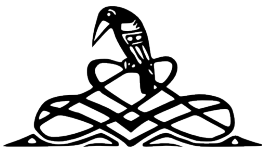
Seminararbeit



Ignaz – Taschner – Gymnasium – Dachau

RAHMENTHEMA DES WISSENSCHAFTSPROPÄDEUTISCHEN SEMINARS:

Codierung und Verschlüsselung von Informationen

LEITFACH:

Mathematik

KURSLEITER/KURSLEITERIN:

Frau Lutz

THEMA DER ARBEIT:

Off-the-Record Messaging am Beispiel des Extensible Messaging and Presence Protocol (XMPP)

VERFASSER / VERFASSERIN:

Michael Kurz

Bewertung	Note	Note in Worten	Punkte	
Schriftliche Arbeit			**x** 3	
Abschlusspräsentation			**x** 1	
			Summe:	
			Gesamtleistung nach § 61 (7) GSO = Summe:2 (gerundet)	

Datum und Unterschrift der Kursleiterin bzw. des Kursleiters

Off-the-Record Messaging

am Beispiel des Extensible Messaging and Presence Protocol (XMPP)

Michael Kurz

Ignaz Taschner Gymnasium Dachau

3. November 2014

Inhaltsverzeichnis

1 Einleitung **1**

2 Instant Messaging **2**
2.1 Proprietäre Protokolle 2
2.2 XMPP 3

3 Off-the-Record Messaging **4**
3.1 Allgemeine Informationen 4
3.2 Technische Umsetzung 4
3.2.1 Diffie-Hellman-Schlüsselaustausch 4
3.2.2 Digital Signature Algorithm 5
3.2.3 Advanced-Encryption-Standard 5
3.2.4 Secure Hash Algorithm 5

4 Praktischer Teil **6**
4.1 Pidgin 6
4.1.1 Installation 6
4.1.2 Einrichten eines Kontos 7
4.1.3 Kontakte hinzufügen 10
4.2 OTR Plugin 11
4.2.1 Installation 11
4.2.2 Konfiguration 11
4.2.3 Starten einer sicheren Konversation 13

5 Schluss **16**

Abbildungen **17**

Literatur **17**

1 Einleitung

Im Bereich der Wissenschaft und Technologie finden Entwicklungen schneller und umfassender denn je statt. Dabei passiert es häufig, dass neuartige Probleme auftauchen, welche es zu bewältigen gilt. Ein Beispiel hierfür ist das Thema Datenschutz und Sicherheit im Internet, welchem besonders nach der Globalen Überwachungs- und Spionageaffäre Aufmerksamkeit geschenkt wird. Kommunikation findet im Internet häufig über sogenannte Instant Messaging Dienste statt und bis heute sind ausgereifte Verschlüsselungstechnologien bei solchen Diensten selten, sind sie doch durchaus notwendig. Denn findet die Kommunikation unverschlüsselt statt, ist es ein Leichtes für Dritte mit etwas technischem Geschick Gespräche abzuhören oder zu manipulieren.

Im Laufe dieser Arbeit wird das Verschlüsselungsverfahren Off-the-Record Messaging besonders am Beispiel des Extensible Messaging and Presence Protocol erklärt. Am Ende der Arbeit soll man außerdem in der Lage sein, sich selbst einen XMPP-Account einzurichten, einen Client zu installieren und eine sichere Konversation mittels Off-the-Record Messaging zu führen.

2 Instant Messaging

Beim Instant Messaging (IM) handelt es sich um eine Form der Online-Kommunikation in der Textnachrichten in (nahezu) Echtzeit übermittelt werden, besser bekannt als Chat. Es gibt viele verschiedene IM-Dienste/Protokolle, wie z.B. ICQ, Skype, XMPP, IRC, usw. Die meisten dieser Dienste, wie auch XMPP, basieren auf dem Client-Server-Modell. Ein Client ist eine Software, welche die Schnittstelle zwischen dem Server, der den Dienst zur Verfügung stellt, und dem Benutzer darstellt. Ein Client kann also mit bestimmten Protokollen (Diensten) umgehen und stellt für den Nutzer wichtige Funktionen zur Verfügung. Bekannte Beispiele für Clients sind alle Webbrowser, welche hauptsächlich für das HTML-Protokoll konzipiert sind, E-Mail Programme, oder eben auch IM-Clients, welche oft den selben Namen tragen, wie das Protokoll (ICQ, Skype, etc.). Ein Protokoll ist dafür zuständig, die Kommunikation zwischen Client und Server zu reglementieren. Es legt geanu fest, wie die Anfrage vom Client und die Antwort vom Server aufgebaut sind. Folgende Grafik veranschaulicht nochmal die Zusammenhänge.

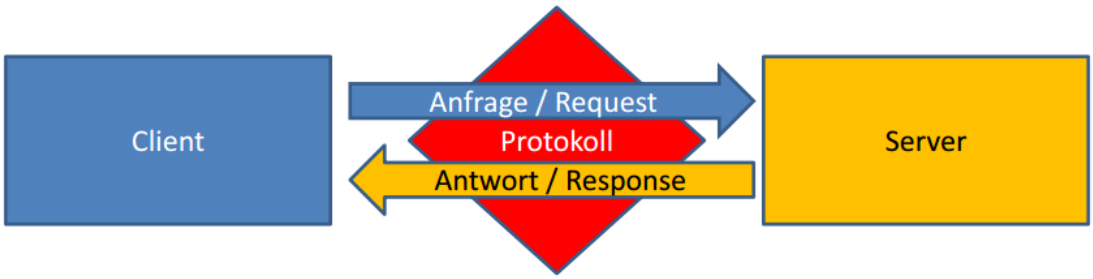


Abbildung 2.1: Client-Server-Modell [2, S. 4]

Üblicherweise hat man beim Instant Messaging eine Kontaktliste, zu der man Nutzer des jeweiligen Dienstes hinzufügen, und schließlich mit ihnen chatten kann. Nahezu alle IM-Dienste verfügen über die Funtkion, Verfügbarkeitsstatus zu setzen, also z.B. Online, Abwesend, Beschäftigt, etc.

2.1 Proprietäre Protokolle

Die meisten IM-Protokolle sind proprietär und nicht offen, wie XMPP. Das bedeutet, dass sie von Firmen entwickelt wurden um einen eigenen IM-Dienst anzubieten. Der Quellcode, welcher auch jederzeit geändert werden kann, ist nicht einsehbar und somit die genaue Funktionsweise des Dienstes unbekannt. Folglich kann auch meist nur der Messenger (Client) der Entwicklerfirma genutzt werden. Ein weiteres, großes Problem proprietärer Protokolle ist die Serverzentralität, was bedeutet, dass jeder Datenverkehr über Server der Firma stattfindet. Man ist also gezwungen, dieser zu vertrauen, wenn man den Dienst nutzt.

2.2 XMPP

Es wird sich in diesem Abschnitt besonders nach den Angaben auf der Entwicklerseite gerichtet. [3]. Das Extensible Messaging and Presence Protocol, auch bekannt unter dem Namen Jabber, ist ein **offenes** und nach RFC 6120, RFC 6121 und RFC 6122 **standardisiertes** Protokoll. Offen bedeutet, dass der Quellcode für jedermann einsehbar und auch anpassbar, also Open Source ist. Abgesehen von diesen, weist XMPP noch weitere Leistungsmerkmale auf, welche das Konzept von XMPP verdeutlichen:

Proven 1998 begann Jeremie Miller an XMPP zu arbeiten und das Protokoll wird stetig weiterentwickelt, was dazu führt, dass es heute ein hohes Maß an Stabilität aufweist. Viele Menschen nutzen XMPP durch öffentliche Dienste, wie ehemals Google Talk oder auch den Facebook-Chat.

Decentralized Im Vergleich zu anderen Diensten kann jeder seinen eigenen XMPP-Server betreiben und man ist nicht darauf angewiesen, einem Unternehmen zu vertrauen. Das Prinzip ist dem von E-Mail ähnlich.

Secure Ein XMPP-Server kann auch in geschlossenen Netzwerken, wie einem Intranet, betrieben werden. Außerdem gehören SASL und TLS[1] zu den Kernspezifikationen.

Extensible Dadurch, dass XMPP XML[2] nutzt, kann das Protokoll jederzeit nach belieben erweitert werden. Viele Erweiterungen wurden bereits in der XMPP-Extension-Protocols (XEP) Serie veröffentlicht.

Flexible XMPP stellt nicht nur einen reinen IM-Dienst dar. Vielmehr Anwendungsmöglichkeiten, wie z.B. Spiele, das verschicken von Dateien, usw. sind gegeben.

Diverse Aufgrund der zahlreichen Möglichkeiten, die XMPP bietet, findet es in den verschiedensten Situationen, Projekten, Unternehmen, etc. Anwendung.

[1] Sicherheitsstandards für Authentifizierung (SASL) und Verschlüsselung (TLS) im Internet
[2] weit verbreitete Auszeichnungssprache

3 Off-the-Record Messaging

3.1 Allgemeine Informationen

Off-the-Record Messaging (OTR) ist der Name eines Protkolls, welches eine sichere IM-Kommunikation zwischen zwei Parteien garantiert. Meist findet das Protokoll Verwendung in Verbindung mit XMPP, da hier günstige Grundvoraussetzungen gegeben sind um OTR zu implementieren. Das Ziel von OTR ist es, ein Vier-Augen-Gespräch mit den entsprechenden Bedingungen zu simulieren. Dies spiegelt sich in den von den Entwicklern festgelegten Leistungsmerkmalen wieder. Auch hier wird sich hauptsächlich nach den Angaben auf der Entwicklerseite gerichtet. [5]

Encryption Die IM-Nachrichten werden verschlüsselt, damit kein anderer die Nachrichten lesen kann

Authentication Es wird sichergestellt, dass die jeweils andere Partei auch die ist, für die man sie hält.

Deniability Während der OTR-Sitzung ist zwar sichergestellt, dass die Nachrichten authentisch, also unverändert sind, jedoch kann nach der Sitzung nicht mehr sichergestellt werden, dass eine Nachricht auch wirklich echt ist. Sie kann nachträglich ohne weiteres manipuliert werden. Das Gespräch bleibt sozusagen folgenlos.

Perfect forward secrecy *„Wenn der (langlebige) private Schlüssel einem Dritten in die Hände fällt, hat dies keine Auswirkung auf die Kompromittierung bisher getätigter Gespräche: Die Gespräche können damit nicht nachträglich entschlüsselt werden.“* [6, Ziele des Projektes, Folgenlosigkeit]

3.2 Technische Umsetzung

Um diese Ziele zu verwirklichen, werden bestehende, bewährte Algorithmen gezielt miteinander kombiniert, woraus dann das Off-the-Record Messaging Protokoll zustande kommt. Die einzelnen Alogrithmen und ihre Funktion werden im Folgenden knapp erläutert.

3.2.1 Diffie-Hellman-Schlüsselaustausch

Der Diffie-Hellman-Schlüsselaustausch ermöglicht, dass sich zwei Parteien auf einen geheimen Schlüssel einigen können, obwohl ihnen zunächst kein sicherer Kanal zur Verfügung steht. Alice und Bob einigen sich auf einen öffentlichen Wert den auch Eve, der potentielle Abhörer, theoretisch wissen kann. Zusätzlich wählen Alice und Bob jeweils für sich noch einen privaten Wert, den sie für sich behalten. Nun werden jeweils die privaten Werte mit dem einen öffentlichen Wert wie im Algorithmus vorgesehen miteinander verrechnet. Dabei findet das Prinzip der Einwegfunktion Verwendung. Alice und Bob tauschen nun diese neuen Werte aus und verrechnen den Wert der jeweils anderen Person wieder mit ihren privaten Werten. Dadurch entsteht ein neuer, gemeinsamer und für Eve unbekannter Wert. Dieser Wert ist nun der Schlüssel für eine symmetrische Verschlüsselung und

wird jede Sitzung erneut ausgehandelt. Sobald dies geschehen ist, werden die privaten Werte von Beginn sofort verworfen, was dann *„Perfect Forward Secrecy“* garantiert.

3.2.2 Digital Signature Algorithm

Eine digitale Signatur gewährleistet die Echtheit einer Nachricht und hat somit den selben Zweck wie eine Unterschrift. Im Prinzip funktioniert das so, dass Alice einen privaten und einen öffentlichen Schlüssel besitzt. Sie kann nun eine digitale Signatur erstellen, indem sie die Nachricht mit ihrem privaten Schlüssel verrechnet. Die Integrität der Signatur kann nun mithilfe des öffentlichen Schlüssels überprüft werden [8, S. 205]. Der Digital Signature Algorithm (DSA) wurde vom National Institute of Standards and Technology (NIST) und der National Security Agency (NSA) entwickelt und gilt derzeit als sicheres Verfahren für digitale Signaturen. Auch hier findet das Prinzip der Einwegfunktion Anwendung.

3.2.3 Advanced-Encryption-Standard

Der Advanced-Encryption-Standard (AES) ist als Gewinner eines Wettbewerbs des NIST der Nachfolger des Data-Encryption-Standard (DES). Er gilt nicht nur als sicher, sondern überzeugte auch weiterhin durch leichte Hard- und Softwareimplementierung und schnelle Performance. Der Algorithmus ist Open Source und stellt ein symmetrisches Verschlüsselungsverfahren dar.

3.2.4 Secure Hash Algorithm

Eine Hashfunktion berechnet aus einem beliebigen Inhalt einen einmaligen HashWert. Die Funktion kann nicht rückgängig gemacht werden. Jeder Hashwert hat, ungeachtet des Inhalts, die selbe Länge und eignet sich z.B. als Prüfsumme für verschiedene Anwendungen. Der Secure Hash Algorithm (SHA-1) erzeugt Hashwerte mit einer länge von 160 bit und wird beim OTR-Protokoll als Message Authentication Code (MAC) verwendet um einzelne Nachrichten zu authentifizieren.

4 Praktischer Teil

In diesem Kapitel wird ausführlich erklärt, wie man selbst auf einem Computer mit dem Betriebssystem Windows einen XMPP-Account anlegt, einen passenden Client installiert und OTR konfiguriert.

4.1 Pidgin

Pidgin ist ein Open Source Multi-Protokoll-Client, was bedeutet, dass der Client neben XMPP noch viele weitere IM-Protokolle unterstützt.

4.1.1 Installation

Das Installations-Setup[3] kann von der Entwicklerseite *https://pidgin.im/* heruntergeladen werden.

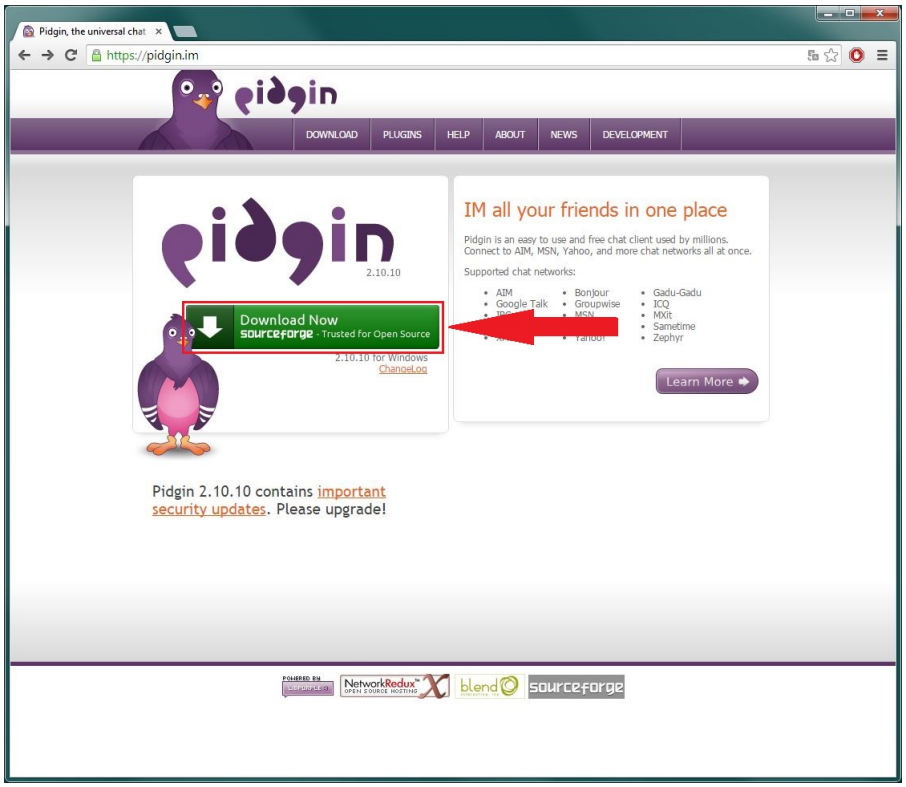


Abbildung 4.1: Webseite der Pidgin-Entwickler

Anschließend wird das Installations-Setup gestartet und das Programm einfach installiert. Dann wird ein beliebiger XMPP-Server ausgewählt. Eine Liste solcher Server findet man beispielsweise unter `https://xmpp.net/directory.php`

[3]Hier wird Version 2.10.10 verwendet

4.1.2 Einrichten eines Kontos

Wird das Programm geöffnet, erscheint zunächst ein Startdialog. Wie in diesem beschrieben, wird die Schaltfläche **Hinzufügen...** gewählt um ein neues IM-Konto zu konfigurieren.

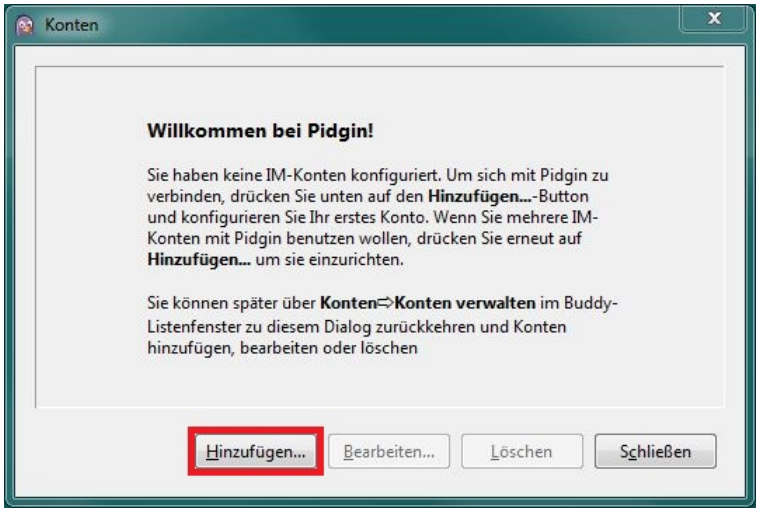


Abbildung 4.2: Startdialog

Nun öffnet sich das Fenster zum Konfigurieren des Kontos, das hinzugefügt werden soll. Wenn man noch keine XMPP-Account besitzt, wird nun ein neuer angelegt. Die Maske wird nach dem in Abbildung 4.7 gezeigten Schema ausgefüllt. Das Protokoll ist XMPP, der Benutzername beliebig und die Domain ist der vorher ausgesuchte Server. Natürlich wird der Account durch ein beliebiges Passwort geschützt. Wichtig ist, die Checkbox **Dieses neue Konto auf dem Server anlegen** zu aktivieren, wenn man das Konto noch nicht registriert hat. Sind alle benötigten Felder ausgefüllt, wird die Schaltfläche **Hinzufügen** gewählt. Je nach gewähltem Server kann es sein, dass sich ein weiterer Registrierungs-Dialog in irgendeiner Form öffnet. Ist dies der Fall, wird der Registrierungsvorgang hier durchgeführt (siehe Abbildung 4.4).

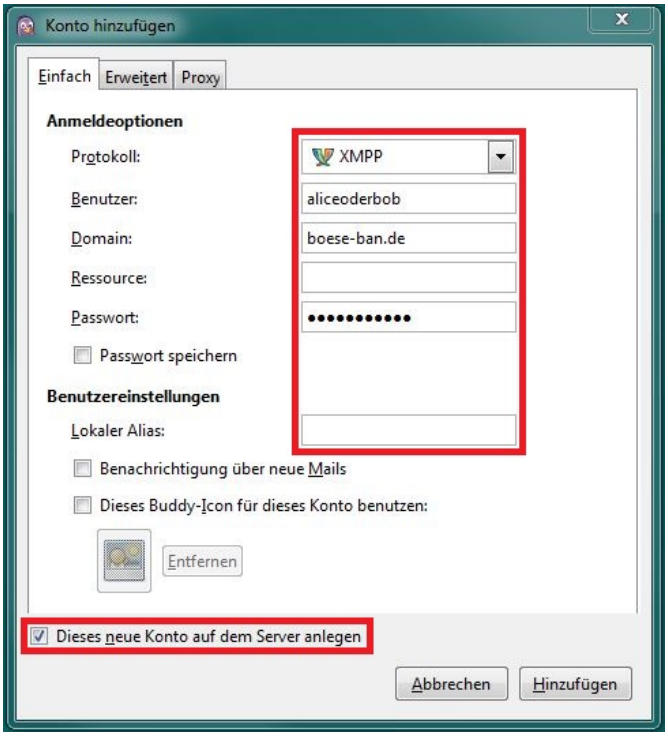


Abbildung 4.3: Fenster zum konfigurieren eines neuen Kontos

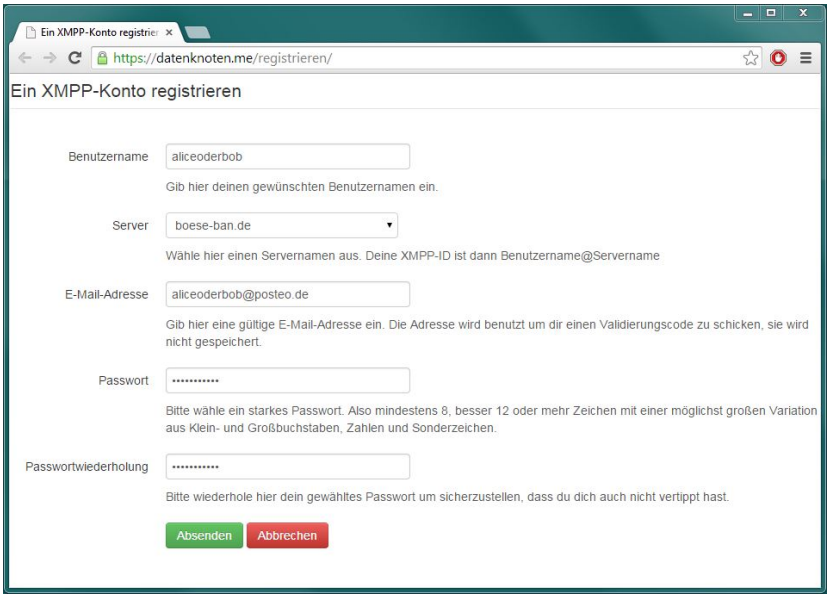


Abbildung 4.4: Externer Registrierungsdialog

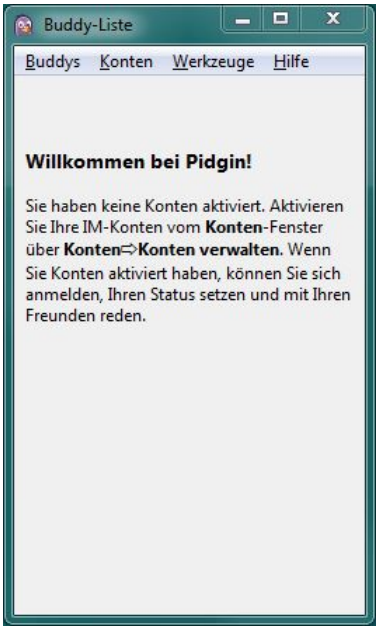


Abbildung 4.5: Hauptfenster

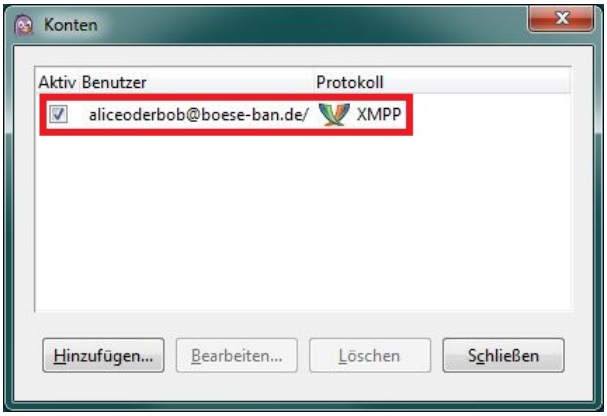


Abbildung 4.6: Kontenverwaltungsfenster

Wie im Hauptfenster zunächst beschrieben muss das Konto durch aktivieren der Checkbox aktiviert werden. Kann die Verbindung hergestellt werden, ist man nun online und das Hauptfenster stellt die Kontaktliste dar.

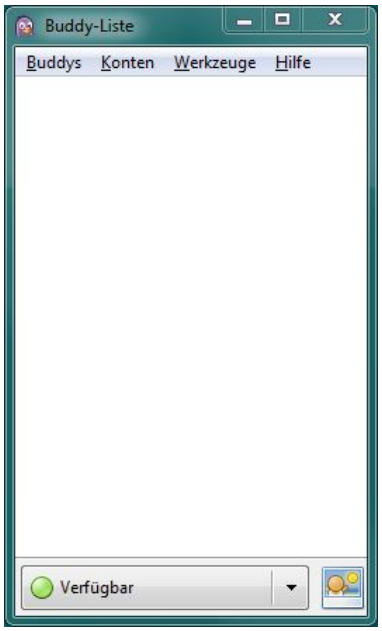


Abbildung 4.7: Kontaktliste ohne Kontakt

4.1.3 Kontakte hinzufügen

Um nun Kontakte zu seiner Kontakliste hinzuzufügen, öffnet man zunächst das Dropdownmenü **Buddys** und wählt dann **Buddy hinzufügen...** In die dafür vorgesehene Zeile gibt man nun die XMPP-ID des Kontaktes ein, den man hinzufügen möchte und wählt anschließend die Schaltfläche **Hinzufügen**. Eine XMPP-ID sieht aus wie eine E-Mail Adresse und setzt sich somit aus dem Benutzernamen und dem Server zusammen (*benutzername@server*).

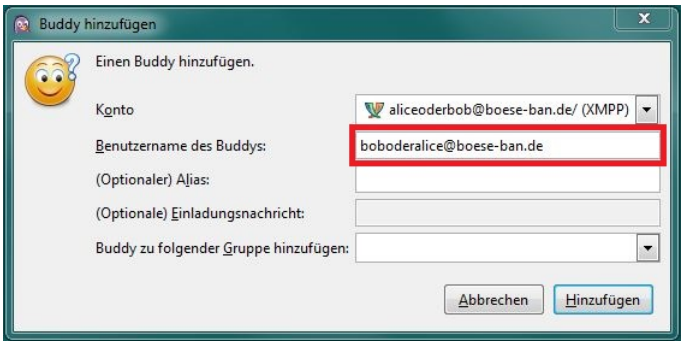


Abbildung 4.8: „Buddy hinzufügen“-Fenster

Der Benutzer, der hinzugefügt werden soll bekommt jetzt eine Anfrage, die er annehmen oder ablehnen kann. Nimmt er sie an, wird er wahrscheinlich eine Anfrage zurückschicken. Man hat die Möglichkeit einem Benutzer Nachrichten zu schreiben, selbst wenn er sich momentan noch nicht in der eigenen Kontaktliste befindet. Autorisiert man einen Kontakt, bestätigt also seine Anfrage, wird man seiner Kontaktliste hinzugefügt.

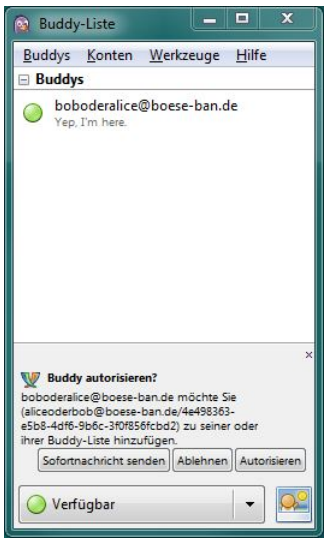


Abbildung 4.9: Authorisierungsanfrage eines Kontaktes

4.2 OTR Plugin

Um OTR zu nutzen muss Pidgin mit Hilfe eines Plugins erweitert werden.

4.2.1 Installation

Das Installations-Setup für das Plugin kann direkt von der Webseite der OTR-Entwickler `https://otr.cypherpunks.ca/` heruntergeladen werden.

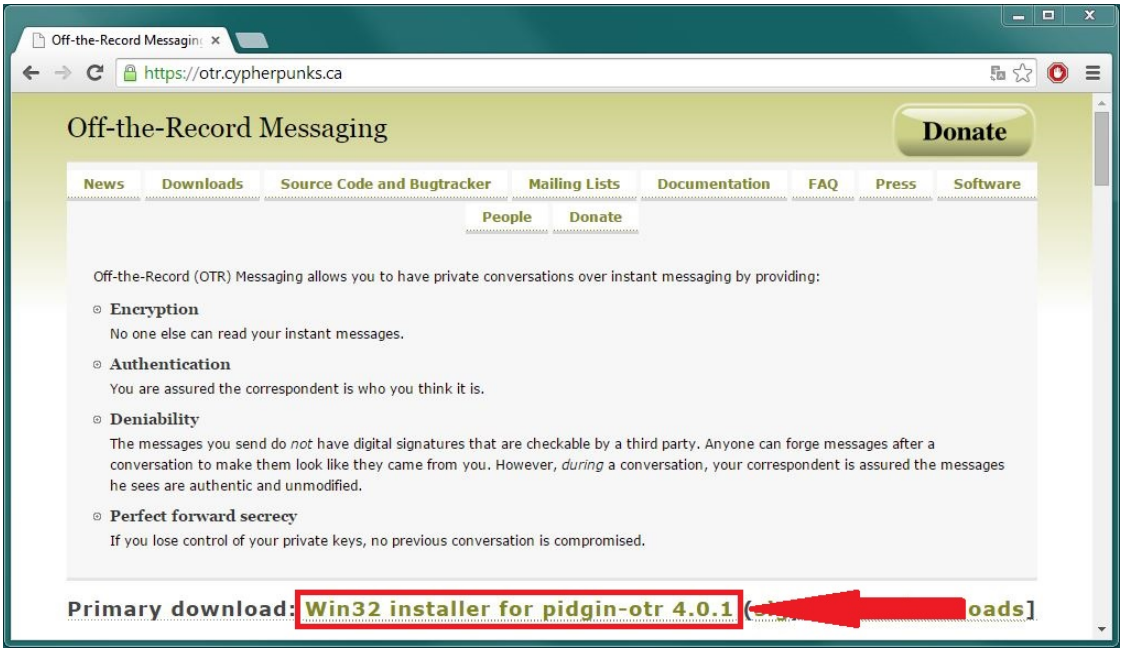


Abbildung 4.10: Webseite der OTR-Entwickler

Das Installations-Setup wird ausgeführt und das Plugin einfach installiert.

4.2.2 Konfiguration

Zunächst muss das Plugin aktiviert werden. Dazu wird in Pidgin das Dropdownmenü **Werkzeuge** geöffnet und **Plugins** ausgewählt. In dem nun geöffneten Fenster muss das Off-the-Record Messaging Plugin aktiviert werden, indem die entsprechende Checkbox aktiviert wird. Anschließend wird die Schaltfläche **Plugin konfigurieren** gewählt.

Jetzt muss einmalig durch auswählen der Schaltfläche **Generieren** ein Fingerprint erzeugt werden. Dieser wird später zur Verifikation benutzt.

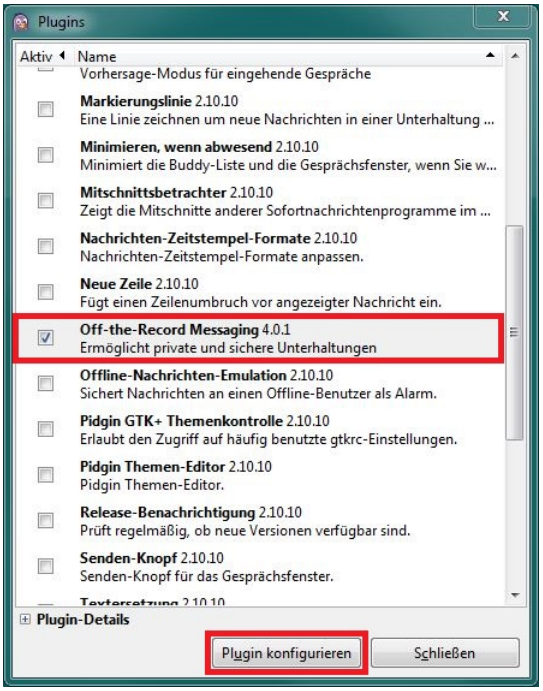


Abbildung 4.11: „Plugins“-Fenster

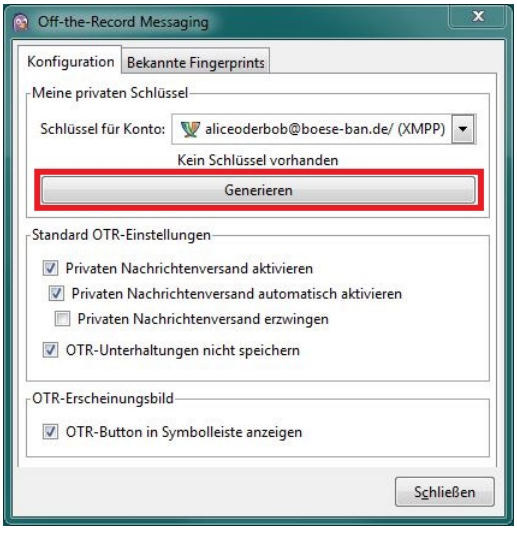


Abbildung 4.12: Fingerprint generieren

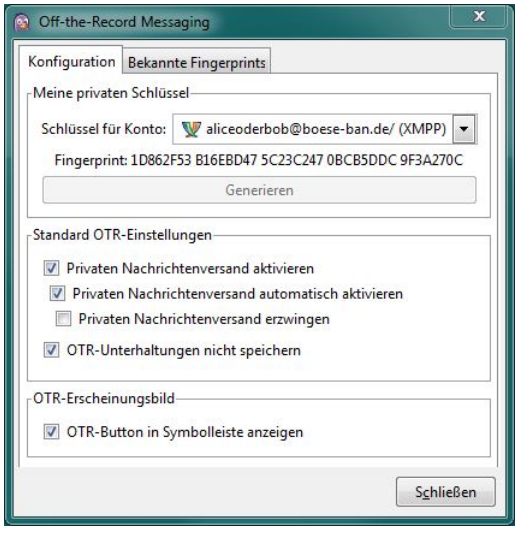


Abbildung 4.13: Fingerprint wurde generiert

4.2.3 Starten einer sicheren Konversation

Um den Chat mit einem Kontakt zu eröffnen muss er in der Kontaktliste doppelgeklickt werden. Zunächst ist die Konversation **Nicht privat**.

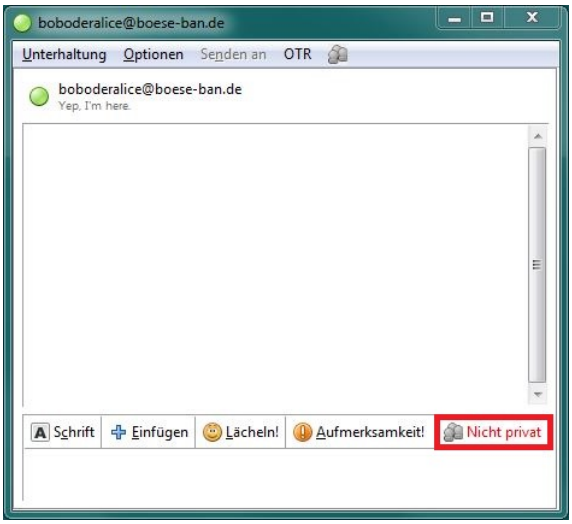


Abbildung 4.14: Unsichere Konversation

Wählt man nun die Schaltfläche **Nicht privat** öffnet sich ein Dropdownmenü in dem **Private Unterhaltung starten** gewählt wird. Das Gespräch findet zwar jetzt verschlüsselt statt, man hat sich aber noch nicht gegenseitig authentifiziert. Das bedeutet, dass man sich nicht sicher sein kann, dass der Kontakt auch wirklich der ist, für den man ihn hält.

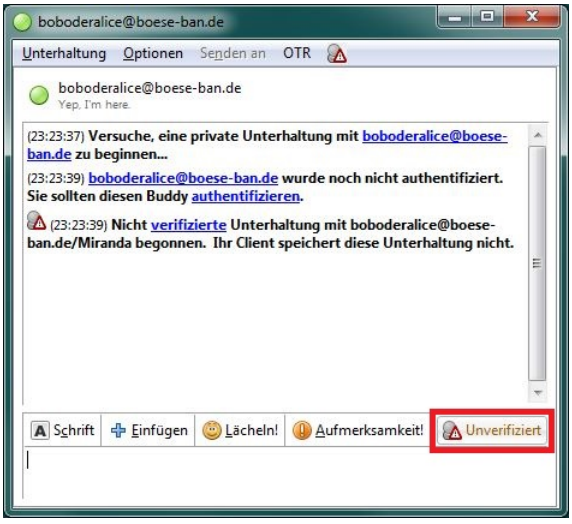


Abbildung 4.15: Unverifizierte Konversation

Um den Kontakt schließlich zu verifizieren, wird die Schaltfläche **Unverifiziert** gewählt. Es öffnet sich ein wieder ein Dropdownmenü in dem **Buddy authentifizieren** gewählt wird. In dem sich jetzt geöffneten Fenster wird die Methode gewählt, mit der man den Kontakt verifizieren möchte. Man hat dabei drei Möglichkeiten.

Frage und Antwort Es wird eine Frage und die dazugehörige Antwort festgelegt. Der Kontakt muss die Frage korrekt beantworten um sich zu verifizieren. Diese Authentifizierungsmethode ist einseitig, d.h. die Authentifizierung muss in beide Richtungen durchgeführt werden.

Abbildung 4.16: Authentifizierung mittels Frage und Antwort

Gemeinsam bekannte Passphrase Es wird ein Passwort festgelegt, dass der Kontakt korrekt eingeben muss um sich zu verifizieren. Diese Authentifizierungsmethode ist beidseitig, d.h. beide Kontakte können sich damit gleichzeitig und aufeinmal verifizieren.

Abbildung 4.17: Authentifizierung mittels Passwort

Manueller Fingerprint-Vergleich Die beiden öffentlichen und einmaligen Fingerprints der Kontakte werden auf ihre Richtigkeit überprüft. Sind die Fingerprints korrekt, findet eine beidseitige Verifizierung statt.

Abbildung 4.18: Authentifizierung mittels Überprüfung der Fingerprints

Der Kontakt ist jetzt verifiziert und die Unterhaltung gilt jetzt als sicher bzw. privat.

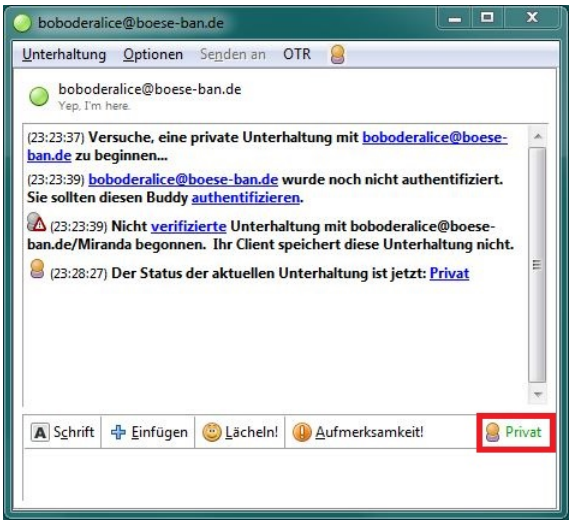


Abbildung 4.19: Sichere Konversation

5 Schluss

Das Off-the-Record Messaging Protokoll bietet eine Verschlüsselung, welche die Kriterien für eine sichere Kommunikation erfüllt. Wie man im Laufe des praktischen Teils jedoch festellen kann, bringt das Einrichten von Client und Plugin einen gewissen Aufwand mit sich. Und da in der Gesellschaft das Bewusststein für Sicherheit im Internet schwach ausgeprägt ist, sind nur wenige bereit diesen Aufwand auf sich zu nehmen.
Abgesehen davon muss man aber auch sagen, dass OTR für viele ungeeignet ist, da es z.B. Schwächen aufweist, was den Einsatz auf mobilen Geräten betrifft. Aber genau in diese Richtung bewegt sich der Trend eindeutig, weshalb es wichtig ist sozusagen den Sprung in den mobilen Bereich souverän zu schaffen und auch die Handhabung zu erleichtern. Dann würden sich womöglich mehr Nutzer auf einen Wechsel von beispielsweise WhatsApp einlassen und eine sicherere Alternative nutzen. Ein erster Schritt wäre aber auch schon bereits vorhandene Systeme wie z.B. den Facebook-Chat um die Verschlüsselung zu erweitern. Die Nutzer müssten sich kaum umstellen und ein Mehrgewinn an Sicherheit wäre schonmal gegeben.
Wie man sieht ist hier noch viel Raum für Entwicklungen und Fortschritt gegeben und in den nächsten Jahren wird auf der technischen Seite auch mit Sicherheit noch viel passieren. Es bleibt nur abzuwarten, ob auch die Gesellschaft noch das nötige Bewusststein entwickeln wird.

Abbildungsverzeichnis

2.1 Client-Server-Modell [2, S. 4] 2
4.1 Webseite der Pidgin-Entwickler 6
4.2 Startdialog 7
4.3 Fenster zum konfigurieren eines neuen Kontos 8
4.4 Externer Registrierungsdialog 8
4.5 Hauptfenster 9
4.6 Kontenverwaltungsfenster 9
4.7 Kontaktliste ohne Kontakt 9
4.8 „Buddy hinzufügen“-Fenster 10
4.9 Authorisierungsanfrage eines Kontaktes 10
4.10 Webseite der OTR-Entwickler 11
4.11 „Plugins“-Fenster 12
4.12 Fingerprint generieren 12
4.13 Fingerprint wurde generiert 12
4.14 Unsichere Konversation 13
4.15 Unverifizierte Konversation 13
4.16 Authentifizierung mittels Frage und Antwort 14
4.17 Authentifizierung mittels Passwort 14
4.18 Authentifizierung mittels Überprüfung der Fingerprints 15
4.19 Sichere Konversation 15

Literatur

[1] `https://de.wikipedia.org/wiki/Client-Server-Modell`

[2] `http://www.hki.uni-koeln.de/sites/all/files/courses/7568/Client-Server-Modell-fin.pdf`

[3] `http://xmpp.org/about-xmpp/technology-overview/`

[4] `http://public.tfh-berlin.de/~s30935/off-the-record-messaging.pdf`

[5] `https://otr.cypherpunks.ca/`

[6] `http://de.wikipedia.org/wiki/Off-the-Record_Messaging`

[7] `http://www.heise.de/security/artikel/Diffie-Hellman-Verfahren-270980.html`

[8] Johannes Buchmann: *Einführung in die Kryptographie*, Heidelberg Dordrecht London New York, ^{5}2010

[9] `http://www.itwissen.info/definition/lexikon/digital-signature-algorithm-DSA-DSA-Algorithmus.html`

[10] http://www.ningelgen.eu/03_Kryptologie/KryptDateien/Kapitel%2008_AES.pdf

[11] http://www.e-teaching.org/glossar/hashwert

[12] https://pidgin.im/about/

[13] https://otr.cypherpunks.ca/help/4.0.1/levels.php?lang=de

Alle Internetquellen wurden erfolgreich am 03.11.2014 aufgerufen.

Erklärung:

"Ich erkläre hiermit, dass ich die Seminararbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen (einschließlich Seiten aus dem Internet) und Hilfsmittel benützt habe."

Bergkirchen ,den 03.11.14

Ort Datum

Unterschrift des Schülers / der Schülerin