# Datenübertragung

## Schritte der Kommunikation

* Übergabe des Briefes mit Link auf App und QR-Code
* Scannen des QR-Codes
* Extraktion des Public Keys aus dem QR-Code und Überprüfung der Signatur
* Ermittlung des Dateinamens aus dem QR-Code
* Herunterladen der Datei
* Entschlüsseln des Session Key, der in der Datei enthalten ist
* Entschlüsseln weiterer Kryptoparameter
* Entschlüsseln des Dokumentes mit den Daten des Trainingsteilnehmers
* Speichern der Daten auf dem Smartphone
* Löschen der Datei auf dem Server

## Arbeitsablauf

* Es gibt einen zentralen Secret Key und dazugehörigen Public Key.
* Für jeden anzulegenden Trainingsteilnehmer wird ein RSA-Key-Pair angelegt. Dieses Keypair dient der Verschlüsselung des Session Keys, mit dem wiederum das eigentliche Dokument verschlüsselt wird.
* Die Trainingsdaten werden in einem Formular eingegeben.
* Es wird ein Session Key generiert, ein Initial Value (iv) als weiterer Crypto-Parameter und diese Daten werden mittels des persönlichen Public Key verschlüsselt.
* Die Trainingsdaten werden mit dem Session Key unter Verwendung einer symmetrischen Verschlüsselung (derzeit AES-512) verschlüsselt.
* Die verschlüsselten Cryptoparameter und die verschlüsselten Trainingsdaten werden in einer Datei zusammengefasst. Da alle Daten mit Crypto-Verfahren auf dem Stand der Technik verschlüsselt werden, ist eine Speicherung im Internet als Voraussetzung zur Nutzung mit der Smartphone-App möglich.
* Es wird ein Brief an den Trainingsteilnehmer generiert, in dem der persönliche Secret Key, der Dateiname und die Signatur dieser Daten (mittels zentralem Secret Key) in einem QR-Code enthalten sind.
* Der persönliche Secret Key wird per Brief und QR-Code an den Benutzer geschickt. Im QR-Code befindet sich neben dem Secret Key der Dateiname und die Signatur, die mit dem zentralen Secret Key erstellt wird. Der zentrale Public Key ist in der App enthalten, womit die Signatur und damit die Gültigkeit des persönlichen Secret Key geprüft wird.
* Der QR-Code des Briefes wird mit der App eingescannt. Es wird die Gültigkeit der Signatur (mittels zentralem Public Key) und damit die Korrektheit der Daten geprüft.
* Sind die Daten gültig, wird die Datei von der App heruntergeladen.
* Aus der Datei werden die verschlüsselten Cryptoparameter extrahiert und mittels des privaten Secret Key entschlüsselt.
* Die Trainingsdaten werden aus der Datei extrahiert und mittels der Cryptoparameter entschlüsselt.
* Anschließend werden die Trainingsdaten in der App gespeichert.
* Nach erfolgreicher Speicherung wird die Datei auf dem Server gelöscht, so dass ein Download der Daten genau einmal möglich ist.
* Der private Public und Secret Key bleiben gespeichert, so dass in einer späteren Version ein versenden weiterer Nachrichten auf dem beschriebenen Weg möglich ist.

## Sicherheitsaspekte

Grundsätzlich ist bei der Übertragung von Daten über das Internet die Sicherheit der übertragenen Daten von hoher Priorität.

Die Sicherheit der Daten umfasst dabei die Vertraulichkeit der Daten (nur befugte Nutzer können die Daten lesen), die Korrektheit der Daten (niemand kann diese Daten unbemerkt verändern) und die Zuverlässigkeit des Absenders (es muss garantiert sein, dass niemand sich fälschlicherweise als Absender ausgeben kann).

Die erste Forderung (Vertraulichkeit) kann durch Verschlüsselung der Daten mittels aktueller Verschlüsselungstechniken gesichert sein. In der Anwendung kommt eine AES-512-Verschlüsselung zur Anwendung, die nach derzeitigem Stand der Technik eine ausreichende Sicherheit bietet. Auch gegen die immer leistungsfähigeren Quantencomputer ist diese Verschlüsselung ausreichend robust.

Die Verschlüsselung wird mittels der OpenSSL-Bibliothek durchgeführt. Diese Bibliothek wird in großem Umfang eingesetzt, so dass kritische Sicherheitslecks sehr schnell erkannt, veröffentlicht und behoben werden. Dieser Ansatz ist einer eigenen Implementierung in jedem Fall vorzuziehen, da die Gefahr von Fehlern in der Implementierung (die einzige bekannte Ursache für erfolgreiche Angriffe auf AES-Verschlüsselungen) sehr hoch ist.

Es wird für jede Übertragung ein neues Paket von Cryptoparametern erzeugt. Diese Parameter werden mittels eines Public Key-Verfahrens (hier derzeit RSA) verschlüsselt. Eine Verschlüsselung der gesamten Daten mittels eines Public Key-Verfahren ist aus Gründen der Performance nicht empfehlenswert. Dies entspricht auch der gängigen Arbeitsweise bei der Übertragung von Daten im Internet.

Für die RSA-Verschlüsselung wird der persönliche Public Key verwendet. Die verschlüsselten Daten können somit nur mittels des persönlichen Secret Key entschlüsselt werden, welcher mittels des QR-Codes im Anschreiben übertragen wird.

Die verwendete RSA-Verschlüsselung ist derzeit als sicher anzusehen. Es ist jedoch absehbar, dass mit der wachsenden Leistungsfähigkeit von Quantencomputern die Sicherheit dieses Verfahrens, welches auf Primfaktoren-Zerlegung beruht, nicht mehr gewährleistet ist. Es wird derzeit an Verfahren gearbeitet, die auch gegenüber Quantenalgorithmen ausreichend robust sind. Sobald das RSA-Verfahren angreifbar ist, muss es durch ein Update der Anwendung ersetzt werden.

Das RSA-Verfahren wird ebenfalls durch die OpenSSL-Bibliothek implementiert. Es gelten hierfür die gleichen Überlegungen wie oben zum AES-Verfahren dargelegt.

Die Sicherheit des bis hierher beschriebenen Ablaufs hängt im wesentlichen an der sicheren Übertragung des privaten Secret Key. Das private Keypair für die RSA-Verschlüsselung wird während des Anlegens des Trainingsteilnehmers generiert.

Der Secret Key wird gemeinsam mit dem Namen der Datei mit den eigentlichen Daten in einer JSON-Datei abgelegt. Über diese Daten wird ein Hash gebildet (SHA-512) und dieser Hash wird mittels des zentralen Secret Key verschlüsselt (Signatur).

Diese Daten (privater Secret Key, Dateiname und Signatur) werden in einem QR-Code codiert und auf dem Begrüßungsanschreiben ausgegeben.

Der Trainingsteilnehmer erhält dieses Schreiben und scannt den QR-Code mittels der App ein. Es wird ein Hash über den enthaltenen privaten Secret Key und den Dateinamen gebildet. Anschließend wird die Signatur mit dem zentralen Public Key entschlüsselt. Die entschlüsselte Signatur und der gebildete Hash werden verglichen. Bei Gleichheit ist sichergestellt, dass diese Daten vom korrekten Absender erzeugt und verschickt wurden. Die Korrektheit des Dateinamens und des privaten Secret Key ist gewährleistet.

Im Anschluss wird die Datei auf das Smartphone heruntergeladen. Die Cryptoparameter für die AES-Verschlüsselungen werden mittels des privaten Secret Key entschlüsselt, dessen Korrektheit im vorherigen Schritt überprüft und bestätigt wurde.

Im Anschluss können die eigentlichen Daten mittels der Cryptoparameter entschlüsselt und gespeichert werden.

Im Anschluss wird die Datei auf dem Server gelöscht.

Zwei Sicherheitsrisiken bestehen bei dieser Vorgehensweise. Der Hauptschwachpunkt ist die Übermittlung des QR-Codes. Hier kann der Weg verwendet werden, auf dem bisher die entsprechenden Daten verschickt werden. Der QR-Code und die darin enthaltenen Daten sind auf demselben Niveau gefährdet wie die Originaldaten, so dass hierfür auch kein höherer Schutz als für die Originaldaten notwendig ist.

Weiterhin hängt die Gesamtsicherheit von der Korrektheit des zentralen Public Key ab. Dieser ist fest in der App codiert, so dass eine Fälschung einen Angriff auf den compilierten Code der App erforderlich macht. Dies ist durch die Sicherheitsmechanismen des Smartphone-Betriebssystems ausgeschlossen (bisher Android). Das Prinzip der festen Codierung eines Public Key wird in ähnlicher Weise seit Jahrzehnten von den Browser-Herstellern verwendet. Dort werden zentrale Zertifikate fest im Browser kodiert, so dass damit die Verschlüsselungsparameter z.B. bei der Übertragung per https überprüft werden können. Eine Aktualisierung des zentralen Public Key muss regelmäßig im Rahmen von Updates der Anwendung vorgenommen werden.