UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN – LEÓN

Facultad De Ciencias y Tecnología

Departamento de Computación

Tema: Evaluación de herramientas para la protección de las comunicaciones en redes sociales y dispositivos móviles.

Monografía para optar al título de

INGENIERO EN TELEMÁTICA

Presentado por:

Br. Yorleni Patricia Mejía Barrera.

Br. Eymi Juniett Peralta Palacios.

Tutor:

Msc. Aldo Rene Martínez Delgadillo.

León, Octubre 2014

Agradecimiento

Primeramente le agradezco a Dios Señor Nuestro por haberme dado vida, paciencia y salud, también por haberme, iluminado y darme la fortaleza para continuar cuando más lo necesitaba, siendo el una fuente de inspiración en los momentos más difíciles.

A mi querida Madre por apoyarme siempre y estar ahí paso a paso durante toda mi carrera profesional y durante toda mi vida desde que nací hasta hoy en día y le agradezco porque siempre me ha dado ánimos de seguir adelante.

A mi Padre por ser un ejemplo para mí, por ayudarme y apoyarme cuando lo he necesitado, por sus consejos acertados en todo momento.

A mi tutor Msc. Aldo Rene Martínez Delgadillo, fuente de inspiración y de sabiduría, gracias por compartir conmigo sus conocimientos por haber confiado en mí.

Br: Yorleni Mejía

Agradecimiento

Primeramente le agradezco a Dios, ser maravilloso que me dio fuerza y fe para creer lo que me parecía imposible terminar.

A mi madre Martha Palacios, mi tía Esmelda Palacios, y familiares por su ayuda incondicional, dedicación y confianza en mí.

A mi tutor Msc. Aldo Rene Martínez Delgadillo por su esfuerzo, dedicación, compromiso, su guía, su persistencia y paciencia.

También expresar mi agradecimiento a todo el personal bibliotecario que me brindaron las puertas de la biblioteca para obtener información.

Br: Eymi Peralta

Dedicatoria

En primer lugar dedico a Dios, por haberme permitido terminar con éxitos mis estudios y poder hacer realidad mi sueño.

A mis padres por la educación que me han brindado, por su ejemplo de luchar en la vida y conseguir las cosas, porque siempre han estado pendientes de mí.

A mis amistades por estar ahí cuando los necesitaba.
A nuestros maestros, por habernos transmitirnos la sabiduría y el conocimiento para triunfar en la vida.

Br: Yorleni Mejía

Dedicatoria

Le dedico primeramente este trabajo a mi Dios por estar conmigo a lo largo de mi vida, por darme esas fuerzas en los momentos que más lo necesitaba.

Con mucho amor a mi madre por haberme brindado todas las herramientas necesarias para enfrentarme a la vida, por su apoyo incondicional y fortaleza a lo largo de este camino.

A mis familiares, por apoyarme siempre.

A mis amistades por estar ahí cuando los necesitaba.

A mis maestros, y principalmente a mi tutor Msc. Aldo Rene Martínez, por transmitirme sus conocimientos para triunfar como profesional.

Br: Eymi Peralta

CONTENIDO

ÍNDICE DE FIGURA

ÍNDICE DE TABLA

I. RESUMEN

Las redes sociales constituyen uno de los principales medios de comunicación que utilizamos en la actualidad. Esto nos presenta un nuevo escenario para la participación ciudadana y para la definición de un modelo de relación entre ciudadanos y medios de comunicación, pero para operar en este medio resulta fundamental tener en cuenta las normas vigentes en materia de protección de datos.

Debido a esta situación, las personas que usan las redes sociales, accediendo a través de sus computadoras o de sus dispositivos móviles, exponen su información a posibles atacantes que emplean sofisticadas técnicas estadísticas para rastrear patrones de comunicación de organizaciones e individuos.

La solución es el uso de herramientas que nos brindan la protección en las comunicaciones, permitiendo el anonimato y cifrado de la información.

.

II. ANTECEDENTES

El presente trabajo monográfico tiene como antecedente un trabajo de tesis de maestría titulado: **"DISEÑO Y DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA EL ANALISIS DE REDES SOCIALES EN TWITTER"** elaborado por el Ing. Wilmer Moisés Matamoros Cáceres. El documento fue desarrollado con la necesidad de estudiar y enfocar la teoría de grafos en un ámbito social, para conocer y dar a conocer como la teoría de redes permite analizar y a su vez plantear soluciones. El trabajo fue dirigido por el Dr. David Fernández Barrero, para el año 2011, en la Universidad de Alcalá de Henares Escuela Politécnica.

Teniendo como punto de referencia el trabajo antes mencionado, desarrollamos un documento donde se expone la seguridad en las redes sociales y en dispositivos móviles adaptándolo y enfocándolo a la carrera de Ingeniería en Telemática de la UNAN-LEÓN.

Es válido mencionar que existen sitios web con información relacionado con este tema; sin embargo, no existe un documento completo por parte de la UNAN-LEÓN como el que se desarrolla en este documento.

III. PLANTEAMIENTO DEL PROBLEMA

Debido al gran crecimiento y popularidad que han tenido las redes sociales, hoy en día se han convertido en las herramientas más usadas por las personas. Todos estos avances han permitido que en nuestros días existan multitudes de procedimiento y tecnologías para proteger las comunicaciones de posibles interceptaciones y manipulaciones. Por desgracia el público en general no sabe cómo proteger su información.

Estas redes sociales se interconectan entre ellas y permite a las personas estar conectadas y en comunicación con familiares y amigos en todo el mundo, permitiendo exponer la información que usted provee en sus páginas de perfil. Sin embargo, la notoriedad de estos espacios sociales no queda exenta de peligros o posibles ataques malintencionados.

Teniendo en cuentas estas problemática, surge el nacimiento de aplicaciones que provean privacidad y seguridad al navegar por estos sitios, donde lo más conveniente sería navegar de forma anónima y usando cifrado en nuestras comunicaciones tanto en redes sociales como en dispositivos móviles.

IV. JUSTIFICACIÓN

Dado al auge que actualmente han cobrado las comunicaciones y específicamente la comunicación en las redes sociales, se considera necesario dotar a las personas con herramientas que les permitan cierto nivel de seguridad al navegar por estas redes, accediendo desde sus computadoras o dispositivos móviles.

4.1. ORIGINALIDAD

Existen documentos que contienen información acerca de las redes sociales; Sin embargo ninguno de ellos se adecúa a la protección en la comunicación, por lo que este trabajo será de gran ayuda al público en general que desee consultar a este documento.

4.2. ALCANCE

Proporcionar información sobre las herramientas y aplicaciones que existen actualmente, relacionadas con la privacidad en redes sociales, donde la comunicación y la transferencia de datos sean de forma eficiente y segura.

4.3. PRODUCTO

El documento presenta las características siguientes:

- Guiado: Describirá paso a paso la forma en la que se deberán instalar y configurar cada herramienta propuesta.

- Ilustrado: Explicará de forma breve y de manera sencilla el funcionamiento de las herramientas seleccionadas, las cuales son usadas para proteger las comunicaciones en redes sociales y dispositivos móviles.

4.4. IMPACTO

Concientizar a los usuarios de cómo proteger su información publicada en la red, poniendo a su disposición herramientas tecnológicas fáciles de usar.

V. OBJETIVOS

5.1. OBJETIVO GENERAL

Evaluar un conjunto de herramientas usadas para la privacidad y protección de las comunicaciones en redes sociales y en dispositivos móviles.

5.2. OBJETIVOS ESPECIFICOS

- Instalar y configurar un conjunto de herramientas actuales que permiten cifrar las comunicaciones en las redes sociales y dispositivos móviles.

- Comparar las herramientas de acuerdo a las características propias de cada aplicación, haciendo énfasis en el nivel de protección, privacidad al navegar y enviar mensajes, esto para tener la garantía mínima de confidencialidad, integridad y autenticidad en las comunicaciones.

- Elaborar un documento guiado con detalles del uso e instalación de las herramientas, con el objetivo de que pueda ser usado por usuarios finales, y con esto crear una cultura de protección en las comunicaciones.

VI. RESULTADOS ESPERADOS

- Instalado y configurado un conjunto de herramientas que permiten a los usuarios cifrar las comunicaciones, tanto en redes sociales como en dispositivos móviles.

- Realizado un estudio comparativo de las herramientas de acuerdo a las características propias de cada aplicación, haciendo énfasis en el nivel de protección, privacidad al navegar y enviar mensajes.

- Elaborado un documento guiado con detalles del uso e instalación de las herramientas, el cual sirve como referencia a usuarios que requieran mantener su información segura.

VII. MARCO REFERENCIAL

7.1. TEORÍA DE LOS SEIS GRADO

La teoría de los seis grados de separación es una hipótesis que intenta probar que cualquiera en la tierra puede estar conectado a cualquier otra persona del planeta a través de una cadena de conocidos que no tiene más de cinco intermediarios (conectando a ambas personas con sólo seis enlaces), algo que se ve representado en la popular frase el mundo es un pañuelo. La teoría fue inicialmente propuesta en 1930 por el escritor húngaro Frigyes Karinthy en un cuento llamado *Chains*. El concepto está basado en la idea de que el número de conocidos crece exponencialmente con el número de enlaces en la cadena, y sólo un pequeño número de enlaces son necesarios para que el conjunto de conocidos se convierta en la población humana entera. Recogida también en el libro "Six Degrees: The Science of a Connected Age" del sociólogo Duncan Watts, y que asegura que es posible acceder a cualquier persona del planeta en tan sólo seis "saltos".

Según esta teoría, cada persona conoce de media, entre amigos, familiares y compañeros de trabajo o escuela, a unas 100 personas. Si cada uno de esos amigos o conocidos cercanos se relaciona con otras 100 personas, cualquier individuo puede pasar un recado a 10.000 personas más tan sólo pidiendo a un amigo que pase el mensaje a sus amigos.

Estos 10.000 individuos serían contactos de segundo nivel, que un individuo no conoce pero que puede conocer fácilmente pidiendo a sus amigos y familiares que se los presenten, y a los que se suele recurrir para ocupar un puesto de trabajo o realizar una compra. Cuando preguntamos a alguien, por ejemplo, si conoce una secretaria interesada en trabajar estamos tirando de estas redes sociales informales que hacen funcionar nuestra sociedad. Este argumento supone que los 100 amigos de cada persona no son amigos comunes. En la práctica, esto significa que el número de contactos de segundo nivel será sustancialmente menor a 10.000 debido a que es muy usual tener amigos comunes en las redes sociales.

Si esos 10.000 conocen a otros 100, la red ya se ampliaría a 1.000.000 de personas conectadas en un tercer nivel, a 100.000.000 en un cuarto nivel, a 10.000.000.000 en un quinto nivel y a 1.000.000.000.000 en un sexto nivel. En seis pasos, y con las tecnologías disponibles, se podría enviar un mensaje a cualquier individuo del planeta. Por ejemplo, imaginemos un limpiabotas de la calle. Este limpiabotas conoce a un portero de un hotel de dos estrellas; dicho portero conoce al dueño del hotel y éste al dueño de un hotel más prestigioso; el dueño de este hotel conoce a una persona que trabaja en la Casa Blanca y esta persona conoce al presidente de los Estados Unidos. En unos pocos enlaces se ha conseguido ligar un limpiabotas con el presidente de los Estados Unidos [1].

7.2. REDES SOCIALES

Las redes sociales son un fenómeno relativamente reciente. Los primeros experimentos datan de 1995, pero no fue hasta el 2003 cuando comenzaron a popularizarse los puntos de encuentro con la aparición de pesos pesados como MySpace. Actualmente, hay dos redes sociales que lideran el ámbito internacional: La ya mencionada MySpace y Facebook [2]. Las Redes Sociales son formas de interacción social, definida como un intercambio dinámico entre personas, grupos e instituciones en contextos de complejidad. Un sistema abierto y en construcción permanente que involucra a conjuntos que se identifican en las mismas necesidades y problemáticas y que se organizan para potenciar sus recursos [3].

El modelo de crecimiento de estas plataformas se basa fundamentalmente en un proceso exponencial, en el que un número inicial de participantes, mediante el envío de invitaciones a través de correos a sus conocidos, ofrece la posibilidad de unirse al sitio web [4].Las redes sociales se han convertido en bases de datos en la que se recogen informaciones de carácter personal y documentos sobre las actividades de la vida real de las personas que hacen uso de ellas. Campos como el estado civil y la ciudad de residencia son habituales en este tipo de páginas y mucha gente cree que rellenando este tipo de campos en nuestro perfil, aun cuando no son obligatorios para el registro, van a ser usuarios más activos, pero la realidad es que estamos poniendo al alcance de cualquiera, datos que no hace mucho tiempo evitábamos dar con tanta facilidad. Nuestra información personal, y más aún este tipo de datos sobre nuestra vida privada deberían estar más valorados, ya que exponerlos con tanta facilidad podría traernos problemas [5]. Son sitios web que ofrecen servicios y funcionalidades de comunicación diversos para mantener en contacto a los usuarios de la red. Se basan en software especial que integra numerosas funciones individuales: blogs, wikis, foros, chat, mensajería, etc. en una misma interfaz y que proporciona la conectividad entre los diversos usuarios de la red [6].

Existen muchos tipos clasificadas según su propósito y ámbito. Sin embargo, podemos hablar de tres grandes categorías:

- *Redes personales.* Se componen de cientos o miles de usuarios en los que cada uno tiene su pequeño “espacio” con su información, sus fotos, su música, etc. Y cada uno se puede relacionar con los demás de múltiples maneras, aunque todas ellas involucran el uso de Internet de una u otra forma. Facebook es una red personal.
- *Redes temáticas.* Son similares a las anteriores aunque se diferencian por el hecho de que suelen centrarse en un tema en concreto y proporcionan las funcionalidades necesarias para el mismo. Por ejemplo, una red de cine, una de informática, de algún tipo de deporte, etc.
- *Redes profesionales.* Son una variedad especial de las anteriores, dedicadas exclusivamente al ámbito laboral, en todas sus vertientes [7].

7.3. REDES SOCIALES MAS USADAS:

Facebook: La red social más popular con decenas de funciones y una extensísima colección de aplicaciones personalizadas, pionera en su campo y muy sencilla de usar. Nació en un campus universitario Harvard en el año 2004 con la finalidad de unir a alumnos y profesores y se extendió rápidamente hasta su uso abierto a todo el mundo en el 2006 con la única condición de tener más de 13 años y disponer de una cuenta e-mail válida. Actualmente tiene más de 200 millones de usuarios [7].

Twitter: Entorno de comunicación gratuito de microblogging que permite a sus usuarios enviar micro-entradas basadas en texto [7].

LinkedIn: Es un sitio web orientado a negocios, fue fundado en diciembre de 2002 y lanzado en mayo de 2003(comparable a un servicio de red social), principalmente para red profesional[8].

Google+ (pronunciado y a veces escrito Google Plus, a veces abreviado como G+: En algunos países de lengua hispana pronunciado Google Más) es un servicio de red social operado por Google Inc. El servicio, lanzado el 28 de junio de 2011, está basado en HTML5. Los usuarios tienen que ser mayores de 13 años de edad, para crear sus propias cuentas. Google+ ya es la segunda red social más popular del mundo con aproximadamente 343 millones de usuarios activos [10].

LiveJournal (a menudo abreviado como LJ): Es el nombre de un sitio de web log que permite a los internautas mantener un periódico o diario en línea. También es el nombre del software de servidor de código abierto que fue diseñado para ejecutarlo [11].

Myspace (escrito antes por la empresa como MySpace) es un servicio de red social propiedad de Specific Media LLC y la estrella de pop Justin Timberlake. Myspace fue lanzado en agosto del 2003 y su base se encuentra en Beverly Hills, California. En agosto de 2011, Myspace contaba con 33.1 millones de visitantes en EEUU.

7.4. VENTAJAS DE LAS REDES SOCIALES

Las redes sociales evolucionan cada vez más rápido, y nosotros como jóvenes lo hacemos con ellas. Para los estudiantes puede ser una ventaja, ya que es una forma muy fácil y rápida de comunicarse, sin importar la distancia en la que se encuentre la otra persona, permite el intercambio de información, para que nuestros compañeros puedan ver nuestros proyectos. Tiene un enorme atractivo en el ámbito educativo, cuanto mayor sea el número de los participantes más atracción genera en los alumnos al poder estar en contacto directo con sus profesores y compañeros de su curso y de otros cursos a los que quizás conozcan de vista pero con los que no ha hablado nunca. Esto permite crear un ambiente de trabajo favorable que es uno de los motivos directos del éxito de las redes sociales, además de fomentar la creatividad [15].

7.5. MODELOS

Como infraestructuras lógicas de ordenamiento de información, presentan una serie de estrategias para la disposición de sus servicios. Dichas estrategias son desplegadas en diferentes niveles y la finalidad de las mismas en su conjunto es la acumulación de "valores".

El análisis y correcta interpretación puede ser bastante útil para poder determinar, desde un punto de vista basado en la necesidad de la libertad de información como forma de progreso humano, la "legitimidad" del servicio que proveen ciertas compañías y/o agrupaciones de personas.

Las estrategias técnicas, aparte de por los equipos humanos que desarrollan el código que genera la parte lógica del aplicativo, pasan por el uso de un modelo concreto para la distribución y el correcto almacenaje de los datos. Podríamos incluir más modelos, como el modelo de protección de la información, el modelo de balanceo de tráfico u otros más, como modelos laborales, modelos de desarrollo de código, etc., pero nos centraremos en explicar los que podemos considerar como pilares básicos para las redes sociales: los modelos de distribución y almacenaje.

7.6. TIPOS DE MODELOS

Modelo centralizado: dicho modelo centraliza los datos en uno o varios puntos con un único gestor y propietario.

Modelo distribuido (P2P): dicho modelo descentraliza los datos en varios puntos y no permite la propiedad o gestión única.

Modelo en "nube" (Cloud): dicho modelo centraliza los datos en uno o varios puntos y el propietario puede ser uno y el gestor puede ser otro. Ambos únicos en su tarea.

Modelo federado: dicho modelo descentraliza los datos en varios puntos y permite diferenciar entre propietarios y gestores, pero sin permitir que sean únicos, si se desea.

7.7. API

Las denominadas Apis en redes sociales son una serie de comandos de conexión, herramientas y funciones, cuyo diseño permite a cualquier usuario que tenga el conocimiento de cómo usarlas, interactuar con procesos internos y externos, así como con la información que se genera, según el nivel de desarrollo de la misma. Se trata de una forma eficiente de ahorro de tiempo y reducción de costes para las redes sociales. Una API, por tanto, es el software que permite interconectar redes, u otras plataformas, entre sí.

Generalmente se utilizan en los procesos de desarrollo de código como fuente de recursos y establecen las normas de conectividad del entorno. Esta última parte es bastante interesante, ya que según el grado de permisividad y otros factores, es posible conocer en buena medida las intenciones tanto técnicas como las que no, de quienes están detrás de las mismas.

Por poner un ejemplo, la red social Twitter permite a sitios web conectarse de varias formas a distintos puntos de su aplicación, como son los mensajes, las actualizaciones, etc. Y para ello han creado un "framework" que permite interconectar y realizar pruebas. También incluyen los procesos de autorización en el manejo de prioridades y de determinadas funcionalidades, así como varias limitaciones [4]. Con todo esto, es posible desarrollar herramientas alrededor de su API y complementar su uso con nuevas funcionalidades.

Por ejemplo, los permisos que tiene la API de Facebook sobre una aplicación móvil se suelen definir en el LBE Privacy Guard. Es curioso que a través de los nombres podemos obtener algo más de información sobre las intenciones que tiene la empresa sobre los datos y el nivel de control sobre el dispositivo que demanda. Por ejemplo, para una aplicación cliente de navegación (Play+) en un móvil, se formulan los siguientes puntos de control:

- Tus cuentas
- Crear cuentas y establecer contraseñas
- Añadir o eliminar cuentas
- Controles de hardware
- Realizar fotografías y vídeos
- Tu ubicación
- Ubicación aproximada (basada en red)
- Ubicación precisa (basada en red y gps)
- Comunicación de red
- Acceso completo a red

- Tu información personal
- Consultar tus contactos
- Modificar tus contactos
- Llamadas de teléfono
- Consultar la identidad y el estado del teléfono
- Almacenamiento
- Modificar tus contactos
- Llamadas de teléfono
- Editar o borrar contenido de usb

Por otro lado, los términos de servicio constituyen un acuerdo legal entre el demandante de servicios y el proveedor de los mismos. En ellos debe quedar bien claro el entendimiento y aceptación del acuerdo de uso y el acuerdo legal y su conformidad, con respecto a la legislación vigente del país seleccionado para albergar la información por parte del proveedor de servicios.

7.8. APIS MÁS USADAS

La API de YouTube nos permite modificar el aspecto y las funciones del reproductor de videos. También es posible crear nuevas aplicaciones que permitan generar, buscar o clasificar videos en YouTube a través de una interfaz propia, en la cual podamos incluir publicidad u otro tipo de contenido. La API brinda funciones para obtener información sobre usuarios y videos.

Google Maps permite insertar mapas en nuestro sitio web. Pero la opción más interesante es la de modificar los mapas agregando información (como puntos geográficos o recorridos de interés), gráficos y videos propios, además de funciones interactivas.

Twitter autoriza la integración de la plataforma con sitios web pero también nos permite crear aplicaciones que procesen información sobre el timeline, el flujo de tweets en tiempo real, los mensajes directos, las listas de favoritos y muchos otros aspectos.

Por último, la *API de Facebook* brinda grandes herramientas para crear juegos y aplicaciones de concursos, como ya hemos detallado en un post reciente[21].

7.9. MINERÍA DE DATOS

La minería de datos es un proceso computacional de búsqueda y selección de patrones en grandes volúmenes de conjuntos de datos que a través del "aprendizaje automático" (IA), la estadística y los sistemas relacionales de bases de datos; trata de extraer la información más relevante en consonancia con lo que se desea obtener [16].

El principal objetivo de la minería de datos es formar estructuras comprensibles de fácil uso y análisis posterior. Un proceso de minería de datos consta de los siguientes pasos generales:

1. Selección del conjunto de datos.
2. Análisis de las propiedades de los datos.
3. Transformación del conjunto de datos de entrada.
4. Selección y aplicación de la técnica de minería de datos.
5. Extracción del conocimiento requerido.
6. Interpretación y evaluación de datos.

7.10. AUTENTICACIÓN DE MENSAJES

Autenticación de mensajes: procedimiento para verificar que un mensaje recibido procede de la fuente a llegada y no ha sido alterado.

Autenticador: el valor usado para autenticar un mensaje en un procedimiento de autenticación.

Procedimientos que se pueden utilizar para generar un autenticador:

- *Mensaje cifrado:* un mensaje cifrado con sistema simétrico o asimétrico sirve como autenticador de dicho mensaje
- *Código de autenticación de mensaje (MAC):* un valor de longitud fija producido a partir de una función pública y una clave secreta.
- *Función Hash:* función pública que genera un valor de longitud fija y único a partir de un mensaje de entrada de cualquier longitud.
- *Firma digital:* utiliza función hash y cifrado asimétrico, ofrece autenticación, integridad y no repudio.

Ejemplo:

Si A y B comparten una clave privada, los mensajes que se envían cifrados están autenticados, sólo el otro lo puede haber cifrado.

- El problema que se plantea es saber si un mensaje recibido es correcto una vez descifrado.
- Si el mensaje en claro es texto podemos analizar si tiene sentido.
- Si el mensaje es una secuencia de bits sin sentido no podemos saberlo.
- Una solución al problema consiste en forzar al mensaje a tener una estructura fácil de reconocer pero que no pueda ser replicada sin la clave.

Ejemplo:

- A añade al mensaje M una secuencia de chequeo de error $F\ (M)$ antes de cifrar. Cifra todo y envía a B el mensaje $EK\ [M||F\ (M)]$.
- B descifra $DK\ [M||F\ (M)]$, genera $F\ (M)$ y lo compara con el $F\ (M)$ recibido.
- Si el chequeo de error fuese externo, es decir se enviase $EK[M]||F(EK\ [M])$, un oponente podría encontrar otra secuencia de bits que diese la misma función de error, cambiarla por $EK[M]$ y crear de esta forma Autenticación
- Con la clave privada del origen y descifrar con la clave pública de origen.
- Así el destino tiene la seguridad de que sólo el origen pudo haber cifrado.
- El mensaje en claro debe tener una estructura que permita al receptor distinguir entre un mensaje correcto y bits aleatorios.
- Se pueden utilizar los mismos ejemplos de estructura del mensaje que hemos visto en sistemas simétricos.
- No es eficiente si el mensaje es largo.
- Añade el matiz del no repudio (al ser la clave secreta no compartida, el receptor no puede falsificar un mensaje para hacer creer que otro usuario se lo había enviado).
- Este mecanismo no ofrece confidencialidad, para ofrecer debemos cifrar con la clave privada de origen (autenticación) y después con la clave pública de destino (confidencialidad).
- El problema de la confidencialidad y autenticación con clave pública es que hay que aplicar el algoritmo de cifrado y descifrado 4 veces en cada comunicación.

- Los algoritmos asimétricos son más lentos que los simétricos.[43]

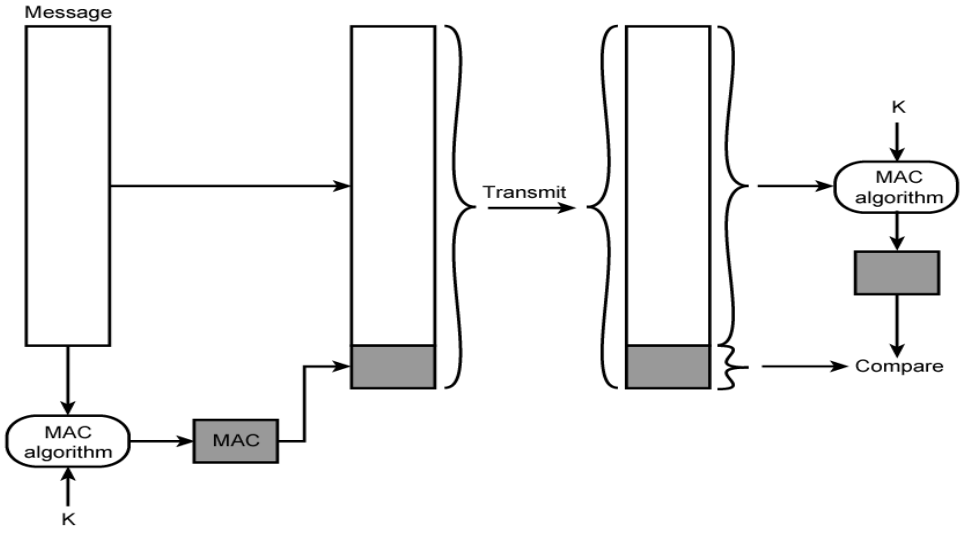


Figura 1: Autenticación de Mensajes mediante el código de autenticación de mensajes (MAC)

1. El receptor está seguro de que el mensaje no ha sido alterado. Sin un oponente altera el mensaje pero no Altera el código, el cálculo del código del receptor será distinto al del código recibido. Como se supone no conoce la clave secreta, no puede alterar el código para hacerlo corresponder con las alteraciones realizadas en el mensaje.
2. El receptor está seguro de que el mensaje es del emisor indicado. Como nadie más conoce la clave, nadie más podría crear un mensaje con un código adecuado.
3. Si el mensaje incluye un numero de secuencia (como el que se usa con X.25, HDLC Y TCP), el receptor puede estar seguro de la secuencia apropiada porque un oponente no puede alterar con éxito el número de secuencia [42].

7.11. CRIPTOGRAFÍA

Desde el principio de la historia intercambiar mensajes cifrados ha jugado un papel destacado. Tanto en la milicia, diplomacia y el espionaje, constituyen la mejor defensa de las comunicaciones y datos que se transmiten, por cualquier canal. Esto es debido a la necesidad de que algunos de los mensajes solo sean conocidos por aquellas personas a las que van dirigidos y no puedan ser interpretados por nadie más que por ellos [42].

En la era de la información como algunos llaman a la época en la que vivimos como se puede pensar la protección de la información es uno de los retos más fascinantes de la informática del futuro. Por poner sólo un ejemplo sencillo y común, un problema de gran actualidad es el asociado con el correo electrónico que se transmite a través de redes y cuyo nivel seguridad deja mucho que desear. Internet es un claro ejemplo de estas amenazas en tanto es un entorno abierto en su sentido más amplio. Por lo visto en estos pocos años de existencia de la llamada red de redes, sobran los comentarios acerca de pérdida de privacidad, accesos no autorizados, ataques y otros delitos informáticos a nivel nacional e internacional.

Ante tales amenazas, la única solución consiste en proteger nuestros datos mediante el uso de técnicas criptográficas. Esto nos permitirá asegurar al menos dos elementos básicos de la seguridad informática, a saber la confidencialidad o secreto de la información y la integridad del mensaje, además de la autenticidad del emisor.

La criptografía es la 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.

Un término más genérico es cristología: el compendio de las técnicas de cifra, conocido como criptografía, y aquellas técnicas de ataque conocidas como criptoanálisis. El hecho de codificar un mensaje para que sea secreto se llama cifrado.

El método inverso, que consiste en recuperar el mensaje original, se llama descifrado. Esto dará lugar a diferentes tipos de sistemas de cifra, denominados criptosistemas, que nos permiten asegurar al menos tres de los cuatro aspectos básicos de la seguridad informática: la confidencialidad o secreto del mensaje, la integridad del mensaje y autenticidad del emisor, así como el no repudio mutuo entre emisor (cliente) y receptor (servidor).

- *Confidencialidad* garantiza que la información sea accesible únicamente a la persona u organismo autorizado a acceder a dicha información.
- *Integridad* asegura que el mensaje no haya sido manipulado.
- *Autenticidad* garantiza que el comunicante que envía el mensaje no es un suplantador.
- *No repudio*, el receptor del mensaje no pueda negar que lo ha recibido

El objetivo de la criptografía es el de proporcionar comunicaciones seguras (y secretas) sobre canales inseguros. Ahora bien, la criptografía no es sinónimo de seguridad. No es más que una herramienta que es utilizada de forma integrada por mecanismos de complejidad variable para proporcionar no solamente servicios de seguridad, sino también de confidencialidad.

La criptografía se usa no sólo para proteger la confidencialidad de los datos, sino también para garantizar su integridad y autenticidad a continuación se mencionan términos de la criptografía:

- *Cifrado:* Proceso de convertir un mensaje inteligible o mensaje en claro, por otro que no pueda ser interpretado, o mensaje cifrado.
- *Descifrado:* Es el procedimiento que invierte el cifrado, es decir, desde un mensaje cifrado recupera el mensaje en claro.
- *Clave:* Es el conjunto de datos necesarios para cifrar o descifrar un mensaje. Aquel que no disponga de la clave no podrá cifrar ni interpretar un mensaje cifrado.

Generalmente. Los sistemas criptográficos se clasifican atendiendo a tres factores independientes:

1. *El tipo de operación utilizado para transformar el texto claro en texto cifrado.* Todos los algoritmos de cifrado se basan en dos principios generales: sustitución, donde cada elemento del texto claro (bit, letra, grupo de bits o letras) se sustituyen por otro diferentes; y transposición, donde los elementos del texto claro se reordenan.
2. *El número de claves usadas.* Si tanto el emisor como el receptor usan la misma clave, el sistema se denomina simétrico, de clave única, de clave secreta, o cifrado convencional. En cambio, si el emisor y el receptor usan cada uno claves diferentes, el sistema se denomina asimétrico, de dos claves, o cifrado de clave pública.
3. *La forma de procesar el texto claro.* Un cifrador de bloque procesa un bloque de elementos cada vez, produciendo un bloque de salida por cada bloque de entrada. Un cifrador de flujo procesa los elementos de entrada continuamente, produciendo la salida de un elemento cada vez [42].

7.12. CLAVE SIMÉTRICA

El cifrado simétrico, también conocido como cifrado convencional, de una clave secreta o de una clave única, era el único que se usaba antes del desarrollo del cifrado de la clave pública a fines de los 70. Aunque hoy continúa siendo el más usado de los cifrados cabe mencionar los principios del cifrado simétrico.

- *Texto claro:* Es el mensaje o los datos originales que se introducen en el algoritmo como entrada.
- *Algoritmo del cifrado:* El algoritmo de cifrado realiza varias sustituciones y transformaciones en el texto claro.
- *Clave secreta:* La clave es también una entrada del algoritmo. Las sustituciones y transformaciones realizadas por el algoritmo dependen de ella.

- *Texto cifrado:* El mensaje ilegible que se introduce como salida. Depende del texto claro y de la clave secreta. Para un mensaje determinado, dos claves diferentes producirán dos textos diferentes.
- *Algoritmo de descifrado:* Es básicamente, el mismo de cifrado ejecutado a la inversa. Toma el texto cifrado y la misma clave secreta y genera el texto claro.

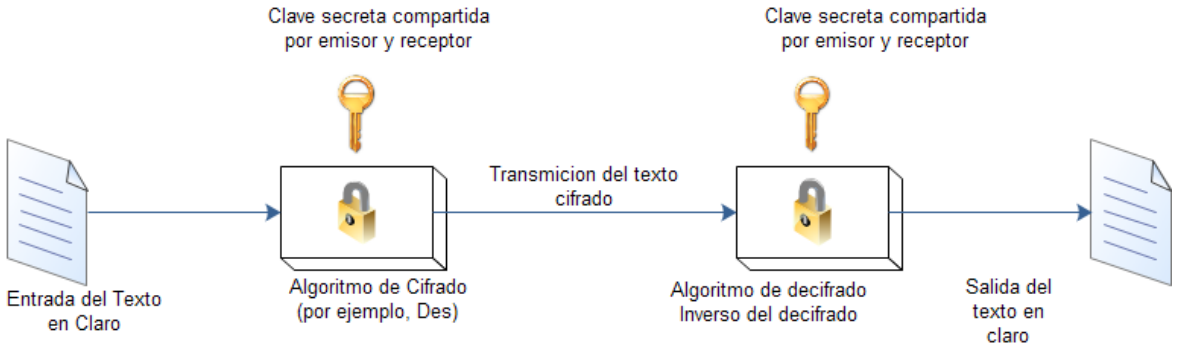


Figura 2: Modelo simplificado del cifrado convencional

Hay dos requisitos para el uso seguro del cifrado simétrico:

1. Se necesita un algoritmo de cifrado robusto. Como mínimo nos gustaría que un oponente que conozca el algoritmo y tenga acceso a uno o más textos cifrados no pudiera descifrar el texto o averiguar la clave. El atacante no debería poder descifrar el texto o averiguar la clave aunque estuviera en posesión de una serie de textos cifrados junto con sus correspondientes textos originales.
2. El emisor y el receptor deben haber obtenido copias de la clave secreta de forma segura y deben guardar de la misma manera. Si alguien descubre la clave y conoce el algoritmo, todas las comunicaciones que se usen son descifrables.

Es importante observar que la seguridad del cifrado simétrico depende de la privacidad de la clave, no de la privacidad del algoritmo. Es decir, se asume que nos es práctico descifrar un mensaje teniendo el texto cifrado y conociendo el algoritmo de cifrado/descifrado. En otras palabras, no es necesario que el algoritmo sea secreto: lo único que hay que mantener en secreto es la clave.

Esta característica del cifrado simétrico es la causa de su uso tan extendido. El hecho de que el algoritmo no tenga que ser secreto significa que los fabricantes pueden desarrollar y han desarrollado implementaciones con chips de bajo coste de los algoritmos de cifrado de datos. Esos chips se pueden conseguir fácilmente y se han incorporado a una serie de productos. Con el uso cifrado simétrico, el problema principal de seguridad consiste en mantener la privacidad de la clave [42].

Ventajas y desventajas de la clave asimétrica

La mayor ventaja de la criptografía asimétrica es que la distribución de claves es más fácil y segura ya que la clave que se distribuye es la pública manteniéndose la privada para el uso exclusivo del propietario, pero este sistema tiene bastantes desventajas:

- Para una misma longitud de clave y mensaje se necesita mayor tiempo de proceso.
- Las claves deben ser de mayor tamaño que las simétricas.
- El mensaje cifrado ocupa más espacio que el original.

Los nuevos sistemas de clave asimétrica basado en curvas elípticas tienen características menos costosas.

Herramientas como PGP, SSH o la capa de seguridad SSL para la jerarquía de protocolos TCP/IP utilizan un híbrido formado por la criptografía asimétrica para intercambiar claves de criptografía simétrica, y la criptografía simétrica para la transmisión de la información.

7.13. ALGORITMOS CRIPTOGRÁFICOS

Los algoritmos del cifrado simétrico más comunes usados son los cifrados de bloques. Un cifrador de bloques procesa la entrada de texto en claro en bloques de tamaño fijo y genera un bloque de texto cifrado del mismo tamaño para cada texto en claro.
Aquí nos centraremos en los cifrados de bloque simétrico más importante el DES (Data Encryption Standard o Estándar de Encriptación de Datos) y el 3DES.

DES (Data Encryption Standard)

El esquema de cifrado más extendido se basa en el DES (Data Encryption Standard) adoptado en 1977 por el Nacional Bureau of Standards, ahora el NIST (Nacional Institute of Standards and Technology), como Federal Información Processing Standard 46.

Descripción del algoritmo

El texto claro tiene una longitud de 64 bits y la clave, de 56; si el texto en claro es más largo se procesa en bloques de 64 bits. La estructura del DES consiste en una pequeña variación de la red de Feistel, que se muestra en la figura 3.

Hay 16 etapas de proceso. Se generan 16 subclaves partiendo de la clave original de 56 bits, una para cada etapa.

- Dado un bloque de N bits (típico 64) éste se dividirá en dos mitades.
- Existirá una función unidireccional F (muy difícil de invertir).
- Se realizan operaciones con la clave k_i sólo con una mitad del bloque, y se permutan en cada vuelta las dos mitades, operación que se repite durante n vuelta (ver figura 3).

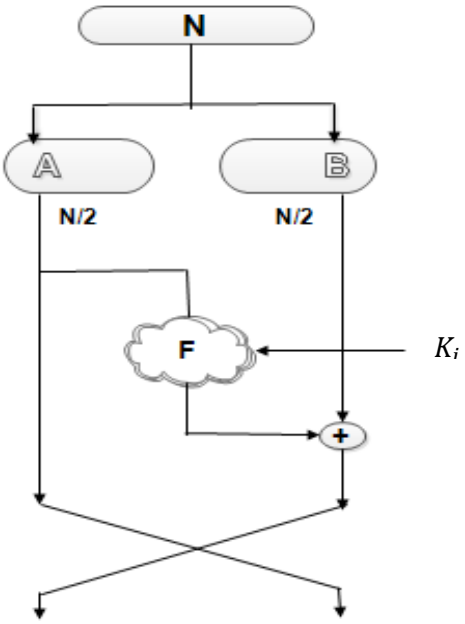


Figura 3: Red clásica de feistel

El proceso de descifrado con el DES es básicamente el mismo que el de cifrado. La regla es la siguiente: Usar el texto cifrado como entrada al algoritmo del DES, pero las subclaves K_i se pasan en orden inverso. Es decir, en la primera etapa se usa K_{16}, K_{16} en la segunda y así sucesivamente hasta K_1en la 16ª y última.

Los aspectos de robustez del DES se engloban en dos categorías: aspectos sobre el algoritmo mismo y aspectos sobre el uso de una clave de 56 bits. Los primeros se refieren a la posibilidad de que el criptoanálisis se realice explotando las características del algoritmo DES. A lo largo de los anos, se han intentado encontrar debilidades que explotar en el algoritmo, lo que ha hecho del DES el algoritmo de cifrado existente más estudiado. A pesar de los numerosos enfoques, nadie ha conseguido descubrir ninguna debilidad grave en el DES.

Un aspecto de mayor importancia es la longitud de la clave. Con una clave de 56 bits, hay 256 claves posibles, que es aproximadamente $7{,}2\ X10^{16}$ claves. Por este motivo, no parece práctico un ataque de fuerza bruta. Suponiendo que, en promedio, se tiene que intentar la mitad del espacio de claves, una única máquina que realice un cifrado DES por microsegundo tardaría más de mil años en romper el cifrado.

Es importante tener en cuenta que para que un ataque de búsqueda de clave no basta con probar todas las posibles claves. A menos que se suministre el texto claro, el analista debe ser capaz de reconocer el texto claro como tal. Si el mensaje es texto claro en inglés, entonces el resultado se obtiene fácilmente, aunque la tarea de reconocimiento del inglés tendría que estar automatizada. Si el mensaje de texto se ha comprimido antes del cifrado, entonces el reconocimiento es más difícil a continuación modos de cifra.

Todos los algoritmos pueden usarse aplicando diversos modos de cifra, entre ellos:

- ECB: Electronic Code Book (libro electrónico de códigos)
- CBC: Cipher Block Chaining (encadenamiento de bloques)
- CFB: Cipher Feed Back (realimentación de bloques)
- OFB: Output FeedBack (realimentación bloque de salida)

Analizaremos cada uno de ellos para el caso del DES, aunque el estudio es extensible a todos los demás ya que en estos modos el cifrador se considera una caja negra.

Especificaciones técnicas finales del DES

- Bloque a cifrar: 64 bits
- Clave: 8 bytes (con paridad, no caracteres ASCII)
- Normas ANSI:
- X3.92: Descripción del algoritmo.
- X3.108: Descripción de los modos de operación (ECB, CBC, OFB).

Visión general del DES

- Cifrador de bloque
- Tipo Feistel
- Longitud de clave de 56 bits
- Realiza 16 vueltas.
- La cifra del bloque central usa técnicas de sustituciones y permutaciones.
- Para poder realizar las sumas or exclusivo, usará permutaciones con expansión y compresión para igualar el número de bits.

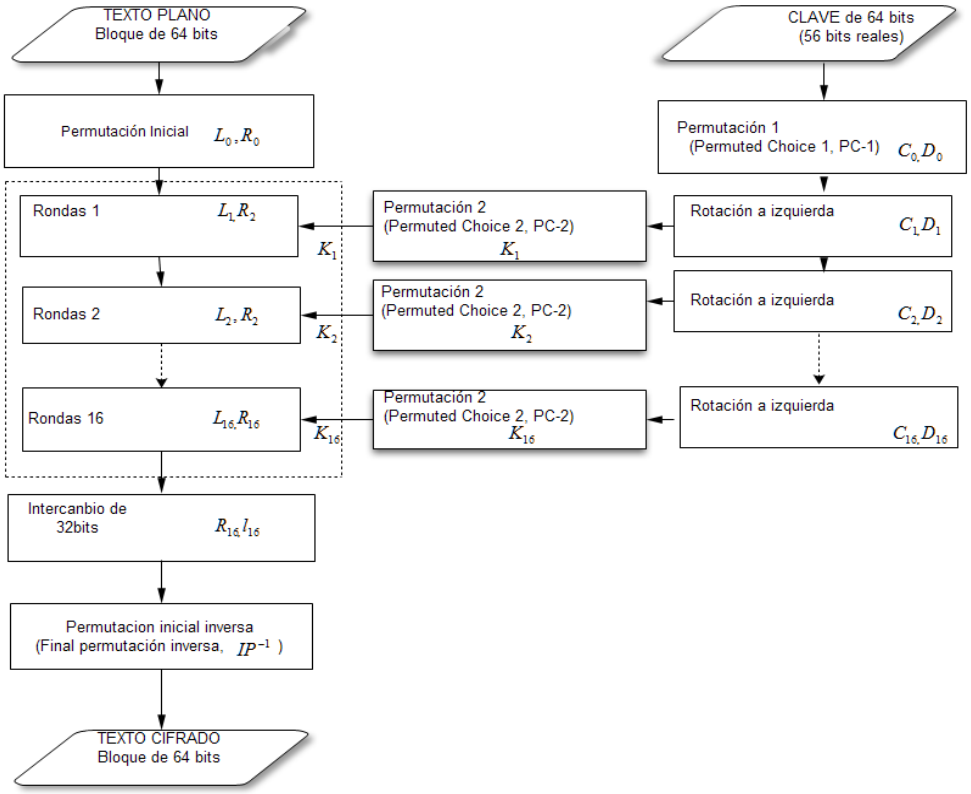


Figura 4: Esquema general del algoritmo DES

La primera etapa es una transposición, una permutación inicial (IP) del texto plano de 64 bits, independientemente de la clave. La última etapa es otra transposición (IP-1), exactamente la inversa de la primera. La penúltima etapa intercambia los 32 bits de la izquierda y los 32 de la derecha. Las 16 etapas restantes son una Red de Feistel de 16 rondas. En cada una de las 16 iteraciones se emplea un valor K_i, obtenido a partir de la clave de 56 bits y distinto en cada iteración.

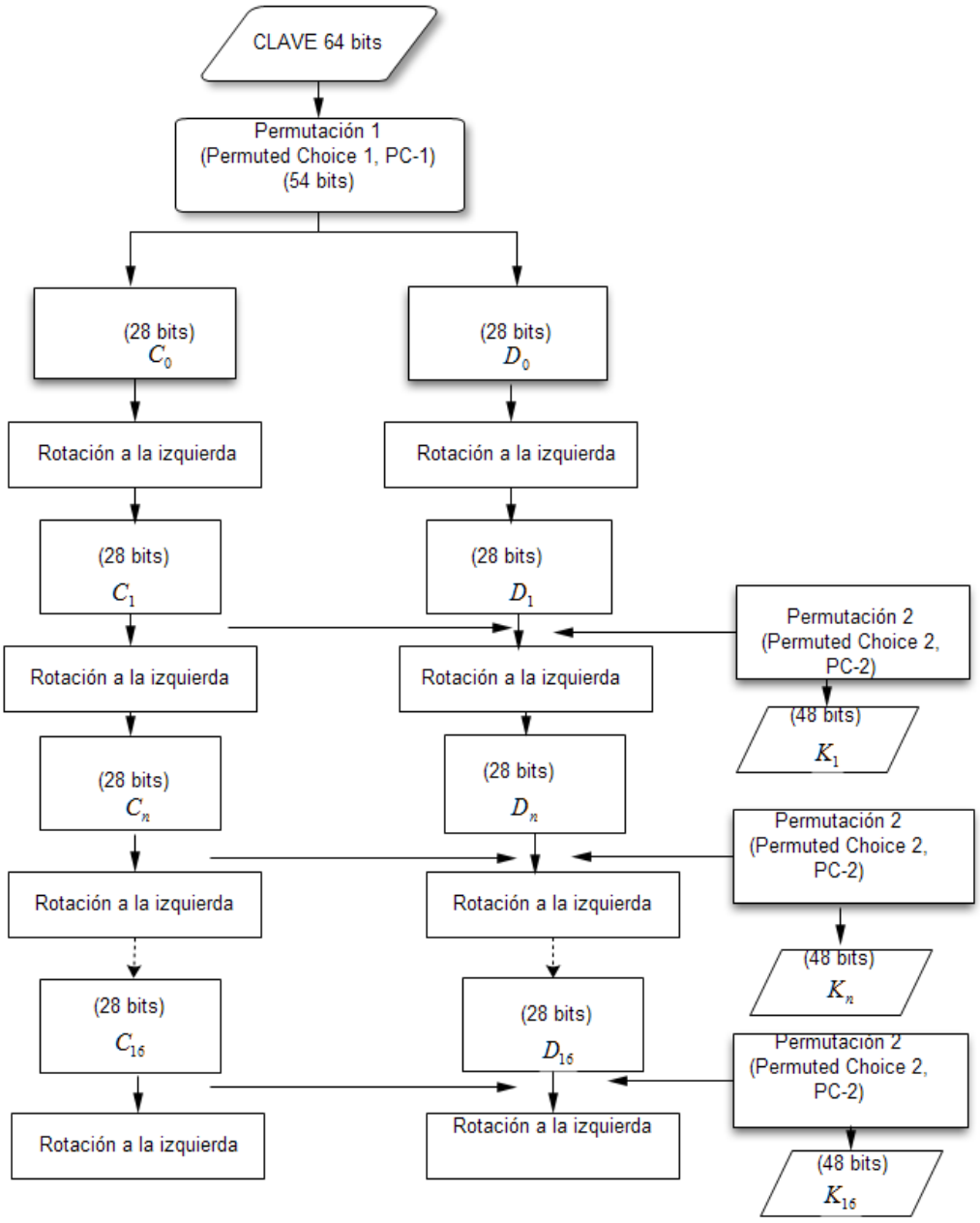


Figura 5: Cálculo de las subclaves, K_i

Se realiza una permutación inicial (PC-1) sobre la clave, y luego la clave obtenida se divide en dos mitades de 28 bits, cada una de las cuales se rota a izquierda un número de bits determinado que no siempre es el mismo. K_i, se deriva de la elección permutada (PC-2) de 48 de los 56 bits de estas dos mitades rotadas. La función f de la red de Feistel se compone de una permutación de expansión (E), que convierte el bloque correspondiente de 32 bits en uno de 48. Después realiza una or-exclusiva con el valor K_i, también de 48 bits, aplica ocho S-Cajas de 6*4 bits, y efectúa una nueva permutación (P).

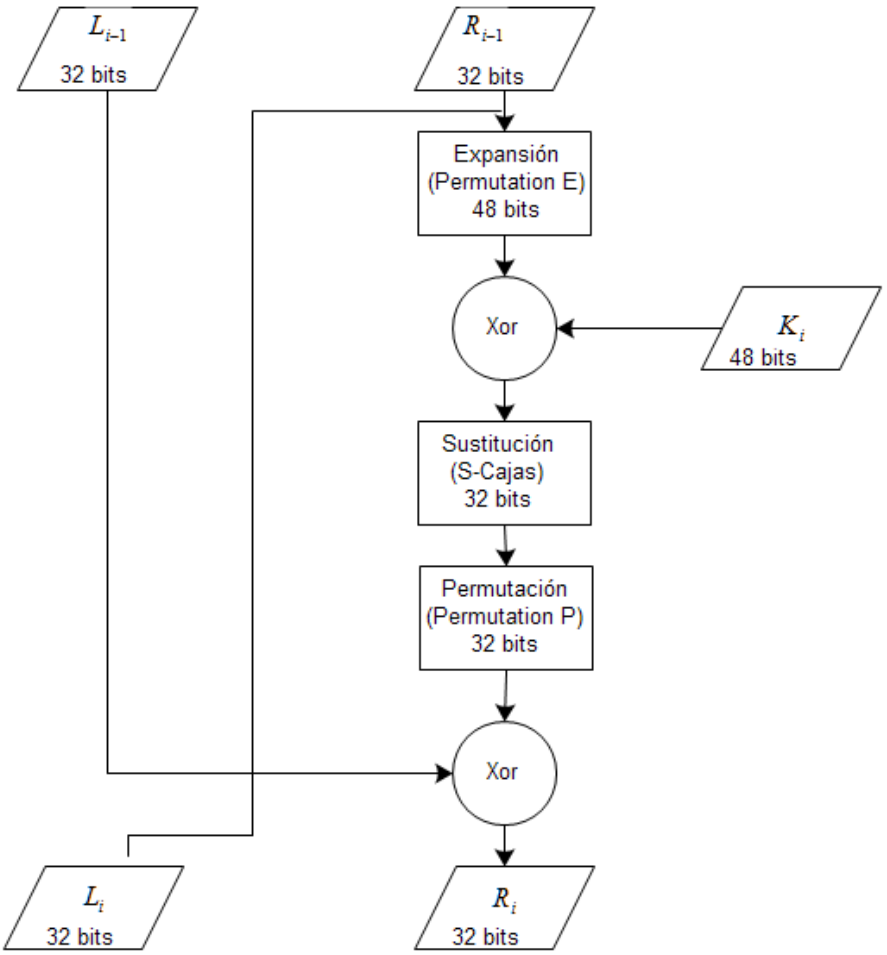


Figura 6: Ronda del algoritmo DES

Para descifrar basta con usar el mismo algoritmo empleando las K_i, en orden inverso.

Ejemplo de operación de cajas S del DES

Sean los bits 7 al 12 los siguientes: 101100

Los bits corresponderán entonces a la entrada de la caja S_2

Para seleccionar la fila tomamos los bits extremos: $10_2 = 2_{10} = 2$

Para seleccionar la columna tomamos los bits centrales: $0110_2 = 6_{10} = 6$

La caja S_2 indica una salida igual a $13_{10} = 1101_2$

S2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Tabla 1: Caja S2 del DES

Por cada 6 bits entrante se tiene 4 bits de salida.

Entrada: 101100 (6 bits)

Salida: 1101 (4 bits)

TRIPLE DES (3DES)

El triple DES (3DES) se estandarizo inicialmente para aplicaciones financieras en el estándar ANSI X9.17 en 1985. El 3DES se incorporó como parte del DES en 1999, con la publicación de FIPS PUB 463.

El 3DES usa tres claves y tres ejecuciones del algoritmo DES. La función sigue la secuencia cifrar-descifrar-cifrar (EDE: Encrypt Decrypt Encrypt) figura 7:

$$C = E_{K3}\,[\,D_{K2}[E_{K1}\,[P]]]$$

Donde

C = texto cifrado

P = texto claro

$E_K\,[X]$ = cifrado de X usando la clave K

$D_K\,[Y]$ = descifrado de Y usando la clave K

El descifrado es simplemente la misma operación con las claves en orden inverso (figura 7):

$$P = D_{K1}\,[\,E_{K2}\,[D_{K3}\,[C]]]$$

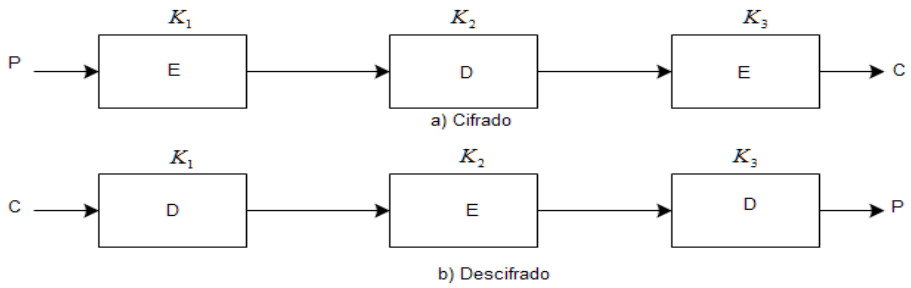


Figura 7: Triple DES

El descifrado del segundo paso no es significativo en términos criptográficos. Su única ventaja es que Permite a los usuarios del 3DES descifrar datos cifrados por usuarios del DES:

$$C = E_{K1}\,[\,D_{K1}\,[E_{K1}\,[P]]] = E_{K1}\,[P]$$

Con tres claves diferentes, el 3DES tiene una longitud efectiva de clave de 168 bits. El FIPS 463también permite el uso de dos claves, con $K_1 = K_3$, lo que proporciona una longitud de clave de 112 bits.

El FIPS 463 incluye las siguientes directrices para el 3DES:

- El 3DES es el algoritmo de cifrado simétrico oficial del FIPS.
- El DES original, que usa una única clave de 56 bits, se mantiene solo para los sistemas existentes. Las nuevas adquisiciones deberían admitir 3DES.
- Se apremia a las organizaciones gubernamentales con sistemas que usan DES a migrar a 3DES.
- Se prevé que el 3DES y el AES (Advanced Encryption Standard) coexistirán como algoritmos oficiales del FIPS, permitiendo una transición gradual hacia el AES.

Es fácil observar que el 3DES es un algoritmo robusto. Debido a que el algoritmo criptográfico que lo sustenta es el DES, el 3DES resulta igual de resistente al criptoanálisis basado en el algoritmo que el DES. Es más, con una clave de 168 bits de longitud, los ataques de fuerza bruta son efectivamente imposibles.

AES (Advanced Encryption Standard)

El 3DES tiene dos atractivos que aseguran su uso durante los próximos anos. Primero, con su longitud de clave de 168 bits evita la vulnerabilidad al ataque de fuerza bruta del DES. Segundo, el algoritmo de cifrado que usa es el mismo que en el DES. Este algoritmo ha estado sujeto a más escrutinios que ningún otro durante un largo periodo de tiempo, y no se ha encontrado ningún ataque criptoanalítico efectivo basado en el algoritmo que no sea la fuerza bruta. Por lo tanto, hay un alto grado de seguridad en la resistencia al criptoanálisis del 3DES. Si la seguridad fuera la única consideración, el 3DES sería una elección adecuada para un algoritmo de cifrado estándar durante las primeras décadas.

El DES original se diseñó para implementaciones hardware de mediados de los 70 y no produce código software eficiente. El 3DES tiene tres veces más etapas que el DES y, por ello es más lento. Debido a este inconveniente, el 3DES no es un candidato razonable para usarlo durante mucho tiempo. Para reemplazarlo, el NIST realizo en 1997 un concurso de propuestas para el desarrollo de un nuevo estándar de cifrado avanzado (AES), que debería ser tan robusto o más que el 3DES y que mejoraría significativamente la eficiencia. Además de esos requisitos generales, el NIST especifico que el AES debía ser un Cifrador simétrico de bloque con una longitud de bloque de 128 bits y permitir longitudes de clave de 128, 192 y 256 bits.

Descripción del algoritmo

La estructura del algoritmo AES está formada por un conjunto de rondas, entendiendo por rondas un conjunto de iteraciones de cuatro funciones matemáticas diferentes e invertibles. Por tanto, el algoritmo se basa en aplicar un número de rondas determinado a una información en claro para producir una información cifrada. La información generada por cada función es un resultado intermedio, que se conoce como Estado, o si se prefiere Estado Intermedio. El algoritmo representa el Estado como un matriz rectangular de bytes, que posee cuatro filas y Nb columnas. Siendo el número de columnas Nb en función del tamaño del bloque: La estructura del algoritmo Rijndael está formado por un conjunto de rondas, entendiendo por rondas un conjunto de iteraciones de cuatro funciones matemáticas diferentes e invertibles.

Por tanto, el algoritmo se basa en aplicar un número de rondas determinado a una información en claro para producir una información cifrada. La información generada por cada función es un resultado intermedio, que se conoce como Estado, o si se prefiere Estado Intermedio. El algoritmo representa el Estado como un matriz rectangular de bytes, que posee cuatro filas y Nb columnas. Siendo el número de columnas Nb en función del tamaño del bloque:

N_b= tamaño del bloque utilizado en bits /32.

De esta manera, si se tiene un bloque con 128 bits se tendría una matriz de cuatro filas y $N_b = 128/32 = 4$columnas quedando una matriz de forma:

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

Tabla 2: Matriz de cuatro filas y columnas

La clave del sistema se representa con una estructura análoga a la del Estado, es decir, se representa mediante una matriz rectangular de bytes de cuatro filas y $N_K$columnas. Siendo el número de columnas $N_K$en función del tamaño de la clave:

N_K= tamaño de la clave en bits/32.

De esta manera, al igual que ocurría con el tamaño del bloque, si se tiene una clave con 128 bits se dispondría de una matriz de cuatro filas y N_K= 128/32 = 4 columnas, quedando una matriz de forma:

K_{00}	K_{01}	K_{02}	K_{03}
K_{10}	K_{11}	K_{12}	K_{13}
K_{20}	K_{21}	K_{22}	K_{23}
K_{30}	K_{31}	K_{32}	K_{33}

Tabla 3: Clave con 128 bits se dispondría de una matriz de cuatro filas y columnas

El número de iteraciones o vueltas de las cuatro transformaciones sobre la Información, o mejor dicho sobre la matriz de estado intermedio depende de la versión del algoritmo que se utilice. Los autores definen que para tamaños de bloques y claves entre 128 y 256 bits (con incrementos de 32 bits) el número de vueltas $N_r$es determinado por la siguiente expresión:

$$N_r = máx(N_K, N_b)\ L + 6.$$

Las rondas para cada uno de los posibles resultados, siendo la primera columna el tamaño de la clave y la primera fila el tamaño del bloque es la siguiente:

Clave/bloque	$\boldsymbol{N_b = 4(128\ bits)}$	$\boldsymbol{N_b = 6(192\ bits)}$	$\boldsymbol{N_b = 8(256\ bits)}$
$\boldsymbol{N_K = 4(128\ bits)}$	10	12	14
$\boldsymbol{N_K = 6(192\ bits)}$	12	12	14
$\boldsymbol{N_K = 8(256bits)}$	14	14	14

Tabla 4: Rondas del Aes

Se puede ver en el gráfico que el proceso de cifrado consiste en la aplicación de cuatro funciones matemáticas invertibles sobre la información que se desea cifrar. Las transformaciones se realizan de forma reiterativa para cada ronda o vuelta definida. Las funciones matemáticas realizadas son:

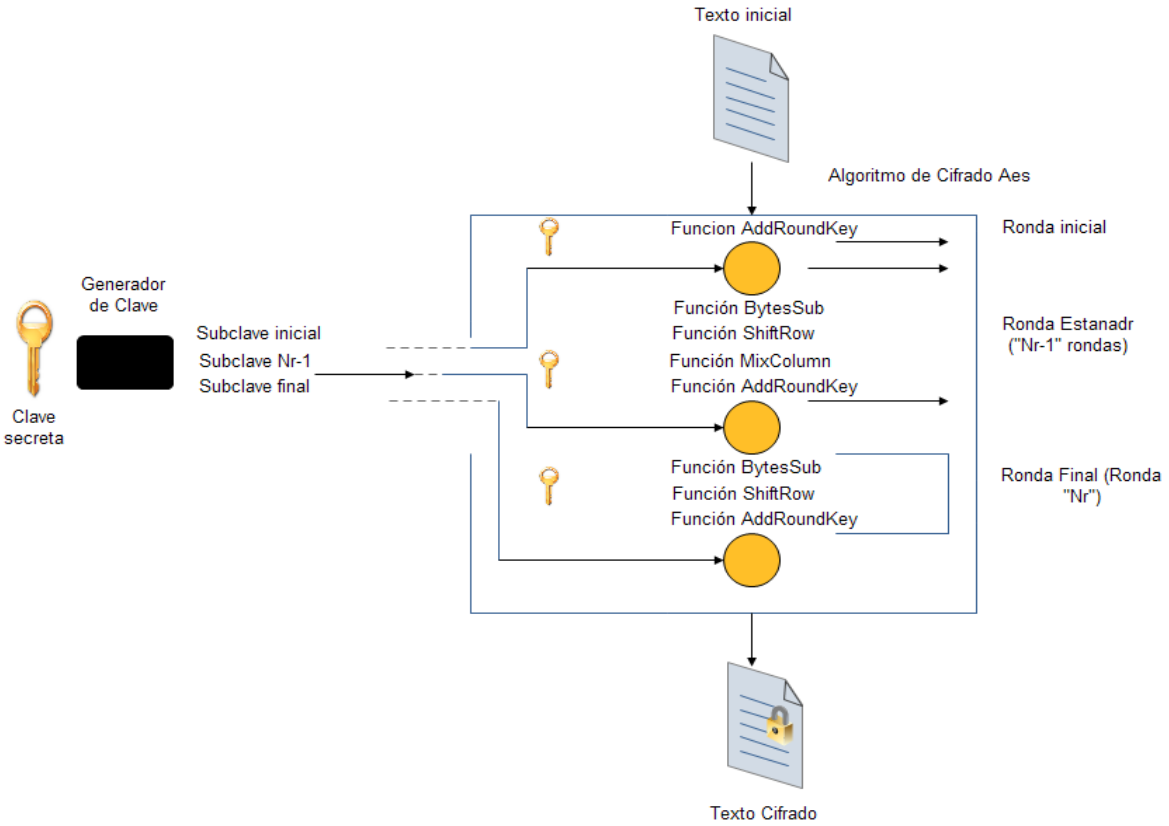


Figura 8: Esquema general del Aes

- **Función AddRoundKey:** Es la primera transformación que se ejecuta, es una suma XOR entre el texto plano y la clave.
- **Función ByteSub:** La transformación ByteSub, es una sustitución no lineal, que opera independientemente en cada byte de la matriz. Cada byte del bloque de entrada es invertido sobre$GF(2^{8)}$ y luego sufre una transformación afín. La operación completa puede realizarse mediante una matriz de sustitución, conocida con el nombre de S-Box. La transformación ByteSub por medio del uso de dicha matriz, consiste en tomar el elemento. (byte) de la matriz A de cuatro filas por cuatro columnas, en formato hexadecimal. Dicho número se transforma en el subíndice necesario para extraer un valor de la matriz S-box.

- **Función ShiftRow:** A continuación de ByteSub, se aplica la transformación ShiftRow. En la misma, los bytes de las tres últimas filas son desplazados cíclicamente en distintas cantidades o posiciones (offsets). En la primera fila el offset es nulo, es decir, que la fila no se desplaza. En la segunda fila, el desplazamiento es de un byte hacia la izquierda, en la tercera es de dos bytes y en la última es de tres bytes, también hacia la izquierda.
- **Función MixColumn:** Se trata de la transformación siguiente a ShiftRow, MixColumns opera columna por columna de la matriz, tomando a cada una como un polinomio de grado tres. Es decir, que las columnas son consideradas como polinomios en el campo y cada una es multiplicada por una fila de la siguiente matriz $Ref(x)$.

En esencia, la información a cifrar se va transformando en la matriz de Estado. Esta matriz de Estado se introduce al cifrador, y sufre una primera transformación, en la ronda inicial, que consiste en una operación or-exclusiva (AddRoundKey) entre una Subclaves generada y la matriz de Estado. A continuación, a la matriz de Estado resultante se le aplican cuatro transformaciones invertibles, repitiéndose este proceso $N_r - 1$ veces, en lo que se conoce como Ronda Estándar. Finalmente, se le aplica una última ronda o vuelta a la matriz de Estado resultante de las $N_r - 1$rondas anteriores, aplicando las funciones ByteSub, ShiftRow y AddRoundKey en este orden. El resultado de la ronda final produce el bloque cifrado deseado. Recordar que AES interpreta el bloque de entrada de 128 bits, como una matriz 4 x 4 de entradas de bytes. Si el bloque es de 192 bits se agregan dos columnas más, y si es de 256 se agregan cuatro columnas más.

A partir de esta matriz combinada, con las funciones que se pasa a explicar en los siguientes apartados, se obtendrá una descripción detallada del algoritmo. Las funciones que se explicarán tienen un propósito fijo y preciso:

- AddRoundKey o etapa de adición de clave.
- SubByte o sustituciones de bytes.
- ShiftRows o etapa de desplazamiento de filas.
- MixColumns o etapas de mezcla de columnas.
- KeySchedule o expansión de clave.

Función AddRoundKey

En esta función, etapa o tratamiento se procede a realizar un XOR byte a byte entre la matriz de estado o matriz intermedia y la matriz de la clave o subclave, dependiendo de la ronda en la que se encuentre.

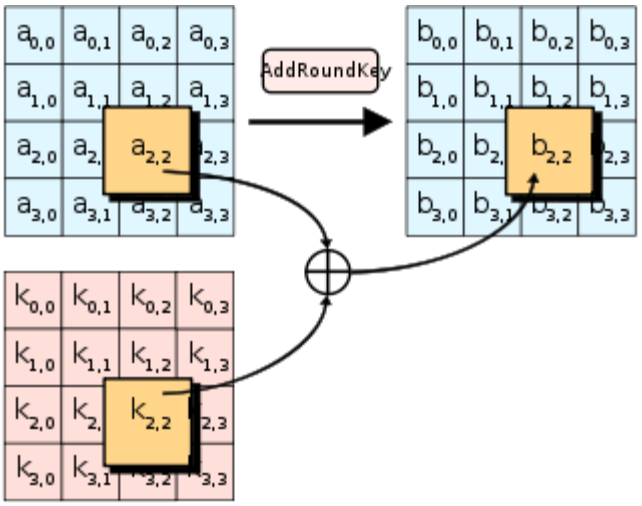


Figura 9: Función AddRoundKey

Función ByteSub

En esta función, etapa o tratamiento se procede a realizar una sustitución no lineal que se aplica a cada byte de la matriz de estado de forma independiente, generando un nuevo byte, es decir, a cada elemento de la matriz de estado se le sustituye por otro byte que depende del primero mencionado. Esta sustitución se lleva a cabo utilizando unas tablas o matrices S-Box invertibles.

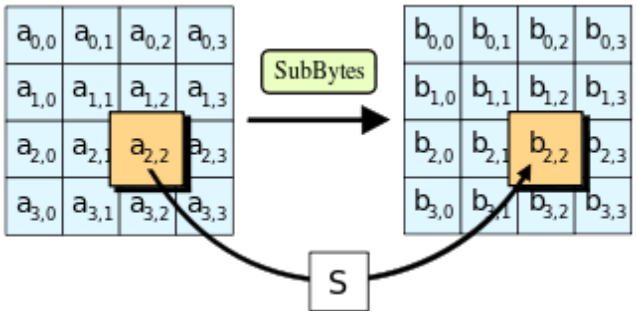


Figura 10: SubBytes

Explicación de la imagen: Se trata de una función no lineal que se realiza a través de una S-box. La S-box se construye: a) calculando el inverso en $GF(2^8)$, y b) calculando la siguiente transformación afín sobre GF (2): $b_i' = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i$. Donde $0 \leq i < 8$, b_i es el iésimo bit del byte y c_i es el iésimo bit del byte c cuyo valor es $\{63\}_{16}$ o $\{011000011\}_2$. En la siguiente tabla mostramos todos los inversos multiplicativos en representación hexadecimal, por ejemplo el inverso de 88 es 9b. Por definición al 00 le asociamos el mismo 00.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	F
0	00	01	8d	F6	cb	52	7b	d1	E8	4f	29	c0	b0	e1	e5	c7
1	74	b4	aa	4b	99	2b	60	5f	58	3f	fd	cc	ff	40	ee	B2
2	3a	6e	5a	f1	55	4d	a8	c9	c1	0a	9B	15	30	44	a2	c2
3	2c	45	92	6c	f3	39	66	42	f2	35	20	6f	77	bb	59	19
4	1d	fe	37	67	2d	31	f5	69	a7	64	ab	13	54	25	e9	09
5	ed	5c	05	ca	4c	24	87	bf	18	3e	22	F0	51	ec	61	17
6	16	5e	af	d3	49	a6	36	43	f4	47	91	df	33	93	21	3b
7	79	b7	97	85	10	b5	ba	3c	b6	70	d0	06	a1	fa	81	82
8	83	7e	7f	80	96	73	be	56	9b	9e	95	d9	F7	02	b9	a4
9	de	6a	32	6d	d8	8a	84	72	2ª	14	9f	88	f9	dc	89	9ª
a	fb	7c	2e	c3	8f	b8	65	48	26	c8	12	4a	ce	e7	d2	62
b	0c	e0	1f	ef	11	75	78	71	a5	8e	76	3d	bd	bc	86	57
c	0b	28	2f	a3	da	d4	e4	0f	a9	27	53	04	1b	fc	ac	e6
d	7a	07	ae	63	c5	db	e2	ea	94	8b	c4	d5	9d	F8	90	6b
e	b1	0d	d6	eb	c6	0e	cf	ad	08	4e	d7	e3	5d	50	1e	b3
f	5b	23	38	34	68	46	03	8c	Dd	9c	7d	a0	cd	1a	41	1c

Tabla 5: Inversos multiplicativos en representación hexadecimal

Lógicamente se cumple que:

Si:

inv x = y

inv y = x

Por ejemplo:

inv c4 = da

inv da = c4

Lo podemos ver en la tabla.

Finalmente el resultado de ByteSub puede resumirse en la siguiente tabla conocida como S-Box AES.

Funciones utilizadas para el cifrado

En esta sección se da la descripción detallada del algoritmo AES. Primero, recordar que AES trabaja con bloques de datos de 128 bits y longitudes de claves de 128, 192, 256 bit. Además AES tiene 10, 12 ó 14 vueltas respectivamente. Cada vuelta de AES consiste en la aplicación de una ronda estándar, que consiste en cuatro transformaciones básicas. La última ronda es especial y consiste de tres operaciones básicas, añadiendo siempre una ronda inicial. Por otro lado, se tiene el programa de claves o extensión de la clave.

Recordar que AES interpreta el bloque de entrada de 128 bits, como una matriz 4 x4 de entradas de bytes. Si el bloque es de 192 bits se agregan dos columnas más, y si es de 256 se agregan cuatro columnas más.

De esta manera el bloque de datos, suponiendo que se dispone de un bloque de 128 bits, es de la manera siguiente:

$$a_{00}\ \ a_{10}\ a_{20}\ a_{30}\ a_{01}\ \ a_{11}\ a_{21}\ \ a_{31}\ a_{02}\ \ a_{12}\ \ a_{22}\ \ a_{32}\ \ \ a_{03}\ a_{13}\ \ a_{23}\ a_{33}$$

Los cuatro primeros bytes conforman la primera columna. Así con los siguientes grupos de cuatro bytes se irán completando las diferentes columnas de manera que se queda una matriz.

De esta misma manera se obtendrán las matrices respectivas para los bloques de 192 y 256 bits, siendo las matrices de 6 ó de 8 columnas, respectivamente. De esta manera la clave, suponiendo se tiene como ejemplo una de 128 bits, es de la manera indicada:

$$K_{00}\ \ K_{10}\ K_{20}\ K_{30}\ K_{01}\ K_{11}\ \ K_{21}\ \ K_{31}\ K_{02}\ \ K_{12}\ \ K_{22}\ \ K_{32}\ \ \ K_{03}\ K_{13}\ \ K_{23}\ K_{33}$$

Los cuatro primeros bytes conforman la primera columna y los siguientes grupos de cuatro bytes irán completando las diferentes columnas de manera que se queda una matriz 4*4. A partir de esta matriz combinada, con las funciones que se pasa a explicar en los siguientes apartados, se obtendrá una descripción detallada del algoritmo

Cálculo de las subclaves

Basándose en el principio de la criptografía moderna, mediante el cual se establece que la seguridad de un algoritmo sólo debe depender de la clave utilizada, se utilizan diferentes subclaves K_i tanto en el cifrado como en el descifrado para que el resultado del algoritmo dependa de una información externa al sistema: la clave del usuario.

Estas subclaves (RoundKeys) se derivan de la clave principal K mediante el uso de dos funciones: una de expansión y otra de selección. Siendo n el número de rondas que aplique el algoritmo, el número total de bits para subclaves que se necesitan para todas las rondas es igual a el tamaño del bloque utilizado multiplicado por n +1. También se puede ver como 4 Hn +1L* N_b bytes. Es decir, por ejemplo, para un tamaño de bloque de 128 bits y 10 vueltas, se necesitan 1408 bits de subclaves: H128*11 = 1408bitsL.

Por tanto, lógicamente, el número de claves que se generan depende del número de rondas empleadas N_r. Después se aplica la siguiente transformación afín en GFH28L, siendo x0, x1, ..., x7 los bits del byte correspondiente, y a su vez y0, y1,..., y7 los bits del byte correspondiente al resultado.

El resultado final de estas dos transformaciones se puede expresar en una tabla de sustitución denominada S-Box aplicada a cada byte, sabiendo que este está expresado en forma hexadecimal, de manera que el elemento más a la izquierda deberá situarse en la columna amarilla y el elemento más a la derecha deberá colocarse en la fila azul, siendo el resultado final el lugar donde coincidan o se unan.

Transformación de SubBytes

Es muy sencillo transformarlo, la cual convierte los datos de 8 bits a otros datos de 8 bits. Por ejemplo, datos de 8 bits "00000000" se transforma en "01100011". Lo que es importante es que los diferentes datos byte siempre se transforman en diferentes datos de 8 bits. Si esta condición se viola, los datos transformados no se pueden recuperar. Tal sistema no puede ser utilizado para el cifrado de datos. Esta condición hace necesaria características Galois Field ajuste.

La definición de SubBytes La transformación es la transformación de serie de las dos transformaciones siguientes.

1. Tome el inverso multiplicativo en Campo Galois GF (2^8), el elemento de 8 bits "00000000" = hex {00} se asigna a sí mismo. El número de 8 bits "01010011" es {53} en hexadecimal. Luego consulte la tabla por X = 5, Y = 3, entonces se obtiene la salida {CA} en hexadecimal. {CA} es "11001010" en formato binario.

Polinomio irreducible es como sigue. $m(x) = x^8 + x^4 + x^3 + x + 1$

2. A continuación, aplicar la transformación afín sobre GF (2^8). Binario "00000000" = {00} Hex se transformará en "01100011" = {63}.

$$\begin{bmatrix} b7 \\ b6 \\ b5 \\ b4 \\ b3 \\ b2 \\ b1 \\ b0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} a7 \\ a6 \\ a5 \\ a4 \\ a3 \\ a2 \\ a1 \\ a0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Si (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)={CAEntonces el cálculo sobre GF (2^8) se realiza como sigue. {CA} se transformará en "11101101 = {ED}.

Totalmente, {53} se convierte en {CA} por inversión, y luego transformado en {ED} por transformar afín. Estos serial transformar es la transformación SubBytes.

SubBytes transformada inversa

SubBytes Inverse La transformación es la operación inversa de la transformación SubBytes. A continuación se ejecutará la transformación de dos de serie siguiente.

1. Se ejecuta la transformación afín inversa. La transformación inversa se da en la siguiente ecuación.

$$\begin{bmatrix} a7 \\ a6 \\ a5 \\ a4 \\ a3 \\ a2 \\ a1 \\ a0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} b7 \\ b6 \\ b5 \\ b4 \\ b3 \\ b2 \\ b1 \\ b0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

2. continuación, realizar la misma inverso multiplicativo en Galois GF campo 2^8 .

Dado que ambos SubBytes y transformaciones inversas SubBytes necesita el inverso multiplicativo en Galois GF.

3. Cuando la señal INV es "0", los datos de entrada en primer lugar va al inverso bloquear luego fue trasladado a afín a transformar, y cuando la señal INV se afirma (= "1"), los datos de entrada se transforma primero en Inv Affin Bloquear y más tarde Inverso en GF (2^8) se calcula.

Desde el afín a transformar es sólo un cálculo utilizando AND y puertas EXOR, cómo calcular el inverso en GF (28) debe ser considerado Weill en la tarea de diseño.

¿Cómo se calcula Inverso en GF .

Más fácil implementación de cálculo inverso en GF (2^8) es el uso de la palabra ROM 256 x 8 bits que posee los datos. Sin embargo, la aplicación ROM no se intersting en el concurso de proyectos.

Otra idea de ejecución se explica cómo sigue.

Supongamos que es de entrada y en GF (2^8), y-1 tienen que ser calculados. En G (2^8),

$$y^{2^8-1} = y^{255} = 1$$

$$y^{-1} = y^{-1} \cdot y^{255} = y^{254}$$

A continuación ejemplo de operación ByteSub usando S cajas.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1ª	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	Ed	20	fc	b1	sb	6a	Cb	be	39	4a	4c	58	cf
6	d0	ef	aa	Fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	Ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	Dc	22	2a	90	88	46	Ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	Dd	74	1f	4b	bd	8b	8ª
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tabla 6: ByteSub conocida como S-Box AES.

Se pide calcular el ByteSub de 5a

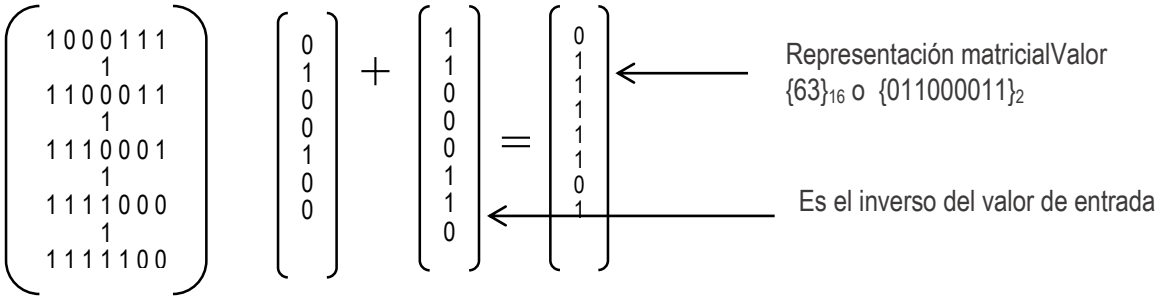
5a = 01011010 = $x^6 + x^4 + x^3 + x + 1$

Nota: 5a en decimal es 90 es por eso que decimos que este es igual a 01011010.

inv (5A) = 22 = 00100010 (según la tabla dada Tabla de inversos en mod 2^8)

Representación de la función ByteSub

La transformación afín anterior queda:



Función ShiftRow

En esta función, etapa o tratamiento se procede a realizar un desplazamiento a la izquierda cíclicamente de las filas que conforman la matriz de estado actual, es decir, rotar los bytes de las filas de la matriz de estado resultante de la transformación anterior a la izquierda.

Se puede decir que consiste en desplazar bloques de un byte hacia la izquierda módulo columna (en este caso 4) dentro de una fila. Así la fila 0 no desplaza, la fila 1 desplaza un byte, la fila 2 desplaza dos bytes y la fila 3 desplaza tres bytes como se muestra.

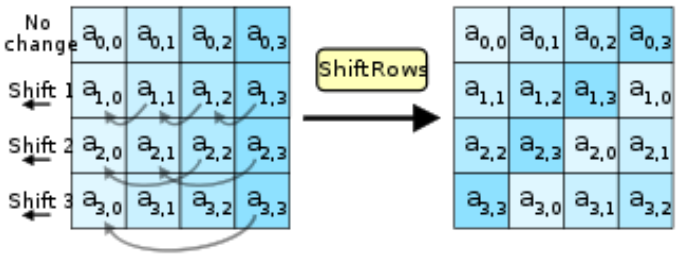


Figura 11: Función ShiftRow

Función MixColumns

En esta función, etapa o tratamiento se procede sobre los bytes de una misma columna de la matriz de estado resultante de la transformación anterior. Las columnas son tratadas como polinomios, cuyos coeficientes pertenecen a $\mathrm{GF}(2^8)$Multiplicando las columnas módulo $x^4 + 1$ por este polinomio fijo, en donde los valores {} están en hexadecimal, que es primo relativo con $x^4 + 1$ y por tanto asegura el inverso.

$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$

Por tanto, recuerde que {03} = x + 1, {02} = x, {01} = 1.

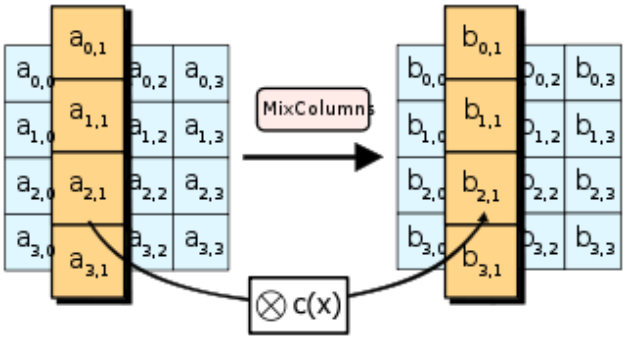


Figura 12: Función MixColumns

Representación matricial de la función MixColumns

$$\begin{pmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{pmatrix}$$

Ejemplo de operación MixColumns

$S'_{0,C} = (\{02\} \bullet S_{0,C}) \oplus (\{03\} \bullet S_{1,C}) \oplus S_{2,C} \oplus S_{3,C}$

$S'_{1,C} = S_{0,C} \oplus (\{02\} \bullet S_{1,C}) \oplus (\{03\} \bullet S_{2,C}) \oplus S_{3,C}$

$S'_{2,C} = S_{0,C} \oplus S_{1,C} \oplus (\{02\} \bullet S_{2,C}) \oplus (\{03\} \bullet S_{3,C})$

$S'_{3,C} = (\{03\} \bullet S_{0,C}) \oplus S_{1,C} \oplus S_{2,C} \oplus (\{02\} \bullet S_{3,C})$

El primer byte de estado $S'_{0,0}$ quedará:

$S'_{0,0} = \{02\}S_{0,0} \oplus \{03\}S_{1,0} \oplus S_{2,0} \oplus S_{3,0}$

$S'_{0,0} = \{02\}e1 \oplus \{03\}fb \oplus 96 \oplus 7c$

$\{02\}e1 = x(x^7 + x^6 + x^5 + 1)$

$\{02\}e1 = x^8 + x^7 + x^6 + x$

$\{02\}e1 = (x^8 + x^7 + x^6 + x) \bmod x^4 + 1 = d2$

$\{03\}fb = (x + 1)(x^7 + x^6 + x^5 + x^4 + x^3 + x + 1)$

$\{03\}fb = x^8 + x^3 + x^2 + 1$

$\{03\}fb = (x^8 + x^3 + x^2 + 1) \bmod x^4 + 1 = 1d$

Operaciones con bytes en AES

- Suma y multiplicación. Son cálculos en Campos de Galois GF (2^8) con 8 bits. Para la reducción de exponente se usará un polinomio primitivo $p(x) = x^8 + x^4 + x^3 + x + 1$.
- Producto por x. Esta operación conocida como xtime(a) al igual que en el caso anterior usa la reducción de exponente. Puede implementarse fácilmente con desplazamientos y operaciones or exclusivo.

Ejemplo de suma en GF (2^8)

Vamos a sumar los valores hexadecimales 57 y 83:

$A = 57_{16} = 0101\ 0111_2$ $B = 83_{16} = 1000\ 0011_2$

que expresados en polinomios dentro de GF(2^8) serán:

$A = 0101\ 0111_2 = x^6 + x^4 + x^2 + x + 1$

$B = 1000\ 0011_2 = x^7 + x + 1$

Sumando: $A+B = (x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) \bmod 2$

$A+B = (x^7 + x^6 + x^4 + x^2 + 2x + 2) \bmod 2$

$A+B = x^7 + x^6 + x^4 + x^2 = 1101\ 0100_2 = d4_{16}$

Y lo mismo se obtiene con la suma Or exclusivo:

$0101\ 0111 \oplus 1000\ 0011 = 1101\ 0100_2 = d4_{16}$

Ejemplo de producto en GF (2^8)

Vamos a multiplicar los valores hexadecimales 57 y 83:

$A = 57_{16} = 0101\ 0111_2$ $B = 83_{16} = 1000\ 0011_2$

Que expresados en polinomios dentro de GF (2^8) serán:

$A = 0101\ 0111_2 = x^6 + x^4 + x^2 + x + 1$

$B = 1000\ 0011_2 = x^7 + x + 1$

$A*B = (x^6 + x^4 + x^2 + x + 1)*(x^7 + x + 1) \bmod 2$

$A*B = x^{13} + x^{11} + x^9 + x^8 + 2x^7 + x^6 + x^5 + x^4 + x^3 + 2x^2 + 2x + 1$

Reduciendo mod 2

$A*B = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$

Este resultado hay que reducirlo por $p(x) = x^8 + x^4 + x^3 + x + 1$

Como el polinomio irreducible es $p(x) = x^8 + x^4 + x^3 + x + 1$

$(x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1) \bmod (x^8 + x^4 + x^3 + x + 1)$

Para dejar los valores dentro de GF (2^8) vemos que un divisor será $x^5 + x^3$ puesto que $x^5x^8 = x^{13}$; $x^5x^4 = x^9$; $x^5x^3 = x^8$ y $x^3x^8 = x^{11}$.

$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$ | $x^8 + x^4 + x^3 + x + 1$

$x^5 + x^3$

$x^{13} + x^9 + x^8 + x^6 + x^5$ *multiplicando p(x) por* x^5

$x^{11} + x^4 + x^3 + 1$ *reduciendo mod 2*

$x^{11} + x^7 + x^6 + x^4 + x^3$ *multiplicando p(x) por* x^3

$x^7 + x^6 + 1$ *reduciendo mod 2*

Resultado: $A*B = x^7 + x^6 + 1 = 1100\ 0001_2 = c1_{16}$

Otros cifradores de bloque simétricos

En vez de reinventar de nuevo la rueda, casi todos los algoritmos de cifrado de bloque simétricos actuales utilizan la estructura básica de bloque de Feistel. Esto se debe a que dicha estructura se comprende muy bien, resultado más fácil determinar la robustez criptográfica de un algoritmo nuevo. Si se usara unaestructura totalmente nueva, podría tener alguna debilidad sutil no detectadainmediatamente por el diseñador.

En la tabla 7 se comparan algunas de las principales características.

Algoritmo	Tamaño de la clave(bits)	Tamaño de bloque (bits)	Numero de etapas	Aplicaciones
Des	56	64	16	Set, Kerberos
Triple Des	112 o 168	64	48	Financial Key management, PGP, S/Mime
Aes	128, 192 o 256	128	10, 12 o 14	Destinados a distribuir DES y 3DES
IDEA	128	64	8	PGP
Blowfish	Variables hasta 448	64	16	Varios paquetes de software
Rc5	Variables hasta 2048	64	Variables hasta 255	Varios paquetes de software

Tabla 7: Algoritmos de cifrado convencional

IDEA

El IDEA usa una clave de 128 bits. Difiere notablemente del DES en la función de etapa así como en la función de generación de subclaves. Para la función de etapa el IDEA no usa cajas S, sino que cuenta con tres operaciones matemáticas diferentes: XOR, suma binaria de enteros de 16 bits, y multiplicación binaria de enteros de 16 bits. Esas funciones se combinan de tal forma que producen una transformación compleja muy difícil de analizar, y por ende muy difícil para el criptoanálisis. El algoritmo de generación de subclaves se basa solamente en el uso de desplazamientos circulares pero ejecutados de manera compleja para generar un total de seis subclaves para cada una de las ocho etapas del IDEA.

Debido a que el IDEA fue uno de los primeros algoritmos de 128 bits de los propuestos para remplazar al DES, ha sido sometido a considerables exámenes y por ahora parece ser muy resistente al criptoanálisis. El IDEA se usa como una alternativa en PGP (Pretty Good Privacy) y también en una serie de productos comerciales.

Descripción del algoritmo idea

El algoritmo IDEA (International Data Encryption Algorithm) es bastante más Joven que DES, pues data de 1992. Para muchos constituye el mejor y más seguro algoritmo simétrico disponible en la actualidad. Trabaja con bloques de 64 bits de longitud y emplea una clave de 128 bits. Como en el caso de DES, se usa el mismo algoritmo tanto para cifrar como para descifrar.

IDEA es un algoritmo bastante seguro, y hasta ahora se ha mostrado resistente a multitud de ataques, entre ellos el criptoanálisis diferencial. No presenta claves débiles, y su longitud de clave hace imposible en la práctica un ataque por la fuerza bruta. Como ocurre con todos los algoritmos simétricos de cifrado por bloques, IDEA se basa en los conceptos de confusión y difusión, haciendo uso de las siguientes operaciones elementales:

- XOR
- Suma modulo 2^{16}

Producto modulo$2^{16} + 1$.

BLOWFISH

El Blowfish consiguió ser rápidamente una de las alternativas más populares al DES. Se diseñó para que fuera fácil de implementar y rápido en su ejecución. También es un algoritmo muy compacto que puede ejecutarse en menos de 5K de memoria. Una característica interesante es que la longitud de la clave es variable, pudiendo alcanzar hasta los 448 bits. En la práctica se usan claves de 128 bits. El Blowfish usa16 etapas.

Utiliza cajas S y la función XOR, como el DES, pero también utiliza sumas binarias. Al contrario que el DES, que utiliza cajas S estáticas, Blowfish usa cajas S dinámicas generadas como una función de la clave. Las subclaves y las cajas S se generan por la aplicación repetida del propio algoritmo a la clave. Se necesita un total de 521 ejecuciones del algoritmo de cifrado Blowfish para producir las subclaves y las cajas S. Por este motivo no es adecuado para aplicaciones en las que la clave secreta cambia frecuentemente.

Este es uno de los algoritmos de cifrado simétrico más robusto hasta la fecha, porque tanto las subclaves como las cajas S se generan por un proceso de aplicaciones repetidas del propio algoritmo, lo cual modifica totalmente los bits haciendo muy difícil el criptoanálisis. Hasta ahora, se han publicado algunos artículos sobre Blowfish, sin que se hayan encontrado debilidades.

RC5

El RC5 se diseñó para tener las siguientes características:

- Adecuado para hardware y software: solo usa operaciones computacionales primitivas que se encuentran comúnmente en los microprocesadores.
- Rápido: para conseguir esto, el RC5 es un algoritmo simple y orientado a palabras. Las operaciones básicas procesan palabras enteras de datos cada vez.
- Número variable de etapas: el número de etapas es un segundo parámetro. Esto permite alcanzar un compromiso entre mayor rapidez y mayor seguridad.
- Longitud de clave variable: la longitud de la clave es un tercer parámetro. Otra vez, posibilita un acuerdo entre velocidad y seguridad.
- Simple: la estructura simple del RC5 es fácil de implementar y facilita la tarea de determinar la robustez del algoritmo.
- Bajo consumo de memoria: la poca necesidad de memoria hace que el RC5 sea adecuado para tarjetas inteligentes y otros dispositivos con restricciones de memoria.
- Alta seguridad: proporciona alta seguridad con los parámetros adecuados.
- Rotaciones dependientes de los datos: incorpora rotaciones (desplazamientos circulares de bits), cuya cantidad depende de los datos. Esto parece fortalecer el algoritmo contra el criptoanálisis [42].

7.14. PRINCIPIOS DE CRIPTOGRAFÍA DE CLAVE PÚBLICA

De igual importancia que el cifrado convencional es el cifrado de clave pública, que se emplea en autenticación de mensajes y en distribución de claves.

Estructura del cifrado de clave pública

El cifrado de clave pública, propuesto por primera vez por Diffie y Hellman en 1976, es el primer avance realmente revolucionario en el cifrado en miles de años. El motivo es que los algoritmos de clave pública están basados en funciones matemáticas y no en simples operaciones sobre los patrones de bits. Además, la criptografía de clave pública es asimétrica, lo que implica el uso de dos claves separadas, a diferencia del cifrado simétrico convencional, que emplea solo una clave.

El uso de dos claves tiene importantes consecuencias en el terreno de la confidencialidad, la distribución de claves y la autenticación. Antes de continuar, deberíamos mencionar algunas confusiones comunes en lo relativo al cifrado de clave pública. La primera es la creencia de que el cifrado de clave pública es más seguro ante el criptoanálisis que el cifrado convencional.

De hecho, la seguridad de cualquier esquema de cifrado depende de la longitud de la clave y el coste computacional necesario para romper un cifrado. No hay nada sobre el cifrado convencional ni de clave pública que haga a uno superior al otro en lo que respecta a la resistencia al criptoanálisis.

Otra equivocación la hallamos en la idea de que el cifrado de clave pública es una técnica con propósitos generales que ha dejado desfasado el cifrado convencional. Por el contrario, debido al coste computacional de los esquemas actuales de cifrado de clave pública, no parece que el cifrado convencional vaya a abandonarse.

Por último, se piensa que la distribución de claves no es importante cuando se usa el cifrado de clave pública, en comparación con los incomodos acuerdos previos que tienen lugar con los centros de distribución de claves para el cifrado convencional. De hecho, se necesita alguna forma de protocolo, que a menudo implica a un agente central, y los procedimientos que tienen lugar no son más sencillos ni más eficientes que los que se requieren para el cifrado convencional.
Un esquema de cifrado de clave pública tiene seis componentes (figura 13):

- **Texto claro:** consiste en el mensaje o los datos legibles que se introducen en el algoritmo como entrada.
- **Algoritmo de cifrado:** el algoritmo de cifrad realiza diferentes transformaciones en el texto claro.
- **Clave pública y privada:** es una pareja de claves que han sido seleccionadas, de las cuales una se usa para el cifrado y la otra para el descifrado. Las transformaciones exactas llevadas a cabo por el algoritmo de cifrado dependen de la clave pública o privada que se proporciona como entrada.
- **Texto cifrado:** es el mensaje desordenado producido como salida. Depende del texto claro y de la clave. Para un mensaje dado, dos claves diferentes producirán dos textos cifrados diferentes.

- **Algoritmo de descifrado:** este algoritmo acepta el texto cifrado y la clave correspondiente y produce el texto claro original.

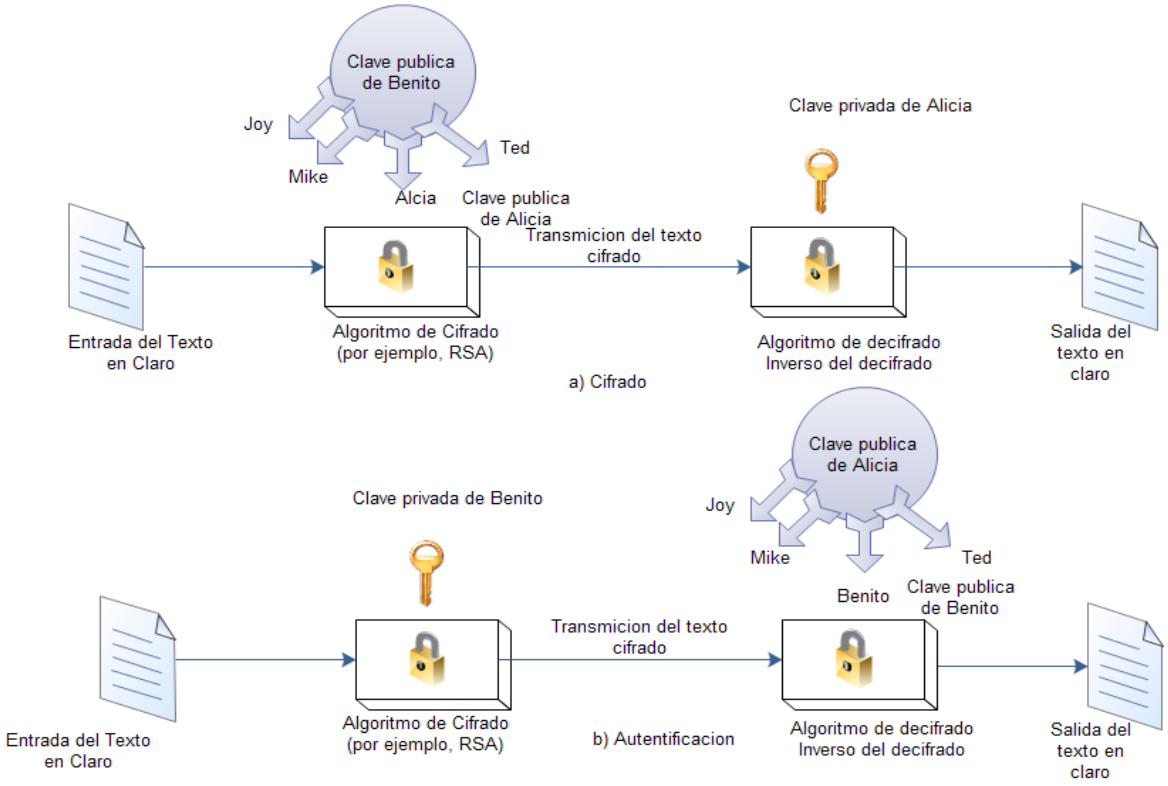


Figura 13: Criptografía de la clave pública

Aplicaciones para criptosistemas de clave pública

Los sistemas de clave pública se caracterizan por el uso de un tipo de algoritmo criptográfico con dos claves, una no se revela y la otra sí. Dependiendo de la aplicación, el emisor usa su clave privada o la clave pública del receptor, o las dos, para realizar algún tipo de función criptográfica. En términos generales, podemos clasificar el uso de criptosistemas de clave pública en tres categorías:

- **Cifrado/descifrado**: el emisor cifra un mensaje con la clave pública del receptor.
- **Firma digital**: el emisor firma un mensaje con su clave privada. Esto se consigue mediante un algoritmo criptográfico aplicado al mensaje o a un pequeño bloque de datos que es una función del mensaje.
- **Intercambio de llaves**: dos partes cooperan para intercambiar una clave de sesión. Hay distintas posibilidades que implican la clave privada de una o de las dos partes.

Algunos algoritmos son adecuados para las tres aplicaciones, mientras otros solo se pueden usar para una o dos de ellas. La tabla 8 indica las aplicaciones que soportan los algoritmos, la tabla también incluye el Estándar de Firma Digital (DSS) y la criptografía de curva elíptica.

Algoritmo	Cifrado/descifrado	Firma digital	Intercambio de clave
RSA	Sí	Sí	Sí
Diffie-Hellman	No	No	Sí
DSS	No	Sí	No
Curva elíptica	Sí	Sí	Sí

Tabla 8: Aplicaciones que soportan los algoritmos

Requisitos para la criptografía de clave pública

El criptosistemas que se muestra en la figura 13 depende de un algoritmo criptográfico basado en dos Claves relacionadas. Diffie y Hellman postularon este sistema sin demostrar que tales algoritmos existen. Sin embargo, si especificaron las condiciones que deben cumplir:

1. Desde el punto de vista computacional, para una parte B es fácil generar una pareja de claves (clave pública KU_b, clave privada KR_b.
2. En términos computacionales, para un emisor A que conozca la clave pública y el mensaje que ha de cifrarse, M, es fácil generar el texto cifrado correspondiente:

$$C = E_{KUb}(M)$$

3. En términos computacionales, para un receptor B es fácil descifrar el texto cifrado resultante usando la clave privada para recuperar el mensaje original:

$$M = D_{KRb}(C) = C = D_{Krb}[E_{KUb}(M)]$$

4. Desde el punto de vista computacional, es imposible que un oponente, conociendo la clave pública, KU_b, determine la clave privada KR_b.
5. Desde el punto de vista computacional, es imposible que un oponente, conociendo la clave pública, KU_b, y un texto cifrado, C, recupere el mensaje original, M.
6. Podemos añadir un sexto requisito que, aunque útil, no es necesario para todas las aplicaciones de clave pública:

Cualquiera de las dos claves relacionadas puede usarse para el cifrado, y la otra para el descifrado.

$$M = D_{KRb}\ [E_{KUb}(M) = D_{KUb}\ [E_{KRB}(M)]$$

7.15. ALGORITMOS DE CRIPTOGRAFÍA DE CLAVE PÚBLICA

Los dos algoritmos de clave pública más usados son el RSA y el DiffieHellman, los cuales, se analizaran a continuación.

El algoritmo de cifrado de clave pública RSA

El RSA es un cifrado de bloque en el que el texto claro y el texto cifrado son enteros entre 0 y n – 1 para algún n. Para algún bloque de texto claro M y un bloque de texto cifrado C, el cifrado y el descifrado son de la siguiente forma:

$$C=M^e \ mod \ n$$

$$M = C^e \ mod \ n = (M^e)^d mod \ n \ M^{ed} \ mod \ n$$

Tanto el emisor como el receptor deben conocer los valores de n y e, y solo el receptor conoce el valor de d. este es un algoritmo de cifrado de clave pública con una clave pública de KU = {e, n} y una clave privada de KR = {d, n}. Para que este algoritmo sea satisfactorio para el cifrado de clave pública, se deben cumplir los siguientes requisitos:

1. Que sea posible encontrar valores de e, d, n tal que M^{ed} = M mod n para todo M < n.
2. Que sea relativamente fácil calcularM^eyC^dpara todos los valores de M < n.
3. .Que sea imposible determinar d dados e y n.

Los dos primeros requisitos se cumplen fácilmente. El tercero se puede cumplir para valores grandes de e y n.

La figura 14 resume el algoritmo RSA. Se empieza seleccionando dos números primos, p y q, y calculando su producto n, que es el modulo para cifrado y descifrado. A continuación, se necesita la cantidad Φ(n), conocida como función totient de Euler de n, que es el número de enteros positivos menor que n y primo relativo de n. Luego, se selecciona un entero de que sea primo relativo de Φ(n) [el mayor común divisor de y Φ(n) es 1]. Finalmente, se calcula d como el inverso multiplicativo de e, módulo Φ(n). Se puede demostrar que d y e tienen las propiedades deseadas.

Generación clave

Seleccionar p, q	p y q primos, $p \neq q$
Calcular $n = p \times q$	
Calcular $\phi(n) = (p-1)(q-1)$	
Seleccionar entero e	$gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calcular d	$de \bmod \phi(n) = 1$
Clave pública	$KU = \{e, n\}$
Clave privada	$KR = \{d, n\}$

Cifrado

Texto claro:	$M < n$
Texto cifrado:	$C = M^e \pmod n$

Descifrado

Texto cifrado:	C
Texto claro:	$M = C^d \pmod n$

Figura 14: Algoritmo RSA

Supongamos que el usuario A ha publicado su clave pública y que el usuario B quiere enviar el mensaje M a A. Entonces B calcula C= $M^e(mod\ n)$ y transmite C. Al recibir el texto cifrado, el usuario A descifra calculando M=$C^d(mod\ n)$.

1. Seleccionar dos números primo , P=17 y q=11
2. Calcular n= pq =17*11=187
3. Calcular Ø(n)=)(p-1)(q-1)= 16*10=160
4. Seleccionamos e tal que e es un primo relativo de Ø (n)= 160 y menor que Ø(n); elegimos e=7.
5. Determinamos d, mod 160 =1 y d< 160. El valor correcto es d=23, porque 23*7=161=10*160+1.

Las claves resultantes son claves publicas KU= {7,187} y la clave privada KR {23, 187}. El ejemplo muestra el uso de estas claves para una entrada de texto claro de M=88. Para el cifrado necesitamos calcular C=88^7 mod 187. Teniendo en cuenta las propiedades de la aritmética modular, podemos realizarlo como sigue:

$$88^7\ mod\ 187 = [(88^4\ mod\ 187) * (88^2\ mod\ 187) * (88^1\ mod\ 187)] mod\ 187$$

$$(88^1\ mod\ 187) = 88$$

$$(88^2\ mod\ 187) = 7744\ mod\ 187 = 77$$

$$(88^4\ mod\ 187) = 59{,}969\ mod\ 187 = 132$$

$88^7\ mod\ 187 = (88 * 77 * 132) mod\ 187\ 894{,}432\ mod\ 187 = 11$

Para el descifrado calculamos M=$11^{23}\ mod\ 187$:

$$11^{23}\ mod\ 187 = [(11^1\ mod\ 187) * (11^2\ mod\ 187) * (11^4\ mod\ 187) * (11^8\ mod\ 187) * (11^8\ mod\ 187)$$

$(11^1\ mod\ 187) = 11$

$(11^2\ mod\ 187) = 121$

$(11^4\ mod\ 187) = 14{,}641\ mod\ 187 = 55$

$(11^8\ mod\ 187) = 214{,}358{,}881\ mod\ 187 = 33$

$(11^{23}\ mod\ 187) = (11 * 121 * 55 * 33 * 33)\ mod\ 187 = 79{,}720{,}245\ mod\ 187 = 88$

Figura 15: Ejemplo del algoritmo RSA

Hay dos enfoques posibles para romper el algoritmo RSA. El primero es el enfoque de fuerza bruta: intentar todas las claves privadas posibles. Así, cuanto mayor sea el número de bits en e y d, más seguro será el algoritmo. Sin embargo, debido a que los cálculos que tienen lugar tanto en la generación de la clave como en el cifrado/descifrado son complejos, cuanto mayor sea el tamaño de la clave, más lento ir al sistema.

La mayoría de las discusiones sobre criptoanálisis del RSA se han centrado en la tarea de factorizar n en sus dos factores primos. Un numero n producto de dos números primos grandes es difícil factorizar, aunque no tanto como solía ser.

Una ilustración llamativa de ello ocurrió en 1977; los tres inventores de RSA lectores de Scientific American a descodificar un cifrado que publicaron en la columna de juegos matemáticos de Martin Gardner. Ofrecieron una recompensa de 100 dólares por la recuperación de una frase de texto claro, algo que, según predijeron, no ocurriría durante unos 40 cuatrillones de años.

En abril de 1994, un grupo que trabajaba en Internet y que usaba unos 1.600 computadores consiguió el premio después de ocho meses de trabajo. En este reto se usó un tamaño de clave pública (longitud de n) de 129 dígitos decimales, alrededor de 428 bits.

Este resultado no invalida al RSA; simplemente significa que debe usarse un tamaño de clave mayor. Actualmente, un tamaño de clave de 1024 bits (300 dígitos decimales aproximadamente) se considera lo suficientemente robusto para casi todas las aplicaciones [42].

Intercambio de clave Diffie – Hellman

El primer algoritmo de clave pública apareció en el artículo de Diffie y Hellman que definía la criptografía de clave pública y que se conoce por el intercambio de clave Diffie – Hellman. Una serie de productos comerciales empleados emplearon esta técnica de intercambio de claves. La finalidad del algoritmo es hacer posible que los usuarios intercambien de forma segura una clave secreta que luego pueda ser usada para el cifrado posterior de mensajes. El algoritmo está limitado al intercambio de claves. El algoritmo de Diffie – Hellman depende para su efectividad de la dificultad de computar logaritmos discretos.

La seguridad del intercambio de claves de Diffie – Hellman se basa en el hecho de que, aunque es relativamente fácil calcular exponenciales modulo un número primo, es muy difícil calcular logaritmos discretos. Para números primos grandes, la última tarea se considera imposible.Aunque sí Diffie-Hellman acuerdo de claves es un anónimo (no autenticado) protocolo de clave-acuerdo, que proporcional base para una variedad de protocolos autenticados, y se utiliza para proporcionar confidencialidad directa perfecta de Transport Layer Security modos efímeros ‘s (denominados como EDH o DHE en función del conjunto de cifrado).

Descripción

Diffie-Hellman establece un secreto compartido que puede ser utilizado para las comunicaciones secretas mientras que el intercambio de datos a través de una red pública. El siguiente diagrama ilustra la idea general del intercambio de claves mediante el uso de colores en lugar de un número muy grande. La parte crucial del proceso es que Alice y Bob intercambian sus colores secretos en una mezcla única. Por último, este genera una llave idéntica que es matemáticamente difícil (imposible modernos superordenadores que hacer en un plazo razonable de tiempo) para neutralizar por otro partido que podría haber estado escuchando sobre ellos. Alice y Bob utilizan ahora este secreto común para cifrar y descifrar sus datos enviados y recibidos.

Tenga en cuenta que el color inicial (amarillo) es arbitrario, pero se acordó de antemano por Alice y Bob. El color inicial se supone que se sabe que cualquier oponente escuchas. Incluso puede ser pública.

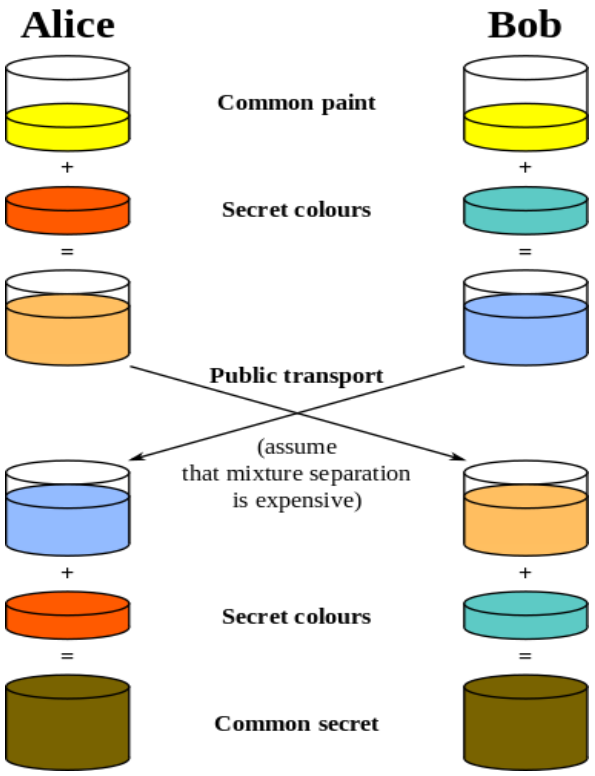


Figura 16: Intercambio de claves DiffieHellman

El más simple y la aplicación original del protocolo utilizan el grupo multiplicativo de enteros módulo *p*, donde es primo y *g* es la raíz primitiva mod *p*. He aquí un ejemplo del protocolo, con valores que no son secretas en azules valores y secretas en rojo. Números enteros pequeños se utilizan para mayor claridad, pero las implementaciones reales requieren el uso de un número mucho mayor de alcanzar la seguridad [44].

	Alicia			Bob		
Secreto	Publico	Calcula	Envía	Calcula	Publico	Secreto
un	p , g		p, g→			B
Un	p , g, A	$g^u = \mod p = A$	La→		p , g	B
Un	p , g, A		←B	$g^b = \mod p = B$	p ,g, A, B	B
Un, s	p ,g , A, B	$B^{un} = \mod p = s$		$A^b = \mod p = s$	p ,g, A, B	b, s

Tabla 9: Algoritmo del intercambio de claves

1. Alice y Bob están de acuerdo en utilizar un número primo $p = 23$ y la base $g = 5$.
2. Alice elige un entero secreto $a = 6$, a continuación, envía a Bob $A = g^{un} \bmod p$

   - $A = 5^{6} \bmod 23$
   - $A = 15625 \bmod 23$
   - $Una = 8$

3. Bob elige un entero secreto $b = 15$, a continuación, envía Alice $B = g^{un} \bmod p$

   - $B = 5^{15} \bmod 23$
   - $B = 30517578125 \bmod 23$
   - $B = 19$

4. Alice calcula $s = B^{un} \bmod p$

   - $S = 19^{6} \bmod 3$
   - $S = 47045881 \bmod 23$
   - $S = 2$

5. Bob calcula $s = A^{b} \bmod p$

1.4 $S = 8^{15} \bmod 23$

1.5 $S = 35.184.372.088.832 \bmod 23$

1.6 $S = 2$

6. Alice y Bob ahora comparten un secreto (la número 2) debido a 6 × 15 es el mismo que 15 × 6.

Tanto Alice y Bob han llegado al mismo valor, debido a que $(g^{un})^{b}$ y $(g^{b})^{una}$ son iguales mod p . Tenga en cuenta que sólo *un*, *b*, y $(g^{ab} \bmod p = g^{ba} \bmod p)$ se mantienen en secreto. Todos los otros valores - p, g, $g^{un} \bmod$ y $g^{b} \bmod p$ se envían en el claro. Una vez que Alice y Bob calcular el secreto compartido que pueden usarlo como una clave de cifrado, conocido sólo por ellos, para el envío de mensajes a través del mismo canal de comunicación abierto.

VIII. HERRAMIENTAS IMPLEMENTADAS EN REDES SOCIALES PARA PRIVACIDAD

8.1. TOR

The Onion Router, en su forma abreviada Tor, es un proyecto cuyo objetivo principal es el desarrollo de una red de comunicaciones distribuida de baja latencia y superpuesta sobre internet en la que el encaminamiento de los mensajes intercambiados entre los usuarios no revela su identidad, es decir, su dirección IP (anonimato a nivel de red) y que, además, mantiene la integridad y el secreto de la información que viaja por ella. Por este motivo se dice que esta tecnología pertenece a la llamada darknet o red oscura también conocida con el nombre de deep web o web profunda.

Tor es una red superpuesta sobre internet que está distribuida para desarrollo, formación y proporcionar un despliegue de una red anónima para las comunicaciones en baja latencia. Está diseñada para conseguir que el encaminamiento de los mensajes entre los host mantenga su privacidad y no se releve la información de los usuarios. Tor se basa en el establecimiento de un sistema virtual utilizando un circuito por capas, un circuito de enrutamiento de cebolla (Goldschlag et al, 1996; Reed et al, 1998.; Syverson et al., 2001).

Enrutamiento cebolla

Enrutamiento cebolla (OR) es una técnica para el anonimato de comunicación a través de una red informática. Los mensajes se cifran en varias ocasiones y luego se envían a través de varios nodos de red llamados enrutadores cebolla. Como alguien pela una cebolla, cada router cebolla elimina una capa de cifrado para descubrir instrucciones de ruta, y envía el mensaje a la siguiente enrutador donde esto se repite. Esto evita que estos nodos intermedios de conocer el origen, destino y contenido del mensaje.

Enrutamiento cebolla fue desarrollado por Michael G. Reed, Paul F. Syverson, y David M. Goldschlag y patentado por la Marina de los Estados Unidos en la patente de EE.UU. N 6.266.704 (1998). A partir de 2009, Tor es la tecnología predominante que emplea el enrutamiento cebolla.

Una cebolla de enrutamiento es una estructura de datos formada por ‘envolver’ un mensaje de texto simple con capas sucesivas de cifrado, de tal manera que cada capa puede ser ‘envolver’ como la capa de una cebolla por un intermediario en una sucesión de intermediarios, con el mensaje original de texto claro único ser visible en la mayoría [26]:

- el remitente
- el último intermediario
- el destinatario

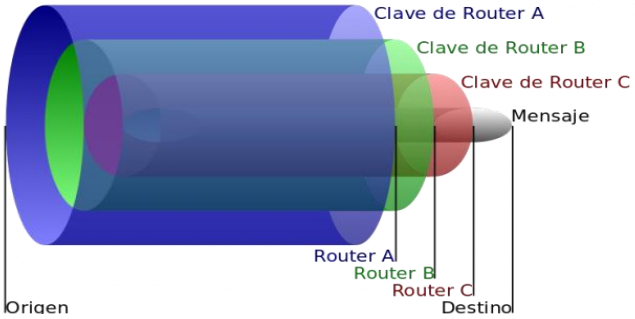


Figura 17: Enrutamiento de Cebolla

El encaminamiento de cebolla aprovecha la idea de Chaum de esconder la relación entre el origen y el destino de una información encapsulando los mensajes en capas de criptografía de clave pública. Aplicando esta idea para implementar un modo de encaminamiento, lo que hacemos es transmitir mensajes con distintas capas de cifrado (por eso se dice que estas redes son 'de cebolla') sobre un camino compuesto por nodos de mezclado (mixes) convertidos ahora en routers (router de cebolla) donde cada router del camino lo que hace es descifrar ('pela' una capa de la cebolla), transmitir y reordenar los mensajes antes de transmitir al siguiente router.

Tor es un conjunto de herramientas para un amplio abanico de organizaciones y personas que quieren mejorar su seguridad en Internet.Usando Tor es posible realizar una conexión a un equipo sin que este o ningún otro tenga posibilidad de conocer el número de IP de origen de la conexión. La red Tor cifra la información a su entrada y la descifra a la salida de dicha red, con lo cual es imposible saber quién envió la información. Sin embargo, el propietario de un servidor de salida puede ver toda la información cuando es descifrada antes de llegar a Internet, por lo que aunque no pueda conocer el emisor sí que puede acceder a la información [27].

La idea de enrutamiento cebolla es proteger la privacidad del remitente y el destinatario de un mensaje, mientras que también proporciona protección para el contenido del mensaje, ya que atraviesa una red.

Enrutamiento cebolla logra esto de acuerdo con el principio de las cascadas de la mezcla de Chaum: viajan los mensajes desde la fuente al destino a través de una secuencia de la representación, que los mensajes de re-ruta en una trayectoria impredecible. Para evitar que un intruso espiando contenido de los mensajes, los mensajes se cifran entre los routers. La ventaja de enrutamiento cebolla es que no es necesario confiar en cada router cooperante; si cualquier router se ve comprometida, la comunicación anónima todavía se puede lograr. Esto se debe a que cada router en una red acepta mensajes, los vuelve a cifrar, y transmite a otro router cebolla. Un atacante con la habilidad de monitorear todos los routers cebolla en una red puede ser capaz de rastrear la ruta de un mensaje a través de la red, pero un atacante con capacidades más limitadas tendrá dificultades incluso si él o ella controlan routers en el camino del mensaje.

Tor sólo permite anonimizar tráfico TCP. Las aplicaciones acceden a la red tor a través del interfaz SOCKS lo cual significa que toda aplicación con soporte SOCKS puede usar tor para realizar comunicaciones anónimas sin necesidad de modificaciones adicionales. El cliente Tor recibe tráfico SOCKS desde nuestras aplicaciones y luego, de forma transparente, se encarga de comunicarse con los routers de la red Tor para enviar las peticiones y posteriormente devolvernos los resultados.

SOCKS es un protocolo que facilita el enrutamiento de paquetes que se envían entre un cliente y un servidor a través de un servidor proxy. Según la pila de protocolos OSI está en el nivel 5 (sesión). Según la pila de protocolos IP está en la capa de aplicación. En los primeros intentos de usar encaminamiento de cebolla se requería un proxy de aplicación para cada protocolo de aplicación soportado. Esto conllevaba mucho trabajo y provocaba que algunos proxys no fueran escritos nunca y por tanto algunas aplicaciones nunca fueron soportadas. Tor usa SOCKS para, de un plumazo, soportar la mayoría de programas basados en TCP sin hacer ninguna modificación.

Observar que cuando navegamos por internet hacemos dos tipos de peticiones:

- Peticiones DNS para que el servidor de DNS que nos diga la dirección IP de una URL
- Peticiones HTTP a las direcciones IP del servidor web que aloja la información.

Si no pasamos por Tor las búsquedas con DNS que hacen los navegadores, pueden ser un problema de privacidad ya que si las peticiones se mandan directamente a través de la red regular un atacante podría deducir qué sitios se están visitando a través de Tor ya que antes de navegar por ellos se pregunta por DNS que IP tienen. Por tanto es necesario redirigir el tráfico de DNS por la red Tor.

Algunas aplicaciones convierten directamente el tráfico del protocolo la capa de aplicación en tráfico SOCKS. Por ejemplo Firefox permite convertir tanto el tráfico DNS como el HTTP a SOCKS y enviárselo al cliente Tor. Otras aplicaciones necesitan redirigir el tráfico del protocolo de la capa de aplicación hacia un proxy que realice la conversión al protocolo SOCKS. Por ejemplo si tuviéramos un navegador que no permitiera el tráfico HTTP y DNS vía SOCKS podría usar privoxy para realizar esta tarea (y podríamos aprovechar para filtrar las peticiones HTTP). Si tenemos una aplicación genérica que no soporta SOCKS y queremos que su tráfico TCP se convierta a formato SOCKS para luego pasarlo al cliente Tor es necesario utilizar una aplicación adicional.

Funcionamiento

La red está formada por una serie de nodos que se comunican mediante el protocolo TLS sobre TCP/IP manteniendo así secreta e íntegra (no modificaciones externas) la información desde un nodo a otro. Hay 2 tipos de entidades:

- **Nodos OR o simplemente OR** (del inglés Unión Router): Funcionan como encaminadores y en algunos casos además como servidores de directorio (DNS) de una especie de servicio de mantenimiento. Los nodos OR mantienen una conexión TLS con cada uno de los otros OR. Las conexiones OR-OR no son nunca cerradas deliberadamente salvo cuando pasa cierto tiempo de inactividad. Cuando un OR comienza o recibe nueva información de directorio él intenta abrir nuevas conexiones a cualquier OR que no esté conectado.
- **Nodos OP o simplemente OP** (del inglés Onion Proxy): Los usuarios finales ejecutan un software local que hace la función de nodo OP y que su función es obtener información del servicio de directorio, establecer circuitos aleatorios a través de la red y manejar conexiones de aplicaciones del usuario. Los OP aceptan flujos TCP de aplicaciones de usuarios y las multiplexa a través de la red OR's. Las conexiones OR-OP no son permanentes. Un OP debería cerrar una conexión a un OR si no hay circuitos ejecutándose sobre la conexión y ha vencido cierto temporizador [32].
- **Servicio de directorio.** El servicio de directorio publica una base de datos que asocia a cada OR una serie de información (router descriptor). Esta información es accesible a todos los OR y a todos los usuarios finales y la usan para tener un conocimiento de la red. Si se tienen pocos servidores de directorio se corre el riesgo tener un punto cuyo fallo puede ocasionar el fallo del sistema completo. Por motivos de backup y de latencia los OR que dan el servicio de directorio mantienen duplicada la información pasándosela de unos a otros. Hay una serie de OR principales (autoridades de directorio) y luego hay otros secundarios que hacen de caches y backup (directory caches). Una lista de algunos servidores de directorio son distribuidos con TOR para facilitar la suscripción a la red (bootstrapping). Los servidores de directorio son en realidad un grupo establecido de ORs confiables. Para dar fiabilidad a la información que da el servicio de directorio as entradas son protegidas criptográficamente con firmas y sólo la información que proviene de ORs aprobados será publicada en la base de datos. Por tanto todo nodo nuevo tiene que ser previamente aprobado y de esta forma se evitan ataques en los que alguien añade muchos nodos no confiables. No hay sistema automático para aprobar OR's; Los administradores del servidor de directorio lo hace manualmente.

Cuando un OR se arranca, recolecta un conjunto de datos que lo describen a él, a su modo de funcionamiento y capacidades. Ejemplos de este tipo de atributos son la dirección IP, nombre amigable para el usuario, versión del software TOR, sistema operativo, clave pública, exit policies (restricciones a cómo puede funcionar el nodo si es el último nodo de un circuito de datos Ej: definir una lista de direcciones IP y número de puertos a los cuales está dispuesto llevar el tráfico. Observar que usando esto se puede hacer que un nodo no pueda actuar como último nodo de un circuito nunca). Toda esta información se publica a través del servicio de directorio.

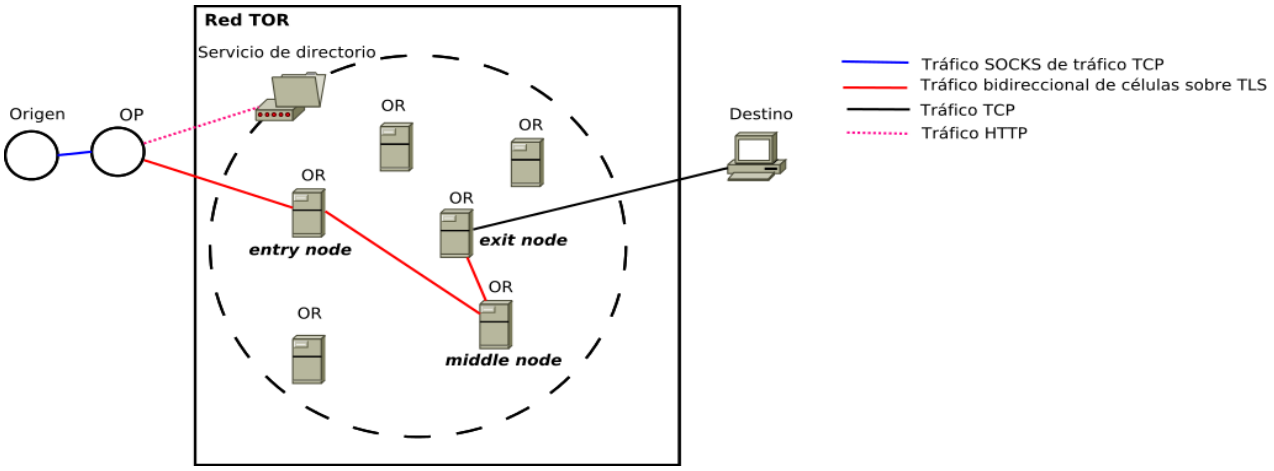


Figura 18: Funcionamiento de la red de TOR

Opinión de la NSA de TOR

The Guardián sigue con las filtraciones de la NSA, y la última se refiere a Tor. Este software permite navegar de forma anónima por Internet y acceder a servidores web ocultos, todo con tráfico cifrado y enrutado de tal forma que es muy difícil detectar la procedencia original.

Tor está basado en la técnica del enrutado cebolla o por capas. A grandes rasgos, lo que hace el cliente es enviar un paquete a un nodo de la red, este lo envía a un segundo nodo, el segundo a un tercer y así siguiendo una ruta más o menos aleatoria hasta que sale de la red Tor, llega a Internet y la respuesta vuelve al cliente original.

La NSA y el GCHQ han utilizado ataques conocidos junto con su poder e influencia para atacar y "desanonimizar" a usuarios de Tor. Sin embargo, no lo han conseguido con el 100% de usuarios ni han logrado atacar a usuarios específicos: por el contrario, lo que hacen es "poner trampas" y ver quién cae. Tampoco han roto la seguridad y cifrado de Tor, los ataques se dirigen más bien a otros programas que use el objetivo para "colarse" en su ordenador.

Lo que la NSA ha hecho con Tor se puede dividir en dos partes: por una, análisis para detectar a los usuarios y por otros ataques para comprometer sus ordenadores y poder espiarles sin problemas de forma continuada.

Análisis: desde usuarios “tontos” hasta revelación cookies

La NSA quiere saber quién está haciendo qué en Internet en todo momento, y Tor se lo pone difícil. Todas las técnicas de análisis están enfocadas a saber quién está detrás de cada petición de Tor, para o bien añadir esos datos a sus registros o bien efectuar más ataques sobre el objetivo posteriormente.

La primera parte consiste en distinguir si una petición a una página web viene de la red Tor o de un usuario normal. No es un paso difícil: Tor deja muchas huellas que hace que su detección sea sencilla. Por ejemplo, TorButton no desvela el número de compilación de Firefox que estás usando. Lo hace por motivos de privacidad, y porque no pasa nada si se detecta una petición Tor: lo que de verdad importa es que no se pueda distinguir entre varios usuarios Tor.

Lo siguiente que hay que hacer es saber quién hay detrás, y la NSA ha desarrollado varias técnicas de análisis.

En las presentación filtrada por The Guardián, la que más me ha llamado la atención es *Dumb users (EPICFAIL)*, o en castellano “Usuarios idiotas (fallo épico)”. La idea es sencilla: fijarnos en fallos de los usuarios que revelen su identidad. Por ejemplo, pongamos que un usuario, Bob, escribe normalmente en un foro bajo el seudónimo *bob8819*. Más tarde, empieza a usar Tor para visitar otras páginas. La NSA no sabe quién es el que está haciendo esas visitas, hasta que de repente Bob escribe en su foro habitual bajo su seudónimo de siempre usando Tor. Con esa información, Bob ha desvelado su identidad y ha permitido a la NSA vincular esas visitas desconocidas de Tor con otras visitas de las que ya conocía el origen.

En resumen, lo que hace la NSA es vincular el tráfico de un usuario Tor a otro tráfico normal (que sabemos de dónde viene) mirando qué partes comunes hay entre los dos. Aquí, las partes comunes eran nombres de usuario y correos. El siguiente nivel es usar las *cookies*.

La idea es aprovechar *cookies* que “sobrevivan” a la navegación con Tor, por ejemplo si se usa mal el navegador con Tor y no se borran al pasar a la navegación normal (no anónima). Según la presentación, las cookies de DoubleClick (red de anuncios) serían una buena forma para identificar usuarios Tor.

Por último, la NSA también podría estudiar la fecha, hora y lugar de conexión de un objetivo y después buscar conexiones a Tor con los mismos parámetros. Sin embargo, es difícil seleccionar los candidatos y los programas que están en marcha no son muy eficaces.

Estudiar la red Tor para descubrir a los usuarios originales. El otro tipo de técnicas de análisis se basan en estudiar la red Tor. La primera es sencilla de entender: la reconstrucción de circuitos.

Si la NSA controla todos los nodos de un circuito, podría vigilar por dónde va el paquete, desde que lo envía el usuario original hasta que sale a Internet. Sin embargo, controlan un número muy bajo de nodos, de tal forma que la probabilidad de que un paquete visite sólo los nodos de la NSA es ínfima, lo que hace la reconstrucción de circuitos casi imposible.

También existe el análisis de tiempos, que es igualmente simple. Por ejemplo, pueden detectar que cuando un paquete entra a la red Tor, sale otro a los 300 milisegundos en otra parte del mundo hacia el ordenador de nuestro amigo Bob. Si se repite mucho ese patrón de latencia, es muy posible que ese tráfico de Tor esté originado por Bob.

Instalación de tor

Tor es un conjunto de herramientas para un amplio abanico de organizaciones y personas que quieren mejorar su seguridad en Internet. Usando Tor es posible realizar una conexión a un equipo sin que este o ningún otro tenga posibilidad de conocer el número de IP de origen de la conexión. La red Tor cifra la información a su entrada y la descifra a la salida de dicha red, con lo cual es imposible saber quién envió la información. Sin embargo, el propietario de un servidor de salida puede ver toda la información cuando es descifrada antes de llegar a Internet, por lo que aunque no pueda conocer el emisor sí que puede acceder a la información.

Paso # 1 Vamos al sitio web https://www.torproject.org/

Paso # 2 Ahora nos descargamos la versión que nos interesa de acuerdo al sistema operativo y de la cantidad de bits de la máquina.

Language	Microsoft Windows *(3.6.1)*	Mac OSX *(3.6.1)*	Linux *(3.6.1)*
English (en-US)	32/64-bit *(sig)*	32-bit *(sig)*	32-bit *(sig)* • 64-bit *(sig)*
العربية (ar)	32/64-bit *(sig)*	32-bit *(sig)*	32-bit *(sig)* • 64-bit *(sig)*
Deutsch (de)	32/64-bit *(sig)*	32-bit *(sig)*	32-bit *(sig)* • 64-bit *(sig)*
Español (es-ES)	32/64-bit *(sig)*	32-bit *(sig)*	32-bit *(sig)* • 64-bit *(sig)*
فارسی (fa)	32/64-bit *(sig)*	32-bit *(sig)*	32-bit *(sig)* • 64-bit *(sig)*
Français (fr)	32/64-bit *(sig)*	32-bit *(sig)*	32-bit *(sig)* • 64-bit *(sig)*

Figura 19: Elección del archivo a descargar

Paso # 3 Y seleccionamos dando clic en la versión que corresponda al sistema operativo.

En nuestro caso nos descargamos el tor del tipo de sistema operativo Linux de 32 bits en español.

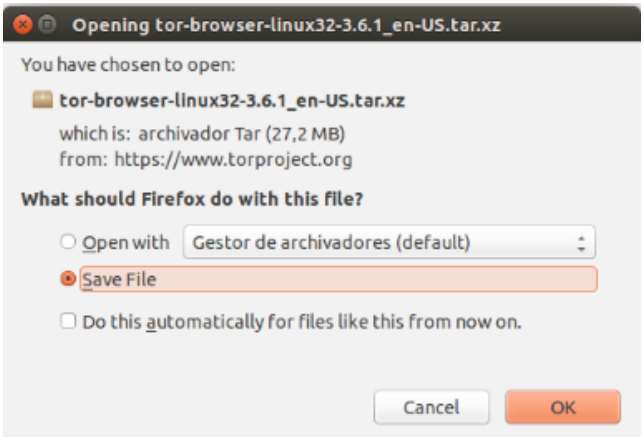


Figura 20: Descarga del archivo

Paso # 4 Vamos a la carpeta de Descargas y damos doble clic en el archivo torbrowser-linux32.3.6.1_en_US.tar.xz

Paso #5 Damos clic en Extraer

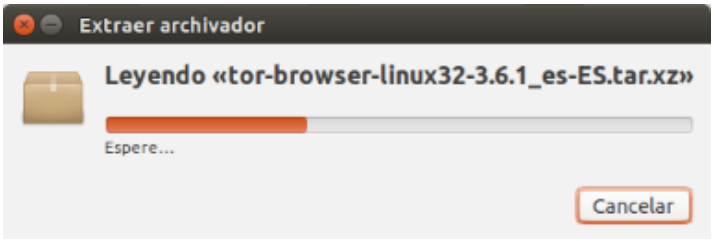


Figura 21: Extracción previo a ejecutar

Paso #6 Abrimos la carpeta torbrowser la que descomprimimos

Paso #7 Damos clic derecho sobre startorbrowser y damos clic en Abrir o lo ejecutamos desde una terminal. /**Startorbrowser** y este de forma automática abrirá el navegador tor.

```
#./ Startorbrowser
```

Este Conectara tu Computadora a la red de tor

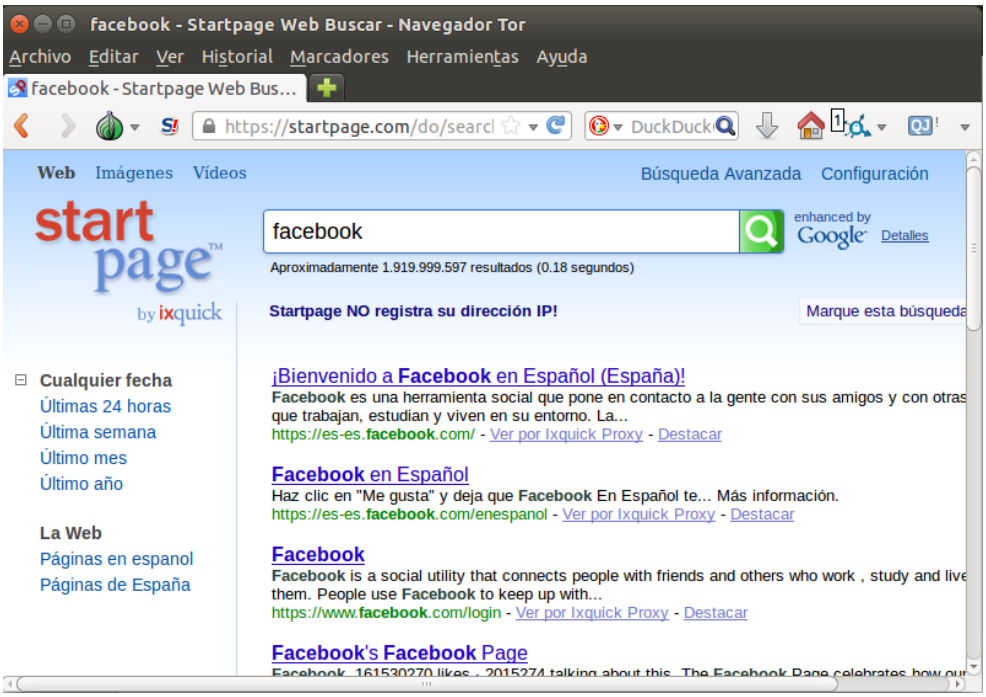


Figura 22: Navegando con el buscador por default de tor

Al navegar por la red de Tor, hay que usar motores de búsqueda específicos que nos permitan la privacidad en este caso hemos utilizado el motor de búsqueda de Start page este lo trae Tor por defecto, pero también podemos usar el duckduckgo que es un buscador similar al de google pero no google.

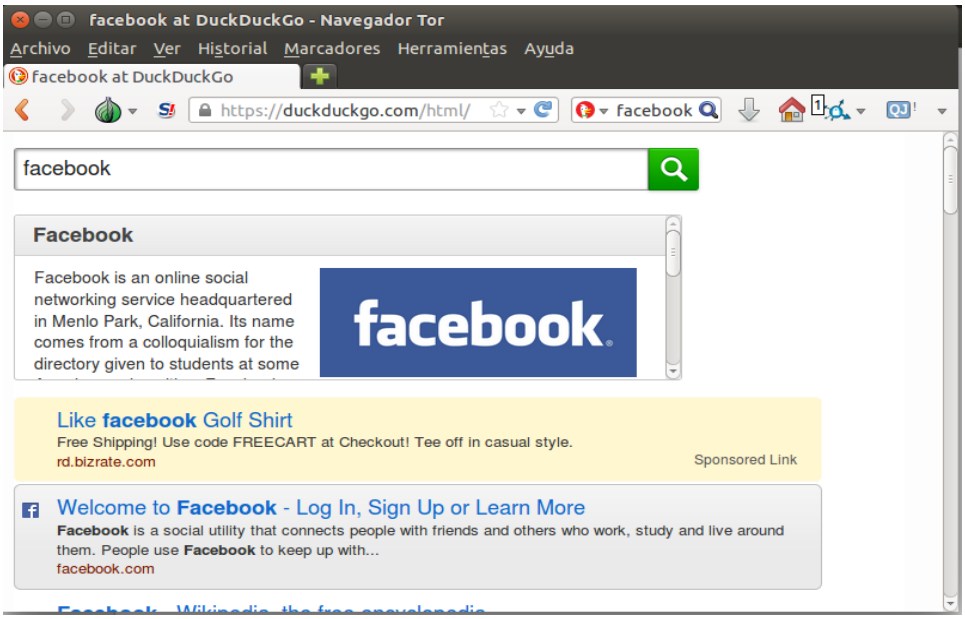


Figura 23: Usando el motor de búsqueda de duckduckgo.

Cuando le damos conectar a la red de tor de forma inmediata nos abre el explorador que está configurado con tor. Si deseamos configurar algo más desde la cebolla verde que se muestra indicando que tor está activo podemos configurar más como deseemos navegar o al iniciar en la pestañas preferencias.

La descarga recomendada para la mayoría de los usuarios es el Paquete de navegador Tor. Este paquete contiene un navegador preconfigurado para navegar en Internet de forma segura a través de Tor y no requiere instalación alguna. Usted descarga el paquete, descomprime el archivo e inicia Tor.

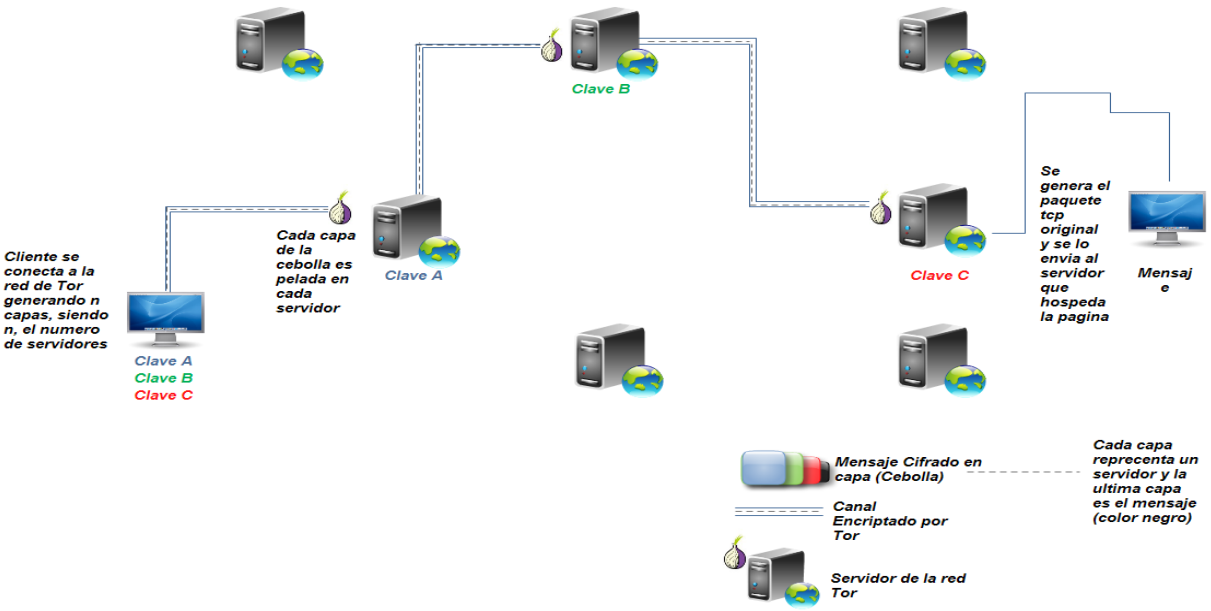


Figura 24: Esquema general del Funcionamiento de Tor

Existen dos maneras diferentes para obtener el software Tor:

Paso # 1 Vamos al sitio web https://www.torproject.org/ y descargamos el Tor

Figura 25: Página Tor

- Después de descargar el Paquete del Navegador Tor, extráigalo en su escritorio o en una memoria USB. Deberá obtener un directorio que contiene algunos archivos. Uno de éstos es un ejecutable llamado “Start Tor Browser” (o “Start-tor-browser”, dependiendo de su sistema operativo).

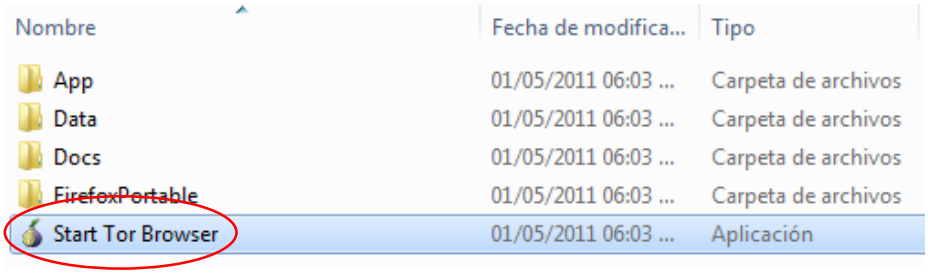


Figura 26: Ejecutando TOR en windows

- Ahora verá un navegador confirmando que usted se encuentre utilizando Tor. Esto se hace desplegando **https://check.torproject.org/**. Ahora ya puede navegar por Internet a través de Tor.

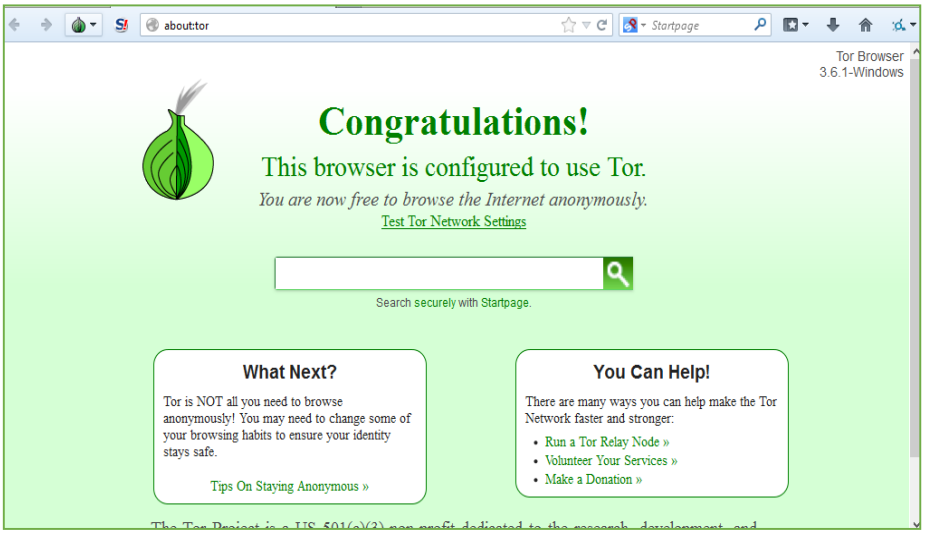


Figura 27: Navegando con Tor en windows

8.2. PROYECTO DE INTERNET INVISIBLE (I2P)

I2P (sigla para Invisible Internet Project, que significa Proyecto de Internet invisible) es un software que ofrece una capa de abstracción para comunicaciones entre ordenadores, permitiendo así la creación de herramientas y aplicaciones de red con un fuerte anonimato [53].

I2P es una red anónima, que presenta una capa simple que las aplicaciones pueden usar para enviarse mensajes entre sí de forma anónima y segura. La propia red está estrictamente basada en mensajes, pero hay una librería disponible para permitir comunicación en streaming confiable sobre ella (TCP). Toda comunicación está cifrada extremo a extremo (en total hay cuatro capas de cifrado usadas cuando se envía un mensaje), e incluso los puntos de los extremos (“destinos”) son identificadores criptográficos (esencialmente un par de claves públicas) [55].

Enrutamiento ajo (Garlic Routing)

Enrutamiento ajo es una variante de enrutamiento cebolla que encripta mensajes múltiples juntos para hacer que sea más difícil para los atacantes para realizar análisis de tráfico. El ajo es uno de los factores clave que distinguen I2P de Tor y otras redes de privacidad o cifrado.

"Garlic" fue quizás usado por los desarrolladores originales de I2P porque implementa un forma de agrupación tal y como Freedman describió, o simplemente para enfatizar las diferencias con Tor. La razón exacta puede haberse perdido en el tiempo. Generalmente, cuando nos referimos al término "garlic", puede significar una de estas tres cosas:

1. Cifrado por capas
2. Agrupa varios mensajes juntos
3. Cifrado ElGamal/AES

Desafortunadamente, el uso del término "garlic" por parte de I2P en los pasados siete años no ha sido siempre preciso; por lo tanto los usuarios están avisados para cuando se topen con el término. Esperemos que la siguiente explicación dejará las cosas más claras.

- Cifrado por capas: El enrutado Onion es una técnica para construir caminos, o túneles, a través de una serie de pares, y después usar ese túnel. Los mensajes son repetidamente cifrados por el creador, y después descifrados por cada salto. Durante la fase de construcción sólo las instrucciones de enrutado para el siguiente salto son expuestas a cada par. Durante la fase de operación los mensajes se pasan a través del túnel, y el mensaje y sus instrucciones de enrutado sólo son mostradas en el punto final del túnel
- Construyendo múltiples mensajes: Michael Freedman definición "garlic routing" como una extensión de onion routing, en la cual varios mensajes son agrupados juntos. El llamó a cada mensaje "bulb, bulbo". Todos los mensajes, cada uno con sus instrucciones de envío, son expuestos en el punto final. Esto permite una agrupación eficiente del "bloque de respuesta" de una ruta onion con el mensaje original.
- Cifrado ElGamal/AES: significa cifrado Gamal/AES+SessionTag (sin múltiples capas).

I2P usa rutado garlic, agrupación y cifrado en tres partes:

1. Para la construcción y enrutado a través de los túneles (cifrado por capas)
2. Para determinar el éxito o fallo del envío de un mensaje de fin a fin (agrupamiento)
3. Para publicar algunas entradas en la base de datos de la red (amortiguando la posibilidad de un ataque de análisis exitoso) (ElGamal/AES).

Hay muchas otras formas de usar esta técnica para mejorar el rendimiento de la red, aprovechando las compensaciones de la latencia del transporte, y derivando datos a través de caminos redundantes para incrementar la seguridad

Agrupación de mensajes de fin a fin.

En la capa por encima de los túneles, I2P envía mensajes de fin a fin entre destinaciones. Dentro de un único túnel usamos El Gamal/AES+SessionTag para el cifrado. Cada mensaje cliente enviado al router a través del interfaz I2CP se convierte un solo Garlic Clove con sus propias instrucciones de envío, dentro de un mensaje Garlic. Las instrucciones de envío pueden indicar una destinación, un router o un túnel [54].

Generalmente, un mensaje Garlic sólo contiene un clove. Aun así, el router agrupa periódicamente dos dientes adicionales en el mensaje Garlic:

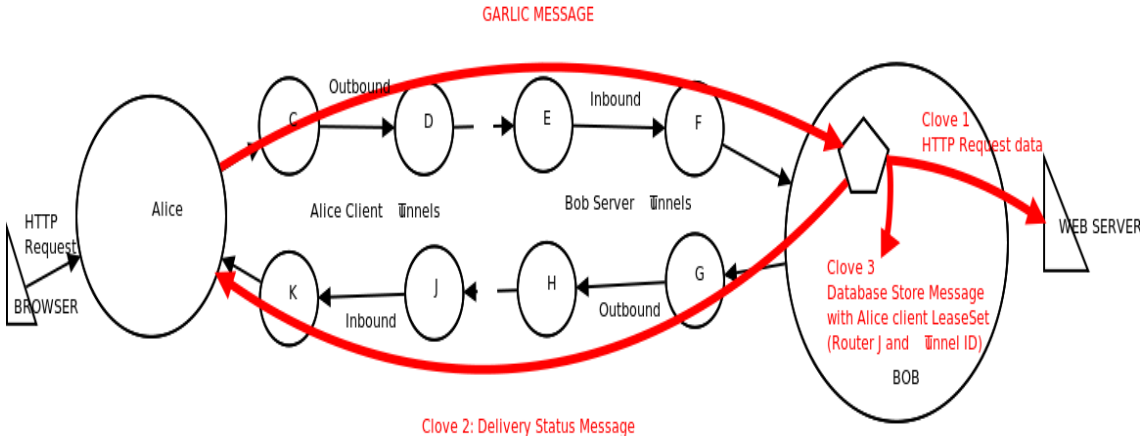


Figura 28: Mensaje Garlic

Funcionamiento de I2P

La red I2P está basada en el concepto de túneles entrantes y salientes, lo cual ofrece facilidad para la adaptación de programas preexistentes a la red I2P. Cada túnel está compuesto por una secuencia de nodos padres, los cuales transportan la información en un sentido unidireccional.

Para a anonimizar los mensajes enviados, cada aplicación cliente tiene su router I2P que crea unos cuantos túneles de entrada y salida una secuencia de pares que pasan el mensaje en una dirección (hacia y desde el cliente, respectivamente). A su vez, cuando un cliente quiere enviar un mensaje a otro cliente, el cliente envía ese mensaje a través de uno de sus túneles de salida hacia uno de los túneles de entrada del otro cliente, eventualmente alcanzando su destino.

La primera vez que un cliente quiere contactar con otro cliente, ambos hacen una consulta contra la, completamente distribuida, "base de datos de red" – una tabla de hash distribuida (DHT) con estructura adaptada basada en l algoritmo Kademlia.

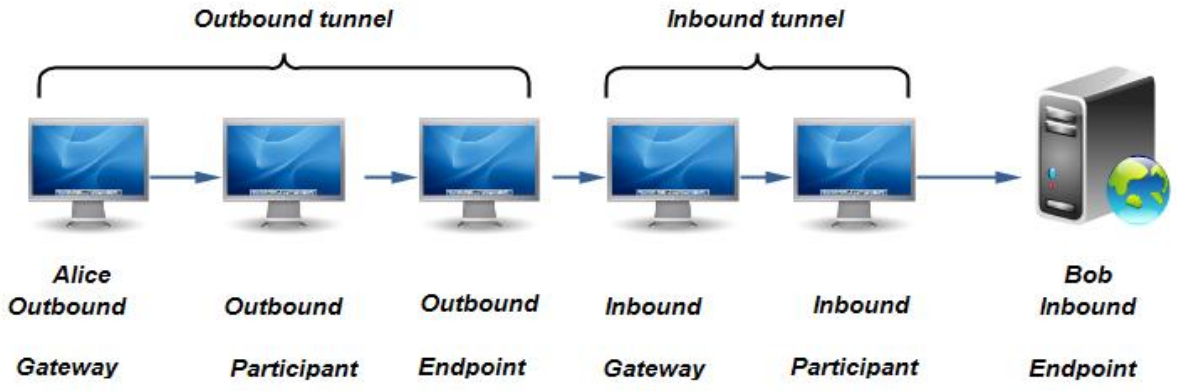


Figura 29: Funcionamiento de I2P

Esto se hace para encontrar eficientemente los túneles entrantes de los otros clientes, pero los mensajes subsiguientes entre ellos normalmente incluyen esos datos, así que no se requieren anteriores búsquedas en la base de datos de red [56].

Existen 2 tipos de túneles: túneles "outbound", de salida envían mensajes hacia fuera desde el creador de túneles, mientras que túneles "inbound", de entrada llevan los mensajes hasta el creador de túneles. Combinando estos 2 túneles los usuarios pueden enviar mensajes a otros usuarios. El remitente ("Alice" en la imagen anterior) crea un túnel de salida, mientras que el receptor ("Bob" en la imagen de arriba) crea un túnel entrante. La puerta de salida de un túnel de entrada puede recibir mensajes de cualquier usuario que serán enviados hasta en punto final ("Bob"). El punto final del túnel de salida necesita enviar el mensaje a la puerta de salida del túnel de entrada. Para conseguir esto, el remitente ("Alice") añade instrucciones al mensaje cifrado. Una vez que el punto final y el túnel de salida descifran el mensaje, tendrán las instrucciones para poder enviar el mensaje la puerta de salida de entrada correcta (la puerta de salida hacia "Bob").

Un tercer concepto crítico es el de la "base de datos de la red" (o "netDb") - un par de algoritmos hechos para compartir los datos de la red. Los 2 tipos de datos compartidos son "routerInfo" y "leaseSets" - la información del router, ruterinfo, da a los routers la información necesaria para poder contactar con un router en particular (sus claves públicas, dirección de transporte, etc), mientras que el leaseSet le da a los routers un número de "leases", asignaciones. Cada una de estas asignaciones especifica una puerta de salida de un túnel, la cual permite alcanzar una destinación específica. La información completa en un lease es:

- Gateway de entrada para un túnel que permite alcanzar una destinación específica.
- Tiempo de espiración del túnel.

Par de claves públicas para poder cifrar el mensaje (para enviarlo a través del túnel y alcanzar su destino).

Los ruters envían su routerInfo directamente a la netDB, mientras que los leaseSets son enviados a través de los túneles de salida (los leaseSets tienen que ser enviados anónimamente para evitar la correlación del ruter con sus leaseSets) [56].

Instalación y configuración en Linux 14.04

Hay 2 formas de instalación:

1. Desde la terminal de ubuntu ejecutamos la siguiente orden:

sudo apt-add-repository ppa:i2p-maintainers/i2p

Este comando agregará el PPA al archivo: /etc/apt/sources.list.d ahora lo que hacemos es actualizar desde consola esta lista de repositorios ejecutando:

```
sudo apt-get update
```

Este comando recuperará la lista más reciente de software de cada repositorio que está habilitada en el sistema, incluyendo el I2P PPA que se ha ‘‘6``añadido con el comando anterior. Hecho esto hora estamos listos para instalar I2P.

```
sudo apt-get install I2P
```

2. Añadir el PPA usando Synaptic, para esto buscamos Synaptic y lo abrimos. Una vez se abra Synaptic, seleccione Repositorios en el menú Configuración.

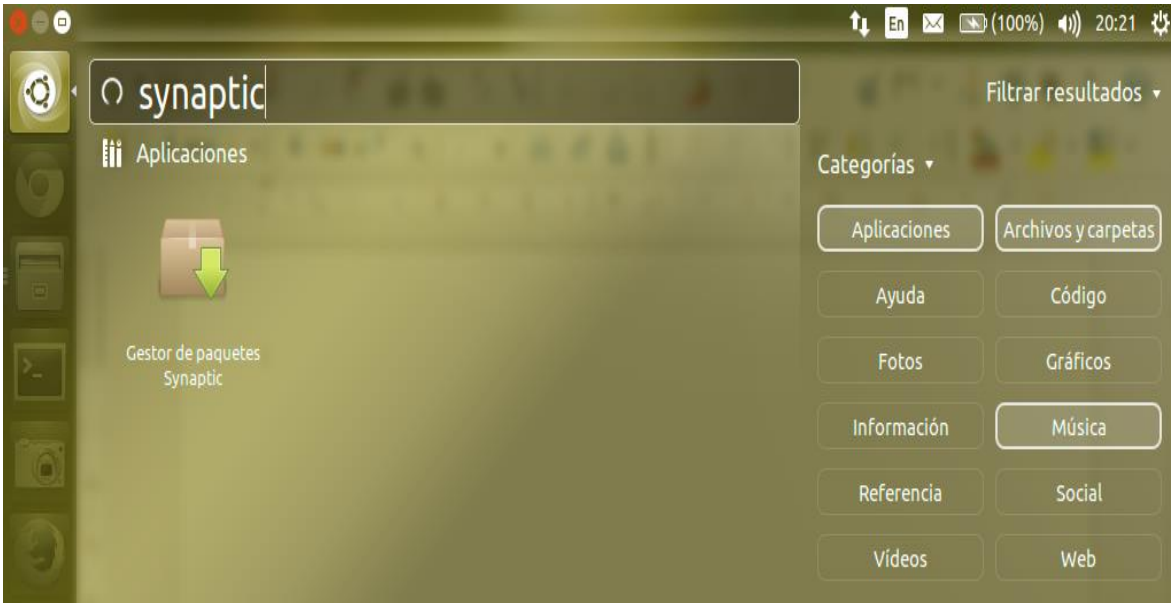


Figura 30: Gestor de paquetes Synaptic

3. Haga clic en la pestaña Otras fuentes y seleccione Añadir. Pegue ppa: i2p-maintainers/i2p en el campo la línea-APT y pulse Añadir fuente. Pulse el botón Cerrar y luego Recargar. En la casilla de Filtrado Rápido escriba i2p y pulse Intro. Cuando devuelva i2p en la lista de resultados, haga clic con el botón secundario sobre i2p y seleccione Marcar para instalación.

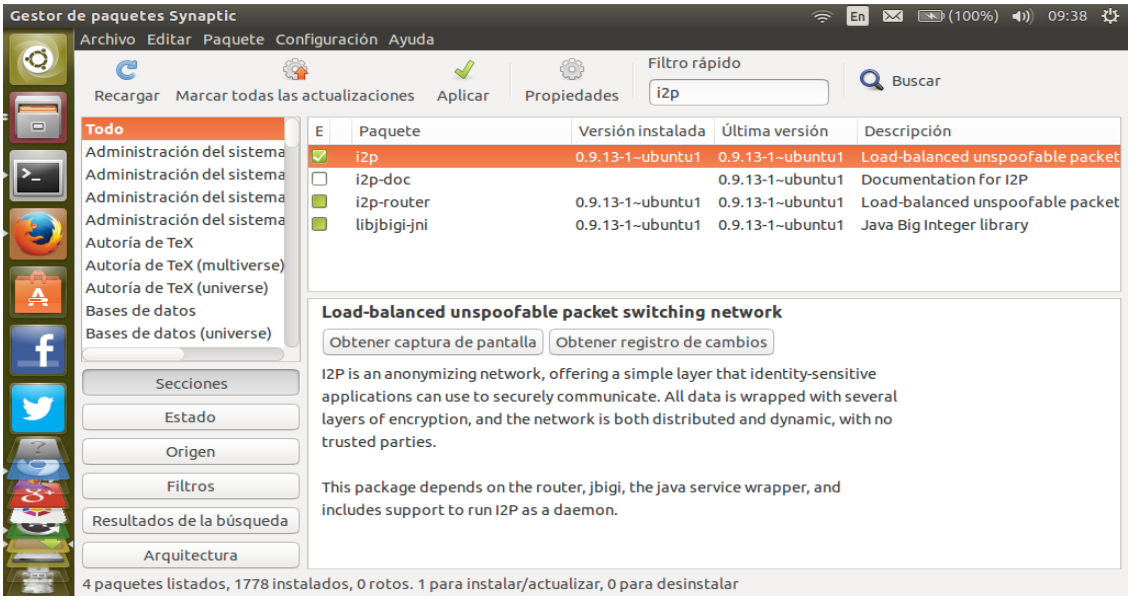


Figura 31: Seleccione Marcar para instalación

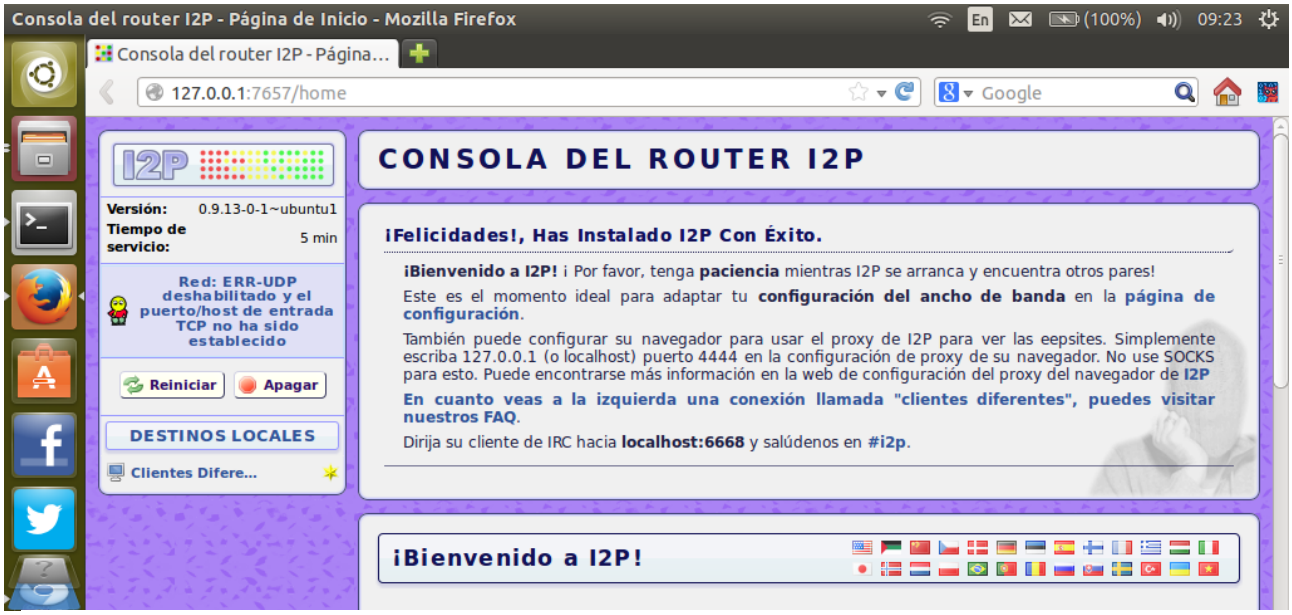


Figura 32: Páginas de inicio de I2p

Después de hacer esto puede que vea un cuadro emergente que diga ¿Marcar cambios adicionales?. Si es así, pulse Marcar y luego Aplicar. Tras completar el proceso de instalación puede pasar a la siguiente parte de la inicialización de I2P, y configurarlo para su sistema. El router I2P se puede iniciar usando estos paquetes I2P de las tres maneras siguientes: “a petición” usando el script i2prouter. Simplemente ejecute “i2prouter start” en una línea de comandos. (Nota: ¡No utilice sudo ni lo ejecute como root!).

Configuración del navegador web

El navegador web necesita ser configurado para poder navegar estos sitios y para utilizar los outproxies disponibles en I2P.Nos vamos a nuestro explorador de firefox.

Desde el menú de herramientas seleccione opciones y abra el menú de configuración de Firefox. Pulse en Avanzado, entonces pulse en la pestaña Red. En la sección de Conexiones, pulse el botón de Configuración.

En la ventana de *Configuración de Conexiones*, pulse el círculo junto a *Configuración manual del proxy,* entonces añada 127.0.0.1, puerto 4444 en el campo Proxy HTTP. Añada 127.0.0.1, puerto 4445 en el campo Proxy SSL. Asegúrese de escribir 127.0.0.1 o localhost en el campo “Sin proxy para”.

Realizado los pasos anterior esto quedaría de la siguiente manera como se muestran en las imágenes.

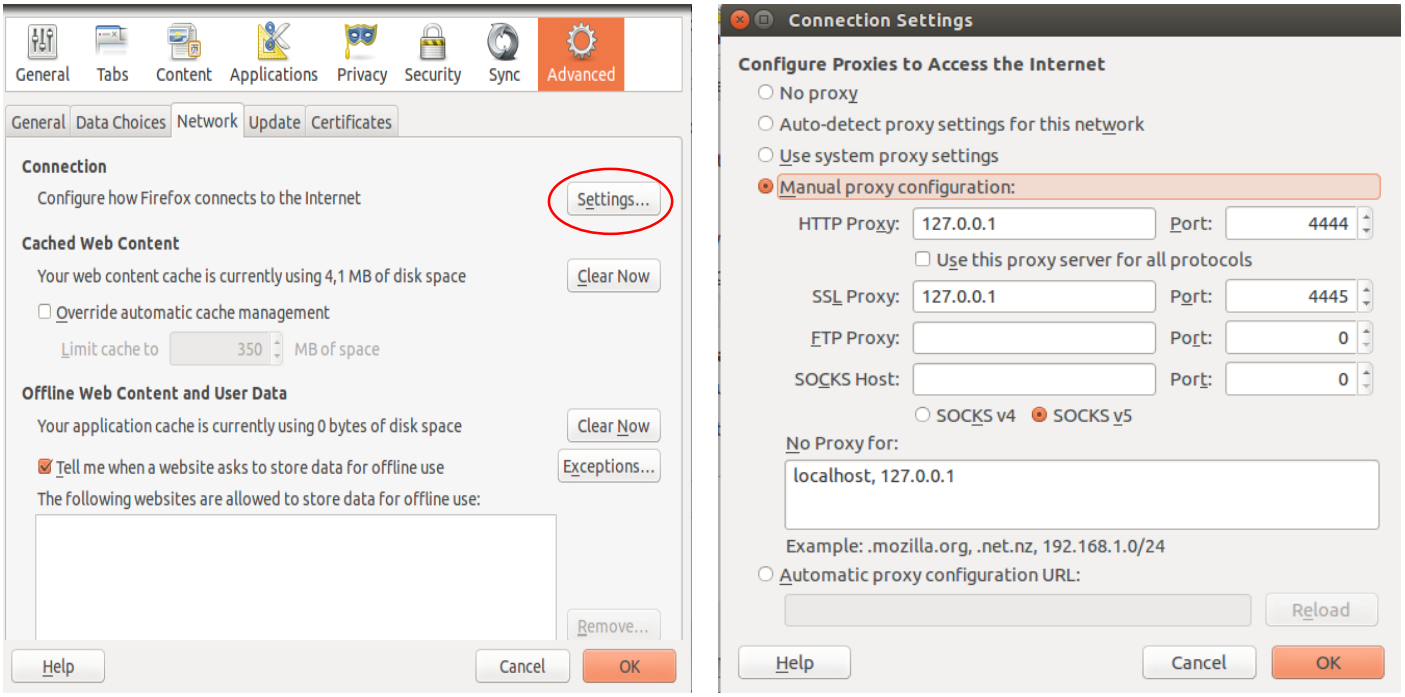


Figura 33: Configurar proxy en el navegador

Figura 34: Navegando con I2P

Con el navegador configurado ya podemos navegar de forma anónima.

Como podemos ver en la imagen estamos navegando de forma natural y sin ningún problema recuerda esto, si tú no tienes corriendo o ejecutando el i2prouter no podrás navegar en tu Firefox te saldrá de la siguiente manera:

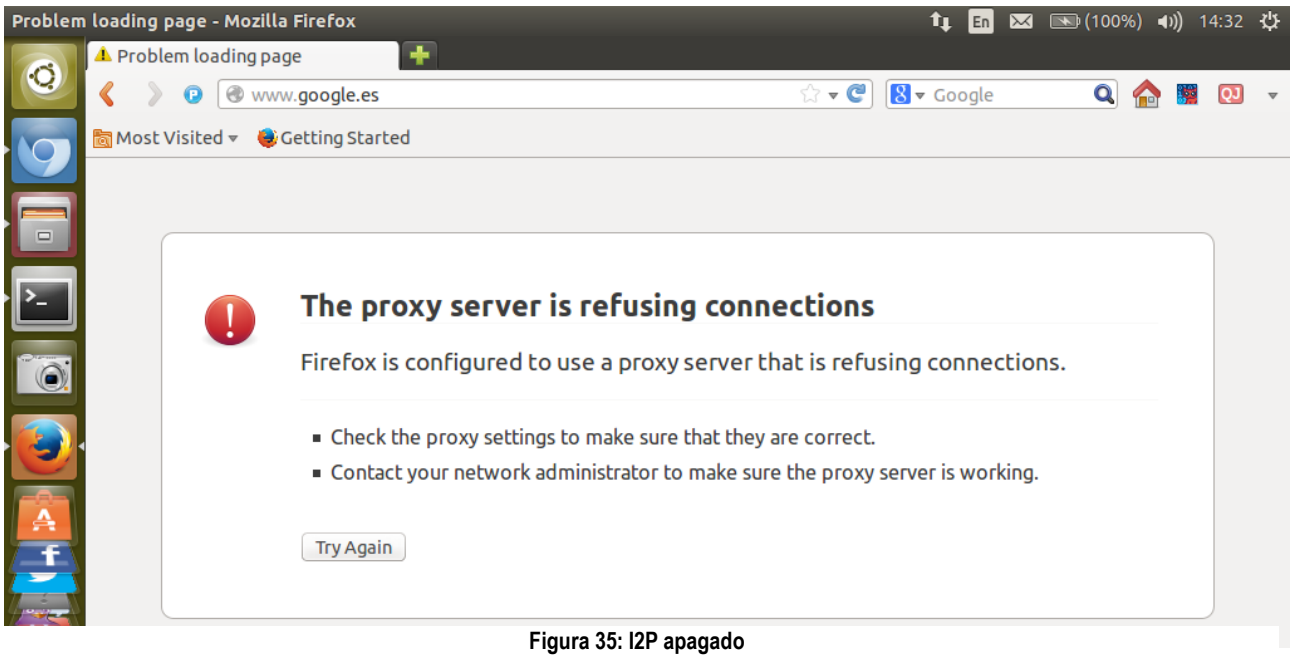


Figura 35: I2P apagado

Instalación y configuración en Windows 8.1

Lo primero es explicar por encima que TOR e I2P son cosas diferentes, básicamente TOR consigue una conexión anónima y encriptada para conectar a internet, pero I2P crea una red encriptada dentro de internet, pero con sus propias web, servidores de correo, etc., que son accesibles desde la mima red I2P pero no desde internet, y lo mismo a la contra, es decir desde una conexión mediante la red I2P no ves el internet "normal" por llamarlo de alguna manera, Así por encima más o menos esa es la diferencia.

Instalación y configuración de I2P en un navegador Chrome

Ahora procedemos a instalar I2P, lo descargamos el instalador grafico desde la página de I2P

http://mirror.i2p2.de/i2pinstall_0.8.6.exe

Una vez descargado lo ejecutamos y lo instalamos, En la primera ventana aparece para elegir el idioma lo demás no tiene complicación lo típico ir dando siguiente e ir aceptando los términos. Una vez terminado se abrirá nuestro navegador predeterminado con la página de I2P. Sino aparece habrá un nuevo icono al lado del reloj que pulsando el botón derecho nos da la opción de abrir dicha página. También podemos abrir nosotros el navegador y acceder a **http://127.0.0.1:7657** deberíamos de ver algo así:

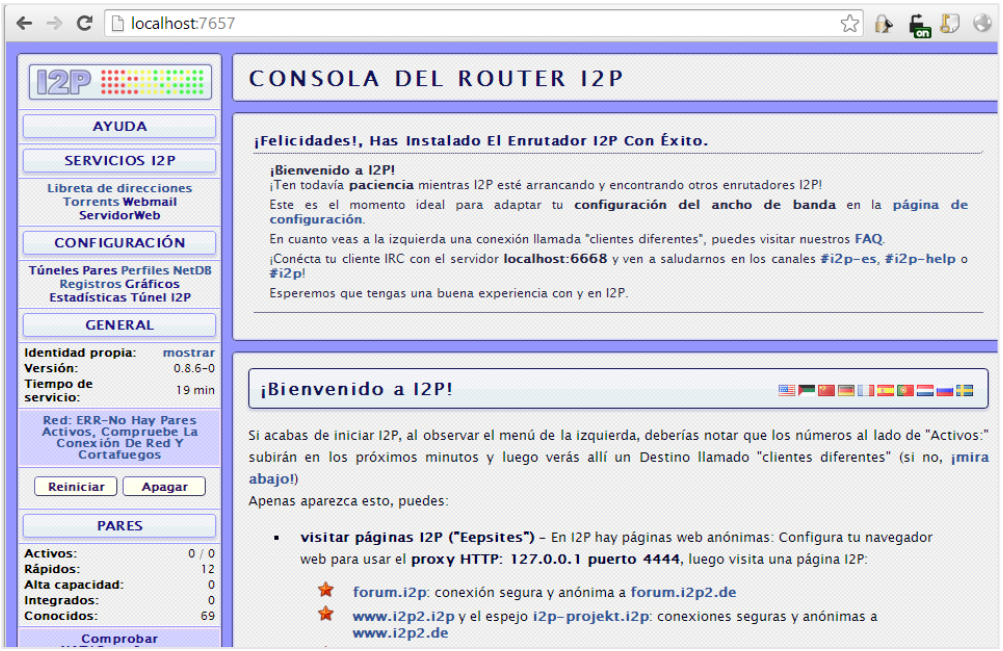


Figura 36: I2P en Windows

Bien ahora vamos a ver como se configura el navegador y el programa de irc. Para el navegador la mejor opción es usar Firefox o Chrome, ya que ambos tiene extensiones para cambiar de proxy, que es lo que hacemos para navegar por la red I2P, para Chrome la extensión se llama Proxy Switch e instala la extensión, luego saldrá la página de configuración en donde tienes que indicarle la dirección del proxy (127.0.0.1) y el puerto (4444) tal que así:

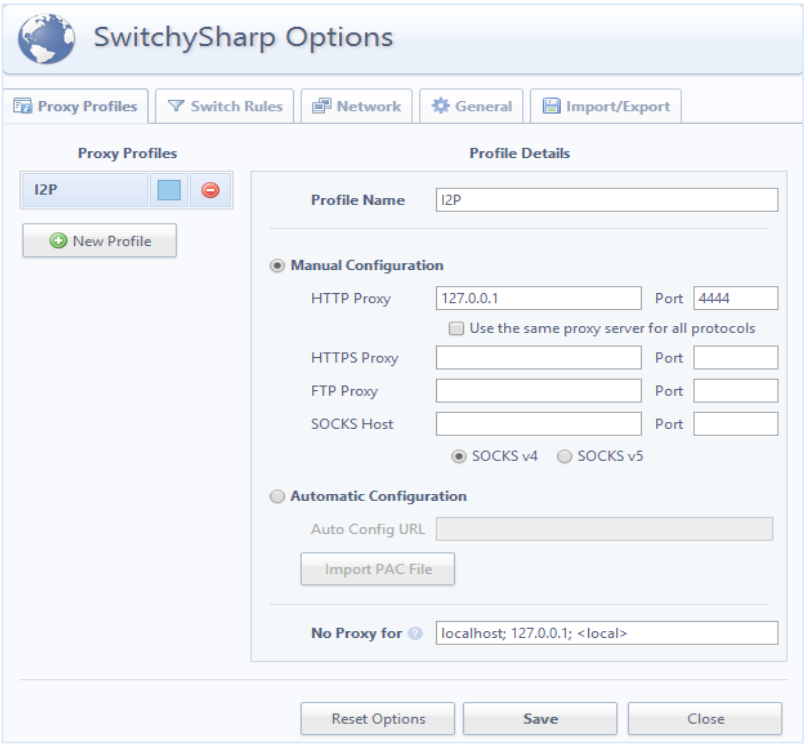


Figura 37: Configurando proxy Switch del navegador de Chrome

Le damos save y listo. Ahora en la parte de arriba a la derecha hay un nuevo icono que es el del proxy Switch y si clic sobre él te dará la opción de elegir el proxy para navegar. Configurado nuestro proxy en el navegador, ahora ya podemos navegar de forma anónima en Windows usando I2P. Para Firefox este navegador se configura igual que lo hemos configurado en Linux.

8.3. GNU PRIVACY GUARD (GPG)

GNU Privacy Guard o GPG es una herramienta de cifrado y firmas digitales, que viene a ser un reemplazo del PGP (Pretty Good Privacy) pero con la principal diferencia que es software libre licenciado bajo la GPL. GPG utiliza el estándar del IETF denominado OpenPGP.

GnuPG utiliza criptografía de clave pública para que los usuarios puedan comunicarse de un modo seguro. En un sistema de claves públicas cada usuario posee un par de claves, compuesto por una clave privada y una clave pública. Cada usuario debe mantener su clave privada secreta; no debe ser revelada nunca. La clave pública se puede entregar a cualquier persona con la que el usuario desee comunicarse. GnuPG implementa un esquema algo más sofisticado en el que un usuario tiene un par de claves primario, y ninguno o más de un par de claves adicionales subordinadas. Los pares de claves primarios y subordinados se encuentran agrupados para facilitar la gestión de claves, y el grupo puede ser considerado como un sólo par de claves [57].

¿Cómo funciona gpg?

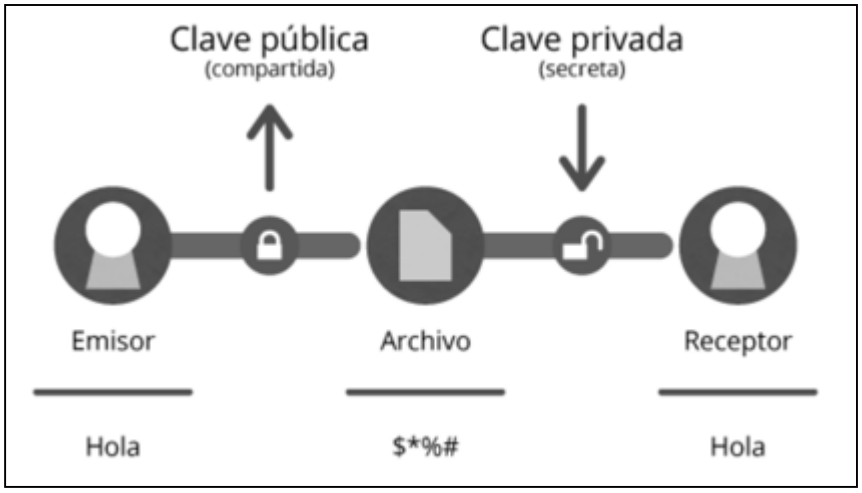


Figura 38: Funcionamiento de GPG

Esta criptografía se basa en dos claves distintas (de ahí el nombre de criptografía asimétrica). Una de las claves se denomina pública y la otra privada. La clave pública (como su nombre indica) puede hacerse pública, por el contrario la clave privada sólo es conocida por el propietario de la misma.

Cuando una persona quiere firmar digitalmente un mensaje usa su clave privada, de esta forma cualquier persona que posea la clave pública del remitente podrá comprobar que el mensaje ha sido firmado correctamente.

Para cifrar un mensaje se usa la clave pública del destinatario, así cuando éste reciba el mensaje podrá usar su clave privada para descifrarlo y por tanto sólo él puede ver el contenido del mensaje [59].

Instalación de GnuPG 2.0.25 en Linux

GnuPG es multiplataforma para descargarlo nos dirigimos a su página oficial https://www.gnupg.org/ para su posterior instalación descargamos la última versión más reciente GnuPG 2.0.25 en nuestro caso hemos descargado la versión bajo Linux, ya que estaremos utilizándolo bajo este sistema operativo en él se harán todas las pruebas necesarias [58].

Lo primero que debemos de hacer es instalar todas las librerías, las descargamos de las siguientes direcciones.

Libgpg-error (ftp://ftp.gnupg.org/gcrypt/libgpg-error/)
Libgcrypt (ftp://ftp.gnupg.org/gcrypt/libgcrypt/)
Libksba (ftp://ftp.gnupg.org/gcrypt/libksba/)
Libassuan (ftp://ftp.gnupg.org/gcrypt/libassuan/)

Cada dirección nos llevara a un directorio en donde elegiremos una versión para instalar, en este caso hemos elegido las versiones más reciente descargamos la comprimida en .tar.gz. Una vez descargadas las procedemos de descomprimir y a instalar de la siguiente manera:

Los pasos predefinidos que generalmente se suelen seguir para instalar un software desde sus fuentes son:

1) $ tar xvfz <nombre_paquete>.tar.gz
2) (o $ tar xvfz <nombre_paquete>.tgz)
3) (o $ tar xvfj <nombre_paquete>.tar.bz2)
4) $ cd <nombre_paquete>
5) $ ./configure
6) $ make

Estos mismo pasos seguimos para cada archivo que instalamos. Instaladas todas las librerías instalamos GnuPG 2.0.25 usando los mismos pasos [60].

Generación de las claves

El primer paso es generar nuestras propias claves (pública y privada) que se necesitan en cualquier sistema de criptografía asimétrica. Para ello usamos la orden:

```
gpg –-gen-key
```

```
yorleni@yorleni-aspire-one:~$ gpg --gen-key
gpg (GnuPG) 1.4.11; Copyright (C) 2010 Free Software Foundation, Inc.

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Por favor seleccione tipo de clave deseado:

  (1) RSA y RSA (predeterminado)

  (2) DSA y Elgamal

  (3) DSA (sólo firmar)

  (4) RSA (sólo firmar)

¿Su selección?:
```

Esta orden es interactiva y nos permite generar las claves basándose en la información que le iremos suministrando:

1. Tipo de clave. Lo normal es seleccionar la opción 1 (RSA)
2. Longitud de la clave. Los 2048 bits que se ofrecen por defecto nos servirán perfectamente.
3. Caducidad de la clave. Es importante darle caducidad a nuestra clave.
4. Nombre y Apellidos. Conviene no usar tildes ni eñes si nuestro nombre las contiene, ya que vamos a publicar esta clave en un servidor y podría haber problemas a la hora de leer nuestro nombre completo.
5. Dirección de correo electrónico. La dirección de correo electrónico asociada a nuestra clave.
6. Comentario. Si lo deseamos podemos añadir información extra, por ejemplo a qué nos dedicamos o para qué usamos la clave.
7. Contraseña. Para poder usar nuestra clave, ésta debe ser desbloqueada, por lo que se pide una contraseña que sólo debemos conocer nosotros y que tendremos que teclear cuando queramos usar la clave.

Conviene elegir una buena contraseña para que nuestra clave privada esté bien protegida. Una vez tecleada (y confirmada) nuestra contraseña, el sistema comenzará a generar las claves. Tal y como muestra la orden, conviene que mientras se están generando las claves, movamos el ratón, usemos el teclado y el disco para generar eventos aleatorios que el sistema pueda utilizar para generar las claves y por tanto tardar menos en generarlas. Cuando termine la generación, nuestras claves estarán en el directorio GnuPG de nuestro directorio personal.

Conviene proteger adecuadamente este directorio para que únicamente nosotros podamos acceder a él, así como realizar una copia de seguridad del mismo por si ocurriera un error grave en nuestro sistema. Es importante recordar que si perdemos la clave privada no podremos firmar ni descifrar mensajes y deberemos crear un nuevo par de claves.

Tras generar las claves podemos verlas con el comando gpg –k que nos muestra nuestro anillo de claves, lo importante de este paso es que veremos la identificación de cada una, que es necesaria para poderlas exportar y enviar.

```
yorleni@yorleni-aspire-one:~$gpg -k
/home/yorleni/.gnupg/pubring.gpg
------------------------------------------------------------------------------------------------------------------------
pub   4096R/F956CCFA  2014-08-07
uid                  yorleni <yorleni.88@gmail.com>
sub   4096R /EAB94A04 2019-08-06
```

Identificador y huella digital de la clave

Cuando se genera una clave, se le asocia un **identificador**. Este identificador permite hacer referencia a nuestra clave en sucesivas operaciones. Cada clave tiene también una **"huella digital"**. Esta huella es un número de 20 bytes que se utilizará a la hora de validar y firmar las claves. El identificador y la huella digital están muy relacionados ya que el identificador de la clave son los **cuatro últimos** bytes de la huella digital. Sería muy difícil que dos claves tuvieran el mismo identificador y estuviéramos trabajando con ambas.

Para averiguar el identificador de nuestra clave podemos usar la orden:

```
$ gpg –-list-keys
```

```
yorleni@yorleni-aspire-one:~$gpg –-list-keys
/home/yorleni/.gnupg/pubring.gpg

------------------------------------------------------------------------------------------------------------------------

pub   4096R/F956CCFA  2014-08-07

uid                  yorleni <yorleni.88@gmail.com>

sub   4096R /EAB94A04 2019-08-06
```

Exportar nuestra clave

Exportar la clave privada no es buena idea salvo que nos la vayamos a llevar a otro sitio o sepamos muy bien lo que estamos haciendo. En cambio, exportar la clave pública es una forma para **hacer llegar** esta clave a las personas con las que nos vayamos a comunicar usando criptografía asimétrica. Para exportar nuestra clave pública a un fichero:

```
yorleni@yorleni-aspire-one:~$gpg --output CPub.gpg --export F956CCFA
yorleni@yorleni-aspire-one:~/gpg$ ls –l |grep “CPub.gpg”

CPub.gpg
```

Publicar nuestra clave pública

Otra forma para distribuir nuestra clave que se usa con bastante frecuencia es la **publicación** en un **servidor de claves**, de esta forma únicamente tendremos que facilitar el identificador de nuestra clave y el servidor en la que la hemos publicado y de esta forma cualquier persona que quiera usar PGP puede acceder a nuestra clave (pública). Hay muchos servidores de claves repartidos por Internet. Algunos de ellos son:

- pgp.rediris.es
- pgp.mit.edu
- subkeys.pgp.net
- pgp.webtru.st

Para publicar nuestra clave ejecutaremos:

```
yorleni@yorleni-aspire-one:~$gpg --send-keys --keyserver pgp.mit.edu F956CCFA
```

A partir de este momento la clave estará accesible desde este servidor específico.

Importar la clave pública de otra persona

En este momento estamos preparados para **importar** las claves de otras personas. Cuando importamos otras claves, éstas se almacenan en lo que se llama nuestro **anillo de claves**. Podemos recibir estas claves por correo electrónico, descargarlas de una página web o importarlas de un servidor de claves (lógicamente para poder importarlas de un servidor de claves, el propietario de la clave tendrá que haberla publicado en el servidor como hemos hecho nosotros).Para esto nos descargamos la clave desde el servidor, y una vez descargada la importamos.

Al tener la clave ya en mi anillo de claves me contesta que no hay cambios. Para realizar la importación desde el servidor tenemos que usar el comando gpg --keyserver [Dirección del servidor] --recv-keys [ID de la clave]. Volvemos a listamos el anillo de clave vemos que ya tenemos la clave pública de Eymi.

```
yorleni@yorleni-Aspire-one:~$ gpg --keyserver pgp.mit.edu --recv-keys C9614963
gpg: solicitando clave C9614963 de hkp servidor pgp.mit.edu
gpg: clave C9614963: clave pública "Eymi <peraltaeymi1989@gmail.com>" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1  (RSA: 1)
yorleni@yorleni-Aspire-one:~$
```

Figura 39: Importando clave

Esta nos permitirá mantener una comunicación donde el intercambio de mensajes será cifrado.

```
$ gpg –-list-keys
```

```
pub   2048R/C9614963 2014-06-20
uid                  Eymi <peraltaeymi1989@gmail.com>
sub   2048R/43108B3D 2014-06-20

yorleni@yorleni-Aspire-one:~$
```

Figura 40: Listando la clave

Cifrar con la clave púbica

Ahora tenemos ya generado nuestro anillo de llave y exportado llaves hemos compartido nuestra clave pública con el usuario Eymi, hecho esto ya podemos intercambiar mensajes porque ambos posemos la clave pública de ambas en este caso clave pública de Yorleni y Eymi.

Por ejemplo a Eymi le tenemos que mandarle un documento, para cifrar el documento usaremos el comando gpg —encrypt —recipient [ID de la clave] [Archivo]

```
yorleni@yorleni-aspire-one:~$echo "Este mensaje será cifrado con gpg" > text.txt
```

Y ya tenemos el archivo listo para mandarlo de forma segura.

```
yorleni@yorleni-Aspire-one:~$ gpg --encrypt --recipient F956CCFA text.txt
yorleni@yorleni-Aspire-one:~$ ls -l |grep "text
texto1.txt     texto.txt      texto.txt.gpg  text.txt       text.txt.gpg
yorleni@yorleni-Aspire-one:~$ ls -l |grep "text.txt"
-rw-rw-r--  1 yorleni yorleni       10 ago  7 17:38 text.txt
-rw-rw-r--  1 yorleni yorleni      606 ago  7 17:39 text.txt.gpg
yorleni@yorleni-Aspire-one:~$
```

Figura 41: Cifrando mensaje

Descifrar un archivo con la clave privada

Eymi nos ha enviado un archivo ahora es el momento de descifrar con la clave pública que habíamos exportado, con el comando gpg -d [Archivo] e introduciendo la contraseña pública.

```
yorleni@yorleni-Aspire-one:~$ gpg -d clave.txt.gpg
gpg: datos cifrados CAST5
gpg: cifrado con 1 frase contraseña
hola estamos probando GPG
gpg: AVISO: la integridad del mensaje no está protegida
yorleni@yorleni-Aspire-one:~$
```

Figura 42: Descifrando mensaje con la clave publica

Firmar archivos

Una de las medidas de seguridad básicas al pasar un mensaje es asegurarnos que el emisor es quien dice ser, para asegurarnos de esto digitalmente existe la firma digital, como GPG usaba los **hash** para crear una firma simple, pero también podemos cifrarlo y a su vez firmarlo, que es lo que haremos con el comando gpg -u [ID de la clave privada] --output [Archivo resultante] --sign [Archivo para firmar] e introduciendo la contraseña de la clave privada [59].

```
yorleni@yorleni-aspire-one:~$echo "Este archivo se firmara con gpg" > firmar.txt
yorleni@yorleni-aspire-one:~$gpg -u EAB94A04 --output firmar.txt.gpg --sign firmar.txt
```

```
yorleni@yorleni-Aspire-one:~$ echo "Este archivo se firmara usando GPG" >My_firm
a.txt
yorleni@yorleni-Aspire-one:~$
```

Figura 43: Escribimos el archivo que deseamos firmar

```
yorleni@yorleni-Aspire-one:~$ gpg -u EAB94A04 --output  My_firma.txt.gpg --sign
My_firma.txt

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "yorleni <yorleni.88@gmail.com>"
clave RSA de 4096 bits, ID F956CCFA, creada el 2014-08-07

yorleni@yorleni-Aspire-one:~$ ls -l | grep "My_firma.txt.gpg"
-rw-rw-r--  1 yorleni yorleni      620 ago  8 14:29 My_firma.txt.gpg
yorleni@yorleni-Aspire-one:~$
```

Figura 44: Firmamos el archivo que vamos a transmitir

```
yorleni@yorleni-Aspire-one:~$ gpg -u EAB94A04 --output  My_firma.txt.gpg --sign
My_firma.txt

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "yorleni <yorleni.88@gmail.com>"
clave RSA de 4096 bits, ID F956CCFA, creada el 2014-08-07

yorleni@yorleni-Aspire-one:~$
```

Figura 45: Archivo firmado My_firma.txt.gpg

8.4. XMPP

Extensible Messaging and Presence Protocol, más conocido como XMPP (Protocolo extensible de mensajería y comunicación de presencia) (anteriormente llamado Jabber), es un protocolo abierto y extensible basado en XML, originalmente ideado para mensajería instantánea.

Con el protocolo XMPP queda establecida una plataforma para el intercambio de datos XML que puede ser usada en aplicaciones de mensajería instantánea. Las características en cuanto a adaptabilidad y sencillez del XML son heredadas de este modo por el protocolo XMPP.

A diferencia de los protocolos propietarios de intercambio de mensajes como ICQ, Y! y Windows Live Messenger, se encuentra documentado y se insta a utilizarlo en cualquier proyecto. Existen servidores y clientes libres que pueden ser usados sin coste alguno.

Tras varios años de su existencia, ha sido adoptado por empresas como Facebook, Tuenti, WhatsApp Messenger y Nimbuzz, entre otras, para su servicio de chat.

Google lo adoptó para su servicio de mensajería Google Talk, y en 2013 anunció que lo abandonaría en favor de su protocolo propietario Hangouts.

Ventajas

Descentralización: La arquitectura de las redes XMPP es similar a la del correo electrónico; cualquiera puede poner en marcha su propio servidor XMPP, sin que haya ningún servidor central.

Estándares abiertos: La Internet Engineering Task Force ha formalizado el protocolo XMPP como una tecnología de mensajería instantánea estándar, y sus especificaciones han sido publicadas como los RFC 3920 y RFC 3921. El desarrollo de esta tecnología no está ligado a ninguna empresa en concreto y no requiere el pago de regalías.

Historia: Las tecnologías XMPP llevan usándose desde 1998. Existen múltiples implementaciones de los estándares XMPP para clientes, servidores, componentes y bibliotecas, con el apoyo de importantes compañías como Sun Microsystems y Google.

Seguridad: Los servidores XMPP pueden estar aislados de la red pública XMPP, y poseen robustos sistemas de seguridad (como SASL y TLS). Para apoyar la utilización de los sistemas de cifrado, la XMPP Standards Foundation pone a disposición de los administradores de servidores XMPP Autoridad de certificación en xmpp.net ofreciendo certificados digitales gratis.

Flexibilidad: Se pueden hacer funcionalidades a medida sobre XMPP; para mantener la interoperabilidad, las extensiones más comunes son gestionadas por la XMPP Software Foundation.

Desventajas

Sobrecarga de datos de presencia: Típicamente cerca de un 70% del tráfico entre servidores son datos de presencia, y cerca de un 60% de estos son transmisiones redundantes. Actualmente se están estudiando nuevos protocolos para aliviar este problema.

Escalabilidad: XMPP también sufre el mismo problema de redundancia en los servicios de chatroom y de suscripción. Actualmente se está trabajando en su solución.

Sin datos binarios: XMPP es codificado como un único y largo documento XML, lo que hace imposible entregar datos binarios sin modificar. De todas formas, las transferencias de archivos se han solucionado usando otros protocolos como HTTP. Si es inevitable, XMPP también puede realizar transferencias codificando todos los datos mediante base64 [33].

Con el protocolo XMPP queda establecida una plataforma para el intercambio de datos XML que puede ser usada en aplicaciones de mensajería instantánea. Las características en cuanto a adaptabilidad y sencillez del XML son heredadas de este modo por el protocolo XMPP. A diferencia de los protocolos propietarios de intercambio de mensajes como ICQ, Y! y Windows Live Messenger, se encuentra documentado y se insta a utilizarlo en cualquier proyecto. Existen servidores y clientes libres que pueden ser usados sin coste alguno. Tras varios años de su existencia, ha sido adoptado por empresas como Facebook, Tuenti, WhatsApp Messenger y Nimbuzz, entre otras, para su servicio de chat. Google lo adoptó para su servicio de mensajería Google Talk, y en 2013 anunció que lo abandonaría en favor de su protocolo propietario Hangouts.

¿Qué puede hacer con XMPP?

La Mensajería y Presencia Protocolo Extensible (XMPP) es una tecnología abierta para el tiempo real comunicación, utilizando el lenguaje de marcado extensible (XML) como formato de base para el intercambio de información. En esencial, XMPP proporciona una forma de enviar pequeños trozos de

XML de una entidad a otra casi en tiempo real. XMPP se utiliza en una amplia gama de aplicaciones, y puede ser adecuado para su aplicación, también. Para imaginar las posibilidades, es útil para romper el universo XMPP abajo en un alto nivel en servicios y aplicaciones. Los servicios se definen en dos especificaciones primarias publicado por el Internet Engineering Task Force (IETF) en http://ietf.org/.

(La serie " RFC "), y en docenas de especificaciones de extensión publicados por la XMPP Normas Fundación en http://xmpp.org/(la serie " XEP "); las aplicaciones son software programas y escenarios de implementación que son de interés común para los individuos y de las organizaciones, a pesar de los servicios básicos que permiten construir muchas otras aplicaciones tipos, así servicios.

En este contexto, un servicio es una característica o función que puede ser utilizado por cualquier aplicación dada. Implementaciones XMPP suelen ofrecer los siguientes servicios principales:

Cifrado Canal: Este servicio, que se define en el [RFC 3920] proporciona cifrado de la conexión entre un cliente y un servidor, o entre dos servidores. Aunque el cifrado de canal no es necesariamente emocionante, es un importante bloque de construcción para la construcción de aplicaciones seguras.

Autenticación: Este servicio, también se define en [RFC 3920] es otra parte de la base para el desarrollo de aplicaciones seguras. En este caso, el servicio de autenticación asegura que las entidades que intentan comunicarse a través de la red se primera autenticado por un servidor, que actúa como una especie de controlador de acceso de acceso a la red.

Presencia: Este servicio, que se define en el [RFC 3921], permite a averiguar acerca de la disponibilidad de la red de otras entidades. En el más nivel básico, un servicio de presencia responde a la pregunta: "¿Es la entidad en línea y disponible para la comunicación, o fuera de línea y no está disponible? Datos de presencia también puede incluir información más detallada (por ejemplo, si una persona está en una reunión).Por lo general, el intercambio de información de presencia se basa en una presencia explícita suscripción entre dos entidades con el fin de proteger la privacidad de los usuarios información.

Las listas de contactos: Este servicio, también se define en [RFC3921], le permite almacenar una lista de contactos o lista, en un servidor XMPP. El más común utilizar para este servicio es una mensajería instantánea lista de amigos, pero cualquier entidad que tenga un cuenta en un servidor puede utilizar el servicio para mantener una lista de entidades conocidas o de confianza (por ejemplo, que puede ser utilizado por los robots).

Mensajes uno a uno: Este servicio, que se define en el [RFC 3920] permite enviar mensajes a otra entidad. El uso clásico de la mensajería de uno - a-uno IM es personal, pero los mensajes pueden ser XML arbitrario , y cualquiera de las dos entidades en una red puede intercambiar mensajes podrían ser bots , servidores , componentes, dispositivos , XMPP habilitado para servicios web , o cualquier otra entidad XMPP .

Mensajería multi –party: Este servicio, que se define en [XEP -0045] permite a unirse a una sala de chat virtual para el intercambio de mensajes entre múltiples participantes, similar a Internet Relay Chat (IRC). Los mensajes pueden ser de texto plano, o puede contener extensiones XML para funciones más avanzadas, como la sala de configuración, la votación en la habitación, y varios mensajes de control de sesión.

Notificaciones: Este servicio, que se define en [XEP -0060] permite generar una notificación y lo han entregado a varios suscriptores. Este servicio es similar a la mensajería multi- partido, pero está optimizado para uno a muchas entregas con las suscripciones a canales explícitos o temas (llamados “nodos “) específicas.

Descubrimiento de servicios: Este servicio, que se define en [XEP -0030] permite saber qué funciones están soportadas por otra entidad, así como cualquier entidad adicional que se asocian con él (por ejemplo, salas alojados en una sala de chat servicio).

Capacidades anuncio: Este servicio, que se define en [XEP -0115] es un extensión al servicio de presencia que proporciona una notación abreviada para el servicio de datos de descubrimiento para que pueda almacenar en caché con facilidad las características que son compatibles con otros entidades de la red.

Formularios de datos estructurados: Este servicio, que se define en [XEP -0004] permite intercambiar formas estructuradas pero flexibles con otras entidades, de forma similar a Formularios HTML. A menudo se utiliza para la configuración y otras tareas en las que necesita reunir información ad -hoc de otras entidades.

Gestión de flujo de trabajo: Este servicio , que se define en [XEP -0050] permite participar en una interacción del flujo de trabajo estructurado con otra entidad, con el apoyo de las acciones de flujo de trabajo típicos, como pasar a la siguiente etapa de un proceso de negocio o ejecutar un comando. Se utiliza a menudo en conjunción con los datos.

Sesiones de medios - Peer -to-peer: Este servicio, que se define en [XEP- 0166], permite negociar y gestionar una sesión de medios con otra entidad. Tal sesión puede ser utilizada para el propósito de chat de voz, chat de vídeo, transferencia de archivos, y otras interacciones en tiempo real.

Arquitectura

Todas las buenas tecnologías de Internet tienen una “arquitectura “, una manera de que varias entidades se ajusta juntos, vincularse y comunicarse. Por ejemplo , la Word Wide Web consiste millones de servidores web que ejecutan el software como Apache , y muchos más millones de web software de los clientes (navegadores) que se ejecuta como Firefox , utilizando todos los protocolos y los datos estándar formatos como HTTP y HTML. Como otro ejemplo , la infraestructura de correo electrónico consiste de millones de servidores de correo electrónico que se ejecutan programas como Postfix , y muchos más millones de clientes de correo electrónico que se ejecutan programas como Thunderbird, todos utilizando protocolos estándar como SMTP , POP e IMAP.

Tecnologías XMPP utilizan una arquitectura cliente-servidor descentralizado similar a las arquitecturas utilizado para el correo electrónico de la red Word Wide Web.

XMPP es un protocolo que varios clientes de chat; por ejemplo pidgin. Dicho cliente contiene un plugin que permite cifrar los mensajes llamados Off The Record (OTR).Por lo tanto al combinar todas esta herramientas, es posible utilizar cifrado en facebook.

Pidgin

Pidgin (anteriormente llamado Gaim) es un cliente de mensajería instantánea multiplataforma capaz de conectarse a múltiples redes (multiprotocolo) y cuentas (multicuenta) de manera simultánea [35]. Pidgin es un programa de chat que te permite conectarte a las cuentas en múltiples redes de chat simultáneamente. Esto significa que usted puede estar chateando con amigos en MSN, hablando con un amigo en Google Talk, y sentado en una sala de chat Yahoo, todo al mismo tiempo.

Pidgin es compatible con las siguientes redes de chat de la caja: AIM, ICQ, Google Talk, Jabber / XMPP, MSN Messenger, Yahoo!, Bonjour, Gadu-Gadu, IRC, Novell GroupWise Messenger, Lotus Sametime, SILC, SIMPLE, MXit , MySpaceIM, y Zephyr. Puede apoyar muchos más con los plugins [36].

Pidgin soporta muchas características de estas redes de chat, como las transferencias de archivos, mensajes de ausencia, iconos de amigos, emoticonos personalizados y notificaciones de mecanografía.

Por si solo no nos da mucha seguridad. Existen 2 plugins de pidgin que permite encriptar el chat a nivel de cliente. Esto es muy bueno porque podemos usar cualquier protocolo de chat y la información pasará encriptada por los servidores. Es decir solo el que escribe y el que recibe podrán leer la información.

Características principales

- Conversaciones mostradas en pestañas
- Posibilidad de conectarse a varia redes simultáneamente
- Registro de conversaciones
- Permite el reemplazo de los nombres de los contactos de la lista
- Muestra un aviso o reproduce un sonido cuando un contacto se conecta/desconecta o cambia de estado.

- Posibilidad de transparencia para las ventanas de contactos y de conversación mediante un plugin
- Transferencia de archivos
- Soporte para Webcam (Sólo disponible en algunos protocolos)

Protocolos soportados

- AOL Instant Messenger
- Bonjour
- Gadu-Gadu
- ICQ
- Internet Relay Chat (comúnmente llamado IRC)
- MSN Messenger de forma nativa, aunque sin funciones audiovisuales.
- MXit
- MySpaceIM
- SILC
- SIMPLE
- Sametime Lotus
- XMPP/Jabber (Facebook entre otros)
- Yahoo!
- Zephyr

Pero puede soportar más protocolos, gracias a complementos no-oficiales: quizá los más destacables son el complemento para Skype y msn-pecan, un protocolo alternativo a la red MSN que ofrece mayor velocidad de conexión y transferencia de archivos.

Cifrado

Los mensajes pueden ser cifrados utilizando plugins:

- Usando cifrado RSA. Al plugin se le llama Pidgin-Encryption
- Usando el protocol Off the record messaging (OTR). Al plugin se le llama OTR-Plugin [37].

Desde una terminal instalamos el paquete

```
Sudo apt-get install pidgin.
```

- Una vez abierto Pidgin hacemos clic en Cuentas-> Gestionar cuentas
- Click en añadir para crear una nueva cuenta.
- La nueva cuenta la configuramos de la siguiente manera:

- Protocolo: XMPP
- Nombre de usuario: Es el de Facebook = www.facebook.com/xxxxxxxxxx)
- Dominio: chat.facebook.com
- Recurso: Pidgin
- Contraseña: aquí la calve de Facebook o sea la cuenta que deseamos vincular a pidgin

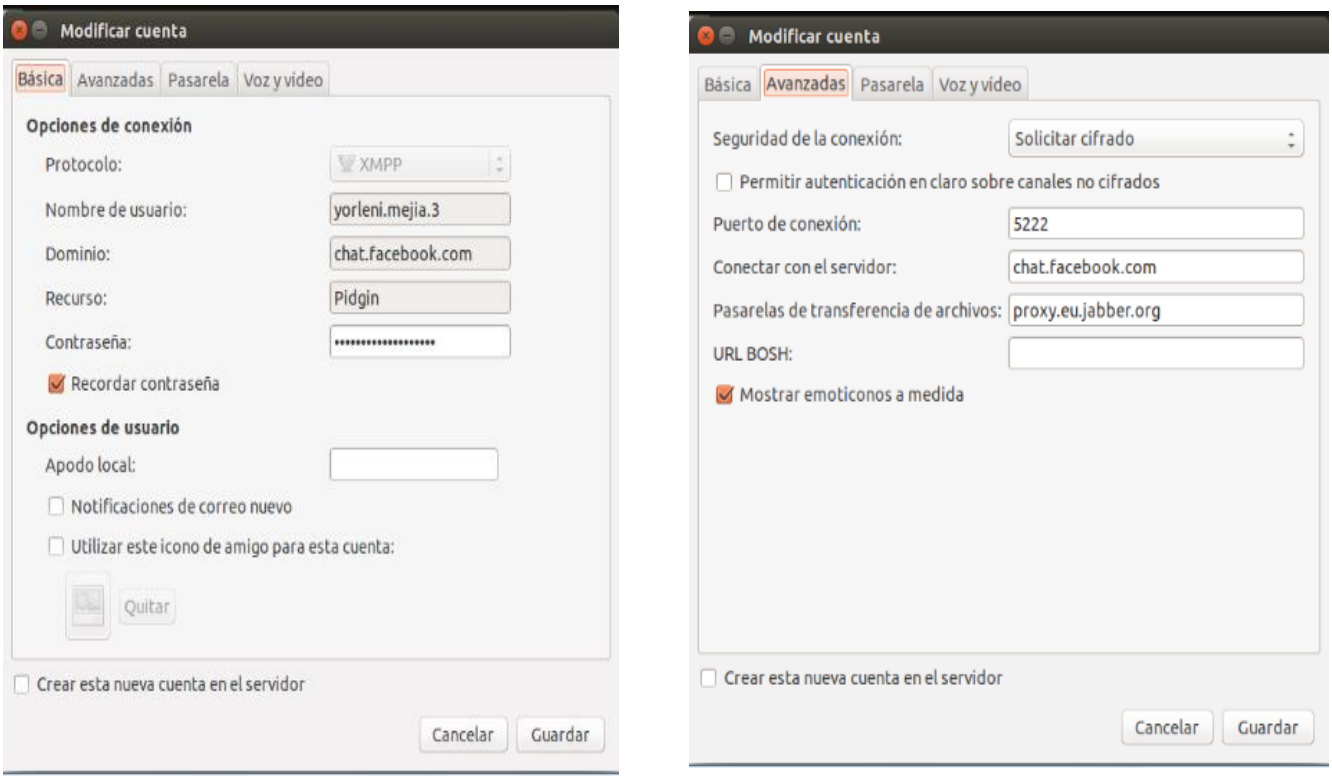


Figura 46: Configuración de la cuenta Facebook en pidgin

Ahora ya poseemos nuestra cuenta de facebook vinculada a pidgin, ya podemos ver todos nuestros contactos aquí e iniciar una sesión de chat este funciona como facebook mensajería instantánea en pidgin puedes usar el estado que desees disponible, ausente entre otros. Ahora lo que haremos en configurar la cuenta de Gmail con pidgin.

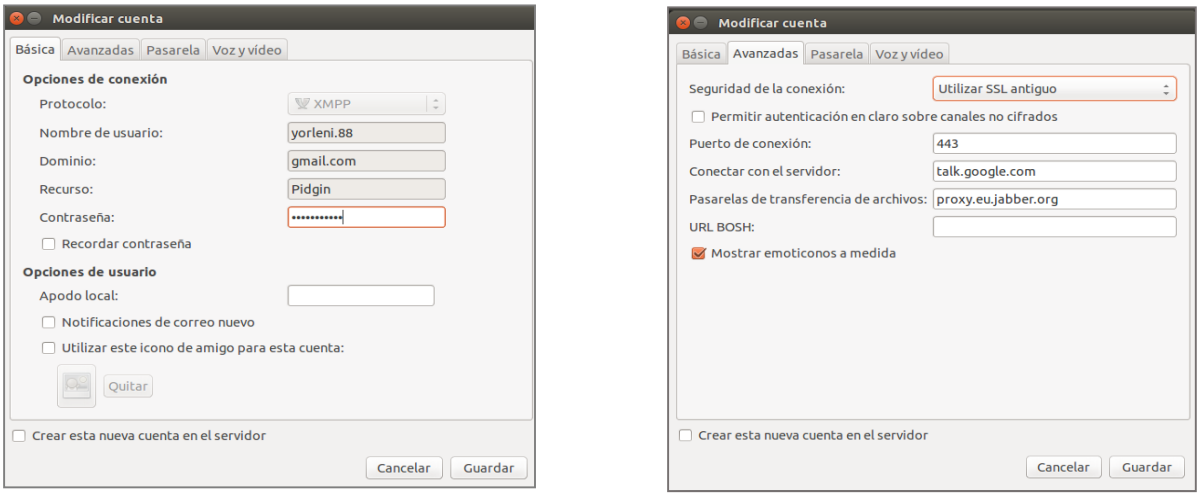


Figura 47: Configuracion de la cuenta Gmail con pidgin

OTR

El protocolo **"Off-The-Record Messaging"**, comúnmente llamado **OTR**, es un protocolo criptográfico que proporciona una fuerte encriptación para conversaciones de mensajería instantánea. OTR utiliza una combinación del algoritmo AES, el protocolo de intercambio de claves Diffie-Hellman y la función hash SHA-1. Además de la autenticación y la encriptación, OTR aporta una confidencialidad directa perfecta.

Características

Aparte de dar encriptación y autenticación (características también proporcionadas por PGP, GnuPG, y X.509 (S/MIME)), OTR ofrece otras características menos comunes:

- *Confidencialidad secreta perfecta*. Los mensajes solo son encriptados con una clave temporal AES para cada mensaje, negociada usando el protocolo de intercambio de claves Diffie-Hellman. El compromiso de cualquier clave criptográfica de vida larga no compromete ninguna conversación previa, incluso si el atacante tiene cyphertexts.

- *Ocultación de la identidad*. Los mensajes en una conversación no tienen firma digital y, cuando una conversación se ha completado, cualquiera podría falsificar un mensaje que aparentemente venga de otro de los participantes en la conversación, asegurando que es imposible demostrar que cierto mensaje fue escrito por cierta persona. Dentro de la conversación, el destinatario puede estar seguro de que un mensaje viene de la persona que ha identificado [38].
- *Autenticación:* Se te garantiza que tu correspondiente es quién tu crees que es.
- *Rechazo:* Después de finalizada la sesión de conversación (chat), los mensajes no pueden ser identificados como procedentes de ti o de tu correspondiente.
- *Cifrado:* Nadie más puede acceder ni leer tus mensajes instantáneos.
- *Perfecta Seguridad Adicional*: Si terceros obtienen tus claves privadas, ninguna de tus conversaciones anteriores estará en peligro.

Instalación de OTR

Lo primero que debemos de hacer es desde terminal instalar pidgin-otr.

sudo apt-get install pidgin pidgin-otr

Una vez que tenga la cuenta de chat configurada, hay que verificar que el módulo de OTR este activo. Para esto en el menú donde se despliegan los contactos conectados dentro del menú herramientas seleccionar complementos. Una vez en los complementos buscar Mensajería Fuera-de-Registro (OTR).

Ahora en configurar complemento generamos una clave y habilitamos las siguientes opciones como se muestra la imagen y damos generar esperamos hasta que se genere la huella.

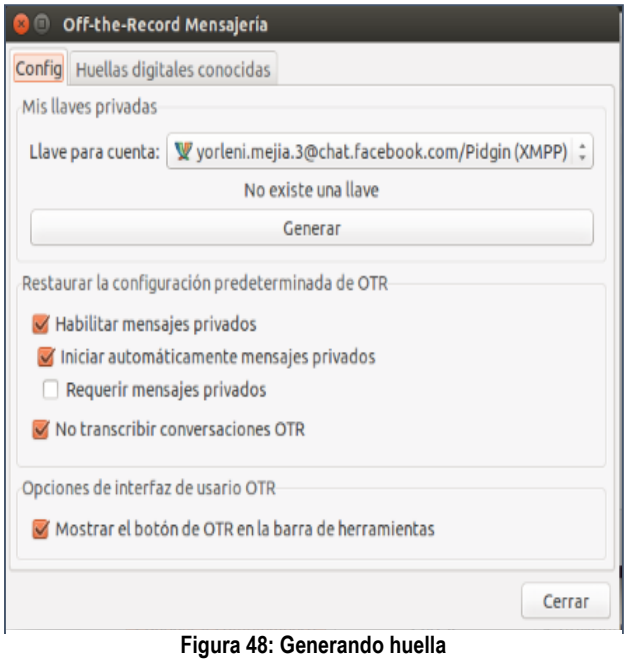


Figura 48: Generando huella

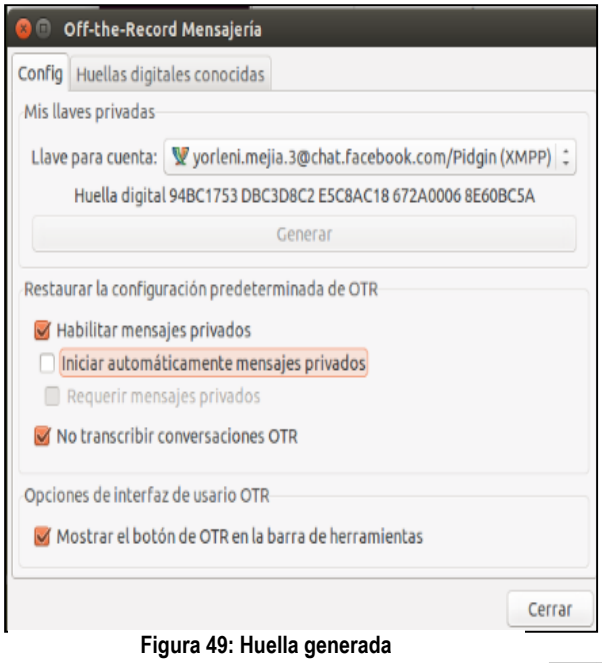


Figura 49: Huella generada

Con eso ya está instalado el complemento de encriptación.

Se puede verificar que está activo si en la parte inferior derecha de la ventana de chat tenemos un mensaje que dice "**No privado**."

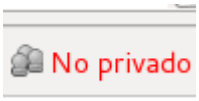


Figura 50: Chat no privado

Esto quiere decir que se está trabajando con un chat no seguro. Si la persona con la que se está chateando tiene instalado OTR, entonces se puede iniciar una conversación privada. Para esto se hace click "**No Privado**" y luego en iniciar conversación privada.

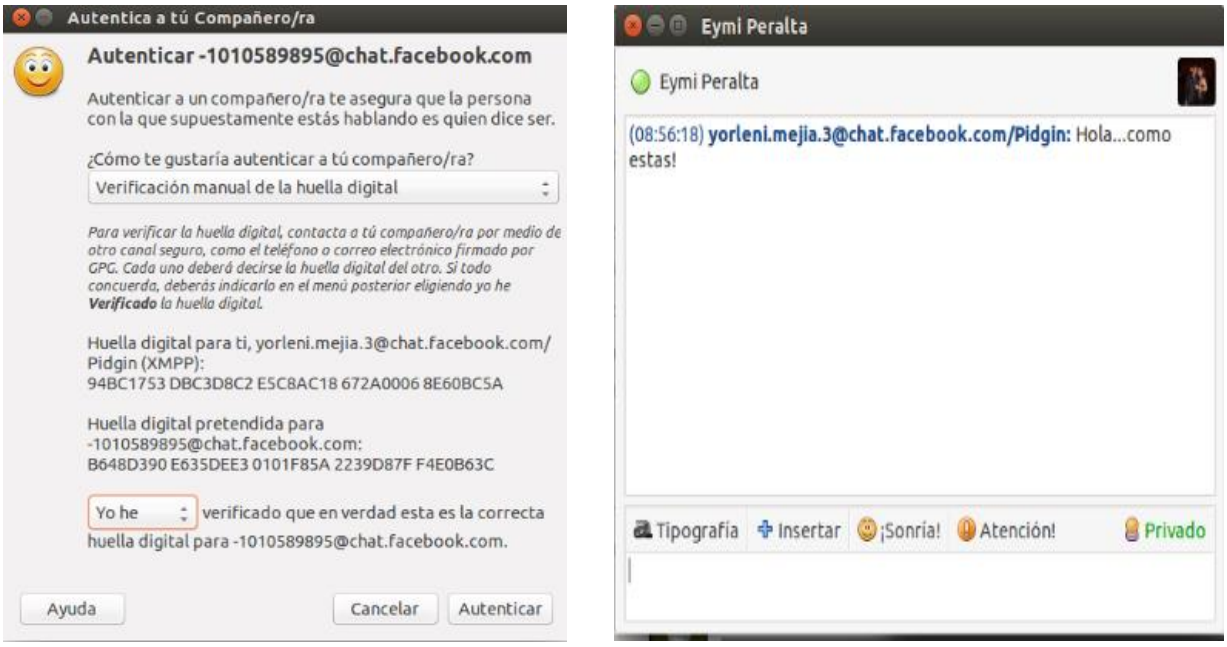


Figura 51: Autentificando a un compañero de chat

Ahora le damos identificar a un compañero.

Ahora ya se tiene una conversación encriptada

Los mensajes pueden ser cifrados utilizando plugins:

- Usando cifrado RSA. Al plugins se le llama Pidgin-Encryption
- Usando el protocol Off the record messaging (OTR).
- Al plugins se le llama OTR-Plugins [37].

8.5. ANONTWI

Anontwi es una herramienta de software libre escrita en python que permite mantener cierta privacidad y anonimato en determinadas redes sociales. Anontwi utiliza OAuth2, por tanto, soporta Identi.ca, es decir statusnet, así como Twitter. Utiliza la tecnología de socks para conectarse a las aplicaciones, permitiendo el uso de proxies (tor) [16].

AnonTwi es un *scrip* en Python que permite enviar y recibir *tweet* y mensajes cifrados mediante AES y HMAC-SHA1 en Twitter, al que se conecta mediante SSL. Otras características son la posibilidad de usar la red TOR, envío de mensajes largos o la falsificación de las cabeceras que envía.

La herramienta, cuyo desarrollo comenzó a principios de 2012 parte de una necesidad; “Tratar de evitar actos de censura en redes sociales” [39].

La herramienta de software libre anontwi, recién salida del horno, está ya disponible para su descarga.

Se trata de un cliente shell en python que permite, entre otras cosas [40]:

- cifrado SSL para comunicarse con Twitter (contra ataques MItM!)
- conectarte desde la red TOR
- introduce falsas cabeceras en los envíos
- permite enviar mensajes largos que divide de forma automática
- utiliza cifrado avanzado: AES128 + Blowfish tanto para Tweets como para DMs (Direct Messages)
- descifra de forma automática.

En el caso del cifrado, Anontwi utiliza la combinación de los algoritmos: AES256 [35] + SHA1 [36] - HMAC [16].El rendimiento obtenido es de 69 caracteres en texto plano en bloques de 140 cifrados.

Instalación de anontwi en ubuntu

Desde la línea de comando de Linux ejecutamos:

```
sudo apt-get install python-crypto python-httplib2 python-pycurl
```

Nota: Recuerda para generar Consumer keys necesario que tengas tu cuenta Twitter porque desde allí lo vamos a generar.

Paso 1) "Consumer" keys:

- Para usar OAuth con Twitter, necesitas estos tokens: 'consumer key' y 'consumer secret'.
- Primero, necesitas crear una aplicación de terceros en tu cuenta de desarrolladores de twitter.
- Entra aquí: https://dev.twitter.com
- En el icono de tu perfil, ve a: "My applications" y/o haz click en: "Create new application"
- Rellena los datos del formulario:
- "Name : El nombre de tu aplicación (Esto será visible cuando mandes un mensaje)"
- "Descripción: Una descripción (por ejemplo: AnonTwi")
- "Website: La página web de tu aplicación (por ejemplo: http://anontwi.sf.net)"
- "Si aceptas las reglas de Twitter, haz click en: 'Yes, I agree'"
- "Resuelve el captcha" ;-)
- "Haz click en: "Create your Twitter application"

  En "My application", haz click en tu application

  Busca: 'OAuth Settings' y guarda tus:
- Consumer key: xxxxxxxxxxxxxxxxxxx
- Consumer secret: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

Ahora que hemos generado nuestra aplicación nos movemos para ver los Consumer key y Consumer secret que hemos generado. En la siguiente imagen se puede ver los Consumer generada.

IMPORTANTE: Debes poner el nivel de acceso de AnonTwi a: Read, write, and direct messages ahora, ya está todo listo para generar tus: secret "Token keys".

Ahora abrimos el fichero “config.py” con un editor de textos, e introduce los tokens. Salvamos los cambios.

```
    APItokens = [{'4nW0u5htmiv3wub1fEQZFhNg1': token[0].strip(),
                  'JM2trlXGdvbIWJIMGL09x54NtZY9pmLuuRopLv0q84je5HODGK' : token[1].strip()}]

    api = open(DIR_TOKENS + FILE_SOURCE_API).readlines()
    APIsources = [{'source_api' : api[0].strip()}]

except:
    """Non GTK"""
    APItokens = [
                {
                  'consumer_key'    : "4nW0u5htmiv3wub1fEQZFhNg1", # enter your consumer_key
                  'consumer_secret' : "JM2trlXGdvbIWJIMGL09x54NtZY9pmLuuRopLv0q84je5HODGK", # enter $
                }
            ]
    APIsources = [
                {
                  'source_api' : "api.twitter.com", #enter your source api
                  #'source_api' : "identi.ca/api" #enter your source api
                }
            ]
```

Figura 52: Colocando Consumer key y secret en el archivo config.conf

Paso 2) “Token” keys:

- Para utilizar AnonTwi como una API de Twitter, necesitas estos tokens: ‘token key’ y ‘token secret’.
- Lanza: ./anontwi - - tokens

```
yorleni@yorleni-Aspire-one:~/Documentos/anontwi$ ./anontwi --tokens
================================================================================

AnonTwi [0.7] - 2012 - http://anontwi.sf.net -> by psy

================================================================================
Getting your API tokens (key & secret)...
================================================================================

Requesting temporal tokens...

Please visit this page and retrieve the pincode to be used
in the next step to obtaining an Authentication Token:

https://api.twitter.com/oauth/authorize?oauth_token=auE0yKzvoNlBJRUWj3WGxZ3hb7k
Q9LbRdNe7QOa0KA

Pincode?
```

Figura 53: Generando tokens

- Sigue el link para leer tu “PinCode” al dar clic en la url que nos indica nos redirige a la página que estamos viendo Aquí nos pide un login esto es nuestro cuenta de twitter.es decir login de twitter.

Figura 54: Autorizando una aplicación

Ahora que ya hemos que ya hemos puesto nuestro datos de entrada este nos muestra el pincode a usar. Recuerda esto si tienes tu cuenta de Twitter abierto no es necesario de login porque ya lo tienes abierto para poder autorizar, este te mostrara directamente el pincode en mi caso mi cuenta la tenía cerrada.

Figura 55: Pincode de autorización

Ahora regresamos a la consola y colocamos te pincode.

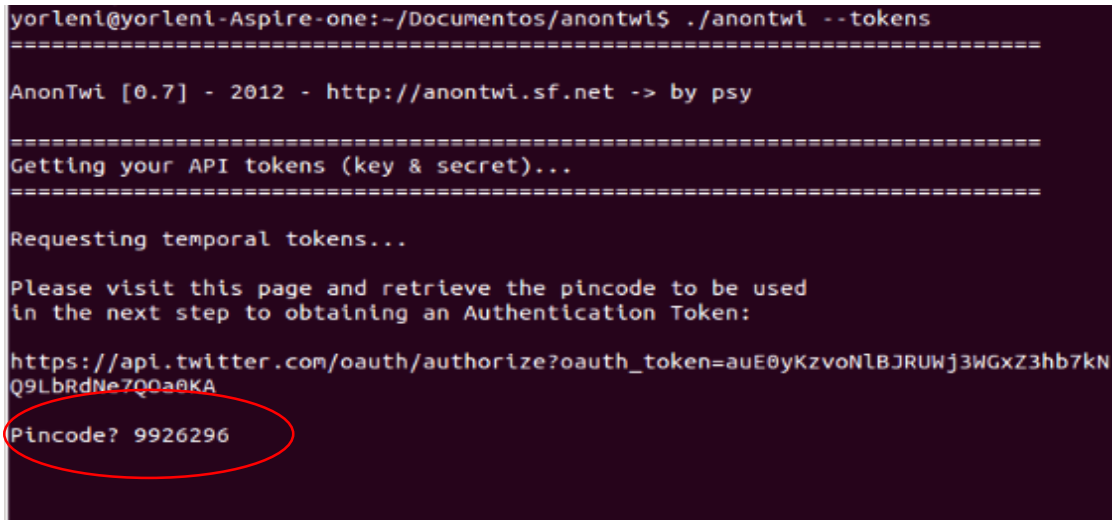


Figura 56: Colocando pincode dado

Al poner el pincode inicia generar todo los tokens y al finalizar nos muestra en pantalla al final las siguientes líneas de comando, lo que hacemos con estas es exportarla. Para eso copiamos, pegamos y ejecutamos.

Figura 57: Variables a exportar

EXPORTAR los tokens como variables de entorno de tu sistema operativo, para no tener que usarlos cada vez que lanzar la herramienta.

Con estos tokens, ya puedes empezar a lanzar comandos de -AnonTwi:./anontwi [-m 'text' | -r 'ID' | -d @user | -f @nick | -u @nick] [OPTIONS] 'token key' 'token secret'

Paso 3) Generar llave PIN para cifrar y descrifrar mensajes: ./anontwi –gen

Comparte esta llave de forma privada con los destinatarios de tu mensaje cifrado. No se la envíes a través de canales inseguros cómo el email, SMS, clientes de chat o Twitter.

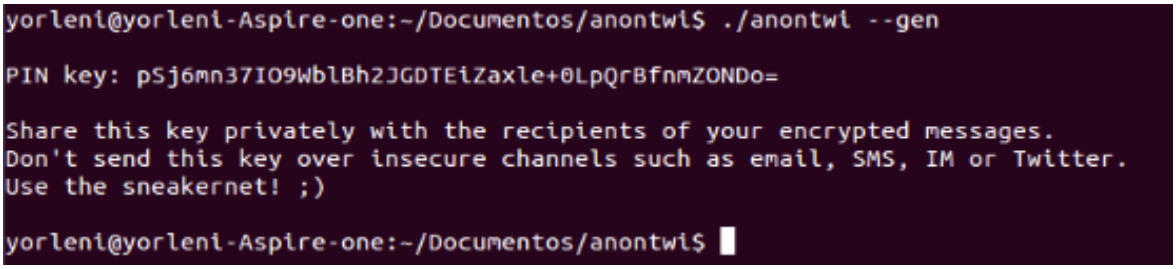


Figura 58: Generando una clave para cifrar

Si lo hiciste, ya puedes empezar a lanzar comandos de AnonTwi: ./anontwi [-m 'text' | -r 'ID' | -d @user | -f @nick | -u @nick] [OPTIONS].

Escribiéremos un mensaje así como se muestra en la imagen:

```
yorleni@yorleni-Aspire-one:~/Documentos/anontwi$ ./anontwi -m "Hello World" -d "
@yorleni_88" --enc --pin pSj6mn37IO9WblBh2JGDTEiZaxle+0LpQrBfnmZONDo=
================================================================================

AnonTwi [0.7] - 2012 - http://anontwi.sf.net -> by psy

================================================================================
Starting to send your DM (direct message)... :)
================================================================================

Message [ Number of words: 11 - Number of waves: 1 ]
-------------
Hello World
-------------
To: @yorleni_88
------

[Info] DM sent correctly!

yorleni@yorleni-Aspire-one:~/Documentos/anontwi$
```

Figura 59: Enviando mensaje encriptado con anontwi

```
yorleni@yorleni-Aspire-one:~/Documentos/anontwi$ ./anontwi -m "Nice tool! Si no
hay problema, vamos hacer un post en nuestro blog" -d "@yorleni_88" --enc --pin
"pSj6mn37IO9WblBh2JGDTEiZaxle+0LpQrBfnmZONDo= "  32NQ199LIYxsvznmuIRl0TlX2  poaG
lx4gtnNV3jYfmAOXkzpUC96EyuMrb4FHnK1uEExMOSDRjs
================================================================================

AnonTwi [0.7] - 2012 - http://anontwi.sf.net -> by psy

================================================================================
Starting to send your DM (direct message)... :)
================================================================================

Message [ Number of words: 66 - Number of waves: 1 ]
-------------
Nice tool! Si no hay problema, vamos hacer un post en nuestro blog
-------------
To: @yorleni_88
------

[Info] DM sent correctly!

yorleni@yorleni-Aspire-one:~/Documentos/anontwi$
```

Figura 60: Envió de mensajes en anontwi

Nota: esta herramienta no funciona en un 100% ya que los mensajes que escribimos no llegan. Porque desde los últimos cambios en la autenticación OAuth que hicieron para la nueva versión de la API de Twitter, se han quedados desconectadas esta herramienta no está actualizada.

Instalación de anontwi en windows

Primeramente tenemos que descargar las librerías que vamos a necesitar para poder trabajar con Anontwi:

Python 2.7 http://www.python.org/getit

httplib2 0.7.4 httplib2.googlecode.com/files/httplib2-0.7.4.zip

Pycrypto 2.3 http://www.voidspace.org.uk/downloads/pycrypto-2.3.win32-py2.7.zip

Nota: Para instalar las librerías solo ejecutamos los .exe. Pero en el caso de la librería httplib2 0.7.4, se ejecuta el setup.py de la siguiente manera:

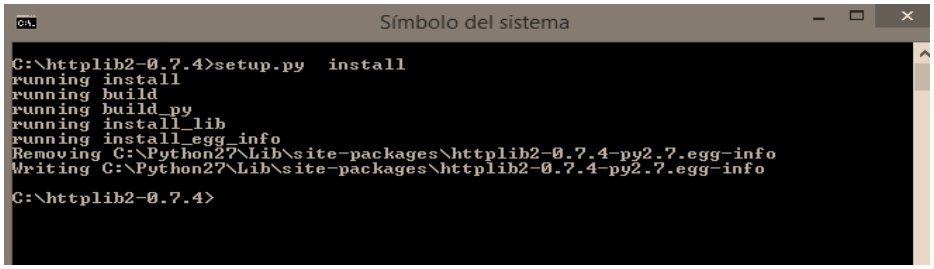


Figura 61: Instalación de librería httplib2

- Una vez instaladas las librerías, podemos descargar el software directamente desde el repositorio Sourceforge de Anontwi**: http://sourceforge.net/projects/anontwi/files/**
- A continuación, tenemos que "declarar" la aplicación en nuestra cuenta de Twitter (aplicación de terceros) y obtener las Consumer Keys o claves de consumidor.
- Para ello accedemos a **https: //dev.twitter.com/apps**, hacemos clic en "**Create new application**" y rellenamos el formulario.

No debemos olvidar dar permisos de "lectura, escritura y DM" a la aplicación (pestaña "Settings"):

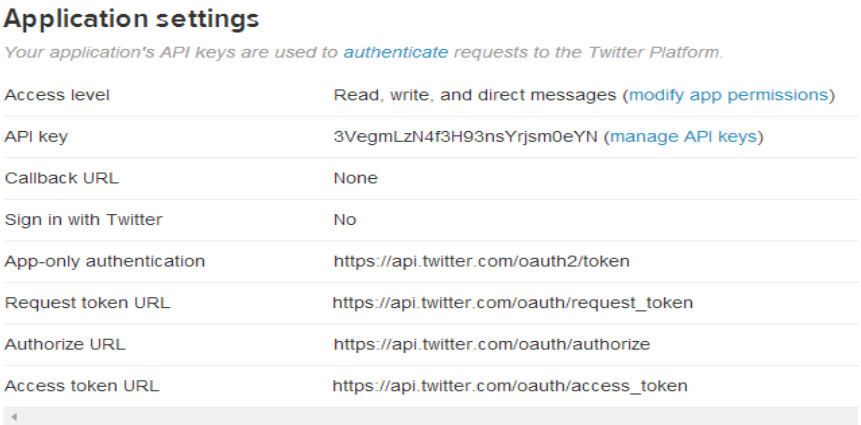


Figura 62: Información de la aplicación Twitter

- Nos interesa las keys

  "Consumer key: xxxxxxxxxxxxxxxxxxx

  Consumer secret: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

- Ahora debemos copiar las "Consumer Keys" y añadirlas al fichero config.py:

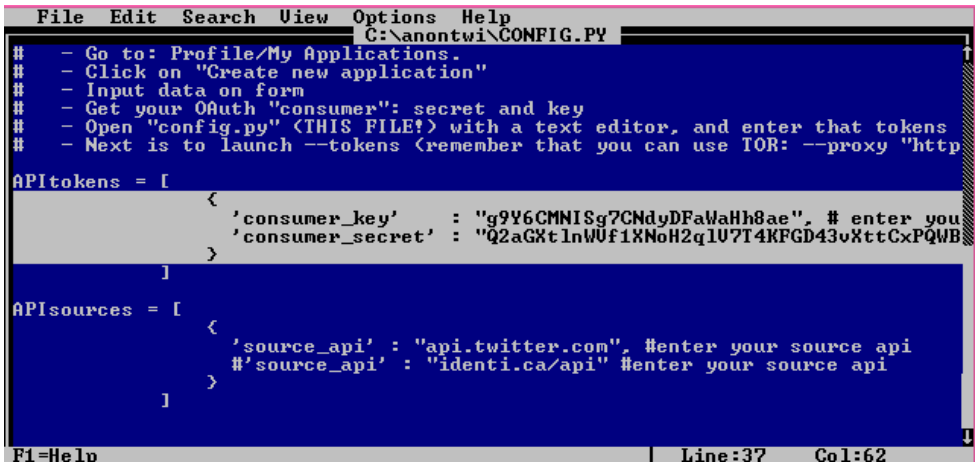


Figura 63: Añadiendo Keys al fichero confi.py

- Y finalmente generar los tokens para el API de Twitter a través de la página de desarrolladores o mediante el comando:

d:\Python27\python.exe anontwi –tokens

USO

. /anontwi [-m 'text' | -r 'ID' | -d @user | -f @nick | --df @nick] [OPTIONS] 'token key' 'token secret'

- Ahora que tenemos todo preparado vamos a probar a mandar un mensaje directo cifrado:

D:\anontwi>d:\Python27\python.exe anontwi -m "hola!" -d "@peraltaeymi"—enc—pin "abcdefghijklmnop" 139367345-VWtNqgrmfqmWs2JYGY01TqNpK86i9NqG9TqNUtRj 7zrMiHJrOBSGgb5kzqHH6R5zPpZYcwCcwS1ShKT6Y

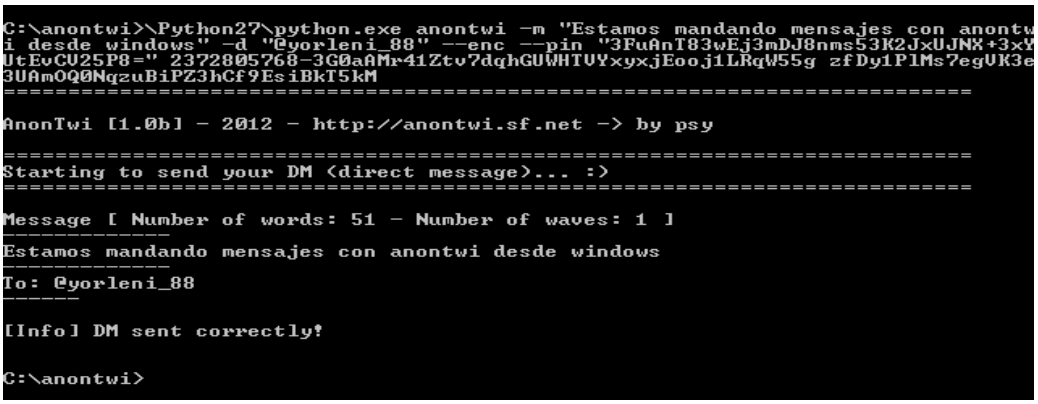


Figura 64: Mandando mensaje Anontwi

- Desciframos el mensaje en raw mediante la clave/PIN que previamente hemos facilitado:

D:\anontwi>d:\Python27\python.exe anontwi—dec "99JgAqyGjvo5RI5dQ+ARrXPj8jiHIdA

SW2c/pM2O93gB7urQXjugpkkS73+Ye6EhiqJ+UIFSG8nom1EAAdFH1i3D2T7vG1i1IFO2wsi26hY=" -pin "abcdefghijklmnop"

8.6. FIREGPG

Permite a los correos webmail implementar seguridad a través de la encriptación de los mensajes. Se ha hablado de un software (Freenigma) que permite cifrar los correos de Gmail, aunque es necesario registrarse en su web. FireGPG es otra solución que permite hacerlo con una extensión para Firefox. Primero necesitas instalar el GnuPG, que es la versión GNU del GPG. Una vez instalada la extensión podremos cifrar y descifrar los mensajes enviados desde Gmail, pero también firmarlos.

La extensión no solo funciona con Gmail, en realidad podemos usarla con cualquier página donde exista un campo de texto, ya que solo habrá que seleccionar la parte que queremos cifrar o firmar y presionar un botón.

FireGPG, es un complemento libre, gratuito, y en castellano para el Firefox, que permite hacer todas estas tareas de forma gráfica, y además incluye un soporte mejorado para Gmail, con lo que el robot que escanea el contenido de vuestros correos para generar publicidad personalizada (sí) deberá jubilarse definitivamente.

Es muy fácil de usar: Una vez instalado y reiniciado el navegador, mostrará un asistente que permitirá generar tus claves sin complicación, y exportar la clave pública con la misma dificultad.

Una vez hemos instalado el software necesario arrancaremos el programa WinPT (que viene con Gpg4win) para gestionar nuestras claves. En el caso de no disponer de un par de claves creadas para nuestra cuenta de GMAIL, deberemos generarla con WinPT y establecer una clave para firmar, cifrar y descifrar los mensajes [45].

Instalación de FireGPG

Esta herramienta corre igual en Linux y en Windows tiene la misma funcionalidad y los mismos pasos de instalación. En este caso se probara solo en Windows para que puedan ver su funcionamiento. FireGPG es un complemento de Firefox bajo GPL que nos ofrece una interfaz gráfica para cifrar, descifrar, firmar o verificar la firma de cualquier texto en una página usando GnuPG. Cabe mencionar que para instalar este complemento usamos la versión 3.6.22 de Mozilla, dado que otras versiones no lo soportan.

- El primer paso que debemos hacer es instalar GnuPG, para ello visitamos la página. http://www.gnupg.org. Después de esto, en la página de descargas de FireGPG instala la extensión pinchando sobre Download FireGPG: http://es.getfiregpg.org/?page=install&lang=es.

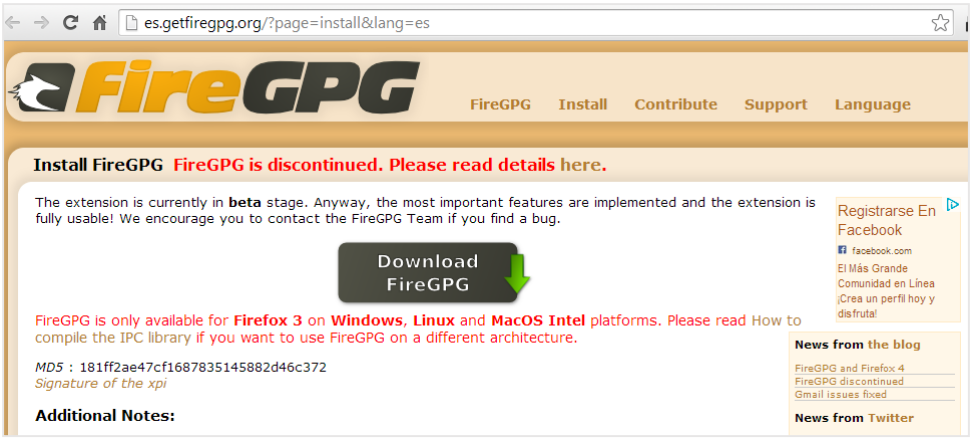


Figura 65: Página de FireGPG

- Nos hace una advertencia sobre la instalación del complemento, y damos en Instalar ahora. Luego reiniciamos el navegador.

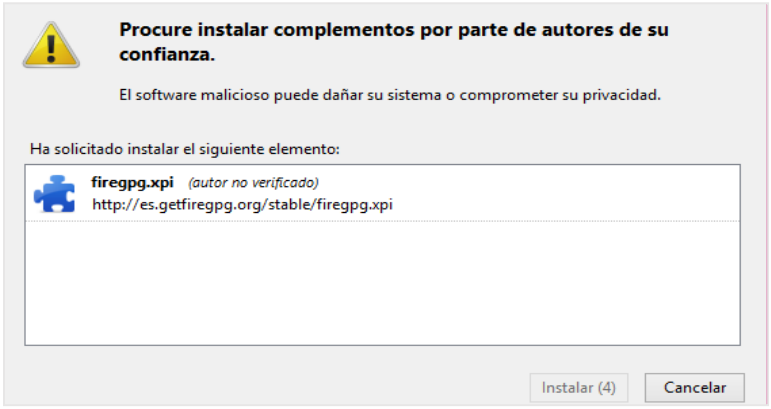


Figura 66: Instalando complemento FireGPG

- Cuando volvemos abrir el Mozil6la Firefox, nos dice que se ha instalado el complemento. Nos da un hipervínculo a Opciones de FireGPG.

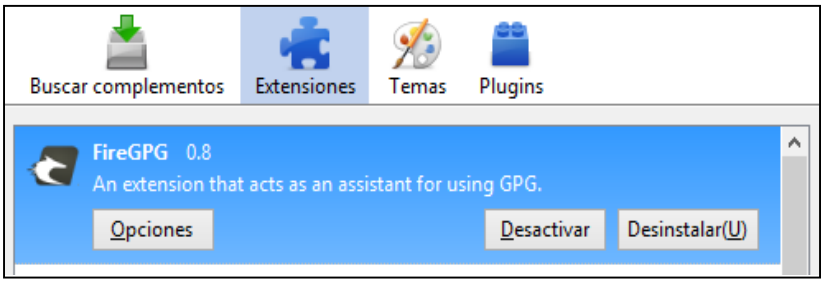


Figura 67: FireGPG instalado

- Entramos a Opciones para Configurar la ubicación de GnuPG For Windows. Por casualidad mira que contiene cada Pestaña. Empecemos por la Pestaña General.

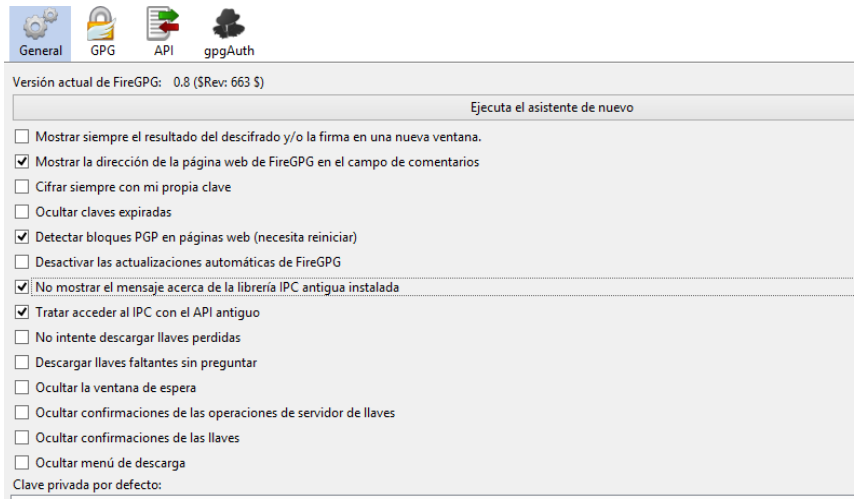


Figura 68: Opciones de GnuPG

- En la Pestaña GPG es la que tenemos que configurar las opciones de GnuPGP. Cabe mencionar que ruta del ejecutable lo encontramos en **C:\Archivos de Programa\GNU\GnuPG\gpg.exe.**

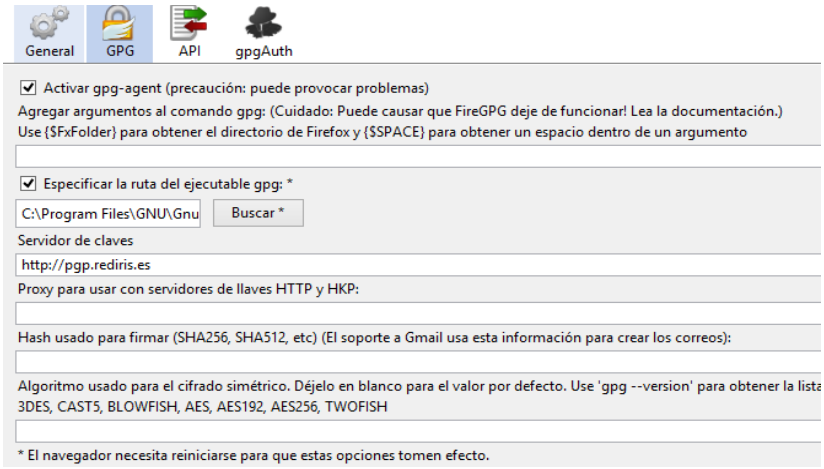


Figura 69: Pestaña GPG

- En la Pestaña API, puedes permite diseñar un sitio Web que utilice GPG sobre el cliente; por ejemplo para autentificar un usuario cuando accede a un panel administrativo.

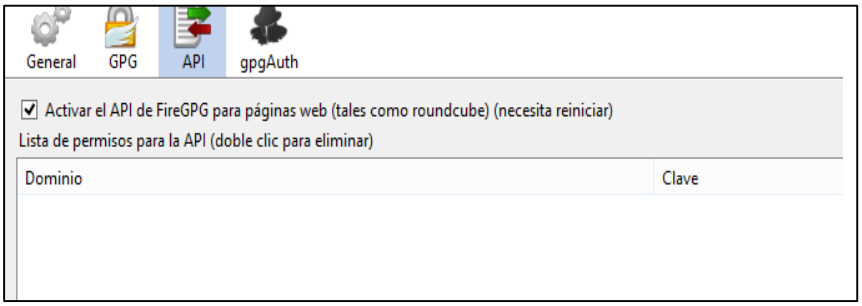


Figura 70: Pestaña API

- En la Pestaña gpgAuth, podemos definir qué hacer cuando se acceda una página Web.

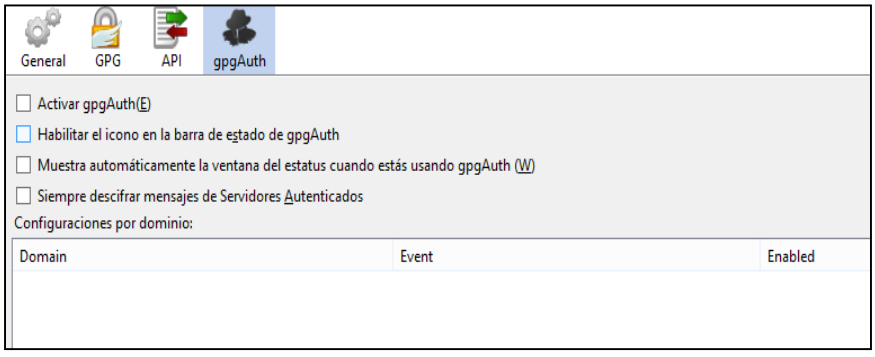


Figura 71: Pestaña gpgAuth

- Damos Aceptar y reiniciamos de nuevo Mozilla Firefox para que tome los cambios y listo abrimos nuestra de Nuevo el Navegador y comprobamos que si está instalado por la pestaña Herramientas o Clic Derecho. Al reiniciar aparecerá el asistente de configuración del navegador, si no es así simplemente tenemos que seguir la secuencia del menú superior Herramientas > FireGPG > Opciones y presionar sobre Ejecuta el asistente de nuevo.

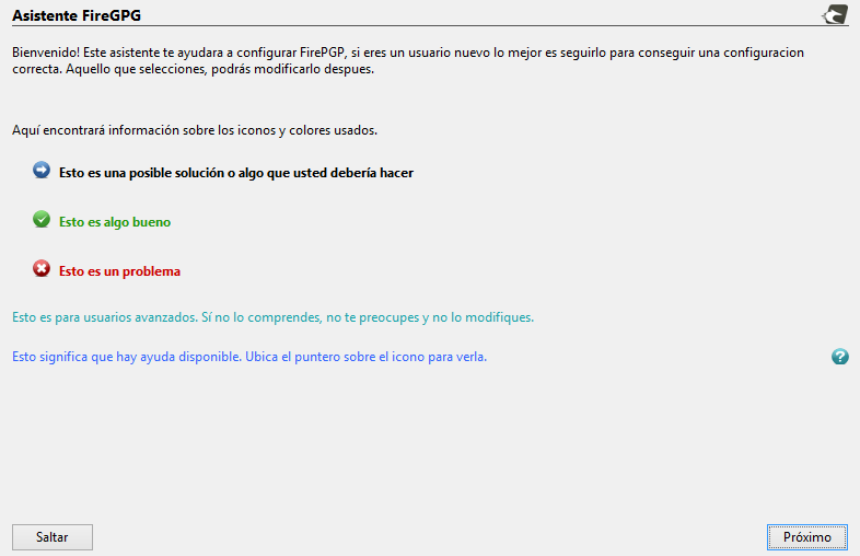


Figura 72: Asistente FireGPG

- Presionamos sobre Próximo. En la siguiente ventana observamos que FireGPG no ha sido capaz de encontrar el software de gestión de claves GnuPG, tendremos que indicarle la ruta presionando sobre el botón Buscar. Si hemos realizado la instalación por defecto de GnuPG en un Sistema Operativo Windows en español dicha ruta será:

C:\Archivos de Programa\GNU\GnuPG\gpg.exe

Hecho esto el asistente mostrará una nueva ventana donde se ha de pulsar sobre Siguiente. Asumiremos que no tenemos experiencia previa con GPG y que por tanto no tenemos nuestro par de claves pública y privada. En la nueva ventana que aparece Establecer un directorio base, que es donde vamos a guardar las claves, luego nos vamos a generar nuestras claves:

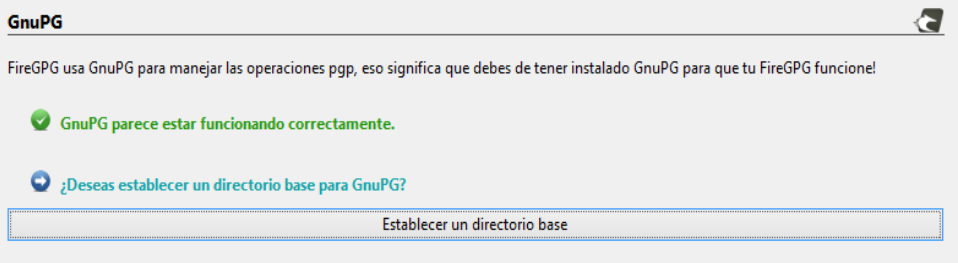


Figura 73: Estableciendo directorio base

Normalmente un par de llaves identifica a una persona. El uso más común de GPG es para cifrar correos y verificar (mediante firma de mensajes) la autenticidad del remitente de los correos (recordemos que SMTP es un protocolo donde se puede falsificar el remitente de forma trivial). Esta es la razón por la cual el asistente solicita nuestro nombre y dirección de correo a la que queremos asociar el par de claves.

El asistente también solicita una contraseña, si bien se pueden generar claves sin contraseña, es recomendable fijar una contraseña (robusta) para que si alguien consigue robar nuestra clave privada no sea capaz de usarla sin tener esa segunda credencial adicional. También es conveniente fijar un periodo de expiración de la clave (1 año por ejemplo), así, aunque tanto nuestra clave privada como la contraseña de esta hubieran sido interceptadas dejarían de valer al año y el atacante no podría continuar suplantándonos.

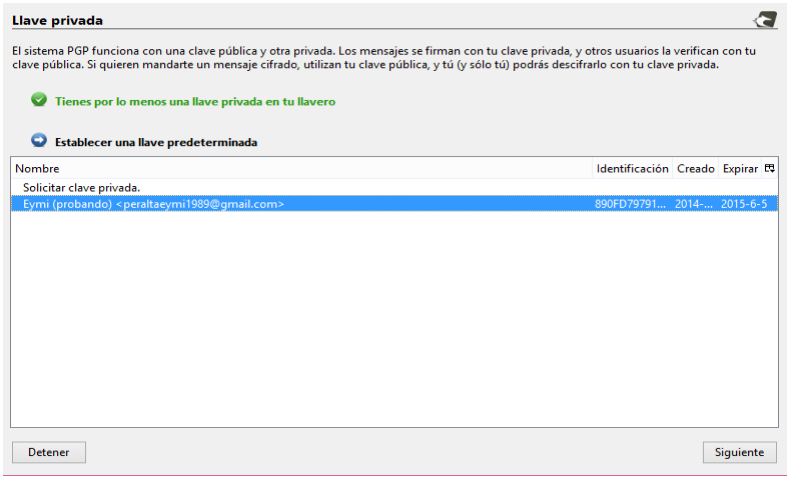


Figura 74: Generación de la llave privada

- El proceso de generación de claves es lento, has de ser paciente. Una vez generadas las claves aparecerá una ventana para integrar la aplicación con Gmail (inyectando botones de cifrado y firmado en la interfaz de Gmail). El desarrollador de la extensión ha descontinuado el soporte con Gmail así que ignoraremos de momento esa funcionalidad, esta guía será genérica para cualquier servicio (Gmail, Hotmail, YahooMail, etc.). Presionamos sobre siguiente en la ventana de configuración de la funcionalidad Gmail.

Figura 75: Pestaña Gmail FireGPG

- Dejamos las opciones por defecto en la nueva ventana del asistente y presionamos sobre siguiente.

Figura 76: Opciones del asistente FireGPG

- La siguiente ventana es ya la ventana final del asistente, simplemente hay que presionar sobre Cerrar.

Cifrando Correo

- Ahora vamos a la cuenta de correo con Gmail y comprobemos que funciona. Y enviamos un correo cifrado. Después de redactarlo lo que queremos que lleve el texto, vamos a Herramientas, FireGPG, Cifrar. Recuerda que también se puede por clic derecho, FireGPG, Cifrar.

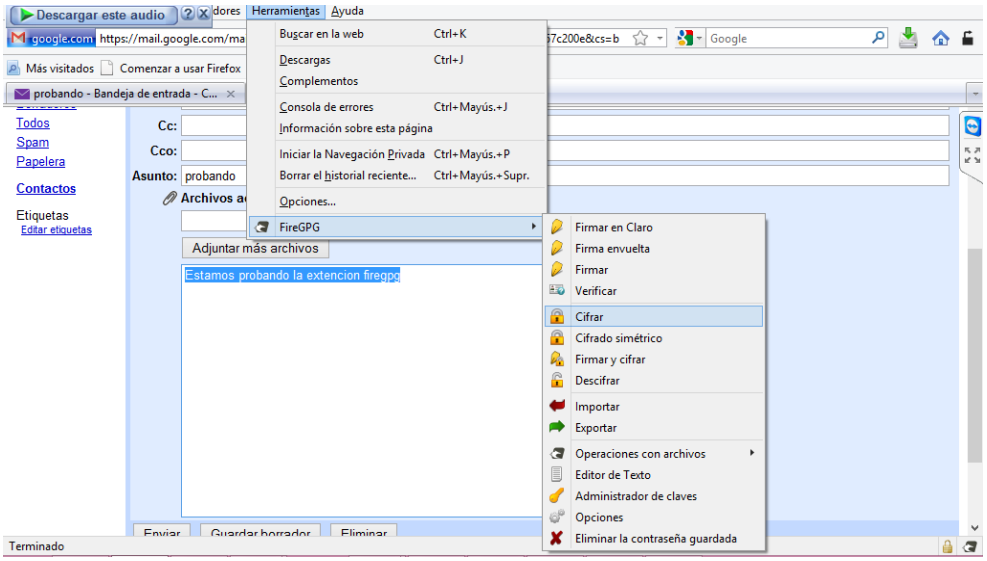


Figura 77: Cifrando correo

- Como podemos ver FireGPG nos tomó las llaves Privadas de los Usuarios creados en GnuPGP, seleccionamos el nuestro, cabe advertir que yo tengo dos cuentas creadas con diferente correo electrónico, damos Aceptar.

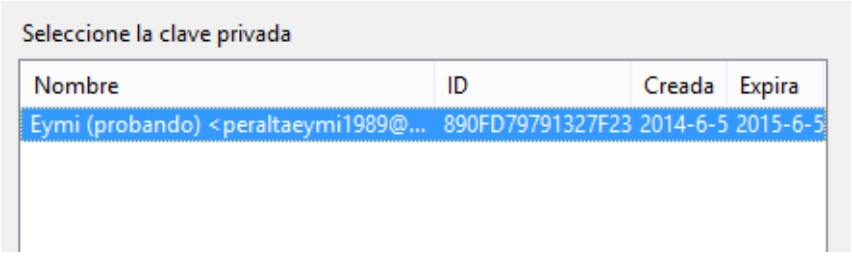


Figura 78: Llave privada del usuario GnuPGP

Nos pide la Contraseña de la clave privada que le dimos en GnuPGP.

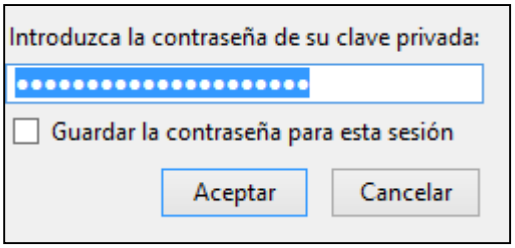


Figura 79: Contraseña de la clave privada

- Como solo dijimos que lo íbamos cifrar copiamos está en el texto del correo quitando lo que habíamos escrito. Porque así es la forma correcta. Le damos Enviar. Al no ser que lo firmemos en claro.

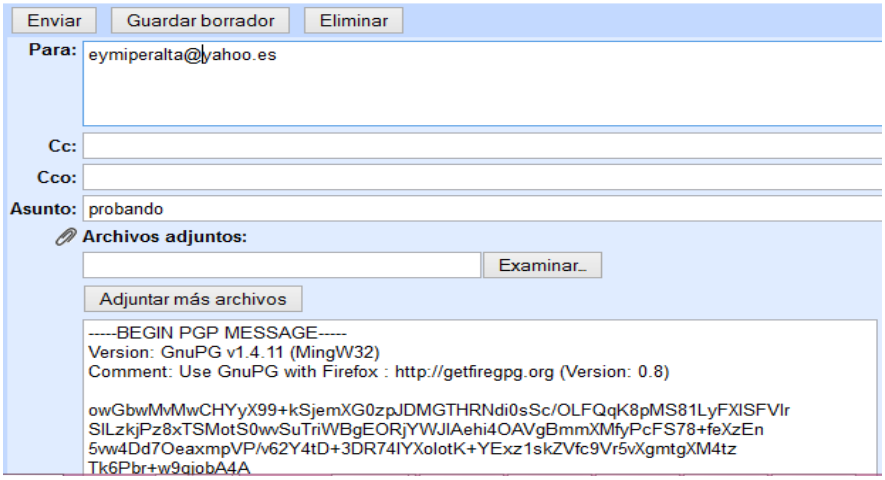


Figura 80: Poniendo texto firmado en el correo

- Comprobamos que si llego el correo y vamos a verificar que tiene el texto cifrado. Lo copiamos en FireGPG o más sencillo lo seleccionamos y le damos clic derecho FireGPG, descifrar.

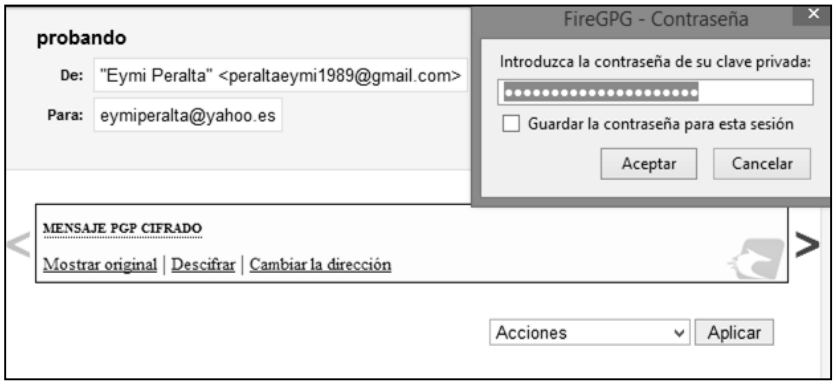


Figura 81: Verificando correo

- Si le damos en mostrar original podemos ver el correo tal como nos lo mandaron:

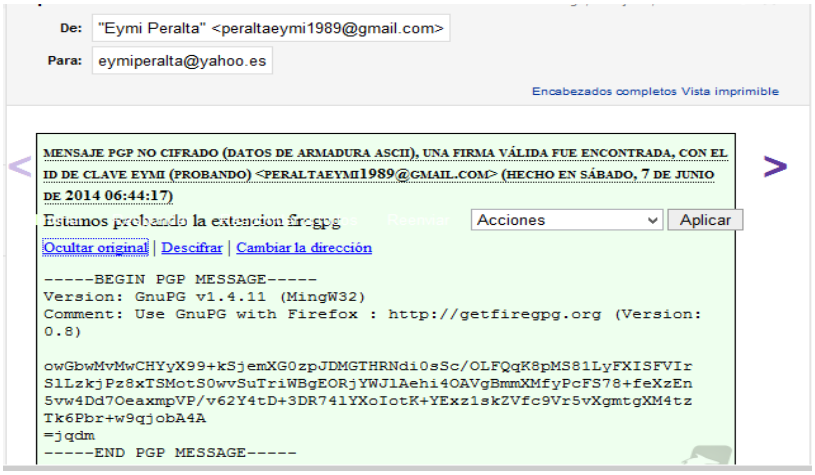


Figura 82: Revisando correo cifrado

- Y desciframos el correo, de esta manera podemos comprobar que el mensaje enviado fue cifrado y llego con éxito.

Figura 83: Descifrando el mensaje

Firmar correos electrónicos

En clase pudimos apreciar que falsificar el remitente de los correos electrónicos es una labor trivial. Una forma de asegurarnos de la autenticidad de la identidad de quien dice escribirnos es mediante la operación de firma.

Recuérdese que cuando se firma lo que se hace es cifrar una función resumen (hash) del mensaje con nuestra clave privada. Componemos un correo electrónico como de costumbre, seleccionamos el contenido a firmar y pulsamos sobre Firmar en Claro en el menú de FireGPG:

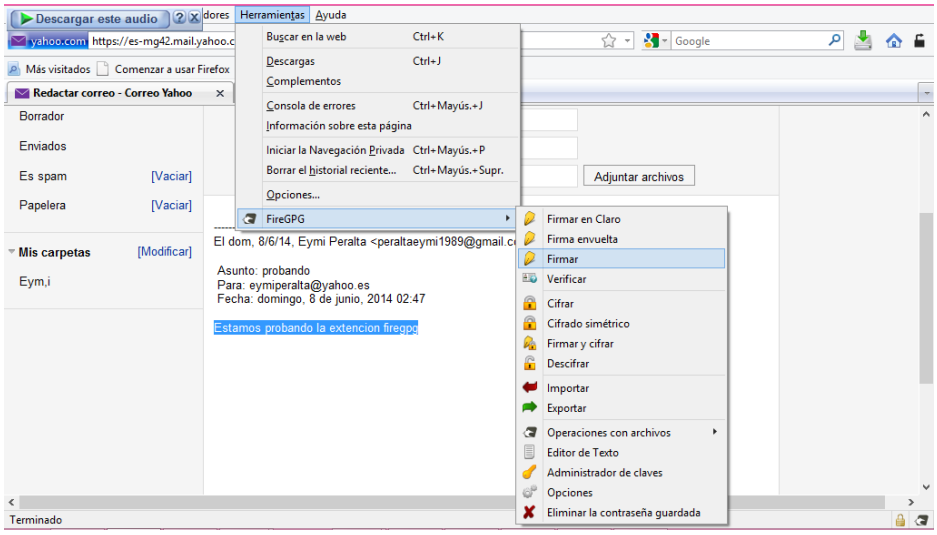


Figura 84: Firmando correo

A continuación se abre una ventana para que seleccionemos la clave privada de la cuenta que ha de actuar como remitente del mensaje.

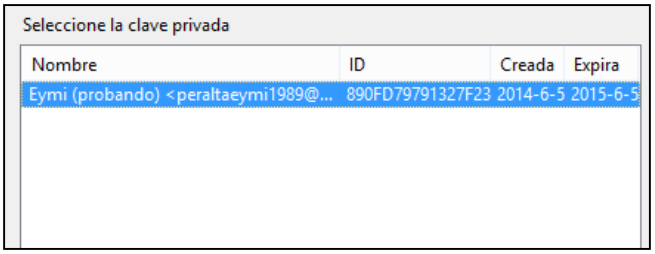


Figura 85: Clave privada

Hacemos la elección y pulsamos sobre Aceptar, se nos solicita la contraseña que le pusimos a la clave privada, una vez introducida aparece el conglomerado de firma. Es este conglomerado el que debemos copiar y sustituir por el contenido del correo electrónico original, es muy importante que se copie todo o FireGPG no será capaz de pasar el mensaje en el lado del destinatario para hacer la comprobación de firma.

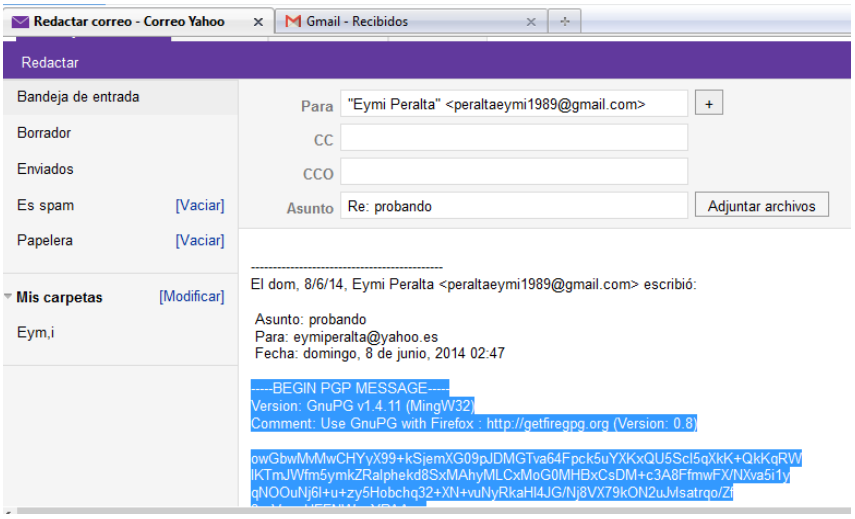


Figura 86: Enviando correo firmado

Comprobar la firma de un correo electrónico

Para comprobar la firma de un correo electrónico lo que se hace es descifrar la secuencia de firma del remitente con su clave pública para obtener el hash origen del contenido que se supone que va firmado. Luego se calcula ese mismo hash para el contenido recibido y se comparan, de ser iguales la firma queda verificada. Todo este proceso está implementado en una única función en FireGPG. Para poder comprobar la firma de un mensaje tendremos que haber importado la clave pública del remitente previamente.

Imaginemos que se recibe un mensaje firmado como el siguiente, basta con seleccionar toda la secuencia PGP y presionar sobre el botón Verificar del menú de FireGPG.

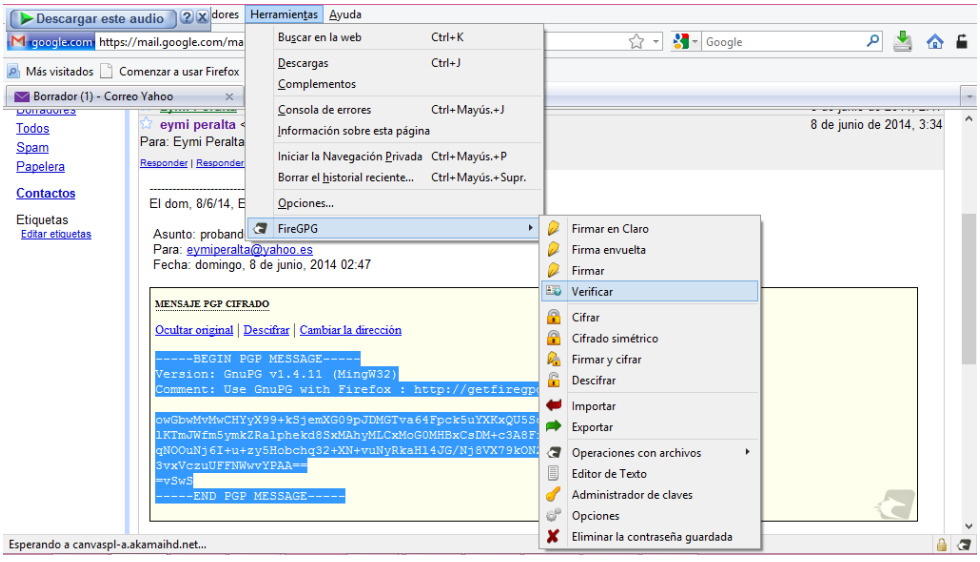


Figura 87: Verificando correo recibido

Si la firma es correcta y tenemos la clave pública del remitente en nuestro gestor de claves FireGPG nos informará de que el remitente es quien dice ser:

Nótese que también se puede cifrar y firmar al mismo tiempo empleando la opción Firmar y Cifrar del menú de FireGPG.

Figura 88: Correo en claro

Nota: FireGPG solo funciona para navegadores Firefox 3.6.23, esta herramienta no está actualizada.

8.7. CRYPTOCAT

Cryptocat es una aplicación web libre destinado a permitir chatear en línea de forma cifrada y segura. Cryptocat cifra las conversaciones (chats) en el lado del cliente (usuario), confiando sólo en el servidor con datos ya cifrados. Cryptocat se ofrece como una aplicación para GNU/Linux y Mac OS X, como una extensión para los navegadores Mozilla Firefox, Google Chrome, 3 y Apple Safari.

Cryptocat proporciona un medio para comunicaciones cifrado interrumpido, proporcionando mayor privacidad que servicios como Google Talk, al mismo tiempo que mantiene un nivel más de accesibilidad que otras plataformas de cifrado de alto nivel, 4 5 y además permite múltiples usuarios en una sala de chat

Funcionalidad

Cryptocat usa el protocolo Off-the-Record Messaging (OTR) para cifra mensajes privados. A partir de que Cryptocat genera pares de llaves para cada conversación (chat), implementa un formulario confidencialidad directa perfecta (perfect forward secrecy). Cryptocat también puede ser utilizado de manera conjunta con Tor a fin de mantener anónima los detalles de conexión de cliente del servidor. El proyecto también planea crear una versión embebida o integrada para dispositivos Raspberry Pi.8 9

El desarrollador de Cryptocat Nadim Kobeissi fue arrestado e interrogado en la frontera de Estados Unidos por el Departamento de Seguridad Nacional de los Estados Unidos en junio de 2012 sobre la resistencia a la censura. Y difundió el incidente en Twitter; lo que provocó cobertura de los medios y un incremento en la popularidad de Cryptocat.

El 2013, la red de Cryptocat migró a Bahnhof, el proveedor de alojamiento web sueco conocido por estar construido en el interior de un refugio anti-nuclear de la Guerra Fría.

Cuando escribes un mensaje usando Cryptocat, cada mensaje es cifrado en tu computador antes de enviarlo. Sólo los receptores previstos tienen la clave para descifrar y leer el mensaje, previniendo así que se revela a “terceras partes”.

Gato Rojo recibe el mensaje y lo descifra usando su clave secreta. El mensaje es configurado para que sólo Gato Gris pueda descifrarlo utilizando su clave secreta. Cuando el mensaje pasa por el servidor, ya está en cifrado. En caso de que se pudiera interceptar, el Perro Espía no podría interceptarlo.

Aunque Cryptocat tiene como objetivo ofrecer una mensajería instantánea fuertemente cifrado, es importante tener en cuenta en lo que Cryptocat no te protege:

- No te anonimiza pese a que tus comunicaciones están cifradas, tu identidad puede rastrearse ya que Cryptocat no oculta tu dirección IP. Para anonimizarte, te recomendamos usar Tor.
- No te protege de que haya programas que almacenen lo que escribes tus mensajes están cifradas mientras viajan por la red, pero eso no quiere decir, necesariamente, que tu teclado esté seguro. Cryptocat no te protege contra hardware o software que almacena lo que escribes y que se mande a “terceras partes”.
- No te protege contra personas de poca confianza los receptores de tus mensajes pueden derivar tus mensajes sin que lo sepas. Cryptocat tiene como objetivo que tus mensajes sólo lleguen a los receptores que escojas, pero no aseguro que esas personas sean de confianza o no.

- Es una nueva aplicación para comunicarse directamente de forma absolutamente privada. Para ello, sobre protocolos como XMPP y OTR, Cryptocat utiliza el clásico cifrado asimétrico, a partir de las claves privada y pública del usuario.
- Es un software gratuito y libre, licenciado bajo la GNU Affero General Public License. Funciona a través del navegador web, gracias a los complementos disponibles para Firefox, Chrome y Safari. En breve, además, se espera la llegada de las aplicaciones para dispositivos móviles.
- El problema de Cryptocat es el mismo que el de aplicaciones como Wickr: va a ser muy difícil que la gente que te rodea pase de GTalk, Facebook Messenger o WhatsApp, para dar el salto a otra cosa. Pero, por lo menos, las alternativas están ahí. Aunque sean para algo puntual [46].

Instalación de Cryptocap

En la instalación de Cryptocat, esta herramienta es multiplataforma ya que funciona igual en Linux y en windows su funcionamiento y instalación no varía en nada, debido que es una extensión que puede ser utilizada en navegadores de Firefox y Chrome. En este caso le mostraremos instalación en Linux.

En la pestaña de herramienta del explorador Mozilla en donde está instalado este complemento nos dirigimos a donde dice Cryptocat desde allí nos aparecerá una ventana en donde nos dirá que coloquemos un nombre de chat puede ser cualquiera y también un usuario este puede ser cualquier user o también elegimos un chat Facebook esta opción la elegimos nosotros, en este caso probaremos los 2 alternativas que nos ofrece.
Nos dirigimos a la herramienta Cryptocat de forma automática inicia a cargar colocamos el nombre de la sala de chat y nuestro nombre este puede ser cualquiera pero, hay que recordar decir a la persona que nombre usaremos para que este nos encuentre hechos esto le damos conectar e inicia a genera claves.

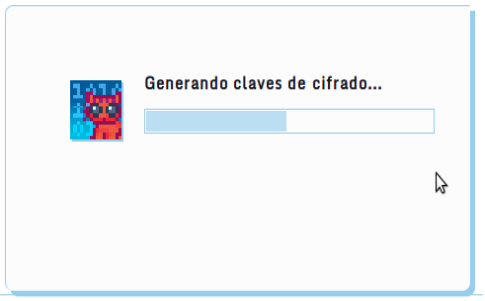


Figura 89: Generación de calves en Cryptocat

Una vez que genera la clave nos muestra un muro, donde aparecen personas conectadas personas que están haciendo uso de esta herramienta. En la parte derecha de esta nos muestra personas es desde allí es en donde podemos elegir a la persona con que deseamos iniciar un chat.

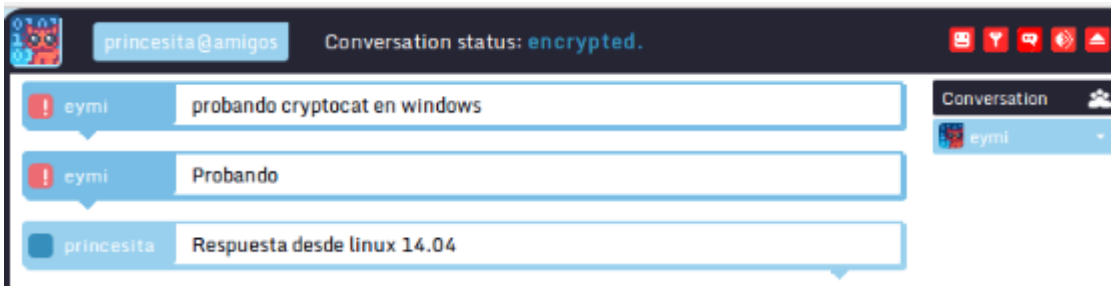


Figura 90: Chat via criptocap

En cuando a la integridad si la persona que estamos chateando es quien creemos que es iniciamos una autentificación ejemplo una pregunta donde la respuesta solo debe de ser conocida por ambas persona que están intercambiando información este caso por princesita y eymi.

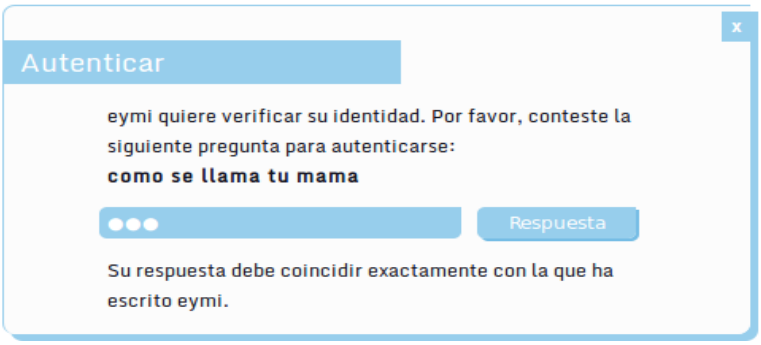


Figura 91: Autentificación de las partes

Si princesita responde bien la respuesta estamos seguros de quien es Eymi.

Cryptocat con Facebook

Una vez que iniciamos Cryptocat lo iniciamos con Facebook este de forma inmediata nos abre una venta como la siguiente:

Figura 92: Cryptocat con facebook

En esta ventana colocamos los datos de la cuenta de Facebook para usar Facebook nuestro correo y clave e inicia a generar claves.

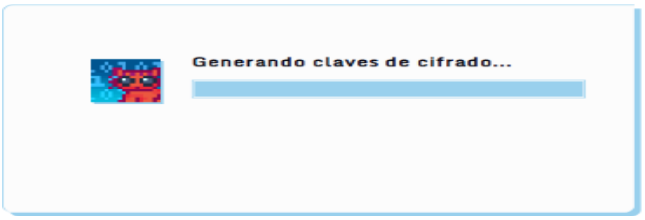


Figura 93: Generación de claves

Una vez que ha generado las claves. Nos carga la interfaz de cryptocap con todos los amigos que están conectado en ese momento con Facebook, los que aparecen con el gato rojo son los que usa cryptocap.

Figura 94: Chat usando cryptocat con facebook

En este caso solo Eymi está haciendo uso de la herramienta, iniciamos un chat con Eymi. Esto es una chat usando cifrado. Podemos ver los mensajes cifrado en Facebook como podemos ver está cifrado mientras que en cryptocap está en claro.

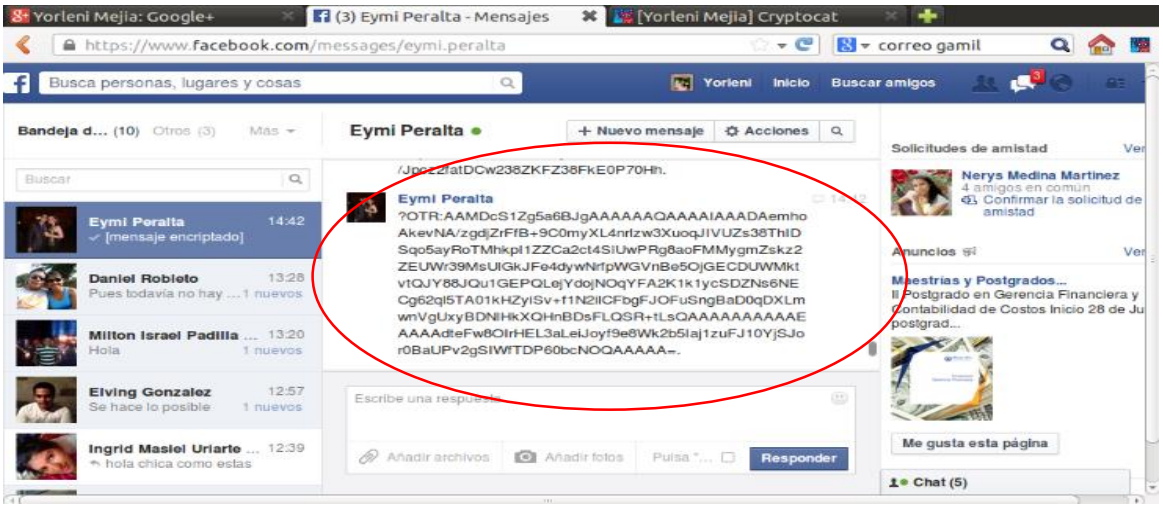


Figura 95: Mensaje encriptado en Facebook

Los mensajes encriptados en Facebook son más largos depende la cantidad de palabras que se escriban desde cryptocap.

En pidgin que es una mensajería de texto el mensaje o los mensajes se verían de la siguiente manera:

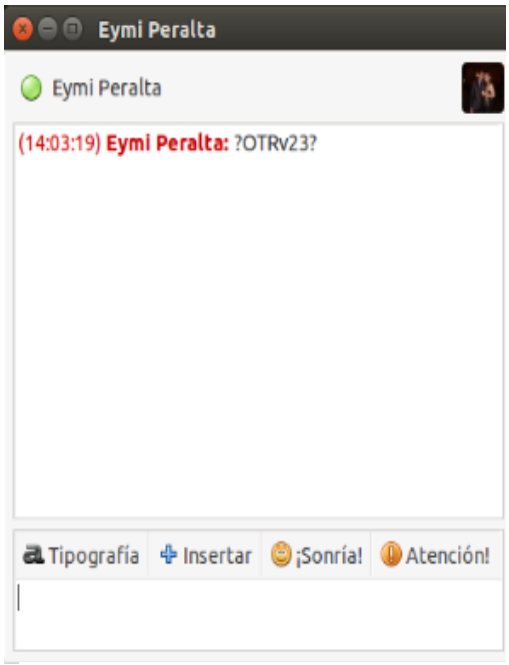


Figura 96: Mensaje encriptado en Pidgin

Estos mensajes van encriptado si ambos están usando cryptocap, pero si elegimos una persona de las que nos apareen conectada del facebook los mensajes viajan en claro sin encriptar.

8.8. Netcat

NetCat es un servicio de red de la computadora para la lectura y escritura en las conexiones de red mediante TCP o UDP. NetCat está diseñado para ser un back-end fiable que se puede utilizar directamente o fácilmente conducido por otros programas y scripts. Al mismo tiempo, se trata de una depuración y una investigación de la red herramienta rica en características, ya que puede producir casi cualquier tipo de correlación el usuario podría necesitar y tiene una serie de capacidades incorporadas.

NetCat se refiere a menudo como una “navaja suiza del ejército para TCP / IP”. Su lista de características incluye el escaneo de puertos, la transferencia de archivos, y la escucha de puertos, y que puede ser utilizado como una puerta trasera.

Características

Algunas de las principales características de netcat son:

- Conexiones salientes o entrantes , TCP o UDP , hacia o desde cualquier puerto
- Absoluto de DNS y avance / retroceso de cheques, con advertencias apropiadas
- Capacidad de utilizar cualquier puerto de origen local,
- Capacidad para utilizar cualquier dirección de origen de la red de configuración local
- Capacidades integradas de escaneo de puertos , con asignación al azar
- Capacidad incorporada de origen - ruta libre
- Se puede leer el comando argumentos de la línea de la entrada estándar
- Slow send modo , una línea cada N segundos
- Volcado hexadecimal de los datos transmitidos y recibidos
- Capacidad opcional para permitir que otro servicio de programa a establecer conexiones
- Opcional telnet -opciones de respuesta [49].

Instalación de Netcat

Netcat es una utilidad imprescindible para todo profesional de la seguridad. Es denominada la “navaja suiza de la seguridad de red” porque sirve para innumerables cosas.

Fue creada en 1995 por “El Hobbit”. La versión original fue desarrollada para sistemas Unix y Linux pero Weld Pond desarrolló la versión para Windows NT en 1998. El código fuente de ambas versiones está disponible.

No tiene interfaz gráfica. Se utiliza desde líneas de comandos.

Lo podemos descargar desde netcat.sourceforge.net una vez descargado lo descomprimimos de la siguiente manera para poder instarlo.

1. $ tar xvfz <nombre_paquete>.tar.gz
2. $ cd <nombre_carpeta_descomprimida>
3. $ ./configure
4. $ make
5. $ sudo make install

Utilidades

1. Chat.
2. Enviar y recibir ficheros.

Chat

En una de las máquinas ponemos el netcat en modo servidor, a la escucha. En la otra, lo ponemos en modo cliente.

Iniciamos conexión con servidor usando la siguiente orden este tiene que iniciarse antes que el cliente para que este pueda responder: nc -l -p 5000

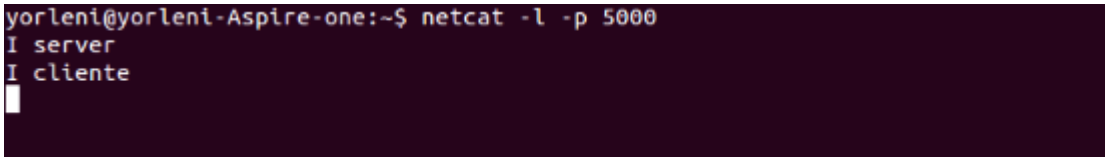


Figura 97: Corriendo netcat en modo servidor

Ahora que ya poseemos una conexión entre cliente servidor podemos iniciar una comunicación vía consola.

nc	NetCat
- l	Indica al netcat que debe actuar como un servidor, es decir, debe poner a la escucha un puerto (Modo Servidor)
- p	Indica el puerto por el que ponemos el servidor a la escucha. Si no se pone, netcat Selecciona un puerto que esté libre de forma aleatoria. Es recomendable usar un puerto superior al 1024.

Tabla 10: Parámetros de Netcat

Cuando lanzamos el netcat en modo cliente, este actúa como lo hace un telnet. De hecho, se podría lanzar la máquina cliente con un telnet (telnet <ip_servidor><puerto_servidor>).

Una vez que hemos lanzado el servidor ya podemos hacer consulta, ahora procedemos y lanzamos el cliente y desde aquí iniciamos una conversación entre 2 laptop diferente de la siguiente manera: nc <ip_servidor><puerto_servidor>

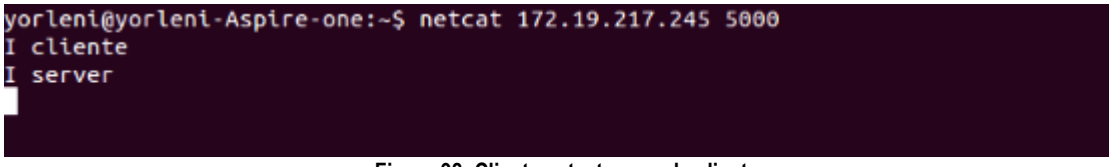


Figura 98: Cliente netcat en modo cliente

Enviar y Recibir Ficheros

Para pasar un fichero de un cliente al servidor, hay que hacer lo siguiente:

Ponemos el servidor a la escucha:

nc -l -p 5000

Pasamos el fichero desde el cliente al servidor:

nc <ip_servidor><puerto_servidor><dirección del archivo (ubicación)

Si queremos que ese fichero se almacene, en vez de ser mostrado, deberíamos lanzar el servidor así:

nc -l -p 5000 > dirección del archivo (ubicación)

Enviamos un documento pequeño desde el cliente:

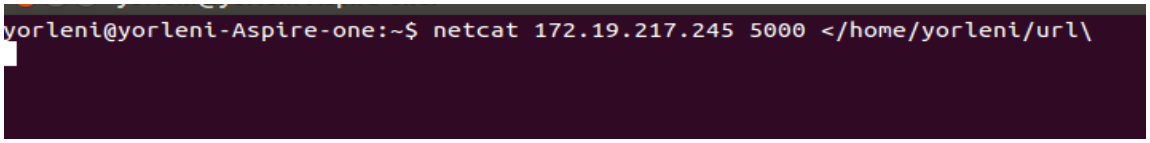


Figura 99: Envió de archivo desde netcat

Este de forma automática será reflejado en el servidor:

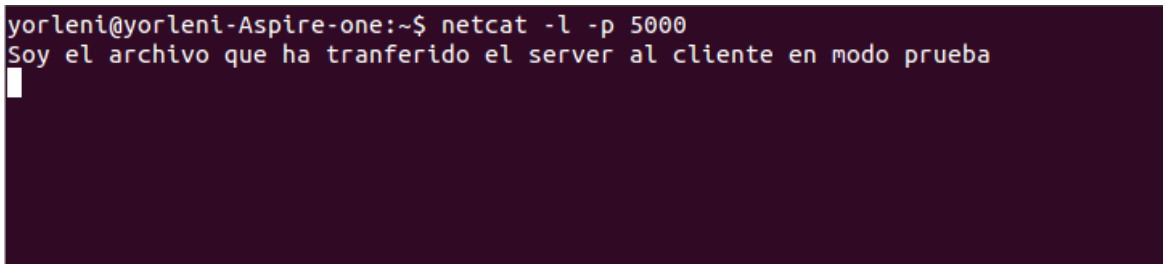


Figura 100: Servidor enviando archivo asía el cliente

Ahora vemos el mensaje del documento que enviamos desde el lado del servidor.

Podemos comprobar, haciendo un netstat, como aparece un nuevo proceso escuchando por el puerto 5000 por todas las interfaces (192.168.1.6:5000).

8.9. SECURE GMAIL

Secure Gmail. Es una extensión para Chrome que nos permite enviar correos cifrados con una contraseña.Esta extensión se funciona igual en Linux y en windows, sin ningún problema. Es muy fácil de usar: cuando esté instalada, al lado del botón "Redactar" aparecerá un botón con un candado.

Figura 101: Candado para enviar un mensaje cifrado

Pulsando ahí empezamos a escribir un correo normal y corriente. Cuando lo enviemos, aparecerá un diálogo en el que se nos pedirá una contraseña y una pista.

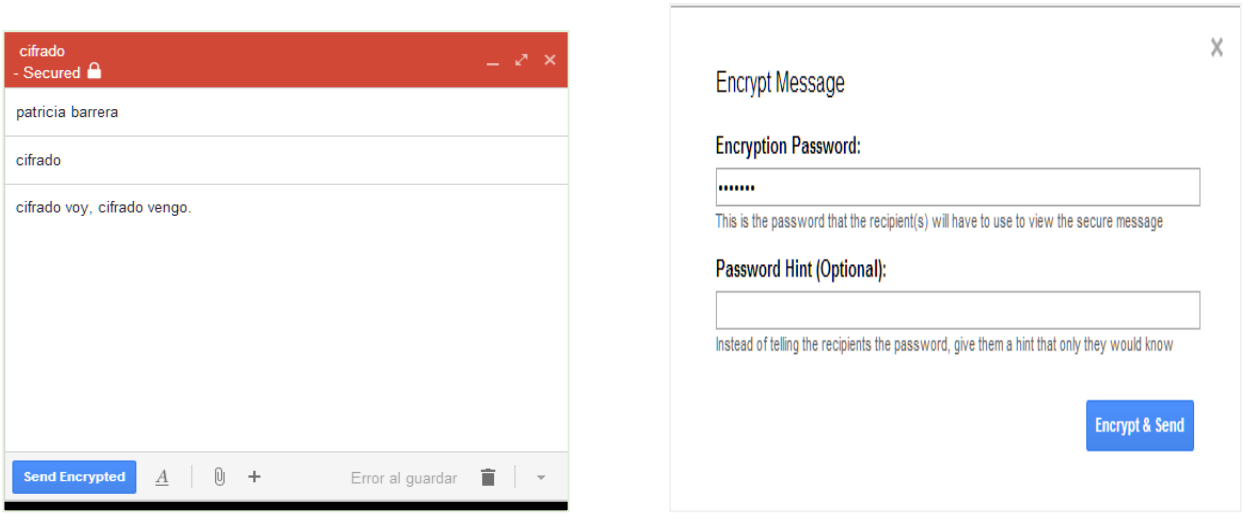


Figura 102: Escribiendo un mensaje para luego cifrar

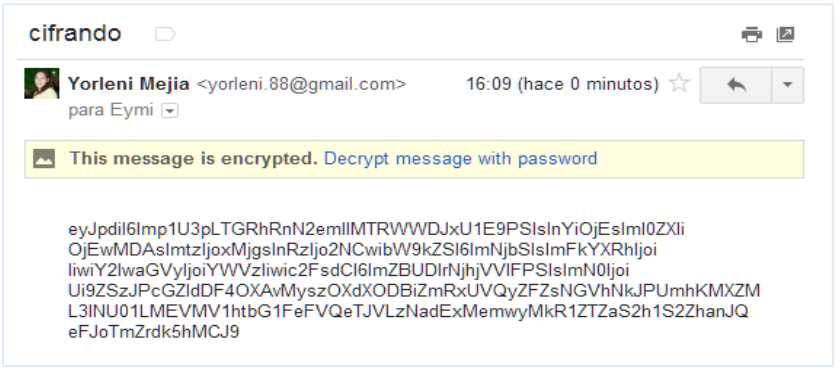


Figura 103: Recibiendo correo

Secure Gmail es código abierto y usa cifrado simétrico AES256. Es menos seguro que usar clave pública/clave privada (aquí tienes que pasar la contraseña a tu compañero de alguna forma) pero sí que es muchísimo más fácil de usar. Si os preocupáis por la seguridad y no queréis que Google tenga la posibilidad de leer ciertos correos, Secure Gmail es una opción muy a tener en cuenta.

8.10. MAILVELOPE

Haga clic en el icono de abajo para ir a la tienda web de Chrome. Elija + Añadir a Chrome para instalar Mailvelope como una extensión en su navegador. https://www.mailvelope.com/

Figura 104: Extensión mailvelope

Una vez que hemos elegido a que navegador deseamos agregar nos manda a la siguiente página, en donde solo damos agregar en mi caso elegí Chrome.

Figura 105: Instalando mailvelope

Si todo ha salido bien hasta ahora, en la barra de tarea lo vemos en forma de un candado  , esto quiere decir que se ha instalado correctamente. Si le damos clic derecho podemos ver las opciones que trae.

Public-Key Crypto graphy

OpenPGP y, por tanto Mailvelope utiliza criptografía de clave pública, que significa una clave se divide en dos partes: **públicos** y **privados** claves con diferentes propósitos:

Clave pública - Se utiliza para cifrar un mensaje. Puede y debe estar disponible para todos.

Clave privada - Se utiliza para descifrar un mensaje. Necesita ser almacenado de forma segura. El acceso está restringido por contraseña.

ntercambio de Claves

Para poder enviar mensajes de correo electrónico cifrados a un compañero, debe tener la clave pública del destinatario. Por lo tanto, antes de la comunicación segura puede suceder entre dos personas, tienen que cambiar sus claves públicas entre sí. Hay varias formas de que las claves públicas pueden ser distribuidas:

- Se envía por correos electrónicos a los socios específicos. Vea la sección Exportar clave para saber cómo se puede hacer esto con Mailvelope
- Publicar la clave en un sitio web para que todos puedan acceder.
- Subir una llave a un servidor de claves.

Manipulación Clave

Haga clic en el icono de bloqueo de Mailvelope en la barra de herramientas de extensión para el navegador para abrir el menú principal.

Generación de claves

Para utilizar Mailvelope, al menos un par de claves (que consiste en una clave pública y privada) debe estar disponible. Tenemos la posibilidad de generar un nuevo par de claves como se explica en esta sección, o importar un par de claves existente, como se describe a continuación. Haga clic en Generar clave para abrir el diálogo de generación de claves:

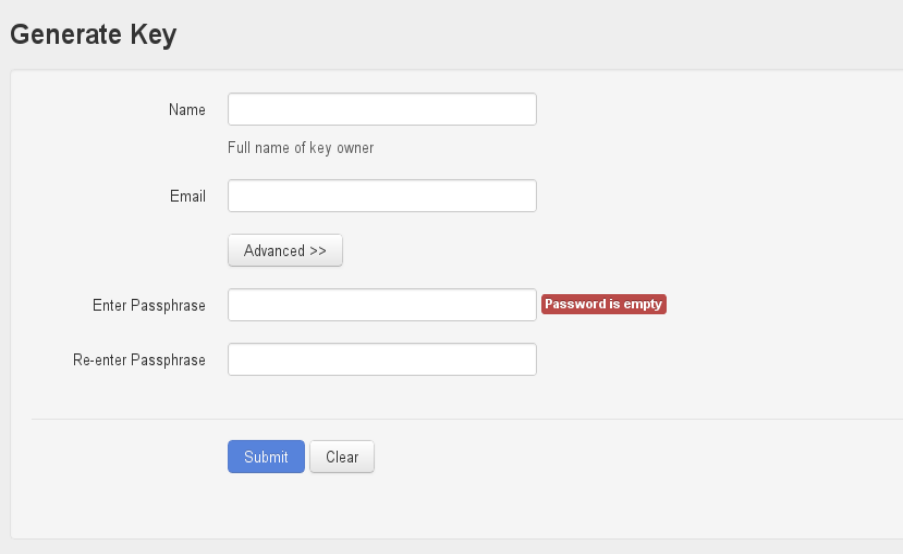


Figura 106: Generando claves RSA

Complete toda la información requerida. Después de hacer clic en Enviar, el proceso de generación de claves se iniciará y el resultado se puede ver mediante la navegación de regreso a la lista de claves.

Figura 107: Proceso de espera de la generación de clave

Si ha salido bien nos muestra el siguiente mensaje: Success.

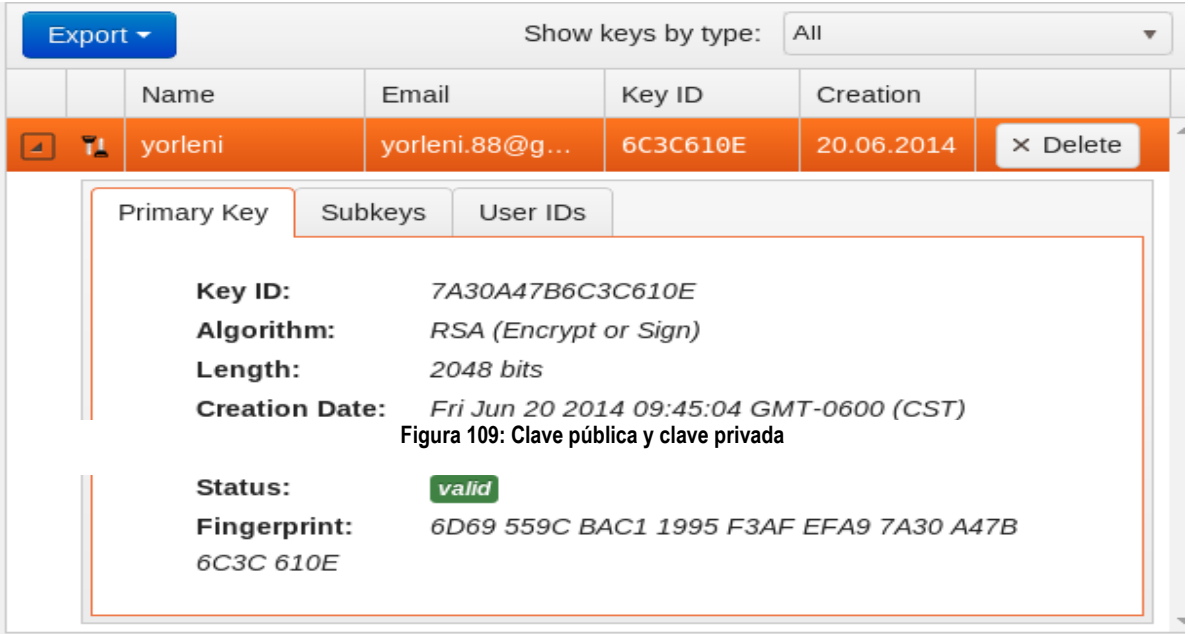


Figura 109: Clave pública y clave privada

Figura 108: Clave generada

En display keys podemos ver la key generada.

Key Export

Funcionalidad de clave de exportación se utiliza para extraer las claves en formato de texto. Podemos usar esto para publicar claves públicas o para hacer una copia de seguridad de un par de claves pública-privada en un lugar seguro.

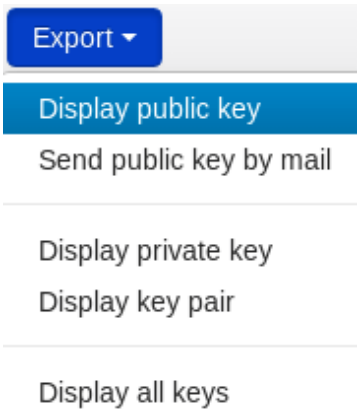


Figura 110: Opciones para ver las claves

Opciones de export

- Visualizar la clave pública: Un emergente aparece con la clave pública. Las opciones posibles son Copiar en el portapapeles y Crear archivo.
- Enviar clave pública por correo electrónico: Esta intentará abrir su aplicación de correo electrónico e introduzca la clave pública como texto en un nuevo correo electrónico. Limitaciones: sólo funcionará si la clave pública no supere una determinada longitud.
- Visualizar la clave privada: Las mismas opciones que la clave pública.
- Visualizar par de clave: Igual que el anterior. Par de claves se muestra en dos bloques principales separados. El método preferido para hacer una copia de seguridad del par de claves completo.

Importando la clave

Importamos la clave de nuestro compañero. Así la tendremos en nuestro anillo de claves y podemos descifrar los mensajes que nos envía.

Enviando mensaje

Mailvelope extiende la interfaz de usuario de correo web (por ejemplo, Gmail, Yahoo, etc) con los controles que permiten la encriptación y desencriptación de correo electrónico.

Al iniciar a redactar un correo, aparece el icono botón de composición  se muestra en todos los correos electrónicos, componen las áreas del proveedor de correo web y pondrá en marcha un editor externo de Mailvelope.

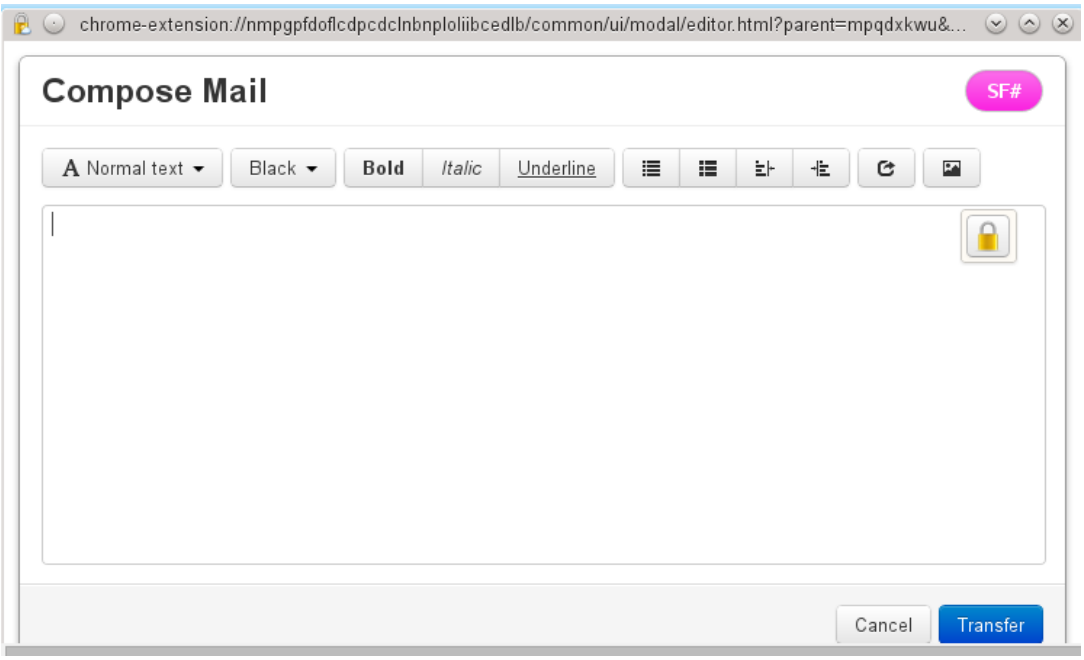


Figura 111: Editor de texto de Mailvelope

El correo electrónico ahora se puede componer. A continuación, haga clic en el botón cifrar  para mostrar el diálogo. Aquí se puede elegir los destinatarios que se permitirá descifrar el mensaje.

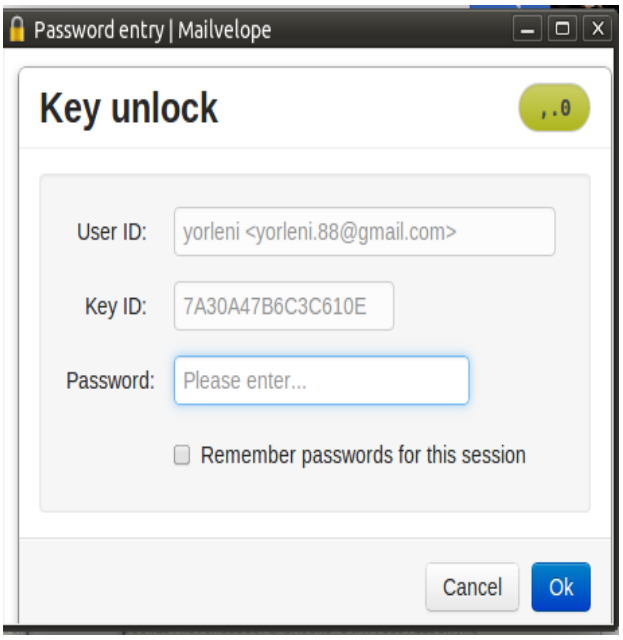


Figura 112: Eligiendo destinatarios

Haga clic en Ok para cifrar el correo electrónico. El texto del correo electrónico será reemplazado por el mensaje cifrado.

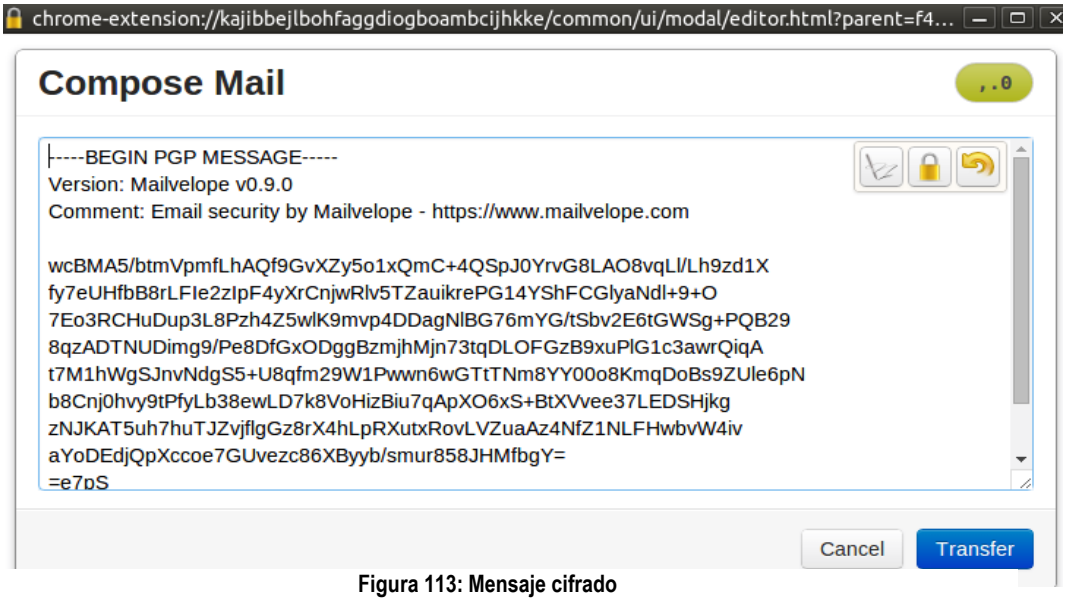


Figura 113: Mensaje cifrado

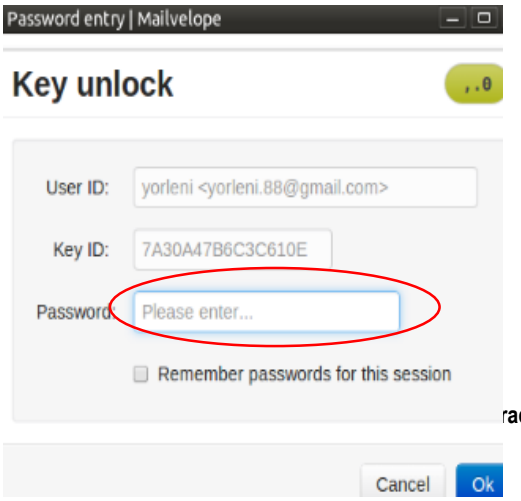
Descifrado de mensajes

Siempre Mailvelope detecta un mensaje cifrado en un correo electrónico que marca con una plantilla:

Figura 114: Recibiendo correo

Este es un mensaje recibido, para poderlo abrir debemos de tener la clave privada de quien lo está mandado para poderlo descifrar.

Nota: introducimos el password en texto claro.



rado

Figura 115: Introduciendo contraseña para el mensaje

Para poder enviar mensaje debemos de intercambiar clave, ya que si no se comparten nunca podremos acceder a lo que dice el mensaje que se nos está enviando.

8.11. Plugins de Firefox obligatorios para usar tor y i2p

Los siguientes addons o plugins de Firefox son “obligatorios” si quieres proteger tu anonimato incluso usando TOR y I2P:

- **NoScript:** evita que Javascript (código que puede ejecutarse en tu navegador) se ejecute aleatoriamente (sirve además para deshabilitar Google Analitics que monitoriza casi todo el tráfico de la red).
- **CS Lite:** permite bloquear las cookies por lista negra. El NoScript de las cookies.
- **TorButton:** para poder usar Tor en el navegador. Añade funciones de seguridad y también modifica el user-agent.
- **RefControl:** modifica la cabecera HTTP Referer, que puede mandar información a un servidor sobre la página web de la que provienes.
- **BetterPrivacy**: te permite evitar y gestionar cookies inborrables que suele usar Google,Ebay, y muchos otros seres oscuros del ciberespacio para rastrear tu navegación.
- **User Agent Switcher:** te permitirá cambiar el perfil y la configuración habitual de tu navegador para que no se pueda identificar como aquél que usas para otras cosas. En principio con TorButton activado esta extensión no es necesaria.
- **QuickJava:** te permitirá activar y desactivar Javascript y evitar así que la página que visites pueda adquirir información adicional sobre tu computadora (p.e. tamaño de tu pantalla). Tampoco es necesaria si bloqueas todo el Javascript con NoScript.
- **Https Everywhere:** es un plugin desarrollado por la EFF Electronic Frontier Foundation para firefox y chrome, y viene instalado por defecto en Tor Bundle.
- Plugin de Firefox para **Scroogle scraper** (búsquedas anónimas en Google) con ssl.

8.12. Plugins de Firefox de alerta y asegurar el cifrado

Los siguientes plugins de Firefox (un navegador libre, muy completo y con muchas extensiones), pueden ayudarte en la tarea de estar alerta y asegurarte el cifrado la identidad de la web que visitas:

- **Force-TLS:** Fuerza al navegador a usar la versión con TSL de una web (si la tiene): obliga a un número de sitios a realizar todas sus solicitudes a través de un canal seguro SSL y mientras algunos sitios, como Amazon, actualmente no tienen la opción de este protocolo seguro, pero las principales empresas como Facebook, Twitter, Google, etc todos permiten usar el HTTPS. Force-TLS ayuda a administrar los sitios que se deben cargar a través de HTTPS solamente. Permitiendo importar / exportar y administrar estos ajustes.
- **Certificate Patrol:** te muestra los nuevos certifiados y los compara con los antiguos. Es una extensión para el navegador Firefox que avisa al usuario cuando el certificado de una página web ha cambiado.
- **Perspectives:** permite comparar los certificados por terceras personas. es una extensión que nos permite navegar en los sitios web de forma segura que utilizan el protocolo HTTPS, validando sus certificados por medio de notarías de red.
- **Safe:** para evitar enviar datos sobre conexiones no seguras.

IX. INTRODUCCIÓN A LA SEGURIDAD EN DISPOSITIVOS MÓVILES

No hace muchos años los terminales móviles se encontraban aislados de muchos riesgos de seguridad al no estar interconectados con la Red. Pero actualmente, la inmensa mayoría incluye mecanismos para que puedan conectarse y descargar contenido de Internet, leer el correo electrónico, etc., y por tanto, enfrentarse a las mismas amenazas de seguridad que los equipos. Es por esto que, se hace necesario un análisis de las prestaciones que incorporan actualmente los terminales móviles y cómo las aprovechan los internautas también, cómo afectan estas prestaciones en los hábitos de los usuarios en cuestión de seguridad.

Los dispositivos móviles constituyen uno de los principales, medio de comunicación que utilizamos en la actualidad. Cada vez son utilizados por más personas, y cada vez se emplean en un mayor número de escenarios y ámbitos diferentes, desde el ocio personal, hasta el acceso a datos corporativos con un alto nivel de confidencialidad, pasando por multitud de aplicaciones de uso cotidiano, como el acceso al correo electrónico o a las redes sociales y la navegación por Internet. Las estadísticas más recientes corroboran el impacto actual de estas tecnologías móviles:
6.800 millones de líneas móviles activas en 2013, dato impresionante teniendo en cuenta que la población mundial es de 7.100 millones.

2.100 millones de líneas de banda ancha móvil activas en 2013, con un crecimiento anual medio de un 40% en los últimos 6 años.

1.700 millones de teléfonos móviles vendidos en 2012, de los cuales más de un 40% eran Smartphone. 100 millones de tablets vendidos en 2012, creciendo vertiginosamente desde 19 millones en 2010.

Habitualmente los dispositivos móviles integran múltiples tecnologías de comunicación inalámbrica, con lo que la información acaba siendo transferida a través de diferentes métodos por el medio de comunicación compartido que es el aire. La mayoría de los dispositivos móviles actuales ofrecen las siguientes tecnologías:

NFC, comunicaciones de corta distancia con grandes expectativas de futuro para aplicaciones como pagos a través de los dispositivos móviles.

Bluetooth, para comunicación con auriculares inalámbricos, para audio con vehículos (llamadas y música), y para intercambiar datos entre móviles o con otros dispositivos, como ordenadores.

Wi-Fi, para acceso a Internet (navegación web, redes sociales, correo electrónico, e innumerables otras aplicaciones) sin consumir el ancho de banda contratado al proveedor de comunicaciones móviles. GSM/GPRS/EDGE (2G), UMTS (3G), LTE (4G), para enviar y recibir llamadas, mensajes SMS, y para acceso a Internet (con aplicación similar a Wi-Fi).

Lamentablemente, todas estas tecnologías presentan vulnerabilidades y riesgos de seguridad. Por ello, es importante conocer cuáles son las principales amenazas asociadas y adoptar las medidas de protección adecuadas para eliminar, o al menos reducir, nuestro nivel de riesgo.

RECOMENDACIONES DE SEGURIDAD GENERALES

Independientemente de la tecnología inalámbrica empleada, se recomienda de manera general deshabilitar el interfaz asociado a las diferentes tecnologías de comunicaciones siempre que no se esté haciendo uso del mismo, con el objetivo de minimizar la posibilidad de ser objetivo de posibles ataques. Los interfaces de comunicaciones del dispositivo móvil deberían permanecer deshabilitados hasta que se desee hacer uso de sus capacidades. Esta recomendación aplica especialmente a los interfaces inalámbricos que no son utilizados ampliamente a lo largo del día, como por ejemplo NFC, Bluetooth, o Wi-Fi.

En entornos corporativos se recomienda hacer uso de capacidades de gestión centralizadas (MDM, Mobile Device Management) para establecer la configuración adecuada de todos los interfaces de comunicaciones inalámbricas de los dispositivos móviles gestionados.

Es muy recomendable verificar y aplicar de forma periódica las actualizaciones de software y firmware proporcionadas por el fabricante del dispositivo móvil y que potencialmente solucionan las vulnerabilidades de seguridad conocidas que afectan a los drivers y a la pila de comunicaciones de las diferentes tecnologías inalámbricas: NFC, Bluetooth, Wi-Fi y 2G/3G.

Para las comunicaciones de datos a través de TCP/IP, tanto Wi-Fi como 2G/3G, se recomienda utilizar, siempre que sea posible, protocolos de comunicación seguros, es decir, autenticados extremo a extremo, y cifrados. Por ejemplo, acceder a sitios web mediante HTTPS en lugar de HTTP, configurar las aplicaciones de correo para que se conecten con el servidor mediante IMAPS en lugar de IMAP, emplear soluciones de mensajería instantánea que hagan uso de cifrado, y así sucesivamente con las diferentes alternativas de protocolos, cuando las haya. No obstante, es importante ser consciente de que en muchas ocasiones el usuario no puede elegir el protocolo o protocolos de comunicación que utilizarán las distintas aplicaciones.

A nivel corporativo es necesario definir una política de seguridad para los dispositivos móviles con capacidades NFC, Bluetooth y Wi-Fi, que analice su configuración por defecto y establezca la lista de dispositivos (NFC y Bluetooth) o redes (Wi-Fi) permitidas, el tipo de información que estos pueden almacenar, y en qué tipo de entornos pueden ser utilizadas estas tecnologías, como por ejemplo dónde llevar a cabo el emparejamiento inicial entre dos dispositivos Bluetooth o si se permite la conexión a redes Wi-Fi abiertas. En ambos casos la concienciación de los usuarios finales acerca de las amenazas existentes juega un papel muy importante. Esta política debe ser monitorizada, por ejemplo, en los puntos principales de acceso a los edificios de la organización, permitiendo identificar dispositivos NFC, Bluetooth o Wi-Fi no autorizados o mal configurados.

Las tecnologías inalámbricas (NFC, Bluetooth, Wi-Fi, 2G/3G…) deben ser incorporadas a las auditorías de seguridad y pruebas de intrusión periódicamente planificadas sobre el resto de tecnologías empleadas por la organización, con el objetivo de identificar anomalías respecto a la política de seguridad definida así como debilidades y vulnerabilidades en los dispositivos móviles y en su utilización.

Finalmente, debe tenerse también en cuenta que las tecnologías inalámbricas son susceptibles de verse amenazadas de manera general por técnicas o ataques de Denegación de Servicio (DoS), siendo trivial la generación de suficiente ruido en el espectro de radiofrecuencia asociado a una tecnología inalámbrica concreta (mediante dispositivos conocidos como jammers) para no permitir el uso de la misma.

9.1. HERRAMIENTAS IMPLEMENTADAS PARA SEGURIDAD EN MÓVILES

9.1.1. Babel

Babel es una aplicación de mensajería iPhone y Android que le permite utilizar su teléfono celular para intercambiar mensajes de texto cifrados. Ofrece el más alto nivel de seguridad en caso de incumplimiento o de espionaje, y está diseñado para mantener los mensajes cifrados, no sólo durante las transmisiones, sino también en la memoria del dispositivo.

Características principales

Los mensajes se cifran no sólo durante la transmisión, sino también en la memoria del teléfono. Los usuarios no tienen que recordar nuevas contraseñas o claves, o compran y utilizan los certificados digitales. Es simple, intuitivo y utiliza un servicio estándar de iOS para contactos y mensajes. No se requiere ninguna conexión de datos, mensajes cifrados se envían a través de SMS estándar (servicio de texto). Si la conexión de datos está disponible, iMessages se utilizan automáticamente.

Los mensajes salientes se cifran utilizando AES de cifrado simétrico con una longitud de clave de 128 bits. Cada mensaje se cifra con una clave aleatoria diferente. Algoritmo Diffie-Hellman se utiliza para el intercambio de claves inicial entre las dos partes. Los mensajes cifrados se pueden intercambiar con cualquier persona que tenga un iPhone con iOS 6 o más reciente.

¿Por qué utilizar BABEL?

- *Compatibilidad con múltiples plataformas:* A diferencia de otras aplicaciones de cifrado de mensajes, puede utilizar BABEL para intercambiar mensajes con usuarios de otras plataformas
- *Privado:* Gracias cifrado de extremo a extremo, ningún tercero o servidor central está implicado en las transmisiones - usted no tiene que confiar en nadie más que su contraparte (contacto).
- *Seguridad*: los poderes de cifrado de la NSA-grado. Además mensaje cifrado a través del aire, que se cifran también en el propio dispositivo, en caso de que su teléfono se pierde o es robado - los mensajes y la privacidad se mantiene seguro y sin tocar.
- *Cifrado:* BABEL utiliza algoritmos probados y estándar para criptografía fuerte - AES para el cifrado de extremo a extremo de mensajes y Diffie-Hellman de claves criptográficas acuerdo
- *Simple:* No se requiere conexión de datos, BABEL utiliza el servicio de mensajería estándar (SMS).
- *Fácil:* Fácil de usar e intuitiva interfaz de usuario - una vez que se intercambian llaves entre usted y su contacto, que son capaces de enviar al instante y recibir mensajes cifrados (toma alrededor de un minuto para comenzar).

Instalación y funcionamiento

Primeramente instalamos la aplicación desde google play, puede ser desde el celular o desde la computadora:

Figura 116: Tienda de aplicaciones Android

Una vez instalada lo abrimos, nos pedirá nuestra clave que queramos poner, será la clave que compartiremos para poder desencriptar el mensaje. Luego de haber entrado con la clave podemos comunicarnos con la otra persona de forma segura ya que los mensajes son encriptados:

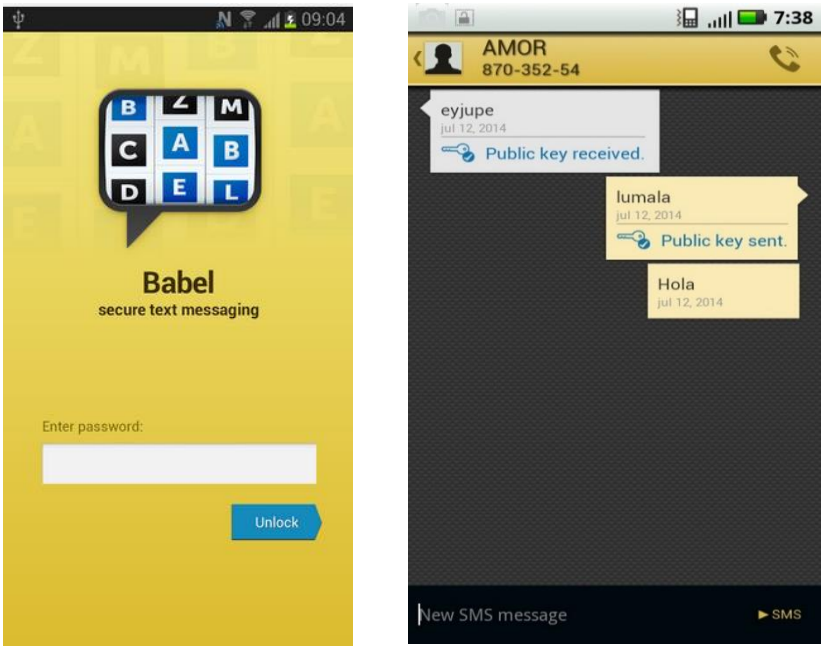


Figura 117: Funcionamiento Babel

9.1.2. Spy Phone

SpyPhone es un programa que se instala en el Celular / Móvil que deseamos monitorear, una vez instalado el mismo le brindara una copia de todos los eventos que la unidad interceptada realice incluyendo todas las llamadas.

La instalación se realiza de forma física. No es posible realizarlo en forma remota. El equipo puede usar la línea antigua o una nueva, funcionara con cualquier SIM/CHIP no importa que compañía utilice. Como primer paso usted debe tener un celular compatible ya sea el que el objetivo este usando o uno nuevo que pueda entregárselo luego de la instalación. Recuerde que el equipo para ser compatible con nuestra aplicación deberá contar con cualquiera de las siguientes plataformas; Android, los (Iphone), Blackberry o Symbian, si no está seguro de que OS es su equipo por favor ingrese a la página de GSM ARENA **http://www.gsmarena.com** y busque su modelo de movil/celular, el detalle técnico le confirmara que plataforma es.

Si el celular que tiene el objetivo a monitorear es compatible solo debe tomarlo "prestado" 10 minutos para descargar, instalar y configurar el programa, una vez hecho esto el programa se tornara invisible trabajando en segundo plano oculto mientras captura todas las comunicaciones las 24 horas los 365 días del año.

¿Qué Funciones de Spyphone?

1. Escucha Ambiental en VIVO
2. Escucha de comunicación celular en VIVO
3. Monitoreo de Envió y Recepción de SMS
4. Alerta e Historial de llamadas entrantes y salientes vía SMS en tiempo real
5. Localización Geográfica por medio de la Red Celular
6. Notificación de Apagado y Encendido
7. Control remoto de activación y desactivación por medio de comandos SMS o Web
8. Informe de Cambio de tarjeta SIM
9. Localización Geográfica por medio del módulo GPS - Geocerca Perímetro Configurable
10. Monitoreo de Envió y Recepción de Emails.
11. Monitoreo de Conversaciones WhatsApp con archivos adjuntos incluido.
12. Monitoreo de Fotografías (tambien reporta las fotos tomados anterior a la instalación)
13. Monitoreo de Videos (tambien reporta los videos tomados anterior a la instalación)
14. Grabación Automática de todas las llamadas recibidas y realizadas
15. Grabación Automática del Sonido Ambiente
16. Monitoreo de Agenda, Contactos por nombre y número telefónico
17. Monitoreo de URLS visitadas
18. Monitoreo de Conversaciones Line
19. Monitoreo de Chat Facebook
20. Monitoreo de Chat Skype
21. Desactivación y Desinstalación en Forma Remota
22. Monitoreo de Chat Viber
23. Monitoreo de Chat Wechat
24. Monitoreo de Conversaciones BlackBerry Messeneger PIN (solo equipos BlackBerry)

Spy Phone App - Mobile Tracker puede registrar:

- Los registros SMS/MMS. Esta aplicación de supervisión les ofrece acceso tanto a los mensajes SMS/MMS enviados, como a los mensajes recibidos. Podrá visualizar todos los detalles de las registraciones (fecha, detalles del expediente), el contenido escrito de los mensajes, la ubicación GPS del teléfono meta como también los ficheros multimedia adjuntos.

- Registros de llamada. El programa para espiar el teléfono podrá seguir y registrar los detalles de las llamadas recibidas y de las llamadas hechas. Tendrá acceso al nombre y al número del contacto, como también a la duración de la llamada.
- Los registros GPS. Los datos GPS también serán enviados con frecuencia al servidor. Podrá ver la colocación del dispositivo en el mapa pero también las coordenadas geográficas de dicha colocación.
- Los registros del Navegador. Esta aplicación de supervisión lleva la cuanta de las páginas visitadas a través del Navegador. Podrá ver el nombre del dominio principal pero también la fecha en la que fue visitado.
- Los registros de sistema. El sistema hará registraciones respecto al apago y al encendido del teléfono. Serán registradas también las modificaciones de los perfiles de sonido, como la activación de los modos Silencioso y Vibraciones.
- Los registros de facebook. El programa monitorizará la actividad del chat o de los mensajes Facebook de los teléfonos meta que tienen acceso root.
- Los registros whatsapp. El programa monitorizará la actividad whatsapp de los teléfonos meta ce tienen acceso ROOT.
- Los registros de imágenes. Este programa utilizado para espiar los teléfonos móviles guardará copias de las fotos hechas o de las imágenes descargadas.
- Los registros de los nuevos contactos añadidos. Este programa para espiar el teléfono registrará los detalles (nombre, número de teléfono) de los nuevos contactos añadidos en la Agenda del teléfono.
- Registros viber, solamente para teléfonos con root.
- Registros skype, solamente para teléfonos con root.
- Registros calendar. Para seguir el teléfono puedes descargar gratis la aplicación de la tienda Google Play o de nuestra página de web. Tendrá un periodo de prueba de 3 días durante el cual tendrá acceso completo a todas las funciones de la aplicación. Ulteriormente, podrá activar para cada teléfono registrado un período de prueba de 2 días más. Después de la expiración del período de prueba tendrá acceso completo a sus registros solamente si se suscribe (ofrecemos uno de los más bajos comisiones de abonamiento).

Instalación la de la aplicación en el teléfono que quiere monitorizar.

Descarga la aplicación de la tienda Google Play e instala la aplicación en el teléfono. Después de haber finalizado la instalación, utiliza el botón de apertura de la tienda Google Play. Cuando abre la aplicación por primera vez, se le solicitará registrar un cuento con una dirección de correo electrónico y una contraseña, que posteriormente utilizará para acceder a sus registros en la página de web.

Figura 118: Instalando SpyPhone

Inicio de sesión en nuestro sitio web con el correo electrónico y la contraseña que ha escrito antes.

Después de aproximadamente una hora, podrá verificar sus registros en la página web. El teléfono meta debe tener conexión a internet para poder mandar las registraciones al servidor.

Figura 119: Monitorizando al celular

9.1.3. Encrypted messages

Esta aplicación permite a las personas comunicarse en privado sin necesidad de utilizar una tercera servidores o servicios de otros fabricantes.

Utilice sitios públicos tales como twitter y facebook para pegar sus mensajes cifrados.

Tiene características tales como cuota de GPS donde se puede insertar su ubicación GPS actual y compartirlo codificado, el receptor recibirá una URL a Google Maps con la ubicación. Construido en los mensajes enlatados también.

Instalación

Primeramente lo descargamos, luego visualizaos su interfaz en donde podemos mandar mensajes encriptados a nuestros contactos, ya sea del móvil como mensaje de texto o como correo:

Figura 120: Mandando mensaje con encrypted message

Figura 122: Mandando mensaje al correo

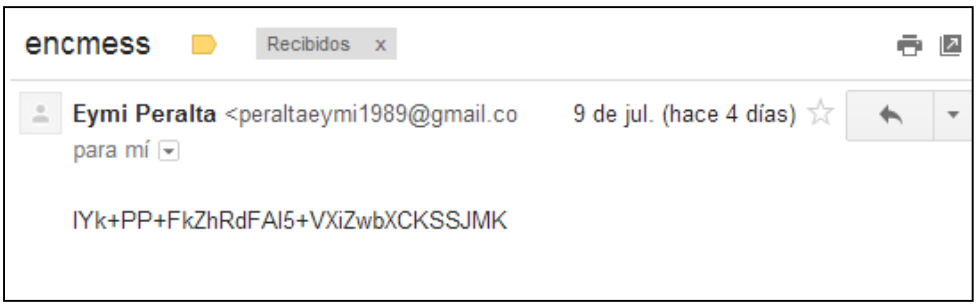


Figura 121: Recibiendo correo

9.1.4. Crypto ms

Crypto ms System es un cliente móvil para enviar sms cifrados. Se establece una contraseña y se compone el mensaje y éste se encripta antes de enviarlos. Es posible establecer una contraseña diferente para cada mensaje privado.

Mensajería segura se utiliza en muchas áreas de negocio con el intercambio de datos y sensible de toda la compañía. Las instituciones financieras, compañías de seguros, los servicios públicos, organizaciones de salud y proveedores de servicios se basan en la protección de la mensajería segura.

Por ejemplo si queremos mandar un mensaje seria como el siguiente ejemplo: Nos vamos al menú, ponemos la clave para desencriptar y podemos ver el mensaje recibido así:

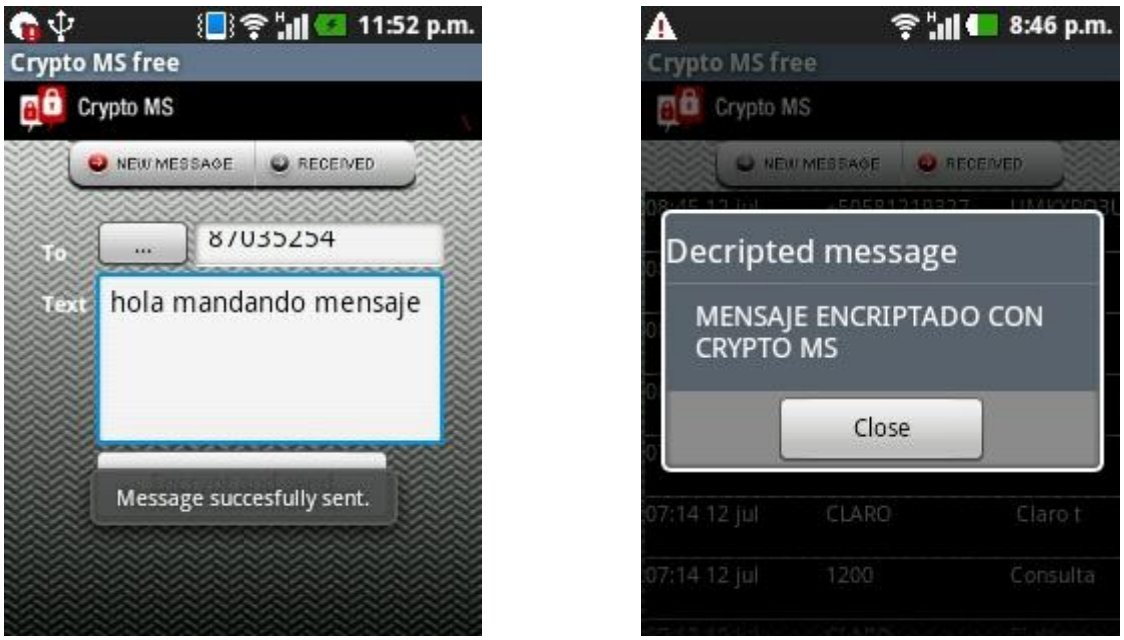


Figura 123: Mandando mensaje con Crypto MS

9.1.5. Orbot

Orbot es una aplicación proxy gratuito que permite a otras aplicaciones que utilizan Internet de forma más segura. Orbot usa Tor para cifrar el tráfico de Internet, es un software libre y una red abierta que le ayuda a defenderse contra una forma de vigilancia que amenaza la libertad personal y la privacidad, confidencialidad en los negocios y las relaciones, y la seguridad del estado conocido como análisis de tráfico.

No acepte imitaciones: Orbot es la forma más segura de utilizar Internet en Android. Orbot rebota el tráfico cifrado en varias ocasiones a través de los ordenadores de todo el mundo, en lugar de que le conecta directamente como VPN y servidores proxy. Este proceso tarda un poco más, pero la privacidad más fuertes y protección de identidad disponible es la pena la espera.

Navegación web privada: Uso con Orweb, la forma más anónima para acceder a cualquier sitio web, incluso si está normalmente bloqueado, supervisado, o en lo oculto Web. Obtenga Orweb: https://goo.gl/s3mLa

Privacidad para apps: Cualquier aplicación instalada puede usar Tor si se dispone de una función de proxy, utilizando los ajustes encontrados aquí: https://goo.gl/2OA1y Uso Orbot con Twitter, o tratar web privada que busca con DuckDuckGo: https://goo.gl/lgh1p.

Privacidad para todos: Tor puede ayudarte a investigar de forma confidencial a un competidor, obtener alrededor de la cuadra Facebook en la escuela, o sortear un cortafuego para ver los deportes en el trabajo.

Es oficial: Esta es la versión oficial del servicio de enrutamiento cebolla Tor para Android.

Instalación y uso

Primeramente descargamos la aplicación en nuestro celular, luego lo ejecutamos e iniciamos el servicio:

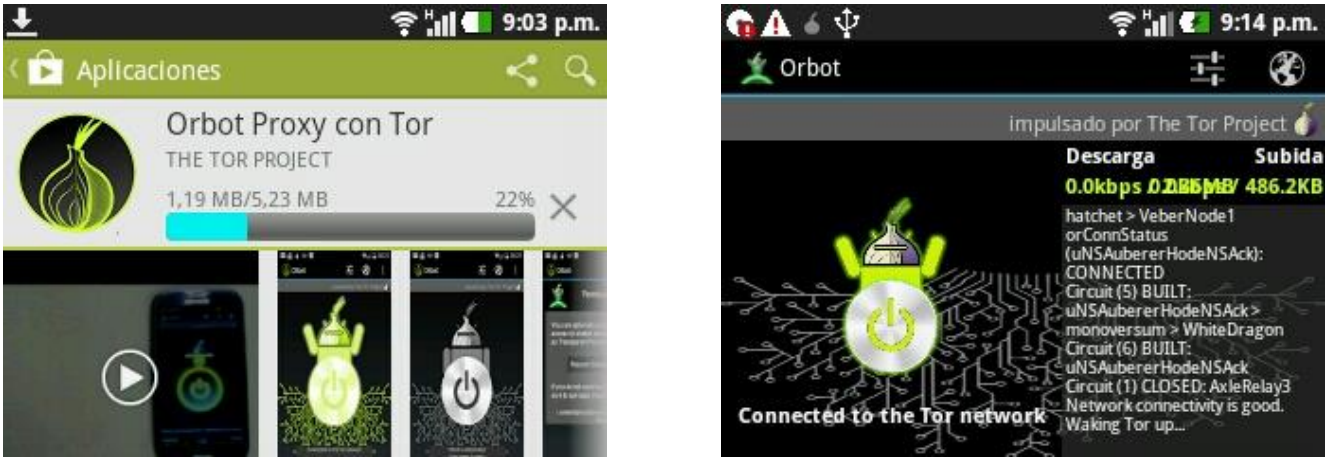


Figura 124: Instalando Orbot

Ahora una vez que ya lo ejecutamos vamos a navegar de forma anónima para comprobar que estamos navegando en la red Tor, pero para ello se descarga el navegador llamado Orweb, y navegamos en facebook:

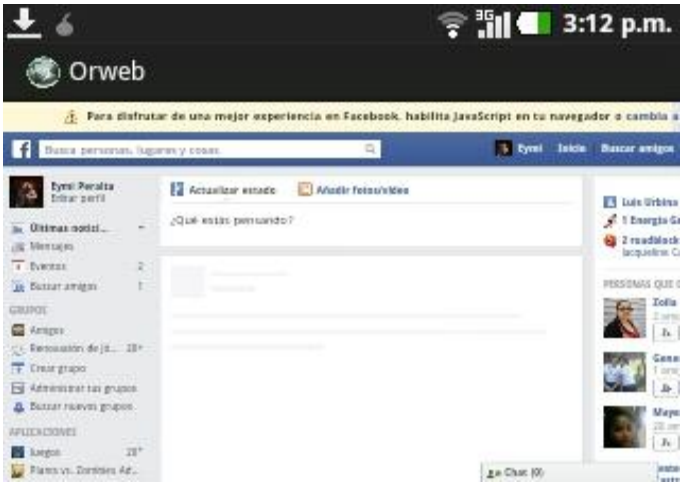


Figura 125: Navegando anónimamente

9.1.6. Otr

OTR significa Off-the-Record. Esto no sólo proporciona cifrado de alta seguridad, sino también una negación. Usando OtrSMS, los usuarios pueden comunicarse con sus amigos mediante el envío de sms al igual que el chat en un ambiente privado. Nadie será capaz de controlar el contenido conversación.

Las operaciones son bastante similares a las aplicaciones normales de sms, por lo que los usuarios encontrarán que es muy práctico. La principal diferencia es que los usuarios tienen que introducir una contraseña para un nuevo contacto OTR. El destinatario debe introducir el mismo. Esta contraseña sólo se utiliza para facilitar el proceso de establecimiento de claves.

La longitud del texto cifrado será promedio 1.3 a 1.4 veces la longitud del texto inicial.

Nota: Todos los SMS enviados antes de la respuesta del nuevo contacto OTR se envían en claro, sin cifrado. Aunque la interfaz de usuario es sólo en inglés por el momento, se puede enviar SMS en cualquier idioma.

Ahora componemos el mensaje a enviar:

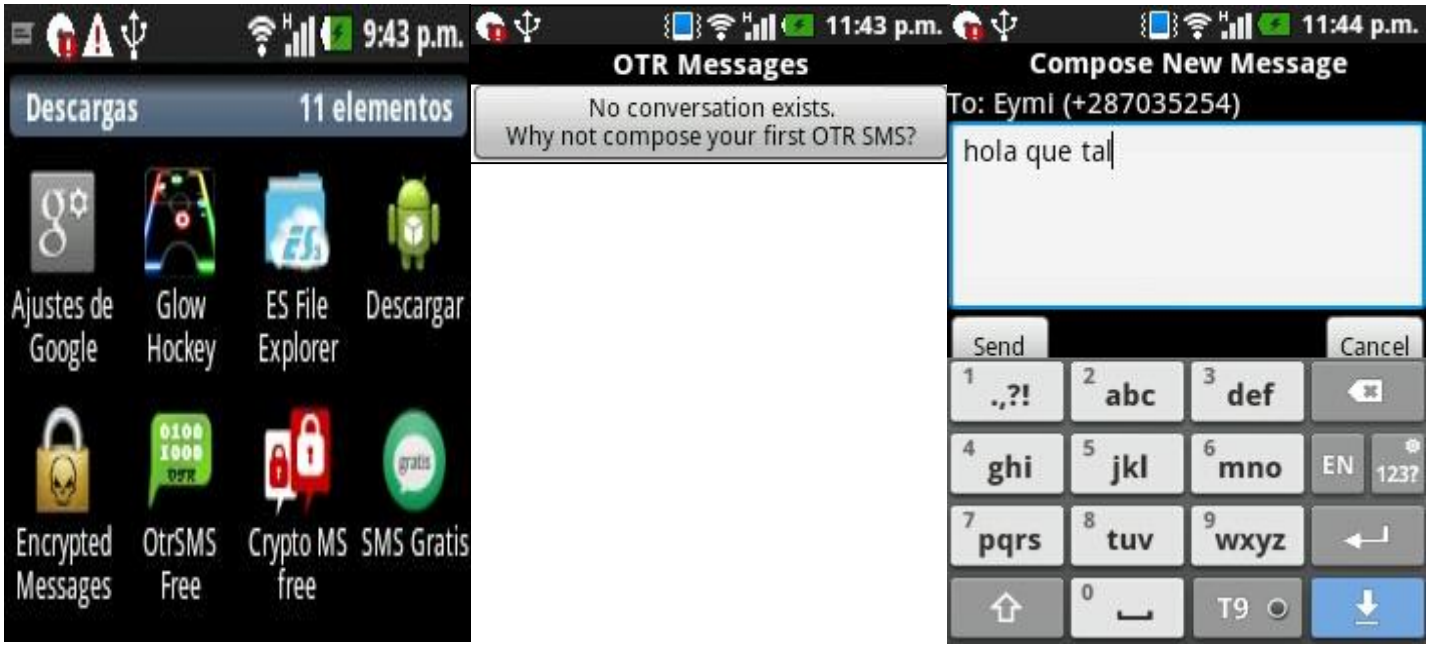


Figura 126: Aplicación Otr Nos pedirá que iniciemos la conversación con un contacto

Nos pedirá una contraseña con la que establecemos las claves:

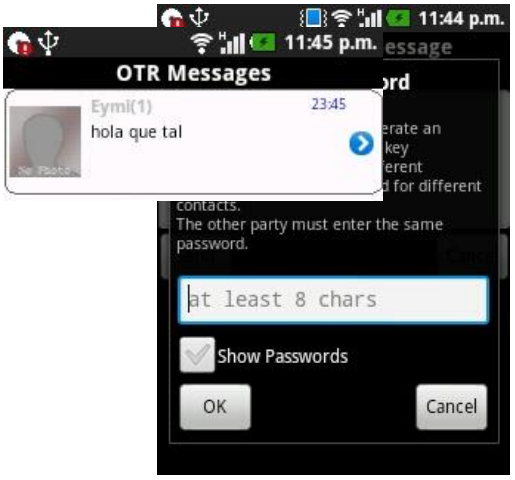


Figura 127: Probando OTR

9.1.7. Phone Tracker

Ofrece SMS gratis, llamadas, GPS y seguimiento de actividad Web de teléfono.

Funcionamiento

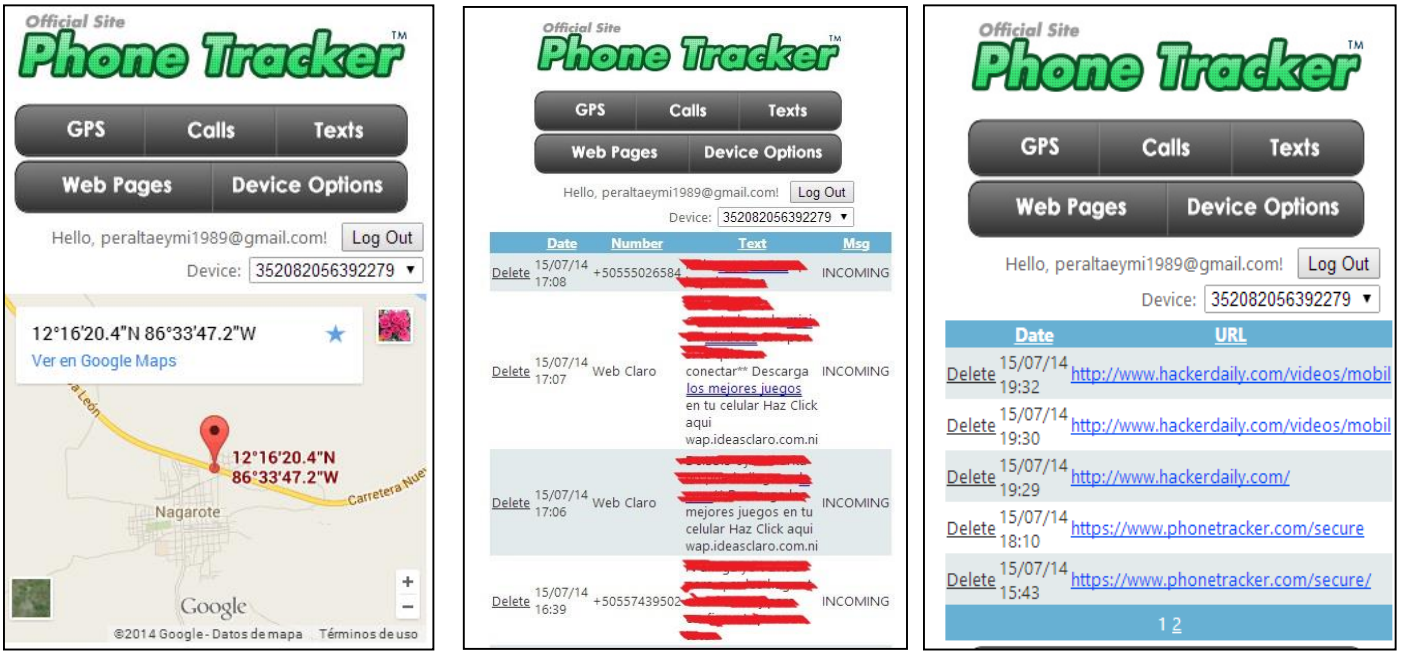


Figura 128: Monitorizando con SpyPhone

Una vez instalado el software, que toma aproximadamente 30 segundos, puede supervisar los siguientes datos:

- Los datos de localización GPS en un mapa trazado (actualizado cada 30 minutos);
- Número de teléfono, fecha, hora y duración de la llamada para todas las llamadas telefónicas salientes; Número de teléfono, fecha, hora y duración de la llamada para todas las llamadas telefónicas entrantes, a menos que los datos se bloquean por la persona que llama;
- El contenido, número de teléfono, la fecha y la hora de todos los mensajes de texto salientes;
- Una lista de sitios web visitados, incluyendo la dirección del sitio web, y la fecha y duración de la visita. Toda esta información está disponible 24 horas al día, 7 días a la semana, a partir de una cuenta que se puede establecer en nuestro sitio web seguro.

9.1.8. Crypto

Crypto es una aplicación que proporciona a cifrar los archivos y los esconden en el sistema. Un variado conjunto de algoritmos se ha construido para proteger mejor sus datos, usted puede elegir entre simplemente ocultar sus archivos a una encriptación completa del DES.Crypto también puede cifrar todos los archivos de una carpeta o una selección de ellos.

Algoritmos soportados:

- Hide (Extensión Modificar)
- OTP (one-time pad)
- RC4 (Rivest Cipher 4)
- TEA (Tiny Encryption Algorithm)
- DES (Data Encryption Standard)

Al abrir la aplicación lo primero que nos solicita es una contraseña que será la clave para poder cifrar y descifrar los archivos.

Seleccionamos el archivo que se quiere cifrar.

Y podemos escoger el método de cifrado.

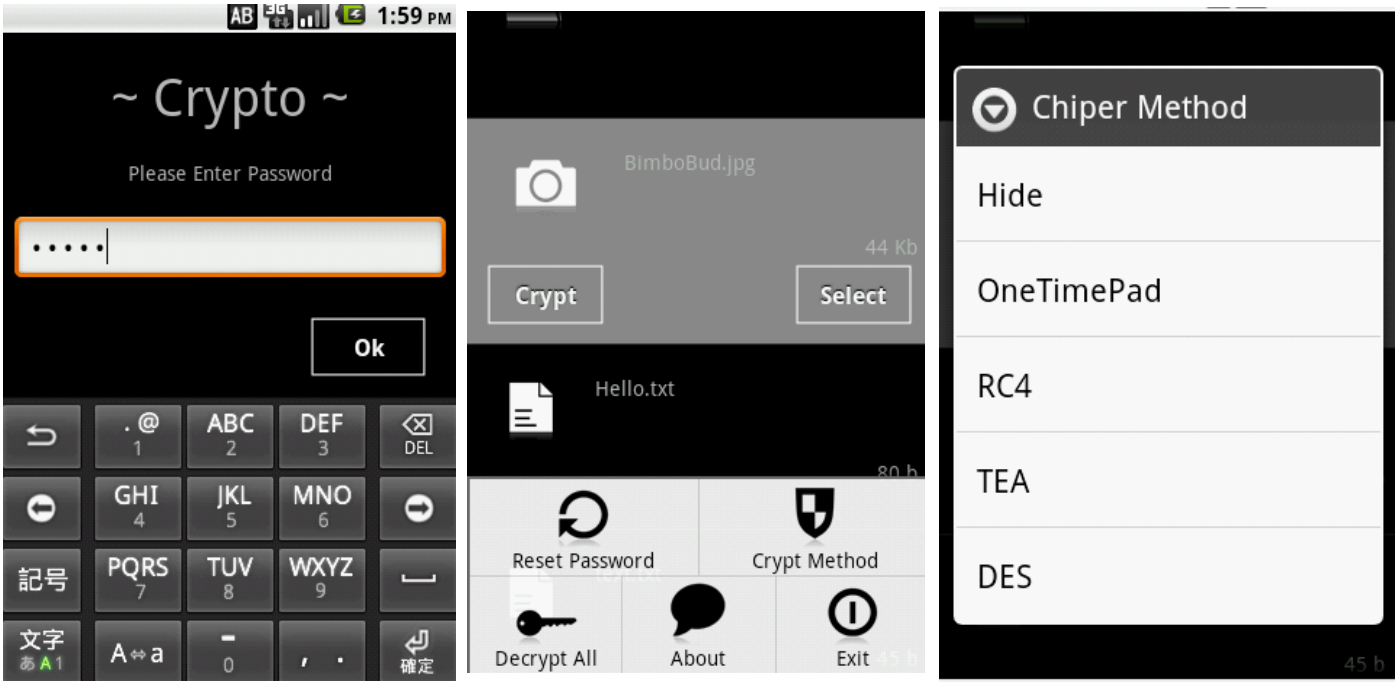


Figura 129: Probando Cryto

Y ahora podemos ver que el archivo está protegido:

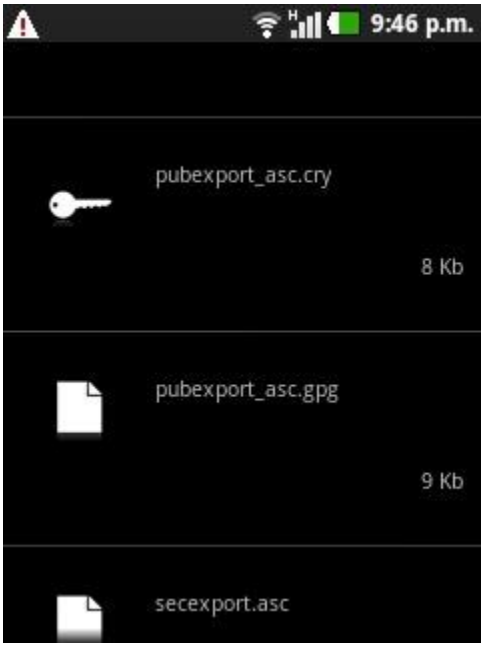


Figura 130: Protegiendo archivo con Cryto

9.1.9. Apg

OpenPGP para Android. Es de código abierto y su objetivo es proporcionar una implementación de OpenPGP similar a GnuPG.

Un tenedor de APG llama Open Key chain añadió la mayor parte de las nuevas características, que ahora se han incluido de nuevo en APG.

Cambios:

- Integración Fix K-9 que fue roto en 1.1.0
- Combinar OpenKeychain desarrollo nuevo en APG
- Nuevo diseño de cajón (tocar icono superior izquierdo)
- El KeyRing sin llave maestra privada Import
- QR compartir tecla
- NFC para compartir tecla
- Correcciones de errores
- Gestionar claves públicas / secretas (por ejemplo GPG / PGP)
- Cifrar / firmar / descifrar / verificar los correos electrónicos y archivos
- Integración con K9

- Soporte de servidor de claves HKP
- Creación de claves básica y edición posible
- API Intención para otras aplicaciones

Primero creamos las claves, la privada y la pública, para ello nos solicitan un correo y generamos las llaves:

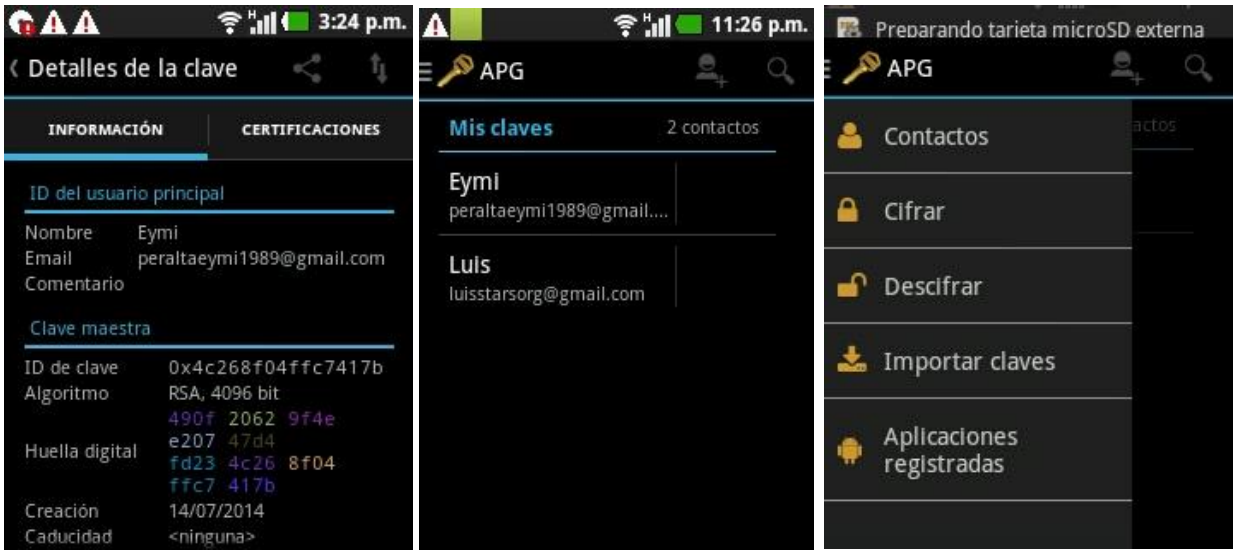


Figura 131: Generando claves con AGP

Ahora si nos vamos al menú podemos ver la opción de cifrar y descifrar, seleccionamos cifrar. Y mandamos el correo con un mensaje cifrado a un destinatario firmado con la llave pública de quien lo manda:

Figura 132: Mandando correo con AGP

Y ahora podemos comprobarlo en el correo, si queremos descifrarlo utilizamos la privada:

Figura 133: Recibiendo correo

Figura 134: Desencryptando correo

X. CONCLUSIONES

A través, del presente trabajo se ha analizado y estudiado las debilidades de las redes sociales, para ello se ha hecho un estudio del funcionamiento en la comunicación de las redes sociales y móviles (más conocido mensajería de textos) en base a lo antes mencionado se ha llegado a las siguientes conclusiones:

- Dado el gran incremento en el uso de las redes sociales con distintos fines y los casos actuales de robo de información personal, el tema de la seguridad no debe dejarse a un lado; más bien, es aconsejable dotar de la mayor seguridad posible a los usuarios finales.
- Mediante una evaluación de las herramientas actuales más usadas, se ha logrado determinar cuáles de esas herramientas poseen las mejores características de protección y seguridad para nuestras comunicaciones.
- A través de las herramientas usadas se ha logrado obtener el nivel de anonimato deseado en el momento de navegar y no ser víctima de un atacante.
- Al elaborar un documento con información guiada y detallada de la instalación y configuración de las herramientas mencionadas, se hace conciencia en estudiantes, profesionales, etc, de la necesidad de la protección de las comunicaciones.

XI. RECOMENDACIONES

Una vez finalizado el presente trabajo se proponen las siguientes recomendaciones para la continuación y mejora en cuanto a la seguridad en redes sociales y dispositivos móviles:

- Darle seguimiento a la seguridad en redes sociales y móviles en cuanto a las actualizaciones de las nuevas herramientas que surjan a lo largo del tiempo.

- Diseñar una aplicación que permita el cifrado de nuestros mensajes a través de los dispositivos móviles en el cual tenga una longitud u capacidad mayor de 140 caracteres.

XII. BIBLIOGRAFÍA

[1] Seis grados de separación. (2010, 30) de marzo. Wikipedia, La enciclopedia libre. Fecha de consulta: 03:10, marzo 1, 2014 from *http://es.wikipedia.org/wiki/Seis_grados_de_separaci%C3%B3n*

[2]<<http://www.wikisaber.es/uploadedFiles/ComunidadWiki/Blogs/Blog_wikisaber_Padres/Redes%20sociales.pdf>> [consulta: 03-03-2014].

[3] <<Redes Sociales>>, Maestros de la Web. [En línea]. Disponible en *http://www.maestrosdelweb.com/editorial/redessociales/*. [Accedido: 03-mar-2014].

[4] <<http://www.lalfasjove.com/data/documentos/Guia_sobre_Redes_Sociales.pdf>>[Consulta: 04-03-2014].

[5] Carrasco Ortega, D., (2011,17) de Noviembre. *Privacidad y seguridad en redes sociales*, *http://recursostic.educacion.es/observatorio/web/es/internet/recursos-online/1015-daniel-ortega-carrasco/*. [Accedido: 04-mar-2014].

[6] <<http://fido.palermo.edu/servicios_dyc/blog/docentes/trabajos/12197_38536.pdf>> [consulta: 05-03-2014].

[7] Suiras E, <<Redes sociales: Definición>>, scribd. [En línea]. Disponible en

http://es.scribd.com/doc/24658747/Redes-sociales-definicion/. [Accedido: 11-marz-2014].

[8] LinkedIn. (2014, 24) de enero. Wikipedia, La enciclopedia libre. Fecha de consulta: 02:00, marzo 3, 2014 from *http://es.wikipedia.org/wiki/LinkedIn*

[9] Pinteares. (2014, 18) de marzo. Wikipedia, La enciclopedia libre. Fecha de consulta: 00:22, marzo 20, 2014 from http://es.wikipedia.org/wiki/Pinteres.t

[10] Google+. (2014,24) de febrero. Wikipedia, La enciclopedia libre. Fecha de consulta:

02:00, marzo 10, 2014 from http://es.wikipedia.org/wiki/Google%2B.

[11] LiveJournal. (2013, 21) de Julio. Wikipedia, La enciclopedia libre. Fecha de consulta:

11:29, marzo 10, 2014 from http://es.wikipedia.org/wiki/LiveJournal.

[12] Tagged. (2014, 20) de febrero. Wikipedia, La enciclopedia libre. Fecha de consulta: 17:04, de marzo, 2014 from *http://es.wikipedia.org/wiki/Tagged.*

[13] Orkut. (2014, 06) de marzo. Wikipedia, La enciclopedia libre. Fecha de consulta: 02:29, de marzo, 2014 from *http://es.wikipedia.org/wiki/Orkut*.

[14] Ning. (2013, 27) de diciembre. Wikipedia, La enciclopedia libre. Fecha de consulta: 20:42, de marzo, 2014 from *http://es.wikipedia.org/wiki/Ning*

[15] <<Análisis de Redes Sociales Conceptos Fundamentales>>, wikibooks. [En línea]. Disponible en *http://es.wikibooks.orgwiki/An%C3%A1lisis_de_Redes_Sociales/Conceptos_Fundamentales/. [Accedido: 17-dic-2013].*

[16] <<Curso de privacidad y protección de comunicaciones digitales>>, cryt4you.Aula Virtual, Universidad Politecnice de Madrid. [En línea]. Disponible en *http://www.crypt4you.com/.* [Accedido: 07-02-2014]

[17]<<Más de la mitad de los españoles no sabe los riesgos que afronta al dar sus datos>>, 20 minutos. [En línea]. Disponible en *http://www.20minutos.es/noticia/1835021/0/cis/encuesta/internet-nuevas-tecnologias/* [Accedido: 08-02-2014]

[18] <<Red de bots Mariposa>>, pandalabs. [En línea]. Disponible en

http://pandalabs.pandasecurity.com/es/red-de-bots-mariposa/ [Accedido: 10-03-2014].

[19]<<El primer ministro turco Erdogan carga contra Twitter>>,post digital. [En línea]. Disponible en *http://postdigital.es/2013/06/03/erdogan-carga-contra-twitter/#sthash.olEdxBdX.dpbs/*[Accedido: 20-02-2014].

[20] <<Cispa apoyo list>>, digitaltrends. [En línea]. Disponible en *http://www.digitaltrends.com/web/cispa-supporters-list-800-companies-that-could-help-uncle-sam-snag-your-data/#!Ad4cP/*[Accedido: 13-02-2014].

[21]<<*Extendiendo las posibilidades de las redes*>>*,4rsoluciones. [En línea].Disponible en http://www.4rsoluciones.com/apis-extendiendo-las-posibilidades-de-las-redes-sociales/*[Accedido: 12-03-2014*]*

[22] Minería de datos. (2014, 17) de marzo. Wikipedia, La enciclopedia libre. Fecha de consulta:

02:00. Marzo 10, 2014 from *http://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos.*

[23] <<*http://arantxa.ii.uam.es/~jms/pfcsteleco/anteproyectos/Gomez_Aparicio_Luis%20Miguel_AntePFC.pdf*>>[Consulta: 21-02-2014].

[24] <<Como funciona la red Tor>>, genbeta. [En línea]. Disponible en

http://www.genbeta.com/seguridad/como-funciona-la-red-tor/[Accedido: 12-03-2014].

[25] <<Artículos útiles>>, centro de artigo. [En línea]. Disponible en *http://centrodeartigos.com/articulos-utiles/article_110600.html/ [*Accedido: 02-mar-2014].

[26] <<Herramientas de Tor>>, cuidatuinfo. [En línea]. Disponible en

https://www.cuidatuinfo.org/herramientas/tor/ [Accedido: 02-marzo-2014].

[27]<< Enrutamiento cebolla, Capacidades, Cebollas, Aplicaciones>>, centro de artigo. [En línea]. Disponible en *http://centrodeartigos.com/articulos-utiles/article_110600.html/ [*Accedido: 12-mar-2014].

[28]<<Redes de anonimato>>, redes cebolla. [En línea]. Disponible en

http://redescebolla.files.wordpress.com/2010/02/redes-de-anonimato-estudio.pdf/[Accedido: 11-mar-2014].

[30] <<Tor parteII>>, pillateunlinux. [En línea]. Disponible en *http://www.pillateunlinux.com/tor-parte-ii/ [Accedido 10-mar-2014].:*

[31] <<Tor parteIV>>, pillateunlinux. [En línea]. Disponible en *http://www.pillateunlinux.com/tor-parte-iv/ [Accedido: 10-mar-2014].*

[32] <<Tor>>. (2014,27) de febrero. Wikipedia, La enciclopedia libre. Fecha de consulta: 10:00. marzo 10, 2014 from *http://es.wikipedia.org/wiki/Tor*

[33]<< Extensible Messaging and Presence protocol>>. (2013,17) de sep. Wikipedia, La enciclopedia libre. Fecha de consulta: 03:52. Marzo 2, 2014 from Wikipedia, La enciclopedia libre. Fecha de consulta: *http://es.wikipedia.org/wiki/Extensible_Messaging_and_Presence_Protocol*

[34] Saint-Andre, P. , Mith, K. ,Remko T. , (2009) "XMPP: The Definitive Guide", Gravenstein Highway North, Sebastopol: O'Reilly Media. [En línea] Disponible en

http://books.google.es/books?id=ChTCQYLIDfoC&printsec=frontcover&dq=xmpp&hl=es&sa=X&ei=QSQjU8X8lpPqkQed9oHIDw&ved=0CD0Q6AEwAQ#v=onepage&q=xmpp&f=false

[35] <<Pidgin>>. (2014,22) de febrero. Wikipedia, La enciclopedia libre. Fecha de consulta*: 10:10, marzo 12, 2014 from http://es.wikipedia.org/wiki/Pidgin*

[36]<<Pidgin, el cliente de chat universal>> (2010,13) de abril. Blogspot. [En línea] Disponible en:

http://alexabreo.blogspot.com/2010/04/pidgin-el-cliente-de-chat-universal.html/ Accedido[10-mar-2014].

[37] <<Pidgin (software)>>. (2014, 22) de febrero. Wikipedia, La enciclopedia libre. Fecha de consulta*: 09:00, marzo 01. 2014 from http://es.wikipedia.org/wiki/Pidgin_(software)*

[38] <<Off the record messaging>> (2014, 8) de febrero. Wikipedia, La enciclopedia libre. Fecha de consulta*: 02:00, febrero 28. 2014 from http://es.wikipedia.org/wiki/Off_the_record_messaging*

[39] <<Anontwi: Cifrado y censura en redes sociales>> (2013,3) de enero. Securitybydefault. [En línea] Disponible en: *http://www.securitybydefault.com/2013/01/anontwi-cifrado-y-censura-en-redes.html/ [Accedido: 2-marzo-2014].*

[40] <<AnonTwi: cliente de Twitter que permite cifrar tweets y mensajería privada>>. (2012.04) de junio. Kriptopolis. [En línea]. Disponible en: *http://www.kriptopolis.com/anontwi/ [Accedido: 11-03-2014].*

[41] <<Transport Layer Security>>, (2014, 15) de marzo. Wikipedia, La enciclopedia libre. Fecha de consulta*: 03:23, marzo 12 from http://es.wikipedia.org/wiki/Transport_Layer_Security.*

[42] Stallings, W., (2004) "*Fundamentos de seguridad en redes. Aplicaciones y Estándares*", segunda edición, Madrid, España: Pearson Educación, S.A.

[43]<< Autenticación Seguridad en Internet>> Universidad de Alcalá [Accedido: 01, de marzo 2014].

[44]<< Diffie-Hellman>>, (2014, 26) de enero. Wikipedia, La enciclopedia libre. Fecha de consulta*: 00:02, marzo 10 fromhttp://es.wikipedia.org/wiki/Diffie-Hellman.*

[45] <<FireGPG: Privacidad para el webmail>>. (2010, 26) Web azul, blogs. [En línea] Disponible en: *https://m.iguel.net/Blog/?entrada=32/ [Accedido: 11-marzo-2014].*

[46] <<Cryptocat>>, (2013,8) de dic. . Wikipedia, La enciclopedia libre. Fecha de consulta*: 10:00, marzo 01 from http://es.wikipedia.org/wiki/Cryptocat.*

[47] <<Privly>>, priv.ly, [En línea] Disponible en: *https://priv.ly/pages/about/ [Accedido: 12-mar-2014].*

[48] <<Facecat>>, (2012. 02) de abril. Pentester. [En línea] Disponible en:

http://tools.pentester.es/facecat/ [Accedido: 02-mar-2014].

[49] <<Netcat>>. (2014, 13) de febrero . Wikipedia, La enciclopedia libre. Fecha de consulta*: 02:00, marzo 02, 2014 from https://en.wikipedia.org/wiki/Netcat.*

[50] <<Sobre anonimato, redes cebolla y demás>> (2011, 24) de enero. Redes Cebolla. [En línea] disponible en: http://redescebolla.wordpress.com/ [Accedido: 12-mar-2014].

[51]<<Red privada virtual>>, (2014, 6) de marzo. Wikipedia enciclopedia libre. Accedido: 09:12, 18 de marzo, 2014. Disponible en *http://es.wikipedia.org/wiki/Red_privada_virtual*.

[52] Laporta L. J., Aguiñiga M. M., (2005) "*Fundamentos de Telemática: Red privada virtual",* Valencia, España: Editorial de la UPV, pp223. [En línea] Disponible en:

http://books.google.com.ni/books?id=zq1r6ed9NIYC&pg=PA223&dq=red+privada+virtual&hl=es&sa=X&ei=wQwpU4aeLsGmkQf7uoDICg&ved=0CDMQ6AEwAA#v=onepage&q=red%20privada%20virtual&f=false.

[53] <<*El Proyecto de Internet Invisible (I2P*)>>, [En línea] Disponible en *https://geti2p.net/es/about/intro* [Accedido: 21-Enero-2014].

[54] <<*Garlic Routing y terminología "Garlic"*>>*,* [En línea] Disponible *en https://geti2p.net/es/docs/how/garlic-routing* [Accedido: 01-Enero-2014].

[55] << *El proyecto de internet invisible*>>, [En línea] Disponible en *https://geti2p.net/es/* [Accedido: 24-Febrero-2014].

[56] <<*I2P*>>, (2014, 09) de mayo. [En línea] Wikipedia, La enciclopedia libre. Fecha de consulta: 15:11, mayo 09 from *http://es.wikipedia.org/wiki/I2P.*

[57] <<*GNU Privacy Guard*>>, (2014, 09) de agosto. Wikipedia, La enciclopedia libre. Fecha de consulta: 20:14, agosto 09 from *http://es.wikipedia.org/wiki/GNU_Privacy_Guard*.

[58] << *GnuPG*>>, [En línea] Disponible en *https://www.gnupg.org/* [Accedido: 10-mayo-1998].

[59] Alonso J. José M., Profesor, autor y desarrollador, <<*GnuPG*>> (2010)<<*http://nibbler.es/cms/documentos/articulos/pdf/criptografia-asimetrica-con-gnupg*>> [consulta: 10-08-2010].

[60] Stemper, Robert (2009), <<*Configure error libgcrypt and libgpg-error*>> [En línea] Disponible en *http://lists.gnupg.org/pipermail/gnupg-users/2009-May/036486.html* [Accedido: 15-Mayo-2009].

XIII. ANEXOS

13.1. TABLA DE SIMILITUDES ANONIMATO TOR & I2P

Tor	I2P
Celda	Mensaje
Cliente	Ruter o Cliente
Circuito	Túnel
Directorio	NetDb, base de datos de la red
Servidor de directorios	Ruter Floodfill o de relleno
Control de entradas	Pares rápidos
Nodo de entrada	Inproxy, proxy de entrada
Nodo de salida	Outproxy, o proxy de salida
Servicio oculto	Eepsite o destinación
Descriptor de servicio oculto	LeaseSet
Punto de introducción	Puerta de salida de entrada
Nodo	Ruter
Proxy Onion	Cliente de I2PTunnel (más o menos)
Relay	Ruter
Punto de reunión	algo así como inbound Gateway + Outbound Endpoint, puerta de salida de entrada + punto final de salida
Descriptor del ruter	Información del ruter
Servidor	Ruter

Tabla 11: Similitudes de TOR & I2P

13.2 . TABLA DE BENEFICIOS DE TOR & I2P

Beneficios de Tor sobre I2p	**1. El número de usuarios es mucho mayor; mucho más visible en entorno académicos y comunidades de hackers** **2. Más eficiente con el uso de la memoria** **3. Los nodos cliente de Tor tienen muy poco gasto de ancho de nada** **4. El control centralizado reduce la complejidad de cada nodo y puede evitar ataques Sybil fácilmente** **5. Menor latencia C, no java.**
Beneficios de I2P sobre Tor	**1. Diseñado para optimizar los servicios ocultos, que son**

	mucho más rápidos que los de Tor **2. Totalmente distribuido y auto organizado** **3. Los pares son seleccionados continuamente según el rendimiento y categoría, en vez de confiar en la capacidad indicada** **4. Los pares Floodfill ("servidores de directorios") cambian, en vez de estar siempre fijos.** **5. Tan pequeña que no ha sido bloqueada o Dosed.** **6. Amigable con las aplicaciones p2p** **7. Conmutado por paquete en vez de conmutado por circuito** **8. Resistencia contra fallos ya que usa varios túneles en paralelo, además de rotar los túneles**

Tabla 12: Beneficios de usar TOR & I2P

13.3. TABLA DE HERRAMIENTAS DE ANONIMATO

	Sistema Operativo	Encaminamiento	Cifrado
Tor	multiplataforma	Cebolla	RSA, AES 128, 3DES
I2P	multiplataforma	Ajo	Simétrico y asimétrico

Tabla 13: Característica de herramientas de Anonimato

13.4. HERRAMIENTAS QUE PERMITEN EL CIFRADO EN REDES SOCIALES

Nombre	Navegador Windows y Linux	Vigencia	Algoritmo usado
Pidgin	Multiplataforma	si	RSA
GPG	Multiplataforma	si	Cifrado Asimétrico

Tabla 14 detalle de las herramientas de cifrado

13.5. EXTENCIONES QUE PERMITEN EL CIFRADO EN REDES SOCIALES

	Extensión	Navegador Windows y Linux	Vigencia	Algoritmo usado
Extensiones para lograr cifrado atreves de la red	FireGPG	Firefox 3.6.23	no	Cifrado simétrico 3DES, AES192, AES256, BLOWFISH
	Cryptocat	Firefox y Chrome	si	OTR
	Secure mail for Gmail	Chrome	Si, en proceso la implementación de firma	Cifrado simétrico de AES256
	Mailvelope	Chrome	Si	RSA clave Asimétrica

Tabla 15: Detalles de extensiones que permiten cifrado

13.6. APLICACIONES IMPLEMENTADAS PARA CIFRADO EN MÓVILES

Aplicación	Versión	Algoritmo	Clave	Longitud
Babel	3.1.1	AES, Diffie-Hellman	Simétrica	128 bits
Crypto MS	1.6.2	AES	Simétrica	128- 192 bits
Encrypted Messages	1	AES	Simétrica	2048 bits
Agp	1.1.0	AES, Hash,	Asimétrica	256 bits, 512 bits
Crypto	1.1	Hide, OTP,RC4,TEA,DES	Simétrica	256 bits

Tabla 16: Detalles del cifrado en móviles