

NFC Key Cards und Transponder (FabCard / FabFire)

Kartenauthentifizierung mit NXP/MIFARE DESFire-Karten. Für die Hardware zum Auslesen dieser Karten siehe auch [Smartcard Reader](#).

- [Funktionsprinzip / Grundlagen](#)
- [FabFire Tools](#)
- [Vorlagen für Karten-Designs](#)

Funktionsprinzip / Grundlagen

Intro

NFC stellt eine Untermenge der RFID-Technologie dar und kann z.B. als benutzbares Endnutzer-Medium in Form Karten- oder Transpondern gekauft werden. Dabei gibt es dutzende verschiedene Typen. Diese unterscheiden sich in Preis, Sicherheit, verwendeter Technologie, Verschlüsselungsverfahren und so weiter. Wir verwenden für FabAccess ausschließlich die MIFARE DESFire Karten von NXP. Diese sind kostengünstig, nicht durch Dritte klonbar.

Ziel der NFC-Verwendung ist zum Beispiel die Verwendung von FabReader v3. FabReader ist ein SmartCard Lesegerät, welches an eine einzelne Maschine (z.B. Drehbank) oder an eine zentrale Stelle montiert werden kann, um damit eine oder mehrere Maschinen (per Keypad) zu bedienen. Durch die Verwendung eines solchen Geräts ersparen wir Nutzern die Verwendung des Smartphones oder Tablets. Das ist besonders in Maschinenbereichen hilfreich, wo z.B. dauernder Schmutz (Fett, Öl) dafür sorgt, dass das private Endgerät unhygienisch verschmutzt wird.

Natürlich kann aber auch jedes Smartphone, Tablet oder PC für FabAccess verwendet werden, wenn sowohl NFC unterstützt, als auch die Borepin App hat.

Basis - Authentifizierung mit NXP MIFARE DESFire Tags

Diese Tags sind in der Lage, den Zugang zu den Daten auf den Karten durch symmetrische Verschlüsselung und die Verwendung eines Diffie-Hellman-Schlüssels zu beschränken, um das Abhören durch eine weiterleitende Partei zu verhindern. Eine Karte hat mehrere „Anwendungen“, die bis zu 32 Dateien enthalten. Eine Datei kann gelesen oder geschrieben werden. Beide Arten des Zugriffs können auf Parteien beschränkt werden, die einen PSK kennen, und zwar auf einer file-to-file Basis.

Grundsätzlich schreiben wir auf den Tag nur die ID des Nutzers. Den Rest regelt das Backend (Datenbank)

Dateien

Das derzeitige System verwendet die Dateien 0001 bis 0003:

Datei 0001 erlaubt öffentlichen (d.h. nicht authentifizierten) den Lesezugriff und enthält die Zeichenketten `FABACCESS`, `DESFIRE` und `1.0` als gepackte Liste von UTF-8 kodierten nullterminierten Zeichenketten.

Beispiel:

- `FABACCESS\0DESFIRE\01.0\0`

Diese Datei dient als Kennung, anhand derer ein Server überprüfen kann, ob er diese Karte verwenden darf.

Datei 0002 erlaubt den öffentlichen Lesezugriff und enthält: Einen URL-codierten Namen des ausstellenden Spaces als URN im Format `urn:fabaccess:lab:<labname>`

Beispiele:

- `urn:fabaccess:lab:innovisionlab`
- `urn:fabaccess:lab:Bibliothek%20Neustadt%20Makerspace`
- `urn:fabaccess:lab:Offene%20Werkstatt%20M%C3%A4rz`

Ein gültiger IRI, der auf die bffh-Instanz verweist, die für dieses Labor läuft. Diese bffh **sollte** vom Internet aus erreichbar sein. Die Verwendung von IP-Adressen zur privaten Nutzung oder von IRIs, die auf solche Adressen verweisen, kann für Spaces hinter restriktiven Firewalls oder aufgrund lokaler Richtlinien erforderlich sein. Der IRI **muss** das „fabaccess“-Schema verwenden und **darf keinen** userinfo-, path-, query- oder fragment-Teil enthalten. Beispiele:

- `fabaccess://innovisionlab.de/`
- `fabaccess://192.168.178.65`
- `fabaccess://fabaccess-server.localnet`

Eine mit Null terminierte Liste von UTF-8 kodierten IRIs mit Kontaktoptionen, um den Kartenaussteller oder -besitzer zu benachrichtigen, falls die Karte verloren gegangen ist. Emittenten **sollten** einen Wert bei der Kartenerstellung festlegen und **können** den Karteninhabern erlauben, Werte ihrer Wahl zu ändern oder hinzuzufügen. Beispiele:

- `mailto:lostcard@innovisionlab.de`
- `https://innovisionlab.de/lostcard`
- `https://werkstatt-märz.de/cardlost.php?action=submitcardlost`

Datei 0003 ermöglicht nach Wahl des Ausstellers den öffentlichen Zugang oder den Zugang mit einem Schlüssel. Sie enthält ein Token, das vom Heimatserver des Karteninhabers verwendet werden kann, um den Karteninhaber zu identifizieren. Das Format des Tokens **darf von keiner anderen Partei** als dem Heimatserver verwendet werden.

FabFire Spezifikation

FabAccess nutzt eine eigene Spezifikation namens FabFire, um die Kommunikation mit DESFire Karten zu ermöglichen.

Application Identifier (AID): `0x464142`

Weitere Details zur konkreten Nutzung von FabFire finden sich unter [FabFire Tools](#).

NFC Implementierungen

- C# - <https://gitlab.com/fabinfra/fabaccess/nfc>
- Rust - https://gitlab.com/fabinfra/fabaccess/nfc_rs

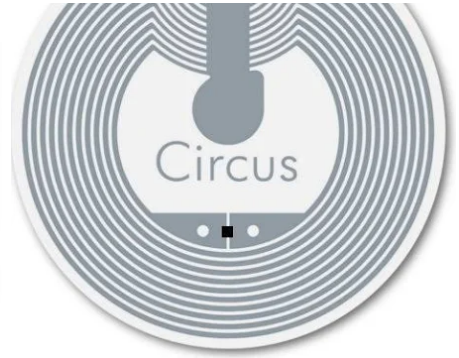
Hinweise zu MIFARE DESFire von NXP

Unsere Client-App Borepin kann neben [QR-Codes](#) auch sogenannte Mifare DESFire Tags von NXP scannen. Hierzu muss die NFC-Funktion des Telefons/Tablets aktiviert sein, auf dem die App läuft. Beim Auflegen des Tags wird die entsprechende Aktion getriggert. Folgendes Datenformat wird auf dem NFC Tag benötigt:

`fabaccess://fabaccess.local/resource/{machine id}`

Beispiel für Tags in Form von Transpondern, Smartcards und Klebetags. Den einen oder anderen dürfte jeder schon mal gesehen haben. Wir empfehlen, die Tags (Karten, Transponder, Kleber) auch optisch indentifizierbar zu machen, z.B. mit dem [FabFire Logo](#).

Kompatibel mit FabFire sind nur MIFARE DESFire EV2 Karten!



FabFire Tools

- FabFire Provisioning Tool: Tool zur Bereitstellung neuer Karten für das FabAccess-Kartensystem. **Unterstützt werden nur DESFire EV2 Karten!**
 - <https://gitlab.com/fabinfra/fabaccess/FabFire-Provisioning-Tool>
- FabFire Adapter: übersetzt MQTT Nachrichten von der Reader-Hardware in die API
 - https://gitlab.com/fabinfra/fabaccess/fabfire_adapter

Vorlagen für Karten-Designs

Wer seine Smartcards selbst branden will, der findet das Design im Chipkartenformat in <https://gitlab.com/fabinfra/design>