May 2020

One time sharing of data between apps on mobile devices

Armijn Hemel

One time sharing of data between apps on mobile devices

Abstract

Sometimes there are situations where data needs to be shared between mobile devices, but where the users of the mobile phones do not want to share phone numbers, e-mail addresses or transfer data via Bluetooth or NFC.

This article explores a solution how with an extra indirection data could be shared in a relatively safe way.

Keywords

mobile, security, sharing, secure sharing, qr code, data sharing

Background

There are situations where data needs to be shared between mobile devices of two persons, but where there is no desire to share any other information, such as phone numbers, e-mail addresses, or to connect phones using Bluetooth or NFC (examples: Android Beam, Airdrop, etc.).

One hypothetical use case would be an app running on a mobile phone where a user can store information about a collection of objects, such as coins, stamps, baseball cards, and so on, and where spare copies can be recorded in a trade list and items wanted recorded in a want list. The two users meet at a trade show or fair and want to make a trade, but not necessarily connect with eachother. To compare the trade list of one user with the want list of another user this data needs to be shared from the device of the first user to the device of the second user, so it can be processed by the application on the second device.

This method requires that the sending device has acesss to an online server that allows public sharing of files via some link, such as Nextcloud instances, WeTransfer, and so on.

Steps

1. a data export is made in the app on the sending device
2. the app on the sending device uploads the data to a server online that allows public file sharing
3. the URL of the file sharing URL is shown as a QR code or other kind of barcode on the sending device
4. the receiving device opens the app that is supposed to receive the data and scans the QR code or other kind of barcode on the sending device using the camera

5. the receiving device receives the file from the server and the app further processes it
6. (optionally) the file is removed from the file sharing service

It might be that to avoid the data being snooped while transferred it needs to be encrypted. This can be done using symmetrical encryption or assymetrical (public/private key).

For symmetrical encryption the process could look as follows:

1. a temporary password is generated on the sending device
2. a data export is made in the app on the sending device, symmetrically encrypted with the generated password
3. the app on the sending device uploads the data to a server online that allows public file sharing
4. the URL of the file sharing URL is shown as a QR code or other kind of barcode on the sending device
5. the receiving device opens the app that is supposed to receive the data and scans the QR code or other kind of barcode on the sending device using the camera
6. the receiving device receives the file from the server
7. the password used to encrypt the archive on the sending device is encoded in a QR code, or other kind of barcode and shown on the sending device
8. the app on the receiving device scans the QR code or other kind of barcode containing the password
9. the app decrypts the data using the password and further processes the data
10. (optionally) the file is removed from the file sharing service

For assymetrical encryption it could look like this:

1. a temporary public/private key pair is generated on the sending device
2. a temporary public/private key pais is generated on the receiving device
3. the public key on the receiving device is encoded in a QR code, or other kind of barcode and shown on the receiving device
4. the app on the sending device scans the QR code or other kind of barcode of the previous step and stores the public key of the receiving device
5. a data export is made in the app on the mobile phone, asymmetrically encrypted with the public key of the receiving device and the private key of the sending device
6. the app uploads the data to a server online that allows public file sharing

7. the URL of the file sharing URL is shown as a QR code or other kind of barcode on the sending device
8. the receiving device opens the app that is supposed to receive the data and scans the QR code or other kind of barcode on the sending device using the camera
9. the receiving device receives the file from the server
10. the public key on the sending device is encoded in a QR code, or other kind of barcode and shown on the sending device
11. the app on the receiving device scans the QR code or other kind of barcode of the previous step and stores the public key of the sending device
12. the app decrypts the data using the public key of the sending device and the private key of the receiving device and further processes the data
13. (optionally) the file is removed from the file sharing service