Aplicación de cifrado contra adversarios clasificadores, para el correo electrónico

Jonathan Arcos Ayala*, Allan U. Zepeda Ibarra†, Sandra Díaz Santiago‡ y Manuel A. Soto Ramos§
Escuela Superior de Cómputo, Instituto Politécnico Nacional
Ciudad de México.
Tel.57-29-6000 ext.52022
Email: *jonas.arc.99@gmail.com, † balaju01@gmail.com,‡sdiazsa@ipn.mx, §msotoa06@yahoo.com.mx

***Resumen*— En este Trabajo Terminal se desarrolló una herramienta de cifrado que protege al correo electrónico, contra un tipo especial de adversario, denominado adversario clasificador. El objetivo de este adversario es analizar una gran cantidad de información, para clasificar al usuario en categorías predeterminadas. La herramienta que se propone utiliza de manera novedosa los CAPTCHAs y una técnica criptográfica conocida como secreto compartido, para facilitar el acuerdo de las claves de cifrado. Adicionalmente, la aplicación utilizará un servidor de llaves para la autenticación de los usuarios para proveer una forma segura de enviar mensajes cifrados y recuperarlos.**

I. 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 [7]. 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 [1]. 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 [4], 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 [2].

Una solución alternativa fue presentada por Golle y Farahat [3] quienes proponen un protocolo que hace uso de una función de cifrado, el cual sustituye cada una de las palabras del mensaje por otra de la misma extensión y frecuencia gramatical, esta función esta pensada para textos en idioma inglés. Para cifrar se utiliza una clave que se genera usando los datos de cabecera que acompañan al 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 tiene con una complejidad de cálculo moderadamente mas alta que las funciones hash estándar.

Este protocolo resulta inseguro para la criptografía moderna 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 se permite que el texto se vea como un texto en inglés pero indistinguible para los clasificadores y por lo tanto este clasifica incorrectamente el mensaje cifrado.

Díaz y Chakraborty [2] propusieron un nuevo esquema que no requiere de un intercambio previo de claves, ni de una infraestructura de clave pública. Puesto que un adversario en cuestión es un programa de cómputo, tal esquema utiliza CAPTCHAs para esconder la clave de cifrado. Este esquema es el que se implementó en el presente trabajo.

El resto del artículo está organizado como sigue, en la Sección II-A se presentan los conceptos básicos asociados con criptografía, en la Sección II-B se describe el esquema propuesto por Díaz y Chakraborty, en la Sección III-A se

muestran los detalles de la implementación y finalmente en la Sección IV se presentan las conclusiones.

II. METODOLOGÍA

II-A. Preliminares

Definición 1: Un esquema de cifrado simétrico está conformado por una tripleta de algoritmos $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, definidos como se describe a continuación:

- El algoritmo generador de claves Gen selecciona una llave K al azar del conjunto de llaves $\mathcal{K}$, esto se denotará como $K \xleftarrow{\$} \mathcal{K}$. Esta clave K será usada por los algoritmos Enc y Dec, esta clave la compartirán emisor y receptor.
- El algoritmo de cifrado Enc, toma como entrada un texto en claro $M \in \mathcal{M}$ y una clave K generada por Gen y regresa un texto cifrado $C \in \mathcal{C}$. Usualmente esto se denota como $C \leftarrow \mathsf{Enc}_K(M)$.
- El algoritmo de descifrado Dec, toma como entrada un texto cifrado C y una llave K y regresa M. Esta operación se denota por $M \leftarrow \mathsf{Dec}_K(C)$. Para que cualquier algoritmo de cifrado simétrico funcione correctamente, se debe garantizar que para todas las llaves posibles en $\mathcal{K}$ y todos los posibles mensajes $\mathcal{M}$,

$$\mathsf{Dec}_K(\mathsf{Enc}_K(M)) = M.$$

Definición 2: Una función hash, es una función de un sólo sentidocuya entrada m es un mensaje de longitud arbitraria y la salida es una cadena binaria de longitud fija. Al resumen o hash de un mensaje m, se le denotará como $H(m)$. Una función hash debe tener las siguientes propiedades:

- Para cualquier mensaje m, debe ser posible calcular $H(m)$ eficientemente.
- Dado $H(m)$, debe ser computacionalmente difícil, hallar un mensaje m', tal que $H(m) = H(m')$.
- Debe ser computacionalmente difícil, hallar dos mensajes m y m' tales que $H(m) = H(m')$.

II-A.1. Esquema de Secreto Compartido de Shamir: El esquema de secreto compartido fue propuesto por Adi Shamir en 1997 [5]. El objetivo de este método es dividir un secreto K en w partes,que son dadas a w participantes. Para recuperar el secreto es necesario tener al menos u elementos de las w partes siendo $u \leq w$. Y no es posible recuperar el secreto si se tienen menos que u partes.

Para construir el esquema del secreto compartido primero es necesario seleccionar un número primo $p \geq w + 1$ el cual define al conjunto $\mathbb{Z}_p$.

El procedimiento para dividir un secreto K en w partes es el siguiente:

1. Se seleccionan w elementos distintos de cero del conjunto $\mathbb{Z}_p$ denotados como x_i donde $1 \leq i \leq w$.
2. Se seleccionan $u-1$ elementos aleatorios de $\mathbb{Z}_p$ denotados como $a_1, ..., a_{u-1}$.
3. Se construye el polinomio y_x de la siguiente forma. Sea

$$y(x) = K + \sum_{j=1}^{u-1} a_j x^j \text{ mód } \quad p \tag{1}$$

   Por medio de este polinomio se calculan los elementos y_i.
4. La salida es el conjunto $S = \{(x_1, y_1), ..., (x_w, y_w)\}$.

Para recuperar el secreto solo tenemos que resolver un sistema de ecuaciones que es definido por el polinomio característico $a(x) = a_0 + a_1 x + ... + a_{u-1}x^{u-1}$.

Posteriormente se seleccionan u pares de elementos (x_w, y_w) con los que obtendremos nuestro sistema de ecuaciones a resolver.

II-A.2. CAPTCHA: Es un programa informático diseñado para diferenciar un ser humano de una computadora, 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* por sus siglas en inglés) [6]. 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 comúnes son imágenes distorsionadas de cadenas cortas de caracteres. Para un humano es generalmente muy fácil recuperar la cadena original de la imagen distorsionada, pero es difícil para los algoritmos de reconocimiento de caracteres recuperar la cadena original de la imagen distorsionada.
Un CAPTCHA en un algoritmo aleatorio G, que recibe como parámetro una cadena de caracteres STR y produce como resultado un CAPTCHA $G(x)$

II-B. Esquema Díaz - Chakraborty

El esquema Díaz-Chakraborty [2] utiliza CAPTCHAs y un algoritmo de clave secreta, para proteger el correo electrónico. Para obtener la clave se genera una cadena al azar, a la se le aplica una función hash, con esta clave se cifra el mensaje. Tanto el mensaje cifrado como el CAPTCHA se envían al receptor. El receptor debe resolver el CAPTCHA, para obtener la cadena, aplicarle la función hash y así obtener la clave de cifrado. Puesto que un adversario clasificador es un programa de cómputo, no podrá resolver un CAPTCHA y por tanto no podrá obtener la clave de cifrado. Este esquema se muestra en la Figura 1.

Protocol $\mathbb{P}(x)$
1. $k \leftarrow \mathsf{STR}$;
2. $k' \leftarrow G(k)$;
3. $K \leftarrow H(k)$;
4. $c \leftarrow E_K(x)$;
5. **return** (c, k')

Fig. 1: Protocolo Díaz-Chakraborty.

Contemplando que es muy común que el usuario no consiga resolver el CAPTCHA Díaz y Chakraborty propusieron una variante del esquema anterior, el cual se describe a continuación. Se genera una cadena de caracteres aleatoriamente llamada STR la cual se codifica a un valor entero. El valor entero es dividido en 5 pares (x, k') por medio del algoritmo de Secreto Compartido, cada uno de los elementos k' de los pares generados es decodificado a su correspondiente valor en cadena de caracteres para posteriormente ser convertidos en CAPTCHAS. Para finalizar la cadena STR se introduce en una función Hash para generar la llave K. Con esta llave se cifra el mensaje de correo y se envía junto con los pares de $(x, CAPTCHA)$. Este esquema se puede observar en la Figura 2.

Protocol $\mathbb{P}'(x)$
1. $k \leftarrow \mathsf{STR}$;
2. $k' \leftarrow \mathsf{ENCD}(k, 0)$;
3. $\{(x_1, k'_1), \ldots, (x_w, k'_w)\} \leftarrow \mathsf{SHARE}^p_{u,w}(k')$;
4. **for** $i = 1$ **to** w;
5. $(k_i, \lambda_i) \leftarrow \mathsf{ENCD}^{-1}(k'_i)$;
6. $c_i \leftarrow G(k_i)$;
7. **end for**
8. $K \leftarrow H(k)$;
9. $C \leftarrow E_K(x)$;
10. **return** $[C, \{(x_1, c_1, \lambda_1), \ldots, (x_w, c_w, \lambda_w)\}]$

Fig. 2: Variante del protocolo Díaz-Chakraborty

Este nuevo esquema se creó pensando en que el usuario pueda tener más oportunidades de recuperar el mensaje cifrado y esto sucede gracias a el algoritmo de Secreto Compartido, ya que no este podemos tener la misma llave repartida en n CAPTCHAS. A continuación se describe las funciones ENCD y ENCD^{-1}, cuyo propósito es convertir una cadena de caracteres a enteros y viceversa.

II-B.1. Codificación de caracteres a enteros: Se tiene un conjunto de caracteres AL compuesto por $AL = \{A, B, ..., Z\} \cup \{a, b, ..., z\} \cup \{0, 1, ..., 9\} \cup \{+, /\}$ con una cardinalidad $|AL| = 64$.

Para obtener una representación binaria de 64 elementos son necesarios 6 bits por lo que para todos los elementos $\sigma \epsilon AL$ existe una cadena binaria. Una vez establecido esto el procedimiento para realizar la conversión es el siguiente:

1. Tomamos una cadena de caracteres y la separamos caracter por caracter y los intercambiamos por su correspondiente número entero en AL $\alpha_0||\alpha_1||...||\alpha_m$
2. Posteriormente cada uno de los enteros lo convertimos en un binario de 6 bits y se concatenan uno detrás del otro $\Psi \leftarrow bin_6(\alpha_0)||bin_6(\alpha_1)||...||bin_6(\alpha_m)$
3. La cadena binaria Ψ la convertimos a entero $v \leftarrow toInt(\Psi)$

II-B.2. Decodificación de enteros a caracteres: También es necesario convertir un entero a una cadena de caracteres y para esto se realiza el proceso inverso:

1. El entero v es convertido en un número binario $z = toBin_6(v)$
2. Separamos z en cadenas de 6 bits y cada una de ellas la interpretamos como un entero $toInt(z_0)||toInt(z_1)||...||toInt(z_w)$
3. Cada uno de estos valores se convierte a su correspondiente caracter en AL se concatenan para generar la cadena de caracteres final.

III. Resultados

III-A. Implementación

El cliente de correo electrónico se programó utilizando la biblioteca grafica GTK3+, el entorno gráfico de GNOME 3 y el lenguaje de programación Python. Para poder inicial el desarrollo de este prototipo es necesario instalar previamente la biblioteca gráfica GTK3+ y el entorno gráfico GNOME 3. Se inició el desarrollo de este prototipo generando una interfaz gráfica que ayude a establecer las configuraciones básicas para conectarse con los servidores de correo electrónico por los protocolos POP3 y SMTP. Esta interfaz también establece la configuración necesaria para conectarse con el servidor de CAPTCHAS y para escoger el protocolo de cifrado de mensajes que se utilizará.

- Configuración básica POP3: El cliente de correo electrónico establece una conexión POP3 con un servidor de correo electrónico para poder descargar los mensajes de un usuario, para poder hacerlo se necesita nombre de host, puerto de comunicación, nombre de usuario, contraseña del usuario y si se utilizara una conexión segura. Todos estos datos son proporcionados por el servidor de correo electrónico con el que se desea comunicar.
- Configuración básica SMTP: El envío de mensajes de correo electrónico se realiza estableciendo una comunicación con nuestro servidor de correo electrónico, para ello se necesita nombre de host, puerto de comunicación, nombre de usuario y contraseña del usuario. Al igual que en la configuración POP3 estos datos son proporcionados por el servidor de correo electrónico con el que se desea comunicar.
- Configuración con el servidor de CAPTCHAS: El envío de la imágenes CAPTCHAS generadas después del cifrado de los mensajes necesitan ser resguardadas en el servidor de CAPTCHAS, para ello es necesario enviar al servidor su dirección de correo electrónico, un nombre de usuario y una contraseña. El servidor valida si el usuario ya esta registrado, en caso contrario el servidor realiza el registro del usuario con los datos proporcionados.

Estas 3 configuraciones se establecen llenando los campos de la interfaz, ver Figura 3, la cual llamaremos ventana de configuración. Esta ventana genera un archivo JSON donde se guardan estos datos para poder ser utilizado mas adelante para el envío y recepción de mensajes de correo electrónico, así como la subida y descarga de las imágenes CAPTCHA y la selección entre los protocolos $\mathbb{P}$ y $\mathbb{P}'$ para cifrar los mensajes de correo electrónico.

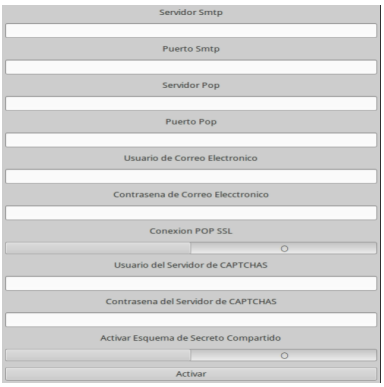

Fig. 3: Ventana de Configuración

Posteriormente se generó una interfaz gráfica principal para visualizar los mensajes de correos electrónicos y una segunda interfaz para la redacción de los mismos.

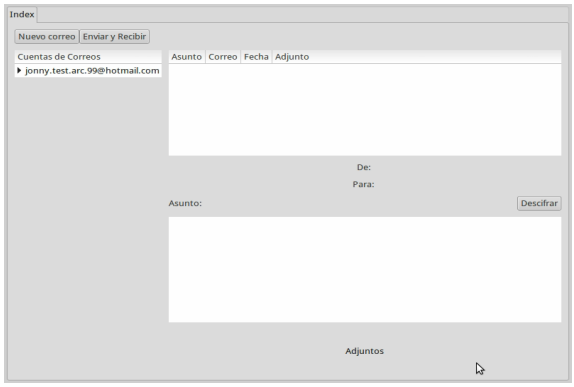

Fig. 4: Ventana Principal

La primera interfaz, también llamada ventana principal, esta dividida en 3 partes: una barra lateral, un listado y un visualizador de mensajes de correo electrónico. En la barra lateral encontramos las carpetas donde se almacenan los correo electrónicos, en el listado encontramos los mensajes de correos electrónicos que se han almacenado en la carpeta seleccionada de la barra lateral y por ultimo tenemos el visualizador de mensajes, el cual despliega la dirección de correo del usuario que mando ese mensaje, los destinatarios a donde fue dirigido el mensaje y por último el cuerpo del mensaje, ver Figura 4.

La segunda interfaz, también llamada ventana de envío de mensajes, tiene un diseño simple para redactar los mensajes de correo electrónico, esta interfaz cuenta con 3 espacios, el primero es para escribir la dirección de correo donde se enviará el mensaje; el segundo espacio es para escribir el asunto que se adjunta al mensaje; y por último espacio es para la redacción del mismo, ver Figura 5. Una vez que se tienen las interfaces listas se procede a darles funcionalidad, para ello se llevaron a cabo las siguientes actividades.

- Cifrado de mensajes de correo electrónico por el protocolo $\mathbb{P}$ y $\mathbb{P}'$: Esta actividad se inicia al redactar un correo electrónico en la ventana de envío de mensaje y pulsar el botón enviar. Lo primero que hace es tomar la fecha actual de la computadora y se concatena con las dirección de correo destino y origen, a esta cadena generada se le obtiene un digesto MD5, el cual será utilizado como firma para el mensaje de correo. Posteriormente se toma el mensaje redactado por el usuario y es enviado a la biblioteca de cifrado, especificando el protocolo a usar. Esta biblioteca nos regresa el mensaje cifrado junto con la ruta de la imágenes CAPTCHAS que descifran el mensaje. Después se toma este mensaje cifrado y se concatena con la firma generada anteriormente y con una cabecera que nos indicará si el mensaje esta cifrado o no al momento de visualizarlo. A continuación toma las direcciones de correo, origen y destino, el asunto redactado y el mensaje cifrado para generar un mensaje de correo electrónico y guardarlo en la carpeta de salida, éste mensaje será enviado posteriormente por el protocolo SMTP al servidor de correos. Por último esta actividad activa el envío de imágenes CAPTCHAS al servidor de CAPTCHAS.
- Envío de imágenes CAPTCHAS al servidor de CAPTCHAS: Esta actividad se inicia al momento de pulsar el botón “Enviar y Recibir” de la ventana principal o al termino del cifrado de mensajes de correo electrónico. Se toma un listado de los mensajes de correo que se tienen pendientes de envío en la carpeta de salida, buscando los CAPTCHAS correspondientes a cada mensaje. Cada uno de estos CAPTCHAS son enviados al servidor junto con las direcciones de correo origen y destino, la firma del mensaje de correo y los datos de configuración del archivo JSON por medio de una petición HTTP. Por último, por cada CAPTCHA enviado exitosamente se envía su correspondiente mensaje al servidor de correo por el protocolo SMTP.
- Envío de mensajes por el protocolo SMTP: Esta actividad se inicia al momento de pulsar el botón “Enviar y Recibir” de la ventana principal o al termino de un envío exitoso de un CAPTCHA. Para hacer el envío de un mensaje de correo electrónico se necesitan los datos de configuración que se tienen el el archivo JSON junto con el mensaje que se desea enviar. En caso de error el

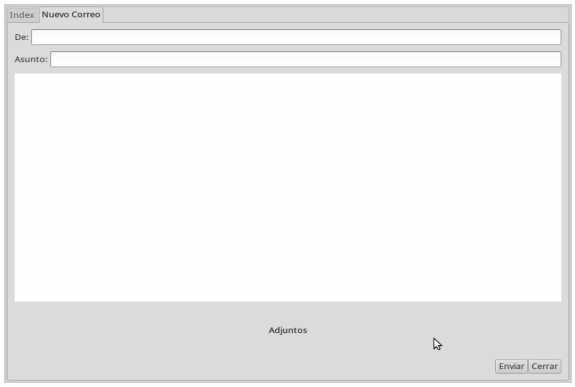

Fig. 5: Ventana de Nuevo Correo

mensaje se almacena en la carpeta de salida.

- Descargar mensajes por POP3: Esta actividad se inicia al momento de pulsar el botón “Enviar y Recibir” de la ventana principal. Para iniciar la descarga de los mensajes de correo electrónico se toman los datos básicos del archivo JSON para establecer comunicación con el servidor. Una vez establecida la conexión el servidor de correo electrónico nos dará uno a uno los mensajes y el cliente de correo electrónico guardará cada mensaje en un archivo txt en la carpeta de entrada.
- Descarga de imágenes CAPTCHAS del servidor de CAPTCHAS: Esta actividad se inicia al momento de pulsar el botón “Descifrar” de la ventana principal. Para saber si el mensaje esta cifrado se busca en el cuerpo del mensaje la cabecera de cifrado de donde obtenemos la firma del mensaje. Con la firma del mensaje se buscan las imágenes de descifrado en la carpeta CAPTCHA, esta carpeta se crea con la instalación del prototipo, en caso de no encontrar las imágenes en la carpeta el cliente de correo hacer una petición HTTP al servidor de CAPTCHAS adjuntando la firma del mensaje, las direcciones de origen y la dirección destino. El servidor contesta enviando la dirección URL de la imágenes de donde el cliente descarga las imágenes y las guarda en la carpeta CAPTCHA. Después de guardarlas, el cliente despliega la o las imágenes CAPTCHA en una ventana para que el usuario lo resuelva, esta ventana la llamaremos ventana de Descifrado. El despliegue de una o mas imágenes dependerá del protocolo que se haya utilizado para cifrar.

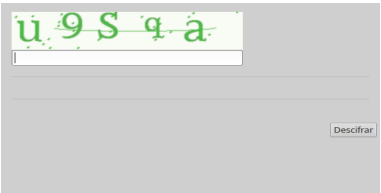

Fig. 6: Ventana CAPTCHAS

- Descifrado de mensajes de correo electrónico por el protocolo $\mathbb{P}$ y $\mathbb{P}'$: Esta actividad se inicia al momento de pulsar el botón “Descifrar” de la ventana de Descifrado. Una vez que el usuario resuelve los CAPTCHAS se toman las respuestas junto con el cuerpo del mensaje cifrado sin la cabecera de cifrado y se envían a la biblioteca de cifrado, la cual nos regresa el texto descifrado. En caso de que los CAPTCHAS sean ingresados incorrectamente el texto regresado por la biblioteca sera ilegible y el cliente de correos lo detectará.

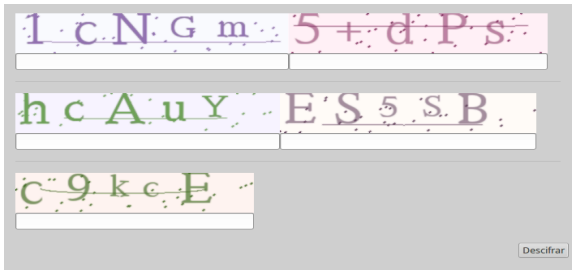

Fig. 7: Ventana Multi-CAPTCHAS

IV. Conclusiones

En el desarrollo de este trabajo terminal se encontraron varios problemas al desarrollar complementos para clientes de correo electrónico comerciales los cuales describiremos a continuación.

El primer problema encontrado fue la gran cantidad de tiempo que se invierte en la investigación, desarrollo, revisión y correcciones de los complementos que se implementan para los clientes de correo estándar, ya que si se desea publicar un complemento con la empresa que desarrollo el cliente, éstos son sometido a una evaluación para verificar que no altere el funcionamiento de otros módulos de su cliente.

Otro problema a tomar en cuenta es la fase de desarrollo en la que se encuentra el cliente de correo que se desea ocupar, ya que si se encuentra en una etapa muy temprana de desarrollo se encontrara poca documentación; las funciones disponibles serán limitadas; y muy probablemente cambien la compatibilidad entre módulos de una versión a otra.

La solución que se implementó fue desarrollar un cliente de correo electrónico que tuviera las funciones básicas de envío y recepción de mensajes de correo electrónico por los protocolos POP3 y SMTP, junto con la implementación de los protocolos $\mathbb{P}$ y $\mathbb{P}'$ del esquema Díaz – Chakraborty para el cifrado y descifrado de los mensajes de correo electrónico por medio de CAPTCHAS. Se observó que el intercambio de clave y la implementación de los protocolos $\mathbb{P}$ y $\mathbb{P}'$ del esquema Díaz - Chakraborty se llevó con éxito. También se concluye que estos esquemas pueden implementarse en los modelos actuales de comunicación de correo electrónico de una manera trasparente al usuario al momento del envío y recepción de los correos electrónicos. Cabe destacar que es la primera implementación funcional que se tiene del esquema de secreto compartido de Adi Shamir para el correo electrónico e inhibiendo los ataques de los agentes clasificadores.

Por último se encontró que las comunicaciones que se establecen actualmente entre los servidores de correo electrónico y los usuarios son canales seguros. Lo cual fue confirmado por las pruebas realizadas a la aplicación, por lo tanto se concluye que el ataque de los adversarios clasificadores se hace en los servidores de correo electrónico donde son almacenados los mensajes en claro y se tiene acceso a un gran número de mensajes para su clasificación.

REFERENCIAS

[1] M. Brodsky. Reflexiones jurídicas sobre el e-marketing en chile. Interactive Advertising Bureau, 2015. http://www.iab.cl/reflexiones-juridicas-sobre-el-e-marketing-en-chile/.

[2] S. Diaz-Santiago and D. Chakraborty. On securing communication from profilers. In P. Samarati, W. Lou, and J. Zhou, editors, *SECRYPT 2012 - Proceedings of the International Conference on Security and Cryptography, Rome, Italy, 24-27 July, 2012, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications*, pages 154–162. SciTePress, 2012.

[3] P. Golle and A. Farahat. Defending email communication against profiling attacks. In V. Atluri, P. F. Syverson, and S. D. C. di Vimercati, editors, *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, WPES 2004, Washington, DC, USA, October 28, 2004*, pages 39–40. ACM, 2004.

[4] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[5] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[6] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. In E. Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 294–311. Springer, 2003.

[7] Wikipedia. Email — Wikipedia, the free encyclopedia, 2015. http://en.wikipedia.org/wiki/Email.