Sichere Unternehmens-kommunikation mit Jabber (XMPP)

SLAC 2008 Magdeburg
12. Dezember 2008

Holger.Zuleger@hznet.de

Agenda

- Einführung
  - Kommunikationsarten im Unternehmen
  - Warum Jabber/XMPP
- XMPP Kommunikationsmodell
  - Client Registrierung
  - Server − Server Kommunikation
  - Skalierung
- Softwareauswahl
  - Serversoftware / Clientsoftware
- Sicherheit
  - Client − Server Verschlüsselung (TLS)
  - Server − Server Dialback
  - Ende zu Ende Verschlüsselung
  - OTR
- Zusammenfassung

Kommunikationsarten im Unternehmen

- Interne Kommunikation
  E-Mail, Telefon, Meeting, Fax, Brief
- Externe Kommunikation
  Brief, Fax, E-Mail, Telefon, Meeting
- Eigenschaften unterschiedlicher Kommunikationsarten

	Medium	**Komm.-form**	**Vorlauf-zeit**	**Reakt.-zeit**	**Multiuser-fähig**
Mail	Text	async	–	≤ 5 Tage	ja
Telefon	Sprache	sync	unbest.	–	jein
pers. Meeting	Bild/ Sprache	sync	Tage - Wochen	–	ja
Brief/Fax	Text	async	–	≥ 1 Woche	nein
Video-konferenz	Bild/ Sprache	sync	Tage	–	ja
SMS	Text	async	–	Stunden	nein
Instant Messaging	Text	sync	Presence Dienst	Minuten	ja

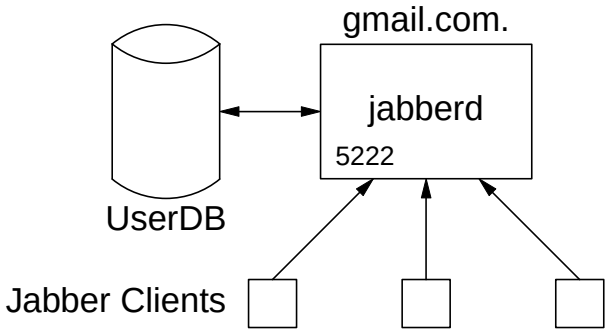
Warum Jabber/XMPP?

- Viele verschiedene Instant Messaging Systeme verfügbar
  IRC, ICQ, AIM, Yahoo, MSN, Skype, Gadu Gadu, QQ, GoogleTalk
- Meist properitäre Systeme
- Separate „Communities“
- Keine Kommunikation zwischen verschiedenen „Communities“ möglich
- Externe Server
- Teilweise sehr bedenkliche Datenschutzbestimmungen
- Jabber / XMPP
  - Offenes Protokoll (Extensible Messaging and Presence Protocol)
  - Dezentrale Server
  - Dedizierter Server für Unternehmen betreibbar
  - Gateway zu anderen IM Systemen

- Einführung
  - Kommunikationsarten im Unternehmen
  - Warum Jabber/XMPP
- XMPP Kommunikationsmodell
  - Client Registrierung
  - Server − Server Kommunikation
  - Skalierung
- Softwareauswahl
  - Serversoftware / Clientsoftware
- Sicherheit
  - Client − Server Verschlüsselung (TLS)
  - Server − Server Dialback
  - Ende zu Ende Verschlüsselung
  - OTR
- Zusammenfassung

XMPP Kommunikationsmodell

Client Registrierung

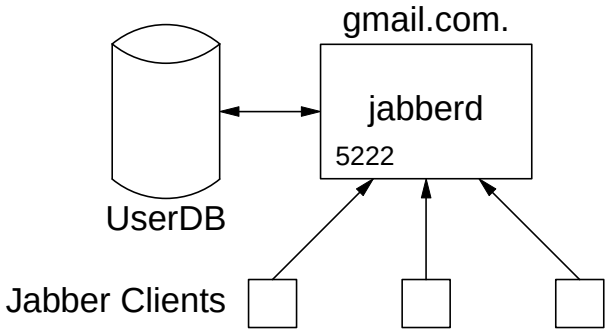


- JabberID ähnelt einer Mailadresse: `uid@do.ma.in./resource`
- Wie finden Clients den Registrierungsserver ?
  - DNS A lookup auf `do.ma.in`
  - DNS SRV lookup auf `_xmpp-client._tcp.do.ma.in.`
- Beispiel:

```
$  dig +noall +answer  SRV _xmpp-client._tcp.gmail.com
_xmpp-client._tcp.gmail.com. 25733 IN   SRV     20 0 5222 talk4.l.google.com.
_xmpp-client._tcp.gmail.com. 25733 IN   SRV      5 0 5222 talk.l.google.com.
_xmpp-client._tcp.gmail.com. 25733 IN   SRV     20 0 5222 talk1.l.google.com.
_xmpp-client._tcp.gmail.com. 25733 IN   SRV     20 0 5222 talk2.l.google.com.
_xmpp-client._tcp.gmail.com. 25733 IN   SRV     20 0 5222 talk3.l.google.com.
```

XMPP Kommunikationsmodell

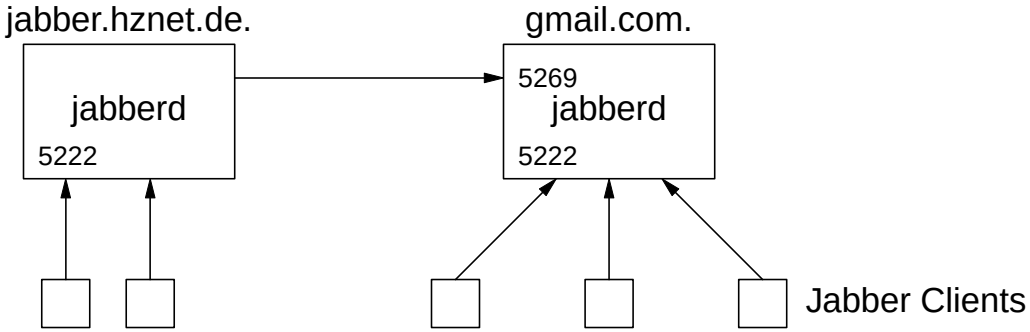
Client Registrierung (2)



- Client baut TCP Verbindung zum Server auf
- Verschlüsselung mittels TLS möglich
  - Serverauthentisierung über Zertifikat
  - Clientauthentisierung über z.B. Username & Password
- Verwaltung der Presence Informationen auf dem Server
- Clients einer Domain können miteinander komunizieren

XMPP Kommunikationsmodell

Server − Server

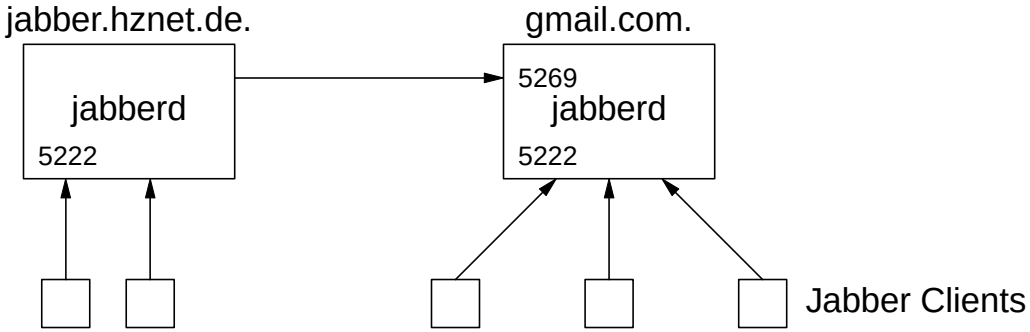


- Bei Jabber ist eine Kommunikation zu fremden Servern möglich
- Wie finden Server fremde Jabber Server ?
  - DNS SRV lookup `_jabber._tcp.foreign.domain.`
  - DNS SRV lookup `_xmpp-server._tcp.foreign.domain.`
- Beispiel:

```
$  dig +noall +answer  SRV _xmpp-server._tcp.gmail.com
_xmpp-server._tcp.gmail.com. 43200 IN   SRV     5 0 5269 xmpp-server.l.google.com.
_xmpp-server._tcp.gmail.com. 43200 IN   SRV     20 0 5269 xmpp-server1.l.google.com.
_xmpp-server._tcp.gmail.com. 43200 IN   SRV     20 0 5269 xmpp-server2.l.google.com.
_xmpp-server._tcp.gmail.com. 43200 IN   SRV     20 0 5269 xmpp-server3.l.google.com.
_xmpp-server._tcp.gmail.com. 43200 IN   SRV     20 0 5269 xmpp-server4.l.google.com.
```

XMPP Kommunikationsmodell

Server − Server (2)



- Domainübergreifende Komunikation möglich
- Server zu Server Kommunikation über separaten Port
  Firewall friendly
- Server zu Server Kommunikation über TLS
  - Authentisierung über Zertifikate
  - In der Praxis nur bei bekannten Domains möglich
  - Skalierungsproblem
- Meistens keine Verschlüsselung bei fremden Jabber Servern

XMPP Kommunikationsmodell

DNS−Konfiguration

- DNS RR für die Domain `jabber.hznet.de`

```
$ORIGIN jabber.hznet.de.
_xmpp-client._tcp    IN SRV  10  0   5222    jabber.hznet.de.
_xmpp-server._tcp    IN SRV  10  0   5269    jabber.hznet.de.
_jabber._tcp         IN SRV  10  0   5269    jabber.hznet.de.
```

- DNS RR für die Domain `gmail.com`

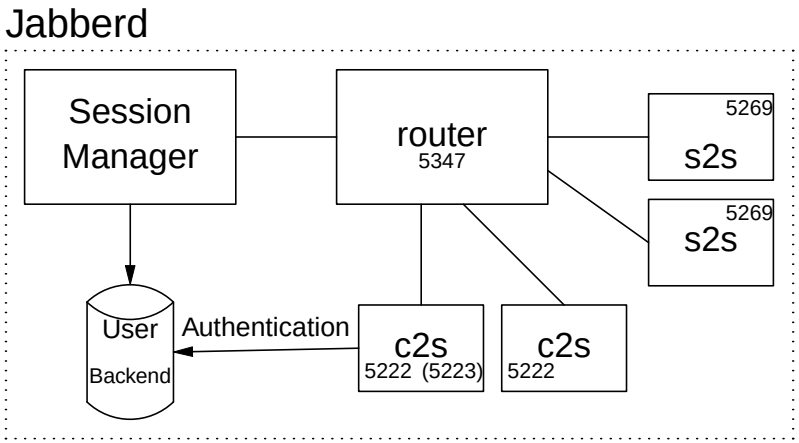
```
$ORIGIN   gmail.com.
_xmpp-client._tcp   IN SRV     20 0 5222 talk4.l.google.com.
_xmpp-client._tcp   IN SRV     5 0 5222 talk.l.google.com.
_xmpp-client._tcp   IN SRV     20 0 5222 talk1.l.google.com.
_xmpp-client._tcp   IN SRV     20 0 5222 talk2.l.google.com.
_xmpp-client._tcp   IN SRV     20 0 5222 talk3.l.google.com.

_xmpp-server._tcp   IN SRV     5 0 5269 xmpp-server.l.google.com.
_xmpp-server._tcp   IN SRV     20 0 5269 xmpp-server1.l.google.com.
_xmpp-server._tcp   IN SRV     20 0 5269 xmpp-server2.l.google.com.
_xmpp-server._tcp   IN SRV     20 0 5269 xmpp-server3.l.google.com.
_xmpp-server._tcp   IN SRV     20 0 5269 xmpp-server4.l.google.com.

_jabber._tcp        IN SRV     20 0 5269 xmpp-server2.l.google.com.
_jabber._tcp        IN SRV     20 0 5269 xmpp-server3.l.google.com.
_jabber._tcp        IN SRV     20 0 5269 xmpp-server4.l.google.com.
_jabber._tcp        IN SRV     5 0 5269 xmpp-server.l.google.com.
_jabber._tcp        IN SRV     20 0 5269 xmpp-server1.l.google.com.
```

XMPP Kommunikationsmodell

Skalierung



- Verschiedene SRV Records und Server Ports gut für Skalierung
- Mehrere Client Server (Prozesse) `c2s`
  Authentication & Registration
- Mehrere Server − Server (Prozesse) `s2s`
- Session Manager `sm`
  Verwaltet alle grundlegenden IM Features
- Routing Instanz für Inter-Komponenten Kommunikation `router`

- Einführung
  - Kommunikationsarten im Unternehmen
  - Warum Jabber/XMPP
- XMPP Kommunikationsmodell
  - Client Registrierung
  - Server − Server Kommunikation
  - Skalierung
- Softwareauswahl
  - Serversoftware / Clientsoftware
- Sicherheit
  - Client − Server Verschlüsselung (TLS)
  - Server − Server Dialback
  - Ende zu Ende Verschlüsselung
  - OTR
- Zusammenfassung

Server Software

Auswahl verfügbarer Jabber-Server
(http://de.wikipedia.org/wiki/Jabber; http://xmpp.org/software/servers.shtml)

- Freie Software
  - djabberd (perl) + libxml (C)
  - ejabberd (Erlang)
  - jabberd 1.4 (C)
  - jabberd 2.x (C)
  - OpenIM (Java)
  - psyced (C)
  - xmpdd.py (Python)
- Freie / Kommerzielle Software
  - Chime (Java)
  - OpenFire (Java)

Client Software

Auswahl der verfügbaren Jabber Clients

- Jabber only Clients
  (http://de.wikipedia.org/wiki/Liste_von_Jabber-Clients;
  http://xmpp.org/software/clients.shtml)
  - Coccinella (mit Whiteboard Funktionalität) (Tcl)
  - Psi (Windows, MacOS, Unix/Linux/BSD)
- Multi-Protokoll Clients
  (http://de.wikipedia.org/wiki/Multi-Protokoll-Client)
  - Adium (Mac OS)
  - Kopete (MacOS, Unix/Linux/BSD)
  - Pidgin (Win, MacOS, Unix/Linux/BSD)
  - Trillian (Win)

Software Auswahl

Server

- Betriebssystem: **Unix**
- Programmiersprache: **C**, C++, Java, Python, Erlang, ...
- Protokoll: IPv4, **IPv6**
- Sicherheit: **TLS**, Kerberos
- Backend: SQL, **BerkleyDB**, **File**, LDAP

☞ Jabberd2 (C)

Client

- Plattformunabhängig: **Unix**, **Windows**, **Apple**
- Protokoll: IPv4, **IPv6**
- Sicherheit: **TLS**, Kerberos, **OTR**, PGP/GPG
- Multiprotokollfähig: ICQ, Yahoo, AIM, MSN, ...

☞ Pidgin, Adium

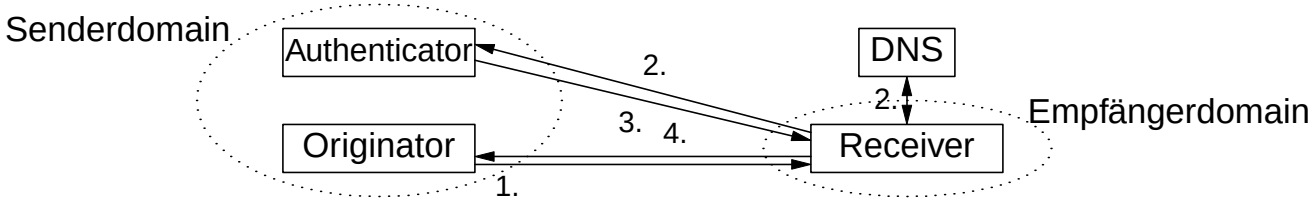
- Einführung
  - Kommunikationsarten im Unternehmen
  - Warum Jabber/XMPP
- XMPP Kommunikationsmodell
  - Client Registrierung
  - Server − Server Kommunikation
  - Skalierung
- Softwareauswahl
  - Serversoftware / Clientsoftware
- Sicherheit
  - Client − Server Verschlüsselung (TLS)
  - Server − Server Dialback
  - Ende zu Ende Verschlüsselung
  - OTR
- Zusammenfassung

Transport Layer Security

- Wird für Client to Server Kommunikation empfohlen
  Aus meiner Sicht ein Muß
- Zwei Verfahren
  - Über separaten Port (5223)
    deprecated
  - Über regulären Port, durch STARTTLS iniitiert
    recommended
- TLS als verpflichtend konfigurieren!
  - Nur 5223 als Listening Port konfigurieren...
  - ... und/oder `<require-starttls/>` einschalten!
- Server Authentisierung wie üblich über Zertifikat
- Clientauthentisierung in der Regel über Username/Passwort
  SASL Mechanismen (müssen von Client **und** Server unterstützt werden)

Server − Server (Dialback)

- Öffentlich erreichbare Jabber Server sind angreifbar
- Wie lässt sich die Identität des sendenden Server verifizieren ?
- Server Identifizierung über Dialback Verfahren (XEP-0220)



1. Verbindungsaufbau nach DNS Lookup durch den Sender
   Sender generiert und überträgt Key (Sender+Empfänger+StreamID+Secret)
2. Empfänger akzeptiert Session erst nach „Dialback“
   Verbindungsaufbau nach DNS Lookup zum (Ab)Sender und Key Übermittlung
3. Senderdomain antwortet ob Key korrekt ist oder nicht
4. Empfänger gibt Rückmeldung an Sender

- Verfahren ermöglicht schwachen Nachweis der Senderdomain-Identität
  Mit DNSSEC starker Nachweis

Server − Server Sicherheit (TLS)

- Dialback Verfahren schützt nicht die Datenübertragung
- TLS (Verschlüsselung) auch für Server zu Server Kommunikation
  - Kein dedizierter Port verfügbar
  - Negotiation erfolgt über STARTTLS
- Authentisierung über Zertifikate (schwierig bei unbekannten Domains)
  - Keine gemeinsame Root CA
  - Viele Root CAs
- Lösungsansätze
  - Gemeinsame Root CA verwenden (xmpp.org)
    Funktioniert nur bei „überschaubarer“ Serveranzahl
  - (Self)signed Cert + Dialback als „weak“ Authentisierung
    Skaliert, aber geringe Sicherheit weil DNS unsicher
  - Cert + Dialback + DNSSEC als sichere Authentisierung ?
    Noch nicht implementiert

Server − Server Sicherheit (Zusammenfassung)

Vier Arten der Server − Server Kommunikation

1. Nachgiebig (Nicht Empfohlen)
   - Server akzeptiert Verbindungen von überall ohne Identifizierung
2. Überprüfbar (Heute üblich)
   - Server wendet Dialback Verfahren zur Identifizierung an
   - Identifizierung basiert auf DNS
   - Secure DNS würde das Verfahren sicherer machen
3. Verschlüsselt (Empfohlen)
   - TLS mit evtl. selbstsigniertem Zertifikat + Dialback
4. Vertraulich (Das möchte man)
   - TLS mit signiertem Zertifikat
   - Kommunikationspartner Vertrauen der CA

Ende zu Ende Verschlüsselung

Ist TLS ausreichend ?

- Aus Unternehmenssicht: Vermutlich „Ja“
  - TLS bietet *Transport* Layer Security
  - Dritte (im Sinne von Externe) haben keinen Zugriff
  - Kontrolle der S2S Kommunikationspartner durch Routingkonfig & Zertifikate
- Aus Nutzersicht: Definitiv „Nein“
  - TLS schützt Transport
  - Nachrichten sind unverschlüsselt auf jedem Server-System
  - Nachrichten sind kopier- und/oder speicherbar

Was ist mit PGP/GPG ?

- Bietet Ende zu Ende Verschlüsselung
- Aber auch Nachweisbarkeit der Kommunikation

Off-the-record Kommunikation (OTR)

Off the record Kommunikation bietet die folgenden Eigenschaften

- **Verschlüsselung** (Encryption)

  Nur die beiden Kommunikationspartner können die Nachrichten entschlüsseln

- **Beglaubigung** (Authentication)

  Der Kommunikationspartner ist sicher identifizierbar

- **Abstreitbarkeit** (Repudiation)

  Niemand (auch nicht der Gesprächspartner) kann den Inhalt der Kommunikation sicher nachweisen

- **Folgenlosigkeit** (Perfect forward secrecy)

  Die ersten drei Eigenschaften gelten auch im Nachhinein

  Selbst wenn einer von beiden seinen Schlüssel verlieren sollte

Privatsphäre in der täglichen Kommunikation

Beispiel: Kaffeeküche

- Authentisierung / Identifizierung
  - Visuell
- Verschlüsselung
  - Wenn die Tür der Kaffeeküche geschlossen ist
  - Abhörmassnahmen sind gesetzlich verboten
  - Aussnahme: Strafverfolgung
- Abstreitbarkeit
  - Keine Zeugen (Aussage gegen Aussage)
  - Niemand darf ein Gespräch zwischen zwei Menschen ohne Einwilligung aufzeichnen
- Folgenlosigkeit
  - Ergibt sich aus den obigen Maßnahmen

Privatsphäre in der täglichen Kommunikation (2)

Beispiel: Telefonie

- Authentisierung / Identifizierung
  - Über Rufnummer (Wir trauen der Telefonanlage / Carrier)
  - Über die Stimme des Gesprächspartners
- Verschlüsselung
  - Abhörmassnahmen sind gesetzlich verboten
  - Schwierig zu prüfen
  - Kommunikationspartner darf nicht ohne Einwilligung auf „mithören“ stellen
- Abstreitbarkeit
  - Keine Zeugen (Aussage gegen Aussage)
- Folgenlosigkeit
  - Ergibt sich aus den obigen Maßnahmen

Aufrechterhaltung der Privatsphäre wird schwieriger

Privatsphäre in der täglichen Kommunikation (3)

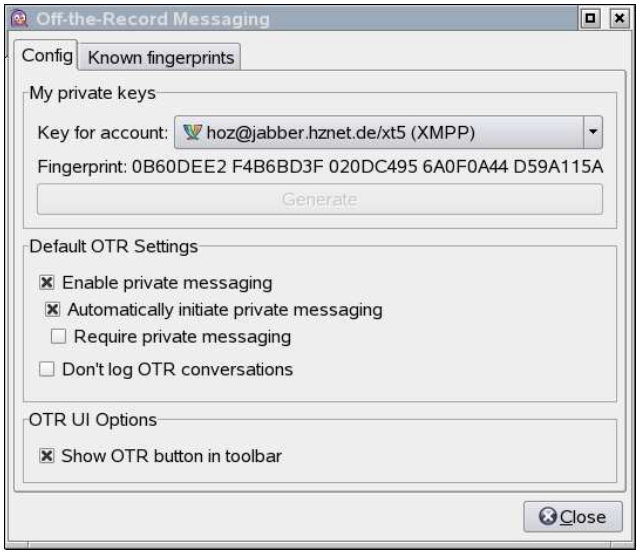
Beispiel: E-Mail

- Authentisierung / Identifizierung
  - Über Email-Adresse (einem einfachen Text Headerfeld)
  - Von jedem beliebig änderbar
  - Nicht vorhanden
- Verschlüsselung
  - Gesetzliche Regelungen das niemand eine E-Mail lesen darf
  - Nicht zu gewährleisten
- Abstreitbarkeit
  - Gewährleistet, da eine E-Mail von jedem ohne Aufwand gefälscht werden kann
- Folgenlosigkeit
  - Sollte sich aus der Abstreitbarkeit ergeben

Aufrechterhaltung der Privatsphäre ist nur durch Verschlüsselung möglich

OTR (Authentisierung)

- Beide Kommunikationspartner erzeugen asymetrischen Schlüssel
  Ausschliesslich zum Signieren **nicht** für die Verschlüsselung genutzt
- Schlüsselalgorithmus DSA ist ein Signaturalgorithmus



- Öffentlicher Schlüsselteil wird bei Verbindungsaufbau übertragen
  Ähnlich `ssh`
- Fingerprint wird nach erfolgreicher Authentisierung lokal gespeichert
  Auch dies ähnlich zu `ssh`

OTR (Authentisierung)

- Über persönliche Frage

OTR (Authentisierung)

- Über Shared Secret

OTR (Authentisierung)

- Über Fingerprint

OTR (Verschlüsselung)

- Austausch eines Schlüssels über Diffie-Hellman
- Authentisierter DH gegen Man-in-the-Middle Attacken
  Durch asymetrischen Key
- Schlüssel hängt nicht von Authentisierungsschlüssel ab
  Perfect Forward Secrecy
- Schlüssel wird nach jeder Nachricht gewechselt
  Sehr kurzlebige Schlüssel
- Werden nach der Nutzung sofort vernichtet
- Message Authentisierung über HMAC
  MAC key ist Hash des Encryption Keys
- Wechselt ebenfalls nach jeder Nachricht
- Wird nach Nutzung veröffentlicht
  Nützlich für Abstreitbarkeit

OTR (Protokoll)

- TLV basiertes Kommunikationsprotokoll
- Alle Binärdaten sind BASE64 encoded
  Encapsulierung der BASE64 Daten durch `?OTR:` und Punkt `.`
- Alle Nachrichten sind einfache Textnachrichten
  Keine Einbindung in spezielles IM Protokoll notwendig
- OTR kann mit allen IM Protokollen verwendet werden
  Auch Proxies stehen zur Verfügung
- OTR Anforderung durch Taggen der Nachricht mit spez. Leerzeichen
  □→□□→→→→ □→□→□→□□ □□→→□□→□
- Plugins für verschiedene Clients stehen zur Verfügung
  Trillian, Pidgin, Miranda, Psi
- OTR ist bei vielen Clients bereits fest integriert
  Adium, climm, MCabber, Kopete (ab 0.50.80)

Zusammenfassung

- Jabber bietet für Unternehmen ausreichende Sicherheit
  - Betrieb dedizierter Server
  - Zugriff für Clients und Remote Server separat steuerbar
  - TLS für verschlüsselte Client − Server Kommunikation
  - TLS für verschlüsselte Kommunikation zwischen „trusted“ Servern
  - Zugriff auf remote Jabber Domains bis auf JID Ebene kontrollierbar
- OTP für Ende zu Ende Sicherheit
  - Einfache Konfiguration und Bedienung
  - Sichere, verschlüsselte Kommunikation
  - Gewährleistung der Privatsphäre
  - „Kaffeküchengespräche“ möglich

Referenzen

XMPP http://xmpp.org
http://jabberd2.xiaoka.com/

Wikipedia
http://de.wikipedia.org/wiki/Jabber
http://de.wikipedia.org/wiki/Xmpp
http://de.wikipedia.org/wiki/Off-the-Record_Messaging

RFCs 3920 (Extensible Messaging and Presence Protocol (XMPP): Core)
3921 (XMPP: Instant Messaging and Presence)
3922 (Mapping XMPP to Common Presence and Instant Messaging)
3923 (End-to-End Signing and Object Encryption for XMPP)

OTR Webseite
http://www.cypherpunks.ca/otr/

H Z N E T

DNSsec, VoIPsec, IPsec, XMPPsec, SMTPsec, WLANsec ...

... DKIM, Kerberos, IMAP, LDAP, ENUM, SIP, ...

... NTP, DNS, DHCP, IPv6, Routing, Switching

CONTENTS