Netze

Modul 13: Messaging

Prof. Dr. Hannes Tschofenig

15. Dezember 2026

Semesterplanung — Vorlesungen

Modul	Dozent	Datum	Thema
1	Rademacher	2. Oktober 2025	Einführung, OSI-Referenzmodell und Topologien
2	Rademacher	9. Oktober 2025	Übertragungsmedien und Verkabelung
3	Rademacher	16. Oktober 2025	Ethernet und WLAN
4	Tschofenig	23. Oktober 2025	IPv4, Subnetze, ARP, ICMP
5	Tschofenig	30. Oktober 2025	IPv6 und Autokonfiguration
6	Tschofenig	6. November 2025	Netzwerksegmentierung
7	Tschofenig	13. November 2025	Routing
8	Rademacher	20. November 2025	Transportschicht und UDP
9	Rademacher	27. November 2025	TCP
10	Rademacher	4. Dezember 2025	DNS und HTTP 1
11	Tschofenig	11. Dezember 2025	HTTP 2 und QUIC
12	Tschofenig	18. Dezember 2025	TLS und VPN
/	/	8. Januar 2026	`Bei Bedarf / TBA`
13	Tschofenig	15. Januar 2026	Messaging
14	Rademacher	22. Januar 2026	Moderne Netzstrukturen

Semesterplanung — Übungen und Praktika

ID	KW	Art	Thema
	40	/	/
UE-1	41	Übung	Topologien und OSI
UE-2	42	Übung	Übertragungen bspw. Kabel
P-1	43	Praktikum	Laboreinführung und Netzwerktools
S-1	44	Video	IPv4
P-2	45	Praktikum	Adressierung
P-3	46	Praktikum	IPv4 und Autokonfiguration
P-4	47	Praktikum	IPv6 und Autokonfiguration
P-5	48	Praktikum	Routing
P-6	49	Praktikum	Switching
P-7	50	Praktikum	Transportprotokolle
S-2	51	Experiment	VPN
S-2	52	Experiment	VPN
	2	/	/
P-8	3	Praktikum	DNS
P-9	4	Praktikum	Webkommunikation

UE - Übung laut Stundenplan in den Seminarräumen

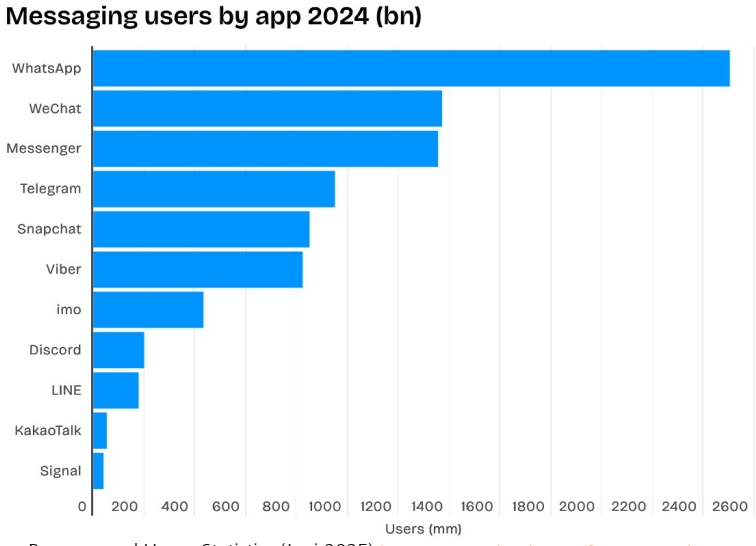
P - Praktikum in C055

S - Selbststudium KEINE Präsenz

Agenda

- Einleitung in Messaging
- Messaging-Apps: Architektur und Protokolle
- XMPP: Geschichte, Anwendungsbereiche und Sicherheitsanforderungen
- XMPP: Sicherheitslösungen
- Off-the-Record (OTR) Messaging: Konzept und Sicherheitsmerkmale
- Zusammenfassung

Messaging Apps sind beliebt!



Quelle: Messaging App Revenue and Usage Statistics (Juni 2025) https://www.businessofapps.com/data/messaging-app-market/

Die Sicherheit von Messaging-Apps ist ein ständig diskutiertes Thema!

Österreich: Ruf nach Ausweitung von Messenger-Überwachung

Noch ist staatliche Malware in Österreich vom Parlament nicht abgesegnet, da gibt es bereits Rufe nach Ausweitung. Koalitionspartner NEOS stellt sich dagegen.

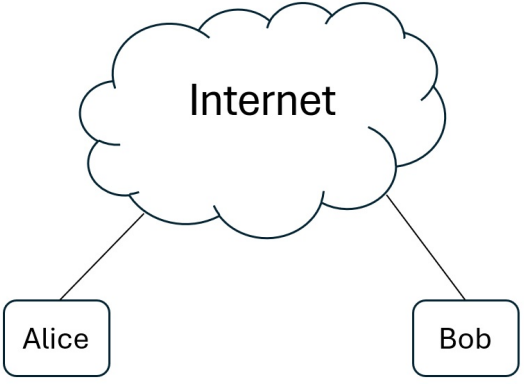
49

Quelle: Heise (27. Juni 2025)

https://www.heise.de/news/Oesterreich-Ruf-nach-Ausweitung-von-Messenger-Ueberwachung-10461718.html

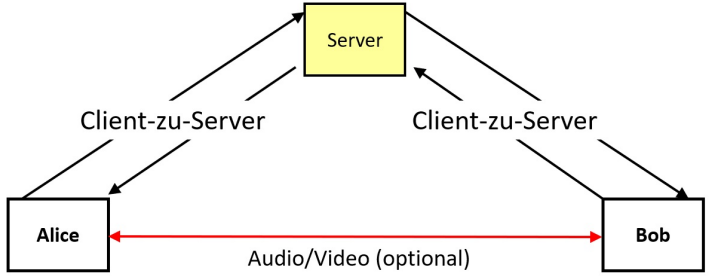
Wie funktionieren Messaging-Apps?

Architektur von Messaging-Apps



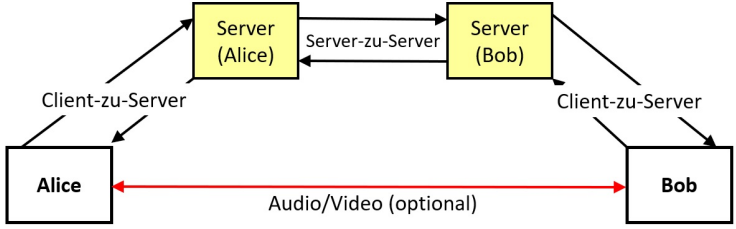
- Welche Herausforderungen gibt es bei diesem Ansatz?

Architektur von Messaging-Apps: Infrastrukturunterstützung



- Nutzer:innen registrieren sich über ihre Messaging-App beim Server, wobei die IP-Adresse übermittelt wird. Der Server vermerkt beispielsweise: ‘Alice ist online und unter der IP-Adresse X.X.X.X erreichbar.’
- Alice übermittelt Nachrichten an Bob über einen zentralen Server.
- Der Server speichert Nachrichten, bis sie zugestellt werden können. Bob könnte beispielsweise offline sein.
- Der Server kann Metadaten (Absender, Empfänger, Zeitstempel) speichern.

Erweiterte Messaging-Architektur: Föderiertes Modell



- Wird nicht von allen Messaging-Apps unterstützt.
- Ähnelt dem föderierten Ansatz von E-Mail und Voice-over-IP (VoIP).
- Erfordert Benutzeridentifikation, die es erlaubt die Domaine des Servers zu ermittlen.

Erweiterte Messaging-Architektur: Peer-to-Peer (P2P)

- Audio- und Video-Daten werden typischerweise direkt zwischen den Endgeräten ausgetauscht.
- Gründe: geringere Latenz und Reduktion der Serverlast.
- NATs und Firewalls erschweren jedoch die direkte Kommunikation erheblich.
- Erfordert NAT-Traversal-Techniken, typischerweise Session Traversal Utilities for NAT (STUN) [7], Traversal Using Relays around NAT (TURN) [8] und Interactive Connectivity Establishment (ICE) [3].

Messaging-Apps: Protokolle

- Eine Vielzahl von Messaging-Apps existiert und diese verwenden (leider) unterschiedliche Protokolle für den Nachrichtenaustausch.
- Beispiele sind XMPP, Matrix, Signal Protocol, SIP, und eine lange Liste von proprietären Protokollen (z.B. Telegram, WhatsApp).
- Manche dieser Protokolle unterstützen eine Ende-zu-Ende-Verschlüsselung.
- Die Kommunikationsendpunkte sind - anders als bei der Interaktion mit Webservern - die Endgeräte der Nutzer:innen, also Smartphones, Tablets oder Laptops.

Wir schauen uns XMPP genauer an!

XMPP: Geschichte

- XMPP, das **Extensible Messaging and Presence Protocol**, wurde ursprünglich außerhalb der IETF unter dem Namen Jabber entwickelt.
- Jabber war mehr als nur eine Sammlung von Spezifikationen - es umfasste sowohl Software (Server und Clients) als auch operative Dienste.
- Jabber Inc. wurde später von Cisco aufgekauft.
- Die Kernspezifikationen von XMPP wurden innerhalb der IETF entwickelt. Die XMPP-Erweiterungsprotokolle (XEPs) wurden jedoch von der XMPP Foundation veröffentlicht: `https://xmpp.org/extensions/`
- Die Inspiration für XMPP war das E-Mail-System, bei dem es jedem möglich ist, einen eigenen E-Mail-Server zu betreiben und mit anderen Nutzern über deren Server zu kommunizieren.

Peter Saint-Andre, Autor vieler XMPP RFCs.

XMPP: Anwendungsbereiche

- XMPP wurde ursprünglich für Instant Messaging entwickelt, fand später jedoch Anwendung in folgenden Bereichen:
- Präsenzinformationen (RFC 6121 [13])
- Sprach- und Videoanrufe (unter dem Namen Jingle - XEP-0166 [4], XEP-0167 [5] und XEP-0262 [11])
- Dateitransfer (XEP-0234 [15])
- Gruppenchat (XEP-0045 [10]), z.B. für Online-Spiele
- Publish-Subscribe (XEP-0060 [6]), z.B. für soziale Medienplattformen
- Internet of Things
- Notfallbenachrichtigungen (XEP-0127 [14])
- Sicherheitsereignisse (XEP-0060 [6])
- und viele mehr ...

XMPP: Highlights

- Die grundlegende Funktionalität ist in RFC 6120 [12] definiert.
- Der Client initiiert die Kommunikation, anschließend erfolgt der Datenaustausch asynchron.
- Ein Stream bezeichnet die zeitlich fortlaufende Interaktion (vergleichbar mit einem Dialog).
- Ein Stream muss zunächst initiiert werden (inklusive Aushandlung von Funktionen).
- Innerhalb eines Streams werden sogenannte „Stanzas“ (XML-Fragmente) ausgetauscht. → Beispiele: Nachrichten, Informationsabfragen (IQ) und Präsenzinformationen
- Es existieren zwei Streams: ein Eingabestream und ein Ausgabestream.
- Typischerweise läuft die Kommunikation über eine einzelne TCP-Verbindung an der Schnittstelle zwischen Client und Server.
- Ähnlich wie beim E-Mail-Verkehr müssen Kommunikationspartner nicht gleichzeitig online sein. → Der Server, der die Nachrichten zwischengespeichert hat, informiert Clients über neue Nachrichten, sobald sie sich wieder verbinden.

XMPP: Sicherheitsanforderungen (1/2)

- Vertraulichkeit, Integrität und Authentizität
  - Ende-zu-Ende- und Hop-by-Hop-Schutz
  - Authentifizierung zwischen Client und Server
  - Sicherheit von Nachrichten und Medien
- Zugangskontrolle
  - Rollenverwaltung in Gruppenchats
  - Sperren oder Ausschließen von Teilnehmenden
- Datenschutz
  - Minimierung der preisgegebenen Metadaten
  - Unterstützung für anonyme oder versteckte Gruppenmitgliedschaften

XMPP: Sicherheitsanforderungen (2/2)

- Zuverlässigkeit und Verfügbarkeit
  - Offline-Nachrichtenübermittlung
  - Synchronisierung über mehrere Geräte
  - Schutz vor Denial of Service Angriffen
- Plausible Deniability (Abstreitbarkeit) → Wird später besprochen
- Forward Secrecy → Ein Angreifer, der den privaten Langzeitschlüssel erlangt, kann keine in der Vergangenheit aufgezeichneten Nachrichten entschlüsseln.
- Post-Compromise Security → Wenn ein Angreifer vorübergehend Zugriff auf ein Gerät erlangt, verliert er im Laufe der Zeit den Zugriff auf zukünftige Nachrichten, da bei jedem Schritt ein neuer Schlüssel generiert wird.

XMPP: Lösung der Sicherheitsanforderungen

- XMPP verwendet TLS für die Verbindung zwischen Client und Server.
- Die Benutzerauthentifizierung erfolgt über das Simple Authentication and Security Layer (SASL) (RFC 4422 [18]).
- Nachrichten können mit dem Signal-Protokoll (XEP-0384 [16]) verschlüsselt werden - oder auch mit Off-the-Record (OTR) Messaging (XEP-0378 [17, 9]).
- Der Server kann auch Gruppeninformationen verwalten (XEP-0045 [10]).

```
<stream from='alice@example.com'
to='bob@example.com' version='1.0'
xml:lang='en'
 xmlns='http://etherx.jabber.org/streams'>
    <message xmlns='jabber:client'>
        <body>Das ist ein Test!</body>
    </message>
</stream>
```

Beispiel für eine XMPP-Nachricht, die von Alice an Bob gesendet wird.

Off-the-Record (OTR) Messaging

- Ein Sicherheitsprotokoll, das für sichere und private 1-zu-1-Kommunikation entwickelt wurde.
- Eigenschaften
  - Ende-zu-Ende-Verschlüsselung
  - Forward Secrecy - sogar jeder Nachrichtenaustausch verwendet einen neuen Schlüssel
  - Plausible Deniability (Abstreitbarkeit) - keine kryptografischen Beweise für die Urheberschaft einer Nachricht.
- Protokollannahme: Die Teilnehmer:innen kennen sich offline und können ein gemeinsames Geheimnis über einen externen Kanal oder durch gemeinsames Wissen etablieren.
- Hinweis: Es existieren verschiedene OTR-Varianten, die grundlegenden Konzepte bleiben jedoch gleich. Die Originalveröffentlichung [1] wurde in [2] überarbeitet und erweitert.

Off-the-Record (OTR): Forward Secrecy und Abstreitbarkeit

- Verwendet Diffie-Hellman-Schlüsselaustausch für **Forward Secrecy**
- Nachrichten werden **nicht digital signiert** (wie es, zum Beispiel, bei sicherer E-Mail-Kommunikation der Fall ist). Digitale Signaturen werden nur zur **Authentifizierung der ausgetauschten DH-Schlüssel** verwendet.
- Dies ermöglicht die **Abstreitbarkeit**.

Notation:

- g^x, g^y: DH-öffentliche Schlüssel,
- k_A, k_B: langfristige private Schlüssel,
- K_A, K_B: langfristige öffentliche Schlüssel

Nachrichtenaustausch:

- Alice $\rightarrow$ Bob: $\text{Sign}(g^x, k_A), K_A$
- Alice $\leftarrow$ Bob: $\text{Sign}(g^y, k_B), K_B$

Hinweis: Dieser Austausch authentifiziert Alice und Bob nicht, da die DH-Schlüssel ephemer sind.

Off-the-Record (OTR) und Message Authentication Codes (MACs)

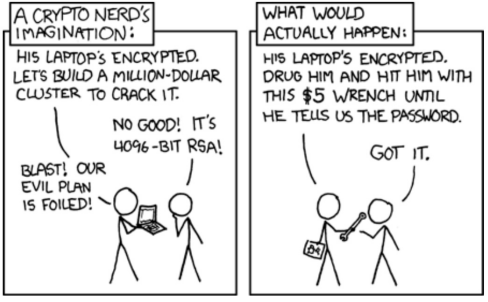
- A $\rightarrow$ B: MAC$(\{g^{x_2}, E(M_1, k_{11})\}, H(k_{11}))$ mit $k_{11} = H(g^{x_1 y_1})$
- MACs mit abgeleitete Schlüssel werden verwendet, um die Nachrichten gegen Manipulation zu schützen.
- Eine digitale Signatur wird nur beim initialen Schlüsselaustausch benötigt.
- Für alle nachfolgenden Schlüsselaktualisierungen werden MACs verwendet, um neue Schlüssel zu authentifizieren - basierend auf einem zuvor etablierten, vertrauenswürdigen geheimen Schlüssel.
- Da jede Person mit Zugang zum MAC-Schlüssel gültige Nachrichten erzeugen kann, ist es unmöglich, die Autor:in einer bestimmten Nachricht zweifelsfrei nachzuweisen.

Off-the-Record (OTR) und Malleable Encryption

- Es soll nicht nur für Bob und Eve unmöglich sein zu beweisen, dass Alice eine bestimmte Nachricht gesendet hat.
- Es soll sogar offensichtlich sein, dass jede beliebige Person die Nachricht verändert oder sogar gesendet haben könnte.
- → Verwendet wird **malleable encryption** mit AES im Counter-Modus (AES-CTR).
- AES im CTR-Modus erlaubt es einem Angreifer, den Geheimtext so zu verändern, dass dies zu vorhersehbaren und sinnvollen Änderungen im Klartext führt - selbst ohne Kenntnis des Schlüssels.
- Dieses Verschlüsselungsschema ist **formbar** (malleable): Ein Bitflip im Geheimtext bewirkt einen vorhersehbaren Bitflip im Klartext.
- Nach jeder Nachricht werden neue, flüchtige DH-Schlüssel erzeugt. Dadurch wird das Zeitfenster verkleinert, in dem alte Nachrichten entschlüsselt werden könnten. → **Post-Compromise Security**

Zusammenfassung

- XMPP ist ein Beispiel für ein Messaging-Protokoll.
- Es verdeutlicht architekturelle Gemeinsamkeiten mit VoIP und Email.
- Messaging-Apps haben viele interessante Sicherheitsanforderungen und daher existieren auch viele Lösungen.
- Off-the-Record (OTR) stellt die ursprünglich Entwicklung dar - es wurde später durch modernere Ansätze in Protokollen wie Signal und Wire ergänzt.
- Im Bereich der Sicherheit bestehen weiterhin Spannungen zwischen Regulierungsbehörden und Technikentwickler:innen.



Quelle: https://xkcd.com/538/

Quellen I

[1] Borisov, N., Goldberg, I., and Brewer, E.
Off-the-record communication, or, why not to use pgp.
In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2004), WPES '04, Association for Computing Machinery, p. 77–84.

[2] Di Raimondo, M., Gennaro, R., and Krawczyk, H.
Secure off-the-record messaging.
In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2005), WPES '05, Association for Computing Machinery, pp. 81–89.

[3] Keränen, A., Holmberg, C., and Rosenberg, J.
Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal.
RFC 8445, July 2018.

[4] Ludwig, S., Beda, J., Saint-Andre, P., McQueen, R., Egan, S., and Hildebrand, J.
Jingle.
https://xmpp.org/extensions/xep-0166.html.

[5] Ludwig, S., Saint-Andre, P., Egan, S., McQueen, R., and Cionoiu, D.
Jingle RTP Sessions.
https://xmpp.org/extensions/xep-0167.html.

[6] Millard, P., Saint-Andre, P., and Meijer, R.
Publish-Subscribe.
https://xmpp.org/extensions/xep-0060.html.

[7] Petit-Huguenin, M., Salgueiro, G., Rosenberg, J., Wing, D., Mahy, R., and Matthews, P.
Session Traversal Utilities for NAT (STUN).
RFC 8489, February 2020.

[8] Reddy.K, T., Johnston, A., Matthews, P., and Rosenberg, J.
Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN).
RFC 8656, February 2020.

Quellen II

[9] Saint-Andre, P.
Best Practices for Handling Offline Messages.
https://xmpp.org/extensions/xep-0160.html.

[10] Saint-Andre, P.
Multi-User Chat.
https://xmpp.org/extensions/xep-0045.html.

[11] Saint-Andre, P.
Use of ZRTP in Jingle RTP Sessions.
https://xmpp.org/extensions/xep-0262.html.

[12] Saint-Andre, P.
Extensible Messaging and Presence Protocol (XMPP): Core.
RFC 6120, March 2011.

[13] Saint-Andre, P.
Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence.
RFC 6121, March 2011.

[14] Saint-Andre, P., and Fletcher, B.
Common Alerting Protocol (CAP) Over XMPP.
https://xmpp.org/extensions/xep-0127.html.

[15] Saint-Andre, P., and Stout, L.
Jingle File Transfer.
https://xmpp.org/extensions/xep-0234.html.

[16] Straub, A., Gultsch, D., Henkes, T., Herberth, K., Schaub, P., and Wißfeld, M.
OMEMO Encryption.
https://xmpp.org/extensions/xep-0384.html.

Quellen III

[17] Whited, S.
OTR Discovery.
https://xmpp.org/extensions/xep-0378.html.

[18] Zeilenga, K., and Melnikov, A.
Simple Authentication and Security Layer (SASL).
RFC 4422, June 2006.