Seguridad en servicios de mensajería instantánea

[!] Actualización: la anterior versión de esta guía se fue quedando obsoleta y vimos necesario actualizarla, dado que la mensajería instantánea es muy utilizada en el día a día y en las actividades relacionadas con la militancia. Los cambios realizados pueden resumirse de esta forma:

- Dejamos de llamarlos sistemas de mensajería seguros y los llamamos "recomendados", ya que, aunque pensamos que son seguros (los más seguros que podríamos recomendar), preferimos prevenirnos, como demuestra el hecho de haber reelaborado esta guía.
- Recomendamos especialmente *Signal* y *SimpleX Chat*.
- Dejamos de recomendar *Matrix*, *Session* y *Briar*.
- Información útil añadida en diversos apartados.
- Frases reformuladas, correcciones ortográficas, etc.

Esperamos que esta nueva versión sea más acorde a los requisitos de seguridad de la militancia revolucionaria.

Resumen (TL;DR)

- **NO utilizar:** servicios de mensajería comerciales como **Whatsapp**, **Telegram** o los **mensajes directos en redes sociales** (Twitter/X, Instagram, etc.).
- **Utilizar:**
 - **Signal:**
 - **Descargar signal.**
 - En Android, mejor aún si se utiliza la aplicación **Molly** en su versión FOSS, en vez de la app oficial de Signal. **Descargar Molly** (Elegir la versión FOSS).
 - Priorizar la utilización de **nombres de usuario** y **evitar compartir el número de teléfono.**
 - **SimpleX Chat:**
 - **Descargar SimpleX Chat.**
 - Se pueden autoalojar servidores propios SMP (transferencia de mensajes) y XFTP (transferencia de archivos). Más información en **la sección dedicada a SimpleX Chat.**

Introducción

Unas de las herramientas más utilizadas por organizaciones para realizar su labor política son los servicios de mensajería instantánea. A diferencia de otros tipos de mensajería cuyas vías recorren el ciberespacio, la mensajería instantánea permite comunicarse de forma rápida, ágil y segura... o al menos esto último se cumpliría si supiéramos elegir bien estas herramientas.

Probablemente nada haya sido más útil para las fuerzas del orden a la hora de detener a militantes y desbaratar manifestaciones u otro tipo de actividades que el mal uso de los servicios de mensajería instantánea o el uso de servicios inseguros para comunicarse y organizar dichas actividades.

En el artículo de hoy, por tanto, pretendemos dar una respuesta a esta problemática que por ser atajada de manera insuficiente le ha costado muy cara a los movimientos revolucionarios en los últimos años.

Para ello, hemos organizado esta guía de tal forma que sea didáctica, no requiera conocimientos avanzados para ser comprendida y transmita de manera clara cuáles son las ventajas de cada servicio de mensajería y en qué contexto es mejor utilizar cada uno de ellos. Empezaremos con la explicación de unos conceptos fundamentales para comprender el funcionamiento de la mensajería instantánea y sus tipologías. Después, haremos un repaso por las mejores alternativas que hemos considerado, explicando las ventajas y desventajas de cada una, así como sus características y los ámbitos y casos de uso a los que mejor se ajustan. También subrayaremos los límites inherentes a los medios de mensajería instantánea, ya que no son medios infalibles, por lo que hay que usarlos con responsabilidad a la hora de compartir contenido sensible a través de ellos.

Topologías

Cliente-Servidor

El diseño Cliente-Servidor es el método más sencillo y usado en la comunicación entre dispositivos. Por un lado, existe un servidor, que no es otra cosa que una computadora preparada específicamente para gestionar peticiones de otras máquinas y darles respuesta. Por otro lado, existen los clientes, que son máquinas (habitualmente computadoras personales, smartphones, etc.) que se dirigen al servidor para hacer peticiones o solicitar servicios.

Pongamos un ejemplo. Cuando se usan Whatsapp o Telegram, se escriben mensajes en el

smartphone y cuando se pulsa el botón de enviar, el smartphone traslada el mensaje al servidor de Whatsapp o Telegram y este envía dicho mensaje al usuario de destino. Así, el servidor actúa como intermediario entre los usuarios finales.

Otra característica muy a tener en cuenta sobre esta topología y que nos parece fundamental para comprender algunos servicios de mensajería de los que hablaremos más abajo, es que cliente y servidor pueden ser totalmente independientes. Por ejemplo, Whatsapp es una aplicación monolítica, fuertemente dependiente de su servidor: para usar el servidor de Whatsapp solo puedes usar la aplicación oficial de Whatsapp. En el lado contrario, estaría, por ejemplo, XMPP, donde el servidor y la aplicación son totalmente independientes. Esto significa que para chatear en un servidor XMPP puedes escoger entre varias aplicaciones creadas por la comunidad, del mismo modo que puedes elegir entre varios servidores gestionados por empresas, fundaciones, gobiernos o personas independientes; cada uno con sus propios intereses y objetivos.

Aunque la arquitectura cliente-servidor es centralizada, en algunos protocolos y sistemas existe la posibilidad de la **federación**.

Ventajas:

- Es el diseño más simple y fácil de desarrollar, desplegar y mantener.
- Al tener un servidor central al que dirigirse, los datos se transmiten más rápido sin tener que andar dando vueltas entre varios nodos.
- No es necesario que los clientes (usuarios) estén conectados al mismo tiempo. El servidor almacena los mensajes y los envía a los usuarios cuando están conectados a la red.

Desventajas:

- Centralización: Si cae el servidor, no hay manera de continuar con la comunicación entre los clientes. Esto implica que el servidor es un punto de fallo crítico, pues toda la comunicación no puede realizarse sin su concurso. Además, si el servidor es comprometido por un atacante, puede tener acceso a toda la información de los usuarios y a los mensajes (aunque estos pueden estar cifrados).
- Es necesario confiar en el servidor central, ya que, al pasar toda la información por él, se le está dando poder absoluto sobre la información.

Peer-to-Peer

El diseño Punto-a-Punto o Peer-to-Peer (P2P a partir de ahora) es un tipo de topología que, a diferencia de la topología Cliente-Servidor, no dispone de un servidor entre los clientes, realizándose la comunicación directamente entre ellos. Es cierto que a veces se

necesita de un servidor central para que gestione las conexiones entre los clientes, pero una vez establecida dicha conexión, el servidor ya no actúa más como intermediario entre la comunicación.

Ventajas:

- No se depende de un servidor central y, por tanto, no existe un único punto de fallo que pueda interrumpir la comunicación entre los nodos.
- Permite una mayor soberanía en la gestión de la información de cada cliente (o nodo), ya que no se depende de la gestión por un servidor central.

Desventajas:

- La comunicación es más compleja, problemática y requiere que ambos nodos estén conectados a la red al mismo tiempo. Esto suele resultar en un peor servicio a la hora de intercambiar mensajes, retrasando su entrega o perdiéndose mensajes durante el tránsito.

Malla

La topología en malla no es más que una topología P2P, pero en la que los nodos o clientes de la red pueden reenviar mensajes a otros nodos de la red. De nuevo, no existe un servidor central y toda la gestión del reenvío de mensajes recae sobre los clientes (nodos) de la red. Sus ventajas y desventajas son prácticamente las mismas que en una topología P2P.

Ventajas:

- Los nodos de la red se apoyan unos en otros para hacer llegar los mensajes a los demás (algo muy útil cuando la distancia es un obstáculo).
- En general, cualquiera puede unirse a una red y convertirse en un elemento más de ésta.

Desventajas:

- Aumenta la complejidad del servicio y, por tanto, aumentan los problemas en la robustez de la red.

Otras cuestiones técnicas

Autoalojar un servidor

Autoalojar o autohostear un servidor significa que cualquiera con los conocimientos

suficientes puede desplegar un servidor propio y gestionarlo a su manera. Esta es una capacidad muy positiva cuando se trata de preservar la soberanía sobre la información, ya que el servidor es propio y, por tanto, no hace falta confiar en un tercero. Por poner un par de ejemplos, un servidor **no autoalojable** sería el servidor de Whatsapp, ya que es de código cerrado y propiedad de Meta Platforms Inc. (Facebook), mientras que un servidor **autoalojable** sería cualquiera de código abierto, como **Prosody** (un software de servidor para el protocolo XMPP).

Para autoalojar un servidor, hay dos opciones. Por un lado, se puede utilizar una máquina propia, abriendo los correspondientes puertos del router o, por ejemplo, pasando el tráfico a través de Tor. Por otro lado, se puede contratar un VPS (Virtual Private Server) de algún proveedor (Amazon AWS, OVH, etc.) y desplegar ahí el servidor. Sin embargo, muchas veces contratar un VPS implica tener que proveer una serie de datos personales (cuenta bancaria, a veces documentación legal, nombres, etc.) que podrían ser reclamados por las autoridades llegado el caso y ser cedidos por el proveedor del VPS (Amazon, OVH o quien sea). En un VPS ajeno tampoco controlamos del todo que solo nosotros tengamos acceso a los datos. Ante esto, hay que saber bien cuándo conviene contratar un VPS y cuándo es mejor alojar el servicio sobre una infraestructura propia.

Puede encontrarse más información sobre autoalojar servicios de forma segura en [nuestra guía sobre aprovisionamiento seguro de servidores](#).

Cifrado punto a punto

El cifrado punto a punto (End-to-End-Encryption o E2EE) es un tipo de cifrado que, tal y como dice su nombre, cifra los mensajes a nivel de cliente o de punto final. Es decir, que el mensaje se cifra en la aplicación cliente y luego se envía al servidor central o al otro cliente, según la topología. Este tipo de cifrado es realmente importante cuando en el servicio de mensajería interviene un servidor central, porque no permite al servidor conocer el contenido del mensaje. Existen otros tipos de cifrado que sí permiten al servidor conocer el contenido del mensaje y esto, obviamente, es un problema. También queremos remarcar que el cifrado E2EE **no asegura una completa privacidad o anonimato**. Aunque esto depende de cada protocolo de comunicación, cifrar el contenido del mensaje no protege frente a otros datos que se pueden extraer como pueden ser: hora de envío del mensaje, localización GPS de una foto, identificador del usuario, identidad del servidor, etc.; éstos son los llamados **metadatos** y hay que tener en cuenta su existencia.

Seguridad de la información

Cuando se trata de mantener la seguridad de la información, hay tres dimensiones que son fundamentales. Aplicadas a los servicios de mensajería, corresponderían a:

- **Confidencialidad:** los mensajes solo pueden ser leídos por sus destinatarios legítimos. Para preservar esta dimensión, se utilizan técnicas de cifrado.
- **Integridad:** los mensajes no deberían poder ser alterados y deberían llegar al destinatario en las mismas condiciones en las que fueron enviados. Hay varias formas de preservar esta dimensión, pero las más usadas son las firmas digitales y los *hashes* (o resúmenes).
- **Disponibilidad:** es la dimensión que trata de que el servicio esté disponible y no se interrumpa la mayor parte posible del tiempo. Existen una infinidad de técnicas para ello y quedan bastante alejadas del campo de los servicios de mensajería, por lo que no serán tratadas aquí.

Centralización, descentralización y federación

La centralización, cuando se habla en lenguaje telemático, significa, a grandes rasgos, que solo existe un servidor (o grupo de servidores que actúan como uno solo) al que las aplicaciones cliente deben dirigirse. Un servicio descentralizado, por el contrario, es aquel en el que participa más de un servidor, a veces pudiendo federarse entre sí.

La **federación** es la capacidad que tienen los servidores de comunicarse entre sí, para extender el alcance de cada sistema cliente-servidor. Un ejemplo de protocolo que permite federación sería XMPP. Un usuario registrado en un servidor XMPP puede enviar mensajes a otro usuario de otro servidor XMPP, si ambos están federados o admiten el envío/recepción entre diferentes servidores. El correo electrónico funciona exactamente igual; por ejemplo, un usuario de GMail puede enviar correos a usuarios de Protonmail, Yahoo,... y viceversa.

Un servicio descentralizado también puede existir cuando no hay servidores de por medio y la conexión es directa entre los nodos, aunque a este tipo de situación se le conoce mejor como un servicio serverless o sin-servidor.

Un servicio centralizado tiene como ventaja principal la mejor gestión de la información del servicio, porque al estar centralizado, no se depende de otras partes que puedan retrasar o obstaculizar dicha gestión. Sin embargo, un servicio descentralizado es más resiliente, porque no depende de un único servidor que, de fallar, dejaría a los usuarios sin servicio.

Servidor, término problemático en la guía

A la hora de usar la palabra servidor pueden resultar confusiones. Hay que diferenciar entre, por un lado, el servidor como software y, por otro lado, el servidor como máquina o dominio. Diferenciarlo es fácil:

- Un servidor como software es, por ejemplo, **Prosody** para el protocolo XMPP. Es un software o programa que puede desplegarse en una máquina para implementar todas las funciones del servidor y dar servicio a usuarios.
- Un servidor como máquina o dominio, en cambio, es el servidor concreto (físico, una máquina) y su nombre. Por ejemplo, *xabber.org* sería un dominio que apunta a una máquina en la que se está ejecutando un software de servidor XMPP (Prosody, **Ejabberd** o el que sea) y *xmpp.is* sería otro dominio apuntando a otra máquina ejecutando otro software de servidor XMPP.

En general, cuando hablemos de que una aplicación de mensajería **dispone de varias implementaciones** de servidores, nos referiremos a que hay varios software de servidor que pueden utilizarse para desplegar un servicio. Va de la mano que, si existen varias implementaciones de servidor como software de código abierto, entonces también será posible que existan varios dominios que los implementen.

Sistemas de mensajería a evitar

Whatsapp, Telegram y mensajes directos de redes sociales

WhatsApp y Telegram son dos de las aplicaciones más usadas por todo tipo de usuarios y, por desgracia, muy usados en las organizaciones políticas del proletariado. Ambas son propiedad de empresas que atienden a sus intereses privados lucrativos, donde es precisamente la información lo primordial dentro de su modelo de negocio.

En el caso de WhatsApp, el código del cliente y el servidor son cerrados, para Telegram en cambio, sólo el código del servidor está cerrado. Al no saber cómo funcionan internamente estos servicios, pero sabiendo claramente que lo que les interesa a estas corporaciones es la información, no hay razón alguna para confiar en estos medios de comunicación, más aún si se trata de organizar y hablar sobre cuestiones sensibles, en términos políticos. Añadido a esto, en ambos casos la ausencia del código del servidor imposibilita el autoalojamiento del servicio, por lo que al ser una red centralizada, cuando el servidor esté caído la comunicación queda interrumpida; WhatsApp ya es famoso por tener repetidas caídas de su servicio.

En cuanto a la privacidad, es bien sabido que el cifrado de estas aplicaciones tiene fallas intencionadas, pues varias han sido las veces que se ha reconocido poder acceder a los mensajes en caso de que las autoridades lo requieran. Telegram ni siquiera tiene el **cifrado de extremo a extremo** activado por defecto y la seguridad de su protocolo de cifrado es puesta en entredicho por los expertos (1 2, 3 y 4). Además, Telegram cumple con la normativa Europea y otras normativas que la sitúan bajo la obligación de proveer la

información (conexiones de usuarios, direcciones IP, mensajes, etc.) en caso de petición judicial.

Con la funcionalidad de chatear o enviar mensajes directos en redes sociales (Twitter/X, Instagram, Snapchat, etc.) ocurre exactamente lo mismo que con las aplicaciones de mensajería comerciales. En muchas redes sociales no existe cifrado (por lo que las conversaciones no son secretas) y en las que existe (muy pocas, de hecho), simplemente **se supone que existe**, pero quién sabe.

Resumiendo, es altamente recomendable **NO** utilizar o minimizar el uso de estos servicios, sobre todo para actividades de organizaciones revolucionarias. Es más, el objetivo de esta guía es proveer alternativas e información para poder prescindir de servicios como Whatsapp o Telegram.

Sistemas de mensajería recomendados

Signal

Topología: Cliente-Servidor

Signal es una conocida aplicación de mensajería instantánea, centrada en la privacidad y la seguridad. Los mensajes, archivos y llamadas de Signal están cifrados punto a punto, por lo que el servidor central no puede acceder al contenido de los mensajes enviados.

El servidor y el cliente son independientes, por lo que pueden utilizarse diferentes clientes para usar Signal. Sin embargo, existen muy pocos de estos clientes alternativos a la aplicación oficial de Signal. Uno de ellos es **Molly**, el cuál recomendamos utilizar en su **versión FOSS**, porque tiene **seguridad añadida** y se está planeando introducir características que podrían ser de utilidad para la militancia como, por ejemplo, el borrado remoto de mensajes (en caso de que el dispositivo móvil acabe en manos de la policía). Por tanto, recomendamos utilizar Molly FOSS, en lugar de la aplicación oficial de Signal.

Hace un tiempo Signal introdujo la posibilidad de configurar nombres de usuario, lo cual supuso un paso adelante y la superación de una desventaja clásica de este servicio, que era la necesidad de compartir el número de teléfono para poder chatear. Ahora, aunque sigue siendo necesario el número de teléfono para el registro, no es necesario compartirlo para poder conversar con otras personas, ya que se puede comenzar a hablar conociendo el nombre de usuario.

Plataformas:

- Android

- iOS
- Microsoft Windows, Linux y MacOS

¿Se puede autoalojar?

Sí, pero **no a efectos prácticos**. El código del servidor de Signal es **abierto** y puede ser desplegado por cualquiera con los correspondientes recursos y conocimientos. Por la cantidad de recursos necesarios y la complejidad en el despliegue, nadie suele desplegar un servidor de Signal y es por ello que decimos que a efectos prácticos no se puede autoalojar.

Mejores aplicaciones cliente:

- **Molly**
- **Signal** (oficial)

Ventajas:

- Cifrado robusto y **post-quantum** (resistente a computadoras cuánticas).
- El código y el protocolo de Signal dispone de una buena reputación y ha sido auditado para comprobar su seguridad.
- La experiencia de usuario es muy amigable. Dispone de todas las características existentes en aplicaciones de mensajería comerciales como Whatsapp.
- Se pueden utilizar **nombres de usuario**, por lo que no es necesario compartir identificadores personales como el número de teléfono.
- En el servidor no se almacena apenas información: solo la fecha de creación y la fecha de última conexión (**más información**).

Desventajas:

- Se necesita proveer el número de teléfono móvil para registrarse.
- A efectos prácticos es complicado autoalojarlo.
- La oferta de aplicaciones cliente es extremadamente reducida.

Casos de uso:

- Es muy recomendable utilizar Signal, sobre todo mediante la aplicación **Molly**, para **comunicaciones relativas a la actividad política y militante**. Es **extremadamente recomendable utilizar los nombres de usuario** para este tipo de actividades. Solo se debe compartir el número de teléfono con gente conocida y de confianza, debido a que el número de teléfono móvil es un identificador personal directo.
- Uso diario con amigos, familia, etc.

SimpleX Chat

Topología: Cliente-Servidor

SimpleX Chat es un sistema de mensajería cifrado basado en el protocolo **SimpleX Chat Protocol** que a su vez se basa en el protocolo **SimpleX Messaging Protocol (SMP)**. Este protocolo consiste en el envío de mensajes por colas (vías) unidireccionales, apoyándose en un servidor intermedio. Esto quiere decir que un usuario necesita una cola para enviar y otra para recibir, pudiendo estas colas estar en diferentes servidores. Además, dispone de mecanismos para ocultar la IP de origen y la sesión a los servidores escogidos por el receptor. Las conexiones entre el cliente y el servidor están cifradas y autenticadas, del mismo modo que los mensajes enviados, que están **cifrados de extremo a extremo** con algoritmos postcuánticos (resistentes a computadoras cuánticas). Para más información técnica, puede consultarse su *whitepaper*.

El sistema de mensajería SimpleX no hace uso de identificadores asignados a los usuarios, sino que usa las direcciones de las colas unidireccionales (simplex) como método para enviar/recibir mensajes. Es como tener una cuenta diferente para cada contacto con el que habla el usuario. Esto es una gran ventaja, ya que permite crear y eliminar cuentas de forma fácil, a la par que mantener sesiones temporales o diferenciar sesiones para cada tipo de actividad. Sin embargo, puede ser un problema de cara a establecer contactos a largo plazo, ya que los identificadores de las colas son difíciles de memorizar (no es como aprenderse un nombre de usuario) y puede que el usuario las vaya cambiando o eliminando sobre la marcha. Toda la información del usuario se almacena en el dispositivo del propio usuario. Los servidores solo adoptan el rol de simples puentes para el envío de mensajes y solo almacenan los mensajes hasta que pueden enviárselos a su receptor. Los servidores también pueden tener una **dirección .onion asociada** para conectarse vía **red Tor**. Puede encontrarse más información en [este enlace](#).

Los mensajes están **cifrados de extremo a extremo** (con el **mismo algoritmo que Signal**), por lo que ni el servidor ni ningún posible interceptor puede ver el contenido de estos mensajes. Solo los usuarios emisor y receptor pueden leerlos. Además, el cifrado es **postcuántico**, protegiendo ante los ataques más sofisticados. También permite **realizar llamadas de audio y video cifradas**, por medio de WebRTC. La base de datos local de la aplicación **SimpleX Chat** está cifrada, de modo que la información y los mensajes recibidos y almacenados en el dispositivo permanecen cifrados. Esta base de datos está diseñada para que sea portátil, por lo que puede exportarse para su uso en otros dispositivos. Dispone de un **modo incógnito** para ocultar la imagen y nombre de perfil a los nuevos contactos, permitiendo conexiones anónimas con usuarios y grupos.

El **modelo de amenaza** expuesto en su documentación aclara muy bien qué puede y qué

no puede hacer un atacante que quisiera sabotear o interceptar mensajes en la red SimpleX y saber entre quiénes y qué se habla.

Además de ser un protocolo abierto, las implementaciones existentes también son de **código abierto** y se pueden autoalojar, tanto el servidor SMP para el envío/recepción de mensajes, como el servidor **XFTP** para la transmisión de archivos. Por tanto, cada organización puede levantar sus propios servidores y crear su propia red SimpleX.

Para ampliar la información sobre SimpleX puede leerse su **guía de uso** y sus **preguntas frecuentes**.

Plataformas:

- Android
- iOS
- Microsoft Windows, Linux y MacOS

¿Se puede autoalojar?:

Sí. Pueden autoalojarse tanto los servidores SMP (**documentación**) para el envío y recepción de mensajes, como los servidores XFTP (**documentación**) para la transferencia de archivos. Para llamadas de audio y vídeo, puede usarse cualquier servidor WebRTC ICE (**guía**)

Mejores aplicaciones cliente:

- **SimpleX Chat**

Ventajas:

- Es la aplicación de mensajería con el diseño más privado y seguro que conocemos, gracias a su sistema de colas, a la ausencia de identificadores de usuario, a la protección de los metadatos, a que oculta la dirección IP al destinatario, etc. Los servidores tiene muy poco conocimiento y no almacenan información privada de los usuarios.
- Cifrado extremo a extremo robusto y postcuántico (como el de Signal).
- Autoalojable, para no depender de servidores de terceros.
- Protocolo abierto, por lo que cualquiera puede desarrollar clientes, servidores, bots, mejoras al protocolo, etc.
- La aplicación de referencia (**SimpleX Chat**) ofrece una experiencia de usuario homologable a la de las aplicaciones de mensajería más populares.

Desventajas:

- Al no disponer de identificadores de usuarios estables, dificulta conocer el nombre o identificador de usuario a largo plazo. Por ejemplo, en Signal el nombre de usuario o el número de teléfono nos puede servir a la larga para hablar con alguien, en caso de que perdamos el acceso a nuestra lista de contactos (en situaciones en las que perdemos el dispositivo o similares). Otros ejemplos son los correos electrónicos o cuentas de redes sociales, que son fáciles de memorizar o compartir. A diferencia de estos ejemplos, las direcciones SimpleX son complejas y volátiles, por lo que se suele depender de estar en contacto de forma frecuente con nuestro destinatario, cara a cara o por otras vías de mensajería.
- Por ahora solo dispone una aplicación cliente **SimpleX Chat**, aunque funciona muy bien.

Casos de uso:

- Es la **opción más segura para el ámbito militante**, pero **requiere compromiso** entre los militantes para estar al día de las direcciones SimpleX de cada uno, por lo que **no es la más cómoda**.
- Grupos de difusión, discusión... Es suficiente con crear un grupo y compartir el enlace, para que cada cual se una. No se depende de estar al día de las direcciones, ya que la gente se une al grupo mediante una dirección pública. Sin embargo, si los titulares del grupo pierden la cuenta SimpleX, se pierde la capacidad de moderación del grupo ("se queda en el limbo").

XMPP

Topología: Cliente-Servidor

XMPP es un protocolo de comunicación extensible de propósito general para el intercambio de mensajes.

En el aspecto técnico, su arquitectura es descentralizada y federable, idéntica a la del correo electrónico. Por ejemplo, un usuario en un servidor (usuario1@dominio1.org) puede comunicarse con otro usuario de otro servidor (usuario2@dominio2.org) si los servidores activan la federación. Una vez más, cliente y servidor son independientes y existen múltiples implementaciones de **clientes** y **servidores** para elegir.

El protocolo no está cifrado de base, por lo que son los clientes los que pueden o no implementar el cifrado de mensajes, mediante tecnologías que se han añadiendo con el tiempo como diferentes extensiones del protocolo. Hoy por hoy, la mejor alternativa de cifrado para XMPP es OMEMO. Por tanto, es importante buscar clientes que lo implementen.

Sin embargo, XMPP es muy problemático a nivel de ingeniería (1 y 2). No es un protocolo cifrado por defecto y su mejor **extensión de cifrado extremo a extremo, OMEMO, adolece de varios problemas**, que lo hace bastante menos seguro que opciones como Signal o SimpleX. Por tanto, si lo recomendamos en esta guía es porque es fácil de autoalojar y tiene varios clientes de código abierto entre los que elegir. En casos donde los requisitos de seguridad no son muy exigentes y donde el servidor es controlado por los propios usuarios, puede ser una buena opción. Sin embargo, solo recomendamos usarlo en estos entornos controlados (un servidor privado de cada organización), sin federación y a posible como un **servicio en la red Tor**, no accesible desde Internet.

Plataformas:

- Android
- iOS
- Microsoft Windows, Linux y MacOS
- Web

¿Se puede autoalojar?:

Sí y además es de los servicios de mensajería más fáciles desplegar. De toda la gama de **implementaciones de servidores XMPP** que existe, la opción más sencilla nos parece la de **Prosody**, aunque **Ejabberd** es otra buena opción, pero más compleja.

Mejores aplicaciones cliente:

- **Conversations** (Android)
- **Monal** (iOS)
- **Dino** (Linux)
- **Gajim** (Linux, Windows)

Ventajas:

- Cifrados como OMEMO, OTR y PGP. Aunque no es la mejor opción en este sentido.
- Fácil de autoalojar y liviano en consumo de recursos.
- Amplia gama de clientes y servidores.
- No requiere identificadores como correo electrónico o número de teléfono para registrar cuentas.

Desventajas:

- Dispone de pocas funcionalidades, aparte de las tradicionales: envío de texto, imágenes, vídeos, documentos, audios y llamadas.
- Adolece de varios problemas de diseño técnico que no lo hacen del todo robusto,

eficiente, etc.

- Su cifrado no es del todo recomendable cuando los requisitos de seguridad son exigentes.
- No protege los metadatos de los mensajes enviados.

Casos de uso:

- Entorno privado y controlado entre personas de confianza. Mejor si se aloja como **servicio en la red Tor**, no accesible desde Internet.

Servicios que dejamos de recomendar

Sabemos que existen muchos más sistemas de mensajería y de los que podríamos llegar a recomendar (por ser de código abierto, porque priorizan la seguridad, son cifrados, etc.) hemos probado e investigado muchos, pero no podemos recomendar ninguno más de los que ya hemos recomendado. En este apartado vamos a ajustar cuentas con algunos de los sistemas de mensajería que recomendábamos hasta ahora, para indicar por qué ya no los recomendamos.

Matrix:

Matrix es un **protocolo** de comunicación de código abierto que ha adquirido mucha popularidad en entornos de organización de software libre y público, gracias a su diseño con enfoque seguro y una gran comunidad de desarrolladores, hasta el punto que está siendo adoptada incluso por las administraciones de diferentes gobiernos. Las conversaciones están cifradas extremo a extremo. Hay que subrayar que Matrix no es un servicio de mensajería, sino un protocolo de mensajería. Dicho de otra manera, Matrix define una serie de reglas, requisitos y detalles técnicos para implementar dicho protocolo. Al ser así, existen muchas implementaciones de **servidores** y **clientes** para Matrix. El usuario dispone de libertad para elegir si conectarse a un servidor existente o a uno **alojado por cuenta propia**.

Históricamente, se han encontrado **agujeros de seguridad importantes en el diseño de la criptografía**. También en la **implementación de alguna librería utilizada para el cifrado**.

Por cuestiones técnicas de diseño, los servidores almacenan ingentes cantidades de metadatos que se sincronizan con o se replican a otros servidores federados cuando un usuario de un servidor contacta con otro usuario. Estos metadatos se transfieren sin cifrar, por mucho que el contenido de los mensajes esté cifrado. Lo anterior supone un gran problema para la privacidad de los usuarios.

Los problemas en la criptografía y en el diseño del protocolo no nos convencen de cara a

recomendar Matrix como una opción segura y privada de mensajería instantánea. Es mejor que muchas de las opciones comerciales, pero peor que otras alternativas como las que hemos recomendado.

Briar:

Briar es una aplicación de mensajería cuyo objetivo principal es poder permitir una comunicación segura y anónima incluso en situaciones en las que la red de Internet no se encuentra disponible. Las cuentas en Briar no requieren de ningún tipo de identificador. Al usuario se le asigna un identificador único dentro de la red y a través de ese identificador puede ser agregado por otros usuarios. Todo el tráfico de Briar pasa por la red Tor, de tal modo que se preserve el anonimato de los usuarios, frente a posibles identificaciones por parte de las fuerzas represivas del Estado. Es más, aunque interviniesen uno de los dispositivos con Briar instalado, no serían capaces (en teoría) de identificar a las personas con las que dicho dispositivo se comunica. La topología es en malla, es decir, que cada nodo (dispositivo) se apoya en los demás para hacer llegar los mensajes. Esto es especialmente útil para Briar, porque dispone de la funcionalidad para poder comunicarse por WiFi o por Bluetooth, en caso de que la red de Internet se encuentre caída o censurada.

Dispone, además, de la posibilidad de crear grupos, foros y blogs, lo que es realmente útil para organizar acciones y flujos de información. No obstante, la experiencia de usuario es bastante mala (la interfaz de usuario es poco amigable y hay problemas para recibir las notificaciones en tiempo real).

Dejamos de recomendarla porque **SimpleX** soluciona infinitamente mejor todos los problemas que quiere afrontar Briar, excepto la posibilidad de usar el sistema de mensajería sin internet, en redes locales (WiFi, Bluetooth) y creando una red en malla. Briar puede ser una buena solución en estos casos, pero son casos muy especiales. Incluso en estas situaciones sin Internet, desplegar amplias redes WiFi o redes locales de otro tipo que permitan una infraestructura de comunicación requiere de una gran implicación de gente con conocimientos técnicos para ello. Además, cuando se producen estas situaciones (sobre todo en guerras o catástrofes naturales), suelen ser situaciones donde también es difícil generar la infraestructura necesaria para que Briar sea una alternativa real. Finalmente, incluso si se tuviese la capacidad para generar redes autónomas, se tendría capacidad para desplegar servidores locales SMP para usar SimpleX. En este caso, Briar solo tendría la ventaja de no tener que desplegar servidor alguno.

Session

Session es una aplicación disponible para varias plataformas, cuyas principales características son la descentralización, el anonimato y que no filtra metadatos. Se apoya sobre una red de nodos que funcionan parecido a la red Tor, pero la red es propia de **Oxen** ([más información](#)). Esto significa que los mensajes rebotan en varios nodos antes de llegar al destinatario, haciendo anónimas a las dos partes. La aplicación está basada en Signal, algo que se ve claramente en la interfaz de usuario. La funcionalidad es parecida, solo que en Session no se depende de un servidor central. En resumen, Session viene a ser un Signal sin números de teléfono en el registro, sin filtrar metadatos y que se apoya en una red parecida a la red Tor para anonimizar a los usuarios y almacenar la información de forma descentralizada.

Un gran problema con Session, que lo deja en términos de seguridad detrás de otros sistemas de mensajería instantánea como Signal o SimpleX, consiste en que **elimina la *perfect forward secrecy* (PFS)**, que es una propiedad que impide que las comunicaciones pasadas no puedan descifrarse en caso de que se haya comprometido una clave de sesión ([más información sobre PFS](#)).

Además, Session está vinculada a **Oxen**, una organización que provee servicios de criptomonedas. Tiene sede en Australia, donde existen leyes que permiten intervención y vigilancia en servicios digitales.

En la comunidad preocupada por la privacidad es un servicio que causa muchas dudas y nosotros no hemos ahondado profundamente en investigarla como para poder recomendarla o prevenir de su uso. En cualquier caso, solo con el hecho de eliminar la PFS mencionada anteriormente, queda en peor lugar que el resto de sistemas de mensajería que recomendamos.

Conclusiones

Tras este repaso por esta selección de aplicaciones de mensajería segura, queremos poner sobre la mesa varias advertencias y dar nuestras recomendaciones finales.

Siempre y cuando sea posible, es preferible no tratar cuestiones sensibles por medios telemáticos. De hecho, en la actividad militante es importante minimizar el uso de estas vías de comunicación, ya que siempre existen posibilidades de fallos o vulnerabilidades desconocidas por los desarrolladores, pero conocidas por las fuerzas del orden. Además, hay que tener muy claro que la comunicación puede ser segura, pero los dispositivos pueden estar comprometidos ("*pinchados*") y, por tanto, se lean todos los mensajes antes de ser enviados y/o nada más ser recibidos. Ahora bien, si es necesario utilizar medios de comunicación como estos, hay que ser serios y utilizar los medios más seguros posibles, para todo tipo de actividad telemática: llamadas, mensajes, transferencia de archivos, etc.

Estos medios de comunicación deberían sustituir los viejos medios de comunicación entre organizaciones. El sistema clásico del correo electrónico o de la telefonía móvil están obsoletos desde hace años en muchos de los aspectos que nos interesan como militantes y deben ir siendo desplazados poco a poco.

Los sistemas de mensajería instantánea que proponemos también ofrecen oportunidades para organizar comunidades y actividades militantes. Por ejemplo, grupos en los que se comparta información pública de organizaciones (los típicos canales) o grupos/foros en los que se realicen debates. Lo importante es saber determinar si es un lugar adecuado para reunir y conectar a personas u organizaciones y contemplar bien los riesgos asociados a cada uso. Una organización puede utilizar Whatsapp o Telegram para organizar canales o grupos informativos entre las amplias masas de trabajadores, donde la privacidad y la seguridad no van a ser un gran problema (si son públicos significa que deben estar disponibles para todo el mundo, con todo lo que conlleva). Pero también podría organizar grupos más privados o selectos, de carácter semi-público o directamente privado, donde el cifrado y la seguridad en general son importantes. Para estos casos, habría que utilizar sistemas de mensajería seguros, por lo que Whatsapp, Telegram y similares no servirían para este objetivo (y habría que evitarlos a toda costa).

Recomendaciones

- **Uso militante:** *Signal* y *SimpleX Chat*. Para usar Signal debe priorizarse hacerlo mediante la aplicación Molly (solo en Android) y mediante el uso de nombres de usuario. Signal permite una mayor estabilidad en los identificadores de cuentas (número de teléfono, nombre de usuario), lo que lo hace más útil para contactos a largo plazo. SimpleX Chat es autoalojable, permitiendo a las organizaciones utilizar sus propios servidores; sin embargo, ponerse en contacto es difícil, porque los identificadores de cuentas son más largos y complejos. Mientras que los usuarios de Signal son fáciles de compartir y transmitir, los de SimpleX requieren o bien estar cara a cara para escanear los códigos QR o bien pasar los identificadores por otros medios, como Signal. Estos servicios permiten cambiar los nombres de usuario (Signal) o recrearlos fácilmente (SimpleX). Ambos servicios de mensajería son los más seguros existentes hasta la fecha y, aunque siempre hay que evitar hablar por medios telemáticos, si hay que hacerlo, mejor hacerlo por estos: para chatear, para llamar y para compartir archivos.
- **Uso diario:** *Signal*, a través de la app *Molly*. Es una aplicación conocida por su seguridad, su usabilidad, su carácter intuitivo y su robustez. Es el sustitutivo ideal para aplicaciones como Whatsapp o Telegram, perfecto para comunicarse con amigos, familia, etc. También es una buena plataforma para hacer grupos de difusión o discusión, ahora que existen los nombres de usuario. Debería sustituir a Telegram en

este último aspecto, aunque Signal limita el número de miembros de un grupo a 1000 (frente a los 200.000 de Telegram). Aunque es cierto que esos grupos de difusión serían públicos y, por tanto, no tendría sentido el cifrado de los mensajes, no es menos cierto que el servidor de Signal no guarda ninguna información de la IP de origen de los usuarios, ni los metadatos, ni datos de interés para la policía (solo guarda la fecha de creación y de última conexión de una cuenta), lo cual es una gran ventaja frente a Telegram, que guarda absolutamente todo lo que puede y lo comparte con las autoridades.