



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Trabajo Terminal I.

**Autenticación mediante Chaffing and
Winnowing en el protocolo HTTP**

2018-B003.

Integrantes:

Blancas Pérez Bryan Israel
Carrillo Fernández Gerardo
Morales González Diego Arturo
Paredes Hernández Pedro Antonio

Directores:

Moreno Cervantes Axel Ernesto
Díaz Santiago Sandra

Índice.

I Trabajo Terminal I	6
1. Introducción	7
1. Objetivos.	8
1.1. Objetivo general.	8
1.2. Objetivos particulares.	8
2. Metodología.	9
2. Marco teórico	12
1. Introducción.	12
2. Métodos de autenticación en internet.	12
2.1. Cookies.	15
3. Introducción a la Criptografía.	17
3.1. Criptología.	17
3.2. Criptografía.	17
3.3. Objetivos de la criptografía.	19
3.4. Aplicaciones de la criptografía.	19
3.5. Criptografía simétrica y asimétrica.	20
4. Chaffing and Winnowing.	22
4.1. Historia	22
4.2. ¿Qué es Chaffing and Winnowing?	22
4.3. Objetivo de Chaffing and Winnowing	24
4.4. ¿Cómo funciona?	25
4.5. Propiedades de Chaffing and Winnowing	26
4.6. All-or-Nothing and the Package Transform (AONT)	27
4.7. Comparando Chaffing and Winnowing contra Cifrado y Esteganografía	29
3. Análisis.	32
1. Prototipo I.	32
1.1. Descripción.	32
1.2. Herramientas a usar.	32

1.3.	Estudio de requerimientos.	35
1.4.	Reglas del negocio.	36
2.	Prototipo II.	36
2.1.	Descripción.	36
2.2.	Herramientas a usar.	36
2.3.	Estudio de requerimientos.	38
2.4.	Reglas del negocio.	39
4.	Desarrollo.	40
1.	Prototipo I.	40
1.1.	Diagrama de casos de uso.	40
1.2.	Descripción de casos de uso.	41
1.3.	Diagrama de flujo de datos (DFD).	46
1.4.	Diagrama de clases.	47
1.5.	Diagrama de secuencia.	48
1.6.	Diagrama de actividades	49
1.7.	Interfaz de usuario.	50
1.8.	Requisitos de diseño.	51

Índice de figuras.

2.1.	Esquema del protocolo de criptografía simétrica.	21
2.2.	Esquema del protocolo de criptografía asimétrica.	21
2.3.	Charles agrega los paquetes inválidos.	23
2.4.	Charles no agrega los paquetes pero multiplexa los flujos. . . .	24
2.5.	Secuencia de Chaffing después del proceso de autenticación. .	25
2.6.	Las dos maneras para el proceso de chaff pueden ser utilizadas. Los paquetes chaff son los mensajes de color rojo.	26
2.7.	Visión general del proceso Chaffing and Winnowing.	26
2.8.	Proceso de Chaffing and Winnowing junto con AONT.	29
2.9.	Visualizando el método Chaffing and Winnowing cómo un es- quema de cifrado.	30
4.1.	Diagrama de casos de uso del Prototipo I.	40
4.2.	Diagrama de flujo de datos del Prototipo 1.	46
4.3.	Diagrama de clases del Prototipo I.	47
4.4.	Diagrama de secuencia del Prototipo I.	48
4.5.	Diagrama de actividades del Prototipo I.	49
4.6.	Logo de la extensión.	50
4.7.	Pantalla inicial. Servicio activado.	50
4.8.	Pantalla inicial. Servicio desactivado.	51

Índice de cuadros.

2.1. Comparación de la aplicación en los distintos métodos de autenticación	13
2.2. Comparación de la seguridad en los distintos métodos de autenticación	14
4.1. DCU: PLCU1	41
4.2. DCU: PLCU2	43
4.3. DCU: PLCU3	44

Glosario.

Cookies Es una pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del navegador. Sus principales funciones son recordar accesos y conocer información sobre los hábitos de navegación e intentos de spyware. 15

CSS Cascading Style Sheets. 33, 35

Flash Aplicación informática englobada en la categoría de reproductor multimedia. 33

HTML Hyper Text Markup Lenguaje. 32–35

HTTP Hypertext Transfer Protocol. 8–11, 32, 35, 41, 42

ID identificador. 12

Identity theft También conocido como "robo de identidad" se produce cuando una persona adquiere, transfiere, posee o utiliza información personal de una persona física o jurídica de forma no autorizada, con la intención de efectuar o vincularlo con algún fraude u otro delito. 15

IU Interfaz de Usuario. 36

Netcape Navegador web de la compañía NetScape Communications. 34

Parte I

Trabajo Terminal I

Capítulo 1

Introducción

En la actualidad la mayoría de los usuarios de internet necesitan guardar contraseñas para sus distintas cuentas en las diferentes páginas web a las que ingresan, ya que recordarlas es un problema debido a la gran cantidad de servicios que se utilizan en la actualidad. Como consecuencia de que la autenticación por contraseña es la más utilizada en los servicio web hoy en día [1], los distintos servicios web han implementado mecanismos de seguridad tales como contraseñas que contengan un mínimo de caracteres determinados, al menos un caracter especial, entre otros. Esto ha provocado que éstas sean más difíciles de recordar y han orillado a los usuarios a optar por guardarlas en medios físicos o digitales para recordarlas cuando sea necesario. Sin embargo, perder esas claves (principalmente con los medios físicos) presenta un grave problema de seguridad, teniendo como consecuencia: perdida de datos sensibles, robo de identidad, robo de cuentas bancarias, etc.

La gran mayoría de servicios web han implementado la función recordar contraseña”, la cual hace que el usuario no tenga que ingresar sus credenciales¹ cada vez que se quiere acceder al servicio. Esta función por lo general hace uso de cookies que es información almacenada en el explorador web del usuario y que representa una amenaza de seguridad ya que permite rastrear la información de navegación del usuario y esto puede ser muy útil para sitios fraudulentos o se puede presentar un robo de cookies con el cual los intrusos podrían hacerse pasar como el usuario entre otros problemas que éstas representan.

El método Chaffing and Winnowing proporciona una excelente confidencialidad de los mensajes sin la necesidad de cifrados o estenografía pero no cumple los demás objetivos de criptografía pero si lo combinamos con algún

¹Credenciales se entiende como los datos que un servicio web requiere para poder acceder al él. Comúnmente son 'usuario' y 'contraseña'.

cifrado asimétrico proporciona un nivel de seguridad muy alto que es lo que pretendemos en este trabajo.

Es por ello, que en este trabajo terminal, propone un nuevo método de autenticación por medio del método de *Chaffing and Winnowing* y con la ayuda de una extensión de Google Chrome, la cual servirá para la inyección de las credenciales de la autenticación del usuario en el protocolo HTTP. Así, si un servicio web tiene este tipo de autenticación disponible, lo podrá validar. El propósito principal de este trabajo es que los usuarios puedan realizar un inicio de sesión más cómodo, seguro y sin la necesidad de recordar sus distintas contraseñas.

1. Objetivos.

1.1. Objetivo general.

Realizar una extensión en Google Chrome que modifique los datos del protocolo HTTP, para permitir que el servidor detecte el método de autenticación propuesto basado en *Chaffing and Winnowing*.

1.2. Objetivos particulares.

- Investigar e implementar el desarrollo de extensiones en Google Chrome.
- Investigar sobre los mecanismos de autenticación.
- Investigar sobre la técnica de *Chaffing and Winnowing* para adaptar su implementación.
- Inyectar el código (la autenticación) en el encabezado HTTP para enviar la petición al servidor.
- Modificar el código del servidor Apache para simular y comprobar el funcionamiento de la extensión.
- Realizar pruebas de seguridad para comprobar la eficacia de la extensión.

2. Metodología.

El proceso de desarrollo que seguiremos estará basado en la metodología de prototipos evolutivos, el cual consiste en la implementación parcial del proyecto cumpliendo con los requerimientos que van surgiendo a lo largo del desarrollo, de esta manera es posible ir experimentando con un prototipo parcialmente funcional e identificar posibles mejoras o fallas con el fin de lograr el objetivo final. Esta metodología está compuesta por las siguientes fases:

- Fase de investigación preliminar.
- Especificación de requerimientos y prototipos
- Diseño técnico
- Programación y pruebas
- Operación y mantenimiento

Primero tendremos la fase de “investigación preliminar”, donde se van a definir las metas principales, después en la fase de “especificación de requerimientos y prototipos”, se hace el diseño básico para dar paso a la creación del primer prototipo correspondiente, y después verificar el cumplimiento de los requerimientos y de ser necesario modificarlo hasta que los cumpla. En la tercera fase (diseño técnico) se realiza un diseño detallado y la documentación necesaria para que en la cuarta fase (programación y pruebas) se implemente y se pruebe el prototipo. Finalmente, en la última fase (Operación y mantenimiento) se hace la liberación y el mantenimiento del prototipo final.

Para nuestro proyecto realizaremos 4 prototipos, los cuales son:

1. Creación de extensión de Google Chrome para interceptar la petición HTTP.
 - Investigación preliminar:
 - Investigar sobre el desarrollo de extensiones en Google Chrome.
 - Especificación de requerimientos y prototipos:
 - Ejecución de la extensión sobre Google Chrome.
 - Detectar petición HTTP e interceptarla.
 - Subir archivo autenticador a la extensión.

- Diseño técnico:
 - Documentación del prototipo.
 - Documentación del prototipo.
 - Desarrollo de la extensión.
 - Pruebas de la extensión.
2. Inyección de código autenticador (Chaffing) en el encabezado HTTP.
- Investigación preliminar:
 - Investigación sobre el método Chaffing and Winnowing.
 - Especificación de requerimientos y prototipos:
 - Lectura del archivo autenticador.
 - Análisis del protocolo HTTP.
 - Inyección del código autenticador sobre el protocolo HTTP.
 - Mandar petición a servidor.
 - Diseño técnico:
 - Documentación del prototipo.
 - Programación y pruebas:
 - Desarrollo del complemento de la extensión.
 - Creación del algoritmo de inyección de código.
 - Pruebas de la extensión.
3. Modificación del servidor Apache para recibir el protocolo con la inyección de código.
- Investigación preliminar:
 - Investigación sobre el servidor Apache.
 - Analizar la arquitectura del servidor Apache
 - Investigación sobre la versión conveniente a modificar.
 - Especificación de requerimientos y prototipos:
 - Recibir petición HTTP de la extensión.
 - Detectar el tipo de autenticación que se usará.
 - Diseño técnico:
 - Documentación del prototipo.
 - Programación y pruebas:

- Descargar la versión del servidor Apache a usar.
 - Modificación del código del servidor Apache para detectar el tipo de autenticación que se usará.
 - Pruebas de funcionamiento.
4. Realización de la autenticación (Winnowing) en el servidor para realizar el login.
- Investigación y pruebas
 - Investigar sobre la implementación de autenticador en distintos servidores
 - Especificación de requerimientos y prototipos:
 - Recibir la petición.
 - Descifrar la petición.
 - Dar respuesta al usuario.
 - Diseño técnico:
 - Documentación del prototipo.
 - Programación y pruebas:
 - Creación de algoritmo que obtenga el código autenticador del protocolo HTTP.
 - Verificación del código autenticador.
 - Responder al usuario.
 - Pruebas de funcionamiento.

Capítulo 2

Marco teórico

1. Introducción.

Dado que este Trabajo terminal relaciona temáticas muy enfocadas a la seguridad y aspectos web, es necesario conocer algunos conceptos e ideas fundamentales tanto para entender el trabajo como para conocer su funcionamiento, por lo tanto será necesario hablar de métodos de autenticación, concepto, objetivos, aplicaciones y tipos de criptografía así como el método de Chaffing and Winnowing que es la parte interesante de todo este trabajo; en este marco teórico tratamos de explicar de manera breve y con un enfoque directo al uso que le daremos en el desarrollo del proyecto.

2. Métodos de autenticación en internet.

Con la evolución de la web, distintas páginas ofrecen ciertos servicios a los usuarios y con la finalidad de dar una experiencia óptima y segura, se requiere que una persona o usuario se identifique para el uso de estos servicios, es aquí donde entra en juego el papel de los métodos de autenticación. Para poder asegurar la confidencialidad de la información manejada en todos estos servicios, es necesario restringir el acceso de este, para esto se utiliza la identificación y la autenticación; la identificación es un procedimiento donde el sujeto es reconocido por algún ID, esto es equivalente al saber una parte de información en específico, mientras que la autenticación es el proceso de validación de si el sujeto es la persona quien dice ser al tratar de identificarse.[21]

Para probar una identidad, el sujeto presenta algo llamado "factor de autenticación", principalmente existen 4:

- El sujeto tiene algo (Token, documento, un material específico, etc.).

- El sujeto conoce algo (contraseña, login, respuesta a una pregunta, etc.).
- El sujeto tiene una característica biológica (huella dactilar, ADN, etc.).
- El sujeto se encuentra en un lugar en específico (dirección IP, información de un lugar en específico, etc.).

Hoy en día, la autenticación por contraseña es el método más utilizado, más que otra cosa por su ventaja principal: su simplicidad de utilización, sin embargo, así como tiene una gran ventaja, la autenticación por contraseña también tiene muchos problemas y desventajas de seguridad.

A continuación, mostraremos algunas tablas comparativas que sirvan para tener una mejor perspectiva de las ventajas, desventajas, vulnerabilidades de los diferentes métodos de autenticación, entre otras cosas:

	Recordar	Otros dispositivos	Acciones	Facilidad	Tiempo	Errores	Recuperación
Contraseñas	1	3	2	3	3	2	3
Otros recursos	2	3	3	3	3	3	2
Contraseñas gráficas	1	1	2	3	3	2	3
Contraseñas dinámicas	1	3	2	2	3	2	2
Tokens	3	1	1	2	2	3	1
Multivariación	1	1	1	3	2	2	1
Criptografía	3	1	1	1	1	2	1
Biométricos	3	3	2	3	2	2	1

Cuadro 2.1: Comparación de la aplicación en los distintos métodos de autenticación

La tabla anterior concentra las siguientes características:

- Recordar: Hace referencia a que tan complicado es que un usuario se acuerde de los datos necesarios para la autenticación.
- Otros dispositivos: El usuario usa una entidad externa para facilitar su autenticación.
- Acciones: Hace referencia a que tantas acciones adicionales se deben de realizar para autenticarse.
- Facilidad: Simplicidad de tecnología.
- Tiempo: Cantidad de recursos temporales que consume el método de autenticación.

- Errores: Posibles errores durante la autenticación.
- Recuperación: Denota la dificultad de recuperar la clave de acceso en caso de pérdida.

En el cuadro No.2 se muestra una tabla comparativa del nivel de seguridad en los distintos métodos de autenticación, donde 1 - baja seguridad, 2 – media seguridad y 3 – alta seguridad.

	Ataque por fuerza bruta	Observación	Hackeo indirecto	Phishing
Contraseñas	1	1	1	1
Otros recursos	2	2	3	3
Contraseñas gráficas	1	1	2	2
Contraseñas dinámicas	2	3	2	2
Tokens	3	3	3	3
Multivariación	1	1	3	3
Criptografía	3	3	3	3
Biométricos	3	3	1	1

Cuadro 2.2: Comparación de la seguridad en los distintos métodos de autenticación

La tabla se enfoca principalmente en los siguientes problemas de seguridad:

- Ataque por fuerza bruta: Se descifra el método de autenticación con una gran cantidad de intentos, usualmente generados por un programa.
- Observación: Cuando se intenta ver directamente los datos necesarios para la autenticación desde una distancia cercana hasta incluso usando binoculares, cámaras o algún otro dispositivo.
- Hackeo indirecto: El usuario confía sus datos del método de autenticación a terceros quienes pueden ser atacados.
- Phishing: Hace referencia a programas que se hacen pasar por entidades confiables para interceptar los datos que desean.

Seguridad en internet En la actualidad, el incremento constante de internet ha impactado directamente en la seguridad de la información que se maneja cotidianamente y por la mayoría de usuarios. Existen infinidad de sitios donde es aplicada la seguridad, ya que sin ésta, se verían afectados

todos los usuarios en sus cuentas, pudiendo verse afectados desde un posible Identity theft (Robo de identidad), hasta la perdida de dinero real dado que la base de algunas de éstas paginas son E-Commerce, estas paginas implican el manejo de tarjetas de crédito, paypal, etc.

Uno de los puntos más críticos de la seguridad en Internet son las herramientas que interactúan de forma directa con los usuarios. Es común escuchar sobre fallas en los sistemas de protección de los servidores más frecuentemente utilizados, por ejemplo Apache, NGINX, IIS, etc. O en los lenguajes de programación en que son escritas las aplicaciones. [9] Sin embargo, la vulnerabilidad más grande dentro de un sistema, son los ataques directos a los usuarios finales durante la autenticación.

2.1. Cookies.

Durante la navegación por internet, la información sobre la computadora puede ser colectada y almacenada. Ésta puede ser de carácter general sobre el equipo y puede ser también información más específica sobre los hábitos de navegación del usuario, toda esta información guardada se le conoce como Cookies[12].

A continuación se muestran los diferentes tipos de cookies que existen para los navegadores.

- **Cookies propias:** Las cookies se gestionan desde el terminal o dominio de un mismo editor.
- **Cookies de terceros:** Las cookies no son enviadas por el propio editor, sino por otra entidad.
- **Cookies de sesión:** Los datos recabados sólo se recogen mientras el usuario está navegando por la página web.
- **Cookies persistentes:** Los datos continúan almacenados en el terminal y se puede acceder a ellos durante un periodo de tiempo determinados.
- **Cookies técnicas:** Permiten controlar el tráfico y la comunicación de datos.
- **Cookies de personalización:** Dejan a los usuarios acceder según algunas características propias que se recogen (navegador, idioma, etc.).

- **Cookies de análisis:** Recogen datos sobre el comportamiento de los usuarios y permiten elaborar un perfil de usuario.
- **Cookies publicitarias:** Recogen datos sobre la gestión de los espacios publicitarios.

Las cookies persistentes son aquellas que se almacenan en el equipo para que las preferencias personales puedan ser retenidas, ayudan a los sitios web a recordar tu información y ajustes cuando los visitas más adelante. Esto conlleva un acceso más rápido y sencillo ya que, por ejemplo, no se tiene que iniciar sesión de nuevo. Además de la autenticación, otras páginas web tienen más funciones para las cookies permanentes, como: selección de idioma, selección de tema, preferencias de menú, marca-páginas internos de la web, o favoritos. [13] Muchos navegadores pueden ajustar el periodo de tiempo en que las cookies persistentes deben ser almacenadas.

Gracias a las cookies persistentes, las direcciones de correo electrónico aparecen por default cuando se abre el correo electrónico, o en páginas de inicio personalizadas cuando se visita en línea un comercio. Si un atacante obtiene acceso puede recopilar información personal del usuario través de estos archivos y poder robar toda información del usuario. Es fácil acceder a estas cookies y obtener fácilmente la información del usuario, por lo que es necesario que el usuario nunca deje vulnerable esta información o en su debido caso borrar cookies al término de cada sesión. Existen diferentes funcionalidades para las cookies, una de las más importantes es la funcionalidad de seguridad, ya que contiene información importante de los usuarios. A continuación se muestran las diferentes funcionalidades de las cookies.

- **Preferencias:** Sirven para que la página se visualice atendiendo a los gustos del usuario, como por ejemplo idioma, región o tamaño de textos.
- **Seguridad:** Se encargan de autenticar a los usuarios y evitar el uso fraudulento de las credenciales por parte de terceros.
- **Procesos:** Son utilizadas para el correcto funcionamiento de la página en el navegador.
- **Publicitarias/Estadísticas:** Se usan para que la publicidad que se muestre sea personalizada.
- **Estados de la sesión:** Obtienen información del comportamiento del usuario en una página web, como por ejemplo el tiempo que pasa en una página, los clicks que realiza o la publicidad que le aparece.

3. Introducción a la Criptografía.

3.1. Criptología.

Para comprender que es la criptografía, es necesario que comprendamos qué es la "Criptología", palabra que proviene del griego «*kryptós*» (oculto) y «*logos*» (estudio). Según la Real Academia Española significa 'estudio de los sistemas, claves y lenguajes ocultos o secretos', sin embargo, brindándole un contexto a esta definición decimos que es el arte y ciencia que se encarga de diseñar sistemas para ocultar mensajes, y buscar la manera de romper dichos sistemas. [2]

La criptología contiene dos ramas, las cuales son: el criptoanálisis y la criptografía. Ésta última es de vital importancia en este trabajo terminal, por lo que a continuación se explicará qué es, su historia, sus objetivos y sus usos que tiene esta rama en la actualidad.

3.2. Criptografía.

"Criptografía", palabra proveniente del griego «*kryptós*» (oculto) y «*graphos*» (escribir), es definida por la Real Academia Española como 'el arte de escribir con clave secreta o de un modo enigmático'. Nuevamente, la definición de la RAE nos da un panorama general de lo que trata esta rama de la criptología, sin embargo, en la actualidad tenemos definiciones más extensas y precisas que nos ayudan a entender las funciones de este arte.

A continuación, se presentan dos definiciones de la criptografía, cabe mencionar que estas definiciones están orientadas al uso de la criptografía en la informática y las telecomunicaciones actualmente.

Jorge Ramio Aguirre nos brinda la siguiente definición en su libro 'Seguridad informática y criptografía'. [3]

"Rama inicial de las matemáticas y en la actualidad también de la informática y la telemática, que hace uso de métodos y técnicas con el objeto principal de cifrar, y por tanto proteger, un mensaje o archivo por medio de un algoritmo, usando una o más claves."

Finalmente, Menezes, Van Oorschot y Vanstone, no brindan en su libro una definición formal de lo que es la criptografía (traducción). [4]

"Estudio de técnicas matemáticas relacionadas con los aspectos de la seguridad de la información tales como la confidencialidad, la integridad de datos, la autenticación de entidad y del origen de datos. La criptografía no comprende sólo a los medios para proveer seguridad de información, sino a un conjunto de técnicas."

A elección del lector elegir aquella definición que le convenza más, nosotros una vez finalizado la exposición de estas definiciones de criptografía, continuaremos con una breve remembranza de la historia de esta disciplina.

En egipto hace 4000 años, tuvo su primera aparición la criptografía, cuando un maestro egipcio escribió la historia de su señor utilizando jeroglíficos poco comunes tratando de imprimir cierta jerarquía. Este primer acercamiento a la criptografía, utilizaba las técnicas de sustitución y transposición de símbolos de una manera similar a la base del concepto general de cifrado. Posteriormente, las antiguas civilizaciones occidentales comienzan a adoptar estas técnicas para mantener determinada información oculta, principalmente con propósitos militares, diplomáticos y religiosos. Mientras la criptografía crecía alrededor del mundo, el criptoanálisis también lo hacía, con el objetivo de hallar el mensaje original a partir de un mensaje cifrado, sin conocer el método utilizado. [5][6]

El la historia conocida después de lo antes mencionado, existe un punto crucial en el desarrollo de la criptografía tal y como la conocemos hoy en día hasta la llegada de las computadoras. Este punto fue la Segunda Guerra Mundial, en donde se construyeron máquinas de cifrado mecánicas y electromecánicas que aceleraban el proceso de cifrado y descifrado. Los alemanes desarrollaron la famosa máquina 'Enigma', que precisamente mediante unos rotores automatizaba el proceso de cifrado y descifrado de mensaje, brindándole al ejército una ventaja considerable en la inversión de tiempo en la comunicación.

3.2.1. Criptografía moderna.

En la actualidad, el término '*criptografía moderna*' hace alusión al desarrollo de esta ciencia en las áreas de la teoría de la información y las comunicaciones. Claude Shannon, considerado por muchos como el padre de la criptografía matemática, en su libro "Sistemas secretos" estableció las bases para las implementaciones de los algoritmos actuales a mediados de los años 50s. [6]

En los años 70s, el público general tuvo acceso al trabajo de Claude, además surgió la llegada de las computadoras y la publicación el primer borrador del algoritmo de criptografía simétrica DES, el cual fue el primer algoritmo público, basado en técnicas matemáticas y criptográficas modernas. Todo ello representó los cimientos para un rápido crecimiento en la criptografía, hasta eventualmente llegar a otro hecho sumamente relevante que determinó gran parte de las transacciones que realizamos hoy en día en internet. Dicho hecho

fue el artículo de las nuevas direcciones de criptografía hecho por Whitfield Diffie y Martin Hellman, el cual trataba de un nuevo método para distribuir llaves, que eventualmente se llamaría criptografía asimétrica.

3.3. Objetivos de la criptografía.

Algunos de los objetivos que se busca con la implementación de la criptografía para la seguridad de la información son los siguientes:

- **Confidencialidad:** este objetivo, también conocido como privacidad de la información, implica mantener en secreto una determinada información, por tanto, sólo aquellas personas que estén autorizadas tendrán acceso a ella.
- **autenticación:** este objetivo, implica hablar de la corroboración de la identidad de una entidad, por tanto, asegura que la entidad de donde la información es originada pueda ser identificada.
- **Integridad:** este objetivo asegura que determinada información no haya sido alterada por personas no autorizadas o por cualquier otro medio no conocido.
- **No repudio:** este objetivo previene que una entidad niegue un envío de información de un acuerdo preestablecido.

3.4. Aplicaciones de la criptografía.

Las aplicaciones que tiene la criptografía son muy variadas, dependiendo del ámbito donde se está aplicando. Las siguientes aplicaciones, son sólo algunas de las tantas que hay y provienen de distintos usos que se le dan a los objetivos que tiene la criptografía aplicada a la seguridad de la información. [6]

- **Autorización:** permiso concreto, de una parte o entidad, para el acceso o la realización de una tarea específica.
- **Validación:** proveer una autorización para el uso o la manipulación de información o recursos.
- **Control de acceso:** restricción de acceso a la información o recurso.
- **Certificación:** respaldo de información por una entidad externa de confianza.

- **Confirmación:** acuse de recibo a servicios que han sido dados.
- **Anonimato:** ocultamiento de la identidad de una entidad.

3.5. Criptografía simétrica y asimétrica.

Anteriormente en la historia de la criptografía, se hizo una rápida mención del nacimiento de estos dos métodos de cifrado, en esta sección se intentará explicar más profundamente sus funciones con el fin de que el lector conozca un poco más y entienda porque hemos decidido utilizar determinado método para cumplir con los objetivos de este trabajo.

3.5.1. Criptografía simétrica.

En la criptografía simétrica, tanto el emisor como el receptor comparten una única llave secreta para cifrar y descifrar la información que se desee transmitir. Esto implica que ambas partes de la comunicación deben tener un acuerdo antes de que se realice la comunicación. La seguridad de este tipo de algoritmos radica en mantener segura la llave secreta, por tanto, si ésta es revelada, cualquiera con acceso a ella puede descifrar el mensaje. Por estas razones, este tipo de criptografía puede ser visto como "criptografía de llave privada". Algunos de los algoritmos más famosos de criptografía simétrica son: DES (Data Encryption Standard), TripleDES, AES (Advanced Encryption Standard) y IDEA (International Data Encryption Algorithm).[17]

En el siguiente esquema se muestran los pasos que sigue un protocolo de criptografía simétrica. Definados 'A' como una entidad que pretende enviar un mensaje a otra entidad llamada 'B'. Luego entonces, ambas partes acordarán una 'Llave Secreta' con la que 'A' cifrará el mensaje utilizando un algoritmo establecido, mandando el resultado (Texto Cifrado) a 'B' que descifrá el mensaje con la misma llave y algoritmo que 'A'.

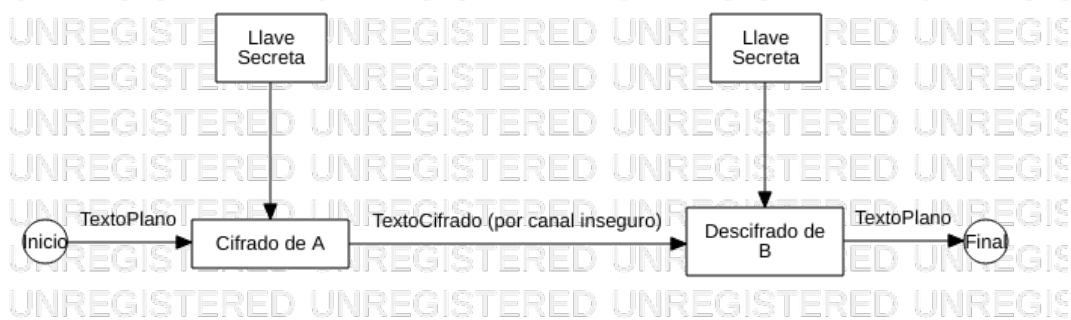


Figura 2.1: Esquema del protocolo de criptografía simétrica.

3.5.2. Criptografía asimétrica.

En los algoritmos de criptografía asimétrica, el receptor posee una llave pública y una llave privada para poder descifrar los mensajes. Por lo que las llaves tanto publica como privada son diferentes, y como sus nombres lo dicen, la llave publica puede ser mostrada a cualquier usuario y la llave privada sólo puede tenerla el usuario propietario del par de llaves. Por lo tanto, podemos llamar llave de cifrado a la llave publica y llave de descifrado a la llave privada. Algunos de los algoritmos más famosos de criptografía asimétrica son: RSA y ElGamal. [17]

En el siguiente esquema se muestran los pasos que sigue un protocolo de criptografía asimétrica. Definamos 'A' como una entidad que desea enviar información a otra llamada 'B'. Luego entonces, 'B' enviará a 'A' su llave pública para que 'A' cifre la información utilizandola. Cuando la información (TextoCifrado) haya viajado a través del canal inseguro para que 'B' la reciba, 'B' descifrará el TextoCifrado con su llave privada.



Figura 2.2: Esquema del protocolo de criptografía asimétrica.

4. Chaffing and Winnowing.

4.1. Historia

Chaffing and Winnowing es una técnica criptográfica que logra confidencialidad sin usar ningún proceso de cifrado para el envío de datos sobre un canal inseguro. El nombre **Chaffing and Winnowing** el nombre proviene de la agricultura: Después de que el grano ha sido cosechado y trillado, el grano es mezclado con paja fibrosa no comestible. La paja y el grano son separados por el movimiento de las hojas y la paja es descartada. Esta técnica fue creada por Ron Rivest y fue publicada en un artículo en línea el 18 de Marzo de 1998. [20] Aunque parece ser similar a un cifrado tradicional y esteganografía, chaffing and winnowing no puede ser clasificado como uno de ellos.

Esta técnica permite el envío de datos evitando la responsabilidad del cifrado de su contenido. Cuando se usa chaffing and winnowing, el emisor transmite el mensaje sin cifrar (texto plano). Aunque el emisor y el receptor comparten una llave, ellos la usan sólo para autentificar. Sin embargo, una tercera parte puede hacer su comunicación confidencial durante el envío simultáneo de mensajes especialmente mensajes diseñados a través del mismo canal.

4.2. ¿Qué es Chaffing and Winnowing?

Chaffing and Winnowing es un nuevo esquema establecido por Rivest en 1998. Este esquema ofrece confidencialidad para el contenido de un mensaje sin involucrarse con cifrado ni estenografía. [17]

El proceso **Chaffing** no hace uso de un cifrado por lo que no tiene una "clave de cifrado". Este proceso consiste en agregar paquetes inválidos (Información innecesaria) al mensaje a enviar, haciendo que el mensaje viaje seguro a la vista de todos los posibles "atacantes".

El proceso de **Winnowing** no emplea algún tipo de cifrado, por lo que al igual que el proceso chaff no tiene una "clave de descifrado". Intentando regular la confidencialidad que provee un cifrado damos paso a la esteganografía y el proceso de winnowing.[20]

Existen dos partes en el envío de mensajes con winnowing: Autenticación (Agregando MACs) y agregando paquetes chaff. Nosotros nos enfocaremos más al uso de paquetes chaff para el envío seguro de información, ya que, el receptor es quien remueve los paquetes chaff para obtener el mensaje original.

Los siguientes esquemas explican como es que se lleva a cabo el proceso de **Chaffing and Winnowing** en diferentes escenarios.

Escenario 1: Alice se está comunicando con Bob en un solo camino de comunicación sobre un canal inseguro y Charles agrega los paquetes de Chaff.



Figura 2.3: Charles agrega los paquetes inválidos.

En el escenario anterior Alice y Bob se están comunicando mutuamente por un canal de comunicación no seguro, en donde son enviados paquetes no cifrados. Alice y Bob comparten la llave de autenticación la cual será usada para el proceso de autenticación. Cuando Alice envía un mensaje a Bob, su mensaje es autenticado de su lado y es enviado a Charles antes de ser enviado a Bob. Charles agrega los paquetes chaff a la secuencia transmitida por Alice, al agregar los paquetes chaff, Charles provee confidencialidad para la comunicación entre Alice y Bob. Pero donde Charles no conoce la llave secreta compartida entre Alice y Bob. Por lo que el proceso de chaffing no necesita ningún conocimiento de la llave secreta de autenticación compartida.

Escenario 2: Alice se comunica con Bob en un camino de comunicación inseguro y en el cual Charles no agrega los paquetes chaff si no que multiplexa los flujos de las otras dos partes (David y Elaine). Este escenario es diferente al anterior, ya que se multiplexa el flujo de datos de Alice y Bob con el flujo de datos de David y Jane, y cuando el paquete llega a Bob el flujo de paquetes de David hacia Jane es el chaff de Bob y es descartado y vice versa para Jane.

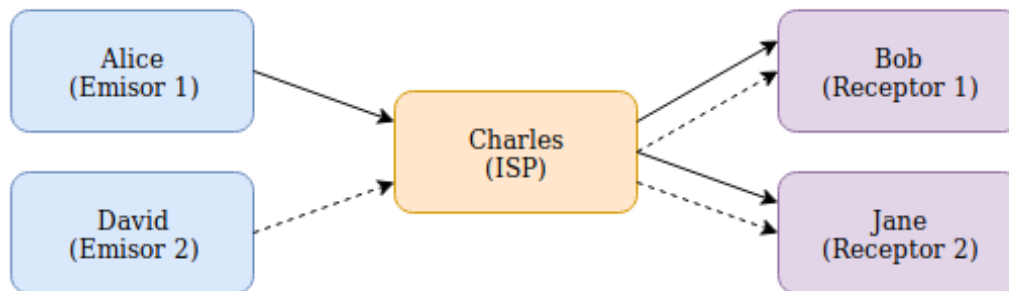


Figura 2.4: Charles no agrega los paquetes pero multiplexa los flujos.

Escenario 3: Alicia se comunica con Bob en un canal de comunicación inseguro y Alice no agrega los paquetes chaff. En este escenario, Alice desarrolla la autenticación de sus mensajes, por lo que Alice aplica chaffing para autenticar los mensajes y producir una secuencia de paquetes que serán transmitidos a Bob por la vía de Charles. Bob lleva a cabo el proceso de winnowing para recuperar el mensaje original.

4.3. Objetivo de Chaffing and Winnowing

El objetivo de seguridad del esquema de chaffing-and-winnowing es proporcionar privacidad en un entorno simétrico. Desde un punto de vista de seguridad, este esquema debe tratarse simplemente como un esquema de cifrado simétrico. Hay algunos procesos de "cifrado" que toman un mensaje y crean un "texto cifrado", y algún proceso de "descifrado" toma el texto cifrado y recupera el mensaje, ambos operando bajo una clave secreta en común. (Para el esquema Chaffing and Winnowing es la clave para el MAC). Estos procesos no se implementan de manera "habitual", pero, de manera abstracta, deben existir, de lo contrario no se logra la privacidad. [18] [19]

"No es una propiedad de seguridad novedosa, sino un conjunto novedoso de restricciones en los procesos dirigidos a lograr una propiedad de seguridad estándar"

"Encontrar-luego-adivinar". Extensión más directa al caso simétrico de la noción de indistinguibilidad.

Haciendo uso de **Chaffing and Winnowing** se asegura que los adversarios no obtengan información del mensaje transmitido a lo largo de un canal de comunicación inseguro entre dos partes.

Rivest propone un esquema, el cual cuenta con tres partes principales. [17]

1. **Autenticación** Es el proceso de descomponer el mensaje original en un paquete más pequeño y complementar cada paquete con un código de autenticación de mensaje (MAC).
2. **Chaffing** Es el proceso de agregar paquetes inválidos (Chaff packets).
3. **Winnowing** Es el proceso de remover paquetes Chaff para obtener el mensaje original en texto plano.

4.4. ¿Cómo funciona?

El esquema de Chaffing and Winnowing deja que cada paquete conste de:

- Un número de serie
- Contenido del paquete
- Código de autenticación de mensaje

Cuando son enviados los paquetes, el mensaje con el texto plano se descompone en pequeños paquetes los cuales contienen datos y el tamaño del paquete original. Entonces, el emisor (Alice) usa el algoritmo de **código de autenticación de mensaje** (MAC) para generar el valor MAC para ser agregado al paquete y el cual se basa en el número de serie, contenido del paquete y la llave autenticación. A continuación se muestra la salida del paquete después del proceso de autenticación.

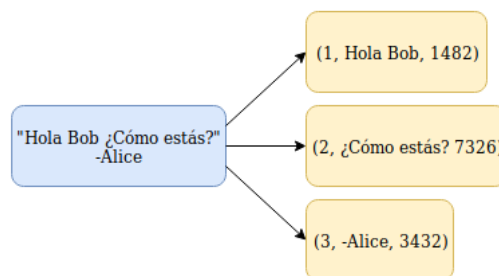


Figura 2.5: Secuencia de Chaffing después del proceso de autenticación.

Esta secuencia de paquetes es enviada a Charles (ISP) para llevar a cabo el proceso de Chaffing. Charles agrega paquetes chaff a la secuencia de paquetes antes de ser enviados por medio del canal de comunicación y ser recibidos por Bob.

Existen dos maneras donde Charles puede enviar la secuencia de chaff hacia Bob. La primera es enviando aleatoriamente mezclados los paquetes chaff para formar una secuencia y la otra manera es enviarlos de manera ordenada por el número de serie seguido del contenido del mensaje. En la siguiente figura se muestra cómo es el proceso de chaff en esta secuencia.

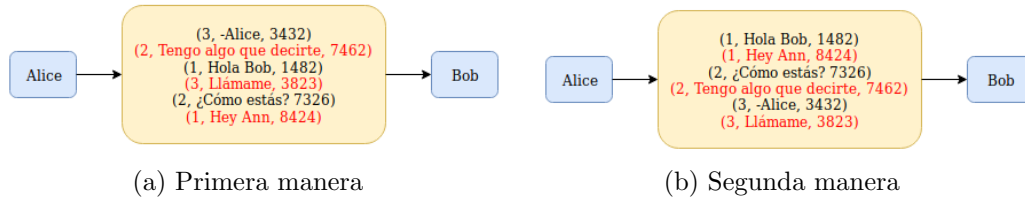


Figura 2.6: Las dos maneras para el proceso de chaff pueden ser utilizadas. Los paquetes chaff son los mensajes de color rojo.

Una vez que la secuencia de chaff llega a Bob, el último proceso es Winnowing. Bob determina la secuencia del mensaje que es válida del paquete chaff usando una función hash para el contenido de cada paquete y la llave de autenticación para re-calcular el MAC y compararlo contra el MAC del paquete recibido, si la comparación falla, el paquete chaff es descartado. Si la comparación es valida, entonces el paquete es parte del mensaje original. La siguiente imagen muestra el proceso completo de chaffing and winnowing.

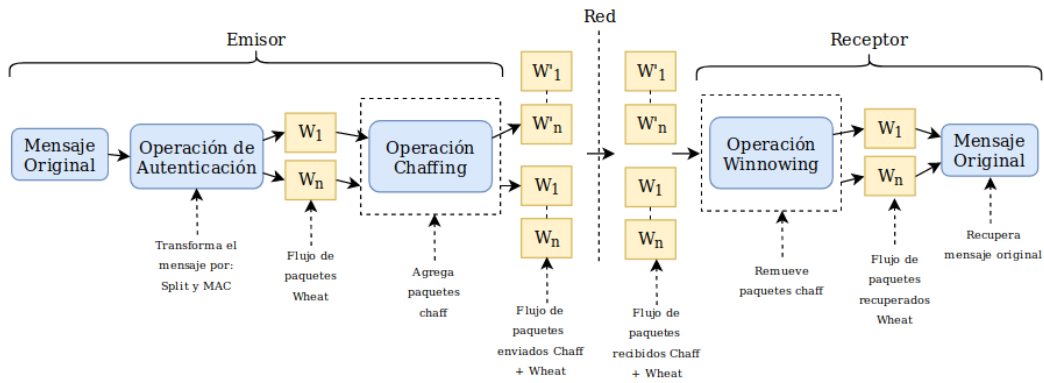


Figura 2.7: Visión general del proceso Chaffing and Winnowing.

4.5. Propiedades de Chaffing and Winnowing

- La técnica de Chaffing y Winnowing no depende de la fuerza del esquema de cifrado para proporcionar confidencialidad debido al hecho de que es muy difícil distinguir la información útil de los paquetes chaff sin la clave secreta. Por lo tanto, la dificultad de distinguir la información útil del chaff proporciona confidencialidad al esquema.
- La operación de Chaffing puede ser realizada por un tercero, ya que la clave secreta compartida no es necesaria en el proceso del mismo.
- Los paquetes de Chaff no tienen que contener datos aleatorios, ya que uno podría usar un mensaje válido con una clave secreta diferente para hacer el paquete de Chaff. Cuando el receptor recibe esos paquetes de Chaff, se verán como paquetes de Chaff, ya que la clave que se usa para volver a calcular el Chaff es diferente de la que los hace.

4.6. All-or-Nothing and the Package Transform (AONT)

All-or-Nothing and the Package Transform es una variación dentro de la técnica Chaffing and Winnowing, donde se mejora la eficiencia de su esquema original. AONT es la transformación de pre-procesamiento que permite a las partes enviar más datos (en términos de bit) por paquete en lugar de solo uno. Este pre-procesamiento es una transformación sin cifrado que toma el mensaje de texto sin formato y produce un mensaje empaquetado que luego se procesa de la manera normal de Chaffing and Winnowing. Las definiciones de la transformación AONT son las siguientes:

1. El algoritmo de transformación es **reversible**: Dado el bloque de mensaje transformado, el receptor puede obtener el mensaje de texto sin formato original.
2. El algoritmo de transformación y su inverso son **computables** de manera eficiente: Lo que significa que es computacionalmente factible recrear el texto original dada la llave privada y recibir todos los paquetes con éxito.
3. La transformación no es **computacionalmente factible**: Esto significa que si se ha recibido parte del paquete de la transmisión, cualquiera que esté intentando leer el mensaje no puede hacerlo ya que la transformación **AONT** requiere que se reciba todo el mensaje, de lo contrario no entrega nada.
4. La transformación es una **técnica sin cifrado**: La técnica de pre-procesamiento no tiene llaves y no hay una llave secreta compartida

involucrada en la operación. Cualquier persona que haya recibido todos los mensajes transformados del paquete puede recuperar el mensaje de texto original.

¿Cómo funciona AONT?

Supongamos que el mensaje de entrada es el siguiente: m_1, m_2, \dots, m_n . Seleccionamos una llave aleatoria K' el cual se usará para la función del paquete de transformación.

Se calcula la secuencia transformada m'_1, m'_2, \dots, m'_s para $s' = s + 1$ como se muestra a continuación:

Tenemos:

$$m_i \otimes E(K', i) \text{ for } i = 1, 2, 3, \dots, s$$

También:

$$m'_{s'} = K' \otimes h_1 \otimes h_2 \otimes \dots \otimes h_s$$

Donde:

$$h_i = E(K_0, m'_i \otimes i) \text{ for } i = 1, 2, \dots, s$$

Donde K_0 es una llave conocida pública fija.

Para que el receptor en el otro extremo obtenga el K_0 , el cual es la llave para el uso de **AONT**, el receptor realiza el siguiente cálculo:

$$K' = m'_s \otimes h_1 \otimes h_2 \otimes \dots \otimes h_s$$

$$m_i = m'_i \otimes E(K', i) \text{ for } i = 1, 2, \dots, s$$

AONT toma el mensaje de texto sin formato de entrada y los transforma, luego crea un bloque para almacenar los mensajes transformados antes de pasar al proceso de autenticación. Después, se genera el paquete Chaff (la cantidad de paquetes Chaff no tiene que ser igual a los paquetes de la información útil).

Esta técnica produce una menor sobrecarga que la sugerencia número 1. El AONT ofrece más confidencialidad al esquema de Chaffing and Winnowing, ya que el adversario debe recibir todo el bloque de mensajes de transformación e identificar correctamente todo el paquete de la información útil para obtener el mensaje de texto original. La siguiente figura muestra la descripción general de Chaffing y Winnowing si se agrega la función AONT.

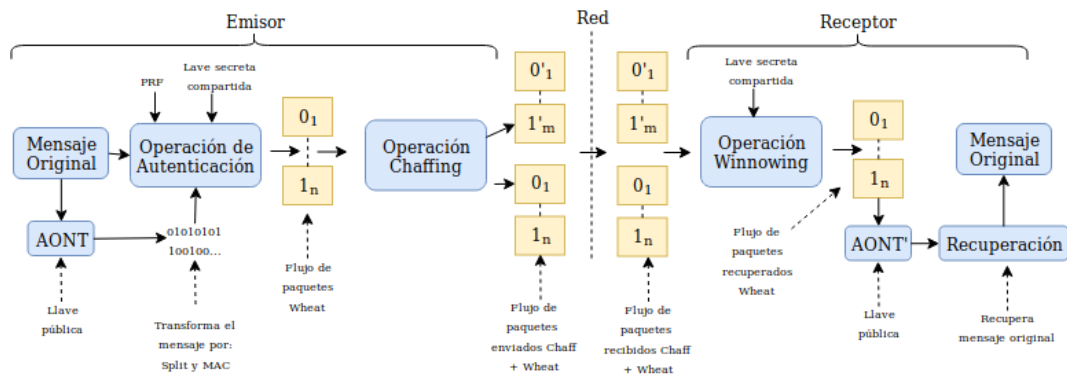


Figura 2.8: Proceso de Chaffing and Winnowing junto con AONT.

¿Cómo AONT puede hacer la diferencia?

1. Requiere menos ancho de banda al transferir paquetes, ya que se pueden transferir más bits en un paquete en lugar de un bit por paquete.
2. Los paquetes Chaff son más fáciles de generar, ya que AONT transforma el mensaje de texto plano en bits aleatorios.
3. La distinción entre Chaffing and Winnowing es más difícil: Si el adversario va a ejercer fuerza bruta en los paquetes, la tarea se ralentizará por el factor del número de bloque de mensajes. Dado que el bloque de mensaje de información adicional se mezcla aleatoriamente dentro de los flujos de paquetes de Chaffing and Winnowing, sin saber que es muy difícil que el bloque adicional proporcione la posibilidad de elegir el bloque de mensaje correcto de los paquetes para obtener el texto plano original.

4.7. Comparando Chaffing and Winnowing contra Cifrado y Esteganografía

En esta sección explicaremos porque Chaffing and Winnowing no puede ser clasificado como una técnica de cifrado o Esteganografía.

4.7.1. Chaffing and Winnowing vs Cifrado

Nosotros podríamos clasificar Chaffing and Winnowing como un método de cifrado, pero volvamos a recordar el principio de un Cifrado. El principal

objetivo de un cifrado es ocultar el mensaje en texto plano de tal manera que oculta su contenido con el uso de una clave de cifrado para el texto cifrado. Por otro lado, en el esquema original de Chaffing and Winnowing, una llave compartida es usada con el fin de autentificar la validación de los paquetes ya sea del emisor o del receptor. Además, Chaffing and Winnowing no hace uso de ninguna técnica de cifrado para ocultar el contenido de un mensaje y que nadie pueda ver dicho mensaje, solo aquellos con la llave correspondiente pueden determinar que paquetes contienen la información válida. A continuación, se muestra como se puede ver el esquema Chaffing and Winnowing como una técnica de cifrado.

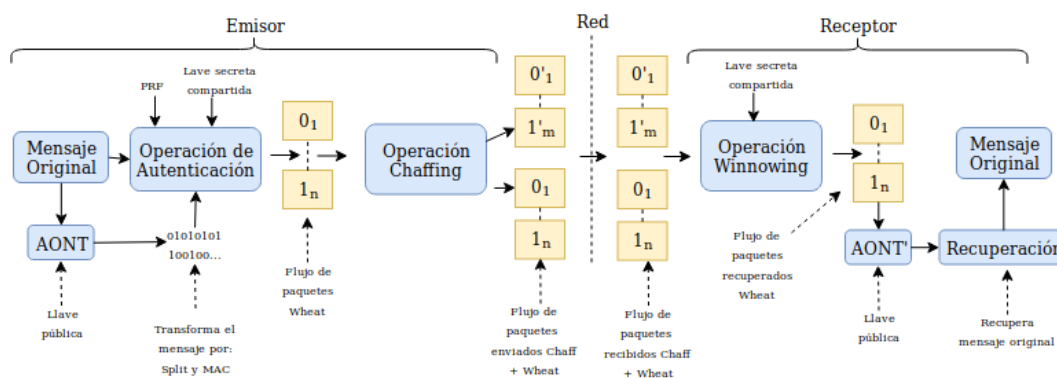


Figura 2.9: Visualizando el método Chaffing and Winnowing cómo un esquema de cifrado.

Chaffing y Winnowing pueden verse como **un tipo especial de esquema de cifrado simétrico**, ya que la operación **chaffing** es similar al "proceso de cifrado". En la operación de chaffing, el texto cifrado se crea para producir un paquete de la información útil no válido que se envía al receptor. Luego, el receptor realiza el "proceso de descifrado", que implica descartar el paquete de desperdicios y recuperar el mensaje original. Ambas operaciones operan bajo una llave secreta común que se usa para derivar el valor MAC.

Pero la diferencia es, *Chaffing y Winnowing* dos partes que no buscan lograr la confidencialidad. El emisor comparte una clave secreta con el receptor para que el receptor pueda usar la clave secreta para autenticarse (si se afirma que el mensaje recibido proviene del remitente deseado). Pero la ganancia de confidencialidad proviene de la dificultad de distinguir el paquete Chaff del paquete de la información útil. Mientras que en el cifrado, la clave se utiliza para lograr la confidencialidad mediante la creación de texto

cifrado que oculta el contenido del mensaje de personas.

Chaffing y Winnowing junto con el esquema AONT, el esquema en sí es muy parecido al cifrado, excepto que la clave que se usa en la transformación AONT se elige aleatoriamente cada vez en lugar de fijarla. Además, el último bloque de mensajes es exclusivo o de la clave y todo el hash del bloque de mensajes está allí para garantizar que cualquier modificación en el bloque de mensajes cambiará la clave K' calculado por el receptor. Por lo tanto, el último bloque de mensajes $m'_{s'}$ está allí solo con el propósito de autenticación. Por lo tanto, Chaffing y Winnowing con el esquema AONT no pueden ser clasificados bajo cifrado.

4.7.2. Chaffing and Winnowing vs Esteganografía

Para algunas personas, Chaffing and Winnowing puede ser clasificado como una técnica esteganografía. Sin embargo, el objetivo principal de la esteganografía es el de ocultar el mensaje original dentro de otro tipo de mensaje, por lo tanto, nadie aparte del emisor y el receptor sabrá que hay un mensaje oculto. Contrario a Chaffing and Winnowing, en donde cualquiera puede ver el contenido del mensaje, ya que este método no trata de esconderlo de los posibles atacantes.

Otra diferencia es que en esteganografía el emisor tiene que ocultar el mensaje el mismo, mientras que en Chaffing and Winnowing no necesariamente es así, ya que una "tercera parte" puede hacerlo.

Por lo tanto, Chaffing and Winnowing no puede ser considerado como esteganografía.

Capítulo 3

Análisis.

1. Prototipo I.

1.1. Descripción.

En este prototipo se busca la creación de una extensión de Google Chrome que pueda interceptar una petición HTTP hecha por el mismo navegador. Mientras la extensión se encuentre habilitada, será capaz de poder recibir las peticiones realizadas por el navegador y evitar que ésta sea mandada al servidor. Además mostrará en otra pestaña del navegador información sobre la petición interceptada.

El propósito de realizar este prototipo es familiarizarse con el manejo de extensiones en el navegador Google Chrome; como es que podemos obtener la información que necesitamos para que posteriormente modifiquemos esta petición y la enviemos al servidor.

1.2. Herramientas a usar.

Software.

Para el desarrollo de software de este prototipo, es necesario hacer mención de algunas de las siguientes herramientas, para tener una idea clara sobre qué herramientas estamos utilizando y porque es que las estamos utilizando:

HTML5.

Hyper Text Markup Lenguaje comenzó mucho tiempo atrás con una simple versión propuesta para crear la estructura básica de páginas web, organizar su contenido y compartir información, todo esto tenía la intención de comunicar información por medio de texto. El limitado objetivo de HTML motivó

a varias compañías a desarrollar nuevos lenguajes y programas para agregar características a la web nunca antes implementadas.

Dos de las opciones propuestas fueron Java y Flash; ambas fueron muy aceptadas y consideradas como el objetivo de la internet, sin embargo, con el crecimiento exponencial del internet, éste dejó de ser únicamente para los aficionados de los computadores y pasó a ser usado como un campo estratégico para los negocios y para la interacción social, ciertas limitaciones presentes en ambas tecnologías probaron ser una sentencia de muerte. Esta falta de integración resultó ser crítica y preparó el camino para la evaluación de un lenguaje del cual hablaremos un poco más a detalle después: JavaScript. Sin embargo, pese a su gran impacto, el mercado no terminó de adoptarlo plenamente y rápidamente su popularidad fue declinando, y el mercado terminó enfocando su atención a Flash. No fue hasta que los navegadores mejoraron su intérprete para JavaScript y la gente se empezaba a dar cuenta de las limitaciones que ofrecía Flash, que JavaScript fue implementado y comenzó a innovar la forma en la que se programaba la web. Al cabo de unos años, JavaScript, HTML y CSS eran considerados como la más perfecta combinación para evolucionar la Web.

HTML5 es una mejora de esta combinación, lo que unió todos estos elementos. HTML5 propone estándares para cada aspecto de la Web y también un propósito claro para cada una de las tecnologías involucradas. A partir de esto, HTML provee los elementos estructurales, CSS se concentra en como volver esta estructura utilizable y atractiva a la vista, y JavaScript tiene todo lo necesario para brindar dinamismo y construir aplicaciones web completamente funcionales. Cabe mencionar que HTML5 funciona diferente dependiendo del navegador y la versión en la que se esté trabajando, algunos soportan más características o diferentes funcionalidades que otros. [8]

JavaScript.

JavaScript es considerado como el lenguaje de programación de HTML y de la web. Es un lenguaje de programación fácil de usar y muy versátil para el ámbito de la comunicación en redes. Los programas, llamados "scripts", se ejecutan en el navegador (Mozilla, Google Chrome, Internet Explorer, etc.) normalmente consisten en unas funciones que son llamadas desde el propio HTML cuando algún evento sucede.

Su primera aproximación a un uso real, fue en mayor parte para "dar vida a una página web", como dar animaciones a un botón, interacciones en

tiempo real, entre otras más. JavaScript fue desarrollado por Netscape, a partir del lenguaje Java, que en ese momento tenía mucho auge y popularidad, y su principal diferencia es que JavaScript sólo "funciona" dentro de una página HTML.

JavaScript fue declarado como estándar del European Computer Manufacturers Association (ECMA) en 1997, y poco después, también fue estandarizado por ISO.[7]

JavaScript es un lenguaje interpretado, usado mayormente como complemento de ciertos objetivos específicos, sin embargo, uno de las innovaciones que ayudó a JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento del código. La clave de los motores más exitosos fue transformar el código de Javascript en código máquina para obtener una velocidad de ejecución mejor que antes. Esto a la vez permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la Web.

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, JavaScript fue expandido en cuestión de portabilidad e integración, a la vez, interfaces de programación de aplicaciones (APIs) fueron incorporando por defecto con cada navegador para asistir a JavaScript en funciones elementales. El objetivo de esto, fue principalmente hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la Web.[8]

Hardware.

En el ámbito del hardware, utilizaremos los equipos de cómputo con los cuales contamos actualmente los integrantes de este equipo, los cuales se especificarán a continuación:

Equipo de hardware utilizado.	
Marca	Asus
Modelo	X550VC
Procesador	Intel Core i5
Tarjeta de video	NVidia GeForce 720
Memoria RAM	12 GB
Disco duro	1TB

Equipo de hardware utilizado.	
Marca	HP
Modelo	Pavilion g4
Procesador	Intel Core i3
Tarjeta de video	Intel Sandybridge Mobile
Memoria RAM	6 GB
Disco duro	500GB

1.3. Estudio de requerimientos.

1.3.1. Requerimientos Funcionales.

PI_RF1. Interceptar petición HTTP. La extensión deberá interceptar la petición HTTP del navegador, en cuanto el usuario realice alguna a través del navegador.

PI_RF2. Deshabilitar extensión. El usuario podrá deshabilitar la extensión, para que ésta no vigile su actividad en el navegador.

PI_RF3. Habilitar extensión. El usuario podrá habilitar la extensión, para que ésta vigile constantemente cuando éste realice una petición HTTP.

PI_RF4. Validar petición. La extensión deberá analizar la petición previamente recibida, y validar si ésta es HTTP(S) o no.

PI_RF5. Mostrar petición. La extensión deberá mostrar en otra pestaña del navegador, la información de la petición que se haya realizado.

PI_RF6. Evitar salida de petición. La extensión deberá evitar que la petición salga red, deteniendola hasta cuando sea necesario.

1.3.2. Requerimientos no Funcionales.

PI_RNF1. Plataforma de implementación. La extensión será implementada en el navegador Google Chrome.

PI_RNF2. Versión del navegador La extensión funcionará a partir de la versión 65.0.3325.181.

PI_RNF3. Tecnologías para la interfaz de usuario Para el sistema se hará uso de HTML, JavaScript, CSS, JSON.

PI.RNF4. Permitir ejecución de JavaScript en Google Chrome.

Para el correcto funcionamiento de la extensión, es necesario que se permita la ejecución de javascript en el navegador Google Chrome.

1.4. Reglas del negocio.

PI.RN1. Extensión habilitada. En cuanto el usuario lo indique por medio de la Interfaz de Usuario, la extensión deberá vigilar la actividad que éste realice en el navegador para interceptar una petición.

PI.RN2. Extensión deshabilitada. De la misma manera, en cuanto el usuario lo indique, la extensión deberá dejar de vigilar la actividad que éste realice en el navegador.

2. Prototipo II.

2.1. Descripción.

En este prototipo se busca que la extensión de Google Chrome pueda modificar la petición HTTP previamente interceptada. La modificación se hará sólo mientras la extensión esté habilitada, y tiene como objetivo inyectar el código autenticador en el encabezado del protocolo. Una vez que dicho código es inyectado, la extensión deberá liberar la petición para que salga a red y llegue al servidor del servicio web correspondiente.

El propósito de este prototipo es utilizar la técnica de *Chaffing and winnowing* en este método de autenticación propuesto, para evitarle al usuario la tediosa tarea de ingresar sus credenciales y brindarle la seguridad necesaria al iniciar de sesión.

2.2. Herramientas a usar.

Software.

Para el desarrollo de software de este prototipo, es necesario mencionar que utilizaremos las mismas tecnologías que en el prototipo anterior y agregaremos unas cuantas herramientas que utilizaremos para poder cumplir correctamente con el prototipo

API FileSystem

Consiste en una API muy útil para trabajar con archivos en el entorno de desarrollo de Google Chrome, soporta la entrada de archivos y directorios desde una computadora personal. La ventaja de esta API es que no es necesario acceder a todos los archivos del usuario si no que se crea una especie de unidad virtual en donde se localizan los archivos que el usuario desee introducir al código, en este caso la extensión. [22]

OpenSSL SSL (Secure Sockets Layer o capa de conexión segura) es un estándar de seguridad global que permite la transferencia de datos cifrados entre un navegador y un servidor web. Es utilizado por millones[23] de empresas e individuos en línea a fin de disminuir el riesgo de robo y manipulación de información confidencial (como números de tarjetas de crédito, nombres de usuario, contraseñas, correos electrónicos, etc.) por parte de hackers y ladrones de identidades. Básicamente, la capa SSL permite que dos partes tengan una conversación”privada. Para establecer esta conexión segura, se instala en un servidor web un certificado SSL (también llamado certificado digital”) que cumple dos funciones: Autenticar la identidad del sitio web, garantizando a los visitantes que no están en un sitio falso y cifrar la información transmitida. Openssl es una api que proporciona un entorno adecuado para encriptar los datos enviados a otra computadora dentro de una red y a su vez desencriptarlos adecuadamente por el receptor, evitando así, el acceso a la información por intrusos con la utilización de sniffer. El conjunto de herramientas OpenSSL es una característica de FreeBSD que ofrece una capa cifrada de transporte sobre la capa normal de comunicación, permitiendo la combinación con muchas aplicaciones y servicios de red. Uno de los usos más comunes de OpenSSL es ofrecer certificados para usar con aplicaciones de software. Estos certificados aseguran que las credenciales de la compañía o individuo son válidos y no son fraudulentos. Si el certificado en cuestión no ha sido verificado por uno de las diversas “autoridades certificadoras” o CA, suele generarse una advertencia al respecto. Una autoridad de certificados es una compañía, que firma certificados para validar credenciales de individuos o compañías. Este proceso tiene un costo asociado y no es un requisito imprescindible para usar certificados, aunque puede darle un poco de tranquilidad a los usuarios. [24]

Hardware.

En el ámbito del hardware, utilizaremos los equipos de cómputo del prototipo anterior los cuales se enlistan a continuación:

Equipo de hardware utilizado.	
Marca	Asus
Modelo	X550VC
Procesador	Intel Core i5
Tarjeta de video	NVidia GForce 720
Memoria RAM	12 GB
Disco duro	1TB

Equipo de hardware utilizado.	
Marca	HP
Modelo	Pavilion g4
Procesador	Intel Core i3
Tarjeta de video	Intel Sandybridge Mobile
Memoria RAM	6 GB
Disco duro	500GB

2.3. Estudio de requerimientos.

2.3.1. Requerimientos Funcionales.

PII_RF1. Inyección de código autenticador. Por medio del método Chaffing se crearán paquetes para agregar al encabezado HTTP que se ha interceptado gracias al prototipo 1. Al agregar dichos paquetes chaff se estará ocultando el mensaje original.

PII_RF2. Liberación de Petición. Se liberará el bloqueo a la petición HTTP impuesto por el prototipo I una vez terminado el proceso de chaffing, para que la petición pueda llegar a servidor correspondiente.

PII_RF3. Subida de código autenticador. La extensión podrá recuperar el archivo que contiene el código autenticador por medio del usuario, quien elegirá dicho archivo, por medio de una interfaz gráfica. Dicho código contiene la llave para que el servidor pueda validar al usuario.

PII_RF4. Inicio de sesión en la extensión. La extensión contará con una interfaz para el ingreso de datos del usuario, los cuales serán usuario y contraseña.

PII_RF5. Obtener el código autenticador. Para el prototipo 2 la obtención del código autenticador se hará del lado del cliente simulando la tarea de una entidad certificadora, permitiendo así generar automáticamente

y descargar esta llave la cual servirá para el "cifrado" del mensaje y la inserción de paquetes chaff en el encabezado de la petición HTTP.

2.3.2. Requerimientos no Funcionales.

PII_RNF1. Tamaño del código autenticador. El tamaño del archivo depende totalmente del tamaño de llave que se desea generar.

PII_RNF2. Carga de archivo en la extensión. Se necesita tener cargado el archivo en la extensión para el correcto funcionamiento de la función chaff.

2.4. Reglas del negocio.

PII_RN1. Petición válida. La extensión modificará la petición siempre y cuando se trate de una petición válida.

PII_RN2. Extensión habilitada. La modificación de la petición HTTP solo se podrá realizar mientras la extensión se encuentre habilitada.

PII_RN3. Inicio de sesión de extensión por usuario. Cada usuario que desee utilizar la extensión solo deberá tener una cuenta con un usuario y una contraseña respectiva a este usuario.

PII_RN4. Acceso a internet. Se debe de contar con acceso a internet para que la extensión pueda enviar al servidor la petición modificada.

Capítulo 4

Desarrollo.

1. Prototipo I.

1.1. Diagrama de casos de uso.

Diagrama de casos de uso general para el prototipo I.

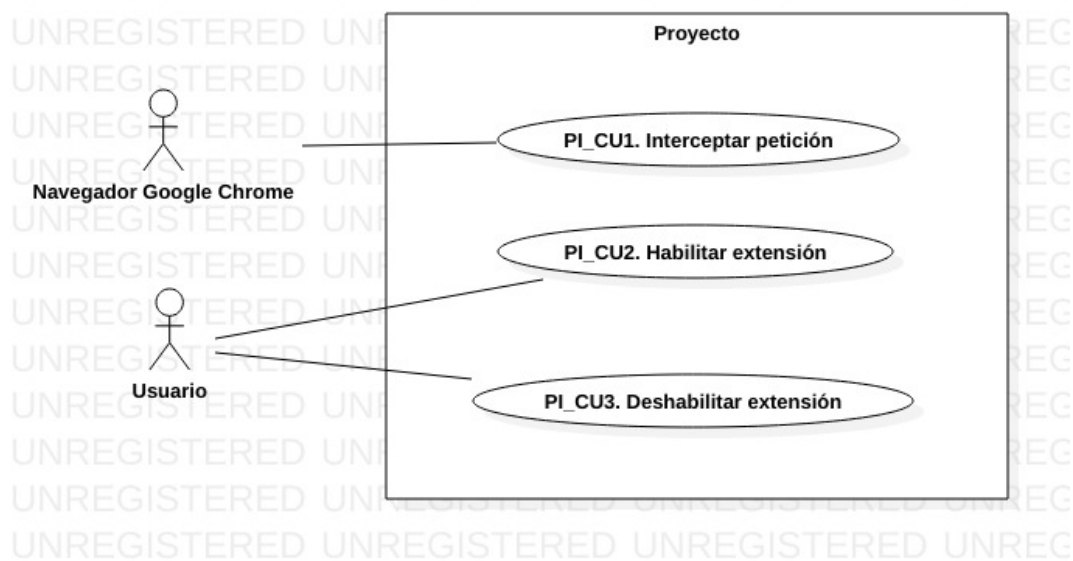


Figura 4.1: Diagrama de casos de uso del Prototipo I.

1.2. Descripción de casos de uso.

Caso de uso: PI_CU1. Interceptar petición.	
Concepto	Descripción
Actor	Navegador de Google Chrome.
Propósito	Este caso de uso permite a la extensión interceptar una petición HTTP, realizada por el navegador Google Chrome por medio de algún agente (sistema o usuario) externo a éste.
Entradas	Petición HTTP realizada por el navegador.
Salidas	Petición HTTP cachada.
Pre-condiciones	Algún agente externo (Sistema o usuario) ha ordenado al navegador mandar una petición HTTP.
Post-condiciones	La extensión, deberá de interceptar la petición para poder modificarla.
Reglas del negocio	-
Errores	La petición no se pudo interceptar. La petición no es tipo HTTP.

Cuadro 4.1: Descripción CU: PI_CU1

... Trayectoria Principal ...

1. ***El Usuario*** o ***El Sistema Externo*** realiza una petición HTTP en el navegador Google Chrome.
2. ***La Extensión*** intercepta la petición antes de que salga a red.
3. ***La Extensión*** debe modificar el contenido de la petición.

... Fin de la Trayectoria Principal ...

... Trayectoria Alternativa 1 ...

1. ***El Usuario*** o ***El Sistema Externo*** realiza una petición que no es HTTP en el navegador Google Chrome.
2. ***La Extensión*** ignora la petición.

... Fin de la Trayectoria Alternativa 1 ...

... Trayectoria Alternativa 2 ...



1. ***El Usuario*** o ***El Sistema Externo*** realiza una petición HTTP en el navegador Google Chrome.
2. ***La Extensión*** no puede interceptar la petición antes de que salga a red.
3. ***La Extensión*** notifica que hubo un error al intentar interceptar la petición.

... Fin de la Trayectoria Alternativa 2 ...

Caso de uso: PI_CU2. Habilitar extensión.	
Concepto	Descripción
Actor	Usuario.
Propósito	Este caso de uso, permite al usuario habilitar a la extensión, para que ésta sea capaz de ver todas las peticiones que realiza el navegador.
Entradas	Indicación de habilitar extensión, mediante interfaz de usuario.
Salidas	Ninguna.
Pre-condiciones	El usuario debe de haber instalado la extensión en Google Chrome y haber permitido su ejecución.
Post-condiciones	La extensión verá todas las peticiones que realice el navegador.
Reglas del negocio	PLRN1.
Errores	No se puede iniciar la vigilancia de la extensión.

Cuadro 4.2: Descripción CU: PI_CU2

... Trayectoria Principal ...

1. **El usuario** da click en el ícono de la extensión .
2. **El usuario** da click en el botón .
3. **La extensión** empieza a vigilar las peticiones que se realicen a través del navegador.

... Fin de la Trayectoria Principal ...

... Trayectoria Alternativa 1 ...

1. **La extensión** no muestra el botón , por ende el usuario no puede dar click.

... Fin de la Trayectoria Alternativa 1 ...

... Trayectoria Alternativa 2 ...

1. **El usuario** no encuentra el ícono de la extensión



... Fin de la Trayectoria Alternativa 2 ...


Caso de uso: PI_CU3. Deshabilitar extensión.	
Concepto	Descripción
Actor	Usuario.
Propósito	Este caso de uso, permite al usuario deshabilitar a la extensión, para que ésta ignore todas las peticiones que se realicen por medio del navegador.
Entradas	Indicación de deshabilitar extensión, mediante interfaz de usuario.
Salidas	Ninguna.
Pre-condiciones	El usuario debe de haber instalado la extensión en Google Chrome y haber permitido su ejecución.
Post-condiciones	La extensión dejará de ver todas las peticiones que realice el navegador.
Reglas del negocio	PLRN1.
Errores	No se puede detener la vigilancia de la aplicación.

Cuadro 4.3: Descripción CU: PLCU3

... Trayectoria Principal ...

1. **El usuario** da click en el ícono de la extensión



2. **El usuario** da click en el botón .

3. **La extensión** deja de vigilar las peticiones que se realicen a través del navegador.


... Fin de la Trayectoria Principal ...

... Trayectoria Alternativa 1 ...

1. *La extensión* no muestra el botón , por ende el usuario no puede dar click.

... Fin de la Trayectoria Alternativa 1 ...

... Trayectoria Alternativa 2 ...

1. *El usuario* no encuentra el ícono de la extensión  .

... Fin de la Trayectoria Alternativa 2 ...

1.3. Diagrama de flujo de datos (DFD).

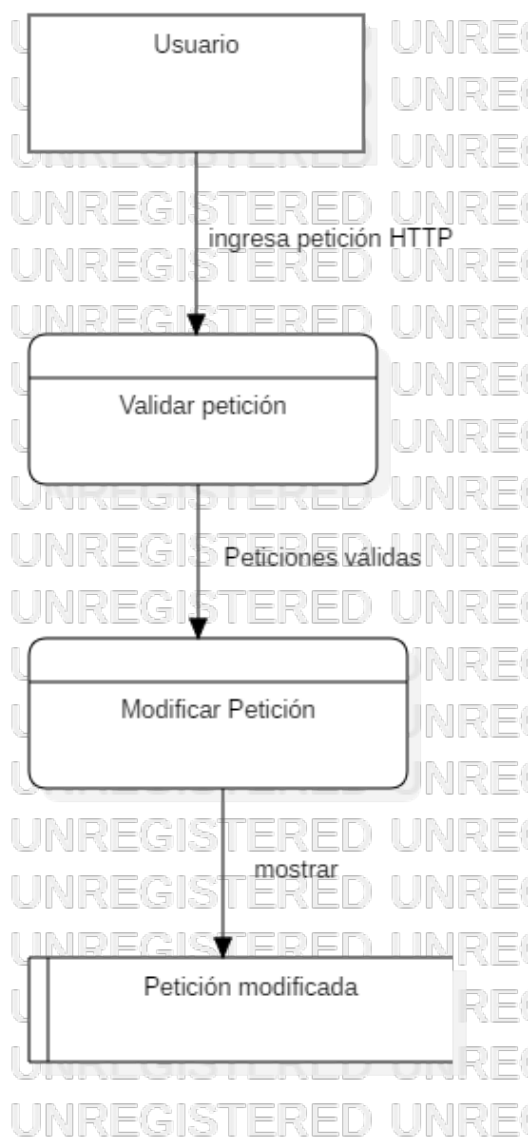


Figura 4.2: Diagrama de flujo de datos del Prototipo 1.

1.4. Diagrama de clases.

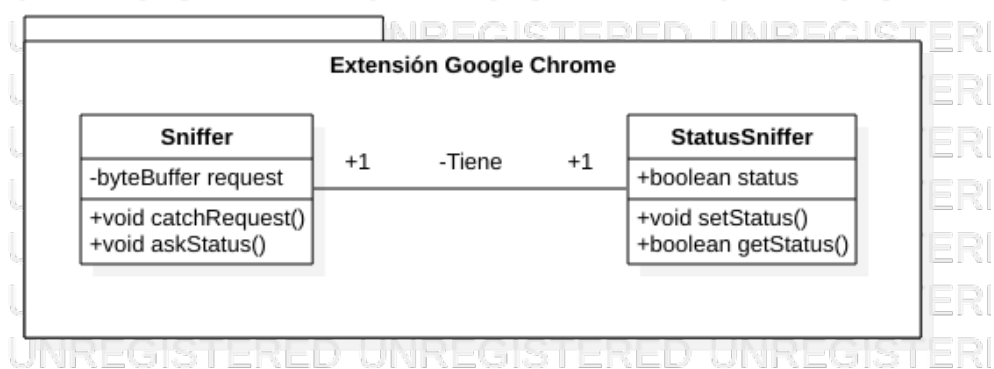


Figura 4.3: Diagrama de clases del Prototipo I.

1.5. Diagrama de secuencia.

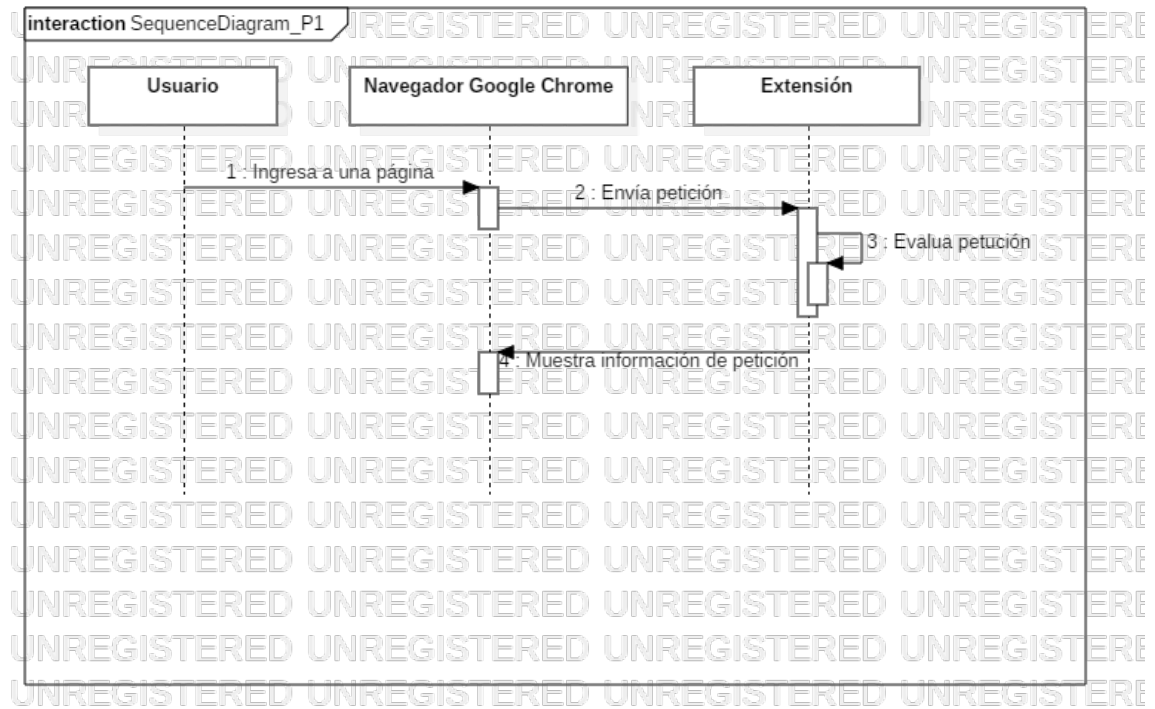


Figura 4.4: Diagrama de secuencia del Prototipo I.

1.6. Diagrama de actividades

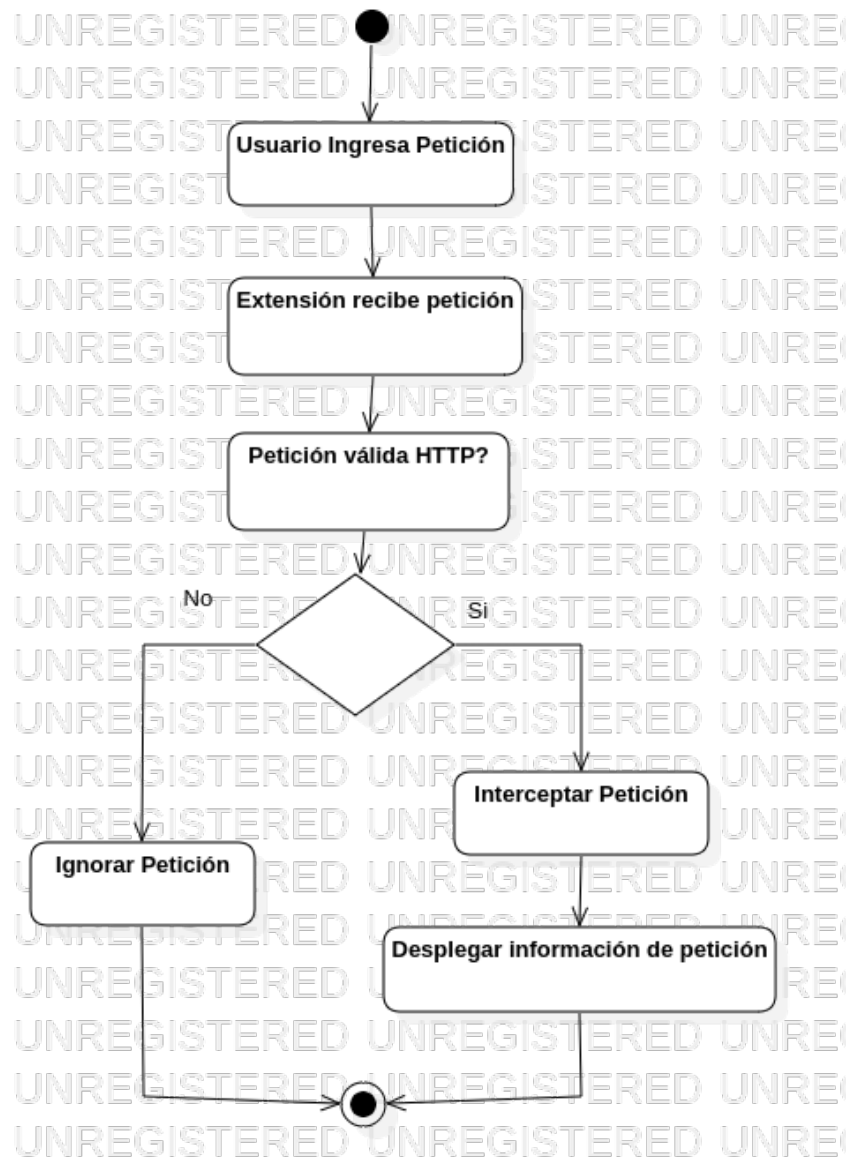


Figura 4.5: Diagrama de actividades del Prototipo I.

1.7. Interfaz de usuario.



Figura 4.6: Logo de la extensión.

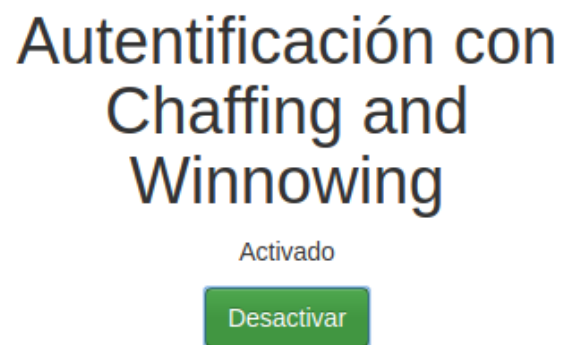


Figura 4.7: Pantalla inicial. Servicio activado.

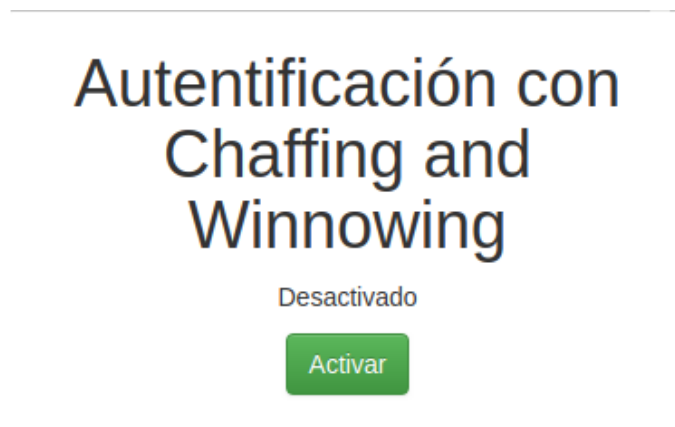


Figura 4.8: Pantalla inicial. Servicio desactivado.

1.8. Requisitos de diseño.

En este apartado, se especificarán las especificaciones de diseño para que el prototipo opere de forma correcta.

1.8.1. Requisitos de ejecución sobre Google Chrome.

Gracias a que Google Chrome es multiplataforma, esto es, funciona en varios sistemas operativos, el funcionamiento del Prototipo I depende exclusivamente de **tener instalado Google Chrome Stable o Google Chrome Developer** en la computadora local. Actualmente, este navegador para dispositivos móviles, no admite la instalación de extensiones, por lo que el funcionamiento depende también de tener una versión de '**Google Chrome Desktop**' instalada en el dispositivo.

Sin embargo, para el Prototipo I utilizamos dos API's, llamadas '*chrome.webRequest*' y '*chrome.storage*', las cuales están disponible sólo a partir de la **versión 28**, por lo que, es necesario tener una versión igual o superior de este software.

Por otro lado, es **necesario que se permita que la extensión se ejecute sobre Google Chrome con ciertos permisos**, los cuales se exponen a continuación.

- Habilitar extensión: activado
- Acceso al sitio web: en todos los sitios
- Permitir acceso a URL de archivo: activado

Bibliografía

- [1] Antonina Komarova, Alexander Menshchikov, “Comparison of Authentication Methods on Web Resources”, St. Petersburg National Research University of Information Technologies, St. Petersburg, Russia, 2016.
- [2] Cryptology, Albrecht BeutelSpacher
- [3] Seguridad informática y criptografía
- [4] Menezes, Van Oorschot y Vanston, Handbook Of Applied Cryptography
- [5] Tesis de la profesora sandra
- [6] Maiorano, Ariel. Criptografía, Técnicas de desarrollo para profesionales.
- [7] <https://www.dtic.upf.edu/tnavarrete/fcsig/javascript.pdf>
- [8] <https://gutl.jovenclub.cu/wp-content/uploads/2013/10/El+gran+libro+de+HTML5+CSS3>
- [9] <https://www.seguridad.unam.mx/historico/documento/index.html-id=17>
- [10] <https://www.seguridad.unam.mx/historico/documento/index.html-id=16?fbclid=IwAR0u8WAXORvBxZ3H-aMzlBhd-6o7g8ycS88eRu7nY1t1XVtCufhEcQ7hWDs>
- [11] Aguilar, A. and Hernández, A. (25 de Abril de 2014). Obtenido de Sugerencias de Seguridad para Sitios Web: <http://www.seguridad.unam.mx/documento-id=1143>
- [12] [https://es.wikipedia.org/wiki/Cookie_\(informatica\)](https://es.wikipedia.org/wiki/Cookie_(informatica))
- [13] <http://www.allaboutcookies.org/es/galletas/cookies-persistentes-utilizados-para.html>
- [14] <http://www.gadae.com/blog/que-son-las-cookies-tipos-de-cookies-y-como-cumplir-la-ley/>

- [15] <https://www.osi.es/es/actualidad/blog/2018/07/18/entre-cookies-y-privacidad>
- [16] http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica_mod
- [17] <http://www.cs.bath.ac.uk/~mdv/courses/CM30082/projects.bho/2007-8/durongdej-r-dissertation-2007-8.pdf>
- [18] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.160.4853&rep=rep1&type=pdf>
- [19] S. GOLDWASSER AND S. MICALI, “Probabilistic encryption,” *Journal of Computer and System Science*, Vol. 28, 1984, pp. 270–299.
- [20] <https://people.csail.mit.edu/rivest/Chaffing.txt>
- [21] © Springer International Publishing AG 2018 A. Abraham et al. (eds.), *Proceedings of the Second International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’17)*, *Advances in Intelligent Systems and Computing* 679, DOI 10.1007/978-3-319-68321-8_11
- [22] <https://developer.mozilla.org/es/docs/Web/API/FileSystem>
- [23] https://www.verisign.com/es_LA/website-presence/online/ssl-certificates/index.xhtml
- [24] <https://www.globalsign.com/es/centro-de-informacion-ssl/que-es-ssl/>