Mobile Instant Messenger und deren Sicherheitsstand

Christian Ochsenkühn
Hochschule Hof,
University of Applied Sciences
Hof, Germany
christian.ochsenkuehn@hof-university.de

ABSTRACT
Dieser Artikel beschreibt den aktuellen Sicherheitsstand mobiler Instant Messenger. Dazu wurde eine Auswahl an Messengern getroffen, welche anhand der sicherheitsrelevanten Kriterien Infrastruktur, Authentizität, Verschlüsselung und Transparenz untersucht wurden. Die Auswahl der mobilen Instant Messenger fiel dabei auf WhatsApp, Threema, myEnigma und Whistle.im. Diese Auswahl und die relevanten Kriterien basieren hierbei auf einem ausführlichen Grundlagen-Kapitel welches zudem einerseits einen Überblick über die Sicherheit von mobilen Instant Messengern geben soll und andererseits die dort vorherrschende Komplexität gängiger kryptographischer Verfahren deutlich machen soll. Ein abschließendes Fazit zeigt somit den aktuellen Stand der Sicherheit im Bereich jener Messenger und stellt einige zukünftige Erwartungen bereit.

Categories and Subject Descriptors
K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – *general, infrastructure, authenticity, encryption, transparency*

General Terms
Security, Verification.

Keywords
Mobil, Sicherheit, Instant Messaging, Kryptographie

1. EINFÜHRUNG
Fast ein Fünftel der deutschen Bevölkerung nutzt derzeit den mobilen Instant-Messenger WhatsApp [31]. Was im ersten Moment wie ein Werbeslogan klingen mag, ist Fakt und spiegelt die derzeitige Verbreitung von mobilen Messengern wieder. Das Verschicken von Nachrichten über mobile Endgeräte erfreut sich nämlich zunehmender Beliebtheit und brachte dem Versand von SMS in den letzten drei Jahren Prognosen zufolge einen Zuwachs um 3,2 Mrd. auf 19,5 Mrd. Nachrichten pro Tag im Jahr 2013. Im selben Zeitraum konnte sich der Versand von Instant-Messaging Nachrichten von 4,4 Mrd. auf stolze 41 Mrd. pro Tag fast verzehnfachen [33].

Neben diesem enormen Anstieg an Instant-Messaging-Nachrichten sind auch die Nutzerzahlen bekannter Messenger deutlich gestiegen. Der Messenger Whatsapp etwa hat im August 2013 die 300 Millionen Nutzer-Marke erreicht. Vier Monate zuvor waren es noch 100 Millionen Nutzer weniger [34]. Eigenen Angaben zu Folge seien im Dezember 2013 dann sogar 400 Millionen monatlich aktive Nutzer erreicht worden [21]. Auch andere Messenger, wie Skype (111% Nutzerzuwachs) und der Facebook Messenger (107%) konnten ihre Nutzerzahlen neben WhatsApp (123%) im 3. Quartal 2013 gegenüber dem 1. Quartal mehr als verdoppeln [32].

Mittlerweile legen Nutzer ihre Schwerpunkte bei der Auswahl eines Messengers aber nicht mehr nur auf die Verbreitung oder eine einfache Bedienung. Spätestens seit den NSA-Skandalen wächst der Bedarf an sicheren Messengern, was eine Vielzahl an mit Sicherheit werbender Messenger zum Vorschein bringt [17]. Auch die vielen Sicherheitslücken des Marktführers WhatsApp, weswegen der Messenger-Dienst immer wieder in der Kritik stand [11][9][18][19], dürfte zu dieser Entwicklung und den vielen Alternativen beigetragen haben. So ist es nicht verwunderlich, dass eine Vielzahl an neuen mobilen Messengern versucht, sich am Markt zu etablieren. Eine Suche nach Messengern in Googles App-Store für Android „Google Play“ bringt beispielsweise bereits hunderte an Vorschlägen zu Tage (Stand 01/2014).

Die folgende Untersuchung einer Auswahl aktueller und relevanter Messenger, soll Aufschluss darüber geben, wo der derzeitige Sicherheitsstand unter diesen mobilen Messengern anzusiedeln ist. Dazu werden zunächst einige allgemeine Grundlagen zu Messengern und deren Sicherheit erörtert, auf denen basierend Vergleichskriterien der Sicherheitsaspekte festgelegt werden. Zudem werden die Grundlagen genutzt, um eine Auswahl an zu untersuchenden Messengern treffen zu können.

2. GRUNDLAGEN
Dieses Kapitel dient zur Definition relevanter Begrifflichkeiten, zur Feststellung zeitgemäßer, genutzter Sicherheitsstandards im Umfeld von Instant Messaging und daraus ableitend zur Beantwortung der Frage: „Was muss ein mobiler Instant Messenger erfüllen um sicher zu sein?“

2.1 Definitionen
Vorweg ist zu sagen, dass grundlegende Begriff der Kryptografie (wie symmetrische und asymmetrische Verschlüsselung, Nonce, Salz, Signatur oder Message Authentication Code (MAC)) hier zum Verständnis vorausgesetzt und somit nicht mehr erläutert werden.

Zunächst soll dazu der Begriff des Mobilen Instant Messengers abgegrenzt werden. Dem allgemeinen Instant Messaging sind grundsätzlich alle Internetdienste zuzuschreiben, die eine „text- oder zeichenbasierte Kommunikation in Echtzeit ermöglichen“ [13]. Im Fall des mobilen Instant Messaging findet diese Kommunikation zwischen mobilen Endgeräten wie Smartphones oder Tablets statt. In diesem Artikel beziehen sich alle folgenden Bezeichnungen wie „Messenger“ oder „Instant Messenger“, wenn nicht anders beschrieben, der Einfachheit halber immer auf die mobile Variante.

Ein weiterer wichtiger Aspekt aller üblichen Messenger ist das Vorhandensein einer Liste an Kontakten, mit denen Nutzer auf synchronem Wege Nachrichten austauschen können [24]. Versendet also ein Nutzer eine Nachricht an einen Nutzer aus dieser Liste, kommt sie direkt auf dem mobilen Endgerät seines Kommunikationspartners an; insofern er eine Verbindung zum Internet aufgebaut hat. Andernfalls verweilt die Nachricht – wie auch bei gewöhnlichen nicht-mobilen Messengern – beispielsweise auf den Servern des jeweiligen Anbieters und wird ausgeliefert, sobald der Empfänger wieder präsent ist [26].

Um den Nutzern in diesem mobilen Umfeld die klassischen Ziele der Informationssicherheit – Vertraulichkeit, Integrität und die Verfügbarkeit ihrer Daten – bieten zu können, empfiehlt das Bundesamt für Sicherheit in der Informationstechnik (BSI) den Anwendungen den Einsatz von Authentisierungsverfahren, integritätssichernden Verschlüsselungsverfahren und kommunikationssichernden Mechanismen [8].

Durch entsprechende Standards sollen Kriterien wie diese festgelegt und umgesetzt werden. Dazu werden im folgenden Punkt verbreitete Kommunikations-Standards genauer untersucht.

2.2 Messenger-Sicherheit

Wenn Messenger heutzutage genutzt werden, so möchten sich Nutzer einiger grundlegender Dinge – die auch für das nicht-mobilen Messaging relevant sind – sicher sein. Wenn beispielsweise Alice also eine Nachricht zu einem weiteren Nutzer Bob versendet, so möchte sie sich sicher sein, dass [27]:

- Bob diese Nachricht erhält
- Bob die Nachricht nur einmal erhält
- Bob sich sicher sein kann, dass Alice die Nachricht gesendet hat
- Bob keine weiteren Informationen (zum Beispiel die IP-Adresse) über Alice ohne ihr Wissen übermittelt bekommt

Zudem soll es einem dritten Nutzer, Mallory, der eigentlich nichts mit der Konversation zwischen Alice und Bob zu tun hat, nicht möglich sein, dass er [27]:

- die Nachricht zwischen Alice und Bob lesen kann
- die Nachricht verändern oder fälschen kann

Erster der beiden Punkte beschreibt hier den Angriff auf die Vertraulichkeit und im zweiten Punkt wird ein Angriff auf die Integrität abgedeckt.

Ein Messaging Protokoll, welches neben der Vertraulichkeit und Integrität zudem die Kriterien Authentizität, Abstreitbarkeit und Folgenlosigkeit unterstützt, ist das Off-the-Record Messaging Protokoll (OTR) [35]. Dieses Protokoll soll als Überblick dienen, wie Sicherheit bei Instant Messengern umgesetzt werden kann. Dazu werden nun jene erwähnte Kriterien und ihre jeweiligen Umsetzungen detaillierter erklärt.

Zu Beginn einer Kommunikation über OTR findet zum Schutz der Authentizität der Authenticated Key Exchange (AKE) nach einer Variante des SIGMA-Protokolls statt [23]. Der Grundgedanke hierbei ist, dass zunächst ein nicht authentisierter Diffie-Hellmann[1] Schlüssel-Austausch (D-H) stattfindet, um anschließend in diesem verschlüsselten Kanal die Authentisierung durchzuführen. Folgendermaßen läuft dies schematisch ab [23]:

1. Alice und Bob besitzen jeweils einen Langzeit-Authentisierungsschlüssel pub_A, respektive pub_B. Anfangs einigen sie sich zudem auf eine für den D-H notwendige (hier 1536 Bit große) Primzahl g und einer Primitivwurzel g modulo p.

2. Bob initiiert nun den AKE mit Alice. Dazu wählt er die Zufallswerte r und x, verschlüsselt anschließend sein g^x über Advanced Encryption Standard (AES[2]) mit dem Schlüssel r und sendet dies zusammen mit einem Hash von g^x an Alice.

3. Auch Alice wählt nun ihren geheimen Zufallswert y und sendet g^y an Bob.

4. Bob berechnet sich nun den Schlüssel $s = (g^y)^x$, falls g^y auch den D-H-Kriterien ($2 \leq g \leq p-2$) entspricht. Anschließend erzeugt er sich zwei AES-Schlüssel c und c‘, und vier MAC-Schlüssel m1, m1‘, m2 und m2‘, indem er s auf verschiedene Arten hasht. Den ersten der MAC-Schlüssel, m1, benutzt er nun um einen MAC M_B, aus seinem g^x, Alices g^y, seinem Langzeitschlüssel pub_B und einer laufenden Nummer $keyid_B$, zu erzeugen. Zu guter Letzt signiert Bob diesen M_B mit seinem Langzeitschlüssel pub_B. Diese Signatur wird nun zusammen mit pub_B und $keyid_B$ zu einem Paket X_B gefasst, mit dem Schlüssel c per AES verschlüsselt und zusammen mit einem weiteren MAC (und m2 als Schlüssel) von diesem AES-Paket, und dem Zufallswert r an Alice gesendet.

5. Alice kann nun mit r das im zweiten Schritt erhaltene g^x entschlüsseln, den zugehörigen Hash abgleichen, nach den D-H-Kriterien verifizieren und wie auch Bob das gemeinsame $s =(g^x)^y$ berechnen. Nun berechnet sie sich durch die gleichen Hashfunktionen auf s wie Bob auch dieselben AES- und MAC-Schlüssel. Mit diesen Werten kann sie nun rückwärts die von Bob empfangene Nachricht schrittweise verifizieren und entschlüsseln und durch Bobs pub_B auch die Signatur verifizieren.

Auch Alice erzeugt sich nun (auf gleichem Wege wie Bob, aber stattdessen mit den Schlüsseln c‘, m1‘ und m2‘) respektive ein M_A, X_A (inklusive pub_A-Signatur), verschlüsselt letzteres Paket, benutzt wieder einen MAC und sendet dies alles an Bob.

6. Ebenso geht Bob nun vor, indem er die Nachrichten schrittweise verifiziert und entschlüsselt, bis auch er durch pub_A die Signatur von Alice verifizieren kann und sich jetzt sicher sein kann, dass die Person mit der er das Geheimnis s teilt auch Besitzerin von pub_A, also Alice, ist.

Ähnlich geht es beim OTR-Nachrichten-Austausch von Statten, um Vertraulichkeit und Integrität zu unterstützen. Dort wird auf eine hybride Verschlüsselung aus AES und Diffie-Hellmann gesetzt [23]. Jedoch wird hier bewusst auf Signaturen verzichtet, um eine Abstreitbarkeit zu gewährleisten. Das heißt, dass sich zum Zeitpunkt der Kommunikation Alice und Bob zwar sicher sein können, dass die Nachricht vom jeweiligen Gegenüber kommt, aber, dass im Nachhinein eine beweisbare und verbindliche Zuordnung einer Nachricht zu einem Absender nicht möglich ist. Dazu wird an jede Nachricht ein MAC mit einem Schlüssel angehängt, welchen zu diesem Zeitpunkt nur Alice und Bob wissen können. Dadurch kann Bob verifizieren, dass die Nachricht von Alice kam. Bei seiner nächsten Nachricht hängt Bob jedoch alle zuvor verwendeten MAC-Schlüssel (also auch den letzten) mit an die Nachricht, so dass im Nachhinein jeder Angreifer die Nachricht mit dem MAC versehen hätte können und somit nicht mehr

[1] Siehe dazu bspw. Diffie, Hellmann 1976 [12]

[2] Siehe dazu bspw. Bogdanov et al. 2011 [6]

beweisbar ist, dass Alice diese Nachricht mit dem MAC verschickt hat [23]. Bekannter ist in diesem Zusammenhang meist das Gegenteil bewirkende Prinzip der Nicht-Abstreitbarkeit, was sicherstellen soll, dass ein Kommunikationspartner seine Aussagen im Nachhinein nicht leugnen kann [30].

Da beim Nachrichten-Austausch über OTR für jede Nachricht ein neuer Schlüssel s berechnet wird, ist auch das Prinzip der Folgenlosigkeit unterstützt. Wenn also ein Schlüssel kompromittiert wird, so kann im Nachhinein trotzdem nicht die komplette Kommunikation entschlüsselt werden [23]. Für den direkten Einsatz im mobilen Umfeld ist OTR aber ungeeignet, da beide Kommunikationspartner online sein müssen, um eine sichere Verbindung aufrecht zu erhalten.

Wie hier bereits deutlich wird, unterliegen Verschlüsselungen einem sehr komplexen Thema und nur die Veröffentlichung eines solchen Verfahrens gewährleistet volle Sicherheit, da dadurch Sicherheitslücken von einer Vielzahl an Leuten aufgedeckt und wenn möglich auch schnellstmöglich wieder geschlossen werden können. Es ist also sinnvoll bei Messengern auf bereits bekannte und geprüfte Verschlüsselungsverfahren zu setzen, oder die eigenen Verfahren offen zu legen. Dieser Ansatz ist deutlich empfehlenswerter als die sog. „Security through Obscurity“, bei der versucht wird durch das Zurückhalten von Informationen zur Verschlüsselung mehr Sicherheit zu gewährleisten [20].

2.3 Vorbereitungen

Aus den bisher erworbenen Informationen werden im folgenden Kapitel Kriterien abgeleitet, anhand derer die verschiedenen Messenger auf ihre Sicherheit hin untersucht werden können. Außerdem wird eine Auswahl an Messengern getroffen, welche dieser Sicherheitsprüfung im nächsten Kapitel unterzogen werden sollen.

Zunächst werden für jeden Messenger einige allgemeine Informationen wie Funktionsumfang, Verbreitung oder Kostenpunkt erläutert. Diese sind für Nutzer häufig interessanter als die Sicherheitsbelange. Im Falle dieses Artikels spielen sie jedoch nur eine Nebenrolle. Anschließend sollen die folgenden sicherheitsrelevanten Kriterien untersucht werden:

Infrastruktur
Zunächst soll die komplette Infrastruktur in Hinblick auf die Vertraulichkeit beleuchtet werden. Das bedeutet, dass auch ohne einen Angriff niemand ungewünschten Einblick in Daten des Nutzers erhalten darf. Gerade im Umfeld von Messengern ist hier wissenswert, wie mit Kontakten und der Kontaktliste umgegangen wird. Aber auch das Verwalten der Schlüssel und somit der mögliche Zugang zu diesen – zum Beispiel von Regierungsorganisationen – ist im Sinne der Vertraulichkeit sicherheitsrelevant. Außerdem sind Informationen über das betreibende Unternehmen des jeweiligen Messengers hilfreich, um den Angaben über die angewendete Verschlüsselung auch einigermaßen vertrauen zu können.

Authentizität
Wie der Name bereits vermuten lässt, soll unter diesem Punkt einerseits Wert auf die erstmalige Registrierung und eventuell zugehörige Verifizierung – falls beispielsweise die Rufnummer zur Identifikation dient – gelegt werden. Andererseits ist das darauf folgende Authentisierungsverfahren aller danach stattfindenden Kommunikationen von Interesse.

Verschlüsselung
Die eingesetzte Verschlüsselung des Messengers sollte aktuellen Sicherheitsstandards entsprechen und somit die Prinzipien der Vertraulichkeit, Integrität und Folgenlosigkeit unterstützen. Das heißt, dass aktuell als unsicher geltende Verfahren sich selbstsprechend negativ auf die Reputation des Messengers auswirken. Ein Nutzer muss sich also sicher sein können, dass seine Nachrichten nicht von unerwünschten Personen gelesen oder verändert werden können, auch wenn im Nachhinein ein Schlüssel kompromittiert werden würde. Das Prinzip der Abstreitbarkeit wird hier nachträglich nicht als Kriterium herangezogen, da sich nach der Analyse herausstellte, dass dies aktuell in keinem der untersuchten Messenger eine Rolle spielt und somit vernachlässigbar ist.

Transparenz
Wie beschrieben, ist das Offenlegen der eingesetzten Verschlüsselungsverfahren und bestmöglich des zugehörigen Quellcodes ein Vorteil und soll somit ebenfalls untersucht werden. Zumindest sollte ein Unternehmen aber Informationen zu den Verschlüsslungs- und Authentisierungsverfahren bereitstellen.

Um all diese Kriterien bestmöglich untersuchen zu können, werden nur Messenger in Betracht gezogen, die zumindest zur Verschlüsselung Informationen bieten. Zudem ist die Bekanntheit, gemessen an den groben Download-Zahlen in Googles App-Store für Android – der zweite große App-Store für iOS bietet hierzu keine Statistiken an – und gemessen an Medienberichten, von Relevanz. Dennoch ist auf Grund der Vielzahl an Messengern und des begrenzten Umfangs dieses Artikels nur eine stichprobenartige und keinesfalls vollständige Auswahl an mobilen Instant Messengern möglich.

Alle ab diesem Punkt folgenden Angaben über die Messenger sind, wenn nicht anders vermerkt, von deren jeweiligen offiziellen Internetauftritten oder den zugehörigen Vorstellungsseiten in Googles [16] und Apples [2] App-Store entnommen.

Von den bereits in der Einführung erwähnten Marktführern WhatsApp, Skype und Facebook Messenger kann letzterer bereits ausgeschlossen werden, da Facebook keinerlei Informationen zur Verschlüsselung bereitstellt. Ebenso ist es mit dem mittlerweile recht verbreiteten VoIP-Messenger (Voice over IP) Viber. Auch hier sind weder Informationen zur Verschlüsselung, geschweige denn zum Firmensitz [7] zu finden. Ebenso hüllt sich Google Hangouts über eine verwendete Verschlüsselung in den Mantel des Schweigens. Da die Auswahl zwangsweise eingegrenzt werden muss, wird auch Skype nicht untersucht. Denn die Software wurde vorwiegend für den Desktop entwickelt, wird eher für VoIP als für Instant Messaging genutzt und stellt nur begrenzte Informationen [29] zu Verschlüsselungen bereit. WhatsApp wird auf Grund der sehr hohen Nutzerzahlen in der folgenden Analyse mit berücksichtigt.

Neben der Marktgröße WhatsApp sollen nun zudem die in relevanten Medien hervorgehobenen und als sicher angepriesenen Neuerscheinungen Threema, Whistle.im und myEnigma untersucht werden [17][28].

3. MOBILE INSTANT MESSENGER

Folgende Auswahl von mobilen Instant Messengern wird unter Betracht der zuvor erarbeiteten Kriterien auf ihren aktuellen Sicherheitsstand untersucht. Wenn in den allgemeinen Beschreibungen von einem „gängigen Funktionsumfang“ die Rede ist, so ist damit die Möglichkeit gemeint, Nachrichten, Gruppen-Nachrichten, Bilder und Videos versenden zu können.

3.1 WhatsApp

Mit einer Verbreitung von mittlerweile etwa 400 Millionen Nutzern ist WhatsApp eine Marktdurchdringung gelungen [21]. Der Messenger ist derzeit ein Jahr lang kostenlos erhältlich, danach wird eine jährliche Gebühr fällig. Nutzbar ist er auf den Plattformen Android, iOS, Windows Phone, BlackBerry und den beiden Nokia-Modellen S60 und S40. Außerdem wird der gängige Funktionsumfang voll unterstützt. In Medien machte der Messenger jedoch häufiger eher durch Sicherheitslücken auf sich aufmerksam [3]. Dies mag das subjektive Vertrauen etwas schmälern, im Folgenden ist aber nur der aktuelle objektive Stand von Bedeutung.

Infrastruktur

Das hinter der Anwendung stehende Unternehmen WhatsApp Inc. mit Sitz in den USA – genauer gesagt im Silicon Valley – wurde von zwei ehemaligen Yahoo-Entwicklern gegründet. An deren Server werden nach eigenen Angaben die Rufnummern des Telefonbuchs eines Nutzers verschlüsselt gesendet, um abzugleichen ob einer dieser Kontakte den Messenger ebenfalls nutzt. Zudem werden Nachrichten nur solange auf den Servern behalten, bis der Empfänger sie erhalten hat. Weitere oder detailliertere Informationen zur Speicherung eventueller Schlüssel, Metadaten oder den Rufnummern werden von offizieller Seite nicht gegeben.

Authentizität

Auch zur Authentizität oder Authentisierung sind keine offiziellen Informationen vorhanden. Wenn man WhatsApp jedoch erstmalig nutzt, so wird die Identität anhand der Rufnummer über eine erhaltene SMS verifiziert. Der Chefentwickler des Messaging-Clients Adium, Alkemade T., versuchte das hierbei genutzte Verfahren nachzuvollziehen [1]. Zudem belegte er seine Ergebnisse teilweise, indem er einerseits die App dekompillierte und andererseits ein zur Ausnutzung seiner gefundenen Sicherheitslücken theoretisch nutzbares Skript veröffentlichte. Da diese aktuell die einzigen Informationen zu diesem Sachverhalt darstellen, sollen sie hier grob geschildert werden.

So werde bei erstmaliger Nutzung die Rufnummer über eine https-Verbindung an die WhatsApp-Server gesendet. Diese senden wiederum eine SMS mit einem Authentifizierungs-Code an das Endgerät des Nutzers, von wo aus dieser Code über https an die Server gesendet wird, um sich zu authentisieren. Bei Erfolg erhält das Nutzer-Endgerät ein Passwort, was für zukünftige Kommunikationen benötigt wird. Dabei wird nämlich ein Challenge-Response-Authentifizierungsverfahren eingesetzt, welches jenes Passwort als gemeinsames Geheimnis zur Lösung der Challenge nutzt. Dazu wird es verwendet, um unter Einsatz der Password-Based Key Derivation Function 2 (PBKDF2[3]) einen Session-Key zu erzeugen, welcher Rufnummer, Challenge und einen Zeitstempel RC4-verschlüsselt. Dieser Cyphertext wird anschließend, mit einem HMAC als Prüfsumme, als Response versendet. Serverseitig wird die gleiche Aktion ausgeführt und anschließend werden die Ergebnisse zur Authentifizierung verglichen.

Verschlüsselung

Offizielle Angaben zur Verschlüsselung liegen ebenfalls nicht vor. WhatsApp schreibt nur, dass die Kommunikation zwischen dem Endgerät und den Servern des Unternehmens vollständig verschlüsselt sei. Dies heißt aber nicht zwingend, dass auch eine Ende-zu-Ende-Verschlüsselung (zwischen den beiden kommunizierenden Nutzern) vorliegt. Laut Alkemade wird auch bei der Verschlüsselung der Nachrichten RC4 verwendet, wobei hier ein grober Fehler in der Kryptographie gemacht werde [1]. WhatsApp nutzt demnach für je eine eingehende und ausgehende Nachricht denselben Session-Key für die Stromchiffrierung mit RC4, weshalb die jeweiligen Klartexte mit derselben Zufallsfolge per XOR verknüpft werden. Die einfache mathematische Gleichung (A xor X) xor (B xor X) = A xor B, wobei A und B für die ein- und ausgehende Nachricht und X für die Zufallsfolge stehen, zeigt, dass wenn beispielsweise Teile von A bekannt sind, ganz einfach auch Teile von B entschlüsselt werden können.

Transparenz

Der Quellcode ist ebenso wie die eingesetzten Verschlüsselungsverfahren nicht öffentlich. Die Betreiber von WhatsApp machen aber auch keine genauen Angaben zu ihrer Verschlüsselung. Sie bestätigen, wie erwähnt, nur, dass eine Verschlüsselung zwischen Client und Server vorhanden ist. Dies macht es nötig auf analysierende Drittparteien zu setzen um überhaupt etwas über die Sicherheit in Erfahrung zu bringen. Jedoch müssen und können deren Angaben nicht hundertprozentig stimmen. Zudem sind deren Annahmen nur schwer belegbar.

3.2 Threema

Der Schweizer Messenger "Threema" ist derzeit kostenpflichtig für die Betriebssysteme Android und iOS zu erwerben. Die Verbreitung kann nicht genau angegeben werden, da für iOS keine Zahlen veröffentlicht werden und da Threema für Android neben Googles App-Store noch einen eigenen Store führt. In Googles App-Store wurde der Messenger zwischen 10.000 und 50.000 mal heruntergeladen (genauere Daten werden von Google nicht bereitgestellt). Threema unterstützt außerdem den gängigen Funktionsumfang, wobei Gruppennachrichten derzeit nur für iOS verfügbar sind.

Infrastruktur

Threema wird vom Unternehmen Kasper Systems, mit Sitz in der Schweiz, betrieben und besteht aus dem Gründer und bei Bedarf aus einigen freien Entwicklern [5]. Auf den Servern des Unternehmens liegen die öffentlichen Schlüssel der Nutzer, die IDs der Nutzer (eventuell verknüpft mit gehashter Rufnummer oder Email-Adresse, sofern er das möchte) und die verschlüsselten Nachrichten, solange bis sie dem Empfänger zugestellt werden können. Im Anschluss werden die Nachrichten von den Servern gelöscht. Weitere Kommunikations-Meta-Daten werden nach eigenen Angaben nicht geloggt. Um den öffentlichen Schlüsseln auf den Servern des Anbieters nicht trauen zu müssen, können diese auch offline verifiziert werden, indem man einen Kommunikationspartner persönlich trifft und die Schlüssel zum Beispiel über einen QR-Code abgleicht. Der private Schlüssel eines Nutzers verlässt dessen Endgerät niemals und kann dort zusätzlich über eine Passphrase geschützt werden. Zur Generierung des privaten Schlüssels führt der Nutzer zufällige Wisch-Gesten auf seinem Smartphone aus, um so einen möglichst zufälligen privaten Schlüssel zu erhalten.

Authentizität

Zur Kommunikation eines Nutzers ist somit mindestens der öffentliche Schlüssel des Empfängers notwendig. Optional kann man seine Rufnummer oder Email-Adresse als Hashwert an den Server schicken, um diese abgleichen zu lassen und somit neue Kontakte zu finden, andernfalls ist man nur über eine ID identifizierbar. Die Funktionsweise der eigentlichen Authentisierung wird nicht genau offen gelegt. Laut Angaben der App-Webseite

[3] Siehe dazu bspw. RFC 2898

muss sich die App aber wohl mit einem, fest in die Anwendung einprogrammierten, öffentlichen Schlüssel des Servers bei den legitimen Servern authentisieren. Diese authentifizieren die jeweilige ID dann mit dem passenden privaten Schlüssel. Zur Prüfung der Authentizität während der Kommunikation wird mit jeder Nachricht zudem ein 128 Bit langer MAC mitgeschickt.

Verschlüsselung
Threema nutzt als 255 Bit starke asymmetrische Verschlüsselung die Elliptic Curve Cryptography (ECC), umgesetzt durch die NaCl Cryptography Library[4], eine offene externe Bibliothek. Dazu wird für jede Nachricht über Elliptic Curve Diffie-Hellman (ECDH) inklusive Hashfunktion und Nonce ein einmaliger 256 Bit langer Schlüssel erstellt, welcher für die symmetrische Verschlüsselung der Nachricht mit XSalsa20 verwendet wird. Der bereits erwähnte MAC dient in diesem Falle ebenfalls der Verhinderung von Manipulationen an den Nachrichten. Als zusätzlichen Schutz vor eventuellem Erraten der Inhalte anhand einer Analyse der Datengröße wird an jede Nachricht eine zufällige Menge an Füllbytes angehängt.

Transparenz
Der Quellcode der Anwendung an sich ist nicht öffentlich einsehbar und eine Prüfung von Experten kann sich das Unternehmen nach eigenen Angaben nicht leisten [5]. Die eingesetzte Library, NaCl, kann hingegen eingesehen werden und die Vorgehensweise wird in einem Paper beschrieben [4]. Zudem ist in Threema ein Validierungs-Log eingebaut, welches über ein kleines C-Programm beweisen soll, dass diese Anwendung auch wirklich NaCl benutzt [36].

3.3 myEnigma

Der Messenger, dessen Name an die Rotor-Schlüsselmaschine Enigma angelehnt ist, kann aktuell kostenlos auf den Plattformen Android, iOS und Blackberry genutzt werden. Die Verbreitung ist, zumindest in Android (zwischen 10.000 und 50.000 Installationen über Googles App-Store), noch eher gering und in etwa mit der von Threema gleichzusetzen. Mit myEnigma können neben dem gängigen Funktionsumfang – auch Gruppenchats werden unterstützt – zudem verschlüsselte SMS verschickt werden.

Infrastruktur
Der Messenger wird von der Schweizer Software-Firma Qnective AG entwickelt. Diese entwickelt und vertreibt Software-Sicherheitslösungen sowohl für den Unternehmens-, als auch für den Konsumenten-Bereich [25]. Deren Server erreichen beim automatischen Telefonbuch-Abgleich des Messengers nur die Hashwerte der Rufnummern. Diese werden über eine Transport Layer Security (TLS) -Verbindung an die Server gesendet und dort mit den gespeicherten Hashwerten abgeglichen, um eventuelle Nutzer des Messengers in die Kontaktliste aufnehmen zu können. Namen zu diesen Nummern werden nicht mit abgespeichert, sondern lokal auf dem Endgerät ergänzt. Außerdem wird die Rufnummer an ein Gerät gekoppelt. Möchte man myEnigma auf einem anderen Gerät mit derselben Rufnummer nutzen, so wird im gekoppelten Gerät eine Warnung gezeigt. Erst mit dortiger Zustimmung (zum Beispiel über die Email-Adresse) ist ein Wechsel des Gerätes möglich.

Authentizität
Bei erstmaliger Nutzung setzt die App auf einen dualen Verifikationsprozess [22]. So muss ein Nutzer seine Rufnummer und eine Email-Adresse angeben. Im Anschluss bekommt er per SMS einen Verifizierungscode geschickt und das zugehörige Einweg-Passwort an die angegebene Email-Adresse. Nach Eingabe dieses Passworts ist der Nutzer registriert und muss sein Passwort auch sogleich ändern. Für die Authentisierung am Server wird aus jenem Passwort per PBKDF2 auf dem Client zunächst ein 256 Bit Schlüssel generiert und anschließend über eine TLS-Verbindung an den Server geschickt. Dadurch verlässt das eigentliche Passwort den Client nie. Anschließend wird der Schlüssel inklusive Salz als Hashwert auf den Servern gelagert und für folgende Authentifizierungen genutzt. Für die Authentisierung beim Kommunikationspartner werden HMACs eingesetzt, was im folgenden Abschnitt mit erläutert wird.

Verschlüsselung
Die Verbindung zwischen Client und Server ist über das TLS-Protokoll geschützt und nutzt dazu eine 2048 Bit Diffie-Hellman Schlüsselvereinbarung und ein 2048 Bit RSA Zertifikat [22]. Ebenso wird für die Ende-zu-Ende-Verschlüsselung Diffie-Hellman (2048 Bit) eingesetzt. Umgesetzt wird dies nach dem Internet Key Exchange (IKE) Protokoll, wobei die Parameter für die Gruppe auf RFC 3526 basieren. Die aus dem Diffie-Hellman-Geheimnis resultierenden Session-Schlüssel werden anschließend für die symmetrische Verschlüsselung und Authentisierung per AES, respektive HMAC, verwendet. Die geheime private (für D-H notwendige) Zahl wird – bei aktiver Kommunikation in dieser Zeit – nach dreieinhalb Tagen erneuert, wodurch eine neue Schlüsselvereinbarung angestoßen wird. Sobald ein Nutzer jedoch länger als 7 Tage offline war, wird bei der nächsten Kommunikation ebenfalls eine neue Schlüsselvereinbarung fällig. Durch alle in diesem Abschnitt genannten Mechanismen wird auch die Folgenlosigkeit garantiert.

Transparenz
Die Betreiber von myEnigma stellen den Quellcode nicht öffentlich zu Verfügung, da sie eine Veröffentlichung nicht als Garantie dafür sehen, dass dieser Code auch wirklich in der App genutzt werde [22]. Außerdem möchten sie dadurch verhindern, dass ihr Code aus sicherheitsrelevanten und kommerziellen Gründen in falsche Hände gerate. Auf Anfrage ist jedoch ein Whitepaper von Qnective erhältlich, welches die zu Grunde liegenden Sicherheitsverfahren beschreibt und auf welche sich in diesem Kapitel bezogen wurde.

3.4 Whistle.im

Whistle.im ist derzeit im Beta-Stadium und wird kostenlos angeboten. Auch dieser Messenger weist laut Googles App-Store eine Verbreitung von etwa 10.000 bis 50.000 Installationen auf Android auf. Neben jener Version für Android wird zudem eine Browser-Version angeboten. Eine iOS- und Windows Phone App sollen folgen. Für den gängigen Funktionsumfang fehlt außerdem noch das Versenden von Bildern und Videos. Gewöhnliche Nachrichten und Gruppenchats sind bereits möglich.

Infrastruktur
Entwickelt wird Whistle.im von zwei deutschen Studenten, welche die komplette Finanzierung des Messengers über Spenden abdecken möchten. Auf deren Servern speichern sie für jeden Nutzer die Id, sein Passwort (gehasht inkl. Salz mit Bcrypt), einen zufällig generierten AES-Schlüssel für Poll- und Push-Benachrichtigungen, den Online-Status und einige Zeitangaben zur Erstellung des Accounts, zum Zeitpunkt des letzten Logins und ein Timestamp für Poll-Benachrichtigungen. Außerdem wird

[4] Siehe dazu bspw. Bernstein D. 2009 [4]

dort aber neben dem öffentlichen Schlüssels des Nutzers im Klartext auch sein privater Schlüssel (per PBKDF2 verschlüsselt) abgelegt. Letzteres hat den Hintergrund, dass die Möglichkeit gegeben werden soll sich mit mehreren Clients (z.B. Smartphone und Browser) einzuloggen. Auch wenn der private Schlüssel auf den Servern verschlüsselt vorliegt, stellt dies dennoch ein Sicherheitsrisiko dar, weswegen der Schlüssel auch vom Nutzer komplett selbst verwaltet werden kann.

Neben den Nutzerdaten werden auf den Servern außerdem Kommunikationsdaten (Kontakte, Gruppen, verschlüsselte Nachrichten, Benachrichtigungs-Daten) abgelegt. Diese sind auf den Webseiten des Messengers detailliert nachzulesen.

Authentizität
Eine Verknüpfung mit Rufnummer oder Email-Adressen wie bei anderen Messengern liegt hier nicht vor. Jedem Nutzer wird stattdessen eine Id zugewiesen, zu welcher er sich ein Passwort für die Authentisierung wählt. Dieses wird wie erwähnt Bcrypt-gehasht auf den Servern gespeichert und jener Hash zur Authentifizierung durch die Server genutzt. Um nun mit einen anderen Nutzer kommunizieren zu können, muss seine Id bekannt sein. Da man diese im Normalfall direkt von seinem Kommunikationspartner bekommt, ist man nicht auf die Sicherheit eines Datenabgleichs der Telefonliste oder ähnlichem angewiesen. Als weiteren Sicherheitsmechanismus wird ein Nutzer benachrichtigt, sobald sich der öffentliche Schlüssel eines Kontakts ändert.

Verschlüsselung
Whistle.im setzt wie die anderen Messenger auch auf eine hybride Verschlüsselung, zusammengesetzt aus 256 Bit AES-CBC für die symmetrische und 2048 Bit RSA für die asymmetrische Verschlüsselung. Signiert werden die Nachrichten zudem mit der Hashfunktion SHA-1 im Zusammenspiel mit RSA. Wie gewohnt wird hierzu mit dem eigenen privaten Schlüssel das Dechiffrat des Hashwerts der Nachricht als Signatur gebildet. Für die Umsetzung der Android-Version nutzen die Entwickler hier beispielsweise die frei verfügbaren Java-Packages java.security und javax.crypto [15].

Transparenz
Die von den Entwicklern gemachten Aussagen zur Verschlüsselung lassen sich durch den offen gelegten Quellcode ihrer Verschlüsselungsklasse für Android und die Browser-Version belegen [15]. Somit sind nicht nur die eingesetzten Verfahren, sondern auch die Umsetzung dieser bekannt und können unter einer GNU General Public License genutzt werden. Weitere Klassen oder die serverseitige Umsetzung sind jedoch nicht offengelegt. Dort vorhandene Sicherheitsrisiken könnten somit unentdeckt bleiben.

4. FAZIT

Wie bereits auf den ersten Blick zu erkennen ist, nutzt ein Großteil aller Messenger-Nutzer WhatsApp. Eben jener bietet aber kaum Informationen zur Sicherheit und ist immer wieder wegen Sicherheitsrisiken in den Medien vertreten. Das Unternehmen dahinter bessert jedoch immer wieder nach, sollte aber zudem noch mehr Transparenz liefern.

Whistle.im, der Messenger aus Deutschland, ist derzeit noch in der Beta-Phase und auch in puncto Umsetzung noch aufholbedürftig [10]. Die Intention hinter whistle.im und die Offenlegung des Quellcodes vermitteln jedoch einen guten Eindruck. Momentan ist der Einsatz insgesamt aber noch nicht empfehlenswert, in Zukunft aber sicherlich einen Gedanken wert.

Die empfehlenswertesten Messenger dieser Auswahl sind aktuell Threema und myEnigma. Beide setzen auf aktuelle Sicherheitsverfahren und eine Ende-zu-Ende-Verschlüsselung, wodurch sie nicht gezwungen werden können, eventuelle Nachrichten preiszugeben. Threema besticht hier durch Sicherheitsfunktionen wie dem Offline-Schlüsselaustausch oder einer Zufallszahl-Erstellung durch Wisch-Gesten, wird aber größtenteils nur von einer Person entwickelt. Dies kann zwar den Vorteil der Unabhängigkeit von Unternehmen oder Organisationen bedeuten, heißt aber auch, dass man den Fähigkeiten und der Sicherheit einer einzelnen Person vertrauen muss. MyEnigma hingegen hat solche sicherheitsbringenden Zusatzfunktionen nicht, wird aber von einem Unternehmen entwickelt welches bereits seit einigen Jahren im geschäftlichen Umfeld Sicherheitslösungen für Kommunikation anbietet.

Letztendlich ist die Nutzung eines solchen mobilen Instant Messengers immer eine Frage des Vertrauens, sei es zum Verfahren oder zum dahinterstehenden Entwickler bzw. Betreiber. Zudem ist es sicherlich möglich, dass sich auf dem höchst frequentierten Markt jener Messenger ein noch sicherer Vertreter finden lässt, zumal immer wieder neue Applikationen in diesem Bereich auftreten. So wird derzeit beispielsweise ein weiterer Instant Messenger, unter anderem vom Gründer von Flattr, entwickelt. Heml.is soll der Messenger heißen und er verspricht laut Webseite Technologien wie das Extensible Messaging and Presence Protocol (XMPP) in Verbindung mit Pretty Good Privacy (PGP) Verschlüsselung.

Wenn jedoch der Anwender einer Software keinen Wert auf Sicherheit legt und zum Beispiel seine Passwörter nicht adäquat wählt und unter Verschluss hält, so hilft auch der sicherste Messenger nichts.

5. REFERENCES

[1] Alkemade T. (2013): Piercing through WhatsApp's Encryption, 08.10.2013, https://blog.thijsalkema.de/blog/2013/10/08/piercing-through-whatsapp-s-encryption/, Letzter Zugriff 18.01.2014

[2] Apple App Store: https://itunes.apple.com/de/genre/ios/id36, Letzter Zugriff 17.01.2014

[3] Beer K. (2013): Whatsapp zählt Nutzerrekord: 400 Millionen Aktive pro Monat, heise online, 20.12.2013, http://www.heise.de/newsticker/meldung/Whatsapp-zaehlt-Nutzerrekord-400-Millionen-Aktive-pro-Monat-2070768.html, Letzter Zugriff 18.01.2014

[4] Bernstein D. (2009): Cryptography in NaCl, University of Illnois, Chicagor, http://cr.yp.to/highspeed/naclcrypto-20090310.pdf, Letzter Zugriff 18.01.2014

[5] Beuth P. (2013): Threema – Ein App, um die NSA zu ärgern, Zeit Online, 14.08.2013, http://www.zeit.de/digital/mobil/2013-07/threema-app-manuel-kasper, Letzter Zugriff 18.01.2014

[6] Bogdanov A., Khovratovich D., Rechberger C. (2011): Biclique Cryptanalysis of the Full AES, *in:*ASIACRYPT 2011, Springer, p. 344-371, http://research.microsoft.com/en-us/projects/cryptanalysis/aesbc.pdf, Letzter Zugriff 17.01.2014

[7] Bulban F., Rohwetter M. (2012): Telefon-App Viber – Ihre Nummern, bitte, Zeit Online, 26.02.2012, http://www.zeit.de/2012/09/Telefonsoftware-Viber/seite-2, Letzter Zugriff 17.01.2014

[8] Bundesamt für Sicherheit in der Informationstechnik: Sicherheitslösungen und Konzepte, Mobile Security, https://www.bsi.bund.de/DE/Themen/weitereThemen/MobileSecurity/Sicherheitsloesungen/sicherheitsloesungen.html, Letzter Zugriff 15.01.2014

[9] Böck H. (2013): Verschlüsselung von Whatsapp ist unsicher, golem.de, 08.10.2013, http://www.golem.de/news/instant-messaging-verschluesselung-von-whatsapp-ist-unsicher-1310-102016.html, Letzter Zugriff 17.01.2014

[10] CCC Hannover (2013): Whistle.im – Fuckup as a Service, nexus, http://hannover.ccc.de/~nexus/whistle.html, Letzter Zugriff 19.01.2014

[11] CureSec (2013): Phishing Google Wallet and Paypal by abusing Whatsapp, 25.10.2013, CVE-2013-6274, Berlin

[12] Diffie W., Hellman M. (1976): New Directions in Cryptography, *in:* IEEE Transactions on Information Theory. 22, NR. 6, p. 644-654, http://www-ee.stanford.edu/~hellman/publications/24.pdf, Letzter Zugriff 17.01.20114

[13] Gabler Wirtschaftslexikon: "Instant Messaging", Springer Gabler Verlag (Hrsg.), http://wirtschaftslexikon.gabler.de/Archiv/81864/instant-messaging-v8.html, Letzter Zugriff 15.01.2014

[14] Gabler Wirtschaftslexikon: „Mobile Business", Springer Gabler Verlag (Hrsg.), http://wirtschaftslexikon.gabler.de/Archiv/498345196/mobile-business-v2.html, Letzter Zugriff 15.01.2014

[15] Github: whistle-im / client / android, https://github.com/whistle-im/whistle-im/tree/master/client/android, Letzter Zugriff 19.01.2014

[16] Google Play Store: https://play.google.com/store?hl=de, Letzter Zugriff 17.01.2014

[17] Gruber A. (2013): Die vermeintlich sicheren Alternativen zu WhatsApp, Zeit Online, 10.08.2013, http://www.zeit.de/digital/mobil/2013-08/sichere-messenger-alternative-whatsapp/, Letzter Zugriff 18.01.2014

[18] Heise Security (2013): WhatsApp-Verschlüsselung ruft Zweifel hervor, 08.10.2013, http://www.heise.de/security/meldung/WhatsApp-Verschluesselung-ruft-Zweifel-hervor-1974767.html, Letzter Zugriff 15.01.2014

[19] Heise Security (2012): Schnüffel-Tool zeigt fremde WhatsApp-Nachrichten an, 11.05.2012, http://www.heise.de/security/meldung/Schnueffel-Tool-zeigt-fremde-WhatsApp-Nachrichten-an-1574066.html, Letzter Zugriff 15.01.2014

[20] Herrero J. (2009): Vulnerabilities and Attack Protection in Security Systems Based on Biometric Recognition, p. 7-8

[21] Koum J. (2013): 400 Million Stories, blog.whatsapp.com, 19.12.2013, http://blog.whatsapp.com/index.php/2013/12/400-million-stories/, Letzter Zugriff 18.01.2014

[22] MyEnigma Whitepaper, 2013, Qnective AG

[23] Off-the-Record Messaging Protocol version 3, https://otr.cypherpunks.ca/Protocol-v3-4.0.0.html, Letzter Zugriff 19.01.2014

[24] Patterson D., Baker C., Ding X., Kaufman S., Liu K., Zaldivar A. (2008): Online everywhere: evolving mobile instant messaging practices, *in:* UbiComp '08 Proceedings of the 10th international conference on Ubiquitous computing, p. 64-73, New York

[25] Qnective: Products and Solutions, https://www.qnective.com/products-services/overview.html, Letzter Zugriff 19.01.2014

[26] RFC 2778: A Model for Presence and Instant Messaging, 02/2000, http://www.ietf.org/rfc/rfc2778.txt, Letzter Zugriff 16.01.2014

[27] RFC 2779: Instant Messaging / Presence Protocol Requirements, http://www.ietf.org/rfc/rfc2779.txt, Letzter Zugriff 19.01.2014

[28] Sawall A. (2013): Verschlüsselter Whatsapp-Konkurrent kommt aus Deutschland, golem.de, 08.08.2013, http://www.golem.de/news/whistle-im-verschluesselter-whatsapp-konkurrent-kommt-aus-deutschland-1308-100879.html, Letzter Zugriff 17.01.2014

[29] Skype FAQ: Verwendet Skype Verschlüsselungstechniken?, https://support.skype.com/de/faq/FA31/verwendet-skype-verschlusselungstechniken, Letzter Zugriff 19.01.2014

[30] Sorge C., Gruschka N., Iacono L. (2013): Sicherheit in Kommunikationsnetzen, Oldenbourg Verlag

[31] Statista (2013): Anteil an der deutschen Gesamtbevölkerung, der die folgenden Geräte und Internetfunktionen nutzt, http://de.statista.com/statistik/daten/studie/260634/umfrage/hinterlassen-von-daten-im-internet-nach-speicherorten/, Letzter Zugriff 15.01.2014

[32] Statista (2013): Apps mit dem höchsten Wachstum der Nutzerzahl weltweit im 3. Quartal 2013 gegenüber dem 1. Quartal 2013, http://de.statista.com/statistik/daten/studie/279903/umfrage/apps-mit-dem-groessten-nutzerzuwachs-weltweit/, Letzter Zugriff 15.01.2014

[33] Statista (2013): Versand von Instant-Messaging-Nachrichten und SMS weltweit 2011 und 2012 und Prognose für 2013 (in Mr.d pro Tag), April 2013, http://de.statista.com/statistik/daten/studie/258398/umfrage/prognose-der-instant-messaging-nachrichten-vs-sms-weltweit/, Letzter Zugriff 15.01.2014

[34] Statista (2013): WhatsApp jetzt mit 300 Millionen aktiven Nutzern, 07.08.2013, http://de.statista.com/infografik/1337/whatsapp-aktive-nutzer-und-verschickte-nachrichten/, Letzter Zugriff 15.01.2014

[35] Stedman R., Yoshida K., Goldberg I. (2008): A user study of off-the-record messaging, *in:* Proceedings of the 4th symposium on Usable privacy and security, p. 95-104, New York

[36] Threema Encryption Validation, threema.ch, https://threema.ch/validation/, Letzter Zugriff 19.01.2014