FabFire und NFC Tags

Benutzer-Kartenauthentifizierung mit NXP/MIFARE DESFire-Karten. Für die Hardware zum Auslesen und Beschreiben dieser Karten siehe auch Smartcard Reader.

Und: Ressourcen scannen mit NFC Tag.

- FabFire Funktionsprinzip / Grundlagen
- Vorlagen für FabReader / FabFire / FabFireCard
- Ressourcen per NTag scannen

FabFire 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 EV2 Karten von NXP. Diese sind kostengünstig und 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 Ressource (z.B. Drehbank) oder an eine zentrale Stelle montiert werden kann, um damit eine oder mehrere Ressourcen (per Keypad) zu bedienen. Durch die Verwendung eines solchen Geräts ersparen wir Nutzern die Verwendung des Smartphones oder Tablets. Das ist besonders in Bereichen hilfreich, wo z.B. dauernder Schmutz (Fett, Öl) dafür sorgt, dass das private Endgerät unhygienisch verschmutzt wird.

Naturlich 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 0004:

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 (Magic Identifier).

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 diesen Space läuft. Diese BFFH-Instanz **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.

Datei 0004

Beschränkt den Lesezugriff auf einen einzigen Schlüssel, der dem Heimatserver des Karteninhabers bekannt ist. Sie ist leer, aber durch die Zugriffsbeschränkung kann der Heimatserver die Karte als echt validieren und damit die Authentifizierung des Benutzers abschließen.

FabFire Spezifikation

FabAccess nutzt eine eigene Spezifikation namens FabFire, um die Kommunikation mit DESFire Karten zu ermöglichen. FabFire nutzt dafür SASL Callbacks für die Authentifizierung (rsasl Bibliothek).

FabFire gibt es in zwei Varianten:

- `X-FABFIRE`
- `X-FABFIRE-BIN`

Der von FabFire verwendete Application Identifier (AID) lautet standardmäßig `0x464142`. Er kann jedoch vom Administrator neu vergeben werden.

Die Verschlüsselung der Keys:

- PICC Masterkey: TDES (oder auch als Triple DES oder 3DES bekannt)
- App Masterkey: AES
- Auth Masterkey: AES

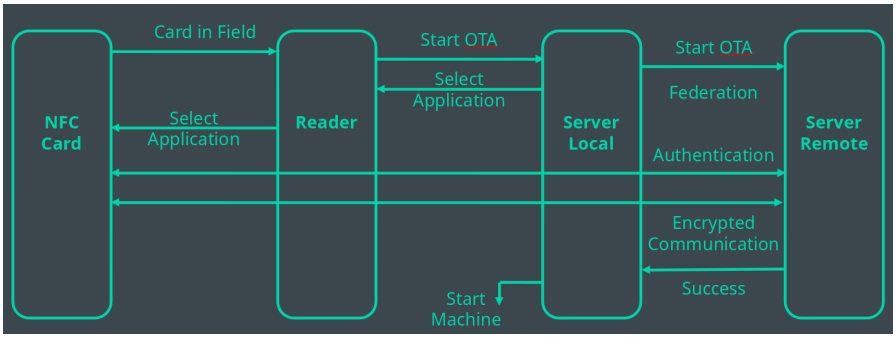
Weitere Details zur konkreten Nutzung von FabFire finden sich unter FabFire Tools.

NFC Implementierungen

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

Kommunikationsschema

Das folgende Schema zeigt grafisch die Kommunikation zwischen NFC Card und unserem Server via FabReader. Alles funktioniert über das OTA-Prinzip (Over the Air).



Hinweise zur Kompabilität mit FabFire

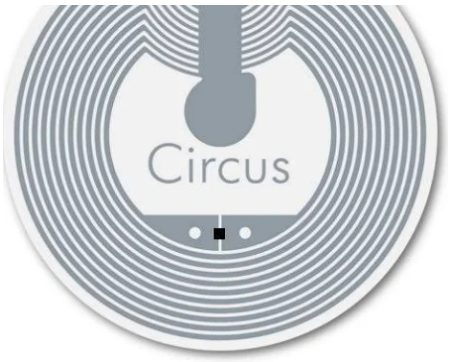
Kompatibel mit FabFire sind nur MIFARE DESFire EV2 Karten!

> Inkompatibel sind z.B. Mifare Ultralight C, Mifare Classic und Mifare NTAG.

In welcher Gestalt können DESFire Tags auftauchen?

Beispiele 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.


NFC


Circus

Vorlagen für FabReader / FabFire / FabFireCard

FabReader und FabFire

Diese Logos dienen dazu NFC-Reader und NFC-Tags zu erkennen und sollte daher auch nur im FabFire Kontext verwendet werden. Das Logo sollte auf der NFC-Reader Fläche oder auf einem Sticker über einem NFC-Tag angebracht werden. Die Icons können je nach Geschmack verwendet werden.

FabFire Logo

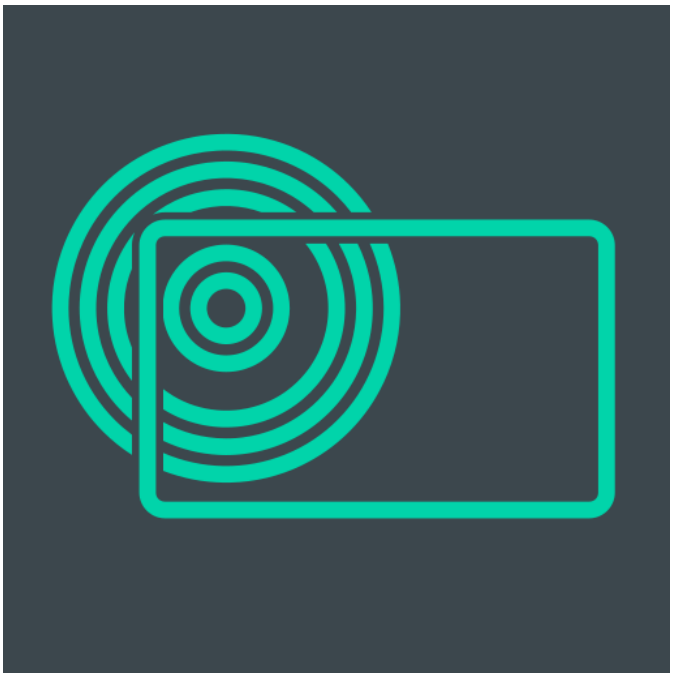
FabReader Logo

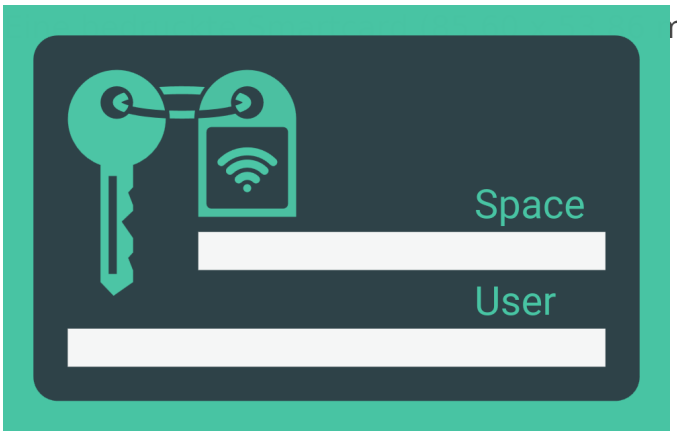
Allgemeines NFC Reader Logo

Falls für das NFC-Projekt kein FabReader, sondern eine andere Leseeinheit verwendet wird, kann auch ein generisches Symbol verwendet werden:

FabFireCards

Das allgemeine FabFireCard Icon sieht wie folgt aus. Es findet aber eher selten praktische Anwendung.





Wer seine Smartcards selbst branden will, der findet fertige Designs in GitLab zum Download. Das Bedrucken ist zum Beispiel mit folgenden Geräten möglich:

- Zebra ZXP 3 - dafür gibt es eine farblich angepasste Vorlage

> Du möchtest Karten mit Daten beschreiben? Das findet sich unter FabFire Tools

Ressourcen per NTag scannen

Allgemeines

Unsere App Borepin erlaubt das Scannen von Ressourcen nicht nur per QR-Code Sticker, sondern auch per NFC Tag. So kürzen wir in der App ab und müssen uns nicht erst durch die Ressourcenliste hangeln.

Wie benutzen?

Hierzu muss die NFC-Funktion des Telefons/Tablets aktiviert sein, auf dem die App läuft. Beim Auflegen des Tags wird die entsprechende Aktion getriggert.

Datenformat

Folgendes Datenformat wird auf dem NFC Tag benötigt:
`fabaccess://fabaccess.local/resource/{machine id}`

NFC Tags können recht einfach mit einer Smartphone App beschrieben werden, zum Beispiel

- auf Android: NFC Tools

> Im Code verantwortlich ist dieser Abschnitt

Wie sehen NFC Tags aus?

Letztendlich nehmen NFC Tags verschiedene Formen wahr, z.B. als Transponder oder Smartcard - quasi genauso wie unsere FabFireCards