portal de su banco en lugar de un sitio falso que podría estar intentando robar sus credenciales.

La seguridad de la banca electrónica y su mejoramiento constituye hoy en día un tema muy extenso que podría ser motivo de otras tantas investigaciones. En este trabajo únicamente nos hemos centrado en verificar cuán vulnerable es un banco a un ataque de fuerza bruta. Futuros estudios podrían enfocarse en el análisis de ataques realizados a los sistemas bancarios de nuestro país, casos de suplantación y *phishing*, y otras fallas que amenacen la seguridad de las instituciones que poseen accesos a través de Internet para proponer mejoras a otros mecanismos empleados.

Sin duda en nuestro país hay mucho por mejorar, más vale hacerlo ahora, antes de que algún *hacker* tome la iniciativa de explotar las vulnerabilidades y esto cause pérdidas irreparables no sólo para los bancos sino sobretodo para sus clientes. No existen sistemas 100% seguros, pero sí existe la manera de mitigar sus fallas y debilidades y acercarnos lo máximo posible a esa seguridad informática deseada.

BIBLIOGRAFÍA

[1] Almagro M. 2000. La seguridad en la banca a distancia. Instituto para la Seguridad en Internet. [Internet]. [Citado 2009, Octubre 19]. Disponible en http://www.instisec.com/publico/verarticulo. asp?id=19.

[2] Banco Pichincha recibe medalla en aniversario. [Internet]. [Actualizado 2006, Abril 11]. Diario Hoy. [Citado 2009, Noviembre 4]. Disponible en: http://www.hoy.com.ec/noticias-ecuador/banco-pichincha-recibe-medalla-en-aniversario-231735-231735.html

[3] Cheswick W, Bellovin S, Rubin A. 2003. Firewalls and Internet Security: Repelling the Wily Hacker. Second Edition. Addison – Wesley. United States of America.

[4] Computer Desktop Encyclopedia. [Internet]. [Actualizado 2007]. [Citado 2009, Noviembre 20]. Disponible en http://www.answers.com/library/Computer%20 Encyclopedia.

[5] CSIM User's Guide (C Version). 2007. Mesquite Software, Inc. Texas. United States of America.

[6] Deitel H, Deitel P. 2003. Cómo programar en C++. Cuarta Edición. Pearson Educación de México, S.A. de C.V. México.

[7] Dierks T, Rescorla E. 2008. RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2. [Internet]. [Citado 2009, Noviembre 20]. Disponible en http://www.ietf.org/rfc/rfc5246.txt.

[8] García C, Romero A. 2004. La expansión de la banca *online* en España. Información Comercial Española. Revista de Economía. [Internet]. [Citado 2009, Octubre 19]. Disponible en http://www.revistasice.com/ cmsrevistasICE/pdfs/ICE_813_89-99__07563D6D319A734F98E4E1 75213263F1.pdf.

[9] Hole K, Moen V, Tjostheim T. 2006. Case Study: Online Banking Security. IEEE Security & Privacy Magazine. Vol 4 No 2. United States of America.

[10] Housley R, Ford W, Polk W, Solo D. 1999. RFC 2459: Internet X.509 Public Key Infrastructure. Certificate and CRL Profile. [Internet]. [Citado 2009, Noviembre 20]. Disponible en http://www.ietf.org/rfc/rfc2459.txt.

[11] Informe sobre fraude online 2005. [Internet]. S21sec. [Citado en 2009, Octubre 19]. Disponible en https://cert.s21sec.com/index.php/es/documentos/func-finishdown/2/.

[12] ITU-T X.680 Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation. [Internet]. [Actualizado 2002]. Disponible en http://www.itu.int/ITU-T/studygroups/com17/languages/X.680-0207.pdf.

[13] Kurose J, Ross K. 2005. Computer Networking: A Top-Down Approach Featuring the Internet. Third Edition. Pearson Education, Inc. United States of America.

[14] Langer N. 2001. Revolución Tecnológica. Alfa-Redi: Revista de Derecho Informático. [Internet]. [Citado 2009, Octubre 19]. Disponible en http://www.alfa-redi.org/rdi-articulo.shtml? x=661.

[15] Ley de comercio electrónico, firmas electrónicas y mensajes de datos del Ecuador. [Internet]. [Actualizado 2002, Abril]. [Citado 2009, Noviembre 20]. Disponible en http://www.conatel.gov.ec/site_conatel/index.php?option=com_docman&task=doc_download&gid=1775&Itemid.

[16] McGann R. 2005. Online Banking Increased 47 Percent Since 2002. ClicZ. [Internet]. [Citado 2009, Octubre 19]. Disponible en http://www.clickz.com/3481976.

[17] McGraw-Hill Dictionary of Scientific and Technical Terms. [Internet]. [Actualizado 2003]. [Citado 2009, Noviembre 20]. Disponible en http://www.answers.com/library/Sci%252DTech+Dictionary-cid-2390044.

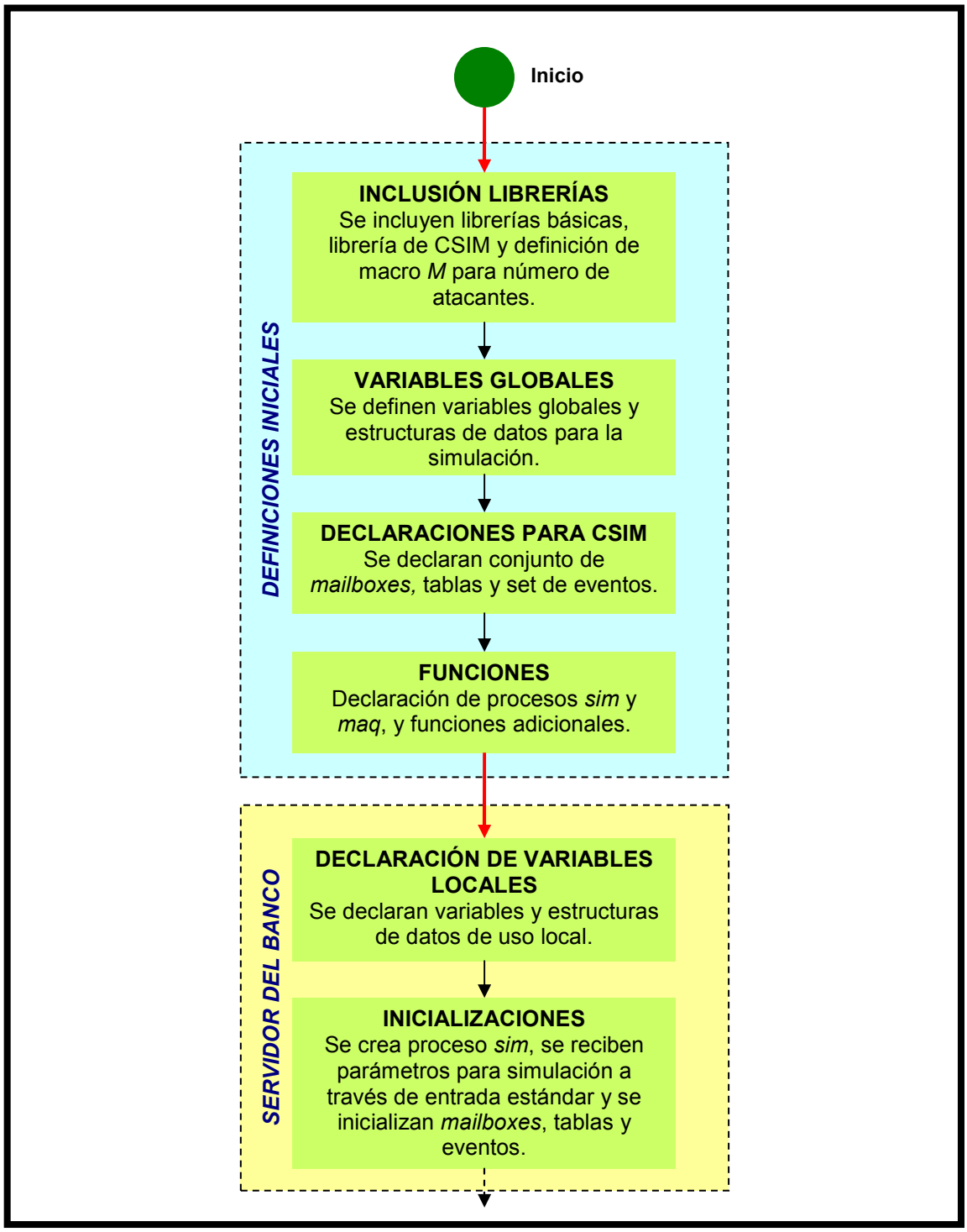
[18] McMillan R. 2005. After theft, bank tightens online security. Network World. [Internet]. [Citado 2009, Octubre 19]. Disponible en http://www.networkworld.com/news/2005/053005bankof america.html.

[19] Myers M, Ankney R, Malpani A, Galperin S, Adams C. 1999. RFC 2560: X.509 Internet Public Key Infrastructure. Online Certificate Status Protocol – OCSP. [Internet]. [Citado 2009, Noviembre 20]. Disponible en http://www.ietf.org/rfc/rfc2560.txt.

[20] Stallings W. 2003. Network Security Essentials: Application and Standards. Second Edition. Pearson Education, Inc. United States of America.

ANEXOS

DIAGRAMA DE BLOQUES DEL PROGRAMA DE SIMULACIÓN



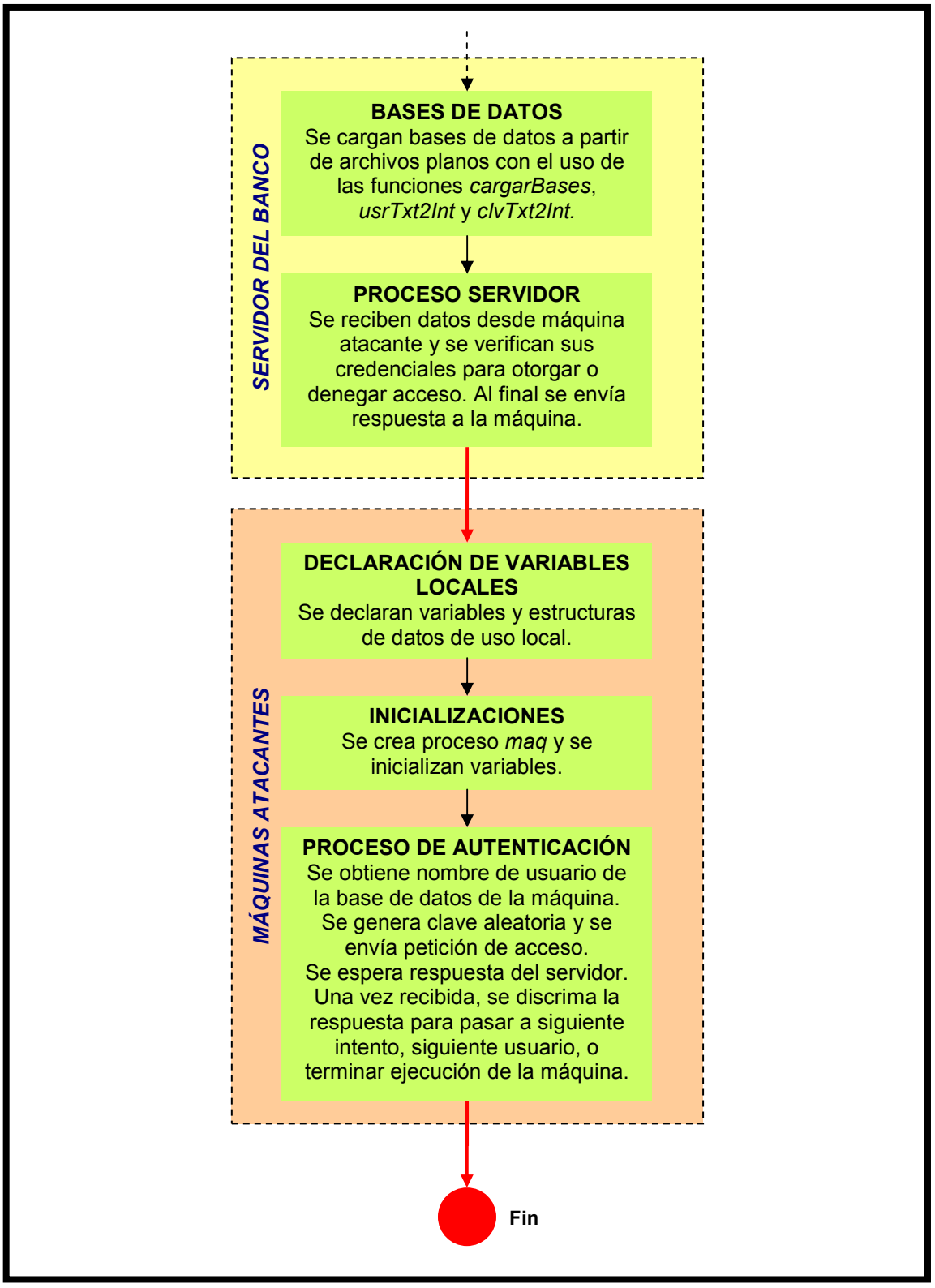


Figura A.1 Diagrama de bloques del programa de simulación.

PROGRAMA DE SIMULACIÓN EN LENGUAJE C Y CSIM

```
// INCLUSIÓN DE LIBRERÍAS
// ----------------------

#include <time.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include <csim.h>
#include <stdio.h>
#define M 10


// VARIABLES GLOBALES
// ------------------

// Variable que controla el nombre de usuario a cuya cuenta se intentará lograr
// el acceso desdecx la máquina atacante
int usuarioActual[M]={0};
// Semilla para generación de claves aleatorias
int semilla=0;
// Numero de veces que la máquina envía información al servidor y éste responde
int numEnv=0,numResp=0;
// Variable que controla el número de máquina atacante
int numMaq=0;
int usuarioInicio=0;
// Contadores para cuentas
int cntTerm=0, cntAcc=0, cntBloq=0, cntNoexiste=0;
// Longitudes de los arreglos que contienen todos los usuarios y los propios del
banco
int longUsrBan=0;
int longUsrTot=1000000;
// Longitud de contraseña
int longClave=0;
// Parámetro usado en lectura de caracteres para
// conversión de clave desde archivo de texto a tipo entero
int paramLongClave=0;
// Número máximo de intentos antes de bloqueo
int intMax=0;
// Arreglo que almacena en el servidor el numero de intentos de acceso a cada
cuenta
int arrIntServ[1000000]={0};
// Cada máquina se inicia con un número de intentos igual a cero
int arrIntMaq[1000000]={0};
// Arreglo que almacena las respuestas obtenidas del servidor para cada máquina
int respInt[M]={0};
// Control de mailboxes
int cntmb=0;
int pocosmb=0;
// Tiempos de inicio de cada máquina
double tMaq[M]={0}, timectrl[M]={0};
// Parámetros tiempo de respuesta de servidor
double tsmin=0;
double tsmax=0;
// Parámetros tiempo entre intentos de acceso
double tmi=0;

// Arreglos que contendrán las base de datos
int bddUsrBan[1000000]={0};
int bddClvBan[1000000]={0};
int bddUsrGlo[1000000]={0};
```

```
// Chequeo de tiempo. Bandera para pausar máquina por tiempo de espera entre
intentos.
int twait[M]={0};

// Estructura que contiene el nombre de usuario y contraseña para autenticación
typedef struct datosCliente
{
	int IDMaqC;
	int nombre;
	int clave;
};

// Estructura para el mensaje de respuesta del servidor
typedef struct autent
{
	//int IDMaqS;
	int nombre;
	int acceso;
};

// Punteros para abrir los archivos
FILE *ptroUsrBan, *ptroClvBan, *ptroUsrGlo;

// DECLARACIÓNES PARA CSIM
// -----------------------

// Declaración de Mailboxes
MBOX maq2Serv_mb; // Mailbox de envío de mensajes de la máquina atacante al
servidor
MBOX serv2Maq_mb[M]; // Mailbox para envío de mensajes del servidor al atacante


// Tabla de tiempos de intento de acceso
TABLE cuentasAccedidas;

// Conjunto de eventos que reactiva la máquina luego de un tiempo de espera entre
intentos de acceso
EVENT etime_set[M];

// FUNCIONES
// ---------

// Procesos de CSIM
void sim(int argc, char *argv[]); // Proceso principal. Hará de servidor del
banco
void maq(int,int); // Proceso para la ejecución de las máquinas atacantes

// Otras funciones
void cargarBases(); // Función para cargar las bases de datos a partir de un
archivo de texto plano
int usrTxt2Int(char[]); // Función para convertir la cadena de caracteres
correspondiente al nombre de usuario a un valor entero
int clvTxt2Int(char[]); // Función para convertir la cadena de caracteres
correspondiente a la clave a un valor entero

// ------------------
// SERVIDOR DEL BANCO
// ------------------
// El proceso 'sim' será el proceso principal que hará de servidor del banco.
void sim(int argc, char *argv[])
{
	// DECLARACION DE VARIABLES LOCALES
	// Estructuras para los datos del cliente y los datos de autentificación
	struct datosCliente *entServ;
	struct autent *respServ;
	// Otras variables
```

```
	int IDMaq; // Identificador de máquina atacante que intenta acceso al servidor
	int u; // Variables de control para lazos y bucles
	int clv2Verif; // Clave proporcionada por el atacante que deberá ser verificada en el servidor
	int respMaq=0; // Índice de máquina a la que se responde el mailbox
	int nuevoAcc; // Bandera que indica si se ha logrado un nuevo acceso o no
	int clienteExiste; // Bandera que indica si el usuario que intenta ganar acceso es o no usuario del banco

	// INICIALIZACIONES
	// Creación del proceso 'sim'
	create("sim");
	// Mensajes de bienvenida en pantalla
	printf("*****************************************************\n");
	printf("**  SIMULACION DE ACCESO A LAS CUENTAS DE UN BANCO  **\n");
	printf("**       MEDIANTE UN ATAQUE DE FUERZA BRUTA         **\n");
	printf("**                                                  **\n");
	printf("**  Realizado por: Juan Almendariz      Junio/2009  **\n");
	printf("*****************************************************\n\n");
	// Niveles definidos por el usuario
	printf("DEFINICION DE NIVELES DE LOS FACTORES PARA SIMULACION: \n");
	printf("-----------------------------------------------------\n");
	printf("Maquinas atacantes: %d\n",M);
	// Recepción de argumentos introducidos por línea de comando
	longUsrBan=atoi(argv[1]); printf("Usuarios del banco: %d\n",longUsrBan);
	longClave=atoi(argv[2]);   printf("Longitud   de   contrasena:   [4,5,6] %d\n",longClave);
	intMax=atoi(argv[3]);printf("Numero maximo de intentos antes de bloqueo: [3,4,5] %d\n",intMax);
	printf("Valores para tiempo de respuesta del servidor: [en s]\n");
	tsmin=atoi(argv[4]);printf("tsmin= %f\n",tsmin);
	tsmax=atof(argv[5]);printf("tsmax= %f\n",tsmax);
	tmi=atof(argv[6]); printf("Intervalo de tiempo entre intentos de acceso: [en s] %f\n",tmi);

	printf("\nPREPARACION DE DATOS:\n");
	printf("-----------------------------------------------------\n");
	// Inicialización mailboxes
	maq2Serv_mb = mailbox("maq2Serv");
	mailbox_set(serv2Maq_mb, "serv2Maq", M);
	printf("Mailboxes creados...\n");

	// Inicialización tablas
	cuentasAccedidas = table ("Tiempos de Acceso a Cuentas");
	printf("Tablas creadas...\n");

	// Inicialización de conjunto de eventos, uno para cada máquina
	event_set(etime_set, "events", M);
	// Se setea a cada evento en estado de 'no ocurrencia'
	for (u=0;u<M;u++)
	{
		clear(etime_set[u]);
	};

	// Estructura de respuesta al intento de acceso
	respServ = (struct autent*) malloc(sizeof(*respServ));
	// Inicialización número de máquinas
	IDMaq=0;

	// BASES DE DATOS
	paramLongClave=longClave+2;
	// Se cargan las bases para la simulación mediante llamada a función
	printf("Cargando bases de datos...\n");
	cargarBases();

	printf("\nEJECUCION DE SIMULACION:\n");
```

```
	printf("-----------------------------------------------------\n");
	printf("Simulando ataque...\n");


	// CREACIÓN DE MÁQUINAS ATACANTES
	for (IDMaq=0;IDMaq<M;IDMaq++)
	{
		// Creación de máquina de acuerdo a identificador y tomando usuario
inicial de base de datos
		hold (0.0);
		maq(IDMaq, IDMaq);
	}

	// PROCESO SERVIDOR
	// Mientras se tenga peticiones de acceso pendientes el servidor las
atiende
	while (msg_cnt (maq2Serv_mb)>0)
	{
		// Se obtiene número de mensajes en mailbox de entrada del servidor
		cntmb=msg_cnt (maq2Serv_mb);
		//printf("Mailboxes %d\n",cntmb);
		// Inicializaciones de variables de control
		nuevoAcc=0;
		clienteExiste=0;
		// Se recibe mailbox desde la máquina)
		receive (maq2Serv_mb, (long*) &entServ);
		// En la respuesta a la máquina se incluirá el nombre de usuario y
el resultado de autenticación
		respServ->nombre=entServ->nombre;
		clv2Verif=entServ->clave;
		respMaq=entServ->IDMaqC;
		// Se realiza la verificacion respectiva haciendo un barrido por
todos los usuarios del banco
		for (u=0; u<longUsrBan; u++)
		{
			if (entServ->nombre == bddUsrBan [u])
			{
				// Si el usuario para el cual se intenta el acceso
efectivamente existe en la bdd del banco
				// se activa bandera indicando que el cliente es real
				clienteExiste=1;
				// Se incrementa el número de intentos registrados en
el servidor
				arrIntServ[u]++;
				// Se pasa a verificar si la contraseña corresponde al
cliente
				if (clv2Verif == bddClvBan [u])
				{
					// Si la contraseña también coindice, se enviará
a la máquina respuesta con acceso logrado
					respServ->acceso = 11;
					// Se sale de la verificación
					break;
				}
				else
				{
					//  Se  incrementa  el  número  de  intentos
registrados en el servidor
					if (arrIntServ[u]==intMax)
					{
						// Si se llegó a numero de intentos máximo
se bloquea la cuenta. Se envia respuesta con bloqueo
						respServ->acceso = 99;
					}
					else
					{
```

```
                                        // Si la contraseña no coincide se envía
respuesta con cliente válido pero contraseña inválida
                                        respServ->acceso = 10;
                                }
                        }
                }
        }
        //  Si  el  cliente  no  existe  se  envía  respuesta  con  cliente  no
existente
        if (clienteExiste==0)
        {
                respServ->acceso = 77;
        }
        // Se simula tiempo de respuesta
        hold(uniform(tsmin/M,tsmax/M));
        //hold(uniform(tsmin,tsmax));
        // Tiempos de espera paralela
        for (u=0; u<M; u++)
        {
                // Procesamiento de tiempo de espera
                if (twait[u]==1 && simtime()>=(tMaq[u]+tmi-0.001))
                {
                        tMaq[u]=simtime();
                        twait[u]=0;
                        set(etime_set[u]);
                }
        }
        // Se envía mailbox a la máquina
        send (serv2Maq_mb[respMaq], (long) respServ);
        //  Se  incrementa  variable  de  control  de  número  de  respuestas
enviadas
        numResp++;
        wait(event_list_empty);
    }
    // Se imprimen variables de control
    printf("\n\nResumen de actividad: \n");
    printf("------------------------------------------------\n");
    printf("Veces que mailboxes superaron limite inferior %d\n",pocosmb);

    printf("Numero envios = %d  Numero respuestas = %d\n", numEnv, numResp);
    printf("Numero cuentas terminadas = %d  \n", cntTerm);
    printf("Numero cuentas accedidas = %d  \n", cntAcc);
    printf("Numero cuentas bloqueadas = %d  \n", cntBloq);
    printf("Numero cuentas con usuario no existente= %d  \n", cntNoexiste);
    // Se muestra reporte de simulación
    report();
    // Termina el proceso
    terminate();
}



// ------------------
// MÁQUINAS ATACANTES
// ------------------

// Máquina que intenta acceso
// Cada máquina intentará el acceso para
void maq (int numMaq, int usuarioInicio)
{
    // DECLARACIÓN DE VARIABLES LOCALES
    // Clave generada para realizar acceso
    int claveGen;
    // Parámetro para generar clave aleatoria
    int limitClv;
    // Usuario actual de cada maquina
    int usuarioMaq=0;
```

```
	// Contador de usuarios de la base de datos de la máquina
	int contUsuario=0;
	// Estructura de datos del cliente para enviar al servidor
	struct datosCliente *salMaq;
	// Estructura de datos de respuesta de la autenticación en el servidor
	struct autent *respDsdServ;

	// INICIALIZACIONES
	// Creación del proceso
	create("maq");
	// Reserva de espacio en memoria para la estructura de datos del cliente
	salMaq = (struct datosCliente*) malloc(sizeof(*salMaq));
	// Se asigna usuario inicial para el cuál la máquina intenta acceso
	usuarioActual[numMaq]=usuarioInicio;
	// Se asigna el identificador para la máquina
	salMaq->IDMaqC = numMaq;

	// PROCESO DE AUTENTICACIÓN
	// Se realiza mientras las máquinas atacantes requieran el acceso
	while (usuarioActual[numMaq]<longUsrTot)
	{
		// Se actualiza usuario para el cual se intenta acceso
		usuarioMaq=usuarioActual[numMaq];
		//  Se  toman  datos  de  usuario  de  la  base  de  datos  global
'usersTot.txt'
		salMaq->nombre = bddUsrGlo[usuarioMaq];
		// Se cambia semilla para que la generación de claves aleatorias se
diferente cada
		// vez que se inicie una nueva máquina
		semilla+= (int) time(NULL);
		//printf("semilla:%d\n", semilla);
		//sleep(3);
		reseed(NIL, semilla);
		// Inicializacion de número de intentos
		arrIntMaq[usuarioMaq]=0;
		// Mientras la cuenta no se bloquee se sigue intentando acceso
		while (arrIntMaq[usuarioMaq]<intMax)
		{
			// Generación de la clave de acceso
			// Se asigna el rango de la clave de acuerdo a su longitud
			limitClv=((int) pow(10,longClave))-1;
			// Se genera clave para el usuario
			claveGen=(int) floor(uniform(0,limitClv));
			// Se asigna clave a estructura de datos para acceso
			salMaq->clave = (int) claveGen;

			// Envío de la petición de acceso
			// Se envia el mensaje al servidor a través de un mailbox
			send (maq2Serv_mb, (long) salMaq);
			// Se incrementa el número de intentos para el usuario/cuenta
			arrIntMaq[usuarioMaq]++;
			// Se incrementa variable de control de envío de peticiones
			numEnv++;

			// Se deberá esperar a que el mensaje sea respondido por el
servidor y recibir tal respuesta
			receive(serv2Maq_mb[numMaq],(long*) &respDsdServ);
			// Se almacena en arreglo la respuesta recibida
			respInt[numMaq]=respDsdServ->acceso;

			// Discriminación de la respuesta recibida
			// Si la respuesta recibida es un acceso logrado se termina
la ejecución para ese usuario y pasa al siguiente
			if (respInt[numMaq] == 11)
			{
				// Se graba tiempo actual de simulación en tabla de
CSIM
```

```
                        record(simtime(),cuentasAccedidas);
                        // Se incrementa número de cuentas accedida
                        cntAcc++;
                        // Se muestra mensaje indicando que se logró acceso
                        printf("Maquina %02d Usuario %07d %d Clave %04d Intento
%d        Tiempo        %8.2f        s\n",numMaq,        usuarioActual[numMaq],
bddUsrGlo[usuarioMaq],claveGen,arrIntMaq[usuarioMaq],tMaq[numMaq]);
                        // Se sale del lazo para pasar al siguiente usuario
                        break;
                }
                // Si la respuesta incluye un usuario válido pero clave
incorrecta se trata nuevamente de lograr el acceso
                // con nueva clave
                if (respInt[numMaq] == 10)
                {
                        // Manejo de tiempo de simulación de la máquina
                        tMaq[numMaq]=simtime();
                        //printf("Maquina  %d  Usuario  %d  Clave  incorrecta
Intento                    %d                    Tiempo                    %f
s\n",numMaq,usuarioActual[numMaq],arrIntMaq[usuarioMaq],simtime());
                        // Control de mailboxes
                        if(cntmb>=3)
                        {
                                // Se setea bandera para indicar que la máquina
está en tiempo de espera entre intentos
                                twait[numMaq]=1;
                                // Se espera ocurrencia de evento que reinicie
la ejecución de la máquina
                                wait(etime_set[numMaq]);
                        }
                        else
                        {

                                pocosmb++;
                                //printf("Mailboxes en decremento %d\n",cntmb);
                        }

                        // printf("Maquina %d reactivada Usuario %d Tiempo %f
%fs\n",numMaq,bddUsrGlo[usuarioMaq],simtime(),tMaq[numMaq]);
                }
                // Si el usuario para el cual se intentaba el acceso es
inválido se pasa al siguiente
                if (respInt[numMaq] == 77)
                {
                        //    printf("Maquina    %d    con    usuario    %d    no
existente\n",numMaq,usuarioActual[numMaq]);
                        // Se incrementa contador de cuentas con usuario no
existente
                        cntNoexiste++;
                        break;
                }
                // Si la máquina fue bloqueada se pasa al siguiente usuario
                if (respInt[numMaq] == 99)
                {
                        // printf("Maquina %d con usuario %d fue bloqueada al
%d           intento           con           clave           %d           Tiempo
%f\n",numMaq,usuarioMaq,usuarioActual[numMaq],claveGen, simtime());
                        // Se incrementa contador de cuentas bloqueadas
                        cntBloq++;
                }
            }
            // Se reinicia el valor almacenado como respuesta recibida del
servidor
            respInt[numMaq] = 0;
            // printf("Maquina %d Usuario %d Tiempo final %10f\n", numMaq,
usuarioActual[numMaq], simtime());
            // Se incrementa número de cuentas terminadas
```

```
            cntTerm++;
            // Se incrementa contador de número de usuarios intentados por la
máquina
            contUsuario++;
            // Se actualiza siguiente usuario para intentar acceso
            usuarioActual[numMaq]+=M;
            // Avisos para visualización de usuarios intentados
            if (cntTerm==100000)
            {printf("                --- Usuarios intentados 100000 ---\n");}
            if (cntTerm==200000)
            {printf("                --- Usuarios intentados 200000 ---\n");}
            if (cntTerm==300000)
            {printf("                --- Usuarios intentados 300000 ---\n");}
            if (cntTerm==400000)
            {printf("                --- Usuarios intentados 400000 ---\n");}
            if (cntTerm==500000)
            {printf("                --- Usuarios intentados 500000 ---\n");}
            if (cntTerm==600000)
            {printf("                --- Usuarios intentados 600000 ---\n");}
            if (cntTerm==700000)
            {printf("                --- Usuarios intentados 700000 ---\n");}
            if (cntTerm==800000)
            {printf("                --- Usuarios intentados 800000 ---\n");}
            if (cntTerm==900000)
            {printf("                --- Usuarios intentados 900000 ---\n");}
      }
      // Termina la máquina una vez que ha intentado para todos los usuarios de
su base de datos.
      printf("Maquina %02d termina ejecucion de %d usuarios a %8.2f s\n",
numMaq, contUsuario, simtime());
      csim_terminate();
}


// Función que convierte el texto leído correspondiente al nombre de usuario
// en un número entero para su posterior operación
int usrTxt2Int (char texto [12])
{
      int text[10] = {48,49,50,51,52,53,54,55,56,57};
      int fact[10]= {9,8,7,6,5,4,3,2,1,0};
      int ind1=0;
      int indc=0;
      int usuario=0;
      for (ind1=0; ind1<=10; ind1++)
      {
            for (indc=0; indc<=10; indc++)
            {
                  if (texto[ind1]==text[indc])
                  {
                        usuario+=indc* (int) pow(10,fact[ind1]);
                  }
            }
      }
      return usuario;
}

// Función que convierte el texto leído correspondiente a la clave del usuario
// en un número entero para su posterior operación
int clvTxt2Int (char texto [8])
{
      int text[10] = {48,49,50,51,52,53,54,55,56,57};
      int fact[6]= {5,4,3,2,1,0};
      int ind1=0;
      int indc=0;
      int clv=0;
      int shift=6-longClave;
      for (ind1=0; ind1<=longClave; ind1++)
```

```
	{
		for (indc=0; indc<=10; indc++)
		{
			if (texto[ind1]==text[indc])
			{
				clv+=indc* (int) pow(10,fact[ind1+shift]);
			}
		}
	}
	return clv;
}

// Función que carga las bases de datos en arreglos
// ------------------------------------------------

void cargarBases ()
{

	// Variables para almacenamiento temporal
	char usuarioCB[12];
	char clvCB[8];
	int cont=0;

	// BASE DE DATOS DEL BANCO
	// ***********************
	// -- Nombres de usuario --
	// Se abre el archivo que contiene los datos de los nombres de usuario
	if (longUsrBan==100000)
	{ptroUsrBan= fopen("usersBank100.txt","r");}
	if (longUsrBan==500000)
	{ptroUsrBan= fopen("usersBank500.txt","r");}
	if (longUsrBan==1000000)
	{ptroUsrBan= fopen("usersBank1000.txt","r");}
	while(cont<longUsrBan)
	{
		// Se toman doce caracteres de cada linea del archivo de texto
		fgets(usuarioCB,12,ptroUsrBan);
		// Se convierte el string en un entero con la función 'user_txt2int'
		bddUsrBan[cont]=usrTxt2Int(usuarioCB);
		//printf("%d\n",bddUsrBan[cont]);
		// Se pasa al siguiente usuario
		cont++;
	}
	printf("Se han leido %d registros de base de datos de usuarios del
banco\n",cont);
	fclose(ptroUsrBan);

	cont=0;

	// -- Contraseñas --
	// Se abre el archivo que contiene los datos de las contraseñas de acuerdo
a su longitud
	if (longClave==4 && longUsrBan==100000)
	{ptroClvBan = fopen("passwordsBank100_4.txt","r");}
	if (longClave==5 && longUsrBan==100000)
	{ptroClvBan = fopen("passwordsBank100_5.txt","r");}
	if (longClave==6 && longUsrBan==100000)
	{ptroClvBan = fopen("passwordsBank100_6.txt","r");}

	if (longClave==4 && longUsrBan==500000)
	{ptroClvBan = fopen("passwordsBank500_4.txt","r");}
	if (longClave==5 && longUsrBan==500000)
	{ptroClvBan = fopen("passwordsBank500_5.txt","r");}
	if (longClave==6 && longUsrBan==500000)
	{ptroClvBan = fopen("passwordsBank500_6.txt","r");}

	if (longClave==4 && longUsrBan==1000000)
```

```
	{ptroClvBan = fopen("passwordsBank1000_4.txt","r");}
	if (longClave==5 && longUsrBan==1000000)
	{ptroClvBan = fopen("passwordsBank1000_5.txt","r");}
	if (longClave==6 && longUsrBan==1000000)
	{ptroClvBan = fopen("passwordsBank1000_6.txt","r");}

	while(cont<longUsrBan)
	{
		// Se toman caracteres de cada linea del archivo de texto
		fgets(clvCB,paramLongClave,ptroClvBan);
		// Se convierte el string en un entero con la función 'user_txt2int'
		bddClvBan[cont]=clvTxt2Int(clvCB);
		//printf("%d\n",bddClvBan[cont]);
		// Se pasa al siguiente usuario
		cont++;
	}
	//printf("%d\n",bddClvBan[0]);
	//printf("%d\n",bddClvBan[1]);
	printf("Se han leido %d registros de base de datos de contrasenas del
banco\n",cont);
	fclose(ptroClvBan);


	// BASE DE DATOS GLOBAL
	// **********************
	// -- Nombres de usuario --
	// Se abre el archivo que contiene todos datos de los nombres de usuario
	cont=0;
	ptroUsrGlo = fopen("usersTot.txt","r");
	while(cont<longUsrTot)
	{
		// Se toman doce caracteres de cada linea del archivo de texto
		fgets(usuarioCB,12,ptroUsrGlo);
		// Se convierte el string en un entero con la función 'user_txt2int'
		bddUsrGlo[cont]=usrTxt2Int(usuarioCB);
		//printf("%d\n",bddUsrGlo[cont]);
		cont++;
	}
	printf("Se han leido %d registros de base de datos de maquinas
atacantes\n",cont);
	fclose(ptroUsrGlo);
}
```

EJEMPLO DE RESULTADO DE LA SIMULACIÓN

```
*****************************************************
**  SIMULACION DE ACCESO A LAS CUENTAS DE UN BANCO  **
**       MEDIANTE UN ATAQUE DE FUERZA BRUTA         **
**                                                  **
**  Realizado por: Juan Almendariz      Junio/2009  **
*****************************************************

DEFINICION DE NIVELES DE LOS FACTORES PARA SIMULACION:
-----------------------------------------------------
Maquinas atacantes: 10
Usuarios del banco: 100000
Longitud de contrasena: [4,5,6] 4
Numero maximo de intentos antes de bloqueo: [3,4,5] 3
Valores para tiempo de respuesta del servidor: [en s]
tsmin= 2.000000
tsmax= 3.000000
Intervalo de tiempo entre intentos de acceso: [en s] 4.000000

PREPARACION DE DATOS:
-----------------------------------------------------
Mailboxes creados...
Tablas creadas...
Cargando bases de datos...
Se han leido 100000 registros de base de datos de usuarios del banco
Se han leido 100000 registros de base de datos de contrasenas del banco
Se han leido 1000000 registros de base de datos de maquinas atacantes

EJECUCION DE SIMULACION:
-----------------------------------------------------
Simulando ataque...
Maquina 05 Usuario 0000495 1700010091 Clave 1136 Intento 1 Tiempo   144.38 s
Maquina 04 Usuario 0061484 1701231589 Clave 4715 Intento 3 Tiempo 18248.75 s
Maquina 02 Usuario 0084232 1701685271 Clave 8187 Intento 2 Tiempo 25466.69 s
                --- Usuarios intentados 100000 ---
Maquina 07 Usuario 0175447 1703512135 Clave 6225 Intento 2 Tiempo 53506.33 s
Maquina 07 Usuario 0190367 1703811222 Clave 9123 Intento 1 Tiempo 57839.39 s
Maquina 05 Usuario 0198555 1703975886 Clave 8338 Intento 2 Tiempo 59769.83 s
                --- Usuarios intentados 200000 ---
Maquina 03 Usuario 0220093 1704407673 Clave 5280 Intento 2 Tiempo 66392.77 s
Maquina 06 Usuario 0261666 1705238242 Clave 8838 Intento 2 Tiempo 79241.32 s
Maquina 07 Usuario 0286987 1705743563 Clave 7086 Intento 2 Tiempo 86419.85 s
                --- Usuarios intentados 300000 ---
Maquina 01 Usuario 0338501 1706773858 Clave 5808 Intento 1 Tiempo 101615.67 s
Maquina 01 Usuario 0378481 1707570964 Clave 2134 Intento 1 Tiempo 113346.37 s
                --- Usuarios intentados 400000 ---
Maquina 09 Usuario 0448349 1708970122 Clave 5159 Intento 1 Tiempo 133894.61 s
Maquina 00 Usuario 0460370 1709210858 Clave 0067 Intento 2 Tiempo 136783.23 s
                --- Usuarios intentados 500000 ---
Maquina 05 Usuario 0498895 1709979205 Clave 9680 Intento 3 Tiempo 150412.31 s
Maquina 09 Usuario 0565399 1711307288 Clave 8167 Intento 2 Tiempo 169066.84 s
Maquina 01 Usuario 0594811 1711896801 Clave 5350 Intento 1 Tiempo 178127.37 s
                --- Usuarios intentados 600000 ---
Maquina 05 Usuario 0615845 1712317609 Clave 6976 Intento 1 Tiempo 185393.65 s
Maquina 09 Usuario 0650029 1712998663 Clave 8759 Intento 3 Tiempo 194396.62 s
Maquina 00 Usuario 0688340 1713767778 Clave 7062 Intento 3 Tiempo 204879.21 s
Maquina 07 Usuario 0686767 1713736088 Clave 2783 Intento 2 Tiempo 205362.23 s
                --- Usuarios intentados 700000 ---
Maquina 04 Usuario 0709814 1714197769 Clave 3301 Intento 2 Tiempo 212580.96 s
Maquina 03 Usuario 0730633 1714614060 Clave 3617 Intento 3 Tiempo 218506.34 s
Maquina 03 Usuario 0740163 1714806336 Clave 2830 Intento 3 Tiempo 221501.32 s
```

```
Maquina 01 Usuario 0763251 1715269435 Clave 3941 Intento 1 Tiempo 228994.13 s
Maquina 09 Usuario 0799919 1716005069 Clave 7159 Intento 3 Tiempo 238757.66 s
                --- Usuarios intentados 800000 ---
Maquina 02 Usuario 0813412 1716274038 Clave 4569 Intento 3 Tiempo 245358.23 s
Maquina 04 Usuario 0839024 1716785678 Clave 5704 Intento 1 Tiempo 252285.33 s
Maquina 01 Usuario 0853201 1717069833 Clave 3849 Intento 3 Tiempo 255774.10 s
                --- Usuarios intentados 900000 ---
Maquina 07 Usuario 0919797 1718403700 Clave 0972 Intento 2 Tiempo 275129.60 s
Maquina 02 Usuario 0934572 1718700444 Clave 0167 Intento 3 Tiempo 282704.70 s
Maquina 06 Usuario 0957516 1719160093 Clave 0264 Intento 2 Tiempo 287424.59 s
Maquina 04 Usuario 0969114 1719390823 Clave 5629 Intento 2 Tiempo 291896.15 s
Maquina 06 Usuario 0974526 1719498550 Clave 6416 Intento 1 Tiempo 292622.94 s
Maquina 00 termina ejecucion de 100000 usuarios a 297826.25 s
Maquina 04 Usuario 0989144 1719789693 Clave 3506 Intento 3 Tiempo 298110.08 s
Maquina 09 termina ejecucion de 100000 usuarios a 298910.36 s
Maquina 07 termina ejecucion de 100000 usuarios a 299150.77 s
Maquina 05 termina ejecucion de 100000 usuarios a 299242.63 s
Maquina 06 termina ejecucion de 100000 usuarios a 299642.87 s
Maquina 01 termina ejecucion de 100000 usuarios a 299647.38 s
Maquina 03 termina ejecucion de 100000 usuarios a 299655.02 s
Maquina 08 termina ejecucion de 100000 usuarios a 299794.03 s
Maquina 04 termina ejecucion de 100000 usuarios a 299951.07 s
Maquina 02 termina ejecucion de 100000 usuarios a 299984.91 s


Resumen de actividad:
-------------------------------------------------
Numero envios = 1199967  Numero respuestas = 1199967
Numero cuentas terminadas = 1000000
Numero cuentas accedidas = 34
Numero cuentas bloqueadas = 99966
Numero cuentas con usuario no existente= 900000

      CSIM (Student) Simulation Report (Version 19.0 for MS Visual C/C++)

                           Tue Sep 22 19:01:18 2009


                     Ending simulation time:    299984.911
                     Elapsed simulation time:   299984.911
                     CPU time used (seconds):      965.656



TABLE 1:  Tiempos de Acceso a Cuentas

      minimum       162.977586          mean               171241.150177
      maximum    298110.813588          variance          8172748051.808767
      range      297947.836002          standard deviation  90403.252440
      observations          34          coefficient of var      0.527929
```

SCRIPT EN LENGUAJE PHP PARA OBTENCIÓN DE CÉDULAS

```
<?php
// SCRIPT QUE OBTIENE CÉDULAS VÁLIDAS DEL PORTAL DEL CNE
// Por: Juan Almendáriz
// Marzo 25 / 2009
// Modificado Diciembre 5 / 2009
// ----------------------------------------------------
// VARIABLES Y CONFIGURACIONES PRELIMINARES
// ***************************************
// Se asigna un tiempo de duración ilimitado al script
set_time_limit(0);
// Variable que contiene el tiempo de inicio del script
$start= time();
// Se abre archivo de texto para escritura de cédulas válidas
$base = fopen("base.txt", "w");
// Número de cédula inicial (9 dígitos) para la búsqueda
$digced=170000000;
// Contador de búsquedas realizadas
$busq=0;
// CÁLCULO DE DÉCIMO DÍGITO Y BÚSQUEDA DE CÉDULA
// *********************************************
// Lazo de ejecución por tantas veces como cédulas válidas se requiera
while ($cedval<100)
{
	// Cálculo del dígito verificador para cáda cédula
	$tocheck=$digced;
	$digit = array();
	$div=100000000;
	for($d = 0; $d < 9; $d += 1)
	{
		$digit[$d]=floor($tocheck/$div);
		$tocheck=$tocheck-$digit[$d]*$div;
		$div=$div/10;
	}
	$sum=0;
	$prod=0;
	$fact = array(2,1,2,1,2,1,2,1,2);
	for($d = 0; $d < 9; $d += 1)
	{
		$prod=$digit[$d]*$fact[$d];
		if ($prod>=10)
		{
			$sum=$sum+$prod-9;
		}
		else
		{
		$sum=$sum+$prod;
		}
	}
	// Décimo dígito de la cédula
	$d10=(10-($sum%10))%10;
	// Se asigna el décimo dígito
	$cedula=$digced*10+$d10;
	// Búsqueda de la cédula en el portal
	$url_cne =
file_get_contents("http://app.cne.gov.ec/lugarvotacion/1221151107.aspx?__EVENTTAR
GET=&__EVENTARGUMENT=&__VIEWSTATE=%2FwEPDwUKLTc3MTY5MzQwOWRk&dropTipo=C&txtCriter
io=".$cedula."&btnConsultar=Consultar&hdnCriterio=&__EVENTVALIDATION=%2FwEWBAKx%2
BJKsBwKT%2F7%2BkBwKvtu%2FvDgKVq7KvCA%3D%3D");
	// Verificación de validez de la cédula mediante string en el código
fuente de la página obtenida
```

```
        if(substr_count($url_cne, '>1<')==1)
        {
                $ced="$cedula\n";
                // Escritura a archivo de cédula válida
                fwrite($base , $ced );
                // Incremento de contador de cédulas válidas
                $cedval += 1;
        }
        // Incremento de cédula para búsqueda
        $digced += 1;
        // Incremento de contador de búsquedas
        $busq += 1;
}
// Finalización archivo de texto
fclose($base);
// Captura de tiempo de finalización
$stop= time();
// Tiempo de ejecución
$time= $stop-$start;
// Resultados mostrados en pantalla
echo "Se han buscado $busq cedulas</br>";
echo "Se han hallado $cedval cedulas validas</br>";
echo "Tiempo de busqueda: $time segundos</br>";
echo "EJECUCION TERMINADA";
?>
```

EJEMPLO DE OBTENCIÓN DE CÉDULAS REALES

Mensajes mostrados en el navegador:

```
Se han buscado 1794 cedulas
Se han hallado 1000 cedulas validas
Tiempo de busqueda: 649 segundos
EJECUCION TERMINADA
```

Contenido del archivo al que se escriben las cédulas obtenidas:

```
1700000035
1700000043
1700000050
1700000068
1700000076
1700000118
1700000126
1700000134
1700000159
1700000175
1700000183
1700000191
1700000209
1700000217
1700000225
1700000308
1700000316
1700000332
1700000357
1700000373
1700000381
1700000399
1700000415
1700000464
1700000506
1700000522
1700000563
1700000589
1700000597
1700000639
1700000647
1700000662
1700000670
1700000720
1700000746
1700000753
1700000761
1700000795
1700000803
...
```

PROGRAMA EN MATLAB UTILIZADO PARA LA GENERACIÓN DE LA BASE DE DATOS DE CÉDULAS FICTICIAS

```
% ------------------------------------------------------------------
% ARCHIVO .m QUE GENERA LAS BASES DE DATOS DE NOMBRES DE USUARIO PARA
% SIMULACIÓN DE ACCESOS A UN SISTEMA BANCARIO EN LÍNEA
% ------------------------------------------------------------------
% Por Juan Alejandro Almendariz
% Mayo 18 / 2009
% ------------------------------------------------------------------

close all
clear all

% CREACIÓN DE LA BASE DE DATOS DE USUARIOS GLOBAL
% ***********************************************
% Generación de nombres de usuario para Pichincha (empiezan con 17)
% Número de usuarios
L=1000000;
% CI inicial (se toman en cuenta solo los 9 primeros digitos)
CIi=170000000;
% Definición de los usernames
users_tot=zeros(1,L);
% A manera de ejemplo se generarán 1 000 000 de supuestos
% números de cédula válidos.
% Se toma en cuenta el numero de CI sin el último dígito
for u=1:L
    % Se genera números de cédula aleatorios usando
    % incrementos uniformemente distribuidos
    username=CIi+ceil(rand(1)*3);
    CIi=username;
    % Se obtienen los 9 digitos de cada CI para
    % calculo del decimo digito
    div=100000000;
    tocheck=username;
    digit=zeros(1,9);
    for d=1:9 % Para los nueve digitos
        digit(d)=floor(tocheck/div);
        tocheck=tocheck-digit(d)*div;
        div=div/10;
    end
    % Calculo del decimo digito. Método Módulo 10
    sum=0;
    prod=0;
    fact=[2 1 2 1 2 1 2 1 2];
    for d=1:9
        prod=digit(d)*fact(d);
        if prod>=10
            sum=sum+1+mod(prod,10);
        else
            sum=sum+prod;
        end
    end
    d10=mod((10-mod(sum,10)),10);
    % Asociacion de los 9 digitos de la CI y el decimo digito
    users_tot(u)=username*10+d10;
end
```

```
% CREACION DE LAS BASES DE DATOS DE USUARIOS DEL BANCO
% ************************************************
% De la base de datos global se escogen aleatoriamente un cierto número de
% usuarios que seran los usuarios del banco
% --- Base de 100 000 usuarios ---
users_bank100=zeros(1,L/10);
% Se escoge un usuario de cada diez usuarios de la base global como usuario
% del banco
for a=0:(L/10)-1
    users_bank100(a+1)=users_tot(a*10+ceil(rand(1)*10));
end
% --- Base de 500 000 usuarios ---
users_bank500=zeros(1,L/2);
% Se escoge un usuario de cada dos usuarios de la base global como usuario
% del banco
for a=0:(L/2)-1
    users_bank500(a+1)=users_tot(a*2+ceil(rand(1)*2));
end
% --- Base de 1 000 000 usuarios ---
users_bank1000=users_tot;

% CREACIÓN DE LA BASE DE DATOS DE CLAVES DEL BANCO
% ***********************************************
% --- Base de 100 000 usuarios ---
% Generacion de las passwords digitales para los usuarios del banco
% Passwords de 4 digitos de longitud
password100_4=round(rand(1,L/10)*9999);
% Passwords de 5 digitos de longitud
password100_5=round(rand(1,L/10)*99999);
% Passwords de 6 digitos de longitud
password100_6=round(rand(1,L/10)*999999);

% --- Base de 500 000 usuarios ---
% Generacion de las passwords digitales para los usuarios del banco
% Passwords de 4 digitos de longitud
password500_4=round(rand(1,L/2)*9999);
% Passwords de 5 digitos de longitud
password500_5=round(rand(1,L/2)*99999);
% Passwords de 6 digitos de longitud
password500_6=round(rand(1,L/2)*999999);

% --- Base de 500 000 usuarios ---
% Generacion de las passwords digitales para los usuarios del banco
% Passwords de 4 digitos de longitud
password1000_4=round(rand(1,L)*9999);
% Passwords de 5 digitos de longitud
password1000_5=round(rand(1,L)*99999);
% Passwords de 6 digitos de longitud
password1000_6=round(rand(1,L)*999999);

% CREACIÓN DE LOS ARCHIVOS QUE CONTIENEN LOS DATOS
% ************************************************
% Se crean los archivos de texto plano que contienen los
% numeros de usuarios y contraseñas

% Base de datos global
fid=fopen('usersTot.txt','w');
fprintf(fid,'%10.0f\n',users_tot);

% Base de datos del banco con 100 000 usuarios
fid=fopen('usersBank100.txt','w');
fprintf(fid,'%10.0f\n',users_bank100);
fid=fopen('passwordsBank100_4.txt','w');
fprintf(fid,'%4.0f\n',password100_4);
fid=fopen('passwordsBank100_5.txt','w');
fprintf(fid,'%5.0f\n',password100_5);
fid=fopen('passwordsBank100_6.txt','w');
```

```
fprintf(fid,'%6.0f\n',password100_6);

% Base de datos del banco con 500 000 usuarios
fid=fopen('usersBank500.txt','w');
fprintf(fid,'%10.0f\n',users_bank500);
fid=fopen('passwordsBank500_4.txt','w');
fprintf(fid,'%4.0f\n',password500_4);
fid=fopen('passwordsBank500_5.txt','w');
fprintf(fid,'%5.0f\n',password500_5);
fid=fopen('passwordsBank500_6.txt','w');
fprintf(fid,'%6.0f\n',password500_6);

% Base de datos del banco con 1 000 000 usuarios
fid=fopen('usersBank1000.txt','w');
fprintf(fid,'%10.0f\n',users_bank1000);
fid=fopen('passwordsBank1000_4.txt','w');
fprintf(fid,'%4.0f\n',password1000_4);
fid=fopen('passwordsBank1000_5.txt','w');
fprintf(fid,'%5.0f\n',password1000_5);
fid=fopen('passwordsBank1000_6.txt','w');
fprintf(fid,'%6.0f\n',password1000_6);

fclose('all');
```

RECURSOS DE HARDWARE UTILIZADOS PARA LA SIMULACIÓN

Debido a la complejidad de las simulaciones y, sobretodo, a su largo tiempo de ejecución, se vio necesaria la utilización de varias computadoras para realizar todas las corridas necesarias para obtener los datos objeto del estudio realizado en este capítulo.

En la tabla siguiente se detallan las características de los equipos que se emplearon para efectuar la simulación. Se destacan, como es lógico, la capacidad de procesamiento de los mismos y la cantidad de memoria RAM disponible.

Tipo	Procesador	Memoria RAM
Servidor *blade*	Quad Core Intel Xeon @ 2.50 GHz	4 GB
Desktop PC	Intel Core 2 Duo @ 2.13 GHz	4 GB
Netbook	Intel Atom @ 2.00 GHz	2 GB

Tabla A.1 Equipos utilizados en la simulación.

Generalmente, la distribución de las corridas de la simulación se la realizó de la siguiente manera: las pruebas para 1 000 000 de usuarios en el servidor *blade*, las de 500 000 usuarios en la *desktop* PC y, las simulaciones para 100 000 usuarios en la *netbook*.