INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo

ESCOM

Trabajo Terminal

"Aplicación de cifrado contra de adversarios clasificadores, para el correo electrónico"

2015-A010

Presentan

Arcos Ayala Jonathan
Zepeda Ibarra Allan Ulises

Directores

Dra. Sandra Díaz Santiago
M. en C. Manuel Alejandro Soto Ramos

Noviembre 2015

Índice

Índice de Figuras

Índice de Tablas

Introducción

Actualmente, una gran cantidad de personas hacen uso del internet y de las nuevas tecnologías para comunicarse. Con ello, también se incrementa la cantidad de información que se transmite y/o almacena. En diversas ocasiones, esta información es susceptible a sufrir distintos tipos de ataques, tales como acceso no autorizado, modificación o destrucción de la misma, entre otros. Adicionalmente, cada día aparecen nuevos tipos de ataques a los sistemas de información. Por lo tanto, surge la necesidad de proteger dicha información.
Una de las tecnologías ampliamente usada para comunicarse es el correo electrónico [1]. Los mensajes que envían y reciben los usuarios de correo electrónico pueden ser de diferentes tipos: personales, transaccionales, de notificación o de publicidad. Por lo tanto, cada vez que se escribe y envía un correo electrónico, se está revelando información acerca de las preferencias y/o intereses del usuario. Estos datos, son el insumo más importante, para distintas entidades, entre las cuales están empresas que realizan publicidad en línea, proveedores de internet, instituciones de gobierno, entre otros [2]. El propósito de tener estos datos puede ser realizar publicidad efectiva, vender los datos a empresas de publicidad o averiguar si determinado usuario es una amenaza para el gobierno. Para obtener información acerca de los intereses y/o preferencias del usuario, se hace uso de programas de cómputo denominados *clasificadores*. Los clasificadores son herramientas informáticas que analizan una gran cantidad de información, haciendo uso de técnicas de aprendizaje máquina [3], y posteriormente clasifican un mensaje en determinada categoría o perfil. En este contexto, los clasificadores pueden constituir una amenaza para algunos usuarios del correo electrónico, por tal motivo de ahora en adelante a los programas que clasifican se les denominará *adversarios clasificadores*.
Ante tal escenario, surge la pregunta ¿cómo se puede proteger un usuario contra los adversarios clasificadores?. Una posible respuesta es hacer uso de algoritmos de cifrado estándar. Sin embargo, hacer uso de tales algoritmos, implica que los participantes en la comunicación acuerden una clave de cifrado. Desafortunadamente, acordar una clave, no es un proceso sencillo para el usuario común. Otra desventaja de esta primera solución, es que los algoritmos de cifrado estándar ofrecen un alto nivel de seguridad, el cual resulta excesivo cuando se consideran los recursos y el objetivo de un adversario clasificador [4].

Capítulo 1

Marco Teórico

En este capítulo hablaremos sobre el correo electrónico y algunos de los intermediarios que hacen posible que este servicio sea usado en toda la red de internet, también hablaremos de las amenazas a las que se tiene que enfrentar este servicio para hacer llegar información de un usuario a otro de una manera segura y cuales han sido las respuestas de los proveedores de servicio de correo electrónico para proporcionar dicha seguridad a los usuarios.

1.1. Correo electrónico

El correo electrónico es el servicio de internet por el cual se pueden enviar mensajes entre 2 usuarios que cuenten con este servicio. El correo electrónico o "e-mail" por sus siglas en inglés, basa su funcionamiento en las oficinas postales. Tomando esa analogía podemos decir que los usuarios tienen cartas (mensajes de correo electrónico) que desean enviar a otros usuarios; estas cartas son enviadas a las oficinas postales (servidores de correo electrónico) donde se almacenan con otras cartas que van dirigidas a otros usuarios en diferentes ciudades; estas oficinas postales envían las cartas a otras oficinas postales si es necesario; y por último cada oficina postal se encarga de colocar el mensaje en el buzón del usuario correspondiente (buzón de correos electrónicos).
Para poder entender la comunicación en correo electrónico necesitamos definir los siguientes conceptos:

- **Mensaje de correo electrónico** Los mensajes de correo electrónico al igual que las cartas tienen la dirección del receptor, la dirección del emisor y el cuerpo del mensaje, pero a diferencia de las cartas los correos electrónicos son enviados por internet y necesitan sus propios formatos. Para poder enviar un mensaje de correo electrónico es necesario tener los siguientes elementos como mínimo.
  Dirección del remitente:
  Esta dirección se compone por dos elementos importantes, el primero es el nombre de usuario seguido de un carácter "@", el segundo elemento es el dominio donde está alojado el servicio de correo electrónico.
  Direcciones de los receptores:
  En un correo electrónico debe haber al menos una dirección de receptor para ser enviado, esta dirección de receptor tiene la misma estructura que la dirección del remitente.
  Contenido del mensaje:
  Es el texto que se desea transmitir entre el remitente y el receptor.

El mensaje de correo electrónico cuenta con más elementos pero estos son opcionales y se pueden consultar en el RFC 2821 extensión MIME.

- **Dominio de internet**
  Es un nombre que identifica a los diferentes dispositivos interconectados en una red sin la necesidad de aprenderse un número de red. Estos dispositivos pueden proporcionar diferentes servicios como el servicio de correo electrónico.

- **Buzón de correo electrónico** Un buzón de correo electrónico es un espacio virtual proporcionado por el servicio de correo electrónico que sirve para almacenar los mensajes enviados y recibidos.

- **Usuario de correo electrónico** Se le conoce como usuario de correo electrónico a la persona que se registra en un dominio para obtener los servicios de mensajería proporcionados por un servidor de correo electrónico.

- **Cliente de correo electrónico** El cliente de correo electrónico es una interfaz por la cual el usuario de correo electrónico puede administrar sus mensajes enviados y recibidos.
  El cliente de correo electrónico puede ser de diferentes tipos como son:

  Cliente de correo electrónico para la pc:
  Estos clientes de correo electrónico son instalados en una computadora y es configurado para sincronizarse con un servidor de correo electrónico cada cierto tiempo o cuando el usuario se lo indique. Una de las principales características de éste cliente de correos es la capacidad de descargar los mensajes del servidor a la computadora para ser leídos sin la necesidad de tener una conexión de internet y en cuanto tiene conexión con el servidor otra vez descarga los nuevos mensajes y envía los que se tienen pendientes en la computadora.

  Cliente de correo electrónico web:
  Los clientes web necesitan un explorador de internet y una conexión permanente a la red para que funcione, estos clientes normalmente son proporcionados por el mismo servidor de correo electrónico y nunca descarga los correo en las computadora donde son utilizados.

  Cliente de correo electrónico para móviles:
  Un cliente de correo electrónico móvil se caracteriza por ser una aplicacion instalada en un dispositivo móvil. Esta aplicación descarga una copia temporal de los últimos mensajes recibidos.

- **Servidor de Correo electrónico** Un servidor de correo electrónico es un programa que se encarga de enviar y recibir los mensajes de correo electrónicos de sus usuarios registrados, este servidor puede recibir mensajes de usuarios de otros servidores de correos que sean dirigidos a sus usuarios registrados.
  Este servidor tiene que seguir algunos estándares que existen en internet para el envío(protocolo smtp) y recepción(protocolo pop3 o imap) de mensajes de correo electrónico.

- **Protocolo SMTP** El protocolo SMTP significa "protocolo para transferencia simple de correo" o "Simple Mail Transfer Protocol" por sus siglas en inglés, el cual se encarga de enviar los mensajes de correo electrónicos entre dispositivos que se encuentran

interconectados en la red o en internet. Este protocolo solo se utiliza para mandar los mensajes entre servidores o entre el usuario emisor y su servidor de correo electrónico. Podemos revisar el protocolo completo en el RFC5321 [6].

- **Protocolo POP3** Este protocolo se encarga de descargar los mensajes del servidor a un cliente de correo electrónico que el usuario haya configurado previamente. Una de las características es que solo se sincroniza para la descarga de los mensajes de correo y no deja una copia de seguridad en el servidor de correo electrónico. Podemos consultar el funcionamiento detallado en el RFC1939 [7].

- **Protocolo IMAP** Este protocolo, al igual que el protocolo POP3, se encarga de la descarga de los mensajes del servidor a un cliente de correo electrónico con la diferencia de que la sincronización entre el servidor de y el cliente es continua, manteniendo una copia de seguridad en el servidor. Con este protocolo es posible tener varios clientes de correo configurados con la misma cuenta y los cambios que se realicen en cualquiera de los clientes de correo se verán reflejados en el servidor y en los diferentes clientes sincronizados. Este protocolo puede ser consultado en el RFC 6851 [8].

Como hemos podido ver en el presente documento, el servicio de correo electrónico es un canal de comunicación que se ayuda de varios elementos para completar la comunicación entre dos usuarios de correo electrónico, no importando que estos estén registrados en 2 servidores de correo diferentes. Pero para poder entender por completo el comportamiento de este servicio tenemos que definir ciertas características de este servicio.
Este servicio establece una comunicación no orientada a conexión, lo que significa que el receptor no necesita estar conectado al servicio de correo electrónico para recibir el mensaje en su buzón de correo electrónico.
Los mensajes enviados por medio del correo electrónico transitan por el internet como archivos en texto plano, esto quiere decir que cualquiera que tenga una copia del mensaje puede abrir el correo electrónico y leer el mensaje siguiendo la estructura del protocolo SMTP.
Además de mandar mensajes tiene la facultad de enviar archivos multimedia en el mismo mensaje de correo electrónico, esta característica ha sido explotada bastante por empresas privadas y de gobierno para tener comunicados a sus empleados, departamentos, proveedores, socios, etc.
El correo permite enviar el mismo mensaje a más de un usuario sin la necesidad de hacer un mensaje para cada usuario, esto lo han utilizado muchas empresas de publicidad para hacer publicidad a gran escala y a muy bajo costo, a este servicio se le llama SPAM.
Este medio de comunicación es bastante rápido ya que se estima que un mensaje de correo electrónico tiene que llegar a su destino a más tardar en 5 minutos, no importando la ubicación geográfica del servidor de correo electrónico.
Las características mencionadas nos dan a notar que el correo electrónico tiene una baja seguridad al momento de enviar los mensajes, ya que la información que mandamos es sumamente fácil de leer por cualquiera que pueda tener una copia del mensaje. Si tomamos en cuenta que un mensaje de tiene que saltar por varios servidores antes de llegar a su destino, es fácil suponer que se puede interceptar una copia del mensaje en el envío de servidor a servidor.
Por estos motivos los servidores de correo electrónico han implementado diversos candados de seguridad para blindar la transferencia de mensajes entre servidores. Una de las maneras

que han encontrado para proporcionar dicha seguridad ha sido a través de la implementación de técnicas criptográficas.

1.2. Criptografia

Es el estudio de técnicas matemáticas relacionadas con la seguridad para proveer, confidencialidad, integridad y autenticación. Estas tecnicas están destinadas a alterar las representaciones lingüísticas de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados.

Como ya se menciono la criptografia provee diferentes servicios al momento del envío de mensajes entre usuarios, estos servicios son:

- **Confidencialidad**: Es que la información sea accesible sólo para aquéllos que están autorizados.
- **Integridad**: Es que la información sólo puede ser creada y modificada por quien esté autorizado a hacerlo.
- **Autenticación**: Es tener la certeza de que el mensaje provino del aparente autor o fuente.
- **No repudio**: Es que la información debe ser accesible para su consulta o modificación cuando se requiera.

1.2.1. Tipos de Ataques

A pesar de la implementación de estas técnicas criptográficas los ataques a las comunicaciones siguen existiendo y estos ataques se han clasificado dependiendo de cuanta información tenga disponible el atacante para poder romper el cifrado y obtener la información deseada. Los tipos de ataques son:

- **Ataque de texto cifrado(Known ciphertext attack)**: Este tipo de ataque tiene acceso sólo al texto cifrado, y no tiene acceso al texto plano, pero es el ataque más débil debido a la falta de información.
- **Ataque de texto plano(Known plaintext attack)**: Este tipo de ataque se supone que tienen acceso a la lista en un número limitado de pares de texto plano y el texto cifrado correspondiente.
- **Ataque de texto cifrado elegido(Chosen ciphertext attack)**: Este ataque es capaz de elegir una serie de textos planos para ser cifrados y tener acceso al texto cifrado resultante. Esto le permite explorar cualquier área del espacio de texto plano que desea y puede permitirle explotar el comportamiento no aleatorio que sólo aparecen con ciertos textos planos.

- **Ataque de texto plano elegido(Chosen plaintext attack)**: Este ataque se puede elegir los textos cifrados arbitrariamente y tener acceso a texto planos después de ser procesados. Para que este ataque se lleve a cabo es necesario tener el extremo receptor de la comunicación y acceso al canal de la comunicación.

1.2.2. Cifrado

Es el procedimiento por el cual la información es transformada con el fin de que esta sea ilegible. Dicha transformación se realiza por medio de una función de cifrado que tiene como parámetro de entrada un archivo de texto conocido como texto plano (Plaintext) y una pieza de información que sirve para verificar el acceso al servicio de cifrado e indica como es la trasformación del texto plano esta pieza es conocida como clave. La conjunción de estas piezas con la función de cifrado da como resultado un texto ilegible conocido como texto cifrado (Ciphertext).
En el siguiente diagrama se ejemplifica el proceso anteriormente descrito:

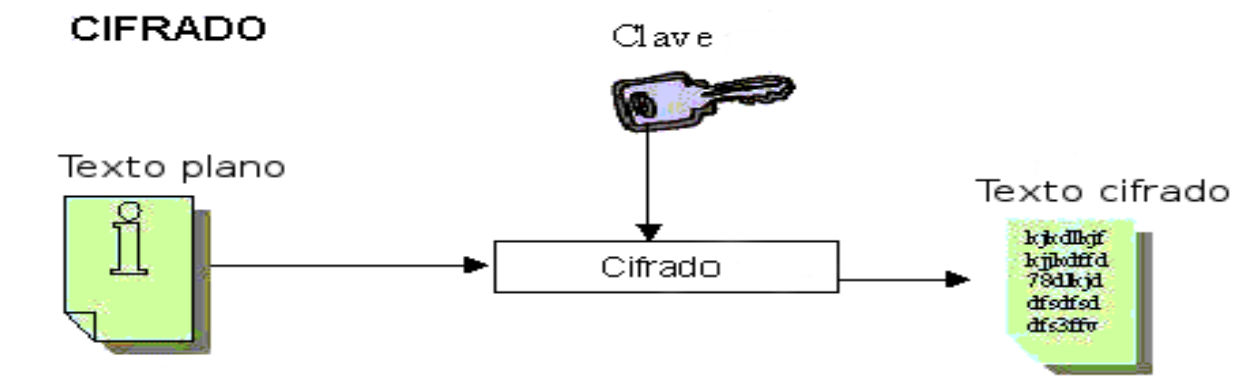

Figura 1.1: Diagrama cifrado

Cifrado Simétrico:
Un sistema de cifrado simétrico es un tipo de cifrado que usa una misma clave para cifrar y para descifrar. Las dos partes que se comunican mediante el cifrado simétrico deben estar de acuerdo en la clave a usar de antemano. Una vez de acuerdo, el remitente cifra un mensaje usando la clave, lo envía al destinatario, y éste lo descifra usando la misma clave.
Un buen sistema de cifrado pone toda la seguridad en la clave y ninguna en el algoritmo. En otras palabras, no debería ser de ninguna ayuda para un atacante conocer el algoritmo que se está usando. Sólo si el atacante obtuviera la clave, le serviría conocer el algoritmo.
Dado que toda la seguridad está en la clave, es importante que sea muy difícil adivinar el tipo de clave. Esto quiere decir que el abanico de claves posibles, o sea, el espacio de posibilidades de claves, debe ser amplio.
Hoy por hoy, los ordenadores pueden adivinar claves con extrema rapidez, y ésta es la razón por la cual el tamaño de la clave es importante en los sistemas modernos. El algoritmo de cifrado DES usa una clave de 56 bits, lo que significa que hay 2^{56} claves posibles. 2^{56}

son 72.057.594.037.927.936 claves. Esto representa un número muy alto de claves, pero una máquina computadora de uso general puede comprobar todo el espacio posible de claves en cuestión de días. Una máquina especializada lo puede hacer en horas. Por otra parte, algoritmos de cifrado de diseño más reciente como 3DES, Blowfish e IDEA usan todos claves de 128 bits, lo que significa que existen 2^{128} claves posibles. Esto representa muchas, muchísimas más claves, y aun en el caso de que todas las máquinas del planeta estuvieran cooperando, todavía tardarían más tiempo que la misma edad del universo en encontrar la clave. [9]
A continuación veremos un diagrama de cómo es que viaja la información por medio de un esquema de cifrado simétrico.

Figura 1.2: Diagrama cifrado simetrico

Cifrado por bloques:
Este tipo de cifrado toma bloques de información del texto plano y produce bloques de información cifrados de un tamaño fijo, normalmente es del mismo tamaño que el bloque de información del texto plano. Los bloques de cifrado tienen que cumplir la condición de ser lo suficientemente grandes como para evitar ataques de texto cifrado, otra condición que se debe cumplir es que la asignación de bloques de entrada a bloques de salida es uno a uno para hacer el proceso reversible.
Para hacer la asignación de bloques los algoritmos de cifrado utilizan sustituciones y permutaciones en los bloques de texto plano hasta obtener un texto cifrado.
La sustitución es el remplazo de un bloque de k bits por otro bloque de k bits en un espacio de 2^k.
La permutación es un tipo de sustitución en el cual se busca que el bloque cifrado obtenido conserve el mismo número de unos y ceros pero no necesariamente en el mismo orden que el bloque de entrada
Los algoritmos de cifrados por bloques funcionan utilizando una función de permutación y/o sustitución donde una de los parámetros es una subclave de la clave que proporciona el usuario, esta función es aplicada sucesivamente a un bloque de texto en claro hasta obtener un texto cifrado. El número de funciones aplicadas sucesivamente depende del nivel de seguridad deseado.

Modos de operación:
ECB: Este modo de operación cuenta con una clave de cifrado, una función de cifrado, los bloques de texto plano para poder generar los bloques texto cifrado. En este modo de operación la creación se los bloques cifrados se hace poniendo un bloque de texto plano con la clave de cifrado en la función de cifrado para obtener un bloque de texto cifrado.

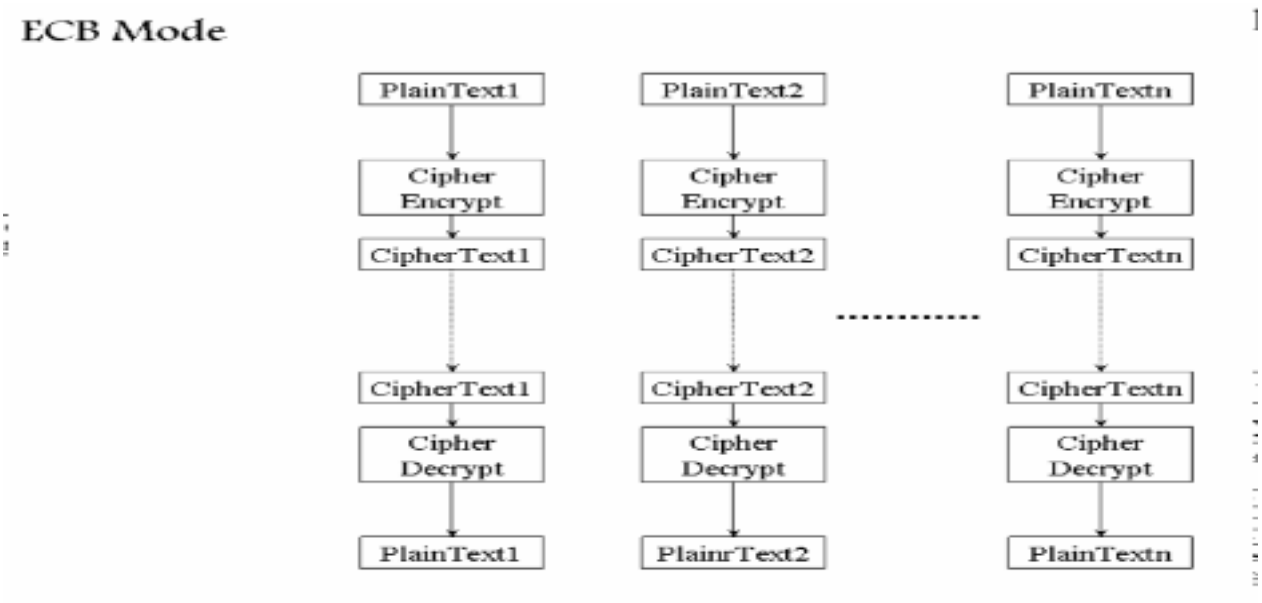

Figura 1.3: Diagrama ECB

Como podemos ver la creación de los bloques de cifrado solo depende de los bloques de texto plano, por lo tanto se pueden realizar de manera directamente e independientemente de los otros bloques de texto plano. A continuación tenemos el diagrama de este modo de cifrado.

CBC: Este modo de operación sigue el mismo procedimiento que el modo EBC con la diferencia de que cada texto plano antes de ser enviado a la función de cifrado se le aplica una operación XOR con el resultado del cifrado del bloque anterior, en caso de que sea el primer bloque de cifrado es necesario un vector de inicialización.

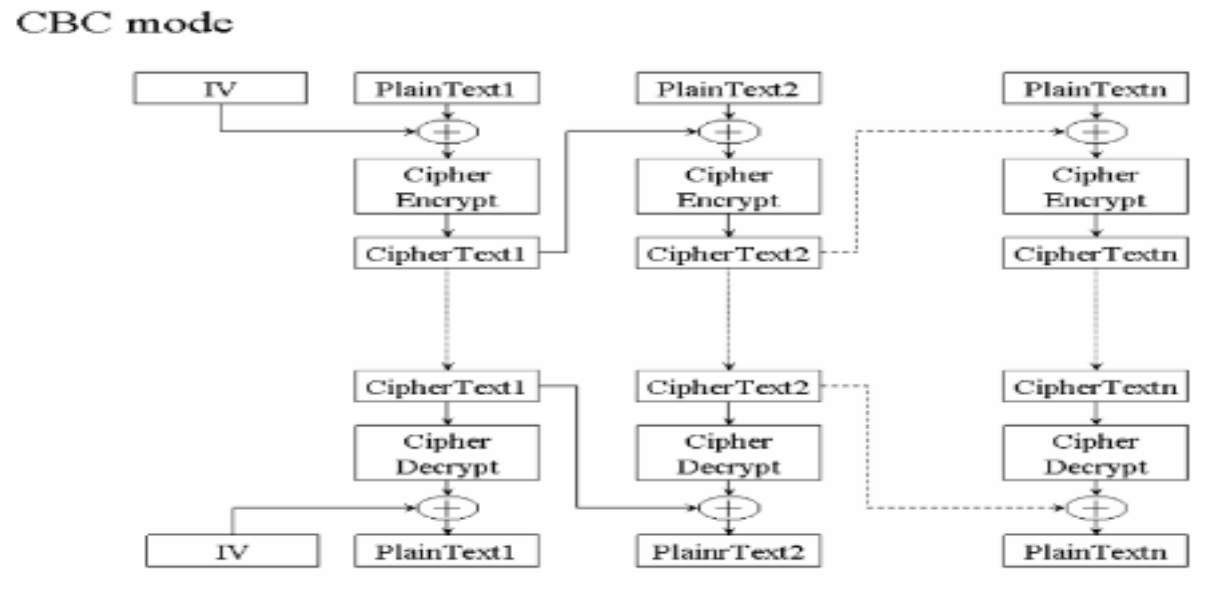

Figura 1.4: Diagrama CBC

CFB: Este modo de operación envía a la función de cifrado la clave de cifrado y un vector de inicialización en lugar del texto plano y el resultado de la función se le aplica una operación XOR junto con el bloque de texto plano y ese resultado es tomado como el bloque de texto cifrado.

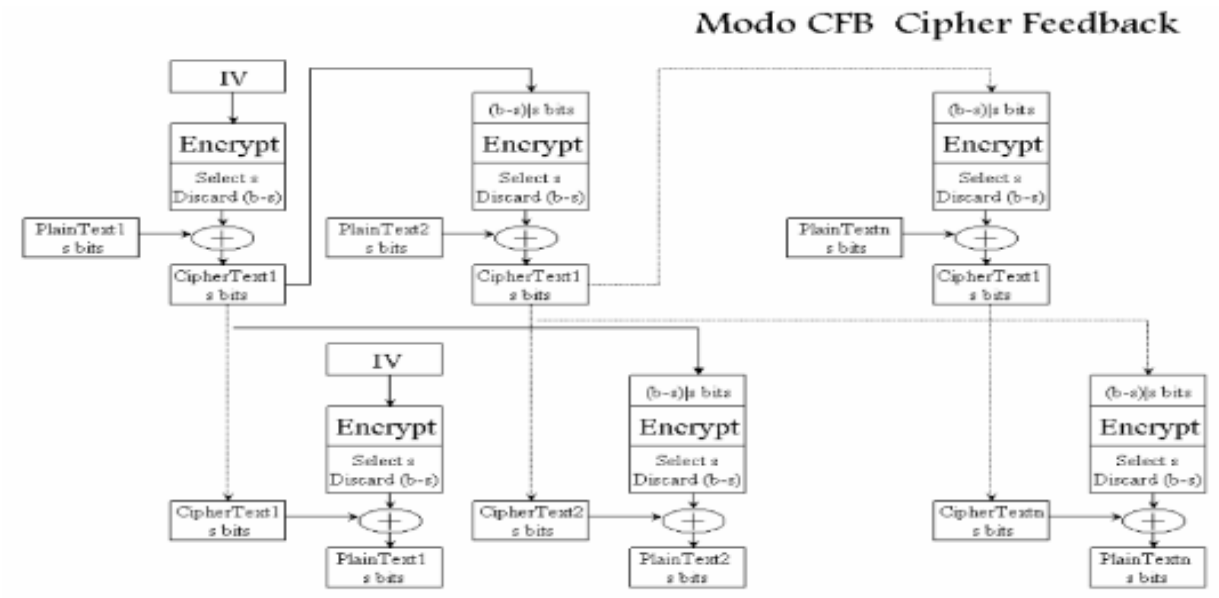

Figura 1.5: Diagrama CFB

El vector de inicialización depende del resultado del bloque anterior ya que se toma el resultado de la función de cifrado como vector de inicialización del bloque siguiente.

CTR: Este modo de operación se ejecuta de la mima manera que el modo OFB con la

diferencia que le vector de inicialización es generado por un bloque llamado contador, el cual cambia una vez que es implementado en otro bloque de cifrado.

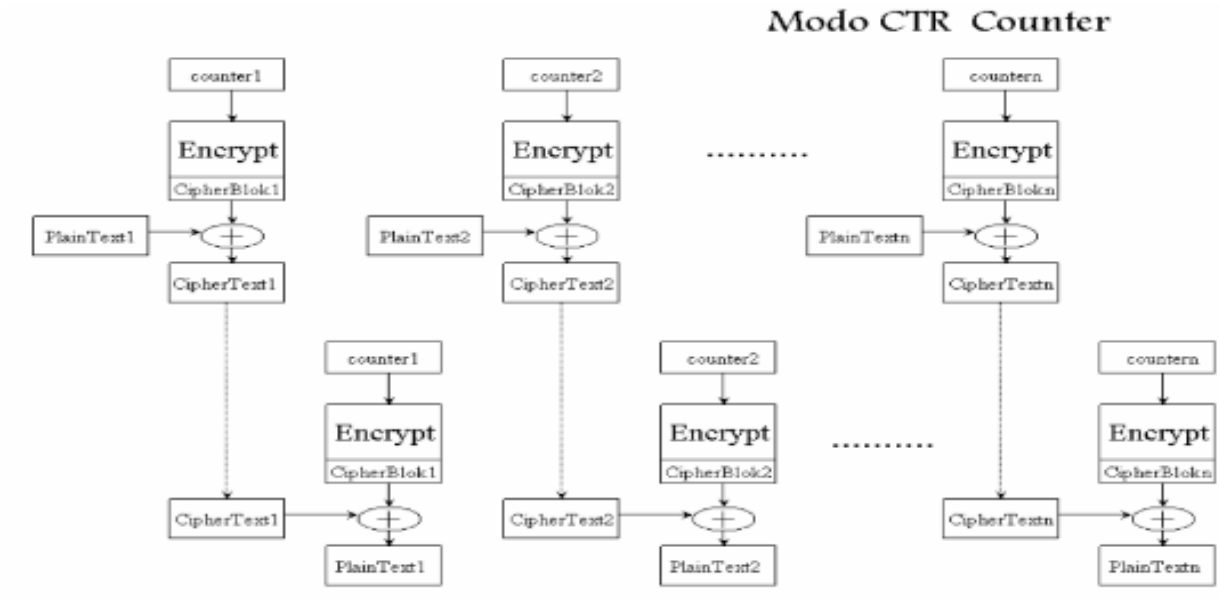

Figura 1.6: Diagrama CTR

1.3. CAPTCHA

son las siglas de prueba de Turing completamente automática y pública para diferenciar computadoras de humanos (Completely Automated Public Turing test to tell Computers and Humans Apart). Es un programa informático diseñado para diferenciar un ser humano de un ordenador, un CAPTCHA es una prueba que es fácil de pasar por un usuario humano pero difícil de pasar por una máquina. Uno de los CAPTCHAs más comunes son imágenes distorsionadas de cadenas cortas de caracteres. Para un humano es generalmente muy fácil de recuperar la cadena original de la imagen distorsionada, pero es difícil para los algoritmos de reconocimiento de caracteres el recuperar la cadena original de la imagen distorsionada. La criptografía moderna se basa en dos corrientes metodológicas que son la criptografía simétrica y la criptografía asimétrica. Estas dos técnicas tienen como propósito ocultar el contenido de un mensaje con el fin de que solo sea leído por aquellos que estén autorizados, a esto se le llama cifrado.

1.4. PGP (Pretty Good Privacy).

Es un programa desarrollado por Phil Zimmermann y cuya finalidad es proteger la información distribuida a través de Internet mediante el uso de criptografía de clave pública, así como facilitar la autenticación de documentos gracias a firmas digitales.
PGP es un sistema híbrido que combina técnicas de criptografía simétrica y criptografía asimétrica, la velocidad de cifrado del método simétrico y la distribución de la claves del método asimétrico.
Cuando un usuario emplea PGP para cifrar un texto en claro, dicho texto es comprimido. La

compresión de los datos ahorra espacio en disco, tiempos de transmisión; después de comprimir el texto, PGP crea una clave de sesión secreta que solo se empleará una vez. Esta clave es un número aleatorio generado a partir de los movimientos del ratón y las teclas que se pulsen; Esta clave de sesión se usa con un algoritmo para cifrar el texto claro; una vez que los datos se encuentran cifrados, la clave de sesión se cifra con la clave pública del receptor y se adjunta al texto cifrado enviándose al receptor.
El descifrado sigue el proceso inverso. El receptor usa su clave privada para recuperar la clave de sesión, simétrica, que PGP luego usa para descifrar los datos. [11]

1.5. Adversarios

Se aplica a la persona o grupo que es rival, competidor o contrario. En criptografía un adversario es un atacante con capacidades especiales que trata de romper la seguridad de un esquema criptografico.
Clasificadores: Los adversario clasificadores son programas que se dedican a observar los mensajes que se intercambian entre los usuarios de correo electrónico, con el fin de clasificarlos e identificar a todos los usuarios que cumplan con cierto criterio. Esta clasificación se hace de manera masiva y la realiza haciendo una búsqueda de palabras clave dentro de los mensajes de los usuarios. Por ejemplo, el clasificador puede estar interesado en los mensajes que contienen la palabra clave "Bomba", así que todos los mensajes que contengan esta palabra serán etiquetados en una clasificación en específico, este proceso se lleva a cabo por medio de técnicas de “Reconocimiento de patrones” y “Aprendizaje Máquina” para encontrar y clasificar los mensajes que intercepta. [5] [4]
La clasificación de estos mensajes tiene diversos usos, ya que pueden ser clasificados con fines demográficos, con fines comerciales o con fines gubernamentales. Todo esto con el propósito de generar las estadísticas de comportamientos e intereses de los usuarios de correo electrónico.

Nosotros nos enfocaremos a un tipo de ataque llamado “ataque de texto cifrado” o COA que hace referencia a su nombre en inglés “ Ciphertext-only attack” [10]. Este ataque solo cuenta con los textos cifrados que va recopilando de un canal o base de datos, estos textos cifrados los utiliza para hacer un análisis criptográfico de cómo se comporta la técnica de cifrado y trata de hallar el texto en claro a partir de los textos cifrados que va recopilando.
Este tipo de ataques es muy común en el internet aunque con muy baja efectividad cuando se implementa en comunicaciones altamente protegidas, y cuando se implementa en canales de comunicación desprotegidos la información obtenida llega a ser muy pobre. En los últimos años se han dado cuenta que si este tipo de adversarios atacan las comunicaciones sin cifrado se obtienes características valiosas sobre los usuarios que utilizan este tipo de canales de comunicación, este tipo de ataques son ejecutados por adversarios clasificadores.

Capítulo 2

Antecedentes

La única referencia que se tiene sobre un esquema criptográfico contra adversarios clasificadores es el que encontramos en el artículo "Defending Email Communication Against Profiling" de Philippe Golle y Ayman Farahat, ambos miembros del "Palo Alto Research Center".
En su artículo se aborda el ataque de adversarios clasificadores sobre los mensajes de correo electrónico, en el cual se proponen un protocolo para la comunicación por correo electrónico.Este protocolo hace uso de una función de cifrado que sustituye cada una de las palabras del mensaje por otra de la misma extención y frecuencia gramatical, esta función esta pensada para textos en idioma ingles. Esta función tiene como parámetro una clave K. Para calcular esta clave se usan los datos de cabecera que acompañan el mensaje los cuales pueden ser dirección del remitente, la dirección del destinatario, la hora a la que se envía el correo electrónico y potencialmente otros campos. Estos datos se introducen en una función hash lenta y el resultado de esta función es la clave K. Estas funciones hash pueden ser funciónes hash conocidas pero con una complejidad de calculo moderadamente mas alta.
Este protocolo resulta inseguro para la criptografia clásica pero es efectivo contra el ataque de clasificadores. Por otro lado este protocolo resuelve dos problemas, le permite a los usuarios calcular la clave de cifrado y descifrado fácilmente ya que los datos del mensaje con que se calcula son públicos, resolviendo así el intercambio de claves. Al usar un cifrado de tipo semántico permite que el texto se vea como una texto en ingles pero indistinguible para los clasificadores y por lo tanto este clasifica incorrectamente el mensaje cifrado.
Tomando en cuenta que esta es la única referencia sobre el tema que estamos abordando, podemos decir que trabajamos en el mismo sentido ya que esta propuesta de solución está basada en el artículo "On Securing Communication from Profilers" [4, 5] en el que se propone algo similar en cuanto al cifrado, nos lleva a tomar muy en cuenta el poder del algoritmo de cifrado que se usará en nuestra propuesta de solución.

Capítulo 3

Objetivos

3.1. Objetivos Generales

Desarrollar una herramienta para un cliente de correo electrónico que permita cifrar el contenido de los mensajes para evitar su clasificación, haciendo uso de tecnicas criptograficas simetricas y un servidor que verifique el envio y recepción de CAPTCHAS entre usuarios.

3.2. Objetivos Específicos

1. Desarrollar una herramienta en un cliente de correo electrónico para el envío y recepción de los correos cifrados y la generación, envío y recepción de CAPTCHAS.

2. Desarrollar un servidor de llaves que reciba, aloje y envíe los CAPTCHAS a los usuarios para descifrar los correos electrónicos.

3. Desarrollar un algoritmo de cifrado y descifrado basado en el envío y recepción de CAPTCHAS.

Capítulo 4

Justificación

La comunicación por medio del Correo electrónico es atacada constantemente y por ellos se han creado diferentes herramientas para asegurar la transferencia de información entre los usuarios. Pero estas herramientas ofrecen un conjunto de servicios como confidencialidad, no repudio, autenticación, entre otros y es porque están pensadas para hacer frente a adversarios mejor capacitados en la adquisición de información de los usuarios de correo electrónico.
Estas herramientas al enfrentarse a adversarios más capacitados necesitan implementar esquemas y técnicas más sofisticadas para establecer una comunicación segura entre usuarios y el mayor reto que se les presenta es el intercambio de claves, porque si un adversario llega a obtener al menos una clave, el esquema de seguridad se considera roto y la comunicación es vulnerable al ataque del o los adversarios que tengan esa clave robada.
Si tomamos en cuenta que los adversarios clasificadores son programas de cómputo que solo lee el contenido del correo y busca palabras específicas no necesitamos tantos servicios criptográficos para detener sus ataques a los correos electrónicos. Sería suficiente con tener un esquema de cifrado que nos proporcione confidencialidad durante el envío de mensajes.
Pero este esquema no solo tiene que preocuparse por la confidencialidad en el envío de los mensajes, también se enfrenta al problema de intercambio de claves para poder descifrar el mensaje por el usuario que recibe el mensaje.
Por lo tanto en este trabajo terminal se propone utilizar CAPTCHAS para el envío de claves entre los usuarios. Los CAPTHCAS contienen una cadena de caracteres que al ser resueltos por un ser humano es posible calcular la clave con que fue cifrado el mensaje, y como nuestro adversario clasificador es un programa de cómputo, le es muy complicado encontrar la clave para descifrar el mensaje y poderlo clasificar correctamente.

Capítulo 5

Propuesta de solución

Tomando en cuenta la información ya vertida en este documento, a continuación se explicará detalladamente la propuesta de solución que hemos ideado.
En la figura 5.1 tenemos el diagrama general del sistema, se puede apreciar la comunicación entre las diferentes entidades que usaremos, que datos se mandan y reciben y por que canales transitan estos datos. A continuación describiremos de manera general como es el proceso de envío y recepción de correos electrónicos ideado para este esquema.

1. Envío
    - El remitente escribe el correo electrónico y le da enviar.
    - El correo electrónico pasa por el complemento del cliente de correo.
    - El cliente genera a partir del correo una clave que usaremos para cifrar el mensaje.
    - Se cifra y se empaqueta el mensaje con el protocolo SMTP.
    - Se coloca una bandera en el mensaje.
    - La clave se convierte en CAPTCHA y es enviada al servidor de CAPTCHAS.
    - Se envía el mensaje de correo electrónico al destinatario.
2. Recepción
    - El receptor abre un correo electrónico cifrado con el presente esquema.
    - El cliente lo descarga del servidor por medio del protocolo POP3 o IMAP.
    - Se hace una petición al servidor de CAPTCHAS para recuperar los CAPTCHAS del correo.

- El usuario resuelve el CAPYCHA y se recalcula la clave de descifrado.
- Se descifra el mensaje y se le muestra al usuario.

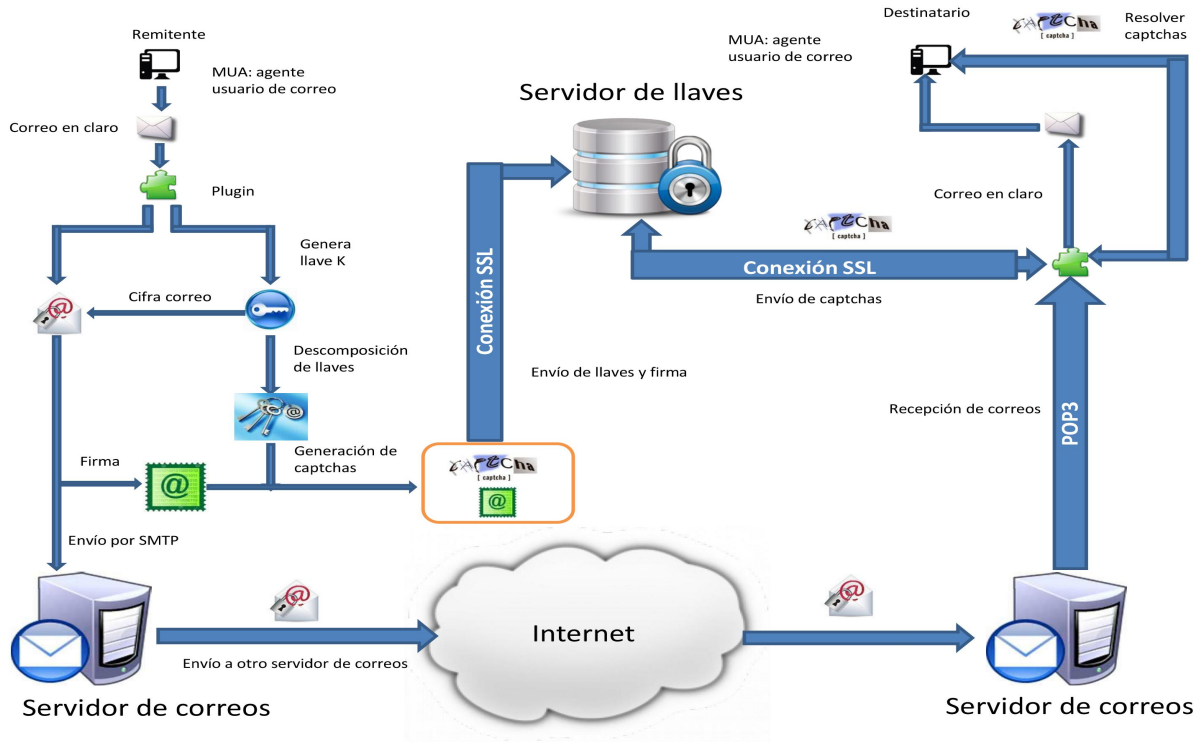

Figura 5.1: Diagrama General del sistema

5.1. Tecnologías

Como ya hemos visto en el esquema anterior necesitamos hacer uso de las herramientas adecuadas para poder desarrollar este esquema de cifrado. Las herramientas que analizaremos se describen en las siguientes secciones.

5.1.1. Cliente de correo electrónico

Un cliente de correo electrónico es necesario para el desarrollo de este proyecto ya que en él se instalará un complemento que cifre el mensaje, envie los CAPTCHAS y descifre los mensajes de correo electrónico. Para ello buscamos un cliente de correo electrónico que cuente con el soporte de los protocolos POP3, SMTP y IMAP; sus licencias son de código libre; soporte la instalación de APIs externas; y tenga soporte en los sistemas operativos ***Windows***, ***IOS*** y ***Linux***. Por lo tanto se investigaron los siguientes clientes de correo electrónico que se encuentra en el mercado:

Cliente de correo electrónico	Sistema Operativo	Protocolos soportados	Código Libre	Agregar funcionalidad	Extra	Gratuita o de paga
eM client	Windows 7, 8 & 10 ; IOS	POP3, SMTP, IMAP, EWS, AirSyn	NO	NO	100 % compatible con gmail y sus APIs	Ambos
Postbox	Windows, IOS	POP3, SMTP, IMAP	NO	SI por medio de APIs	Sincronización con Dropbox, OneDrive, Facebook y Twitter	Ambos
Zimbra	Windows, IOS & Linux	POP3, SMTP, IMAP	SI	SI por medio de APIs	Una plataforma de nivel empresarial y capas se soportar sincronización con múltiples servicios	Ambos
Opera Mail	Windows, IOS & Linux	POP3, SMTP, IMAP	SI	NO	La plataforma para desarrollar en Opera se actualiza cada semana	Gratuito
Thunderbird	Windows, IOS & Linux	POP3, SMTP, IMAP	SI	SI por medio de APIs	Cliente de correo versátil y fácilmente escalable y una comunicad de desarrollo bastante amplia	Gratuito

- El cliente de correo electrónico ***eM client*** tiene una sincronización a 100 % con las cuentas de ***Gmail*** y sus APIs, cuenta con una versión gratuita y una versión de paga; puede hacer migración de mensajes de correo electrónico y contactos de diversos clientes de correo electrónico y tiene una compatibilidad con muchos servidores de correo electrónico. [15]
  Su desventaja es que su código es cerrado y permite agregar funcionalidades.

- El cliente de correo electrónico ***Postbox*** esta soportada en los sistemas operativos ***Windows 7*** o posteriores y ***IOS***, esta aplicación es generada por el servidor de correo electrónico ***Postbox*** por lo tanto cuenta con una sincronización al 100 % con este servidor, también soporta otros servidores de correo como ***Gmail*** y ***Outlook***; este cliente puede sincronizarse con ***Dropbox***, ***Onedrive*** y redes sociales como ***Facebook***, ***Twitter***, entre otras. Es posible agregar más funcionalidades con la instalación de APIs.
  Una desventaja de esta aplicación es que su código es cerrado, pero gracias a que

esta basado en código de ***Mozilla*** puedes desarrollar APIs para agregarle tus propias funciones. [16]

- El cliente de correo electrónico ***zimbra*** es la aplicación más completa que se analizó, tiene compatibilidad con el servidor ***zimbra*** pero es capaz de soportar otros servidores de correo electrónico, se encuentra en los 3 sistemas para PC, ***Windows***, ***IOS*** & ***Linux***, da la facilidad de agregarle funcionalidades por medio de la instalación de APIs y gracias a que su código es abierto se pueden programar funciones propias. Este cliente cuenta con la versión gratuita y la versión de paga. Una gran ventaja que tiene es que puedes certificarte en el desarrollo APIs para este cliente de correo electrónico. [17] La única desventaja que encontramos en este cliente de correo es que la plataforma es demasiado grande y el tiempo que se necesita invertir al estudio del código es demasiado y el tiempo de desarrollo de este proyecto es muy corto.

- ***Opera mail*** es un cliente de correo electrónico que salió al mercado recientemente y se puede instalar en los sistemas operativos ***Windows***, ***IOS*** & ***Linux***, es capaz de comunicarse con diversos servidores de correo electrónico y su código es de libre acceso. Su principal desventaja es que las funcionalidades que se quieran agregar no pueden ser adquiridas por medio de la instalación de complementos o APIs. [18]

- Por último tenemos a ***Thunderbird*** que es un cliente de correo electrónico desarrollado por ***Mozilla***, este cliente es de código abierto y la instalación de APIs para agregar funcionalidad es demasiado sencilla; es un cliente de correo que puede ser instalado en los sistemas operativos ***Windows***, ***IOS*** y ***Linux***. [19]

Por lo tanto el cliente de correo electrónico que utilizaremos será ***Thunderbird*** al ser un cliente de correo electrónico casi tan completo como ***zimbra*** pero con la facilidad de desarrollar APIs más rápido.

5.1.2. Lenguajes de programación.

Uno de los objetivos que se tienen en este proyecto es generar un complemento para un cliente de correo electrónico por lo tanto al escoger a ***Thunderbird*** como cliente tenemos que programar con el lenguaje que fue desarrollado para tener la mayor compatibilidad.
Para el desarrollo de este proyecto utilizaremos [19]:

- Lenguaje Python
- HTML 5
- CSS3

A pesar de ser una aplicación de escritorio este cliente de correo electrónico está construido con lenguajes web.

5.1.3. Tipos de CAPTCHAS

En nuestro esquema de cifrado es necesario esconder la palabra que genera clave en un CAPTCHA para ser enviado a otro usuario y descifre el mensaje, pero existen varios tipos de CAPTCHAS que podemos utilizar [20].
Los CAPTCHAS se pueden clasificar de la siguiente forma:

- CAPTCHAS de texto: Este tipo de CAPTCHAS genera una pregunta sencilla donde la respuesta a la pregunta es la respuesta al CAPTCHA.
- CAPTCHAS de imágenes: Este tipo de CAPTCHAS nos muestran en una imagen una cadena de caracteres distorsionados, esta cadena de caracteres es la repuesta del CAPCHA transformada en una imagen.
- CAPTCHAS de audio: Este tipo de CAPTCHAS se caracterizan por ser un audio con ruidos de fondo, donde nos dirán la respuesta oculta.
- CAPTCHAS de video: Este tipo de CAPTCHAS nos muestran un video de unos cuantos segundos donde una palabra aparece alrededor de la pantalla, esta palabra es la respuesta al CAPTCHA.
- CAPTCHAS de acertijos: Este tipo de CPATCHAS es versatil, ya que se trata de pequeños acertijos que resolver donde la respuesta no es una palabra si no una acción o serie de acciones. Los CAPTCHAS de acertijos pueden ser armar un rompecabezas pequeño, seleccionar la imagen que no corresponde, unir figurar geométricas, etc.

Para poder decidir qué tipo de CAPTCHAS se utilizará se consideró los siguientes puntos:

- Facilidad de creación.
- Peso en byte del CAPTCHA.
- Forma del despliegue.
- Tipo de respuesta.

Por lo tanto nosotros necesitamos un tipo de CAPTCHA con poco peso, que su respuesta sea una cadena de caracteres y su forma de despliegue sea fácil de implementar.
Consideradas las necesidades anteriores las opciones son CAPTCHAS de imágenes y CAPTCHAS de texto, pero en este proyecto utilizaremos los CAPTCHAS de imágenes porque se almacenaras en los CAPTCHAS las claves de cifrado de los mensajes de correo electrónico y estas son cadenas de caracteres que no se les puede generar una pregunta coherente.

5.1.4. Bases de datos para almacenar los CAPTCHAS.

Para escoger un gestor de base de datos que controle la información delos usuarios registrados en nuestra aplicación, la información de los mensajes que envían entre usuarios y los CAPTCHAS asociados a los mensajes para ser descifrados consideramos 3 características principales en un gestor base de datos relacional:

- Rapidez en transferencias de información.
- Número de usuarios conectados que soporta.
- Facilidad de comunicación entre los lenguajes Python, HTML con los gestores de base de datos.

Los 3 gestores que se analizaron fueron SQLite, MySQL y PostGreSQL.
SQLite es un gestor de base de datos sumamente ligero que soporta hasta 2 terabytes de información, esta base de datos es compatible con python y lenguajes de programación web. Este gestor por su misma ligereza es posible ser implementada en dispositivos móviles, pero no es recomendable cuando múltiples usuarios están haciendo peticiones al mismo tiempo ya que su rendimiento baja [21].
MySQL es un gestor de base de datos que se caracteriza por su transferencia al hacer consultas de datos almacenados; es uno de los gestores libres más utilizados en aplicaciones web y cuenta con diferentes APIs para hacer la comunicación con los lenguajes Python, PHP, C++, etc. Y soporta peticiones de múltiples usuarios gracias a la implementación de hilos.
PostGreSQL es un gestor de base de datos que puede hacer operaciones complejas como subconsultas, transacciones y rollbacks's. Soporta las peticiones de múltiples usuarios pero en cuanto a la velocidad de transferencia de información, comparado con MySQL, es muy lenta pero lo compensa manteniendo una velocidad casi invariable a pesar de que la base se mantenga con un número grande de registros. [22]
Se eligio el gestor de base de datos MySQL porque el proyecto necesita mayor rapidez en la transferencia de información más que generar filtros muy complejos para la búsqueda de información.

Capítulo 6

Análisis y Diseño

6.1. Diagramas de caso de uso

Para el desarrollo de esta propuesta se muestran los siguientes diagramas, estos muestran el diseño que nos da una idea mas clara de como quedara el sistema.

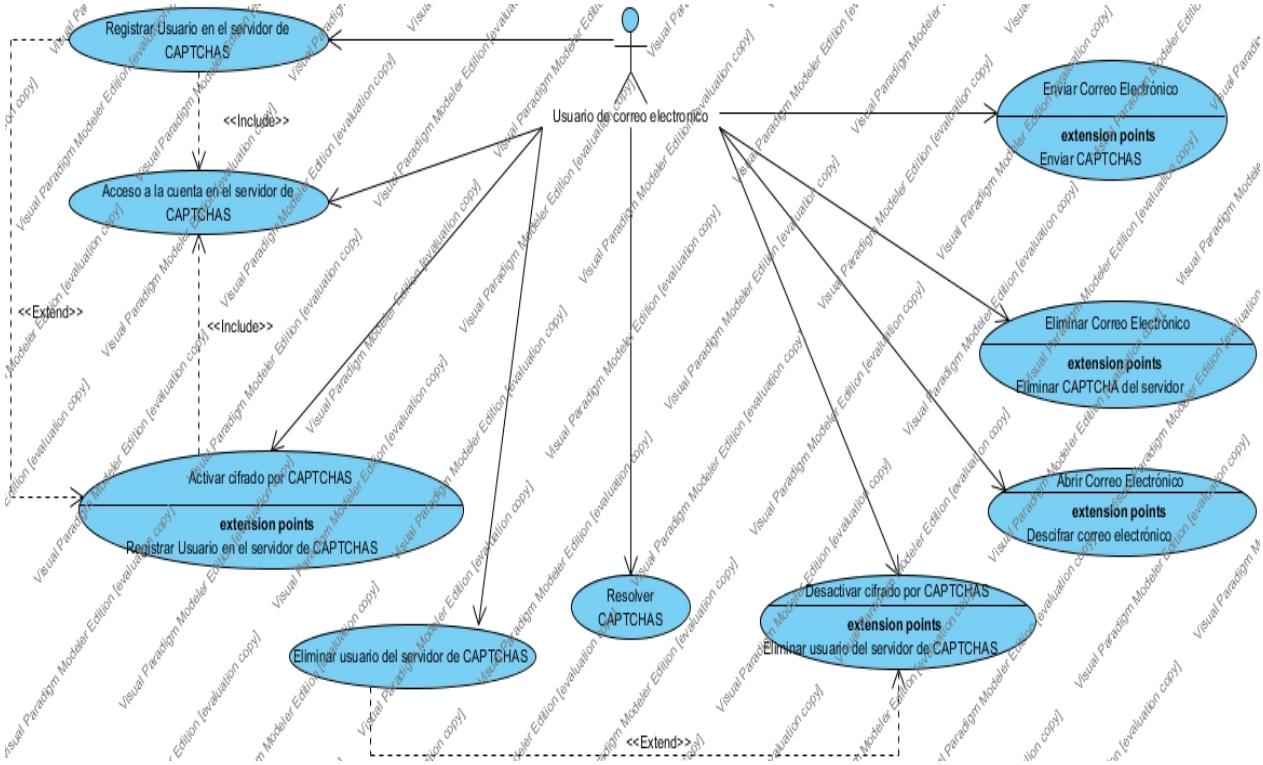

Figura 6.1: Diagrama General de caso de uso

6.1.1. Diagrama de caos de uso CU2 Registrar usuario en el servidor de CAPTCHAS

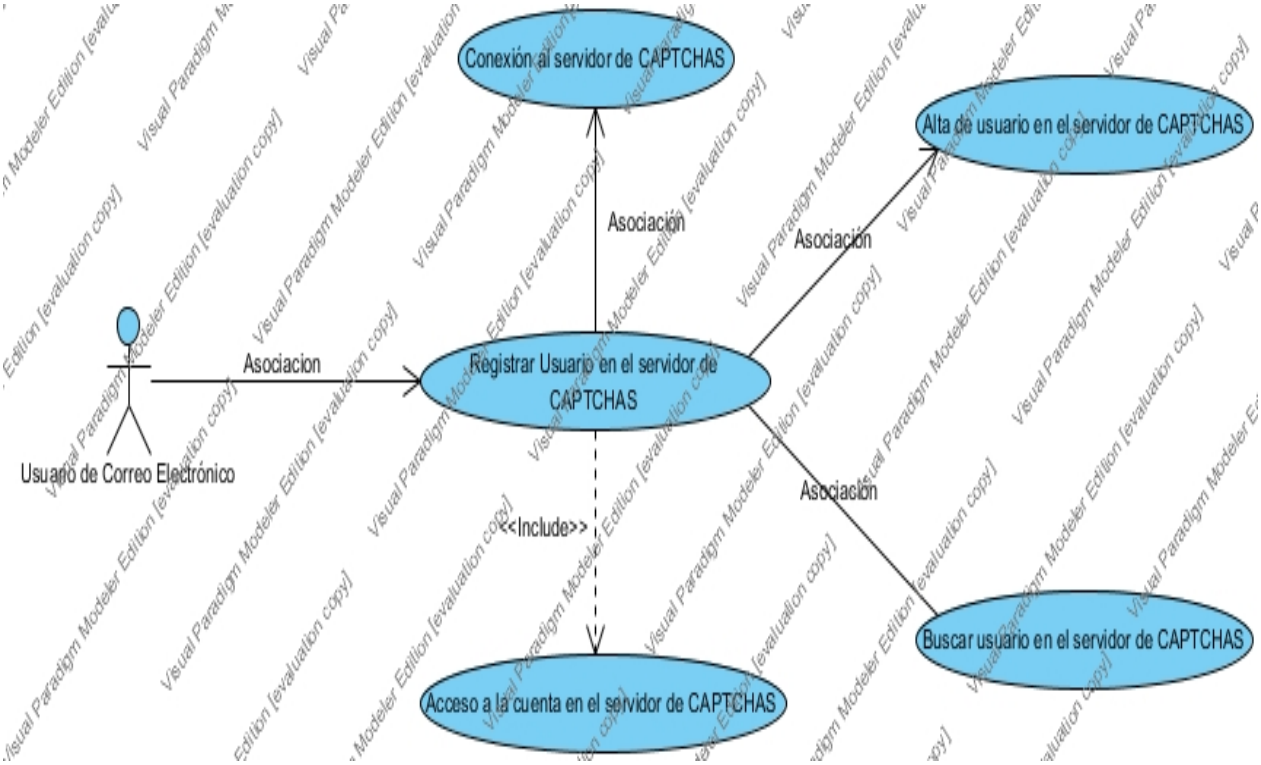

Figura 6.2: Diagrama de caos de uso CU2 Registrar usuario en el servidor de CAPTCHAS

Caso de Uso	CU2 Registrar Usuario en el servidor de CAPTCHAS		
Actor	Actor1. Usuario de Correo Electrónico		
Descripción	Describe los pasos necesarios para registrar un nuevo usuario en el servidor de CAPTCHAS.		
Pre-condiciones	Tener una cuenta de correo electrónico.		
Post-condiciones	Activación del módulo de cifrado por CAPTCHAS.		
Puntos de inclusión	Acceso a la cuenta en el servidor de CAPTCHAS.		
Puntos de extensión			
Flujo principal		Actor/Sistema	Acción a realizar
	1	Actor	El usuario selecciona la opción registrarse en el servidor de CAPTCHAS
	2	Sistema	El cliente de correo contesta un formulario con la información necesaria para dar de alta en el servidor de CAPTCHAS.
	3	Actor	Completa el formulario y oprime el botón de registrar.

	4	Sistema	El sistema valida los datos proporcionados por el usuario.
	5	Sistema	Se conecta con el servidor y valida si el usuario ya está registrado. <FA01 - Usuario ya registrado><FA02 - Falla en la conexión con el servidor>
	6	Sistema	Manda la información del usuario y lo da de alta.
	7	Sistema	Despliega el siguiente mensaje .El usuario se ha dado de alta correctamente"
			Fin del flujo principal.
	FA01 - Usuario ya registrado.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Sistema	Despliega el siguiente mensaje .El usuario ya está registrado favor de proporcionar otra cuenta de correo electrónico"
	2		El flujo continúa en el paso 3 del flujo principal.
			Fin del flujo alternativo
	FA02 - Falla en la conexión con el servidor.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Sistema	Despliega el siguiente mensaje "La conexión con la red es nula o limitada, favor de realizar esta operación más tarde"
	2		El flujo continúa en el paso 1 del flujo principal.
			Fin del flujo alternativo

Tabla 6.1: Descripción CU2.

6.1.2. Diagrama de casos de uso CU3 Acceso a la cuenta en el servidor de CAPTCHAS

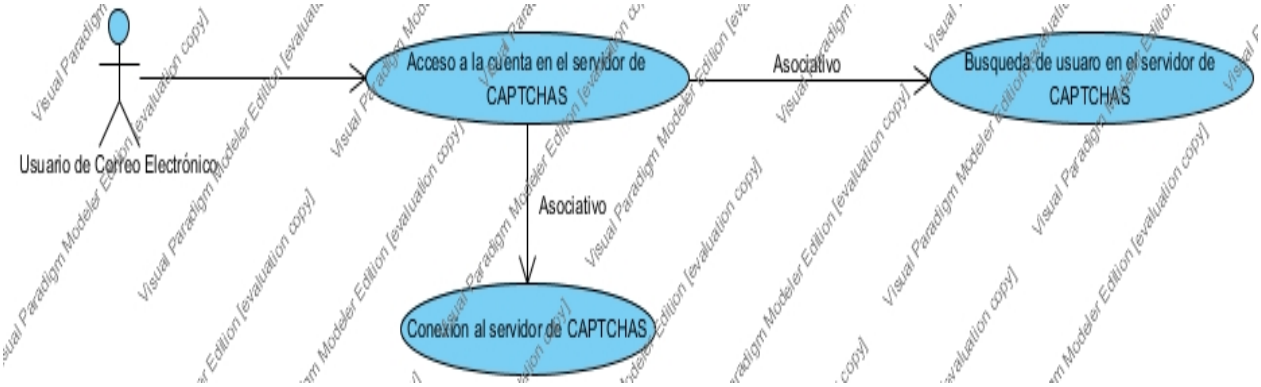

Figura 6.3: Diagrama de casos de uso CU3 Acceso a la cuenta en el servidor de CAPTCHAS

6.1.3. Diagrama de casos de uso CU4 Abrir Correo Electrónico.

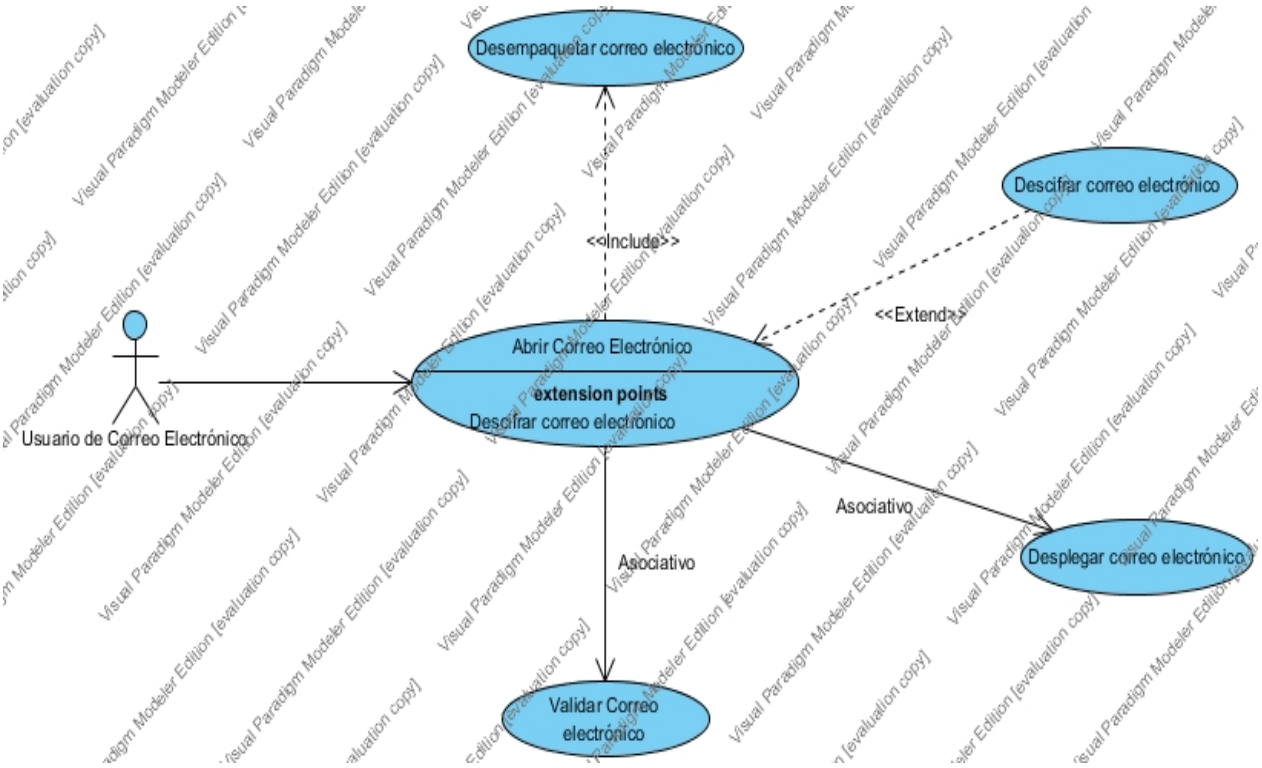

Figura 6.4: Diagrama de casos de uso CU4 Abrir Correo Electrónico.

Caso de Uso	CU4 Abrir correo electrónico		
Actor	Actor 1. Usuario de correo electrónico		
Descripción	Describe los pasos necesarios para abrir un mensaje de correo electrónico.		
Pre-condiciones	1. Iniciar sesión con su servidor de correo electrónico. 2. Descargar el correo electrónico que se desea abrir.		
Post-condiciones	Despliegue del mensaje de correo electrónico descifrado.		
Puntos de inclusión	Desempaquetar correo electrónico		
Puntos de extensión	Descifrar correo electrónico		
Flujo principal		Actor/Sistema	Acción a realizar
	1	Actor	El caso de uso comienza cuando el usuario selecciona el correo que desea abrir.
	2	Sistema	El sistema manda a llamar a la función de desempaquetar correo electrónico.
	3	Sistema	Valida si el mensaje viene timbrado. <FA01 - El mensaje no viene timbrado>
	4	Sistema	Invoca al caso de uso <CU Descifrar correo electrónico>
	5	Sistema	Recibe el mensaje de correo electrónico descifrado
	6	Sistema	Despliega el contenido completo del mensaje al usuario
			Fin del flujo principal.
	FA01 - El mensaje no viene timbrado.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Sistema	El flujo continúa en el paso 6 del flujo principal.
			Fin del flujo alternativo

Tabla 6.2: Descripción CU4.

6.1.4. Diagrama casos de uso CU5 Activar cifrado por CAPTCHAS.

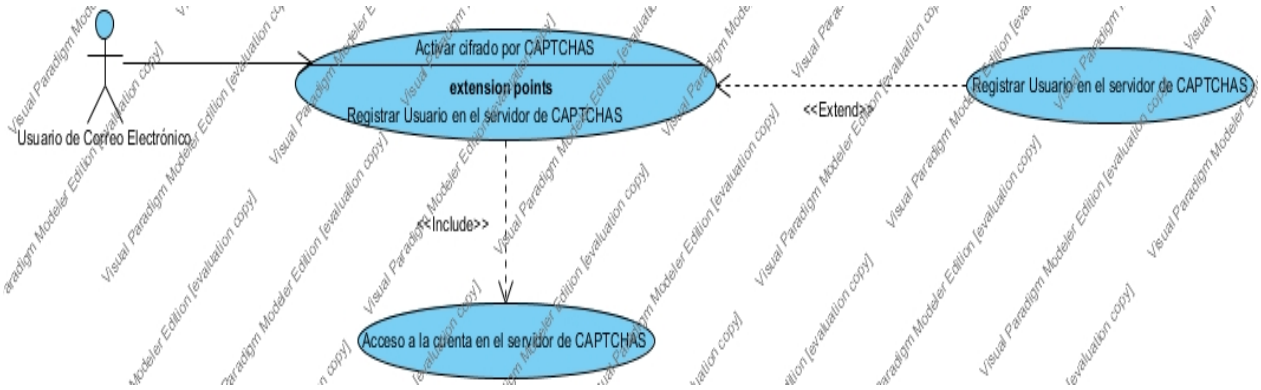

Figura 6.5: Diagrama casos de uso CU5 Activar cifrado por CAPTCHAS.

Caso de Uso	CU5 Activar cifrado por CAPTCHAS		
Actor	Actor 1. Usuario de correo electrónico		
Descripción	Describe los pasos necesarios para activar el módulo de cifrado CAPTCHAS en el cliente de correo electrónico.		
Pre-condiciones	1. Instalar el módulo de cifrado por CAPTCHAS		
Post-condiciones	Activación del cifrado y descifrado por CAPTCHAS.		
Puntos de inclusión			
Puntos de extensión	Registrar usuario del servidor de CAPTCHAS		
Flujo principal		Actor/Sistema	Acción a realizar
	1	Actor	El caso de uso inicia cuando el actor seleccionar la opción .Activar cifrado por CAPTCHAS"
	2	Sistema	El sistema verifica si la dirección de correo del usuario está registrada en el servidor de CAPTCHAS<FA01 -Usuario no registrado en el servidor>
	3	Sistema	Despliega una ventana con el mensaje .Activación del módulo de cifrado por CAPTCHAS"
			Fin del flujo principal.
	FA01 -Usuario no registrado en el servidor.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Sistema	El sistema despliega una ventana con las opciones de Registrarseż Çancelar". <FA02 Cancelar activación>

	2	Actor	Oprime el botón de Registrarse"
	3	Sistema	El sistema invoca al caso de uso <CU Registrar usuario en el servidor de CAPTCHAS>
	4	Sistema	El sistema obtiene una respuesta satisfactoria del registro
	5		El flujo continúa en el paso 2 del flujo principal.
			Fin del flujo alternativo
	FA02 - Cancelar activación.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Actor	El Actor selecciona Çancelar"
	2	Sistema	Cierra la ventana de selección
	3		El flujo continúa en el paso 1 del flujo principal.
			Fin del flujo alternativo

Tabla 6.3: Descripción CU5.

6.1.5. Diagrama de casos de uso CU6 Descifrar correo electrónico.

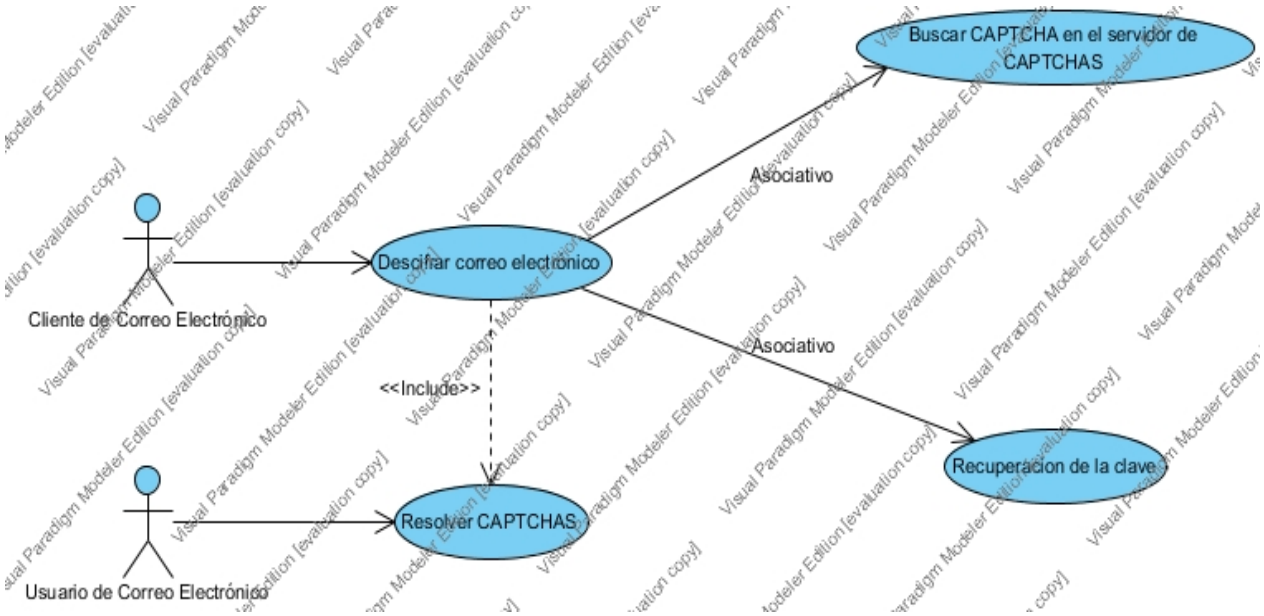

Figura 6.6: Diagrama de casos de uso CU6 Descifrar correo electrónico.

Caso de Uso	CU6 Descifrar correo electrónico.		
Actor	Actor 1. Usuario de correo electrónico		
Descripción	Describe los pasos necesarios para desactivar el módulo de cifrado CAPTCHAS en el cliente de correo electrónico		
Pre-condiciones	1. Activar cifrado por CAPTCHAS. 2. Registrar usuario en el servidor de CAPTCHAS		
Post-condiciones	Desactivación del cifrado y descifrado por CAPTCHAS.		
Puntos de inclusión			
Puntos de extensión	Eliminar usuario del servidor de CAPTCHAS		
Flujo principal		Actor/Sistema	Acción a realizar
	1	Actor	El caso de uso inicia cuando el actor seleccionar la opción "Desactivar cifrado por CAPTCHAS"
	2	Sistema	El sistema despliega la venta con las opciones de "Desactivar cifrado¿ .Eliminar usuario«FA01 - Eliminar usuario>
	3	Actor	Selecciona la Desactivación del cifrado por CAPTCHAS
	4	Sistema	El sistema desactiva el módulo de cifrado por CAPTCHA

			Fin del flujo principal.
	FA01 - Eliminar usuario.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Actor	El Actor selecciona .Eliminar usuario"
	2	Sistema	El sistema despliega una ventana con las opciones de .Aceptarz Çancelar"para confirmar la eliminación del usuario. <FA02 - Cancelar acción eliminar usuario>
	3	Actor	Oprime el botón de .Aceptar"
	4	Sistema	Establece la conexión con el servidor de CAPTCHAS <FA03 - Fallo en la conexión con el servidor>
	5	Sistema	Busca y elimina al usuario de la base de datos desplegando la confirmación del servidor.
	6	Actor	Oprime el botón de .Aceptar"
	7	Sistema	Desactiva el módulo de cifrado por CAPTCHA
			Fin del flujo alternativo
	FA02 - Cancelar acción eliminar usuario.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Actor	El Actor selecciona Çancelar"
	2	Sistema	Cierra la ventana de confirmación
			Fin del flujo alternativo
	FA03 - Fallo en la conexión con el servidor.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Sistema	Despliega una ventana de alerta con el mensaje "No se ha podido establecer la conexión con el servidor, es probable que no se tenga conexión a internet. Favor de intentarlo más tarde"
	2	Actor	Cierra la ventana de alerta
	3		El flujo continúa en el paso 1 del flujo principal

			Fin del flujo alternativo

Tabla 6.4: Descripción CU6.

6.1.6. Diagrama de caos de uso CU7 Eliminar CAPTCHA del servidor.

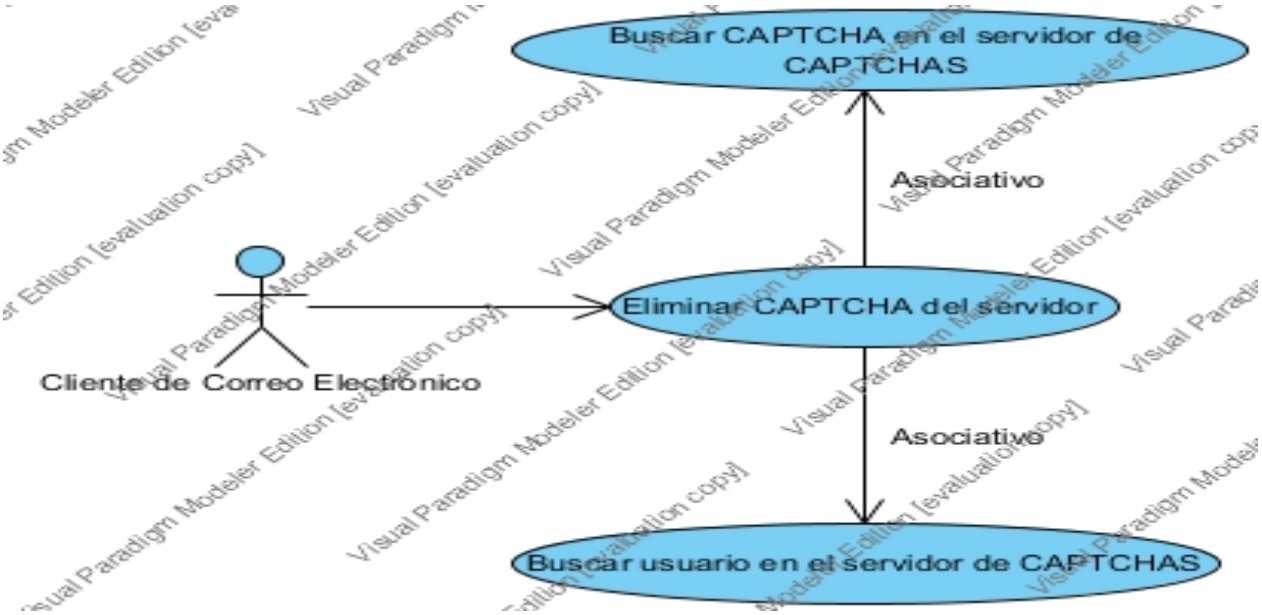

Figura 6.7: Diagrama de caos de uso CU7 Eliminar CAPTCHA del servidor.

Caso de Uso	CU7 Eliminar CAPTCHA del servidor.		
Actor	Actor 1. Cliente de correo electrónico.		
Descripción	Describe los pasos necesarios para eliminar los CAPTCHAS del servidor de CAPTCHAS.		
Pre-condiciones	1. Solicitar eliminar un mensaje de correo electrónico		
Post-condiciones			
Puntos de inclusión			
Puntos de extensión			
Flujo principal		Actor/Sistema	Acción a realizar
	1	Actor	Solicita eliminar CAPTCHA del servidor de CAPTCHAS
	2	Sistema	Busca al usuario y el CAPTCHA a eliminar en el servidor de CAPTCHAS
	3	Sistema	Elimina el CAPTCHA solicitado
	4	Sistema	Regresa la confirmación de que se eliminó el CAPTCHA.
			Fin del flujo principal.

Tabla 6.5: Descripción CU7.

6.1.7. Diagrama de casos de uso CU8 Eliminar correo electrónico.

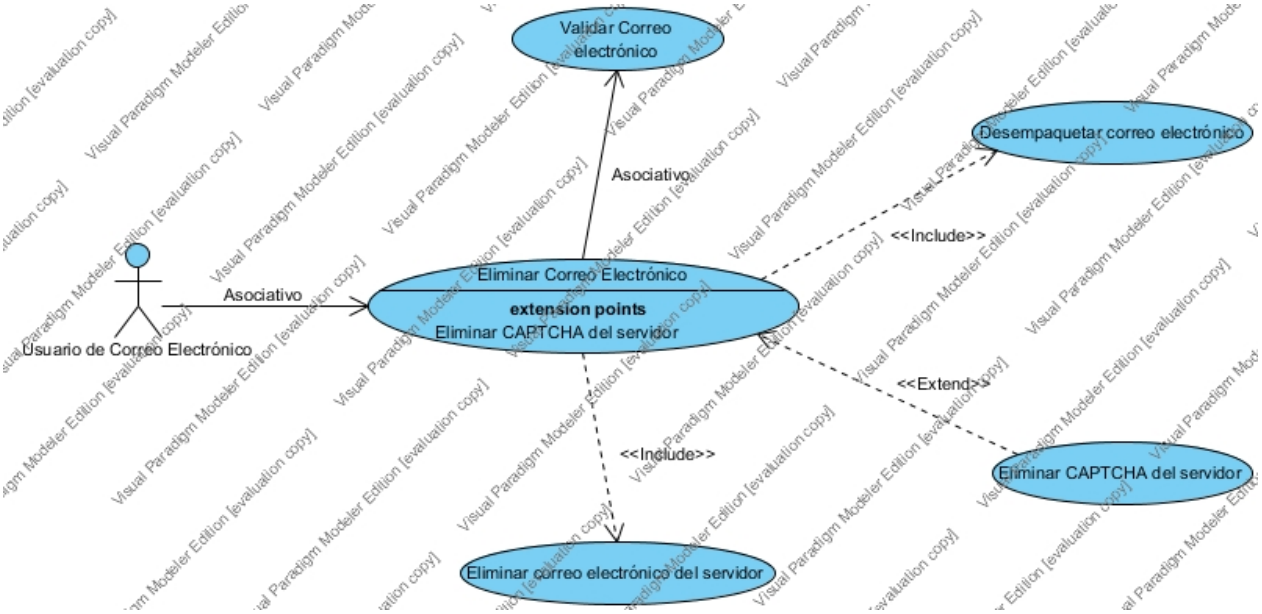

Figura 6.8: Diagrama de casos de uso CU8 Eliminar correo electrónico.

Caso de Uso	CU8 Eliminar correo electrónico.		
Actor	Actor 1. Usuario de correo electrónico.		
Descripción	Describe los pasos necesarios para eliminar un mensaje de correo electrónico.		
Pre-condiciones	1. Seleccionar un mensaje de correo electrónico		
Post-condiciones	Mensaje y CAPTCHA eliminados.		
Puntos de inclusión	1. Desempaquetar correo electrónico. 2. Eliminar correo electrónico del servidor		
Puntos de extensión	Eliminar CAPTCHA del servidor de CAPTCHAS		
Flujo principal		Actor/Sistema	Acción a realizar
	1	Actor	Selecciona un mensaje de correo electrónico a eliminar.
	2	Sistema	Desempaqueta el mensaje de correo electrónico
	3	Sistema	Valida si el mensaje esta timbrado. <FA01 - Mensaje no timbrado>
	4	Sistema	Invoca al caso de uso <CU Eliminar CAPTCHA del servidor>
	5	Sistema	Elimina el mensaje de correo electrónico y despliega el mensaje .El correo se ha eliminado correctamente"
			Fin del flujo principal.
	FA01 - Mensaje no timbrado.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Sistema	El sistema continúa a partir del paso 5 del flujo principal.
			Fin del flujo alternativo

Tabla 6.6: Descripción CU8.

6.1.8. Diagrama de casos de uso CU9 Enviar CAPTCHAS

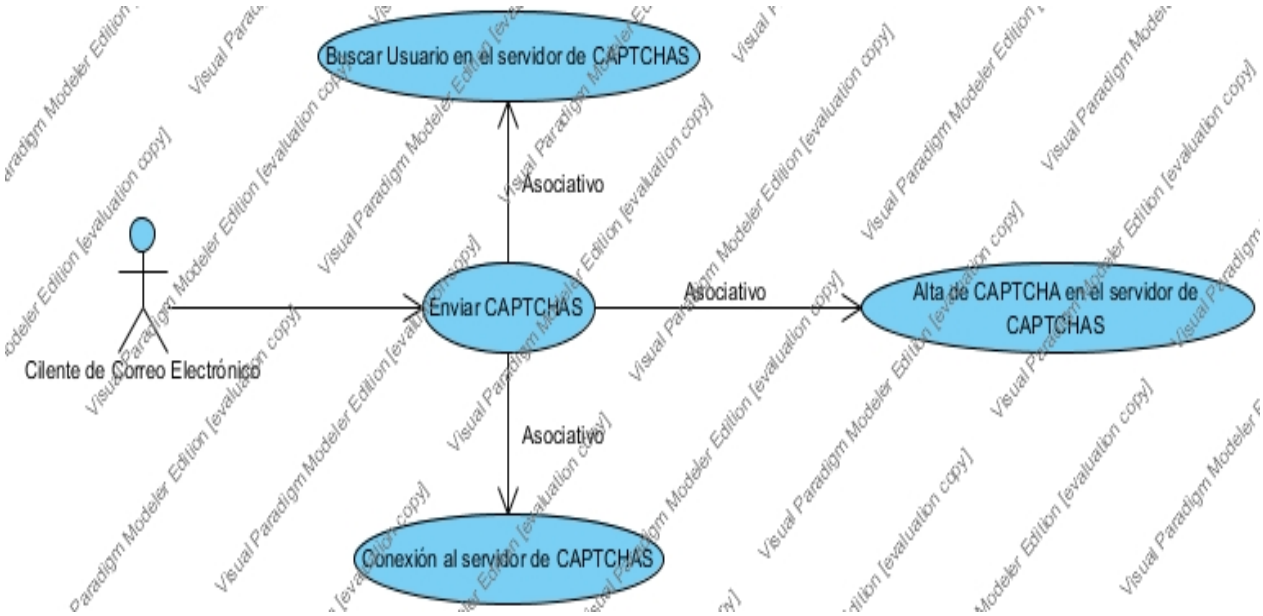

Figura 6.9: Diagrama de casos de uso CU9 Enviar CAPTCHAS

Caso de Uso	CU9 Enviar CAPTCHAS		
Actor	Actor 1. Cliente de correo electrónico.		
Descripción	Describe los pasos necesarios para enviar el CAPTCHA el servidor de CAPTCHAS		
Pre-condiciones	1. Solicitar el envió de un nuevo mensaje de correo electrónico.		
Post-condiciones	Envío del CAPTCHA al servidor de CAPTCHAS		
Puntos de inclusión			
Puntos de extensión			
Flujo principal		Actor/Sistema	Acción a realizar
	1	Actor	Solicita el envío de CAPTCHA al servidor
	2	Sistema	Abre la conexión y busca al usuario en el servidor de CAPTCHAS
	3	Sistema	Da de alta el CAPTCHA en el servidor asociándolo con el usuario.
	4	Sistema	Regresa la confirmación de que se dio de alta el CAPTCHA
			Fin del flujo principal.

Tabla 6.7: Descripción CU9.

6.1.9. Diagrama de casos de uso CU10 Enviar correo electrónico.

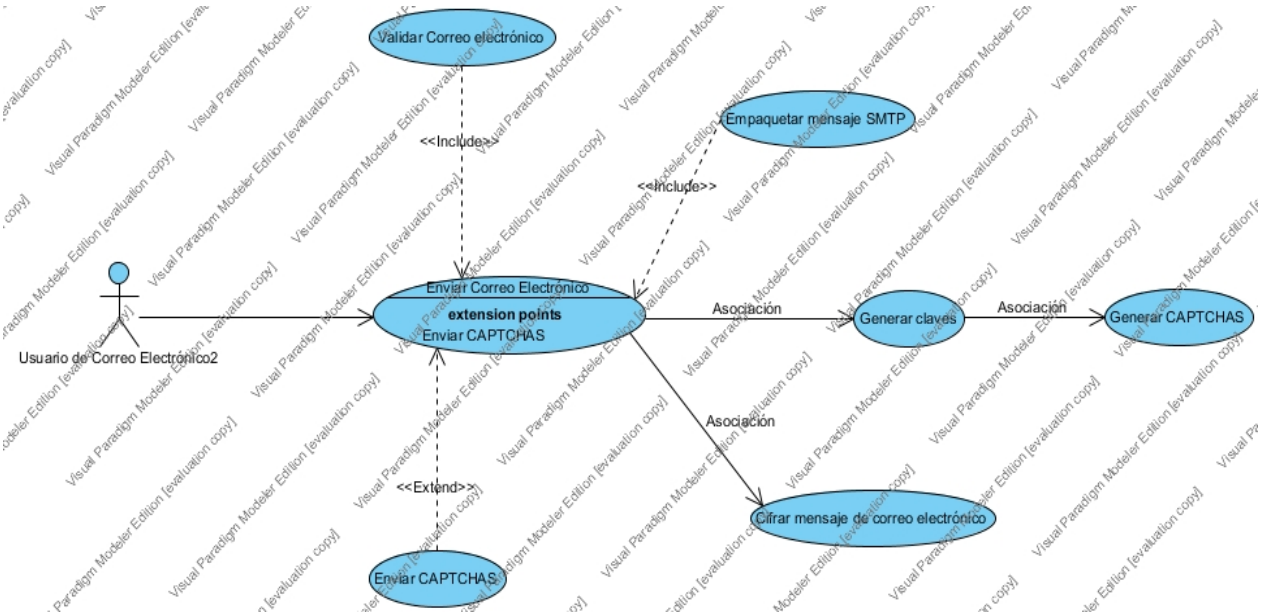

Figura 6.10: Diagrama de casos de uso CU10 Enviar correo electrónico.

Caso de Uso	CU10 Enviar correo electrónico.		
Actor	Actor 1. Usuario de correo electrónico.		
Descripción	Describe los pasos necesarios para enviar un mensaje de correo electrónico cifrado a otro usuario de correo electrónico.		
Pre-condiciones	1. El usuario tiene que redactar un mensaje de correo electrónico que contenga la dirección del destinatario.		
Post-condiciones	Envió de un mensaje cifrado al servidor de correo electrónico y el registro del CAPTCHA en el servidor de CAPTCHAS.		
Puntos de inclusión	1. Validar correo electrónico. 2. Empaquetar mensaje de correo electrónico SMTP.		
Puntos de extensión	Enviar CAPTCHA		
Flujo principal		Actor/Sistema	Acción a realizar
	1	Actor	Oprime el botón .Enviar"
	2	Sistema	Valida que el mensaje de correo electrónico contenga los datos mínimos.<FA01 - Campos no completados>
	3	Sistema	Genera una llave de cifrado
	4	Sistema	Con una palabra aleatoria se genera el CAPTCHA y cifra el mensaje de correo electrónico.
	5	Sistema	Toma el mensaje cifrado y es empaquetado para enviarse al servidor de correo electrónico
	6	Sistema	Toma el CAPTCHA y se envía al caso de uso <CU Enviar CAPTCHA>
	7	Sistema	Despliega el mensaje de .envío satisfactorio"
			Fin del flujo principal.
	FA01 - Campos no completados.		
Flujo alternativo		Actor/Sistema	Acción a realizar
	1	Sistema	Notifica al usuario cuales campos han sido mal proporcionados, para poder enviar el mensaje correctamente
	2	Actor	Modifica los campos solicitados
	3		El flujo continúa en el paso 1 del flujo principal
			Fin del flujo alternativo

Tabla 6.8: Descripción CU10.

6.2. Diagramas a bloques

A continuación se presentan los diagramas a bloques, en donde se muestra cuál es la secuencia de procesos a realizar. Esto servirá para comprender cómo se comunican los diferentes módulos de manera interna, y cómo hacen los procesos.

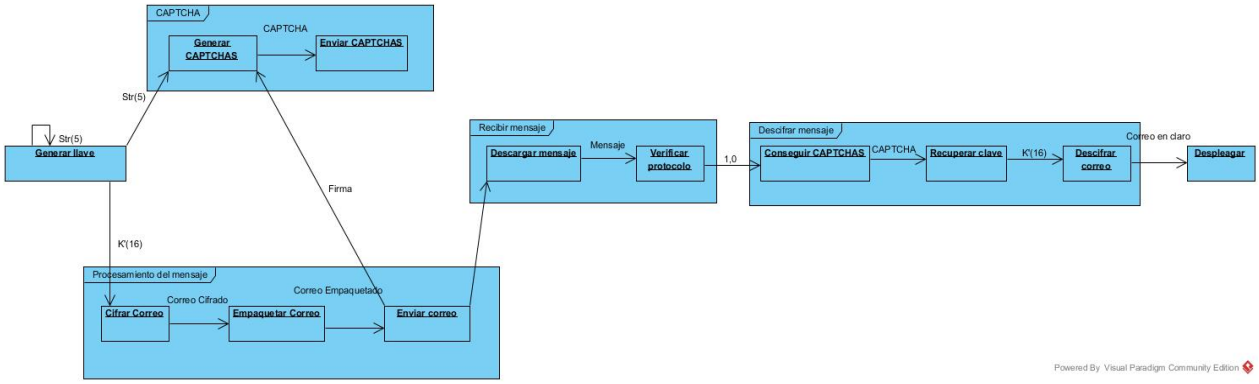

Figura 6.11: Diagrama a bloque 0 general del sistema

	Generar clave	**Generar CAPTCHA**	**Procesamiento del mensaje**	**Recibir mensaje**	**Descifrar mensaje**	**Desplegar**
Entradas	*Señal de activación	*Cadena de 5 caracteres: Str(5)	*Clave de 16 bytes: K'(16). *Mensaje de correo.	*Correo Cifrado	Verificación (1,0)	*Correo en claro
Salidas	*Cadena de 5 caracteres: Str(5) *Clave de 16 bytes: K'(16)	*Señal de envió	*Correo Cifrado	*Verificación (1,0)	*Correo en claro	
Descripción	Se activa el proceso generar clave, este crea una palabra de 5 caracteres (Str(5)), procesa la palabra Str(5) por medio de una función hash obteniendo una palabra de 256 caracteres (K(256)) y recorta esta clave a una palabra de 16 caracteres (K'(16)).	Toma la entrada Str(5) y la convierte en una imagen CAPTCHA, Posteriormente inicia una conexión con el servidor de CAPTCHAS para mandarlo a este.	Cifra el mensaje de correo con la clave K'(16), posteriormente lo firma y genera un timbre para saber que fue creado con este esquema y lo empaqueta para su envió.	El cliente hace una petición al servidor y descarga el mensaje de correo electrónico, lo des empaqueta verifica la firma y el timbrado para saber de quién viene y si está cifrado bajo este esquema.	Se hace una petición al servidor de CAPTCHAS, se descargan los CAPTCHAS asociados al correo, ya con el CAPTCHA este se resuelve y se recupera la cadena Str(5), esta se pasa por una función hash y se recupera K(256), esta se corta a K'(16), con esto se descifra el mensaje.	Se muestra el correo descifrado en la interfase del cliente de correo electrónico.

Tabla 6.9: Diagrama a bloques 0 general

Figura 6.12: Diagrama a bloques 1 Generar clave

	Generar Str(5)	**Aplicar Función Hash**	**Recortar Hash K'**
Entradas	*Llamada a Función	*Cadena de 5 caracteres: Str(5)	*Digesto K(128)
Salidas	*Cadena de 5 caracteres: Str(5)	*Digesto K(128)	*K'(16)
Descripción	Toma una función random módulo 67, para formar una palabra con 5 caracteres aleatorios tomados del siguiente conjunto.Anillo67- .,+*[a-z][A-Z]	Se pasa la cadena Str(5) por una función hash SHA-1 para obtener un digesto único de esta palabra.	Se copian a otro string lo primeros 16 caracteres del digesto K(128) para formar la clave K'(16)

Tabla 6.10: Diagrama a bloques 1 general clave

	Cifrar
Entradas	*Clave K'(16) *Mensaje de correo
Salidas	*Correo cifrado
Descripción	Se cifra el mensaje con un algoritmo de llave simétrica (AES o DES) usando una llave de 16bytes o 128bits.

Tabla 6.11: Diagrama a bloques 2 Cifrar Correo

Figura 6.13: Diagrama a bloques 3 Empaquetar Correo

	Empaquetamiento SMTP	**Timbrar Correo**
Entradas	*Mensaje Cifrado	*Correo Empaqueta
Salidas	*Correo Empaquetado	*Correo Timbrado
Descripción	Se toma el correo y se integra en el formato del correo marcado en el RFC822	Se timbra el mensaje colocando una marca después del final del mensaje. Para señalar que el correo enviado está cifrado bajo este protocolo.

Tabla 6.12: Diagrama a bloques 3 Empaquetar Correo

Figura 6.14: Diagrama a bloques 4 Enviar correo

	Abrir conexión SMTP	**Envió de Correo por SMTP**
Entradas	*Petición	*Correo empaquetado
Salidas	*Canal de comunicación	*Confirmación de envió
Descripción	Se genera una petición para conexión SMTP	Se manda el correo electrónico al servidor por medio del protocolo SMTP

Tabla 6.13: Diagrama a bloques 4 Enviar correo

Figura 6.15: Diagrama a bloques 5 Generar CAPTCHA

	Generar respuesta del CAPTCHA	**Firmar CAPTCHA**	**Transformar palabra**
Entradas	*Cadena de caracteres: Str(5)	*Firma	*Señal de confirmación
Salidas	*Señal de confirmación	*Archivo Firmado	*Imagen CAPTCHA
Descripción	Genera un archivo con la respuesta del CAPTCHA	Se firma el CAPTCHA por medio de un Hashing del mensaje.	Convierte el Str(5) en una imagen distorsionada que llamaremos CAPTCHA

Tabla 6.14: Diagrama a bloques 5 Generar CAPTCHA

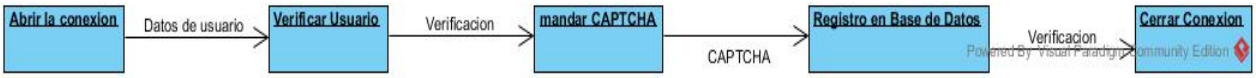

Figura 6.16: Diagrama a bloques 6 Enviar CAPTCHAS (Usuario existente)

	Abrir conexión	**Verificar Usuario**	**Mandar CAPTCHA**	**Registrar en base de datos**	**Cerrar Conexión**
Entradas	*CAPTCHAS	*Datos de Usuario	*Verificación de usuario	*Datos de usuario *CAPTCHA	Verificación
Salidas	*Datos de usuario	*Verificación de usuario	*CAPTCHA	Verificación	
Descripción	Se genera una petición para poder entablar una conexión con el servidor de CAPTCHAS	Se verifica la existencia del usuario en el servidor, si existe se le da acceso	Ya verificado el usuario se manda el CAPTCHA al servidor	Se registran los datos del CAPTCHA en la base de datos y se envía una verificación	Se cierra la conexión y se guardan los datos

Tabla 6.15: Diagrama a bloques 6 Enviar CAPTCHAS (Usuario existente)

Figura 6.17: Diagrama a bloques 7 Enviar CAPTCHAS (Usuario inexistente)

	Abrir conexión	**Registrar usuario en Base de Datos**	**Verificar Usuario**	**Mandar CAPTCHA**	**Registrar en base de datos**	**Cerrar Conexión**
Entradas	*CAPTCHAS	*Datos de Usuario	*Verificación de registro	*Verificación de usuario	*Datos de usuario *CAPTCHA	Verificación
Salidas	*Datos de usuario	*Verificación de registro	*Verificación de usuario	*CAPTCHA	Verificación	
Descripción	Se genera una petición para poder entablar una conexión con el servidor de CAPTCHAS	Se da de alta al usuario en la base de datos	se le da acceso al usuario	Ya verificado el usuario se manda el CAPTCHA al servidor	Se registran los datos del CAPTCHA en la base de datos y se envía una verificación	Se cierra la conexión y se guardan los datos

Tabla 6.16: Diagrama a bloques 7 Enviar CAPTCHAS (Usuario inexistente)

Figura 6.18: Diagrama a bloques 8 Descargar mensaje

	Abrir conexión al servidor de correo	**Descargar mensaje por IMAP o POP3**
Entradas	*Señal de activación	*Confirmación
Salidas	*Confirmación	*Correo electrónico
Descripción	El receptor se conecta al servidor de correo electrónico e inicia la sesión	Descarga del servidor de correo electrónico todos los mensajes que aún no se hayan descargado.

Tabla 6.17: Diagrama a bloques 8 Descargar mensaje

Figura 6.19: Diagrama a bloques 9 Verificar protocolo (con protocolo valido)

	Abrir mensaje	**Verificar mensaje**
Entradas	*Correo electrónico	*mensaje
Salidas	*Mensaje	*verificación
Descripción	Se toma el mensaje descargado del servidor y se des empaqueta para dejar solo el texto del mensaje	Se verifica que el mensaje tenga la bandera correspondiente a que está cifrado con este esquema

Tabla 6.18: Diagrama a bloques 9 Verificar protocolo (con protocolo válido)

	Abrir mensaje	Verificar mensaje
Entradas	*Correo electrónico	*mensaje
Salidas	*Mensaje	*verificación
Descripción	Se toma el mensaje descargado del servidor y se des empaqueta para dejar solo el texto del mensaje	Si la verificación es negativa se manda directamente al bloque de Despliegue

Tabla 6.19: Diagrama a bloques 10 Verificar protocolo (con protocolo inválido)

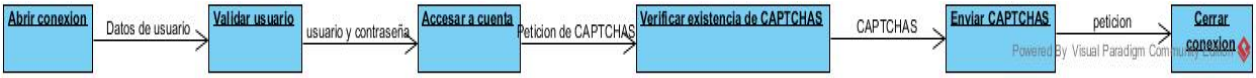

Figura 6.20: Diagrama a bloques 11 Conseguir CAPTCHAS (Usuario existente)

	Abrir Conexión	Validar Usuario	Accesar a cuenta	Verificar existencia de CAPTCHA	Enviar CAPTCHA
Entradas	*Confirmación	*Datos usuario	*Contraseña	*Petición de CAPTCHAS	*Confirmación
Salidas	*verificación	*Contraseña	*confirmación	*confirmación	*CAPTCHA
Descripción	Se abre una conexión con el servidor de CAPTCHAS	Se verifica que el usuario este dado de alta en el servidor mandándole una petición a la base de datos, si el usuario existe se accesa	Si esta dado de alta en el servidor se manda la contraseña para que pueda tener acceso a los CAPTCHAS de su cuenta	Se verifica que los CAPTCHAS que están ligados al mensaje que realizo la petición existan	Si existen estos CAPTCHAS son enviados de regreso al mensaje

Tabla 6.20: Diagrama a bloques 11 Conseguir CAPTCHAS (Usuario existente)

Figura 6.21: Diagrama a bloques 12 Recuperar clave

	Resolver CAPTCHA	**Aplicar Función Hash**	**Recortar llave**
Entradas	*CAPTCHA	*Cadena de 5 caracteres: Str(5)	*Digesto K(128)
Salidas	*Str(5)	*Digesto K(128)	*K'(16)
Descripción	Se despliega el CAPTCHA para que el usuario pueda resolverlo	Se pasa la cadena Str(5) por una función hash SHA-1 para obtener un digesto único de esta palabra	Se copian a otro string lo primeros 16 caracteres del digesto K(128) para formar la clave K'(16)

Tabla 6.21: Diagrama a bloques 12 Recuperar clave

	Descifrar
Entradas	*Clave K'(16) *Mensaje de correo
Salidas	*Correo descifrado
Descripción	Se descifra el mensaje con un algoritmo de llave simétrica (AES o DES) usando una llave de 16bytes o 128bits.

Tabla 6.22: Diagrama a bloques 13 Descifrar correo

Capítulo 7

Desarrollo de prototipos

Objetivo del prototipo: Conocer el uso, funcionamiento e implementación de herramientas de cifrado, hashing y generación de CAPTCHAS, con el fin de conocer la integración de estos módulos en diferentes lenguajes de programación.

7.1. Prototipo 1

Se implementó un módulo de cifrado de mensajes de texto en lenguaje C++. Tratando de simular el proceso de cifrado del esquema que se esta usando.
La primera parte del proceso es abrir el mensaje para lo cual se estan usando los metodos estandar definidos en las bibliotecas nativas de C++, posteriormente se generara una palabra aleatoria de 5 caracteres usando una función Rand() %100 y transformando el valor de salida a un char.
Al resultado se pasa por una función de hashing, esta función no es nativa de ninguna biblioteca estándar de C++ ni de C, por lo que se tuvo que conseguir una en internet y probar que efectivamente funcionara como lo necesitamos.
Posteriormente este hash se usara como llave para cifrar el mensaje que ya abrimos, para esto necesitaremos una función AES o DES, ninguna de estas es estándar de alguna biblioteca de C o C++, así que tendremos que buscarlas y verificar su funcionamiento.

Conclusión: Podemos ver que en C++ el proceso es simple pero se necesita buscar muy bien las bibliotecas externas que se usaran, ya que no siempre están funcionando correctamente, en algunos casos están ni siquiera compilan.
Este caso fue particularmente evidente al buscar una biblioteca de C o C++ que pudiera realizar el cifrado con AES o DES, encontrándonos con bibliotecas que cifraban mal ya que al meter la misma llave no descifraban e incluso bibliotecas que no logramos hacer que compilaran.

7.2. Prototipo 2

Se implementó un módulo de cifrado, descifrado y generación de CAPTCHAS en Python, simulando el proceso antes del envío del correo y el que se hace después de la recepción de

los correos electrónicos.
Para este se usó el formato estándar del correo electrónico especificado en el RFC 822, también se usaron bibliotecas ya estandarizadas de python para la implementación de las funciones de hashing, funciones de cifrado y descifrado (AES o DES), funciones aleatorias y la generación de los CAPTCHAS.

Conclusión: Se logró generar todo el proceso de envío y parte del proceso de recepción de mensajes. En cuanto a el envió se logró leer el mensaje, crear una palabra a partir de funciones random, crear la llave con esa string y cifrar el correo exitosamente, además de esto se logró leer el archivo de mensaje de correo electrónico y cifrar únicamente el mensaje que viene en este.
Por su parte el módulo de generación de CAPTCHAS mostró muchos problemas para generarlos, ya que no logramos hacer que el intérprete pudiera encontrar correctamente las funciones de la biblioteca de generación CAPTCHAS por lo que al no poder generar un CAPTCHA la recuperación no se puede realizar como se planteó, para verificar únicamente que las funciones sirven se implementó el descifrado del mensaje en el mismo método.

Objetivo Generar una imagen un CAPTCHA a partir de una cadena de caracteres ingresada desde una interfaz gráfica.

7.3. Prototipo 3

Este prototipo se construyó en 2 partes; la primera parte fue la interfaz gráfica y sus herramientas, y la segunda en las herramientas para generar la imagen a partir de una cadena de caracteres.
Para la interfaz gráfica se utilizaron las siguientes herramientas para desarrollar este prototipo:
Biblioteca Qt y Qt creator: Utilizamos esta biblioteca para generar la interfaz gráfica con la que ingresara la cadena de caracteres y el IDE Qt Creator para facilitar la gestión de las clases.
La interfaz gráfica consta de un apartado para ingresar la cadena de caracteres y un botón para convertir la cadena a una imagen de CAPTCHAS.
Para generar CAPTCHAS se utilizaron las siguientes herramientas:
Lenguaje PHP: se utilizó para genera las imágenes CAPTCHAS con la cadena de caracteres proporcionada anteriormente.
En un principio se buscó una biblioteca que generara las imágenes CAPTCHAS en el lenguaje C++ pero su implementación no estaba optimizada y necesitaba ser adaptada casi en su totalidad que tener el funcionamiento deseado, por esta razón buscamos otra biblioteca que se adaptara más a la funcionalidad del prototipo, por lo tanto se optó por utilizar el lenguaje PHP ya que tiene librerías optimizadas para generar imágenes CAPTCHAS.
Conclusión. La generación de imágenes CAPTCHAS es rápida y fácil de implementar, pero durante la investigación nos dimos cuenta que el cliente de correo "Thunderbird" está desarrollado en el lenguaje de programación Python y al no tener una biblioteca nativa en el lenguaje C++ para convertir una cadena de caracteres en CAPTCHAS y decidimos cambiar de lenguaje de programación.

Objetivo del prototipo. Instalar y configurar un servidor de correo electrónico para el envío de mensajes de correo electrónico entre diferentes usuarios.

7.4. Prototipo 4

Instalación y configuración de un servidor de correo electrónico y un servidor DNS.

Para el desarrollo de este prototipo fue necesario instalar el servidor de correo electrónico con el protocolo pop y imap, un cliente de correo electrónico web, un servidor DNS y el servidor HTTP Apache. Estos 3 servicios fueron levantados en una computadora con un sistema operativo Xubuntu 15.04; primero se instaló el servidor HTTP [12], posteriormente pasamos a instalar el servidor DNS y configurar un dominio [13]; seguimos con la instalación del servidor de correo electrónico y los protocolos pop y imap; y por último se instaló y configuro el cliente de correo web [14].
Para la instalación de servidor HTTP fue necesario seguir los siguientes pasos:

- Abrimos una terminar en Ubuntu y escribimos el comando: "sudo apt-get install apache2"
- Abrimos como administrador el archivo /etc/apache2/sites-enabled/00-default.conf y escribimos la siguiente configuración:

```
<VirtualHost *:80>
    ServerAdmin nombredelsitio@example.com
    ServerName nombredelsitio
    ServerAlias www.nombredelsitio.com
    DocumentRoot /var/www/nombredelsitio.com/public_html/
    ErrorLog /var/www/nombredelsitio.com/logs/error.log
    CustomLog /var/www/nombredelsitio.com/logs/access.log combined
</VirtualHost>
```

- Levantamos el servicio http con el siguiente comando: "sudo service apahce2 start"
- Para verificar la instalación abrimos un explorador y escribirlos en la barra de búsqueda la siguiente dirección: http://localhost/ y nos aparecerá la siguiente pantalla.

Una vez instalado el servidor HTTP proseguimos a instalar el servidor DNS, para levantar este servicio es necesario seguir los siguientes pasos:

- Seleccionamos un nombre de dominio, para fines prácticos nuestro dominio privado será "correocifrado.edu".
- Abrimos una terminar en Ubuntu y escribimos el siguiente comando: "sudo apt-get install bind9"
- Realizar una copia de respaldo del archivo de configuración original con el comando "cp /etc/bind/named.conf.local /etc/bind/named.conf.local.original"

- Editamos el archivo de configuración con: “nano /etc/bind/named.conf.local”
- Agregamos al final del archivo lo siguiente:

```
zone "correocifrado.edu" {
type master;
file "correocifrado.edu.zone";
};

zone "10.168.192.in-addr.arpa" {
type master;
file "db.192.168.10";
};
```

- Procedamos a crear los (nuevos) archivos de zona, esos archivos contienen los registros del DNS y en Ubuntu se encuentran en el directorio /var/cache/bind/ “nano /var/cache/bind/db.isti.edu.ni.zone”
- En el archivo agregamos el siguiente texto:

```
  $ORIGIN correocifrado.edu.
$TTL 86400              ; 1 dia
@       IN      SOA ns.correocifrado.edu. info.correocifrado.edu. (
        2014112401      ; serie
        6H              ; refresco (6 horas)
        1H              ; reintentos (1 hora)
        2W              ; expira (2 semanas)
        3H              ; m nimo (3 horas)
)

@       IN      NS      ns
@       IN      MX 10   mail
ns      IN      A       192.168.10.10
mail    IN      A       192.168.10.10
www     IN      A       192.168.10.10
```

- De igual manera el archivo de zona de búsqueda inversa:
  nano /var/cache/bind/db.192.168.10
- Agregamos la siguiente configuración:

```
   $ORIGIN 10.168.192.in-addr.arpa.
$TTL 86400      ; 1 dia
@       IN  SOA ns.correocifrado.edu. info.correocifrado.edu. (
        2014112401      ; serie
        6H              ; refresco (6 horas)
        1H              ; reintentos (1 hora)
        2W              ; expira (2 semanas)
        3H              ; m nimo (3 horas)
```

```
)

@       IN      NS      correocifrado.edu.
10      IN      PTR      correocifrado.edu.
10      IN      PTR      mail.correocifrado.edu.
10      IN      PTR      www.correocifrado.edu.
```

- Procedemos a re-iniciar el servicio con el comando “service bind9 restart”
- Cambiar el primero de los servidores DNS por la IP del nuestro: “nameserver 192.168.10.10”
- Lo único que quede es realizar las pruebas en el cliente “nslookup www.correocifrado.edu”

Seguimos con la instalación del servidor de correo electrónico y los servicios del protocolo pop y imap con la aplicación courier-pop y courier-imap:

- Abrimos una terminal y escribimos el siguiente comando: “sudo apt-get install postfix”
- Durante la instalación nos aparecerá una pantalla de configuración, damos enter para aceptar la configuración.
- En tipo genérico de configuración de correo seleccionaremos "Sitio de Internet".
- A continuación indicaremos el nombre de sistema de correo, normalmente la dirección del dominio registrado, en nuestro caso çifradocorreo.net".
- Con esto veremos que postfix terminara de instalarse y procedemos a editar el archivo “/etc/postfix/main.cf”.
- Añadiremos al final del fichero main.cf las líneas:

```
  inet_protocols = ipv4
home_mailbox = emails/
```

- Una vez guardado el archivo que editamos procedemos a reiniciar el servidor con el comando “sudo /etc/init.d/posrfix restart”

Una vez instalado el servicio de correo electrónico procedemos a instalar el courier-pop y el courier-imap.

- Abrimos una terminal el Ubuntu y escribimos el siguiente comando “sudo apt-get install courier-pop”.
- Nos mostrará una ventana de configuración de courier-base, seleccionamos “NO”.
- Procedemos a instalar courier-imap con el siguiente comando “sudo apt-get install courier-imap”.
- Esperamos a que finalice la instalación.

Por ultimo necesitamos instalar una aplicación webmail para enviar correos entre usuarios del correo electrónico.

- Abrimos una terminar en Ubuntu y escribimos el siguiente comando “sudo apt-get install squirrelmail”.
- Tras la instalación de SquirrelMail lo configuraremos ejecutando el siguiente comando “sudo squirrelmail-configure”
- Selecciónanos la letra D y damos enter.
- En este nuevo menú tecleamos la opción courier y damos enter.
- Nos dará un informe de la configuración que seleccionamos y damos enter para continuar.
- Regresamos al primer menú, ahora tecleamos el número 2 y damos enter.
- Seleccionamos en este nuevo menú la opción 1 y damos enter nuevamente.
- Nos pedirá nuestro nombre de dominio, en nuestro caso es el dominio que configuramos en el servidor DNS “correocifrado.net”
- Nos regresara al menú principal y tecleamos la letra Q para salir de la configuración.
- Preguntara si queremos guardar los cambios y tecleamos la letra Y.
- Por ultimo ejecutamos el siguiente comando para levantar SquirrelMail en Apache “sudo ln -s /usr/share/squirrelmail /var/www/webmail”
- Reiniciamos el servicio apache con el comando “sudo service restart apache2”.
- Ya podremos entrar a la aplicación escribiendo el explorador “www.correocifrado.edu/webmail”

Para poder enviar correos se necesitan usuarios que desean enviar mensajes entre usuarios, primero crearemos un usuario

- Abrimos una terminal de Ubuntu y escribimos el siguiente comando “sudo adduser nombreusuario”.
- Introduzca la nueva contraseña de UNIX: introduciremos la contraseña para el usuario, es importante que sea segura (números, letras, mayúsculas y minúsculas) pues con el usuario y la contraseña podremos acceder vía web al servidor de correo electrónico desde cualquier parte del mundo.
- Vuelva a escribir la nueva contraseña de UNIX: repetiremos la contraseña.
- Full Name: introduciremos el nombre completo, por ejemplo .Alicia Robles Maldonado".
- Room Number: Número de oficina.
- Work Phone: teléfono del trabajo.
- Home Phone: teléfono particular.
- Other: otros datos del usuario.

- Respondemos "S.ª la pregunta "¿Es correcta la información?". Y ya tendremos el usuario creado en el sistema operativo, que también servirá como usuario (buzón) para el servidor de mail.
- Ahora generamos el buzón con el siguiente comando “sudo maildirmake /home/nombreusuario/emails”
- Cambiamos los permisos de las carpeta emails con el comando “sudo chdown nombreusuario:nombreusuario /home/nombreusuario/emails -R

Para crear otro usuario es necesario repetir los pasos anteriores.

Referencias

[1] Email, Internet: http://en.wikipedia.org/wiki/Email, Mayo, 2015

[2] Interactive Advertising Bureau, Marcelo Brodsky, "Reflexiones jurídicas sobre el e-marketing en Chile", Internet: http://www.iab.cl/reflexiones-juridicas-sobre-ele-marketing-en-chile.

[3] D. Jurafsky, Text Classification, Stanford University Natural Language Processing.

[4] S. Díaz Santiago y D. Chakraborty., "On Securing Communication from Profilers." Proceedings of International Conference on Security and Cryptography, Secrypt 2012, pp.154-162, Rome, Italy, 2012.

[5] Philippe Golle and Ayman Farahat. "Defending Email Communication Against Profiling Attacks" Proceedings of the 2004 ACM workshop on Privacy in the electronic society, ACM New York, NY, USA ©2004pp 39-40

[6] J. Klensin, "Simple Mail Transfer Protocol", Patent 5321, October 2008.

[7] J. Myers, "Post Office Protocol - Version 3", Patent 1939, May 1996.

[8] A. Gulbrandsen, "Internet Message Access Protocol (IMAP) - MOVE Extension", Patent 6851, January 2013.

[9] sime, Internet: https://www.gnupg.org/gph/es/manual/c190.html#AEN201, Noviembre2015.

[10] Ciphertext-only attack Internet: https://en.wikipedia.org/wiki/Ciphertext-only_attack,Noviembre2015.

[11] PGP Internet: https://es.wikipedia.org/wiki/Pretty_Good_Privacy, Noviembre2015

[12] HTTP Internet: http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=506, Noviembre2015

[13] DNS Internet: http://www.servermom.org/install-apache-php-mariadb-ubuntu-15-04/2208/, Noviembre2015

[14] web Internet: https://www.howtoforge.com/tutorial/ubuntu-perfect-server-with-apache-php-myqsl-pureftpd-bind-postfix-doveot-and-ispco Noviembre2015

[15] EM Internet: http://www.emclient.com/, Noviembre2015

[16] BOX Internet: https://www.postbox-inc.com/, Noviembre2015

[17] ZIM Internet: https://www.zimbra.com/, Noviembre2015

[18] OPERA Internet: http://www.opera.com/es-419/computer/mail, Noviembre2015

[19] THUN Internet https://www.mozilla.org/es-ES/thunderbird/, Noviembre2015

[20] CIT Internet http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.444.8759&rep=rep1&type=pdf, Noviembre2015

[21] DB Internet https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-syste Noviembre2015

[22] SQL Internet http://danielpecos.com/documents/postgresql-vs-mysql/, Noviembre2015