

*Dieser Text und seine Abbildungen sind quelloffen und können von jedem urheberrechtsfrei genutzt oder weiter veröffentlicht werden.
Stand: Oktober 27 / 2014*

GoldBug - Secure Instant Messenger

Deutsches Benutzer-Handbuch

des sicheren Sofort-Nachrichten-Programms GoldBug

Abbildung 1: GoldBug-Logo-Claim



Secure Instant Messenger

<http://goldbug.sf.net>

Was ist GoldBug?

GoldBug ist ein sicherer Instant Messenger.

Mit der Nutzung von GoldBug (GB) kann Du sicher sein, dass kein unerwünschter Dritter Deine Gespräche belauschen kann. Private Nutzer-zu-Nutzer Kommunikation verbleibt im privaten, geschützten Raum.

Dafür nutzt GoldBug starke Vielfach-Verschlüsselung, auch hybride Verschlüsselung genannt, mit verschiedenen Ebenen von moderner Verschlüsselungs-Technologie von etablierten Verschlüsselungs-Bibliotheken - wie libgcrypt (bekannt von GnuPG) und OpenSSL.

Zum Beispiel werden damit mehr als 8 öffentlich/private Schlüssel zur Verschlüsselung erstellt - basierend auf den Verschlüsselungsalgorithmen RSA, oder wahlweise auch ElGamal und NTRU. Weiterhin bietet die Applikation auch dezentrales und verschlüsseltes E-Mail und auch dezentralen öffentlichen E*IRC-Gruppen-Chat an.

Wie in jedem Nachrichtenprogramm können auch Dateien geteilt und versandt werden. Mit den Werkzeugen "Rosetta CryptoPad" und dem "File-Encryptor" kannst Du auch Text und/oder Dateien sicher verschlüsseln.

Warum ist es wichtig, dass Du Deine Kommunikation verschlüsselst?

Abbildung 2: Oliver Stone Portrait



"The question is not 'do you have something to hide?'"

The question is whether we control
or others controls us." - Oliver Stone

<https://www.youtube.com/watch?v=0U37hl0n9mY>

Heutzutage sind fast alle kabellosen WIFI-Internetnetze mit einem Passwort geschützt.

In wenigen Jahren werden Klartext-Nachrichten oder E-Emails an Freunde (im Folgenden immer w/m gemeint) über das Internet ebenso verschlüsselt sein.

Das ist keine Frage, ob man etwas zu verbergen hat oder nicht, es ist die Frage, ob wir selbst unsere Kommunikation kontrollieren - oder sie durch andere, dritte kontrolliert wird.

Es ist letztlich eine Frage des Angriffs auf das freien Denken und Streichung der Annahme einer Unschuldsvermutung ("Im Zweifel für den Angeklagten" - wenn jeder Bürger überhaupt auf eine Anklagebank gehört).

Demokratie erfordert das Denken und die Diskussion von Alternativen im Privaten wie auch in der Öffentlichkeit.

Starke Multi-Verschlüsselung (sogenannte "hybride Verschlüsselung") sichert die Erklärungen der Menschenrechte in breit konstituiertem Konsens und ist eine digitale Selbstverteidigung, die jeder erlernen und nutzen sollte.

Der GoldBug Messenger bemüht sich, ein einfach zu nutzendes Werkzeug für diesen Anspruch zu sein.

Ähnlich der Sicherheitsentwicklung beim Automobil wird sich auch die E-Mail-Verschlüsselung entwickeln: ist man beim Automobil zunächst ohne Sicherheitsgurt gefahren, fahren wir heute mit Sicherheitsgurten und zusätzlich Airbags oder drittens noch ergänzenden Sicherheits-Informationssystemen. Die unverschlüsselte Plain-Text-Email oder Chat-Nachricht hat ausgedient.

Woher kommt der Name "GoldBug"?

Der Gold-Käfer ("[The GoldBug](#)") ist eine Kurzgeschichte von Edgar Allan Poe: "In der Handlung geht es um William LeGrand, der kürzlich einen gold-farbenen Marienkäfer entdeckte. Sein Kumpel, Jupiter, fürchtet nun, dass LeGrand besessen wird in seiner Suche nach Reichtum, Erkenntnis und Weisheit, nachdem er mit dem Goldkäfer in Kontakt gewesen ist - und geht daher zu einem weiteren Freund von LeGrand, ein nicht mit Namen benannter Erzähler, der zustimmt, seinen alten Freund zu besuchen.

Nachdem LeGrand sodann auf eine Geheime Botschaft gestoßen ist und diese erfolgreich entschlüsseln konnte, starten die drei ein Abenteuer als Team.

Der Gold-Käfer - als eine der wenigen Stücke in der Literatur - integriert Verschlüsselungstexte als Element der Geschichte. Poe war damit der Popularität von Verschlüsselungstexten seiner Zeit weit voraus als er im Jahre 1843 "The Gold-Bug" schrieb, in dem der Erfolg der Geschichte sich z.B. um solche ein solches Kryptogramm und

metaphorisch um die Suche nach dem Stein der Weisen drehte.

Der Gold-Käfer war ein sofortig eine viel gelesene Geschichte und war äußerst populär und von den Literaten das meist untersuchte Werk von Poe während seiner Lebenszeit. Seine Ideen halfen ebenso das Schreiben verschlüsselter Texte und so genannter Kryptogramme bekannt zu machen" (vgl. auch engl. Wikipedia).

170 Jahre später hat Verschlüsselung mehr Gewicht denn je. Verschlüsselung sollte ein Standard sein, wenn wir Kommunikation über das unsichere Internet senden.

GoldBug hat Alternativen zu RSA

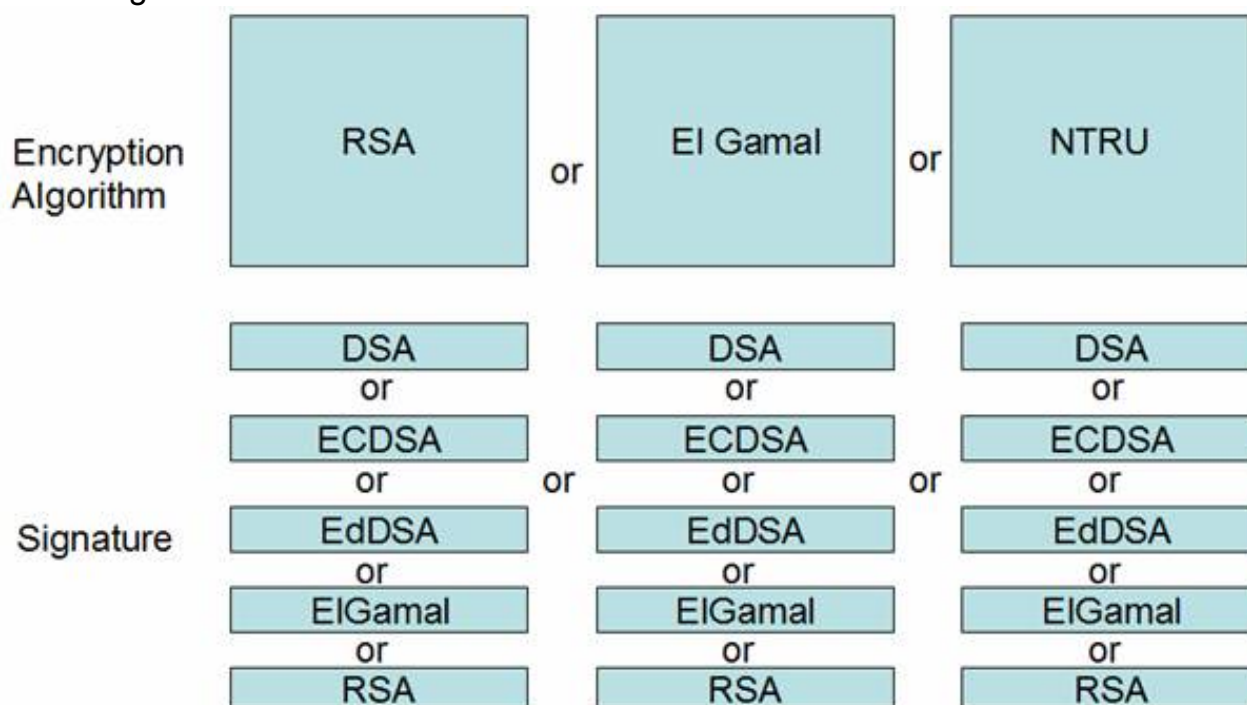
GoldBug Messenger hat verschiedene Alternativen zu RSA, falls dieser Verschlüsselungs-Algorithmus einmal unsicher würde (z.B. durch Quantum Computer). Bislang gilt RSA aber mit entsprechend großer Größe des Schlüssels weiterhin als sicher. Neben RSA hat GoldBug die Verschlüsselungsalgorithmen ElGamal und auch NTRU implementiert.

Bei den bei Verschlüsselung ebenso optional verfügbaren Signaturverfahren besteht sogar noch eine größere Auswahl: DSA, ECDSA, EdDSA, ElGamal und RSA.

Natürlich kann jeder Nutzer seine individuelle Schlüsselgröße einstellen, die "Cipher", den "Hashtype", ferner "Iteration Count", und die Salz-Länge ("Salt Length") - für die Erstellung der Schlüssel für die Verschlüsselung oftmals typische Kriterien.

Der Vorteil ist, dass jeder Nutzer dieses individuell für sich definieren kann.

Abbildung 3: Alternativen zu RSA



RSA - ElGamal und NTRU im Vergleich

[NTRU](#) ist ein asymmetrisches Verschlüsselungsverfahren, das 1996 von den Mathematikern Jeffrey Hoffstein, Jill Pipher und Joseph Silverman entwickelt wurde. Es basiert lose auf Gitterproblemen, die selbst mit Quantenrechnern als nicht knackbar gelten. Allerdings ist NTRUEncrypt bisher nicht so gut untersucht wie gebräuchlichere Verfahren (z.B. RSA). NTRUEncrypt ist durch IEEE P1363.1 standardisiert (vgl.

<https://de.wikipedia.org/wiki/NTRUEncrypt> sowie <https://en.wikipedia.org/wiki/NTRU>).

RSA (nach den Personen Rivest, Shamir und Adleman) ist ein asymmetrisches kryptographisches Verfahren, das sowohl zur Verschlüsselung als auch zur digitalen Signatur verwendet werden kann. Es verwendet ein Schlüsselpaar, bestehend aus einem privaten Schlüssel, der zum Entschlüsseln oder Signieren von Daten verwendet wird, und einem öffentlichen Schlüssel, mit dem man verschlüsselt oder Signaturen prüft. Der private Schlüssel wird geheim gehalten und kann nur mit extrem hohem Aufwand aus dem öffentlichen Schlüssel berechnet werden. (vgl. <https://de.wikipedia.org/wiki/RSA-Kryptosystem> sowie [https://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](https://en.wikipedia.org/wiki/RSA_(cryptosystem))).

Das **ElGamal**-Verschlüsselungs-Verfahren oder ElGamal-Kryptosystem ist ein im Jahr 1985 vom Kryptologen Taher Elgamal entwickeltes Public-Key-Verschlüsselungsverfahren, das auf der Idee des Diffie-Hellman-Schlüsselaustauschs aufbaut. Das ElGamal-Verschlüsselungsverfahren beruht, wie auch das Diffie-Hellman-Protokoll, auf Operationen in einer zyklischen Gruppe endlicher Ordnung. Das ElGamal-Verschlüsselungs-Verfahren ist beweisbar IND-CPA-sicher unter der Annahme, dass das Decisional-Diffie-Hellman-Problem in der zugrundeliegenden Gruppe nicht trivial ist. Verwandt mit dem hier beschriebenen Verschlüsselungsverfahren (aber nicht mit diesem identisch) ist das Elgamal-Signaturverfahren (Das ElGamal-Signatur-Verfahren ist in GoldBug nicht implementiert). ElGamal unterliegt keinem Patent. (vgl. https://en.wikipedia.org/wiki/ElGamal_encryption sowie <https://de.wikipedia.org/wiki/Elgamal-Verschl%C3%BCsselungsverfahren>).

Was ist das Echo Protokoll?

Mit dem Echo Protokoll ist - einfach ausgedrückt - gemeint, dass

- **jede Nachrichten-Übertragung verschlüsselt ist...**

Beispiel:

SSL (AES (RSA* (Nachricht)))

*) anstelle von RSA kann ebenso ElGamal oder NTRU genutzt werden,

- **... und im Echo-Netzwerk sendet jeder Verbindungsknoten jede Nachricht an jeden verbundenen Nachbarn.**

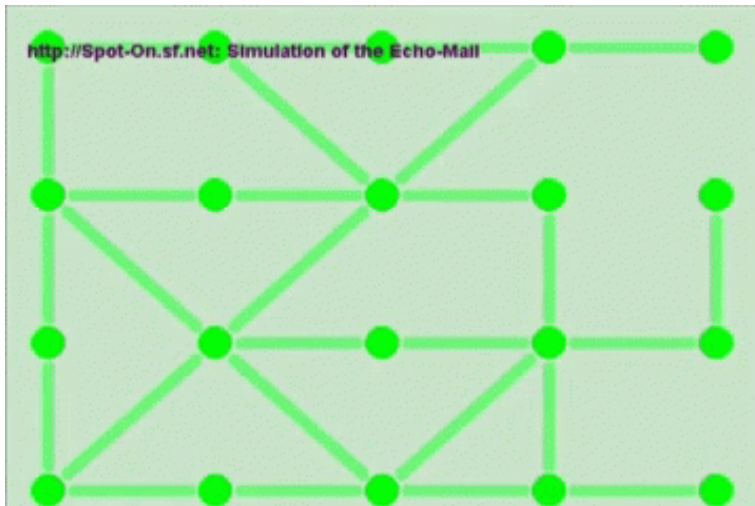
Zugrunde liegt das sogenannte "**Kleine-Welt-Phänomen**": Jeder kann jeden über sieben Ecken in einem peer-to-peer oder friend-to-friend Netzwerk irgendwie erreichen - oder aber einfach über einen im Freundeskreis installierten gemeinsamen Echo-Chat-Server.

- Der **Modus des "Halben Echos"** sendet eine Nachricht nur einen Hop, d.h. z.B. von Bob zu Alice. Alice sendet die Nachricht dann nicht mehr weiter (wie es beim Vollen Echo der Standard ist).
- Neben Vollem Echo, Halben Echo gibt es drittens noch das **Adaptive Echo** (AE). Hier wird die Nachricht nur an Nachbarn oder Freunde versandt, wenn diese einen Verschlüsselungs-Token kennen, also zuvor gespeichert haben. Wer den Token nicht kennt, an den wird die Nachricht nicht weitergeleitet.

- Schließlich kennt das Echo noch **Echo Accounts**. Eine Art Firewall. Hiermit kann sichergestellt werden, dass nur Freunde, die den Account-Zugang kennen, sich verbinden können. So kann ein Web-of-Trust erstellt werden, also ein Netzwerk ausschließlich unter Freunden. Es basiert nicht auf dem Schlüssel für die Verschlüsselung, sondern ist davon unabhängig. D.h. Du musst nicht Deinen öffentlichen Schlüssel auch noch mit Deiner IP-Adresse verknüpfen oder gar im Netzwerk bekannt geben.

Grundsätzlich sendet im Echo jeder Knoten die Nachricht an jeden Knoten

Abbildung 4: Simulation des Echo-Netzwerkes



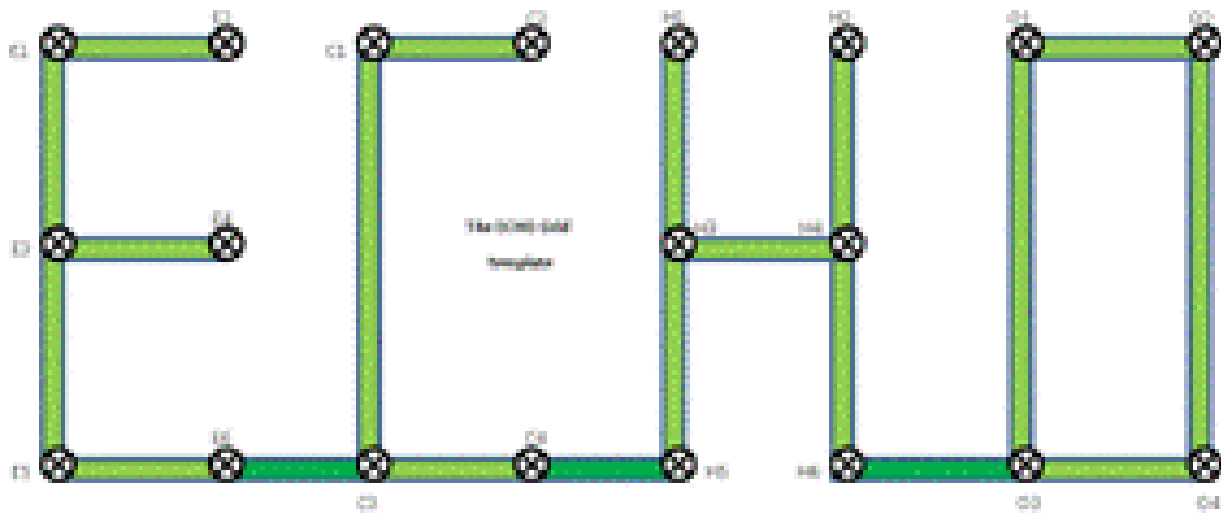
Wenn Du eine Nachricht ein zweites Mal erhalten solltest, so wird sie in einem temporären Zwischenspeicher verglichen (anhand des Hashwertes für diese Nachricht) und verworfen ("Congestion Control").

Schließlich: man kann ebenso mit der GoldBug Applikation unechte Nachrichten ("Fakes") und simuliert Kommunikationsnachrichten ("Impersonated Messages") aussenden. Einmal ist die Verschlüsselung keine Verschlüsselung, sondern stellt pure Zufallszeichen dar, die von Zeit zu Zeit ausgesandt werden, und das andere Mal wird eine menschliche Unterhaltung simuliert, die ebenso nur auf durcheinander-gewürfelte Zufallszeichen beruht. So kann die Analyse von Nachrichten erschwert werden, wenn dritte Aufzeichner ("Recorder") Deine sämtliche Kommunikation zwischen-speichern und aufzeichnen sollten, was ggf. anzunehmen ist.

Das ECHO-Grid

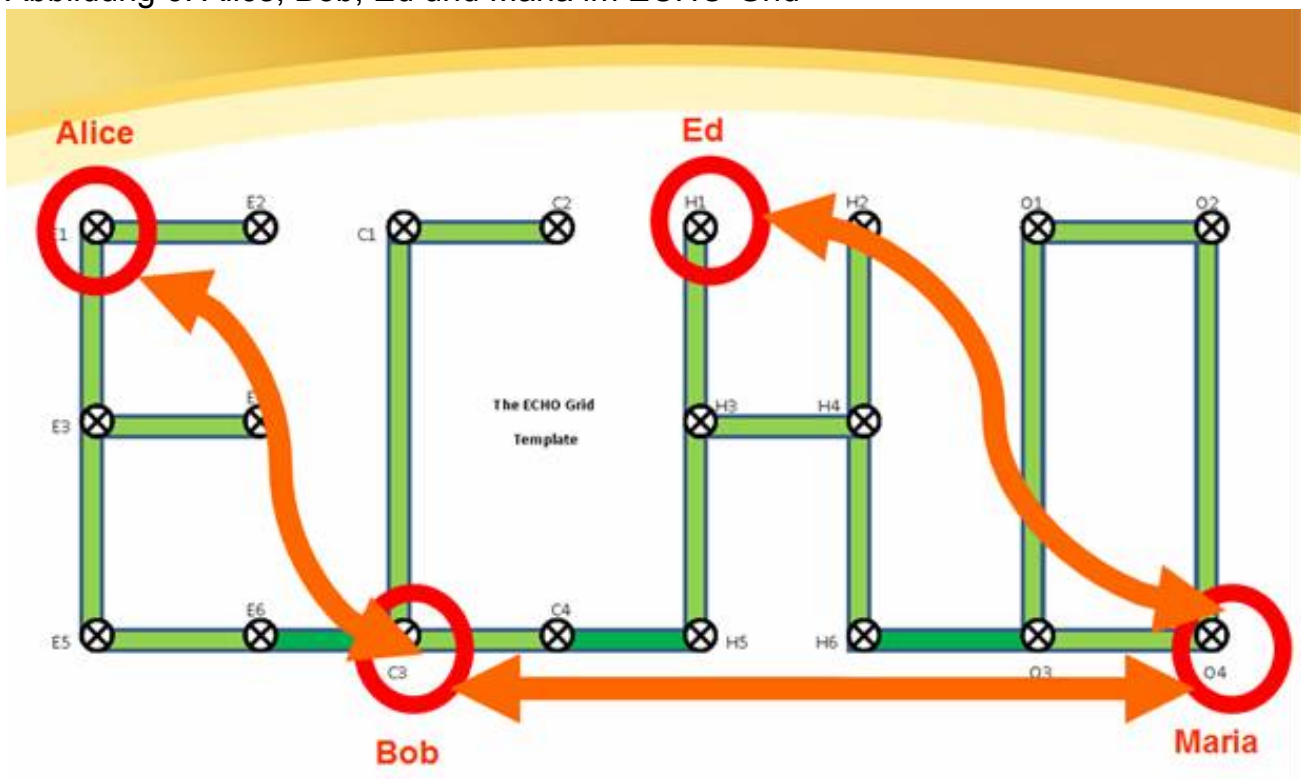
Wenn Studierende über das Echo-Protokoll reden und unterrichten, dann zeichnen wir einfach ein ECHO-Grid mit den Buchstaben E-C-H-O und nummerieren die Knotenpunkte von E1 bis O4 und verbinden die Buchstaben mit einer Verbindungslinie am Boden. Beispielsweise bezeichnet dann die Verbindung E1-E2 eine IP-Verbindung zu einem Nachbarn.

Abbildung 5: Das ECHO-Grid



Wenn die einzelnen Kontenpunkte nun Schlüssel tauschen, so entstehen Verbindungen, die als neue Ebene auf der Ebene der IP-Verbindungen des P2P/F2F-Netzwerkes entstehen.

Abbildung 6: Alice, Bob, Ed und Maria im ECHO-Grid



Beispiele des Schlüssel-Austausches von Alice, Bob, Ed und Maria.

- Alice (IP=E1) und Bob (IP=C3) haben ihren öffentlichen Schlüssel getauscht und sind über die folgenden IP-Nachbarn verbunden: E1-E3-E5-E6-C3.
- Bob (C3) und Maria (O4) sind ebenso Freunde, sie haben ihre öffentlichen Schlüssel

für die Verschlüsselung ebenso getauscht: und nutzen die IP-Verbindungen der Nachbarn: C3-C4-H5-H3-H4-H6-O3-O4.

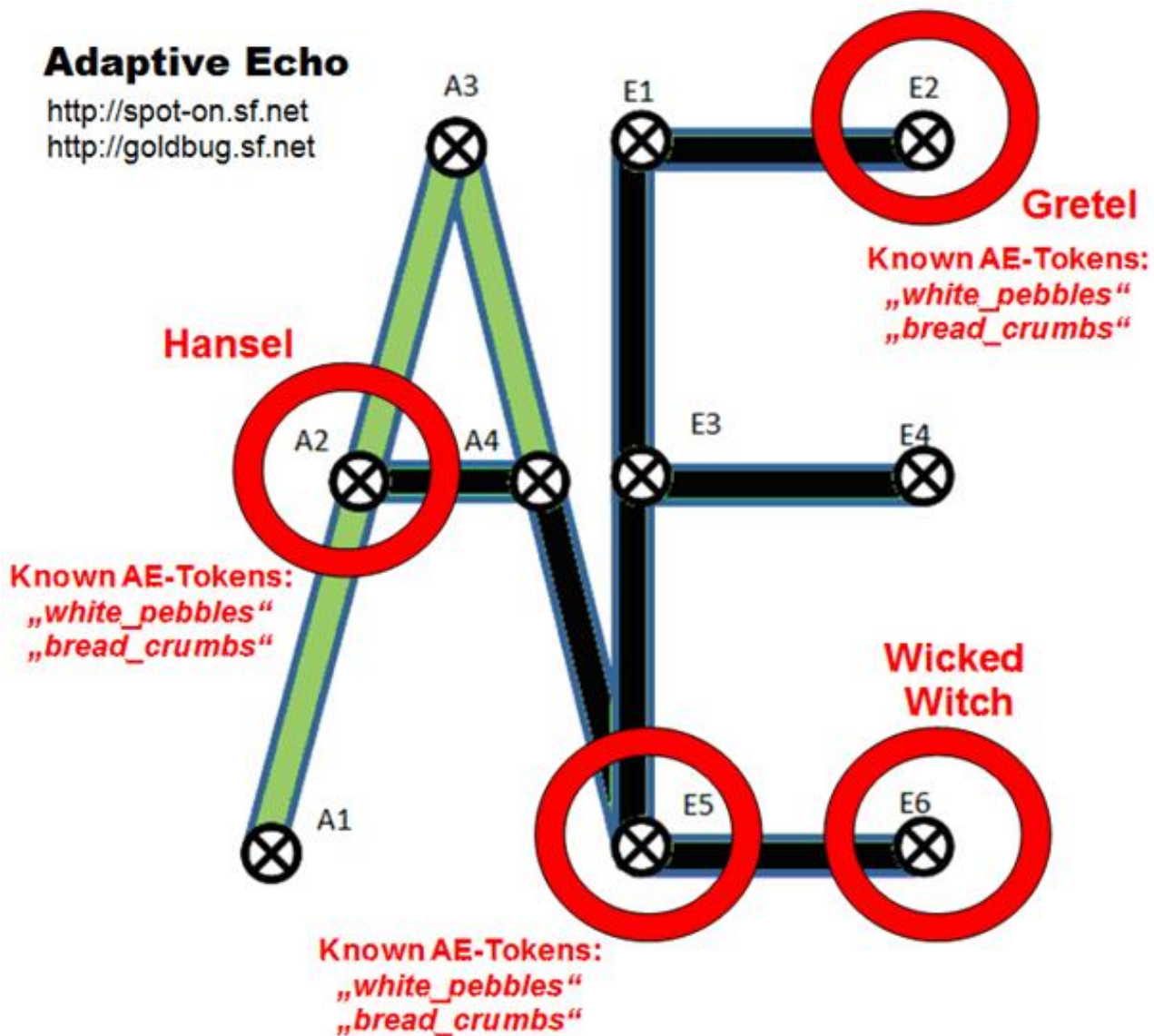
- Schließlich, Maria (O4) ist eine Freundin von Ed (H1). Sie kommunizieren entweder über den Weg: O4-O3-H6-H4-H3-H1 oder sie nutzen den Pfad von: O4-O2-O1-O3-H6-H4-H3-H1. Da im Echo Protokoll ja jeder IP-Nachbar jede Nachricht an jeden verbundenen IP-Nachbarn sendet, wird der Pfad erfolgreich sein, der die Nachricht am schnellsten übermittelt.
- Direkte IP-Verbindungen von Nachbarn wie z.B. E1-E3 können durch die Erstellung eines sog. "Echo-Accounts" abgesichert werden: Keine andere IP-Adresse als E1 kann zu dem sogenannten "Listener" des Nachbarn E3 verbinden. Über diese Methode kann ein Web-of-Trust erstellt werden - ohne von Schlüsseln zur Verschlüsselung abhängig zu sein - noch brauchst Du einen Freund, mit dem Du Deinen Chat oder E-Mail-Schlüssel tauschst.
- Sogenanntes „Turtle hopping“ wird im Echo-Netzwerk wesentlich effizienter: Wenn Ed und Alice einen sogenannten "Starbeam-Magneten" zur Dateiübertragung tauschen, dann transportiert das Echo Protokoll die Pakete über den Weg H1-H3-H5-C4-C3-E6-E5-E3-E1. Maria ist nicht in der Route, aber sie wird die Pakete ebenso über das volle Echo erhalten, wenn sie den StarBeam-Magnet kennt. Vorteil ist, dass das Hopping nicht über die Schlüssel geht, sondern über die IP-Verbindungen (z.B. das Web-of-Trust). Grundsätzlich ist alles immer verschlüsselt, also warum nicht den kürzesten Weg nehmen?
- Ein sogenannter "Buzz" bzw. "ge-echo-ter IRC Channel" (e*IRC) - Raum kann z.B. durch den Knotenpunkte O2 erstellt oder "gehostet" werden. Da nur der Nutzer Ed den Buzz Raum Namen kennt, bleiben alle anderen Nachbarn und Freunde außen vor. Vorteil: Du kannst mit unbekannten Freunden in einem Raum sprechen, ohne mit diesen einen öffentlichen z.B. RSA - Schlüssel jemals getauscht zu haben. Stattdessen nutzt Du einfach einen Einmal-Magnet ("one-time-magnet") für einen "buzz" / "e*IRC" Raum.
- Maria ist eine gemeinsame Freundin von Ed und Bob und sie aktiviert die C/O (care of)-Funktion für Emails: Das erlaubt Ed, E-Mails an Bob zu senden, obwohl er offline ist, denn: Maria speichert die E-Mails zwischen, bis Bob dann online kommt.
- Weiterhin: Alice erstellte eine sogenannte virtuelle "E-Mail Institution". Das ist nicht vergleichbar mit einem POP3 oder IMAP Server, da die E-Mails nur zwischengespeichert werden: Ed sendet dazu seinen öffentlichen E-Mail-Schlüssel an Alice - und Ed fügt den Magneten der "Email Institution" von Alice bei sich in seinem Programm ein. Nun werden auch die E-Mails von Bob and Ed bei Alice zwischengespeichert (in der E-Mail-Institution), selbst wenn Maria offline sein sollte.

Es ist hilfreich, die Beispiele auf obiger Grafik nachzuvollziehen.

Adaptives Echo (AE) und seine AE-Tokens

Für die Erklärung des "Adaptiven Echos" kann ein weiteres Echo-Grid mit den verbundenen Buchstaben A und E gezeichnet werden.

Abbildung 7: Das „Hänsel und Gretel“ - Beispiel des Adaptiven Echos



Wenn Du, Dein Chat-Freund und ein eingerichteter dritter Kontenpunkt als Chat-Server denselben AE-Token ("Adaptive-Echo Token") in das Programm einfügen, dann wird der Chat-Server Deine Nachricht nur zu Deinem Freund senden - und nicht zu allen anderen verbundenen Nachbarn oder Nutzern wie es normalerweise bei dem Vollen Echo Modus der Fall wäre. Mit einem AE-Token wird kein anderer Deine Nachricht erhalten oder einsehen können. Damit können potentielle "Recorder" ausgenommen werden, also mögliche Nachbarn, die möglicher- und anzunehmender Weise den gesamten Nachrichtenverkehr aufzeichnen und sodann versuchen wollen, die mehrfache Verschlüsselung aufzubrechen, um an den Nachrichten-Kern zu kommen.

Hänsel und Gretel – ein Beispiel für den Adaptiven Echo Modus:

Wenn Knotenpunkt A2, E5 und E2 denselben AE-Token einsetzen, dann wird Kontenpunkt E6 keine Nachricht erhalten, die der Knotenpunkt A2 (Hänsel) und der Knotenpunkt E2 (Gretel) austauschen werden. Denn, der Knotenpunkt E5 lernt über den bekannten Token "Weisse Kieselsteine" ("white_pebbles"), die Nachrichten nicht an den Kontenpunkt E6, die "Böse Hexe" ("Wicked Witch"), zu senden. Ein lernendes, sich anpassendes ("adaptives") Netzwerk.

Ein "Adaptives Echo" Netzwerk enthüllt dabei keine Ziel-Informationen (vergleiche auch "[Ants Routing](#)"). Denn - zur Erinnerung: Der Modus des "Halben Echos" sendet nur einen Hop zum verbundenen Nachbarn und das "Volle Echo" sendet die verschlüsselte Nachricht zu allen verbundenen Knotenpunkten über eine nicht weiter spezifizierte Hop-Anzahl.

Während "Echo Accounts" andere Nutzer quasi als Firewall oder Berechtigungskonzept beim Verbinden fördern oder behindern, halten hingegen "AE-Tokens" Graphen- oder Pfad-Exklusivität vor - und zwar für Nachrichten, die über Verbindungsknoten gesandt werden, die den AE-Token kennen.

Chat-Server Administratoren können ihre Token mit anderen Server-Administratoren tauschen, wenn Sie sich untereinander vertrauen (sogenanntes "Ultra-Peering for Trust") und ein Web-of-Trust definieren.

In Netzwerk-Laboren oder zuhause mit drei, vier Rechnern kann man das das Adaptive Echo einfach austesten und seine Ergebnisse dokumentieren:

Nutze "SPOTON_HOME" als Datei im Binärverzeichnis, um mehrere Programminstanzen auf einer einzigen Maschine zu launchen und zu verbinden - oder nutze einfach ein Netzwerk mit drei oder mehreren Computern. So dann folge diesem Ablauf:

1. Erstelle einen Knotenpunkt als Chat Server.
2. Erstelle zwei Knotenpunkte als Klient.
3. Verbinde die beiden Klienten zum Chat Server.
4. Tausche Schlüssel zwischen den Klienten.
5. Teste die normale Kommunikationsfähigkeit beider Klienten.
6. Setze einen AE-Token auf dem Server.
7. Teste die normale Kommunikationsfähigkeit beider Klienten.
8. Setze denselben AE-Token nun auch in einem Klienten.
9. Notiere das Ergebnis: Der Server-Knotenpunkt sendet die Nachricht nicht mehr an andere Knotenpunkte aus, die den AE-Token nicht haben bzw. kennen.

Dieses Beispiel sollte einfach replizierbar sein.

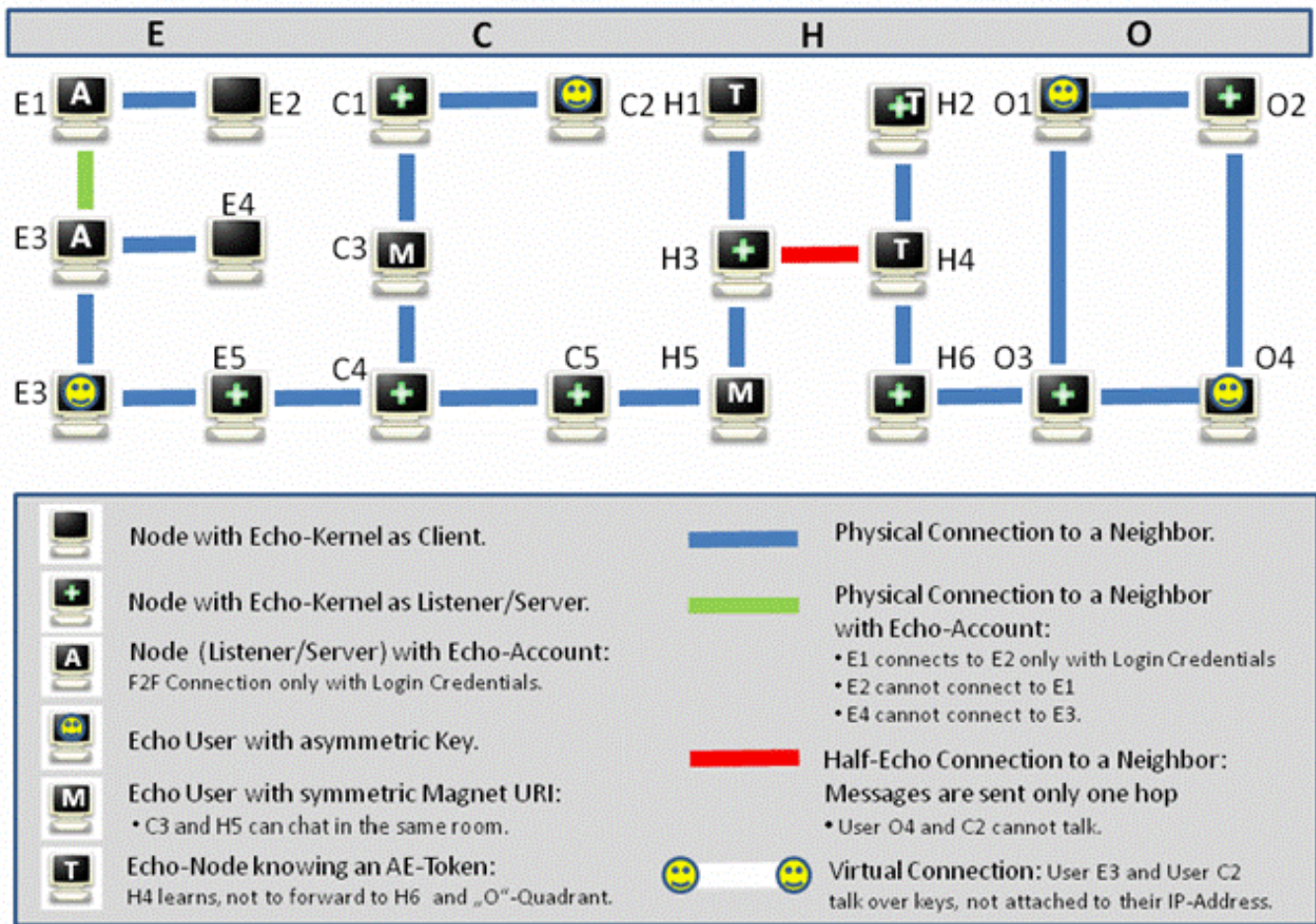
Wie das ECHO Protokoll funktioniert

Nimmt man nun die verschiedenen Methoden und Optionen zusammen, kann folgende Graphik einen komplexeren Überblick bieten.

Abbildung 8: Wie das ECHO PROTOCOL funktioniert

How the ECHO PROTOCOL works

Full Echo | Half Echo | Adaptive Echo (AE) | Echo Accounts



In der Graphik abgebildet sind die unterschiedlichen Nutzungsbeispiele von "Full Echo", "Half Echo", "Adaptive Echo" sowie "Echo Accounts".

Unterschieden wird zwischen physischen IP-Verbindungen und virtuellen Verbindungen zu Schlüsseln. Schlüssel sind daher nicht zwingend einer IP-Verbindung zugeordnet.

Nutzer können darin asymmetrische öffentliche Schlüssel, aber auch Magnet-URIs mit symmetrischen Verschlüsselungsdetails sowie Tokens und Account-Credentials tauschen.

Verbindungsknoten können Verbindungen erlauben und verbieten ebenso wie Nachrichten dediziert adressiert oder auslassend adressiert senden.

Dementsprechend entstehen unterschiedliche Kommunikations-Szenarien.

Beispiele:

- Nutzer H4 hat einen AE-Token. Er sendet keine Nachrichten (über den Verbindungsknoten H6) in den O-Quadranten, wenn H4 den Token nicht kennt.
- Wenn H3 eine Nachricht an H4 sendet, dann sendet H4 diese Nachricht ebenso nicht weiter, da es sich um eine Verbindung des „Halben Echos“ handelt.
- Der Nutzer E1 lässt den Nutzer E2 nicht verbinden, da er das Login für den Echo-Account nicht kennt.

- d. Nutzer O1 und O4 chatten miteinander und kennen nur ihren öffentlichen Schlüssel für die Verschlüsselung.
- e. Nutzer H3 und C5 chatten über einen URI-Magneten im gleichen Gruppen-Chat-Raum (auch Buzz oder e*IRC genannt).

Screenshot: Passwort-Definition, Schlüssel-Erstellung und Kernel-Aktivierung

Der GoldBug Messenger hat ein Interface und einen Kernel. Beide sind als Binärdatei gegeben (also unter Windows als GoldBug.exe und Kernel.exe bezeichnet).

Mit der Benutzeroberfläche (Interface oder auch GUI genannt (Graphical User Interface, GUI = GoldBug.exe)) muss vor jedem Start der Kernel aktiviert werden, der dann die direkten Verbindungen zu Freunden oder über einen gemeinsamen Chat-Server bzw. das Echo-Netzwerk koordiniert.

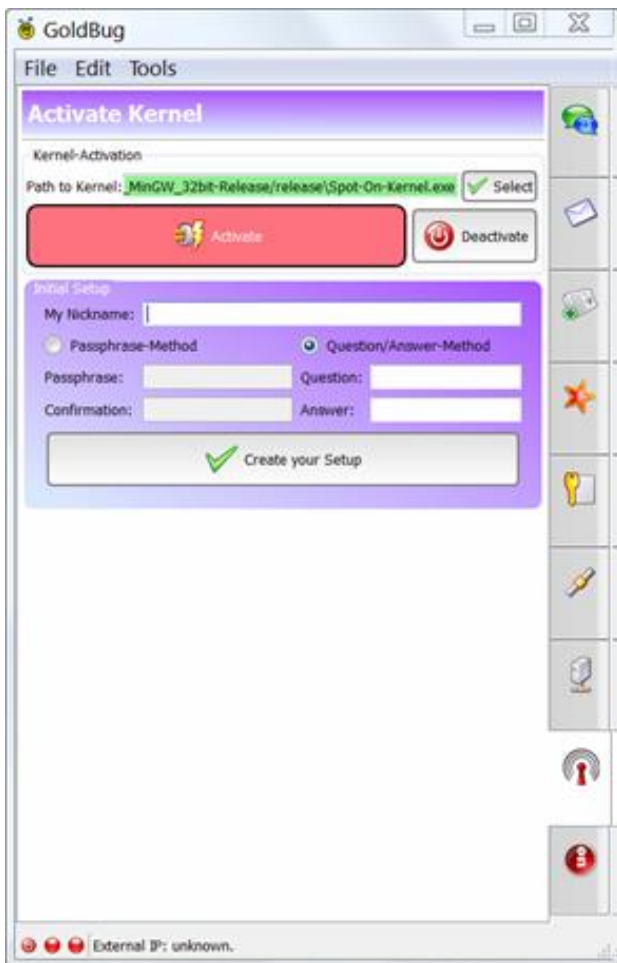
Bevor das Programm jedoch überhaupt gestartet werden kann, muss zunächst das "Initiale Setup" eingerichtet werden, d.h. Du musst Dir Deine Schlüssel für die Verschlüsselung erstellen. Dazu werden derzeit 8 Schlüssel generiert, was auf langsamen Maschinen ca. bis zu einer Minute dauern kann.

Ebenso ist eine Passphrase für den Messenger zu wählen, die jedes Mal als Login abgefragt wird, nachdem man das Programm, die Goldbug.exe, gestartet hat.

Das Passwort muss mindestens 16 Zeichen lang sein. Wem das zu lang ist, der kann ein Passwort auch dreimal wiederholen wie z.B. "passwort_passwort_passwort", jedoch ist das Passwort dann nicht so sicher wie eines mit zufälliger Zeichenkette.

Wenn Du GoldBug das erste Mal startest, gebe in dem blauen Kasten einen Nicknamen ein und definiere eine Passphrase. Dazu gibt es zwei Methoden: die Passphrase-Methode oder die Frage-Antwort (Question/Answer) Methode.

Abbildung 9: Setze Passwort, generiere Schlüssel und aktiviere den Kernel



Die zwei Methoden lassen sich wie folgt unterscheiden:

- **Passphrase-Methode:** Hash (Passphrase + Salt), das bedeutet ein "[salted Hash](#)" wird genutzt. Bei der Erstellung des Passwortes wird dieses nicht lokal gespeichert, sondern nur der Hash der Eingabe.
- **Q/A-Methode:** Hash (Question, Answer), das bedeutet ein "[HMAC](#)" wird genutzt. Und: Weder die Frage noch die Antwort wird auf Deiner Maschine gespeichert und kein Salz wird durch die Maschine per Zufall generiert. Anstelle einer Frage kannst Du natürlich auch zwei Passwörter ohne das Fragezeichen eingeben. Bitte beachte, dass hier die Frage und die Antwort bei späteren Logins exakt eingeben werden muss, wie sie definiert wurde und hier kein zweiter Eingabecheck ("Confirmation") wie bei der Passwort-Methode erfolgt.

Als bald die Schlüssel generiert sind, kannst Du den Kernel aktivieren. Drücke dazu auf den roten Knopf "Aktivieren" und vergewissere Dich, dass der Datei-Pfad zur Kernel.exe angegeben ist und somit grün hinterlegt ist.

Bei der erstmaligen Aktivierung wird die IP-Adresse des Projekt-Chat Server als Nachbar hinzugefügt und dieser dient als temporärer Chat Server, über den Du mit Deinen Freunden testweise chatten kannst, bis ihr Euch einen eigenen Verbindungsknoten auf einem Webserver oder Zuhause erstellt habt oder direkt verbindet. Bitte den Test-Server des Projektes nur für wiss. Testversuche benutzen.

Wenn ihr ohne Server direkt verbinden wollt, muss einer von beiden einen sogenannten Listener im Tabulator Chat-Server erstellen und für den Port die Firewall freigeben und den Port im Router zusätzlich an seine Maschine weiterleiten (s.u. ausführlicher).

Wenn Du das erste Mal den GoldBug Messenger startest, fragt Dich ein Pop-up Fenster, ob Du den Kernel aktivieren willst. Ansonsten bei allen weiteren Starts musst Du nach dem Login den roten "Activate Kernel" Knopf drücken, bevor es losgehen kann. Wenn er grün ist, läuft der Kernel.

Wenn Du die GUI schließt, wird der Kernel noch weiterlaufen. Es empfiehlt sich also, erst den Kernel zu deaktivieren und dann die Gui zu schließen. Ein weiteres Pop-up Fenster wird Dich aber in jedem Falle fragen, ob beides geschlossen werden soll. Ansonsten betreibst Du den Kernel ohne GUI, was ja manchmal auf einem Webserver gewünscht ist, damit niemand sich in die offene Benutzeroberfläche einschalten kann.

Du kannst den Kernel auch aktivieren/deaktivieren, indem in der Status Bar unten links die erste LED gedrückt wird. Wenn sie grün leuchtet, ist der Kernel aktiv, wenn sie rot leuchtet, ist der Kernel ausgeschaltet.

Deine generierten Schlüssel sind im Unterpfad „/.spot-on“ gespeichert. Wenn Du einen neuen Login mit neuen Schlüsseln aufsetzen willst und alle Nutzerdaten gelöscht werden sollen, dann lösche diesen Pfad einfach und starte neu. Gleiches kann im Hauptmenü mit "!!!_Total_Database Erase_!!!" erreicht werden.

Beschrieben ist bislang die minimale Sicht auf das Interface: Über das Hauptmenü kann man ebenso zwischen „voller Ansicht“ oder „minimaler Ansicht“ wählen. Wer sich mit Computern nicht so gut auskennt, kann sollte die minimale Ansicht wählen, da es die ggf. nicht benötigte Optionsvielfalt ausblendet. Keep it simple.

Bei nicht-minimaler Ansicht zeigen sich im Tabulator "Activate Kernel" noch folgende Elemente.

- Pfad zum Kernel: Hier kannst Du den Pfad zum Kernel eingeben. Ist der Kernel mit der "spot-on-kernel.exe" im Pfad richtig angegeben, dann ist der Pfad grün hinterlegt. Andernfalls schaue, wo die ausführbare Datei des Kernels liegt oder kopiere sie ebenso zur ausführbaren Datei der GUI (goldbug.exe) bzw. passe den Pfad entsprechend an.
- PID: Die PID Nummer kennzeichnet die Prozess-ID, mit der in Windows die ausführbare Datei gekennzeichnet ist. Du findest auch im Windows Task Manager die Prozess-IDs.
- Simulacra: Diese Funktion sendet bei Aktivierung der Check-Box eine "simulierte" Chat-Nachricht ins Echo Netzwerk. Diese „Fake“-Nachricht besteht aus reinen Zufallsziffern und macht es Analysten schwerer, verschlüsselte Nachrichten mit echten und zufälligen Nachrichten zu unterscheiden. Simulacra ist ein Begriff, der sowohl aus dem Film "[Die Matrix](#)" als auch in der Philosophie [Baudrillards](#) nicht unbekannt ist.
- Impersonator: Neben zufälligen Fake-Nachrichten kann mit dem GoldBug-Programm auch ein Chat simuliert werden, als wenn eine echte Person von Zeit zu Zeit chattet und Antworten aussendet. Auch diese Nachrichten sind mit reinen Zufallsdaten gefüllt, variieren jedoch - simuliert an einer echten Chat-Unterhaltung.
- Create Settings: Für die Schlüssel-Erstellung solltest Du einen Schlüssel größer als 2048 Bit wählen und kannst auch weitere Optionen wie Algorithmus, Hashtype, Cipher, Salz-Länge oder Iteration Count selbst wählen.
- Mit der "Regeneration"-Funktion kannst Du einzelne Schlüssel auch neu generieren - mit neuen Werten und Optionen. Checke dazu die Check-Box, setze die Werte und regeneriere den jeweiligen Schlüssel. Dann musst Du Deinen neuen Schlüssel jedoch wieder Deinen Freunden zur Verfügung stellen, denn der Schlüssel ist Deine Kommunikations-ID.

Tausche mit einem Freund den Schlüssel und ein erster Chat kann beginnen. Setze den Schlüsseltausch wie folgt um:

Die Schlüssel tauschen und einfügen

Du und Dein Partner, beide Freunde, müssen jeweils ihren öffentlichen Schlüssel tauschen d.h. auskopieren und sodann den Schlüssel des Freundes im Tabulator: „Freund hinzufügen“ („Add Friend/Key“) eingeben.

Dein Freund kann seinen Schlüssel per E-Mail oder auch über ein anderes Chat-Programm senden. Kopiere diesen dann in diesem Tabulator ein und drücke den „Hinzufügen“-Knopf am unteren Rand.

Du findest Deinen eigenen Schlüssel ebenso im Tabulator „Freunde hinzufügen“ („Add Friend/Key“). Über den großen Knopf ("Kopiere Schlüssel") oben kannst Du Deinen Schlüssel in die Zwischenablage auskopieren.

GoldBug nutzt eine öffentliche/private Schlüssel-Infrastruktur, wie sie auch z.B. von GnuPG bekannt ist. Der öffentliche Schlüssel kann getauscht werden und der private Schlüssel bleibt verschlüsselt auf Deiner Festplatte.

Die unterschiedlichen Funktionen von GoldBug haben entsprechend aus Sicherheitsgründen auch verschiedene Schlüsselpaare. Für Email wird ein anderer Schlüssel als für den Chat benutzt. Es gibt aber in dem Auskopieren-Knopf die Funktion, alle Schlüssel in einem einzigen Text auszukopieren. Kopiere hier den vollen Text und sende diesen zu Deinem Freund.

Ebenso macht es Dein Freund und Du fügst den Schlüssel des Freundes in das Textfeld ein.

(Ggf. kann es notwendig sein, mit der rechten Maustauste im Kontext-Menü einen neuen Freund als Freund zu bestätigen (Make-Friend-Funktion). Dieses kommt meistens dann zum Einsatz, wenn ein Freund seinen Schlüssel online in einer direkten IP-Verbindung sendet. Diese Funktion ist im Interface von Spot-on gegeben, in der GoldBug Benutzeroberfläche steht dieses nicht zur Verfügung, so dass beide immer den Schlüssel kopieren und einfügen. Sollte aber ein Freund den Spot-on Klienten nutzen und eine direkte IP-Verbindung zu einem Nutzer des GoldBug-Klienten aufbauen, dann wäre es theoretisch möglich, den Schlüssel auch per IP-Verbindung zu transferieren anstelle vom Copy/Paste).

Sodann erscheint der Freund mit seinem Nick-Namen im Tabulator Chat oder Email.

Besonderheit: Repleo

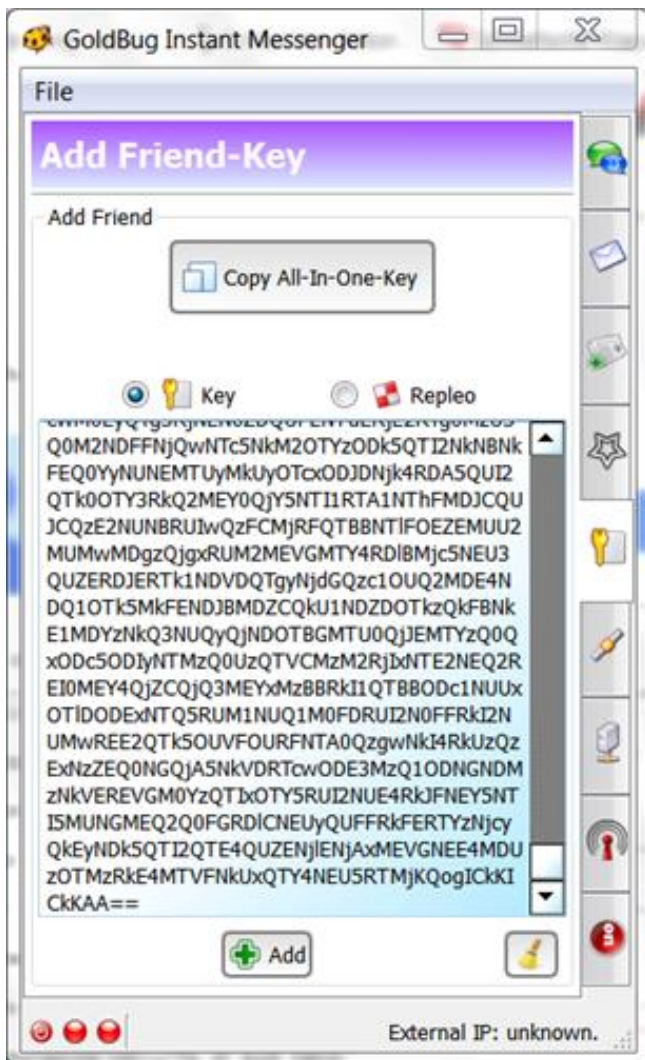
Wenn Du schon einen Schlüssel Deines Freundes erhalten hast und diesen eingefügt hast, nunmehr aber Deinen öffentlichen Schlüssel nicht preisgeben willst, ihn nicht bei einem E-Mail-Programm gespeichert wissen willst, dann kannst Du auch mit dem erhaltenen Schlüssel Deines Freundes Deinen eigenen Schlüssel verschlüsseln. Das nennt man dann REPLEO.

Beim Repleo wird also Dein öffentlicher Schlüssel mit dem öffentlichen Schlüssel Deines Freundes bereits verschlüsselt.

Auch ein Repleo kann dann Dein Freund in den Kasten des Tabulators "Add Friend/Key" einfügen.

Ein Schlüssel startet immer mit einem Buchstaben "K" oder "k" und ein Repleo startet mit einem "R" oder "r". Du kannst über dem entsprechenden Einfügekasten mit zwei Radio-Knöpfen bestimmen, ob es sich um einen Key oder ein Repleo handelt.

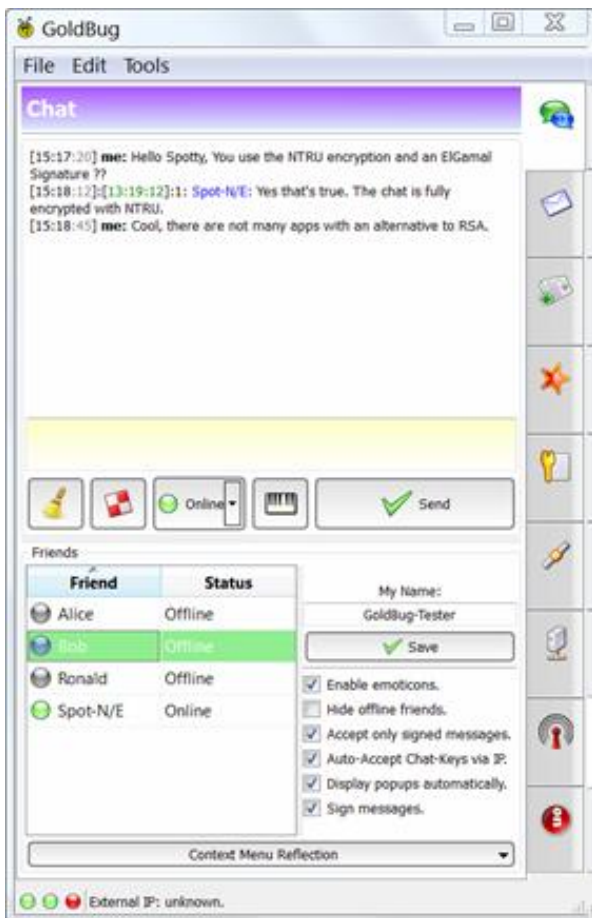
Abbildung 10: Tabulator Key: Schlüssel einfügen und auskopieren



Einen ersten sicheren Chat beginnen

Du findest nach erfolgreichem Key-Tausch Deinen Chat-Freund im Tabulator "Chat". Damit der Chat funktioniert, sollten beide Teilnehmer idealerweise die gleiche und aktuellste Version des Programmes nutzen, ihre Schlüssel generiert und ausgetauscht haben und zum einem Netzwerk-Knoten oder Chat-Server im Web verbunden sein. Wenn die ersten beiden LEDs in der Status-Zeile unten grün leuchten und der Names des Freundes im Chat-Tab auftaucht, sieht es schon gut aus.

Abbildung 11: Chat Tabulator



Wenn der Online-Status des Freundes blau (abwesend), rot (beschäftigt) oder grün (gesprächsbereit) aufleuchtet, kann der Chat beginnen. Entweder markiere den Freund in der Tabelle und chatte aus dem Tab heraus, oder doppelklicke mit der Maus auf den Freund und ein Pop-Up Chat Fenster für diesen Freund öffnet sich.

Abbildung 12: Pop-up Chat Fenster mit Doppelklick starten



Der Vorteil im Chat-Tab zu chatten ist, dass man gleich mehrere Freunde markieren kann, so dass die Nachricht alle Freunde erreicht. Wenn Du den Pop-up Chat nutzt dann musst Du nicht mehr auf die Markierung eines Freundes im Chat-Tab achten.

Zusätzliches Sicherheitsmerkmal: MELODICA

MELODICA steht für “**M**ulti **E**ncrypted **L**ong **D**istance **C**alling” – ins Deutsche übersetzt in etwa: „Vielfach-verschlüsseltes Anrufen über eine lange Distanz“

Es bezeichnet, einen Freund wie mit einem Telefon anzurufen – nur, dass damit eine sichere Ende-zu-Ende Verschlüsselung aufgebaut wird.

Die Ende-zu-Ende Passphrase – auch Gemini genannt - sollte zwischen beiden Teilnehmern geheim bleiben. Daher ist die elektronische Übertragung immer ein Problem, wenn diese potentiell abgehört werden kann. GoldBug hat dieses Übertragungsproblem dadurch gelöst, indem das Gemini mit einer symmetrischen Verschlüsselung durch einen nochmals verschlüsselten Kanal übertragen wird.

Gemini ist der Begriff für Zwilling, d.h. es bezieht sich auf beide Teilnehmer, die die Passphrase somit kennen sollten.

Abbildung 13: Das MELODICA Symbol



Der MELODICA-Knopf erzeugt somit einen „Call“, einen Anruf, bei dem das Ende-zu-Ende verschlüsselnde Passwort übertragen wird.

Genau genommen sind dieses zwei Schlüssel, denn das Gemini wird durch einen weiteren Schlüssel authentifiziert. Dieses wird auch [MAC-Hash](#) genannt.

Du kannst mit dem MELODICA-Knopf die Verschlüsselung jederzeit erneuern. D.h. das Paradigma des „Perfect Forward Secrecy“ ist um zwei Komponenten erweitert worden. Einerseits kann man die Ende-zu-Ende Passphrase manuell definieren und sie auch sofortig, also „instant“ jederzeit erneuern. Daher wird von „Instant Perfect Forward Secrecy“ (IPFS) gesprochen.

Im Vergleich bieten viele andere Tools nur einen Key pro Online Sitzung an und man kann die Verschlüsselungsphrase auch nicht manuell edieren.

Als weiteren Clou besteht bei GoldBug nunmehr auch die bislang einzigartige Möglichkeit, ein neues Gemini durch den Kanal eines bestehenden Gemini zu senden. Hier wird der Ende-zu-Ende Schlüssel durch eine Ende-zu-Ende Verbindung gesandt.

Die symmetrische Verschlüsselungsphrase wird also nicht mit einer asymmetrischen Verschlüsselung verschlüsselt (z.B. RSA oder ElGamal oder NTRU) und dann durch einen sicheren Kanal (SSL) von Punkt-zu-Punkt gesandt, sondern wird selbst mit dem bestehenden Gemini verschlüsselt und sodann erst durch die beschriebene Methode gesandt.

Sichere Transportverschlüsselung entsteht, wenn ein Messenger einen manuell definierten symmetrischen Schlüssel mit einem bestehenden symmetrischen Schlüssel encodiert und dann mit einem asymmetrischen Schlüssel zusätzlich verschlüsselt. Und dieses Packet durch eine sichere Verbindung gesandt wird.

Die Möglichkeit das Passwort sekundlich oder für jeden oder innerhalb eines jeden Anrufes erneuern zu können, macht es Angreifern somit sehr schwer, die Ende-zu-Ende-Verschlüsselung der MELEODICA-Funktion aufbrechen zu können.

Die Verschlüsselung ist dabei nicht neu, sondern lediglich der Verfahrensprozess ist

ausgeklügelt implementiert, um Sicherheit zu bieten.

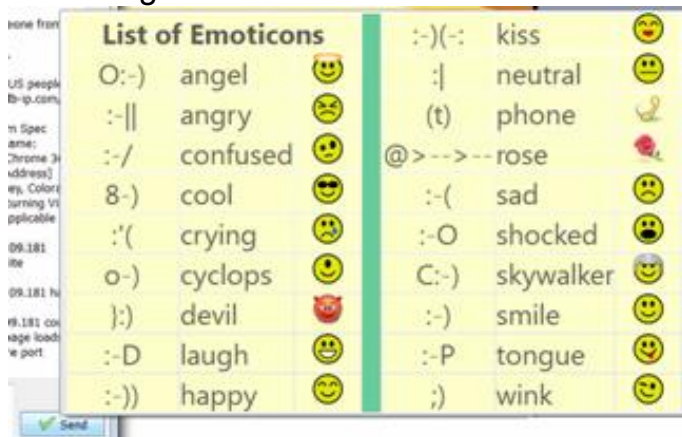
Ende-zu-Ende Verschlüsselung wird durch einfaches Knopf-Drücken so einfach wie telefonieren: Einfach den Hörer aufnehmen oder wieder auflegen. Zu jeder Zeit bleibt die Kommunikation asymmetrisch verschlüsselt und die symmetrische End-zu-Ende Verschlüsselung kann einfach hin zugeschaltet werden - und auch durch asymmetrische oder symmetrische Verschlüsselung (innerhalb eines SSL-Kanals) erneuert werden. Das ist ein neuer architektonischer Standard, den diese Methode etabliert.

Emoticons

GoldBug nutzt ein ganzes Bouquet an Emoticons - auch Smileys genannt.

Um die Hilfe zu nutzen, doppelklicke auf einen Freund, so dass sich ein Pop-up Chat-Fenster für den privaten Chat öffnet. Nun gehe mit der Maus über den Senden-Knopf. In einem dann erscheinenden Tooltip werden die Smileys angezeigt und mit Eingabe des ASCII Codes werden die Emoticons im Chat dargestellt. Im Chat-Tab besteht in den Optionen des rechten Seiten-Splitters auch die Möglichkeit, die graphische Darstellung von Smileys auszuschalten.

Abbildung 14: Liste der Emoticons



P2P E-Mail: ohne Vorratsdatenspeicherung

Neben der Chat-Funktion hat der GoldBug Messenger auch ein E-Mail-System integriert und wird neben den Gruppen-Chat-Funktionen zur ganzen Kommunikations-Suite.

Der E-Mail-Klient ist peer-to-peer basiert, d.h. die Emails werden über das Netzwerk der verschlüsselten Verbindungen gesandt.

Dieses Netzwerk wird durch die integrierte Architektur des Spot-on-Kernels bereitgestellt, wie sie auch der E-Mail Klient des Teams um BitMail.sf.net in einer frühen Version nutzt.

Wie schon dargestellt, nutzt die E-Mail Funktion einen anderen Schlüssel zur Verschlüsselung als die Chat-Funktion. So kannst Du also einen Freund zum Chat hinzufügen, aber das E-Mail verweigern. Sinnvoll ist es jedoch, immer alle Schlüssel als Ganzes aus zu kopieren, dann hat man seinen Freund auch in allen Funktionen präsent (neben dem URL Schlüssel und dem Rosetta Schlüssel, zwei Funktionen, die später noch beschrieben werden).

Natürlich kann auch für die E-Mail Funktion wieder die Sicherheit eines Repleos genutzt werden, wenn man seinen E-Mail-Schlüssel nicht der Öffentlichkeit preisgeben will.

Das interessante an der GoldBug E-Mail-Funktion – und hier unterscheidet es sich ggf. von anderen p2p E-Mail Implementierungen - ist, dass es möglich ist, Email auch zu Freunden zu senden, die offline sind.

Hierzu bestehen zwei verschiedene Methoden:

Die eine Methode ist, dass ein dritter, gemeinsamer Freund genutzt wird, um die Emails dort zwischen zu speichern.

Wenn Alice und Bob also einen Chat-Server im Web auf ihrem Webserver einrichten, und alle drei Ihre Schlüssel getauscht haben, fungiert der Webserver wie ein E-Mail-Postfach, wie wir es von POP3 oder IMAP her kennen.

Grundsätzlich benötigen die E-Mails jedoch keine zentralen Server, es kann auch ein dritter Freund zuhause sein, der kontinuierlich online bleibt. Es macht daher Sinn, mehr als einen Freund in seiner Liste zu haben und gemeinsame Freund mit anderen Freunden zu vernetzen, die als Zwischenspeicher fungieren können. Da alle E-Mails verschlüsselt sind, können die Freunde, die eine Cache-Funktion zur Verfügung stellen, Deine E-Mail auch nicht lesen.

Du hast bei GoldBug die Wahl, ob die E-Mails authentifiziert oder nicht authentifiziert, also einfach nur verschlüsselt, ohne Nachweis, dass der Schlüssel auch zu Dir gehört, gesandt werden.

Besonderheit: Zusätzliche Verschlüsselung mit einem „GoldBug“ setzen

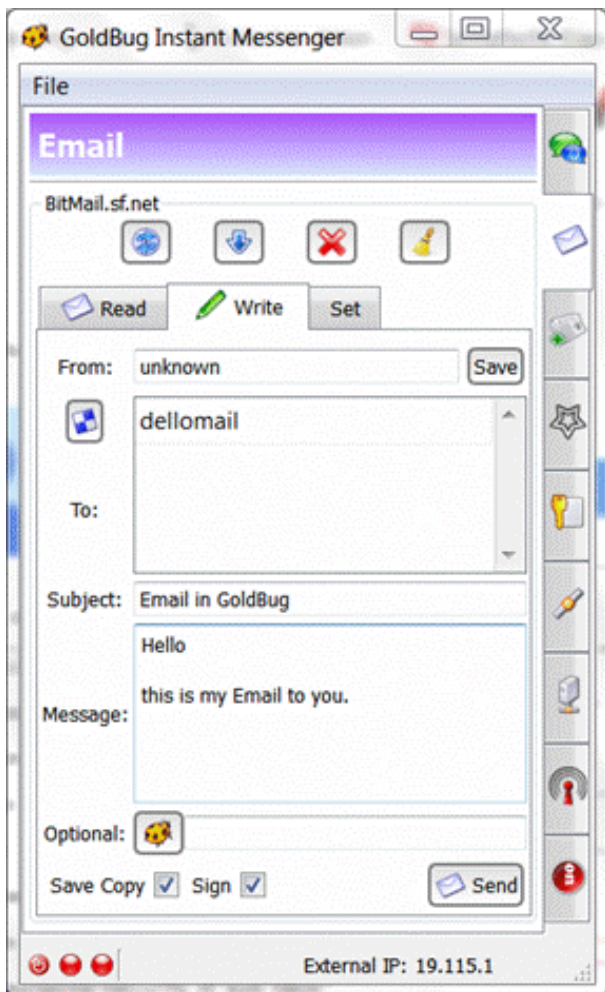
Nicht nur die Software nennt sich GoldBug, sondern auch die Funktion im E-Mail Klienten, auf das Email ein zusätzliches Passwort zu setzen.

E-Mails, auf die ein „GoldBug“-Passwort gesetzt wurde (vergleiche später unten die Beschreibung der File-Transferfunktion „StarBeam“, hier heißt das zusätzliche Passwort „Nova“) können vom Empfänger nur gelesen werden, wenn sie das entsprechende GoldBug – also den Goldenen Schlüssel für das Passwort kennen.

Du solltest daher Deine Freunde informieren, wenn Du ihnen E-Mails sendest, die zur Öffnung noch ein zusätzliches Passwort benötigen.

Das kann z.B. in den E-Mails zu seiner Frau sein, dass man die E-Mails immer mit dem Stadtnamen zusätzlich verschlüsselt, in dem die Hochzeit stattfand.

Abbildung 15: E-Mail Tabulator



Um die Care-Of (C/O) Caching-Funktion zu aktivieren, muss in dem Sub-Tabulator "Email-Settings" die Check-Box "Care-Of" aktiviert sein, wenn man als dritter Freund zwei anderen Freunden das zwischenspeichern der E-Mails im eigenen Klienten ermöglichen will und sie beide auch in der E-Mail-Kontaktliste einfügt.

Die zweite Methode ist die Einrichtung einer virtuellen E-Mail Institution.

Hierzu ist es ebenso notwendig, die C/O-Funktion mit der Check-Box zu aktivieren. Sodann ist eine virtuelle E-Mail-Institution zu erstellen und den Magnet Freunden zur Verfügung zu stellen, die in diesem Postfach dann zwischen-speichern. Zusätzlich muss der Node, der die E-Mail-Institution einrichtet, auch die öffentlichen E-Mail-Schlüssel der Freunde, die in seiner Institution speichern sollen, in seinen Node einkopieren. Der Vorteil gegenüber der ersten Methode ist jedoch, dass der öffentliche E-Mail-Schlüssel des Nodes, der die Institution einrichtet, niemandem bekannt gegeben werden muss.

E-Mail Attachments können ebenso einer Email angehängen werden und werden automatisch verschlüsselt.

C/O und E-Mail Institutionen einrichten

Hier wird nochmal anhand eines Beispiels beschrieben, wie die C/O Funktion des E-Mails und die Einrichtung einer virtuellen E-Mail-Institution Schritt für Schritt umgesetzt wird.

1. Aktiviere die C/O-Funktion in dem Tabulator für Email Settings.
2. Erstelle eine Institution und wähle einen Namen und eine Adresse für die Institution.

3. Beispiel: Name= "GB-Postfach" und Adresse = „Dotcom“
4. Füge den E-Mail Schlüssel eines Freundes in Deinen Klienten ein und lasse die Freunde Deinen E-Mail-Magnet von Deiner Institution in Ihren Klienten einfügen. Der Magnet wird ähnlich wie dieser aussehen:

magnet:?in=GB-Postfach&ct=aes256&pa=Dotcom&ht=sha512&xt=urn:institution

Du erkennst einen E-Mail-Magnet an seiner Endung: URN=Institution. Dann weißt Du, dass der Magnet kein Buzz-Gruppen-Chat Magnet ist und auch kein Star-Beam-Magnet für den Dateiaustausch - denn diese hätten die Endung „buzz“ bzw. „starbeam“.

So dann wird Dein Node die E-Mails Deiner Freunde zwischenspeichern – auch für Adressaten, die ggf. offline sein sollten.

Du (als Ersteller einer E-Mail-Institution) brauchst Deinen eigenen E-Mail-Schlüssel nicht mit den Freunden / Subscribern Deiner Institution zu tauschen.

Du kannst Deinen E-Mail Schlüssel in einem Gruppen-Chat Raum mit dem Ersteller einer E-Mail-Institution tauschen. Der Austausch-Prozess von Key & E-Mail-Magnet muss also keine weiteren Identitäten vermitteln.

Ge-Echo-ter IRC

Der GoldBug Messenger verfügt neben E-Mail und Chat auch über eine Gruppen-Chat-Funktion. Diese funktioniert ähnlich einem IRC-Chat. Die Übermittlung der Nachrichten an alle Gruppen-Teilnehmer erfolgt auch hier wieder vollständig verschlüsselt über das Echo-Protokoll. Letztlich können in dem p2p-Netzwerk alle Teilnehmer eines Gruppenchats mitlesen, die einen bestimmten symmetrischen Ende-zu-Ende-Schlüssel kennen, der den Chat-Raum definiert.

Es wird daher von ge-echo-tem IRC oder auch kurz e*IRC gesprochen, dass dem IRC-Chat neue Optionen eröffnet, da die Transportwege des e*IRC-Chats ebenso verschlüsselt sind - wie heute normale POP3- oder IMAP-E-Mails auch zumindest eine Transportverschlüsselung z.B. mit TLS 1.3 aufweisen.

Auch der althergebrachte IRC-Chat wird daher zunehmend solche Sicherheitsfunktionen berücksichtigen. Der e*IRC Chat kann dazu das Modell einer neuen IRC-Generation darstellen.

Auch die Verschlüsselungs-Details des Gruppenchats werden wieder über einen Magneten definiert (definiert mit Endung URN = Buzz).

Zum Start des GoldBug-Programms wird der Entwickler-Chat-Raum geöffnet, der als Beispiel dienen kann.

Um einen eigenen Kanal beizutreten, gebe einfach den Raumnamen ein oder nutze die oben angesprochene Methode des Magnet-Links. Der Magnet-Link hat neben dem Raumnamen zusätzliche Werte für die Verschlüsselung eingebettet wie z.B. Schlüssel, Hash oder Cipher für den Verschlüsselungstyp.

Wenn Du nur den Raumnamen eingibst, und keinen Magnet-URI, werden die zusätzlichen Verschlüsselungsdetails auf den Wert 0000 gesetzt und die Verschlüsselung des Raumes erfolgt auf Basis des Raumnamens.

Wenn Du alle Werte eingegeben hast, drücke den Knopf „Join/Beitreten“ – und wenn Du einen Magnet eingefügt hast, dann nutze im Pull-Down Menü den Befehl „de-magnetize“. Der Magnet wird wieder in seine Einzelbestandteile zerlegt und der Raum wird auf Basis der Verschlüsselungswerte erstellt und betreten.

Wenn der Raum geöffnet ist, kann Du den Raum auch als Bookmark abspeichern oder den entsprechenden Magnet-UIR auch jederzeit aus Deinen Chat-Raum-Bookmarks auskopieren und an Deine Freunde senden, um sie in einen Raum einzuladen.

Um eine Nachricht zu senden, gebe etwas Text ein und drücke den Senden-Knopf.

Der e*IRC Chat-Raum kann öffentlich oder privat sein, das hängt davon ab, wie sehr Du den Magnet bzw. die einzelnen Verschlüsselungswerte bekannt gibst.

Als öffentlichen e*IRC Chat Raum kannst Du den Magnet-URI auf Deiner Webseite bekannt geben und jeder weiß, wie er in Deinen Chat-Raum kommen kann - mit "de-magnetize".

Letztlich funktioniert es wie ein IRC hat, nur mit dem Unterschied, dass der Internet-Provider und weitere Rooting-Server nicht in die Kommunikation hineinsehen können, da sie ja verschlüsselt ist - wie Deine Verbindung beim Online-Banking auch.

Es macht also keinen Unterschied mehr, ob Du mit Freunden sprichst oder Deinem Bankberater.

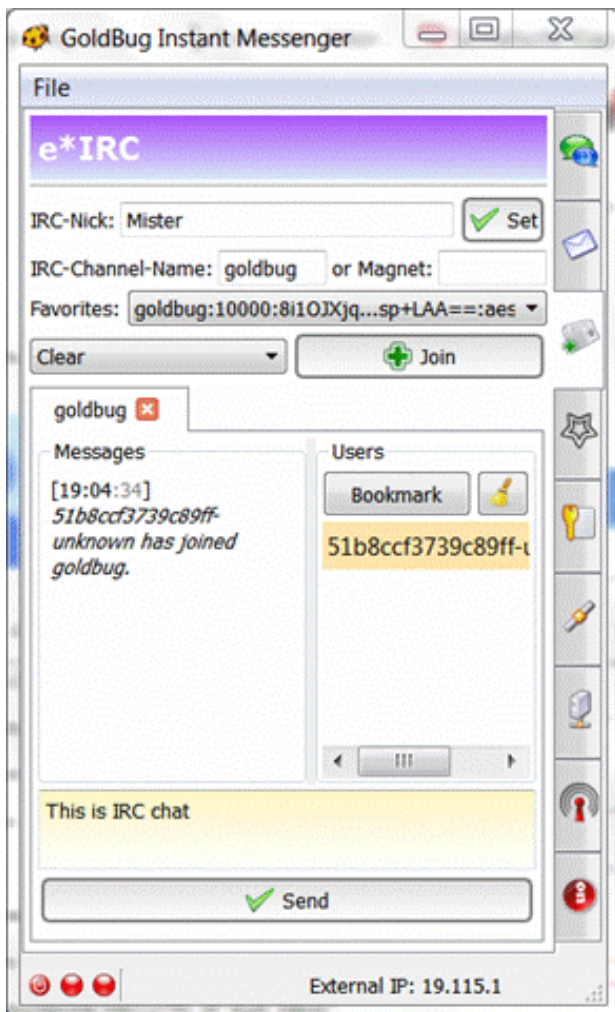
Wenn Du den Chat-Raum als privaten Raum, nutzen willst, kannst Du den Magnet-URI sogar mit Freunden teilen, ohne ihren öffentlichen (asymmetrischen) Schlüssel zu tauschen. Erstelle einfach einen One-Time-Magnet bzw. -Room und schütze Deinen öffentlichen Chat-Schlüssel.

Diese Funktion ist eine der Besonderheiten des GoldBug Programmes, dass Du einfach verschlüsselt chatten kannst ohne vorher asymmetrische Schlüssel tauschen zu müssen oder aber in einem privaten IRC Raum den asymmetrischen Schlüssel geschützt tauschen kannst.

GoldBug ermöglicht mit dem Repleo und dem Schlüsseltausch über einen One-Time-Magneten (OTM) für einen privaten Chatraum einen gesicherten Schlüsseltransfer und zusätzlich müssen öffentliche Schlüssel nicht mehr öffentlich sein.

Während andere Applikationen den öffentlichen Schlüssel mit allen Freunden oder gar in einem [DHT](#) teilen und z.T. auch die eigene IP-Adresse an den Schlüssel knüpfen, ist die oben vorgestellte Architektur wesentlich sicherer und zukunftsweisender.

Abbildung 16: IRC-Gruppenchat über das Echo



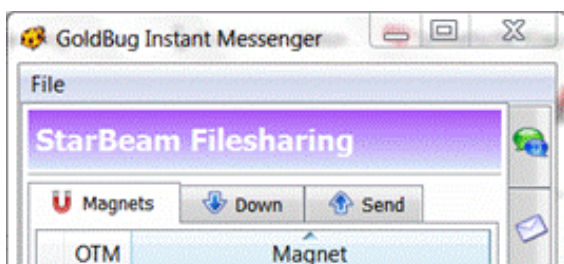
FileSharing: mit StarBeam

Wie in jedem Messenger kann auch in GoldBug Filesharing zwischen mehrerer Personen oder ein File-Transfer zwischen zwei definierten Personen umgesetzt werden.

Dazu gilt es auf folgende Schritte hinzuweisen:

- Hinzufügen oder Erstellen eines SB-Magneten
- Optional: Verschlüsselung der Datei mit einer Passphrase: „Nova“ genannt
- Optional: Verschlüsselung der Datei mit der Datei-Verschlüsselung in GoldBug.
- Wähle die Datei und einen SB-Magnet aus: Transferiere die Datei verschlüsselt

Abbildung 17: StarBeam Tabulator für Dateitransfer



Der Tabulator „StarBeam“ für das File Sharing besteht aus drei Sub-Tabulatoren: einen für das Hochladen, einen für das Herunterladen und einen für das Erstellen oder Hinzufügen von SB-Magneten.

Viele Nutzer kennen es noch von einem Emule oder Torrent Klienten: einfacher kann es nicht sein: Upload, Download und ein Tab zum Einkopieren des Magnet-URI.

SB-Magneten und Novas

Ein [Magnet-URI](#) ist ein Standard, der aus vielen File-Sharing Programmen bekannt ist (vielfach im Gnutella Netzwerk) und ebenso eDonkey/Emule ed2k-Links oder auch Torrent-Links entspricht.

Die Weiterentwicklung des Magnet-URI Standards durch die dem GoldBug Messenger zugrunde liegende Spot-On Bibliothek liegt in der Ausgestaltung des Magnet-URI mit Verschlüsselungswerten.

Magneten werden also genutzt, um ein Bündel an kryptologischen Informationen zu erstellen oder zusammen zu halten.

SB-Magnet-URIs werden daher in der Community auch als Crypto-Torrents bezeichnet, da sie wie ein Torrent-Link auf einer Webseite verlinkt werden können und den Zugang zu einer Datei - oder gar als ein Channel für verschiedene Dateien - verlinkt werden können.

Durch diesen Dual-Use-Effekt kann ein Magnet auch nicht einer einzelnen Datei oder einer bestimmten IP-Adresse zugeordnet werden. Auch ein Dateiname taucht in dem Crypto-Torrent bzw. SB-Magnet nicht auf, wie es dennoch selbst bei den - gegenüber Gnutella, Emule und Torrent-Link - fortschrittlicheren Verlinkungen beispielsweise von Offsystem.sf.net oder Retroshare.sf.net ist.

Während hingegen zahlreiche Meinungen das Verlinken von Gnutella, Edonkey und Torrent-Links kritisch sehen, besteht bei einer Kollektion aus Verschlüsselungswerten keine Veranlassung dazu, diese Werte zu diskreditieren. Deine Homepage bzw. unabhängige Portale finden mit StarBeam also fortschrittlichste Technologie.

Neben den strategischen Entscheidungen der Auswahl eines Link-Standards geht es bei dem Nutzungsaspekt jedoch auch um die Sicherheit des Dateitransfers zwischen zwei privaten Nutzern.

Für den Ablauf der privaten Datei-Übertragung von Freund zu Freund einige weitere Hinweise:

Bevor Du eine Datei versendest, kannst Du überlegen, ob Du sie einfach per Email an ein Email innerhalb von GoldBug anhängst. Dieses ist die Variante der Wahl, wenn die Datei kleiner als 10 MB ist.

Größeren Dateien sollten ausschließlich über die StarBeam-Funktion übertragen werden.

Vor einem Versand kannst Du auch überlegen, die Datei auf der Festplatte zu verschlüsseln. Dazu hält der GoldBug Messenger im Hauptmenü unter Werkzeuge/Tools das Werkzeug für die Dateiverschlüsselung bereit. Mit einer doppelten Passphrase wird die Datei darin verschlüsselt.

Manche packen die Dateien auch in ein Zip und verschlüsseln dieses vor dem Versand oder Upload. Die Zip-Verschlüsselung ist jedoch sehr leicht zu knacken mit 96 Bits, insofern sollte man also einen Schlüssel nutzen wie er heute für RSA mit mindestens 2048 Bits empfohlen wird.

Egal wie Du Deine Datei nun auch vorbereitest - so wie sie ist, als plain-Binärdatei, oder verschlüsselt mit dem GoldBug-Tool über StarBeam - wird sie ja wiederum mit dem Echo-Protokoll mehrfach verschlüsselt.

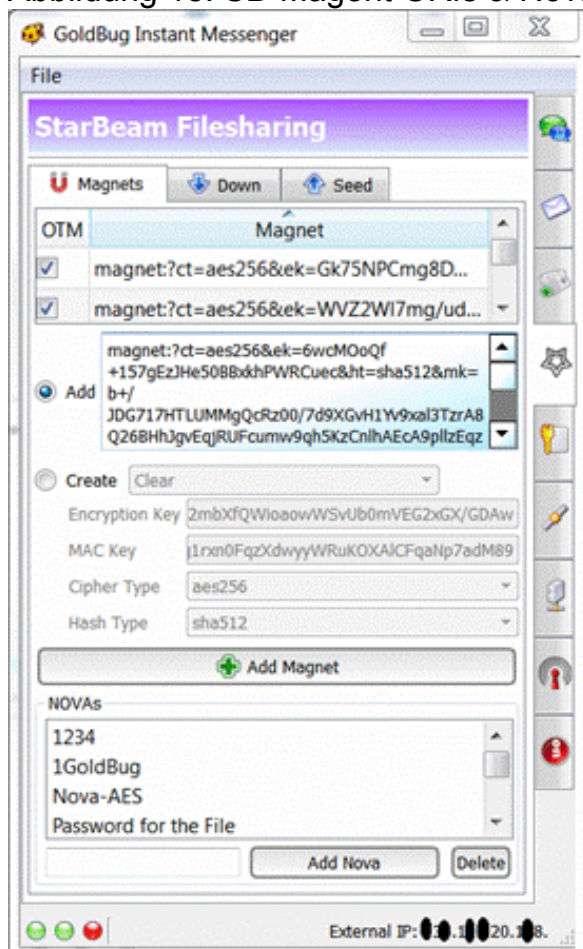
Genau wie man beim Email ein zusätzliches Passwort auf ein Email setzen kann („GoldBug“ bei der E-Mail-Funktion genannt, siehe oben) kann man auch auf die Datei – genauer auf den verwendeten Magnet-UIR zum Dateitransfer ein weiteres Passwort setzen. Dieses wird „Nova“ genannt. Selbst wenn der Dateitransfer erfolgreich geglückt ist oder auch ein dritter Unbekannter die bisherige Vielfach-Verschlüsselung kacken könnte (was nicht anzunehmen ist), wird mit dem Nova-Passwort eine Ende-zu-Ende Verschlüsselung eingeführt, die so lange sicher ist, wie das gemeinsame Passwort ausschließlich bei beiden Partner unter Verschluss ist.

Um eine Datei zu senden, muss ein verschlüsselter Kanal erstellt werden. Dieses funktioniert wieder mit der Erstellung eines Magneten (am Ende gekennzeichnet durch URN=SB-StarBeam).

So wird Datei-Packet für Datei-Packet – auch Datei-Chunk oder Datei-Link genannt - über diesen Kanal verschlüsselt übertragen mittels des HTTPS-Protokolls (das auf TCP, UDP und auch SCTP Verbindungen aufsetzen kann). Es ist daher eine interessante Fragestellung, ob ein Transfer einer großen, verschlüsselten Datei mittels StarBeam über das SCTP, TCP oder UDP Protokoll ceteris paribus fehlerfrei und am schnellsten übertragen wird.

Somit wird deutlich, dass in StarBeam keine bestimmte Datei getauscht wird, sondern es werden grundsätzlich nur verschlüsselte Kanäle getauscht. Sozusagen ein „Wurmloch“, um bei dem Begriff der „Stars“ zu bleiben. Und dieser Kanal wird durch einen Magnet-URI-Link definiert.

Abbildung 18: SB-Magnet-URIs & Novas



Idealerweise hast Du für jede Datei einen eigenen Magnet-URI. Das wäre sodann ein One-Time-Magnet (OTM), ein Magnet, der nur einmal genutzt wird für eine Datei. (OTM entspricht also dem Gedanken eines OTP - einem [One-Time-Pad](#): eine Zeichenfolge, die nur einmalig genutzt wird. OTP wird oftmals in kryptologischen Prozessen als entscheidend angesehen,

um Sicherheit herzustellen).

Du kannst aber einen Magnet-URI auch dauerhaft nutzen, dann ist das wie ein abonnierter Video-Kanal, in dem z.B. jeden Montag eine Datei versandt wird.

Das eröffnet z.B. auch Torrent-Portalen ganz neue Möglichkeiten, es muss gar kein Portal mehr existieren, in dem tausende an Links verlinkt sind. Das Portal selbst braucht nur einen einzigen Magnet-URI in diesem dezentralen Echo Netzwerk, um sodann konsekutiv, nach und nach, eine Datei nach der anderen durch das Wurmloch zu senden. Wer fürchtet, dass der verbundene Nachbar einen Dateiversand missbilligen könnte, braucht dann nur von p2p auf f2f umzustellen und mit Echo-Accounts ein [Web-of-Trust](#) zu erstellen. Verbinde Deinen Node nur zu einem vertrauten Freund, indem Du die Credentials des Echo-Accounts teilst sowie einen Magnet-URI für Deinen Datei-Kanal.

Als bald Du eine Datei über den Magnet-URI transferiert hast, kannst Du den Magnet-URI also löschen oder beibehalten. Erstellst Du den Magneten als OTM und aktivierst die Check-Box für OTM, dann löscht er sich nach Dateitransfer von selbst. Mensch, das ist ja wie bei Mission Impossible.

So kannst Du Dein Tagebuch Deiner Urlaubsreise mit Deiner Schwester teilen und sicher über das Internet übertragen, ohne es unverschlüsselt irgendwo hochladen zu müssen.

Das Werkzeug des GoldBug-File-Encryptor kann man natürlich auch nutzen, wenn man eine Datei irgendwo in einen Online-Hoster hochladen will.

Da diese jedoch die Dateien ggf. kontrollieren und verschlüsselte Dateien mit einem Fragezeichen versehen werden, obwohl es ein Ausrufezeichen sein sollte, macht es Sinn, die verschlüsselte Datei gleich von Punkt zu Punkt, von Freund zu Freund über GoldBug zu transferieren.

Wie genannt empfiehlt es sich, auf den Datei-Transfer wenigstens ein sog. Nova als zusätzliche Passphrase zu setzen.

Denn, wenn die Übermittlung des SB-Magnet-URI abgehört werden sollte – Du musst den Crypto-Torrent ja irgendwie online an Deinen Freund übertragen – dann kann jeder, der den Magnet-URI kennt, auch die Datei ebenso empfangen. Daher macht es Sinn, die Datei mit einem Nova zu schützen – ein Passwort, das beide Freunde ggf. mündlich, in der Vergangenheit oder über einen zweiten Kanal getauscht haben.

Das Nova baut auch auf dem Ende-zu-Ende Verschlüsselungsstandard AES auf (wenn Du Dir nicht ein eigene Passphrase ausdenkst). Und: Es muss - bevor - der Dateitransfer beginnt, in dem Node des Empfängers hinterlegt worden sein.

Wenn ein Empfänger ein Datei-Packet, ein Chunk oder Link, empfangen hat, ist er in der Lage, dieses nochmal hochzuladen – auch in andere Magnet-URI-Kanäle – oder es nochmals in den gleichen Kanal zu geben. Dieses ist ähnlich einer Rewind-Funktion: Die Datei wird einfach nochmal wie auf einem Kassetten-Recorder oder MP3-Spieler über das Echo-Netzwerk erneut abgespielt. Die Datei kann auch viele Stunden oder Tage später nochmals versandt werden. Jeder, der eine Kopie über den Magnet-URI-Kanal erhalten hat, wird zu einem Satelliten, und kann die Daten erneut in ein Wurmloch oder besser: StarBeam-Magnet-URI wieder einspielen.

Um den Transfer durchzuführen, benötigst Du nur eine Verbindung zu einem Nachbarn oder Freund und kannst diese auch mit einem Echo-Account absichern, so dass nur Freunde untereinander verbinden können.

Die Übertragung mit dem Echo-Protokoll ist effektiver, als sie über ein Protokoll ähnlich dem „[Turtle Hopping](#)“ (siehe Wikipedia) laufen zu lassen, da hier je nach Ausgestaltung des Echo-Netzwerkes (Volles Echo, halbes Echo, Adaptives Echo, Superecho) und der grundsätzlichen Verschlüsselung Knotenpunkte mit nur geringer Bandbreite nicht als Flaschenhals wirken müssen, sondern über weitere Echo-Wege die gewünschten Download-Geschwindigkeit optimieren.

Eine Datei übertragen

Wenn Du einen Magnet-URI definiert oder generiert hast, erscheint er nicht nur im Sub-Tab für die Magneten, sondern auch in der Tabelle im Sub-Tab für den Upload/Seed.

Wähle in der Check-Box einen SB-Magnet aus. Ebenso wähle die Datei.

Schließlich entscheide Dich noch, ob Du auf den Transfer ein zusätzliches Password – Nova genannt – setzen möchtest. Für einen ersten Test kann man dieses erst mal auslassen.

Die Chunk-Grösse (Pulse-Size) kann man so wie vor-definiert belassen.

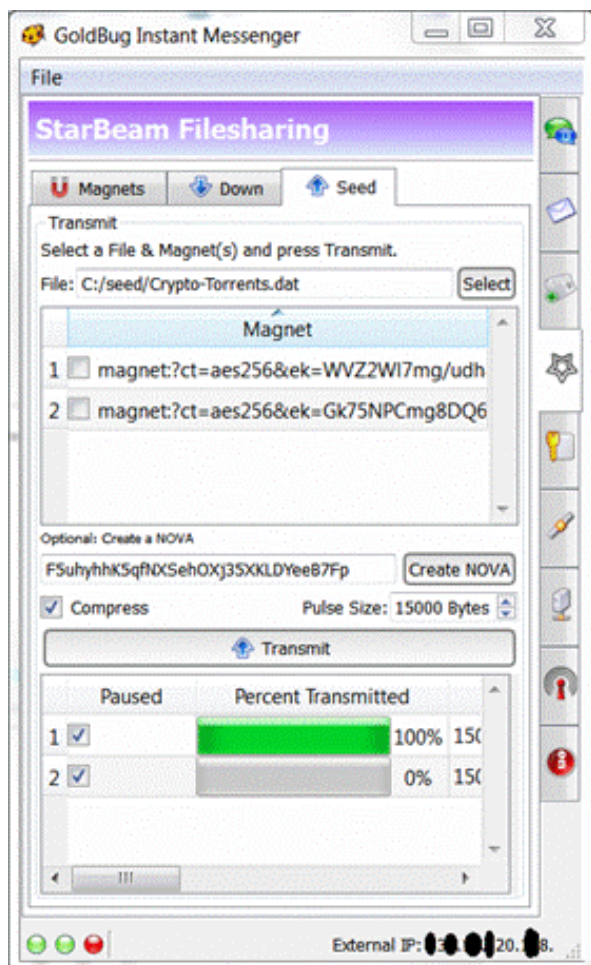
Das das Echo als HTTP-Post oder -Get übertragen wird, entspricht der Transfer einer Webseite. Wenn die Pulse-Size größer gemacht wird, wird die Webseite sozusagen länger, die übertragen wird.

Sodann drücke den Knopf „Transmit“/„Übertragen“.

Schließlich kopiere den Magnet-URI aus und sende diesen Deinem Freund. Wenn er ihn einkopiert hat, kannst Du den Transfer mit der Deaktivierung der Pause-Funktion starten.

Den Magnet-URI kann man in dem rechten Seitensplitter zur Transfertabelle auskopieren.

Abbildung 19: Eine Datei übertragen



StarBeam Downloads

Um mit StarBeam eine Datei zu laden, benötigst Du wiederum einen SB-Magnet-URI bzw. umgangssprachlich manchmal auch Crypto-Torrent genannt. Dieses findest Du auf Webseiten verlinkt bzw. kannst Du diesen von einem Freund, der Dir eine Datei senden will, erhalten.

Kopiere sodann den Magnet-URI in den Sub-Tab für die Magnet-URIs einfach ein. Teile Deinem Freund mit, dass Du den Magnet-URI eingefügt hast und er die Übertragung starten kann.

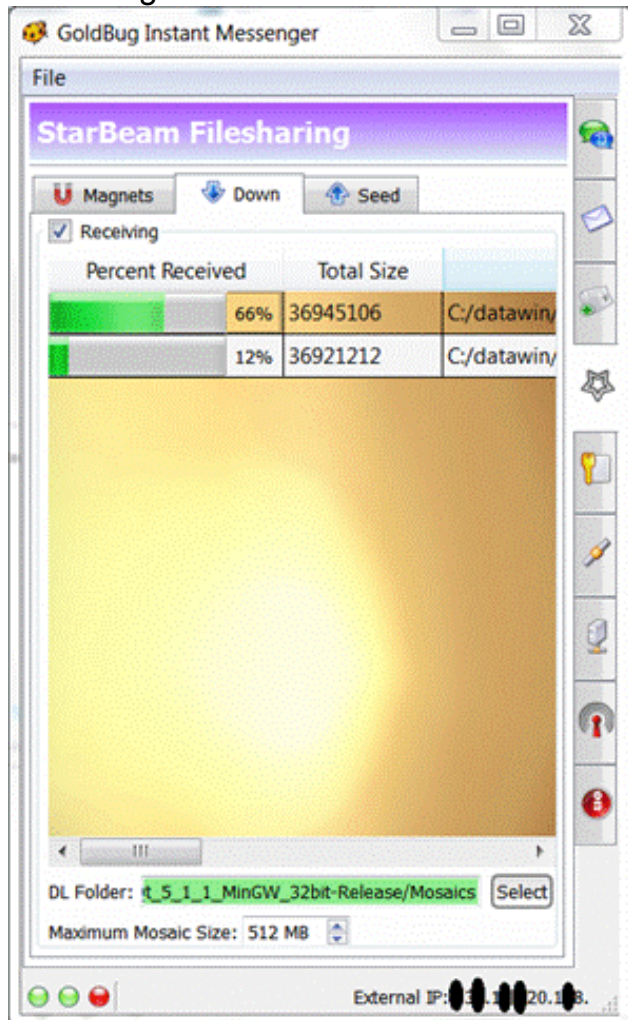
Zuvor solltest Du im Download-Sub-Tab noch die Check-Box „Receiving“ / „Empfang“ aktivieren.

Sodann sollte der Download starten, alsbald ein Sender über das Echo und durch den Kanal des Magneten eine Datei sendet.

Mit den weiteren Einstellungen auf dieser Seite kannst Du noch die Größe und den Pfad für den Download-Bereich definieren.

Die erfolgreich heruntergeladenen Teile werden Mosaics genannt. Die zu übertragenden Dateien werden Links (bzw. in der Community auch: Chunks) genannt.

Abbildung 20: Download von Dateien



Sollte eine Datei mal nicht erfolgreich übertragen worden sein, kann dieses mit dem StarBeam-Analyser Werkzeug überprüft werden.

Dieses stellt fest, ob alle Mosaike vorhanden sind oder ob noch Links bzw. Chunks fehlen. Wenn noch Links fehlen, erstellt der SB-Analyser einen Magnet-URI, den der Freund nochmals in seinen Upload-Tab eingeben kann. Dann werden nur die noch fehlenden Links bzw. Chunks erneut gesandt.

Die Datei würde sich aber auch vervollständigen, wenn der Sender sie dreimal am Tag über das Echo mit der „Rewind“ (= „Erneut senden“)-Funktion sendet.

Beachte, dass ein Magnet ein Kanal ist, und vorhandene Dateien in Deinem Mosaik-Pfad sodann erneuert werden, wenn kein One-Time-Magnet genutzt wird.

Eröffnen StarBeam-Magnet-URIs also neue Wege des Denkens beim Thema der Anwendung von Crypto-Torrents über das Echo-Protokoll?

Einen ersten Setup einrichten

Wie ein allererstes Profil eingerichtet wird, ist oben schon erläutert worden. Nick-Name und zweimal ein 16-Stelliges Passwort eingeben. Fertig. Wahlweise eine Frage-/Antwort-Phrase anstelle des Passwortes wählen.

Im folgenden geht es nunmehr um das Setup des Netzwerkes. Wenn Du den GoldBug Messenger erstmalig ausprobierst, wirst Du über den Projektserver verbunden. Freunde von Dir ebenso, so dass die Software von Euch getestet werden kann.

Sodann – wenn die Grundfunktionen deutlich sind - planen fortgeschrittenere Nutzer sicherlich auch die Nutzung eines eigenen Chat-Servers oder die Verbindung ohne Chat-Server direkt zwischen zwei Freunden.

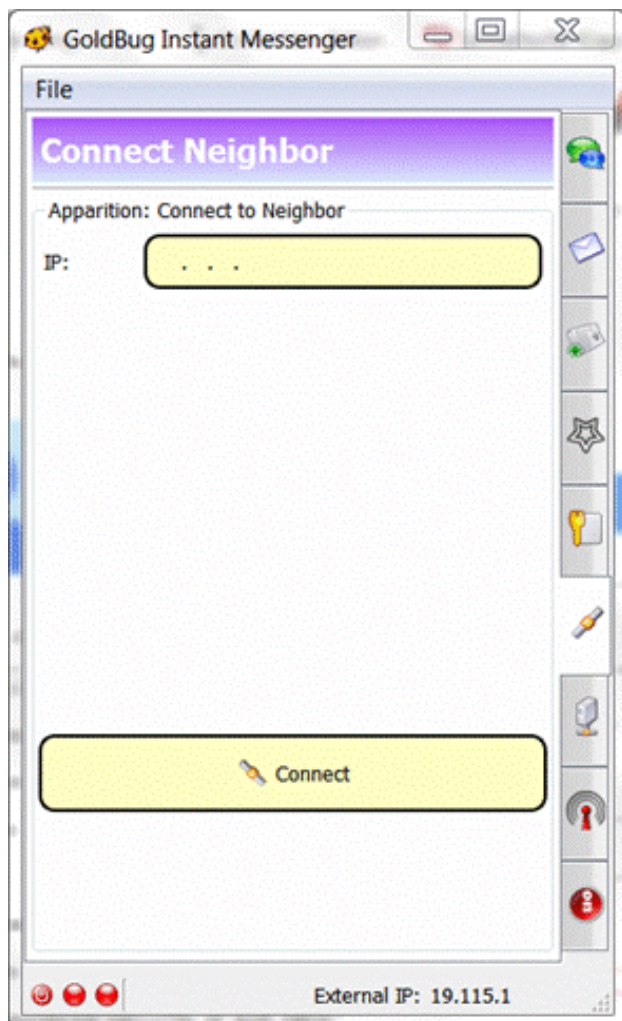
Die nächsten Schritte erläutern daher die

- Verbindung zu einem Nachbarn/Chat-Servers,
- die Erstellung eines eigenen Chat-Servers bzw. Listeners
- sowie weitere Details, die in der Nicht-Minimal-Sicht eingeblendet werden können.

Um es einfacher und für den Beginn übersichtlicher zu gestalten, wähle im Hauptmenü die Option „minimale Ansicht“ aus.

Gehe sodann auf den Tabulator: "Nachbar verbinden". Hier zeigt sich ein Eingabefeld für die IP-Adresse des Nachbarn bzw. des Webserver, auf dem ein Spot-On-Kernel läuft bzw. ein Freund ebenso einen GoldBug Messenger nutzt.

Abbildung 21: Eine IP-Adresse als Nachbarn hinzufügen.



Gebe in das Feld die IP-Adresse des Nachbarknoten ein. Mit den Punkten sind jeweils drei Stellen der IP Adresse getrennt. Umfasst ein Block nur zwei Stellen, z.B. in 37.100.100.100, dann kann die 37 in dem ersten Block beliebig platziert werden oder als 37 auf den ersten beiden Positionen eingegeben werden.

Sodann drücke den „Verbinden“-Knopf.

Die IP-Adresse ist sodann auf dem voreingestellten Port 4710 hinterlegt.

Wenn eine Fehlermeldung erscheint, dann ist diese IP-Adresse schon eingegeben. Um alle Nachbarn zu löschen, kannst Du dann den Knopf „Alle Nachbarn löschen“ drücken und die IP-Adresse erneut eingeben.

Wahlweise kann im Installations-Pfad `./spot-on` auf der Festplatte auch die Datei „neighbors.db“ gelöscht werden. Sie bildet sich sofort neu und ist dann leer.

Wenn der Kernel aktiviert ist (linke, erste LED in der Statuszeile leuchtet grün) und der Nachbar verbunden ist (mittlere LED leuchtet grün) ist alles erfolgreich installiert und online.

Eine IP-Adresse einzugeben und den Verbindungsknopf zu drücke, sollte gelingen. Wer mehr Details sehen will, kann die minimale Ansicht auch gegen die volle Ansicht wechseln.

In dieser Ansicht wird deutlich, dass neben der IP-Adresse auch der Port der IP-Adresse individuell konfiguriert werden kann. Standardmässig nutzt GoldBug den Port 4710.

Ferner kann der Klient auch über IPv6 betrieben werden sowie einen Listener ansteuern, der über das [Dynamische DNS](#) verlinkt ist. Damit gibt man dann keine Nummernfolge bei der IP ein, sondern einen Domain-Namen.

In der darunter liegenden Box können weitere Sicherheitsoptionen definiert werden.

Einen Chat-Server oder Spot-on-Kernel einzurichten bedeutet, einen sogenannten „Listener“ einzurichten, so der technische Begriff.

Dieser wird standardmäßig für das TCP Protokoll eingerichtet, GoldBug ist jedoch auch dafür ausgestattet, einen Listener über das UDP oder drittens auch SCTP Protokoll einzurichten. Beide letztgenannten Protokoll sind ideal für VOIP oder Streams.

Daher kann in den Verbindungsoptionen auch definiert werden, ob Dein Klient sich über TCP, UDP oder SCTP zum Nachbarn bzw. Server verbinden soll.

Der Nachbar oder Listener des Servers kann auf SSL-Verbindungen verzichten, dann wird die Übertragung nicht über HTTPS, sondern nur über HTTP geregelt.

Ein Listener kann die Sicherheitsoption setzen, ein permanentes SSL-Zertifikat zu erzeugen. Damit wird der bei SSL bestehende [Diffie-Hellman-Schlüsselaustausch](#) bzw. -Verhandlungsprozess nicht in jeder Sitzung neu verhandelt, sondern ein Angreifer müsste schon einen Aushandlungsprozess in der Vergangenheit kennen, um hier einzugreifen. Es kann aber sein, dass der Server bzw. Listener sein SSL-Zertifikat mal erneuert, daher macht es ggf. Sinn, Ausnahmen („Exceptions“) zuzulassen, wenn man eine Verbindung einfacher erstellen will und diese zusätzliche Sicherheitsebene nicht perfektionieren will.

Ebenso kann man seinerseits die Schlüsselgröße für die SSL-Verbindung definieren und auch bestimmen, dass Verbindungen unterhalb einer bestimmten SSL-Schlüsselgröße gar nicht erst aufgebaut werden. Einmal wird also definiert, was der Nachbar an SSL-Schlüsselgröße braucht und das andere Mal wird definiert, welche Schlüsselgröße Du von einem Server bzw. Nachbarn erwartest.

Schließlich besteht die Option, dass der Klient bestimmt, ob er zu dem Nachbarn mit vollem oder halbem Echo verbindet. Bei halbem Echo wird das Nachrichtenpaket nur an den Nachbarn einen Hop über die direkte Verbindung gesandt. Angenommen, Dein Freund hat den Webserver eingerichtet und sitzt auch davor und Du möchtest nicht, dass Deine Echo-Pakete an dritte und seine Freunde gehen, dann kannst Du mit dem Halben Echo definieren, dass Deine Pakete nach Erhalt durch den Server nicht weiter verbreitet werden. Damit chattet ihr über eine direkte IP-Verbindung. Beide Teilnehmer sehen beim Halben Echo die IP-Adresse des Freundes und Chat-Partners. Beim Vollen Echo muss der Chat Freund nicht

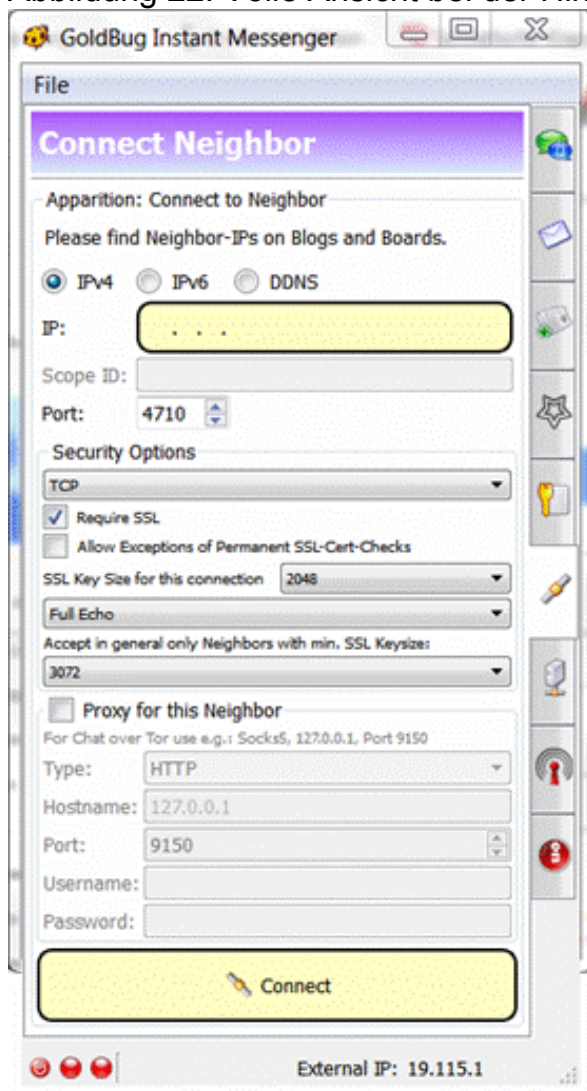
Administrator des Knotenpunktes sein, sondern kann wie ein zentraler Chat-Server mehrere Klienten miteinander verbinden.

Wenn Du GoldBug als Klient über einen Proxy in der Firma, hinter einer Firewall bzw. einem Proxy der Universität oder auch über das Anonymisierungsnetzwerk Tor laufen lassen willst, kannst Du die Proxy-Details für einen Nachbarn einfügen.

Als Klient kannst Du Dank des HTTP-Protokolls aus jeder IT-Umgebung verbinden, wenn Du in dieser Umgebung auch mit einem Browser surfen kannst.

Entscheidend ist, einen Knotenpunkt im Web mit einem GoldBug Node zu adressieren, der ggf. nicht vom Port her durch Deine Firewall bzw. den Proxy limitiert wird. Falls das der Fall ist, bitte doch Deinen Freund, den GoldBug-Chat-Server auf dem Port 80 oder dem Port 443 statt 4710 einzurichten und diesen ggf. mit Login-Daten für einen Echo-Account zu versehen und diese Dir zur Verfügung zu stellen.

Abbildung 22: Volle Ansicht bei der Hinzufügung eines Nachbarn



Wenn Du Deinen GoldBug-Chat über das Tor-Netzwerk betreiben willst, geht dieses ebenso sehr komfortabel, so dass ein Tor-Exit Node nur den Verschlüsselungstext von GoldBug sehen wird.

Hierbei liegt der Chat-Server wieder im normalen Web außerhalb des Tor-Netzwerkes; vereinzelte Teilnehmer der Tor-Community sind gerade dabei, die Installation eines GoldBug Chat-Servers/Listeners innerhalb des Tor-Netzwerkes zu designen.

Da das Echo-Protokoll nicht zwingend einen DHT benötigt, sondern nur eine einfache HTTP-

Verbindung zu einem Nachbarn, die potentiell über das Tor-Netzwerk abgebildet werden kann, ist es eine sehr einfache Architektur, Chat sicher über einen Proxy oder ein Proxy-Netzwerk zu betreiben.

Hier sind ggf. auch Potentiale für weitere Tests, Versuchsbeschreibungen und Dokumentationen gegeben, um die Synergie-Effekte der einzelnen Klienten innerhalb und außerhalb des Netzwerk gemeinsam nach vorne zu bringen und informationstechnologisch zu erforschen.

Wenn Du einen Proxy z.B. in Deiner Firma oder Universität mit dem GoldBug-Messenger nutzen oder austesten willst, dann ist dieses unkritisch, denn es wird eine SSL/TLS- bzw. HTTPS-Verbindung aufgebaut – was für die Proxy-Administratoren kaum unterschiedlich ist wie eine SSL/HTTPS Verbindung zu einer HTTPS-Webseite beim Banking oder dem Einloggen in Dein Web-E-Mail.

Verschlüsselter Traffic bleibt verschlüsselter Traffic und über die Ports 443 oder 80 kann jeder GoldBug Freund erreicht werden.

Kontakt hinzufügen durch Tausch eines Schlüssels

Wie ein Freund hinzugefügt wird und der Schlüssel getauscht wird, ist oben schon erläutert worden. Nachdem im vorherigen Abschnitt die Verbindung zu einem Chat-Server erläutert wurde, bist Du mit zwei grünen LED-Lampen in der Status-Leiste und einem Freund im Chat-Tab normalerweise in der Lage, einen Chat zu beginnen.

Wenn das nicht der Fall ist, prüfe, ob beide Freunde die gleiche Version des Programmes nutzen.

Sodann mag es für die fortgeschrittenen Nutzer darum gehen, mal einen eigenen Chat-Server auszutesten oder über eine direkte Verbindung von Zuhause zu Zuhause zu verbinden und auch den eigenen Router der heimischen Internetverbindung zu definieren.

EMPP Chat Server einrichten

Wenn Du Dich in der minimalen Ansicht befindest, ist ein Chat-Server bzw. Listener ebenso schnell eingerichtet wie im zuvor beschriebenen Tabulator eine Verbindung zu einem Nachbarn hergestellt ist.

Nochmal zur Erinnerung: Im Tabulator "Verbindung herstellen", verbindest Du Deinen GoldBug mit einem anderen Knoten oder Nachbarn, und mit dem Tab "Chat-Server" erstellst Du einen Server bzw. Listener, so dass andere zu Dir verbinden können.

Egal welche Methode, Nachrichten kannst Du immer senden, wenn die zweite oder dritte LED in der Statuszeile leuchtet und ein Nachbar verbunden ist.

Die rechte (dritte) LED in der Statuszeile zeigt also an, dass Du einen eigenen Chat-Server auf Deinem Computer eingerichtet hast.

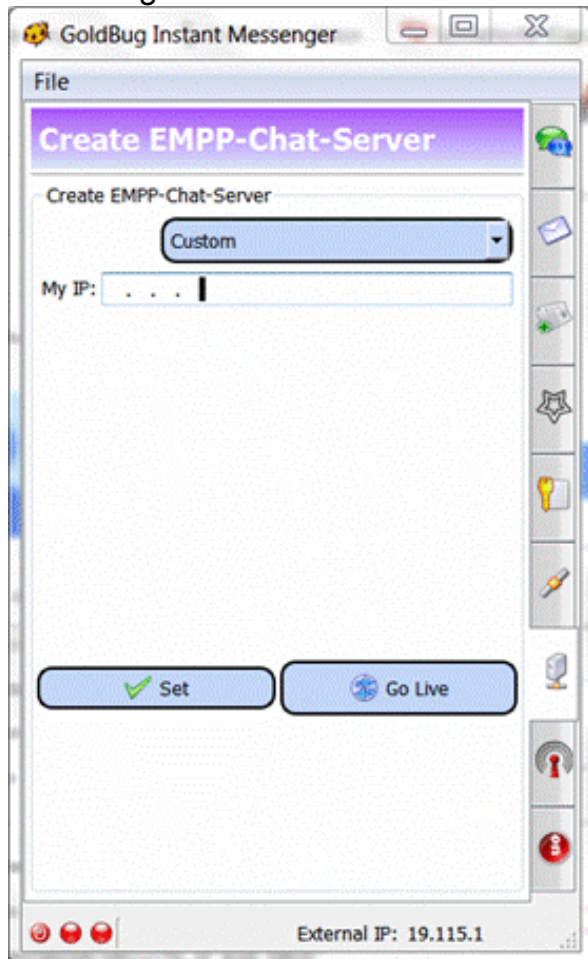
Dazu musst Du die lokale IP-Adresse Deiner Maschine eingeben. Es handelt sich hier nicht um die IP-Adresse des Routers, sondern um die Netz-IP-Adresse des Gerätes, auf dem Du GoldBug Installiert hast. Auch hier erhält man über das Pulldown Menü eine Auswahl und kann die lokale IP wählen. Als Port wird dann wieder automatisch 4710 definiert.

Dücke den Knopf „Set“ und der Eintrag Deines Listeners ist erfolgreich, wenn die dritte LED leuchtet.

Wenn Du einen Klienten hast, der an Deinem Server ist, oder Du im "Verbinde-Nachbar"-Tabulator von Dir aus zu einem anderen Chat-Server oder Freund verbunden bist, kannst Du sodann auch den Kopf „Go Live“ klicken. Damit wird Dein Chat Server über die bestehenden Verbindungen Deinen verbundenen Freunden bzw. Nachbarn sowie auch deren Freunden mitgeteilt. "Go Live" meint also "Broadcast IP+Port" Deines Chat-Servers an Deine Freunde und Nachbarn. Dann können Sie auch zu Deinem Chat-Server automatisch verbinden. Du musst also keine IP-Adresse mehr mitteilen oder die Freunde Deine IP-Adresse manuell eintragen. Alles geht dann automatisch und Dein Server steht Deinen Freunden und deren Freunden als Peer zur Verfügung.

So einfach kann ein Chat-Server erstellt werden.

Abbildung 23: Deine IP-Adresse als Chat-Server konfigurieren



Das Echo Protokoll wird für den Messaging Bereich bzw. für die Chat-Server Erstellung ab und an auch als "EMPP" bezeichnet und steht für "Echoed Messaging and Presence Protocol" - sicherlich auch in Anlehnung zum XMPP Protokoll, das hinsichtlich Verschlüsselung als wenig elaboriert gilt und aufgrund der schlechten Nachrüstbarkeit von Verschlüsselungsmöglichkeiten und -optionen auch bei Kryptologen und Datenschützern auch hinsichtlich der Architektur trotz bestehender Popularität technisch als antiquiert gelten mag.

Wenn Du in der nicht-minimalen Ansicht noch zusätzliche Merkmale definieren willst, ist eine oft genutzte Funktion die des Echo Accounts.

Markiere in der Tabelle den Listener, den Du erstellt hast und gebe dann die Account Credentials ein, also Name und Passwort.

Teile Deinem Freund mit, wie der Accountname und das Passwort dafür lautet und er wird, wenn er den Nachbarkontakt herstellt, über ein Pop-up Fenster gefragt, diese Credentials

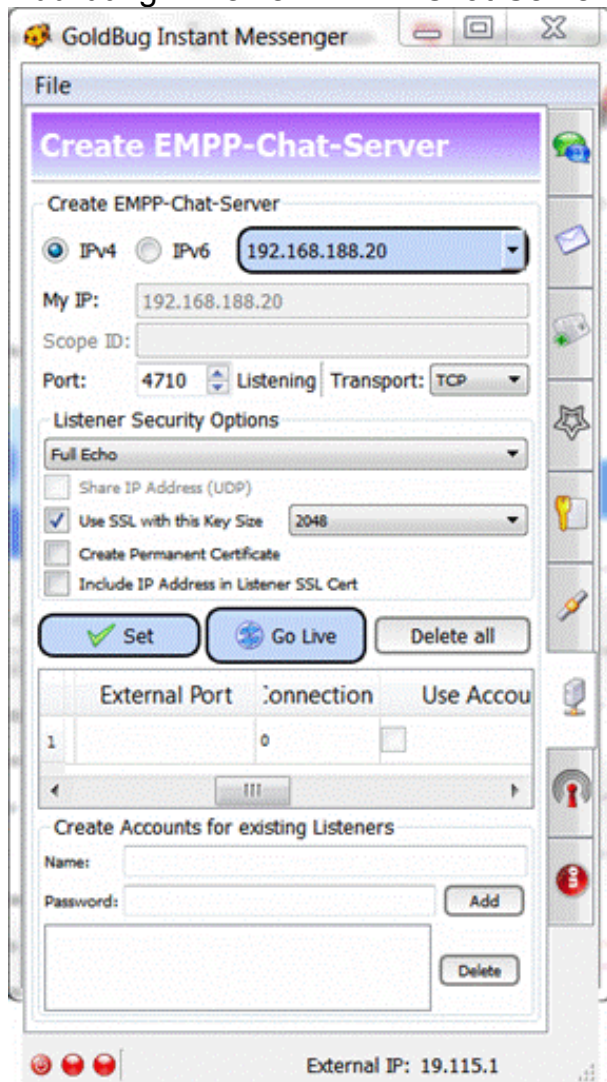
einzugeben.

Ebenso kannst Du auch wieder zwischen IPV4 und IPV6 wählen, wenn Du einen Listener/Chat-Server erstellen willst. Auch können mehrere Chat-Server erstellt werden, indem ein anderer Port gewählt wird. Teste verschiedene Listener mit Port 4710 oder 80 oder 443 und entscheide, ob Du diese Listener für Freunde mit einem Echo Account definieren willst, oder für einfacher aufzubauende Verbindungen im Peer-Modus ohne Account-Login betreibst.

Echo Account definieren, ob Du ein F2F Netzwerk oder ein P2P Netzwerk aufbaust, denn mit den Account Credentials erstellst Du eine Web-of-Trust, mit dem nur Deine vertrauten Freunde mit dem Login-Passwort verbinden können.

Wenn Du einen Peer betreibst, kannst Du z.B. auch auf einer LAN-Party eines geschlossen Netzwerkes mit dem Go-Live Knopf allen Teilnehmern mitteilen, dass Dein Knotenpunkt einen Listener für die Gäste eröffnet hat.

Abbildung 24: einen EMPP-Chat-Server einrichten



Sicherheitsoptionen erlauben in der erweiterten Ansicht bei der Erstellung eines Chat-Servers/Listeners weiterhin die SSL-Schlüsselgröße zu definieren sowie auch ein permanentes SSL-Zertifikat vorzuhalten.

Auch kannst Du – falls Du eine dauerhafte, stabile IP-Adresse hast - diese in das SSL-Zertifikat einbinden.

Diese drei Maßnahmen machen es Angreifern schwerer, das SSL-Zertifikat auszutauschen oder zu faken – denn es würde sofort erkannt, wenn ein untergeschobenes anderes Zertifikat sich als das originäre ausgeben wollte: weil z.B. der Klient kein neues, sondern das alte, permanente Zertifikat erwartet oder weil die IP-Adresse darin fehlt oder nicht stimmig ist. Auch

die SSL-Schlüsselgröße definiert dieses.

Erstellung eines Servers/Listeners Zuhause hinter einem Router / Nat:

Wenn Du keinen Webserver hast oder keinen allgemeinen Nachbarn im Web findest, kannst du auch einen Chat-Server zuhause hinter Deinem Router einrichten. Dein Freund muss das dann nicht, er kann direkt als Klient zu Deinem Listener verbinden. Aber einer von beiden muss einen Listener erstellen. Wenn Du dieses hinter Deinem Router/Nat zuhause machen willst, nimm wie geannt die lokale IP-Adresse der Maschine für den Listener z.B. 192.168.121.1.. Sodann muss Du in Deinem Router auch den Port weiterleiten, d.h. Port 4710 muss vom Router weitergeleitet werden an 192.168.121.1: 4710. Sodann muss der Kernel - Spot-on-Kernel.exe - sowie auch die GoldBug.exe in Deiner Windows Firewall erlaubt sein. Wenn Du alles korrekt weitergeleitet hast, kann der Freund an Deiner (externen) IP-Adresse des Routers (siehe z.B. unter www.whatismyip.com) und dem Port 4710 seinen Klienten verbinden.

Wichtig ist nur, dass Dein Router den Kontaktversuch aus dem Internet am definierten Port an Deine lokale Maschine weiterleitet.

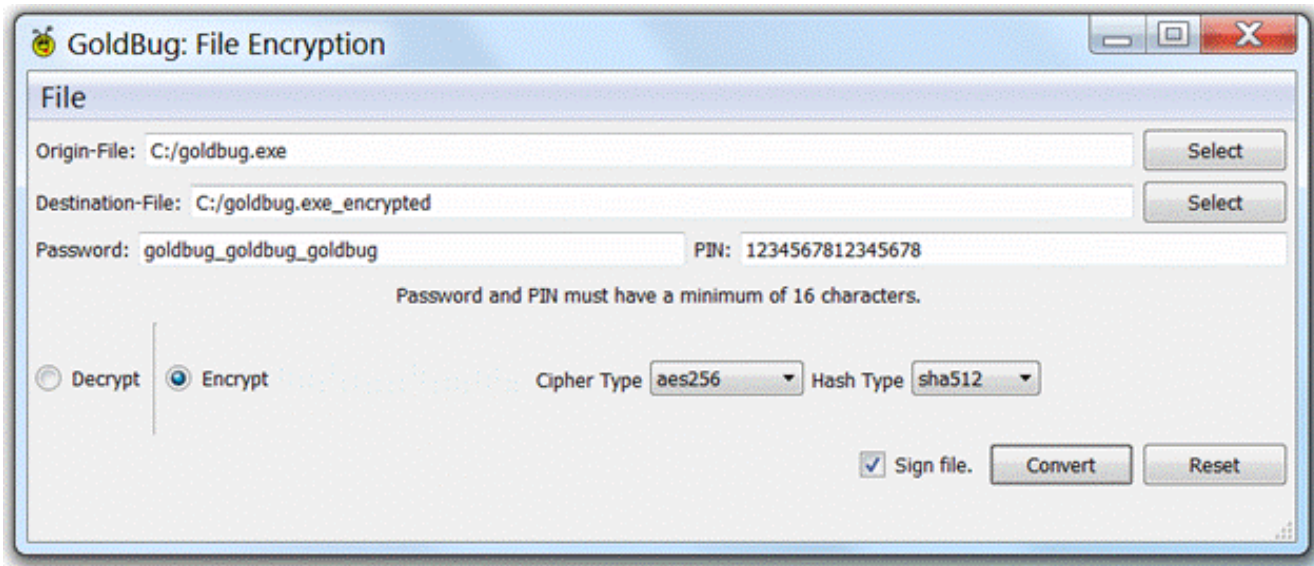
Dieses ist ein übliches und sicheres Verfahren und öffnet kein beliebigen Zugang zu Deinem Rechner, sondern über den Port und die Applikation wird dabei wie bei vielen anderen Programmen definiert, dass auch nur Packet in diesem Sinne zugelassen werden.

Du kannst und musst dieses alles selbst definieren und GoldBug enthält keinen Code, der automatisch Ports im Router weiterleitet, oder öffnet oder gar automatisch einen Listener einrichtet. Somit ist es sicherer und bedarfsgerechter als andere Applikationen, die im Sinne der Nutzerfreundlichkeit sich selbst konfigurieren und diese Mühe zwar abnehmen, jedoch auch vielen Nutzern, die die technischen Details der Portweiterleitung, Port-Öffnung und Listener-Definiton kennen, unwissend als Standardeinstellung anbieten. Wenn Du also das erste Mal davon hörst, sei gewiss, dass andere Programme dieses oftmals alles automatisch einstellen und die Tatsache, dass das vorliegende Programm diese Optionen als manuelle Einstellungen durch Dich selbst ermöglicht, sollte Dich nicht abschrecken, es einmal auszuprobieren und in die von Dir eingestellte Technik zu vertrauen, weil sie nunmal so funktioniert wie beschrieben hinsichtlich Port-Freigabe, ggf. Port-Weiterleitung und Einrichtung eines Listeners.

Werkzeuge: Verschlüsselung von Dateien

GoldBug verfügt über zusätzliche Werkzeuge für die Verschlüsselung. Im Hauptmenü unter Werkzeuge findest Du das Werkzeug für die Verschlüsselung von Dateien auf Deiner Festplatte ("File Encryption Tool")

Abbildung 25: Werkzeug zur Datei-Verschlüsselung



Damit kannst Du eine Datei von der Festplatte bestimmen, sodann den gleichen Pfad angeben und eine beliebige Endung bzw. Änderung des Dateinamens wählen - sodann Passwort und Pin eingeben (beides natürlich wieder mindestens 16 Zeichen) und mit den Radio-Auswahl-Knöpfen definieren, ob die Datei ver- oder ent-schlüsselt werden soll. Cipher- und Hash-Type sind ebenso definierbar wie auch eine Signatur in die Verschlüsselung optional eingebaut werden kann, damit sichergestellt wird, dass die Verschlüsselung auch durch Dich erfolgte (und niemanden anderes).

Das Werkzeug zur Dateiverschlüsselung ist ein Angebot, um z.B. potentiell unsichere Truecrypt Container zu ersetzen bzw. ergänzend zu verschlüsseln oder um einzelne Dateien zu sichern, bevor Du sie transferierst - sei es als E-Mail in GoldBug, über StarBeam in GoldBug oder auch über konventionelle, unsichere Wege - oder einfach, um sie auf Deiner Festplatte oder bei Speicherung in Online-Speichern wie Dropbox oder Megaupload zuvor zu verschlüsseln.

Werkzeuge: Das Rosetta CryptoPad

Das Werkzeug Rosetta Crypto Pad hat seinem Namen vom [Stein of Rosette](#), der in London im Museum steht (siehe Wikipedia). Er gilt als Übersetzungshilfe für ägyptische Hyroglyphen in weitere Sprachen.

Das in GoldBug enthaltene Rosetta Cryptopad besteht aus zwei Schüsseln - wie auch Chat und E-Mail derartig eigene Schlüssel haben. Tausche auch hier mit einem Freund den Rosetta-Schlüssel, gebe Text ein, wähle den Freund und ob es sich um Ver- oder Entschlüsselung handelt - und drücke den Knopf "konverieren".

Sodann wird unten der Output angezeigt und diesen kannst Du einfach mit der Kopierfunktion auskopieren und über konventionelle Online-Kommunikationswege wie @-E-Mail oder einen anderen Chat versenden. Slow-Chat durch manuelle Verschlüsselung Deines Chat-Textes.

Es ist eine Alternative zu GnuPG (bzw. basiert es ja ebenso auf der GnuPG zugrunde liegenden Bibliothek Libgcrypt).

Abbildung 26: Das Rosetta-CryptoPad



Further Implementations & GB Features

- **Accounts:** Enter your password to the account, it is not transferred to the server, just a hash comparison is done on both sides.
- All data on your hard disk (.db files) is strong encrypted.
- **Gemini** (end-to-end encryption key) is secured by a **MAC Gemini Hash**.
- Secure Key Transfer: **Repleo** encrypts your public key.
- Chat over Tor with GoldBug.
- **Instant Forward Secrecy** with MELODICA Button: Change the encryption key end to end whenever you want.
- Set an **additional password for emails** (based on AES).
- Send p2p Emails to offline friends.
- **Email-Signatures:** Decide, if you want to send and receive authenticated emails or just non-authenticated.
- **StarBeam (SB):** Transmit your file into a network of encrypted packets anonymously. TCP & UDP transport for the echo protocol: UDP is ideal for echoed VoIP.

Further Development

- Spot-On is the underlaying library for the GoldBug Instant Messenger.
- Spot-On has as well a gui and is full of adjustable options, GoldBug aims to be a desktop/mobile messenger with a smaller set of options to fit mobile or tablet devices.
- Spot-On is a c++ library as an exploratory research project investigating an encrypted communication and data transfer protocol, called the "echo protocol" or short "EMPP" protocol: Echoed Message and Presence Protocol. The package which includes the 'libspot-on' library, is found here: spot-on.sf.net

It enables personal and group messaging, decentral p2p email, echoed IRC/Buzz Chat Channels and secure Filetransfer with multi-encryption (SSL, RSA (PGP / GnuPG) / ElGamal, AES, libgcrypt, OpenSSL etc). IP Addresses are detached from Encryption Keys. It is programmed in c++ and is the underlaying library for chat, email and messaging applications like the GoldBug Instant Messenger App.

Spot-On can be deployed by every c-developer into chat and filesharing apps.

Liste an möglichen Kriterien für weitergehende Evaluationen

1. Tiered application: kernel and user interface processes.
2. Use proxy capabilities?
3. Send email messages to offline friends?
4. Send email with encrypted attachments?
5. Having different Keys for Chat, Email, Cryptopad, Filetransfer etc.?
6. Is the key stuck to your IP Address?
7. Mutual access authentication?
8. No hashing of a file and sending it with hash and senders/receivers ID to neighbors, so it is identifiable?
9. Are there alternatives to RSA, like ElGamal or NTRU? Can a NTRU-user chat to a RSA-user?
10. You can use SSL or not? Selectable SSL ciphers?
11. Selectable hash algorithms?
12. Just need connectivity, no key exchange, keys are optional?
13. You are more autonomous?
14. Trust is not needed, or can be added as you define it?
15. Technical simplicity?
16. Anonymous seeds?
17. You cannot determine, who is reading which message (as you have no destination ID or info added)?
18. Free of Web of Trust-Graphs and no mapping of connections ?
19. Its different, its fun?
20. Local database stores all info in encrypted .db' s?
21. Re-encode support of locally-encrypted data.
22. Optional authentication of messages ?
23. You can communicate without public keys, using Magnets ?
24. Support for TCP and UDP and SCTP communications?
25. Support the multi-layer of encryption
26. Having multi encryption? e.g. SSL + RSA + AES ? Or even Ciphertext over SSL + RSA +

AES (Rosetta-Cryptopad ciphertext sent over encrypted channels)?

27. Multiple listeners are possible?
 28. A kernel is given? Multi-threaded?.
 29. IRC-like channels?
 30. Simple IP-based firewalls?
 31. You can define many points of connections?
 32. Do scramblers send out fake messages?.
 33. You can store messages in friends ?
 34. You have the option to use an end-to-end key for communication?
 35. You have the option to renew the end-to-end key each time you want (not only session based)?
 36. Encrypted file transfer protocol (StarBeam)?
 37. Using a one time magnet (OTM) for a crypto channel?
 38. Having ipv6 support?
 39. Having Qt 5 and up deployed ?
 40. Hops are not forwarding, no routing, is it always a wrap the message new and send to just to your friend? router-less and forwarding-less protocol?
 41. Sending a message to a friend to his dedicated connection and not to all connections?
 42. Hiding the key exchange online?
 43. Use several encryption keys on one filetransfer?
 44. Adding a passphrase on a file transfer ?
 45. Use it as client without a listener?
- ... over 40 criteria, someone could analyse and write about in her/his master thesis - with these different implementations in different tools compared.

The digital encryption of your private communication in the context of ...

Principles of the protection of private speech, communication and life:

Universal Declaration of Human Rights, 1948 (Art. 12)

No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.

<http://www.un.org/en/documents/udhr/index.shtml#a12>

http://en.wikipedia.org/wiki/Universal_Declaration_of_Human_Rights

International Covenant on Civil and Political Rights, 1966 (Art. 17)

1. No one shall be subjected to arbitrary or unlawful interference with his privacy, family, home or correspondence, nor to unlawful attacks on his honour and reputation.

2. Everyone has the right to the protection of the law against such interference or attacks.

<http://www.ohchr.org/EN/ProfessionalInterest/Pages/CCPR.aspx>

http://en.wikipedia.org/wiki/International_Covenant_on_Civil_and_Political_Rights

European Convention on Human Rights, 1950 (Art. 8)

1. Everyone has the right to respect for his private and family life, his home and his correspondence.

2. There shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others.

<http://conventions.coe.int/treaty/en/Treaties/Html/005.htm>

http://en.wikipedia.org/wiki/European_Convention_on_Human_Rights

Charter of Fundamental Rights of the European Union, 2000 (Art. 7, 8)

Article 7. Respect for private and family life

Everyone has the right to respect for his or her private and family life, home and communications.

Article 8. Protection of personal data

1. Everyone has the right to the protection of personal data concerning him or her.

2. Such data must be processed fairly for specified purposes and on the basis of the consent of the person concerned or some other legitimate basis laid down by law. Everyone has the right of access to data which has been collected concerning him or her, and the right to have it rectified.

3. Compliance with these rules shall be subject to control by an independent authority.

http://en.wikisource.org/wiki/Charter_of_Fundamental_Rights_of_the_European_Union

http://en.wikipedia.org/wiki/Charter_of_Fundamental_Rights_of_the_European_Union

Basic Law e.g. for the Federal Republic of Germany, 1949 (Art. 2 Abs. 1 i. V. m. Art. 1 Abs. 1)

Article 2 [Personal freedoms]

(1) Every person shall have the right to free development of his personality insofar as he does not violate the rights of others or offend against the constitutional order or the moral law.

Article 1 [Human dignity – Human rights – Legally binding force of basic rights]

(1) Human dignity shall be inviolable. To respect and protect it shall be the duty of all state authority.

<https://www.btg-bestellservice.de/pdf/80201000.pdf>

http://en.wikipedia.org/wiki/Basic_Law_for_the_Federal_Republic_of_Germany

Further: Article 1 and Article 10:

Art. 1 [Human dignity – Human rights – Legally binding force of basic rights]

(1) Human dignity shall be inviolable. To respect and protect it shall be the duty of all state authority.

(2) The German people therefore acknowledge inviolable and inalienable human rights as the basis of every community, of peace and of justice in the world.

(3) The following basic rights shall bind the legislature, the executive and the judiciary as directly applicable law

Art. 10 [Privacy of correspondence, posts and telecommunications].

Secrecy of correspondence - Fernmeldegeheimnis (Art. 10 Abs. 1

Grundgesetz)

§ 88 Abs. 1 Fernmeldegeheimnis - Telekommunikationsgesetz:

(1) Dem Fernmeldegeheimnis unterliegen der Inhalt der Telekommunikation und ihre näheren Umstände, insbesondere die Tatsache, ob jemand an einem Telekommunikationsvorgang beteiligt ist oder war. Das Fernmeldegeheimnis erstreckt sich auch auf die näheren Umstände erfolgloser Verbindungsversuche.

(2) Zur Wahrung des Fernmeldegeheimnisses ist jeder Diensteanbieter verpflichtet. Die Pflicht zur Geheimhaltung besteht auch nach dem Ende der Tätigkeit fort, durch die sie begründet worden ist.

(3) Den nach Absatz 2 Verpflichteten ist es untersagt, sich oder anderen über das für die geschäftsmäßige Erbringung der Telekommunikationsdienste einschließlich des Schutzes ihrer technischen Systeme erforderliche Maß hinaus Kenntnis vom Inhalt oder den näheren Umständen der Telekommunikation zu verschaffen. Sie dürfen Kenntnisse über Tatsachen, die dem Fernmeldegeheimnis unterliegen, nur für den in Satz 1 genannten Zweck verwenden. Eine Verwendung dieser Kenntnisse für andere Zwecke, insbesondere die Weitergabe an andere, ist nur zulässig, soweit dieses Gesetz oder eine andere gesetzliche Vorschrift dies vorsieht und sich dabei ausdrücklich auf Telekommunikationsvorgänge bezieht. Die Anzeigepflicht nach § 138 des Strafgesetzbuches hat Vorrang.

(4) Befindet sich die Telekommunikationsanlage an Bord eines Wasser- oder Luftfahrzeugs, so besteht die Pflicht zur Wahrung des Geheimnisses nicht gegenüber der Person, die das Fahrzeug führt oder gegenüber ihrer Stellvertretung.

§ 206 Verletzung des Post- oder Fernmeldegeheimnisses

(1) Wer unbefugt einer anderen Person eine Mitteilung über Tatsachen macht, die dem Post- oder Fernmeldegeheimnis unterliegen und die ihm als Inhaber oder Beschäftigtem eines Unternehmens bekanntgeworden sind, das geschäftsmäßig Post- oder Telekommunikationsdienste erbringt, wird mit Freiheitsstrafe bis zu fünf Jahren oder mit Geldstrafe bestraft.

(2) Ebenso wird bestraft, wer als Inhaber oder Beschäftigter eines in Absatz 1 bezeichneten Unternehmens unbefugt 1.

eine Sendung, die einem solchen Unternehmen zur Übermittlung anvertraut worden und verschlossen ist, öffnet oder sich von ihrem Inhalt ohne Öffnung des Verschlusses unter Anwendung technischer Mittel Kenntnis verschafft, 2. eine einem solchen Unternehmen zur Übermittlung anvertraute Sendung unterdrückt oder 3. eine der in Absatz 1 oder in Nummer 1 oder 2 bezeichneten Handlungen gestattet oder fördert.

(3) Die Absätze 1 und 2 gelten auch für Personen, die 1. Aufgaben der Aufsicht über ein in Absatz 1 bezeichnetes Unternehmen wahrnehmen, 2. von einem solchen Unternehmen oder mit dessen Ermächtigung mit dem Erbringen von Post- oder Telekommunikationsdiensten betraut sind oder 3. mit der Herstellung einer dem Betrieb eines solchen Unternehmens dienenden Anlage oder mit Arbeiten daran betraut sind.

(4) Wer unbefugt einer anderen Person eine Mitteilung über Tatsachen macht, die ihm als außerhalb des Post- oder Telekommunikationsbereichs tätigem Amtsträger auf Grund eines befugten oder unbefugten Eingriffs in das Post- oder Fernmeldegeheimnis bekanntgeworden sind, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft.

(5) Dem Postgeheimnis unterliegen die näheren Umstände des Postverkehrs bestimmter Personen sowie der Inhalt von Postsendungen. Dem Fernmeldegeheimnis unterliegen der Inhalt der Telekommunikation und ihre näheren Umstände, insbesondere die Tatsache, ob jemand an einem Telekommunikationsvorgang beteiligt ist oder war. Das Fernmeldegeheimnis erstreckt sich auch auf die näheren Umstände erfolgloser Verbindungsversuche.

http://www.gesetze-im-internet.de/gg/art_10.html
http://en.wikipedia.org/wiki/Secrecy_of_correspondence
<http://de.wikipedia.org/wiki/Briefgeheimnis>
<http://de.wikipedia.org/wiki/Fernmeldegeheimnis>
<http://de.wikipedia.org/wiki/Postgeheimnis>
http://www.gesetze-im-internet.de/tkg_2004/__88.html
http://www.gesetze-im-internet.de/stgb/__206.html

United States Constitution: Search and Seizure (Expectation of Privacy, US Supreme Court)

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

<http://www.usconstitution.net/const.html>

Webseite

Weitere Informationen finden sich auf der Webseite:

<http://goldbug.sf.net>