# Standards

**S/Chat**

**Lastenheft**

*Elias Frantar, Wolfram Soyka, Gary Ye*

# 

Inhaltsverzeichnis

1 Einführung 2

2 Zielbestimmung 2

3 Produkteinsatz 2

4 Produktfunktionen 2

4.1 Verschlüsselung 2

4.2 Userfunktionen 3

5 Produktdaten 4

5.1 Private Daten 4

5.2 Kontaktdaten 4

5.3 Nachrichten 4

6 Zwingende Randbedingungen 5

6.1 Produktumgebung und Systemintegration 5

 Hardware Entwicklung: ADT fähiges Device (Mac, Linux, Windows) 5

 Software Entwicklung: Android Development Tool 5

 Hardware Laufumgebung: Android fähiges Device 5

 Software Laufumgebung: Android 2.3 aufwärts 5

6.2 Schnittstellen 5

7 Vertragsgegenstand 6

7.1 Lieferumfang 6

7.2 Produktleistungen 6

7.3 Produktbezogene Leistungen 6

8 Qualitätsanforderungen 7

9 Ergänzungen 8

10 Glossar 8

Lastenheft S/Chat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Version* | *Autor* | *QS* | *Datum* | *Status* | *Kommentar* |
| 0.9 | Frantar |  | 16.09.2013 | Draft |  |
| 1.0 |  | Ye | 16.09.2013 | Überarbeitet |  |

# Einführung

Mit Aufdeckung des NSA-Skandals wird das Thema Datensicherheit zu einem essenziellen Bestandteil der Softwareentwicklung. Aktuell existieren nur sehr wenige Programme, die eine sichere (es werden bewährte Standards verwendet und die Daten werden nur auf Endgeräten entschlüsselt) Kommunikation ermöglichen. Von diesen wenigen sind wiederrum nur sehr wenige benutzerfreundlich und werden daher kaum von Privatpersonen eingesetzt.

Deshalb wollen wir in unserem Projekt eine benutzerfreundliche App entwickeln, die es den Usern ermöglicht, sicher Nachrichten zu versenden, ohne jegliches Grundwissen über Verschlüsselungen besitzen zu müssen.

# Zielbestimmung

Es soll eine Android App entwickelt werden, die es Usern ermöglicht sicher (so verschlüsselt, dass ein unfreiwilliges Entschlüsseln mit höheren Kosten verbunden ist als, sämtliche mit dieser Applikation versendete Daten wert sind) miteinander über das Internet zu kommunizieren.

# Produkteinsatz

Es soll den Usern ermöglicht werden im Alltag sicher über das Internet zu kommunizieren, ohne dass die Nachrichten von Externen unbefugt mitgelesen werden können.

# Produktfunktionen

## Verschlüsselung

Sämtliche Nachrichten sollen verschlüsselt versendet und abgespeichert werden. Zusätzlich werden sie noch digital signiert. Dazu müssen zahlreiche verschiedene Schlüssel („Sessionkeys“) generiert werden.

/**LF010**/ **Private / Public Key-pair generieren**

Beim erstmaligen Starten der Applikation wird ein zufälliges Schlüsselpaar generiert. Der öffentliche Teil des Schlüsselpaares wird mit Kommunikationspartnern ausgetauscht. Der private Teil wird zur Entschlüsselung sämtlicher Nachrichten verwendet.

/**LF020**/ **Sessionkeys generieren**

Zum Verschlüsseln jeder Nachricht muss ein eigener symmetrischer Sessionkey zufällig generiert werden.

/**LF030**/ **Nachrichten signieren**

Jede zu versendende Nachricht muss digital signiert werden, damit der Empfänger ihren Absender eindeutig feststellen kann.

## Userfunktionen

Der User kann folgende Funktionen der App benutzen.

/**LF110**/ **Neuen Schlüssel erstellen**

Ein User kann auf Wunsch ein neues Schlüsselpaar generieren lassen. Es ist zu beachten, dass alle alten Konversationen nach Benutzung dieser Funktion erst nach erneutem Schlüsselaustausch wieder fortgeführt werden können.

/**LF120**/ **Nachrichten versenden**

Ein User kann eine Nachricht verschlüsselt an einen Kontakt verschicken. Eine Nachricht besitzt eine maximale Länge.

/**LF130**/ **Nachrichten anzeigen**

Beim Öffnen eines Chats mit einem anderen User werden alle alten abgespeicherten Nachrichten automatisch geladen und angezeigt.

/**LF130**/ **Nachrichten löschen**

Ein User kann eine seiner gespeicherten Nachrichten löschen. Die Nachricht wird natürlich nur auf seinem Gerät gelöscht nicht aber auf dem des Kommunikationspartners.

/**LF140**/ **Kontakte hinzufügen**

Ein User kann einen neuen Kontakt durch Eingabe dessen Publickeys seiner Kontaktliste hinzufügen.

/**LF140**/ **Kontakte löschen**

Ein User kann einen Kontakt aus seiner Kontaktliste löschen. Dadurch werden auch sämtliche mit diesem Kontakt ausgetauschte Nachrichten gelöscht.

# Produktdaten

## Private Daten

Folgende Schlüsseldaten werden abgespeichert und von der Applikation verwendet. Sie können aber nicht vom User ausgelesen (mithilfe dieser App) werden.

/**LD010**/ folgende Schlüssel sollen abgespeichert werden:

* Privater Schlüssel
* Masterschlüssel zur Schlüsselgenerierung

## Kontaktdaten

Ein Kontakt wird als Name und dazugehöriger Public-Key abgespeichert.

/**LD110**/ folgende Daten werden über Kontakte abgespeichert:

* Name
* Public Key

## Nachrichten

Es sollen alle empfangenen und gesendeten Daten verschlüsselt abgespeichert werden.

/**LD210**/ folgende Daten sollen zu den Nachrichten abgespeichert werden:

* Sender
* Sessionkey
* Nachrichteninhalt
* Uhrzeit
* Digitale Signatur des Senders

# Zwingende Randbedingungen

## Produktumgebung und Systemintegration

## Hardware Entwicklung: ADT fähiges Device (Mac, Linux, Windows)

## Software Entwicklung: Android Development Tool

## Hardware Laufumgebung: Android fähiges Device

## Software Laufumgebung: Android 2.3 aufwärts

## 6.2 Schnittstellen

* In diesem Projekt sollten nur Userschnittstellen vorhanden sein

# Vertragsgegenstand

## Lieferumfang

* App
* Vollständiger Sourcecode

## Produktleistungen

**Benutzerfreundlichkeit:**

/**LL010**/ Der User wird nicht direkt mit dem Verschlüsselungsprozess konfrontiert

/**LL020**/ Der Publickey wird dem User in einer weitergebbaren Form angezeigt (Bsp. a-z A-Z 0-9 Zeichenbereich)

**Nachrichten:**

/**LL110**/ Nachrichten sollen sicher und zuverlässig verschickt werden können

**Sicherheit:**

/**LL210**/ Zur Verschlüsselung sollen internationale Standards implementiert werden

/**LL220**/Zur geheimen Kommunikation sollen geeignete Protokolle ausgewählt werden

**Internet:**

/**LL310**/ Die Internetgeschwindigkeit des Users soll bei laufender Applikation nicht beeinträchtigt werden

/**LL320**/ Das erstmalige Generieren des Schlüsselpaares sollte nicht länger als 10 Sekunden dauern

/**LL330**/ Das Laden der gespeicherten Nachrichten sollte nicht länger als 5 Sekunden dauern

## Produktbezogene Leistungen

* Zusätzlich zum Sourcecode werden auch noch die Servereinstellungen offengelegt
* Die Applikation sollte auf Geräten ab Android-Version 2.3 funktionieren

# Qualitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Produktqualität** | **sehr gut** | **Gut** | **normal** | **Irrelevant** |
| **Funktionalität** |  | **x** |  |  |
| **Zuverlässigkeit** | **X** |  |  |  |
| **Benutzbarkeit** | **X** |  |  |  |
| **Effizienz** |  |  | **x** |  |
| **Änderbarkeit** |  |  | **x** |  |
| **Übertragbarkeit** |  |  |  | **x** |
| **Datensicherheit** | **X** |  |  |  |

**Funktionalität:**

Die Applikation soll eine überschaubare Menge an Funktionen bereitstellen.

**Zuverlässigkeit:**

Das Versenden von Nachrichten soll sehr zuverlässig (bei guter Internetanbindung) funktionieren.

**Benutzbarkeit:**

Die Applikation muss auch von nicht-Experten im Bereich Verschlüsselung ohne Probleme einfach bedient werden können.

**Effizienz:**

Die Effizienz ist bei diesem Projekt nicht sehr von Bedeutung, da die Leistung modernerer Smartphone sämtliche Verschlüsselungsalgorithmen ohne Performanceprobleme berechnet werden.

**Änderbarkeit:**Änderbarkeit steht bei diesem Projekt nicht im Vordergrund. Es müssen aber trotzdem Daten geändert werden können.

**Übertragbarkeit:**Die Übertragbarkeit ist irrelevant, da dieses Projekt nur auf dem Betriebssystem Android laufen soll.

**Datensicherheit:**

Die Datensicherheit ist der wichtigste Punkt dieser Applikation.

# Ergänzungen

**Verschlüsselungsstandards:**

* Es sollen bevorzugt bewehrte Verschlüsselungsstandards und Protokolle verwendet werden
* Der Sourcecode soll nach Fertigstellung des Projekts offengelegt werden.

# Glossar

* **Asymmetrische Verschlüsselung:** Ver- und Entschlüsselungen geschehen durch unterschiedliche Schlüssel (bei symmetrischen Verfahren durch denselben Schlüssel)
* **Public/Private Key-pair:** Ein asymmetrisches Verschlüsselungsverfahren benötigt immer einen öffentlichen Schlüssel zur Verschlüsselung und einen privaten Schlüssel zur Entschlüsselung von Nachrichten. Der öffentliche Schlüssel ist allen Sendern bekannt, der private aber nur dem Empfänger.
* **Session Key:** Jede Nachricht wird aus Performancegründen nur mit einem symmetrischen Sessionkey verschlüsselt, der anschließend asymmetrisch verschlüsselt wird.
* **Digitale Signatur:** Mithilfe von asymmetrischen Verschlüsselungsverfahren können Nachrichten digital signiert werden