# Zur Nutzung von XMPP (Jabber) mit OTR und OMEMO

Kontakt (Kritik | Hilfe | Rückfragen):
*rc4@jabber.systemli.org* [OTR-Fingerprint: *619CCB09 D120E17E 84CEBE8B BCC51AAE 5E17FB68*]

Inhalt:



## Einleitung

Vorweg: *Dieses Dokument ist nicht auf die möglichst sichere Nutzung von Jabber ausgelegt sondern soll vielmehr einen Einsteig bieten.*

Zum Nachrichtenaustausch gibt es die verschiedensten Protokolle. Das sind genaue Ablaufpläne wie Nachrichten gesendet, übertragen und empfangen werden, in welchem Format sie versendet werden und was beachtet werden muss, damit keine Fehler geschehen.

Ein solches Protokoll ist *XMPP* (Extensible Messaging and Presence Protocol). Es existiert unter dem Namen *Jabber* seit 1999, wurde 2004 modifiziert und wird (mittlerweile stark abgewandelt) unter anderem von Diensten wie Google Mail oder Facebook verwendet.

Jabber ermöglicht neben der direkten Kommunikation zweier Menschen auch Gruppenchats und Datentransfer (welchen wir nicht nutzen werden). Zudem kann die Kommunikation verschlüsselt werden, z.B. Mit *PGP*, *OTR* oder *OMEMO*.
*PGP* (Pretty Good Privacy), vielleicht von der Mail-Verschlüsselung bekannt, werden wir nicht behandeln.
Weiter unten findet sich noch Allgemeines zu "sicherer" Kommunikation.

XMPP in Kombination mit Verschlüsselung kann die meisten Anforderungen zum Chatten erfüllen und ist somit eine gute Alternative zu den meisten Messengern, auch zur E-Mail. Mal von Fotos und GIF's abgesehen, daran wird noch gearbeitet. Für einen Kanal (wie u.a. von *Telegram* bekannt) kann eine Schleuder verwendet werden. Also ein Programm mit eigenem Account, das automatisch alle eingehenden Nachrichten an bestimmte Empfänger*innen verteilt.

Um *XMPP* (und Verschlüsselung) zu nutzen benötigen wir Programme, die das Protokoll verstehen (und die Verschlüsselung anwenden können). Solche Clients gibt es in unterschiedlicher Form, mit unterschiedlichen Funktionen. Später werden ein paar behandelt.

Auch wenn ein anderer Eindruck entstehen mag: *Jabber/XMPP* kann natürlich auch unverschlüsselt genutzt werden. Der Schritt zu einer frei verfügbaren und von Konzernen (mit Verschlüsselung als Geschäftsmodell) unabhängigen Kommunikation ist allerdings verhältnismäßig klein und meist relativ unkompliziert.

**Anonymität:**

Das Protokoll *XMPP* kann sehr Meta-Daten-arm sein. Also keine verräterischen Informationen über Standort, Gerät, Kontakte etc. weitergeben. Leider sind bestimmte Meta-Daten nicht zu vermeiden. Um Daten außutauschen nutzen wir das Internet(protokoll). Dieses Protokoll benötigt immer eine Zieladresse und eine Absendeadresse. Diese Internetprotokoll-Adressen (kurz IP-Adressen) können mit Telefonnummern verglichen werden. Genau wie bei einer Telefonverbindung müssen alle Punkte zwischen Anrufer*in und angerufener Person wissen woher der Anruf kommt und wohin er geht. Beim Internet sind das der jeweilige Internetkonzern, Netzknotenpunkte und alle die Zugriff darauf haben. Nun weiß der eigene Internetkonzern welcher realen Adresse ein bestimmter Internetanschluss und somit auch eine IP-Adresse zuzuordnen ist. Wer diese Zuordnung vermeiden möchte kann die eigene IP-Adresse mit Anonymisierungs-Diensten wie dem *Tor*-Netzwerk verschleiern. Später werden auch Möglichkeiten zu Nutzung des *Tor*-Netzwerkes angeschnitten.

**Begriffserklärungen:**

Nun sollen noch ein paar Begriffe und ihre Verbindung zueinander klar gestellt werden:

Ein **Account** (ein Benutzer*innen-Konto) ermöglicht die Nutzung von Internet-Diensten. Der Account hat eine Identität, alson einen Namen und wird mit einem Passwort an reale Identitäten, also Personen gebunden.

Der Account wird auf einem **Server** angelegt. Das ist ein Computer, der bestimmte Programme installiert hat, um zB. Email- / Jabber-Nachrichten zu verteilen oder auf dem eine Webseite gespeichert ist. Der Server kann über das Internet angesprochen werden, um zB. eine Nachricht zu versenden oder um eine Webseite herunterzuladen (die dann im Internet-Browser angezeigt wird). Wenn ein Server Dienste (Mail, Online-Banking oder ein Spiel) anbietet liegen auf dem Server Listen mit Accounts. So kann eine Nachricht oder ein Bankkonto einer bestimmten Identität zugeordnet werden.

Um die Server anzusprechen brauchen wir meistens **Clients**. Das sind Programme nach bestimmten Ablaufplänen, also Protokollen, die mit den **Servern** kommunizieren. Ein Internet-Browser fragt eine Webseite an, lädt diese auf den eigenen Computer und zeigt sie dann an. Ein Jabber-Client meldet sich mit den Zugangsdaten eines Accounts auf dem Server an, um Nachrichten zu empfangen und zu versenden.

Um zu verschleiern was Client und Server (oder Server und anderer Server) austauschen wird die Kommunikation verschlüsselt. Diese **Transportverschlüsselung** wird mit den Methoden SSL oder TLS realisiert (das *s* bei *https* steht für SSL). Um aber sicher zu gehen, dass eine Nachricht erst von der Empfänger*in gelesen werden kann, wird **Ende-zu-Ende-Verschlüsselung** verwendet (zB. PGP, OTR oder OMEMO). Diese Verschlüsselunsmethoden (auch Chiffre, eng. Cipher) arbeiten meistens mit einem öffentlichen und einem privaten Schlüssel (eng. *Key*). Der öffentliche Schlüssel, public key, kann mit einem Klickschloss verglichen werden: Alle können es schließen, aber nur mit einem Schlüssel kann es wieder geöffnet werden. Dieser Schlüssel ist der private key, der private Schlüssel.

# Clients

Im Folgenden sollen ein paar Clients exemplarisch vorgestellt werden. Alle Anwendungen sind komplett OpenSource (der Quellcode ist öffentlich, d.h. öffentlich nachvollziehbar). Ein gemeinsamer Nachteil ist, dass keine der Anwendungen lokale Daten (wie Account-Passwort oder kryptographische Schlüssel) verschlüsselt abspeichert. Es ist zu empfehlen, das entsprechende Gerät zu verschlüsseln und es abzuschalten sofern es nicht genutzt wird (natürlich auch bei Bullen-Kontakt). Anleitungen und Tipps zur Installtion finden sich weiter unten.

## TorMessenger

*Betriebssysteme: Linux, Windows, macOS*
Kommentar:
*TorMessenger* ist auf hohe Sicherheit ausgelegt. Es ist z.B. nicht möglich, aus Versehen unverschlüsselte Nachrichten zu versenden. Zudem wird der Datenverkehr automatisch über Tor geleitet. Das verwendeten Jabber-Protokoll ist sehr Meta-Daten-arm. *OTR* ist vorinstalliert. Leider ist *OMEMO* noch nicht eingebaut. Somit sind Gruppenchats nicht möglich.

## Conversations

*Betriebssystem: Android*
Kommentar:
*Conversations* ist eine einigermaßen Energie sparende Anwendung die *OTR* und *OMEMO* unterstützt. *PGP* und *Tor* können mit weiteren Apps ebenfalls genutzt werden.

## ChatSecure

*Betriebssystem: iOS*
Kommentar:
*ChatSecure* unterstützt *PGP*, *OTR* und *OMEMO*. Außerdem ist Tor direkt implementiert.

## Gajim

*Betriebssyteme: Linux, Windows, Un*x*
Kommentar:
*Gaijm* bietet mit vielen Protokollen und Erweiterungen etliche Funktionen – was natürlich auch immer eine Schwachstelle sein kann. Mit *Tor* oder *TorBrowser* kann *Gaijm* über das Tor-Netzwerk senden und empfangen. Zudem wird neben und *OTR* auch *OMEMO* unterstüzt.

## Pidgin

*Betriebssyteme: Linux, Windows, BSD, Solaris*
Kommentar:
Ähnlich wie *Gajim* bringt *Pidgin* sehr viele Möglichkeiten mit (nicht zu letzt auch Twitter, IRC und Skype) – und damit auch viel Angriffsfläche. Mit *Tor* oder *TorBrowser* kann *Pidgin* über das *Tor*-Netzwerk senden und empfangen. Eine Erweiterung um OMEMO nutzen zu können findet sich hier: *https://github.com/gkdr/lurch*. Für MacOS erweist sich die Installtion der Erweiterung derzeit als komplizierter (*https://github.com/gkdr/lurch/issues/8*).

## Server

Nun läuft unser ganzer Datenverkehr über einen Server. Diesem sollten wir vertrauen können – zumindest so weit, dass keine unnötigen Meta-Daten gespeichert und notwendige Daten (wie Kontaktlisten, letztes Online-Datum) verschlüsselt abgespeichert werden. Ein anderer Aspekt ist die Kommunikation zwischen unterschiedlichen Servern, diese sollte sinnvoll verschlüsselt von statten gehen. Bei der Serverwahl kann mensch in Betracht ziehen, auf einem weiteren Server ein Ausweich-Konto anzulegen. Damit könnte eine Kommunikation auch möglich sein, wenn der Staat einen Server unnutzbar macht. So geschehen während dem Aufkommen von Pegida: Der Server von *systemli.org* wurde regelmäßig durch zu viele gleichzeitige Anfragen zum Absturz gebracht. Daraufhin legten sich viele Leute einen Ersatz-Account mit gleichem Nutzer*innenamen bei *ccc.de* an.

Der Account, bzw die Account-Adresse hat die Form: *benutziname@server.xy* (also gleicher einer E-Mail-Adresse). Oft begegnet uns auch eine Adresse die auf *@jabber.server.xy* endet.

Hier ein paar Server die nach obigen Kriterien cool sind:

systemausfall.org *(jabber.systemausfall.org)* – bisher kein *OMEMO*-Support; gerade nicht erreichbar

*riseup.net* – einfach mit E-Mail-Account; kein *OMEMO*

ccc.de (*jabber.ccc.de*) – Registrierung direkt im Client möglich; es gab Vorwürfe bzgl. gespeicherten Daten

systemli.org (*jabber.systemli.org*) – auch *OMEMO*

*conversations.im* – kostet 8 Euro pro Jahr

*jabber.cat* – kann alles; nach 3 Monaten offline werden Accounts gelöscht

*5222.de* – keine Zecken, ansonsten alles verfügbar; wenig gespeicherte Daten

## Zur Nutzung

**Kontakte / Buddys hinzufügen, verwalten und anschreiben:**

*Mit Kontakten und Buddys ist das gleiche gemeint.*

Was alle Clients gemeinsam haben: Irgendwo gibt es einen Knopf um **Kontakte hinzuzufügen**. Bei dem *TorMessenger* ist das: *File* (in der Menü-Leiste) und dann *Add Contact* auswählen. Bei den meisten Clients wird nun eine Kontaktanfrage von dem eigenen Account an einen Anderen gestellt. Sobald die Nutzer*in des anderen Accounts die Anfrage bestätigt, wird automatische eine Anfrage zurück an den eigenen Account gestellt. Eine Besonderheit des *TorMessenger* ist, dass die angefragte Person manuell die Rückanfrage stellen muss. Das ist zu beachten, denn Sinn des Hinzufügens von Kontakten ist, dass wir sehen, wann der andere Account online/offline ist. Wenn

mehrere Accounts gleichzeitig mit einem Client genutzt werden, ist es wichtig aufzupassen mit welchem Anfragen gestellt und erwidert werden.
Als kleiner Tipp:
Bei *Pidgin*, *Gajim* und *TorMessenger* kann mit Rechtsklick auf die Kontaktliste ausgewählt werden, dass auch Kontakte, die offline sind angezeigt werden. Wahrscheinlich auch bei den meisten anderen Clients...

Je nach eurer Nutzung könnt ihr mit euren Freund*innen vereinbaren, keine Namen oder Kontexte für Buddies/Kontakte einzuspeichern (beim Umbenennen von Kontakten). Bei sehr vielen Kontakten kann es hilfreich sein, **Tags** zu verwenden (Gruppe 1 für alle Leute aus der Stadt, Gruppe 2 für die Leute aus der überregionalen Vernetzung, ...).

Mit einem Doppelklick (oder Einfachem bei Touch-Geräten) auf einen Kontakt wird eine **Konversation gestartet**. Danach muss die Verschlüsselung extra gestartet werden.
Verschlüsselung: Irgendwo im Chat-Fenster findet sich meist ein Symbol (oft ein Schloss) mit dem eine private, also **verschlüsselte Konversation** begonnen wird, was meistens ein paar Momente dauert.

**Links** nicht direkt anklicken, sondern kopieren und im Browser (zB. *Firefox* oder gleich dem *TorBrowser*) öffnen. Mit der Nutzung des *TorBrowsers* kann hier vermieden werden, dass ein Personen-Netzwerk ersichtlich wird, z.B. indem plötzlich 13 IP-Adressen der selben Stadt die gleiche Webseite ohne Anonymisierung aufrufen.

Wenn eine **Konversation beendet** wird ist es unter Umständen wichtig a) die private Unterhaltung zu beenden und b) bei Gelegenheit das Chat-Fenster zu schließen. Zudem sollte beim Start einer privaten Konversation beachtet werden, dass die private Unterhaltung, also die Verschlüsselung frisch gestartet oder erneuert wird.

Im XMPP-Protokoll ist die Nutzung von sog. **Ressourcen** und von einer **Status**-Meldung vorgesehen.

Eine Ressource wird genutzt wenn ein Jabber-Account von mehreren Geräten verwendet wird. Nach der Account-Adresse wird dann z.B. das verwendete Gerät angezeigt:
benutziname@server.xy/*handy*. Die Ressource könnt ihr meistens in den Account-Einstellungen setzen.

Die Status-Meldung bietet meisten eine Reihe von vorgefertigten Standarts wie *Verfügbar* oder *Beschäftigt* (bei dem *TorMessenger* sind das dann auch schon alle). Zudem kann ein eigener Text hinzugefügt werden. Aus Erfahrung ist es angenehm, wenn Leute ihren Status zumindest zu *Beschäftigt* ändern, falls sie gerade nicht direkt am entsprechenden Gerät sind.
Achtung: Wenn eine Status-Meldung im Client ausgewählt/geschrieben wird, gilt das für alle genutzten Accounts.

Ein Manko bei der Nutzung von Jabber ist, dass nicht automatisch festgestellt wird, ob ein Kontakt offline ist. Deswegen liegt es in der Verantwortung von jeder Einzelnen den **Status auf *offline*** zu stellen, bevor der Laptop zugeklappt oder ausgeschalten wird. Mit dem Start oder Erneuern einer privaten Konversation kann festgestellt werden ob eine andere Person tatsächlich online ist (weil die kryptografischen Schlüssel ausgetauscht werden müssen).

Es bietet sich auch an, den **Client automatisch starten** zu lassen. Praktisch jedes Betriebssystem bietet eine einfache Möglichkeit Programme nach dem Hochfahren starten zu lassen. Damit steigt die Verfügbarkeit und somit die Nützlichkeit.

**Verifizieren:**

Um sicher zu stellen, dass eure Kommunikation nicht durch einen man-in-the-middle-Angriff (kurz MITM) mitgelesen oder auch manipuliert werden kann, muss die Identität des individuellen Schlüssels (engl. Key) der Kommunikationspartner*innen eindeutig festgestellt werden. Zunächst die Skizze eines solchen Angriffs anhand einer PGP-verschlüsselten E-Mail:

> Person A möchte an Person B eine Mail senden. Person B nutzt den PGP-Key Z. Das bedeutet sie hat den private key Z auf ihrem Computer gespeichert und kennt das entsprechende Passwort. Damit Person A verschlüsselt an sie schreiben kann, braucht sie den public key Z von Person B. Bei der Übermittlung des public key Z von Person B zu Person A wird die Mail von den Bullen abgefangen und der Inhalt wird so verändert, dass statt public key Z nun ein public key X an die Mail angehängt ist. Diesen empfängt Person A und verschlüsselt an public key X eine Nachricht (z.B. einen Treffpunkt). Die Nachricht wird von den Bullen abgefangen und da sie im Besitz ihres private key X sind können sie damit die Nachricht (die technisch an sie verschlüsselt wurde) öffnen (da sie ja zuvor mit public key X verschlüsselt wurde). Nun könnten sie (z.B. um nicht aufzufliegen) die Nachricht kurzerhand an Person B weiterleiten und zuvor mit ihrem public key Z verschlüsseln. Person B würde wahrscheinlich nicht merken, dass die Nachricht während der Übermittlung abgefangen, entschlüsselt und neu verschlüsselt wurde. Ebenso wenig weiß Person A, dass ihre Nachricht mit dem public-key X der Bullen verschlüsselt wurde.

| | | |
|---|---|---|
| [Person B] ----------[Mail]----------> | [Bullen] ----------[Mail*]-------> | [Person A] |
| versendet public key Z | fangen EMail mit public key Z ab und ersetzen diesen durch ihren public key X | empfängt EMail mit public key X |
| [Person A] ----------[Mail]----------> | [Bullen] ---------- --[Mail*]-------> | [Person B] |
| verschlüsselt mit public key X | entschlüsseln mit private key X und public key Z | entschlüsselt ihrem private key Z |

Im Client findet sich meist ein Symbol (oft ein Schloss) das eine Verifikation anbietet. Hier können wir zwischen *Frage-und-Antwort*, *Gemeinsames Geheimnis* und *Manueller Fingerprint-Abgleich* wählen. Von *Frage-und-Antwort* ist abzuraten da oftmals Fragen genutzt werden, die auch von außen mit ein wenig Kontext-Wissen gelöst werden können. Vielleicht kann bei einer Begegnung ein *gemeinsames Geheimnis* ausgemacht werden oder der Fingerprint kopiert und über einen anderen Kanal abgeglichen werden, zB. über einen Jabber-Kontakt, der bereits mit beiden Kommunikationspartner*innen verifiziert ist. Der Fingerprint darf nicht im verschlüsselten, unverifizierten Jabber-Chat abgeglichen werden. Wer Jabber anonym nutzen möchte sollte Fingerprint und Account-Name nicht über unverschlüsselte Kanäle austauschen.

Wir unterscheiden bei sicherer verschlüsselter Kommunikation zwischen fünf Kriterien:

**Verschlüsselung (Encryption)**

Niemand kann die Nachrichten mitlesen. Im Fall von OTR wird dies durch AES (Advanced Encryption Standard) gewährleistet.

**Beglaubigung (Authentification)**

Die Sicherheit, dass die Empfänger*in diejenige ist, für die man sie hält. Dafür muss allerdings extern der Fingerabdruck des Schlüssels geprüft werden. Dieser Abgleich muss out-of-band (also offline), oder mittels eines bereits verifizierten Systems (PGP-Mail) geschehen. Andernfalls ist eine Man-In-The-Middle-Attacke nicht erkennbar.

**Glaubhafte Abstreitbarkeit (Plausible Deniability)**

Eine Information kann nicht eindeutig entschlüsselt werden. Es kann argumentiert werden, dass keine eindeutige Auflösung der Verschlüsselung möglich ist.

**Perfekte Abstreitbarkeit (Deniability)**

Es sind Verfahren, um Informationen (oder deren Ursprung) so zu verbergen, so dass deren Existenz oder Ursprung nicht nachgewiesen werden kann. Dies bedeutet einerseits, dass die verwendete Mathematik und deren Umsetzung in ein Programm nach außen nicht erkennbar ist und sich als etwas anderes tarnt.

**Folgenlosigkeit (Perfect Forward Secrecy)**

Wenn der dauerhafte private Schlüssel Bullen in die Hände fällt, hat dies keine Auswirkung auf die Kompromittierung bisher getätigter Gespräche: Die Gespräche können damit nicht nachträglich entschlüsselt werden. Das wird durch die Generierung von Sub-Schlüsseln für jede Konversation gewährleistet. Zukünftige Unterhaltungen können selbstredend mitgelesen werden.

Ein Beispiel von OTR:

> “Im Gegensatz zur Übertragung verschlüsselter Nachrichten mit OpenPGP [...] kann man beim Off-the-Record Messaging später nicht mehr feststellen, ob ein bestimmter Schlüssel von einer bestimmten Person genutzt wurde (Prinzip der glaubhaften Abstreitbarkeit, englisch plausible deniability). Dadurch lässt sich nach Beenden der Unterhaltung von niemandem (auch keinem der beiden Kommunikationspartner) beweisen, dass einer der Kommunikationspartner eine bestimmte Aussage gemacht hat.” (*https://de.wikipedia.org/wiki/Off-the-Record_Messaging*)

Wie dargestellt erfüllt OTR alle Kriterien, genauso steht es bei der Weiterentwicklung von OMEMO (eine Weiterentwicklung von OTR). Im Gegensatz dazu nutzt PGP immer nur ein und den selben Schlüssel. Bei OTR und OMEMO wird der Schlüssel zu Beginn einer jeden verschlüsselten Konversation verändert.

Für OMEMO muss festgestellt werden, dass es sich um eine relativ neue Technik handelt. Es gab bisher weder ein Audit (Untersuchung ob geforderte Standards, bzw. Sicherheit von einer Software erfüllt wird) noch besteht jahrelange Erfahrung damit. Andererseits wird eine Version von OMEMO

von der Messenger-App Signal verwendet, welche reltiv professionell und unter vielen Augen entwickelt wird.

| ***Vergleich:*** | **PGP** | **OTR** | **OMEMO** |
|---|---|---|---|
| *Encryption* | X | X | X |
| *Authentication* | X | X | X |
| *Plausible Deniability* | - | X | X |
| *Deniability* | - | X | X |
| *Perfect Forward Secrecy* | - | X | X |
| *Gruppenchats* | - | - | X |
| *Datentransfer* | X | - | (X) |
| *Nutzung auf mehreren Geräten* | X | (X) | X |
| Verschlüsselte Offline-Nachrichten (d.h. die Empfänger*in ist offline) | X | - | X |

*Anmerkungen:*
Der *Datentransfer* mit OMEMO-Verschlüsselung ist vorgesehen, wird aber derzeit noch nicht von den Servern unterstützt. Die App *Signal*, welche OMEMO nutzt, ermöglicht den Datentransfer bereits.
OTR kann theoretisch auf *mehreren Geräten* mit dem dem gleichen Schlüssel verwendet werden. Allerdings wird für jede neue private Konversation ein wenig „Salz“, also Zufallswerte, in die Verschlüsselung gestreut. Das ist bei OTR Notwendig um *Perfect Forward Secrecy* zu ermöglichen. Deshalb kann eine Konversation mit OTR-Verschlüsselung immer nur von einem Gerät aus geführt werden.

**Vorsicht bewahren:**

Auch wenn eine Konversation verschlüsselt ist sollte sich nicht komplett darauf verlassen werden:

Das eigene Gerät oder das der Gesprächspartner*in könnte von den Behörden infiltriert sein. Vielleicht wird der Bildschirm mitgelesen, vielleicht werden gespeicherte Konversationen kopiert, vielleicht, ganz vielleicht, gibt es eine bisher unbekannte Schwachstelle in der Verschlüsselung-Methode. Nun setzen Behörden zwar immer mehr auf das Infiltrieren von Endgeräten, doch ist davon auszugehen, dass solche Angriff noch die Ausnahme sind – sie müssen individualisert sein, kosten Geld.

**Schwachstelle Samrtphone:**

Falls ihr sensible Daten über Jabber kommuniziert und/oder anonym bleiben möchtet ist es empfohlen das Gerät zu verschlüsseln um das Auslesen von Zugangsdaten oder Verschlüsselungskey (PGP / OTR / OMEMO) zu erschweren. Außerdem sollten regelmäßig Updates durchgeführt werden.
Die Nutzung eines Smartphones sollte immer mit der kritischen Betrachtung der Möglichkeiten einhergehen, die bestehen, um die Nutzer*in zu überwachen.

**Account anlegen:**

Hier am Beispiel systemli.org: Besucht *https://www.systemli.org/service/xmpp.html,* klickt *Registrieren* und wählt ein Passwort mit mind. 20 Stellen. Überlegt, wie einfach das Passwort zu erraten ist mit privaten Informationen über euch. Alternativ kann es sinnvoll sein eine Schlüsseldatenbank wie *keepassx* zu benutzen und sich das Passwort zufällig generieren zu lassen: *https://privacy-handbuch.de/handbuch_21j2.htm* (Mitte der Seite).

**Client installieren und einrichten:**

Auf diese Messenger wird nicht tiefer eingegangen:

**ChatSecure**

**Pidgin** ...verhält sich ähnlich bei Gajim (folgt unten), es gib zudem viele Tutorials

(zB. *https://wiki.systemli.org/howto/jabber_eng*)

**Conversations** ...ist über den PlayStore kostenpflichtig. Nicht so in der altenativen App-Verwaltung *F-Droid*. Diese kann von *https://f-droid.org/* heruntergeladen werden. Wählt in den Downloads die Datei Fdroid.apk aus womit selbige installiert werden sollte. Wahrscheinlich muss zuvor in den Einstellungen die Installation aus unbekannter Herkunft aktiviert sein. Nach dem Starten und Aktualisieren der App kann *conversations* gesucht und installiert werden.

*TorMessenger* und *Gajim* werden nachfolgend detailierter erklärt.

Der *TorMessenger* bietet sich als Client an, denn Anonymität und Nutzbarkeit standen im Vordergrund der Entwicklung. Zur Nutzung von OMEMO, was vor allem in Gruppen sinnvoll sein kann, gibt es für Laptops und Desktop-CPs gerade nur einen Client, *Gajim*, der (einigermaßen) unkompliziert eingerichtet und verwendet werden kann.

**TorMessenger**

*Diese Anleitung ist eine leicht abgeänderte Kopie dieser Webseite:*
*https://privacy-handbuch.de/handbuch_24s2.htm*
*Deswegen stimmen auch Bilder und Text nicht immer überein.*

Ladet die aktuelle Version des TorMessenger von der Webseite
*https://trac.torproject.org/projects/tor/wiki/doc/TorMessenger#Downloads Download*
herunter und entpackt sie. Fertig.
TorMessenger ist eine portable Anwendung, man kann ihn auf einem USB-Stick entpacken und damit auch problemlos in Live-Systemen wie TAILS verwenden. Tor ist im Packet enthalten und wir unabhängig vom TorBrowser automatisch gestartet.

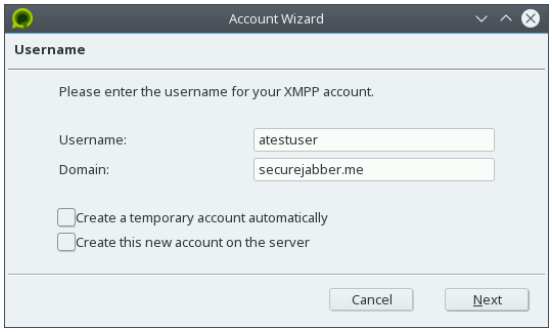
Unter Linux installiert man den TorMessenger mit folgenden Schritten:

1. Nach dem Download ist das Archiv zu entpacken: > tar -xaf tor-messenger-*

2. Danach kann man den TorMessenger in das Anwendungsmenü des Desktops in die Programmgruppe "*Internet*" einfügen, um den Start zu vereinfachen:
   > cd tor-messenger
   > ./start-tor-messenger.desktop --register-app
   Alternativ kann man den TorMessenger aus dem Dateimanager oder von der Konsole starten, indem man die Datei "*start-tor-messenger.desktop*" als Programm aufruft.
3. Der TorMessenger enthält zur Zeit nur ein englisches Wörterbuch für die Rechtschreibkontrolle. Man kann die installierten Hunspell-Wörterbücher mit TorMessenger nutzen. Dafür kopiert man die Wörterbücher in das Verzeichnis "*dictionaries*".

   >cd tor-messenger
   >cp-rf /usr/share/hunspell/*Messenger/dictionaries/

4. *Jabber Account einrichten:* Beim ersten Start erscheint der Assitent zum Einrichten eines Accounts. Später kann ein Account unter "*Tools - Accounts*" hinzugefügt werden.

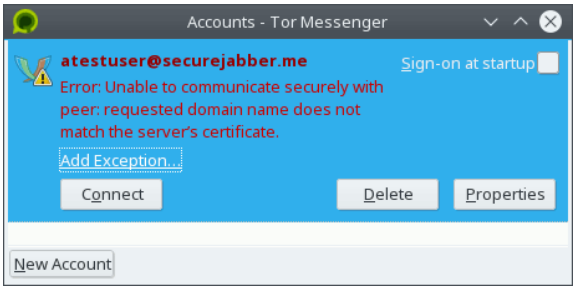



   1. Als erstes wählt man das XMPP-Protokoll.
   2. Im nächsten Schritt gibt man den Usernamen und die Domain ein. Wenn der Account bereit auf dem Server vorhanden ist, deaktiviert man die Option zum Erstellen des Account auf dem Server.

      Außerdem bietet der TorMessenger die Option, einen Wegwerf-Account zu erstellen. Wenn man diese Option aktiviert, dann ein Account auf dem Server *jabber.otr.im* angelegt. Es wird ein zufällig generiertes Passwort verwendet und ein OTR-Schlüssel für die Ende-zu-Ende Verschlüsselung erzeugt.

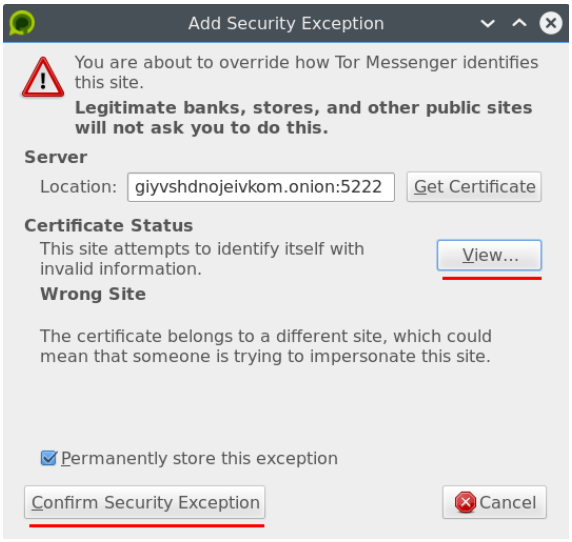
      Den Account kann man einfach wieder löschen, aufgrund eines Bug wird aktuell der OTR-Schlüssel für den Account nicht gelöscht und bleibt auf der Festplatte und muss per Hand in "*tor-messenger/Browser/TorBrowser/Data/Browser/profil.default/otr.priavte_key*" gelöscht werden.

   3. Im Idealfall tragen wir nun den Hidden-Server (also ein Server “im” TorNetzwerk) unserer Anbieterin ein. Damit vermeiden wir, dass unser Datenverkehr an einem Ausgang des Tor-Netzes mitgelesen und im schlimmsten Fall von einem großen Geheimdienst (zu denen der BND mitunter einen guten Draht hat) statistisch ausgelesen werden könnte – um uns zu deanonymisieren. Diese findet ihr meist auf den Webseiten der Dienste. Bei Systemli.org ist das: *x5tno6mwkncu5m3h.onion.*




   4. Abschließend wird eine Zusammenfassung angezeigt und man kann die Option zum automatischen Verbinden beim Start aktivieren.

5. Wenn man die Hidden Services der XMPP-Server nutzt, bekommt man beim ersten Aufbau der Verbindung einen Fehler. Die SSL-Zertifikate sind in der Regel nicht für die Hidden Service Adressen mit der Domain (also Endung) *irgendwas.onion* gültig. Man muss auf den kleinen Link "*Add Exception...*" unter der Fehlermeldung klicken und kann dann das Zertifikat selbst verifizieren.




6. *Der nachfolgende Tipp kann wichtig sein, sollte euch aber nicht aufhalten falls es zu kompliziert wird*:

   Zur Prüfung des SSL-Zertifikates klickt man in dem folgenden Dialog auf den Button "*View*" und vergleicht dem Fingerprint des Zertifikates mit den Daten, die der Betreiber auf der Webseite veröffentlicht hat oder mit den Werten, die das IM-Repository für diesen Server ermittelt hat: *https://xmpp.net/list.php* (mit Strg + f suchen).
   Wenn der SHA1 bzw. SHA2 Fingerabdruck übereinstimmt, kann man die Ausnahme dauerhaft bestätigen.

**OTR-Verschlüsselung:**

TorMessenger unterstützt OTR für die Ende-zu-Ende Verschlüsselung. Um die OTR-Schlüssel zu verwalten, wählt man den Menüpunkt "*Tools - OTR Preferences*".

Standardmäßig wird für jeden Account auch ein OTR-Schlüssel beim Anlegen des Account erzeugt und es wird die OTR-Verschlüsselung erzwungen. Man muss diesen Menüpunkt eigentlich nur aufrufen, um den Fingerabdruck zu vergleichen.

**SSL/TLS-Verschlüsselung:**

*Der nachfolgende Tipp kann wichtig sein, in Zukunft wichtig werden, sollte euch aber nicht aufhalten falls es zu kompliziert wird. SSL/TLS sind Methoden zur verschlüsselten Kommunikation zwischen dem Client und dem Server (eine sog. Transportverschlüsselung). Diese ist zB. für die Übertragung der Account-Daten wichtig*:

Die SSL/TLS-Einstellungen des TorMessengers sind etwas lax. Man kann aber besser TLS-Verschlüsselung in den erweiterten Einstellungen erzwingen. Wenn man die Einstellungen unter "*Tools - Preferences*" öffnet, findet man in der Sektion "*Advanced*" auf dem Reiter "*General*" den Button für den "*Config Editor*".

Dort kann man folgende Werte setzen:

1. TLS 1.2 erzwingen: security.tls.version.min = 3
2. Alle Verschlüsselungsmethoden bis auf die (derzeit) als sicher Eingestuften deaktivieren:
   security.ssl3.ecdhe_rsa_aes_256_gcm_sha384 = true
   security.ssl3.ecdhe_ecdsa_aes_256_gcm_sha384 = true
   security.ssl3.ecdhe_rsa_chacha20_poly1305_sha256 = true
   security.ssl3.ecdhe_ecdsa_chacha20_poly1305_sha256 = true
   security.ssl3.ecdhe_rsa_aes_128_gcm_sha256 = true
   security.ssl3.ecdhe_ecdsa_aes_128_gcm_sha256 = true
   security.ssl3.* = false
3. Insecure Renegotiation verbieten:
   security.ssl.require_safe_negotiation = true
   security.ssl.treat_unsafe_negotiation_as_broken = true
4. OCSP abschalten: security.OCSP.enabled = 0
5. Strenges Certifikate Pinning erwingen (z.B. für Add-on Updates):
   security.cert_pinning.enforcement_level = 2

Wenn mit diesen Einstellungen keine Verbindung zum Jabber-Server mehr möglich ist, dann sollte man sich einen Account auf einem anderen Server erstellen.

## Gajim

*Diese Anleitung ist eine leicht abgeänderte Kopie dieser Webseite:*
*https://privacy-handbuch.de/handbuch_63.htm*

*Installation von Gajim:*

- Windows Nutzer laden die Installationsdatei von der Download Webseite herunter und starten die Setup-Datei als Administrator.
- Für Debian und abgeleitete Linux-Derivate sind folgende Pakete zu installieren:

  > sudo apt install gajim python-axolotl python-protobuf python-potr python-qrcode
  > sudo apt install kwalletcli aspell-de libgtkspell0 python-openssl

  Die Pakete *python-axolotl* und *python-protobuf* werden für das OMEMO-Plugin benötigt. *kwalletcli* ist optional und ermöglicht es, die Zugangsdaten verschlüsselt in der Passwortverwaltung KWallet zu speichern. *aspell* und die Wörterbücher aus *aspell-de* werden für die Rechtschreibprüfung verwendet.

*Installation der Plugins:*

Nach der Installation startet man Gajim, überspringt den Assistenten zur Accounteinrichtung und öffnet die Plugin-Verwaltung unter "*Ändern – Plugins*".

Auf dem Reiter der verfügbaren Plugins wählt man das OTR- und OMEMO-Plugin sowie HTTPUpload + URLImagePreview und klickt auf den Install-Button.

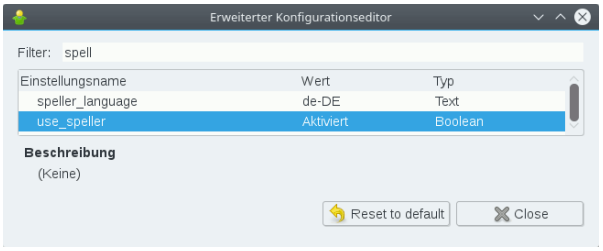
Nach der Installation muss man die Plugins noch aktivieren. Dafür wechselt man zum Reiter "*Installiert*" und aktiviert die frisch installierten Plugins.

### Konfiguration von Gajim

Bevor man einen Account erstellt, kann man noch einige Anpassungen der Konfiguration vornehmen. Dazu öffnet man den Konfigurationsdialog (Menüpunkt: "*Ändern - Preferences*") und klickt sich einmal durch die Einstellungen. Auf dem Reiter "*Erweitert*" kann man die bevorzugten Programme konfigurieren und Einstellungen zur Privatsphäre vornehmen:

Wichtig ist, die Aufzeichnung von verschlüsselten Chats zu deaktivieren.

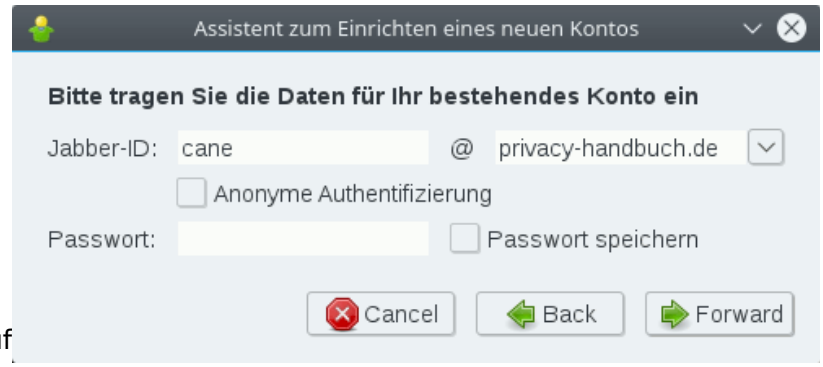
Wenn man den "*Erweiterten Konfigurationseditor*" öffnet, kann man die Rechtschreibprüfung aktivieren. Dazu trägt man die gewünschte Sprache ein (Deutsch: "de_DE")
und aktiviert die Rechtschreibprüfung,
wie es im Screenshot links zu sehen ist.




Gajim verwendet das Programm "*aspell*" für die Rechtschreibprüfung. Die Wörterbücher für die gewünschte Sprache und die "*libgtkspell0*" müssen ebenfalls installiert sein.

*Account erstellen:*

1. Um einen Jabber/XMPP Account einzurichten öffnet man die Account - verwaltung (Menüpunkt: *Ändern - Konten*) und klickt auf den Button *Hinzufügen*.
   Es öffnet sich der Assistent zur Konfiguration neuer Konten. Wenn man bereits einen Account hat, die Ist Konfiguration einfach (siehe links).




2. Wenn man einen neuen Account auf einem Jabber Server erstellen möchte, wählt man zuerst den Server. Das Drop-Down Menü enthält eine Liste von Jabber Servern. Danach wählt man den Namen und das Passwort für den neuen Account.

Einige Server (zB. *jabber.ccc.de*) verlangen die Lösung für ein Captcha, um Robots (Programmen) die Anmeldung zu erschwerden. Damit sollen bestimmte Angriffe verhindert werden. Mit Klick auch Forward wird der neue Account auf dem Server erstellt.

*Account konfigurieren:*

In den erweiterten Einstellungen des Account kann man nach dem Erstellen des Account noch einige Kleinigkeiten anpassen.

Gajim möchte z.B. wieder alle Unterhaltungen protokollieren (das betrifft auch verschlüsselte Konversationen). Außerdem kann man das Verfolgen von Konversationen auf anderen Geräten deaktivieren, da es nicht mit der OTR-Verschlüsselung kompatibel ist, man sieht dann nur unlesbaren OTR-Kauderwelsch. Mit PGP und OMEMO gibt es keine Probleme.

*Der nachfolgende Tipp kann wichtig sein, sollte euch aber nicht aufhalten falls es zu kompliziert wird*:

Außerdem kann man in dem Konfigurationseditor die Einstellungen für die SSL/TLS-Verschlüsselung anpassen. Standardmäßig verwendet Gajim TLS v1.0+ und und ansonsten nur schwache Verschlüsselungsmethoden, die nicht mehr mit den aktuellen Empfehlungen kompatibel ist. Für jeden Account kann man folgende Werte anpassen:

tls_version = 1.2
cipher_list = ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384

Welche Verschlüsselungsmethoden der genutzte Server unterstützt, kann mit dem CryptCheck geprüft werden, indem folgende URL im Browser aufgerufen wird:

*https://tls.imirhil.fr/xmpp/<domain.tld>*

Welche Verschlüsselungsmethoden Gajim unterstützt, kann zumindest unter Linux mit folgendem Kommando geprüft werden:

> openssl ciphers -v

### Gajim + Tor Onion Router

Gajim ist unserer Meinung nach nicht für die Kombination mit Tor Onion Router geeignet. Es ist eine Proxy Konfiguration für Tor vorbereitet, aber Gajim enthält Bugs, welche die Anonymität und Sicherheit bei der Verwendung von Tor gefährden.

## Outro

Bitte **verbreitet den Reader** weiter und ermutigt eure Umfeld zur selbstbestimmt(er)en Kommuniktion mit Jabber/XMPP.

Über Rückfragen (auch zu anderen Themen) und Kritik freuen wir uns!

Kontakt:
*rc4@jabber.systemli.org* [OTR-Fingerprint: *619CCB09 D120E17E 84CEBE8B BCC51AAE 5E17FB68*]

**Noch ein paar weiterführende Links:**

*https://conversations.im/compliance/* - Vergleich von XMPP-Servern (*nicht aktuell*)

*https://status.jabber.cat/* - Erreichbarkeit einiger Jabber-Server

*https://www.privacytools.io/* - bieten einen guten Überblick.

*https://privacy-handbuch.de/* - technisch sehr detailierte Anleitung zu digitaler Privatspähre

*https://ruptur3.github.io/toolkit/#* - sehr viele nüztliche Links. Von Linux Mint ist abzuraten, lieber Ubuntu nutzen.

*https://riseup.net/en/security* – guter Einstieg zu digitaler Sicherheit

*https://riseup.net/en/security/network-security/tor/#riseups-tor-onion-services* – nüztliche Hidden Services (also direkt im Tor-Netzwerk erreichbar)

*https://www.systemli.org/en/service/* - alternative nützliche Dienste

*https://www.heise.de/ix/heft/Hinter-Schichten-2268444.html* – eine kurze Einführung zum Tor-Netzwerk

Und off topic:

*https://capulcu.blackblogs.org/neue-texte/band-iii/* - Broschüre zum digitalen Angriff

*Autonome Grüppchen*