Carte étudiante européenne

Ces pages de documentation doivent vous permettre d'intégrer votre ESUP-SGC dans le projet ESC (European Student Card).

Ces pages sont avant tout techniques ; pour une bonne compréhension du projet, merci de vous référer au site officiel : https://europeanstudentcard.eu

Une présentation (slides) datant de juin 2021 est également disponible en complément de cette page wiki : ESUP-SGC, son intégration dans le projet Carte Étudiante Européenne [CNCEU - Juin 2021]

Le projet de carte étudiante européenne comporte plusieurs volets techniques que nous allons aborder un par un :

Impression et utilisation du QR-Code

Présentation

Pour fonctionner, ESUP-SGC imprime puis encode la carte en 2 étapes successives et distinctes.

Pour réaliser l'enrôlement de la carte au moment de l'encodage après impression, un qr-code est utilisé.

Ce qr-code correspond au qr-code standardisé/normalisé par le projet ESC.

C'est là le premier usage que fait ESUP-SGC du QR-Code de la carte européenne.

Le QR-Code européen consiste en une url dans laquelle on retrouve l'url d'accès au serveur ESC http://esc.gg/ suivi d'un numéro de carte 'ESCN' qui est un identifiant codé en hexadécimal, formaté pour une meilleur lecture avec des - supplémentaires.

Exemple : d88b02c1-894e-1038-a711-001999465982

Cet ESCN est construit pour être unique, ESUP-SGC utilise directement la librairie escn-generator disponible depuis les repositories centraux de maven (le code source y est également donné) : https://search.maven.org/artifact/eu.europeanstudentcard/escn-generator

Cet identifiant de carte est construit via l'heure système suffixé d'un code qui correspond au pic de l'établissement lui-même préfixé par un numéro supplémentaire.

Dans l'exemple de d88b02c1-894e-1038-a711-001999465982 on retrouve ainsi le code pic 999465982 de l'Université de Rouen Normandie.

On note que la "version" hexadecimal correspond à d88b02c1894e1038a711001999465982, soit une chaîne hexadecimale de 16 octets.

ESUP-SGC ne propose pas cependant d'utiliser le QR-Code pour identifier la carte dans un usage courant après édition.

ESUP-SGC préfère en effet l'usage du sans-contact au scan de QR-Code.

Mise en oeuvre

La configuration de la génération de ce QR-Code se fait dans src/main/resources/META-INF/spring/applicationContext-crous.xml

```
<bean id="escUidFactoryService" class="org.esupportail.sgc.services.esc.EscUidFactoryService">
<property name="pic" value="le-code-pic-de-letablissement"/>
<property name="prefixe" value="1 " /> <!-- si plusieurs sgc, l'établissement doit les distinguer par un
préfixe différent -->
<property name="qrCodeUrlPrefixe" value="http://esc.gg/" /> <!-- pour l'instance ESC de pre-production on
mettra http://pp.esc.gg/ -->
</bean>
```

On active l'utilisation du QR-Code européen (en lieu de l'usage de l'EPPN en tant que QR-Code) en allant dans l'interface web, "Admin" > "Configurations" > "QRCODE_ESC_ENABLED" que l'on doit mettre à true.

Enregistrement d'une carte dans ESC-R

Présentation

Pour que la carte puisse être reconnue par un service d'un établissement européen partenaire, il faut que celui-ci puisse la récupérer techniquement depuis ESCR qui est la plateforme d'échange (le hub ou Router) des identifiants de cartes/étudiants du projet ESC.

Enregistrer une carte c'est donc enregistrer à la fois l'étudiant et la carte dans ESCR.

ESUP-SGC permet à ce que ce soit sur demande de l'étudiant que l'enregistrement ait lieu (RGPD). Ensuite, ESCR elle-même propose à l'étudiant de diffuser au cas par cas ses identifiants aux services des établissements partenaires.

L'identifiant de carte est l'ESCN.

Confère cette documentation Renater, l'identifiant de l'utilisateur ESI (European Student Identifier) pour les établissements français est construit ainsi : urn:schac:personalUniqueCode:int:esi:fr:<INE>

ESUP-SGC reconstruit simplement cet identifiant depuis l'INE : la connaissance de l'INE est donc obligatoire ici.

Mise en oeuvre

La configuration d'esup-sgc en tant que client ESCR pour enregistrer les cartes se fait dans src/main/resources/META-INF/spring/applicationContext-crous.xml

```
        <bean id="europeanStudentCardService" class="org.esupportail.sgc.services.esc.ApiEscrService">
                <property name="enable" value="true"/>
                <property name="webUrl" value="https://api.europeanstudentcard.eu/v1" />
                <property name="key" value="clef-a-recuperer-auprès-d-escr" />
                <property name="restTemplate" ref="restTemplate" />
                <property name="countryCode" value="FR"/>
                <property name="picInstitutionCode" value="le-code-pic-de-letablissement"/>
                <!--
                Type of cards. Possibles values are :
                1 - passive card, with no electronic
                2 - Smartcard without European common data zone
                3 - Smartcard with European common data zone
                4 - Smartcard on which application may be installed by service providers
                 -->
                 <property name="cardType" value="2"/>
        </bean>
```

Pour proposer à certains "userType" (populations) d'utilisateurs d'activer leur carte dans ESCR, il faut renseigner le paramètre DISPLAY_FORM_EUROPEAN_CARD dans l'interface web, "Admin" > "Configurations".

Le paramètre ENABLE_EUROPEAN_CARD permet quant à lui de sélectionner les userType pour lesquels on active par défaut la carte dans ESCR.

Récupération d'une carte depuis ESC-R

Présentation

Pour qu'une carte extérieure puisse être reconnue par ESUP-SGC, il faut déclarer ESUP-SGC dans ESCR.

Ainsi ESUP-SGC sera présenté aux étudiants adhérents au dispositif ESC, et ils pourront accepter que leur carte et informations personnelles soient transmises à votre ESUP-SGC.

Mise en oeuvre

Depuis votre interface ESCR https://router.europeanstudentcard.eu/remote-service en tant que gestionnaire, vous pouvez déclarer une application compatible ESCR.

Au niveau d'esup-sgc, vous pouvez mettre dans security.properties

```
accessRestrictionWSescr=hasHeader('escr-key','123456789ABCDEF')
```

Notez qu'à la fois le nom de l'entête http (escr-key ici) et sa valeur (123456789ABCDEF) sont libres. Il faudra simplement reporter les mêmes dans l'interface ESCR.

En supposant que votre ESUP-SGC est accessible en https://esup-sgc.univ-ville.fr, vous devrez ainsi remplir les champs demandés lors de l'enregistrement de votre service ainsi :

- API root url : https://esup-sgc.univ-ville.fr/wsescr
- Api Key Header name : escr-key
- Api Key : 123456789ABCDEF
- Activation Endpoint : /activate
- Deactivation Endpoint : /deactivate
- Card added Endpoint : /addcard
- Card deleted Endpoint : /deletecard
- Student updated Endpoint : /updatestudent
- Student deleted Endpoint : /deletestudent

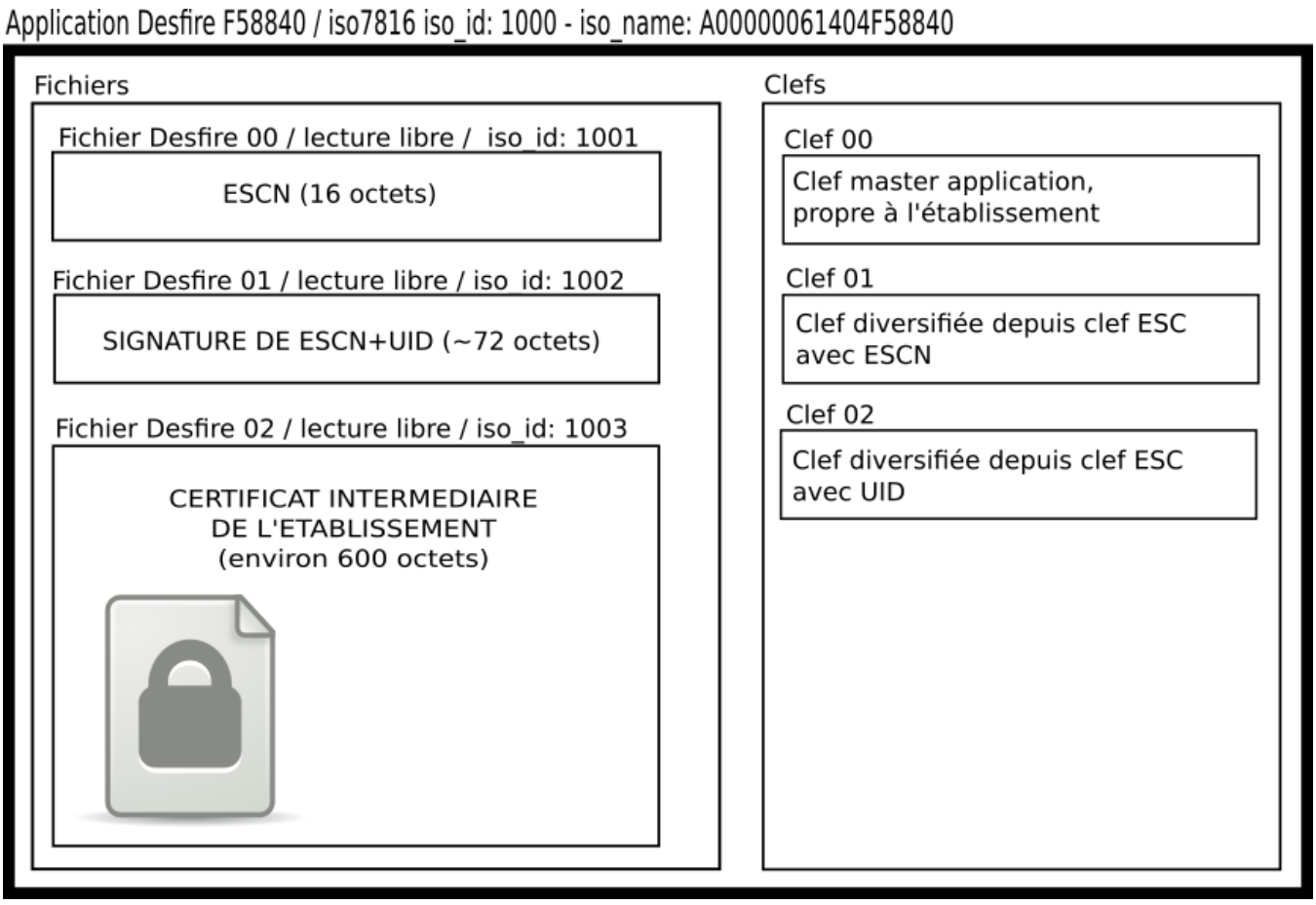
Ecriture et lecture électronique de la DEUINFO

Présentation

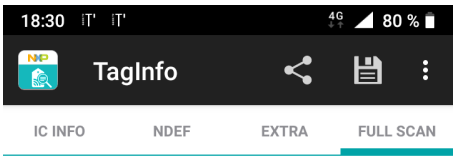
La DEUINFO (**D**ata **E**uropean **U**niversity **Info**) correspond à une application Desfire dans laquelle on indique l'identifiant de carte ESCN (présenté déjà au travers du QR-Code) de manière électronique.

La DEUINFO propose cet identifiant en clair, et donc lisible très simplement. Dans le même temps, la DEUINFO propose des mécanismes de chiffrement permettant de garantir de manière satisfaisante le caractère officiel de la carte.

L'application est ainsi construite suivant ce schéma :



En accès libre, une application type TagInfo de NXP sous Android permet d'afficher directement les 3 fichiers de la DEUINFO, on retrouve ainsi notamment en clair l'ESCN sous sa forme hexadecimal dans le fichier 00, soit d88b02c1894e1038a711001999465982 ici :



- File ID 0x00: Standard data, 16 bytes
  - Communication: plain
  - Read key: free access (0x0E)
  - Write key: master key (0x00)
  - Read/Write key: master key (0x00)
  - Change key: master key (0x00)
  - Contents:

```
[0000] D8 8B 02 C1 89 4E 10 38 |··%··N%8|
[0008] A7 11 00 19 99 46 59 82 |·%%·FY·|
```

- File ID 0x01: Standard data, 72 bytes
  - Communication: plain
  - Read key: free access (0x0E)
  - Write key: master key (0x00)
  - Read/Write key: master key (0x00)
  - Change key: master key (0x00)
  - Contents:

```
[0000] 30 46 02 21 00 D0 D5 E4 |0F%!%···|
[0008] 36 2D 9B 69 94 92 3A 0C |6-·i··:%|
[0010] 4F E7 6E A3 F3 26 4F 67 |O·n··&Og|
[0018] 40 CB 1E A3 0B E6 69 37 |@·%·%·i7|
[0020] 12 38 EC 59 B8 02 21 00 |%8·Y·%!%|
[0028] C1 BF 5B 9C 70 6D 54 AA |··[·pmT·|
[0030] 2D AF 48 98 7D 45 E9 0F |-··H·}E·%|
[0038] 0F 7D 73 F5 3E 61 7B EC |%}s·>a{·|
[0040] EF 21 6C E8 BD 34 35 5C |·!l··45\|
[0048] [none]
```

- File ID 0x02: Standard data, 616 bytes

Afin de permettre une compatibilité avec d'autres technologies que Mifare Desfire, la DEUINFO est également spécifiée pour supporter la norme ISO/IEC 7816-4 : https://www.iso.org/fr/standard/77180.html

Mise en oeuvre

La mise en oeuvre de l'encodage de l'application DEUINFO se fait principalement côté esup-nfc-tag-server. De même esup-nfc-tag-server permet également de lire et vérifier l'ensemble des mécanismes d'une DEUINFO.

Une partie correspondant aux mécanismes de signatures et certificats se fait cependant côté ESUP-SGC.

Au préalable, vous aurez suivi la procédure ESC permettant de générer vos clefs et certificats d'établissements propres à la DEUINFO. Vous aurez également récupéré la clef de base ESC (clef partagée entre établissement mais qui doit restée secrète) permettant la diversification des clefs selon ESCN et UID.
https://router.europeanstudentcard.eu/docs/deuinfo

ESUP-SGC

Dans applicationContext-crous.xml, modifiez le bean escDeuInfoService pour le décommenter au besoin et faire pointer la clef et certificat vers la clef et certificat récupéré depuis la procédure ESC :

```
        <bean id="escDeuInfoService" class="org.esupportail.sgc.services.esc.EscDeuInfoService">
        <property name="pic" value="le-code-pic-de-letablissement"/>
                <property name="deuInfoPrivateKey" value="classpath:META-INF/security/esc/ca.intermediate.key.
der"/>
                <property name="deuInfoPublicKey" value="classpath:META-INF/security/esc/ca.intermediate.cert.
der" />
        </bean>
```

Utilisez openssl au besoin pour convertir vos pem en der - dans src/main/resources/META-INF/security/esc/ vous retrouvez 2 scripts bash d'exemple pour ce faire.

La récupération éventuelle des certificats des établissements partenaires lors de la lecture et validation de la DEUINFO d'une carte partenaire utilise l'API europeanStudentCardService déjà paramétré.

Le cardType envoyé à ESCR doit cependant être revu. En codant la DEUINFO, il doit maintenant passer à 3. esup-sgc propose d'envoyer un cardType différent suivant la date d'encodage, ainsi si vous encodez la DeuInfo à partir du 25 mars 2021 à 10H41 par exemple, il faut que toutes les cartes encodées après cette dat soient envoyées dans ESCR avec un cardType de 3 (et non plus 2).

Pour ce faire, vous pouvez paramétrer votre bean europeanStudentCardService ainsi :

```
        <bean id="europeanStudentCardService" class="org.esupportail.sgc.services.esc.ApiEscrService">
                <property name="enable" value="true"/>
                <property name="webUrl" value="https://api.europeanstudentcard.eu/v1" />
                <property name="key" value="clef-a-recuperer-auprès-d-escr" />
                <property name="restTemplate" ref="restTemplate" />
                <property name="countryCode" value="FR"/>
                <property name="picInstitutionCode" value="le-code-pic-de-letablissement"/>
                <!--
                Type of cards. Possibles values are :
                1 - passive card, with no electronic
                2 - Smartcard without European common data zone
                3 - Smartcard with European common data zone
                4 - Smartcard on which application may be installed by service providers
                 -->
                 <property name="cardType" value="2"/>
        <property name="cardTypes">
          <map>
             <entry key="2021-03-25 10:41:00" value="3"/>
          </map>
        </property>
        </bean>
```

ESUP-NFC-TAG

Ecriture

Dans la structure de votre carte dans laquelle se trouve déjà par exemple votre applciation de contrôle d'accès, vous allez rajouter une nouvelle application ainsi :

```
 <!-- Application DEUINFO de la carte étudiante européenne
               nok A3 : ISO enbaled, 3 AES keys
               amks OB :  configuration changeable, free directory list access without master key
-->
<bean class="org.esupportail.nfctag.beans.DesfireApplication" p:desfireAppId="F58840" p:amks="0B" p:nok="A3" p:
isoId="1000" p:isoName="A00000061404F58840">
        <property name="files">
                <util:list>
                        <!-- ESCN File
                                                        communicationSettings 00 : communication plain text for
clear access
                                                        accessRights E000 : - read access clear 'E'
                                                                            - write access '0' master
key only
                                                                            - read/write access clear
'0'
                                                                           - change acces rights '0'
master key only
                                                                            - fileSize : 16bits ->
000010 -> 100000.
                        -->
                        <bean class="org.esupportail.nfctag.beans.DesfireFile"
p:fileNumber="00" p:communicationSettings="00" p:accessRights="E000" p:isoId="1001"
p:tagWriteApi-ref="escnDeuInfoTagWriteEsupSgc" />

                        <!-- Signature (71 ou 72 octets) - taille variable  -->
                        <bean class="org.esupportail.nfctag.beans.DesfireFile"
p:fileNumber="01" p:communicationSettings="00" p:accessRights="E000" p:isoId="1002"
p:tagWriteApi-ref="signatureDeuInfoTagWriteEsupSgc" />
                        <!-- Certificat - taille variable  -->
                        <bean class="org.esupportail.nfctag.beans.DesfireFile"
p:fileNumber="02" p:communicationSettings="00" p:accessRights="E000" p:isoId="1003"
p:tagWriteApi-ref="certDeuInfoTagWriteEsupSgc"/>
                </util:list>
        </property>
        <property name="keys">
                <util:list>
                        <!-- master key of deuInfo application : private app master key of the university -->
                        <bean class="org.esupportail.nfctag.beans.DesfireKey"
p:keyNo="00" p:keyVer="00" p:key="00000000000000000000000000000000" />
                        <!-- deuinfo master key diversified with ESCN -->
                        <bean class="org.esupportail.nfctag.beans.DesfireKey"
p:keyNo="01" p:keyVer="00" p:desfireKeyService-ref="escnDeuInfoDiversifiedKeyService" />
                        <!-- deuinfo master key diversified with CSN -->
                        <bean class="org.esupportail.nfctag.beans.DesfireKey"
p:keyNo="02" p:keyVer="00" p:desfireKeyService-ref="csnDeuInfoDiversifiedKeyService" />
                </util:list>
        </property>
</bean>
```

La clef 00000000000000000000000000000000 pouvant être changé ici par ce que vous voulez (clef propre à l'université).

Pour que cela fonctionne vous devez en plus également ajouter 5 beans supplémentaires référencés dans cette configuration :

```
    <bean id="escnDeuInfoTagWriteEsupSgc" class="org.esupportail.nfctag.service.api.impl.TagWriteRestWs">
                <property name="idFromCsnUrlTemplate" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/getEscDeuInfoEscn?csn={0}"/>
    </bean>

    <bean id="signatureDeuInfoTagWriteEsupSgc" class="org.esupportail.nfctag.service.api.impl.TagWriteRestWs">
                <property name="idFromCsnUrlTemplate" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/getEscDeuInfoSignature?csn={0}"/>
    </bean>

    <bean id="certDeuInfoTagWriteEsupSgc" class="org.esupportail.nfctag.service.api.impl.TagWriteRestWs">
                <property name="idFromCsnUrlTemplate" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/getEscDeuInfoCertificat?csn={0}"/>
    </bean>

    <bean id="csnDeuInfoDiversifiedKeyService" class="org.esupportail.nfctag.service.desfire.
DesfireDiversificationService">
                <property name="baseKey" value="11111111111111111111111111111111"/>
    </bean>

    <bean id="escnDeuInfoDiversifiedKeyService" class="org.esupportail.nfctag.service.desfire.
DesfireDiversificationService">
                <property name="baseKey" value="11111111111111111111111111111111"/>
                <property name="tagWriteApi">
                        <bean class="org.esupportail.nfctag.service.api.impl.TagWriteRestWs">
                                <property name="idFromCsnUrlTemplate" value="https://esup-sgc.univ-ville.fr
/wsrest/nfc/getEscDeuInfoEscn?csn={0}"/>
                    </bean>
                </property>
    </bean>
```

Ici la clef 11111111111111111111111111111111 est donnée 2 fois de suite et elle doit correspondre à la clef de base de diversification fournie par ESC. Cette clef est partagée par tous les établissements, il est important de faire en sorte de ne pas la divulguer !

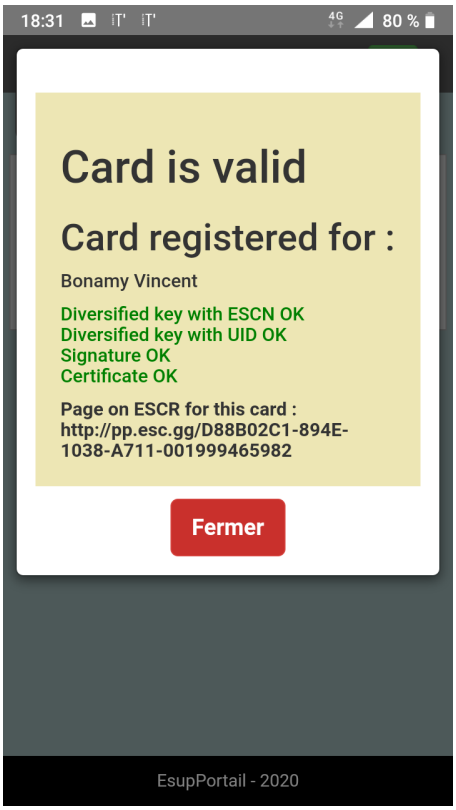
Lecture

Pour la lecture/vérification de la DEUINFO d'une carte au travers d'esup-nfc-tag (desktop ou droid), au niveau d'esup-nfc-tag-server toujours on peut positionner les beans suivants :

```
   <bean id="desfireDeuinfoReadConfig" class="org.esupportail.nfctag.service.api.impl.DesfireReadDeuinfoConfig">
                <property name="baseKey" value="11111111111111111111111111111111"/>
                <property name="description" value="DEUINFO"/>
   </bean>

   <bean id="iso7816DesfireDeuinfoReadConfig" class="org.esupportail.nfctag.service.api.impl.
Iso7816ReadDeuinfoConfig">
                <property name="description" value="Check Iso7816 DEUINFO"/>
   </bean>

    <bean id="esupSgcDeuinfoExtApi" class="org.esupportail.nfctag.service.api.impl.AppliExtRestWs">
            <property name="isTagableUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/isTagableDeuinfo"/>
            <property name="validateTagUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/validateTagDeuinfo"
/>
            <property name="getLocationsUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/locationsDeuinfo"
/>
            <property name="displayUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/deuinfo"/>
            <property name="description" value="DEUINFO"/>
    </bean>

    <bean id="tagIdCheckApiDeuinfoEsupSgc" class="org.esupportail.nfctag.service.api.impl.TagIdCheckRestWs">
                <property name="tagIdCheckUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/tagIdCheckDeuinfo"/>
                <property name="idFromEppnInitUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/idFromEppnInit"/>
                <property name="description" value="Deuinfo"/>
    </bean>
```

On pourra ainsi créer une "application esup-nfc-tag" 'DEUINFO" depuis esup-nfc-tag-server en sélectionnant ces beans DEUINFO pour l'ensemble des paramètres proposés.

Celle-ci permet ainsi notamment de valider techniquement l'ensemble de la DEUINFO, validation des clefs, certificats, ... :



La validation ci-dessus proposée par esup-nfc-tag se fait via la technologie Mifare Desfire et est complète vis à vis des vérifications possibles sur la DEUINFO implémentée dans une Mifare Desfire.

En utilisant non pas Mifare Desfire, mais la norme ISO/IEC 7816-4, la lecture de la carte créée depuis esup-sgc/esup-nfc-tag est également possible ; celle-ci est alors plus simple et légère, elle ne permet pas de vérifier les clefs par exemple, mais elle a l'avantage d'être standardisée et donc possible sur bon nombre de technologies de cartes.

Pour ce faire, on peut passer les APDU suivants normalisés ISO 7816 :

```
# select deuinfo app
00A4040009A00000061404F5884000
# select ESCN file
00A40000021001
# read
00B0000000
# select sign file
00A40000021002
# read
00B0000000
# select cert file
00A40000021003
# read
00B0000000
# read next (cert > 256 bytes)
00B0010000
# read next (cert > 512 bytes)
00B0020000
```

Voici un exemple de ce que ça donne (depuis scriptor ici) :

```
vincent@debian-i7:/tmp$ scriptor -r 'Identive Identive CLOUD 4500 F Dual Interface Reader [CLOUD 4700 F
Contactless Reader] (53201322201041) 01 00'
```

```
Using given card reader: Identive Identive CLOUD 4500 F Dual Interface Reader [CLOUD 4700 F Contactless Reader]
(53201322201041) 01 00
Using T=1 protocol
Reading commands from STDIN
00A4040009A000000061404F5884000
> 00 A4 04 00 09 A0 00 00 06 14 04 F5 88 40 00
< 90 00 : Normal processing.
00A40000021001
> 00 A4 00 00 02 10 01
< 90 00 : Normal processing.
00B0000000
> 00 B0 00 00 00
< D8 8B 02 C1 89 4E 10 38 A7 11 00 19 99 46 59 82
90 00 : Normal processing.
00A40000021002
> 00 A4 00 00 02 10 02
< 90 00 : Normal processing.
00B0000000
> 00 B0 00 00 00
< 30 45 02 20 79 DC F9 A1 C0 2C C5 1F 61 73 28 02
E1 CF EB 65 6A 1B 9E 72 F4 6E E2 D7 12 8A B4 7B
D2 E0 7F 2D 02 21 00 B3 83 1A BD 51 8F 17 36 A2
AE 4A 31 07 1B 7F 75 FF C3 2F 3E E5 85 FD 9A F7
E5 55 09 1D 5F 8F A0 90 00 : Normal processing.
00A40000021003
> 00 A4 00 00 02 10 03
< 90 00 : Normal processing.
00B0000000
> 00 B0 00 00 00
< 30 82 02 64 30 82 02 09 A0 03 02 01 02 02 02 10
00 30 0A 06 08 2A 86 48 CE 3D 04 03 02 30 81 8E
31 0B 30 09 06 03 55 04 06 13 02 46 52 31 0B 30
09 06 03 55 04 08 0C 02 37 36 31 0E 30 0C 06 03
55 04 07 0C 05 52 6F 75 65 6E 31 29 30 27 06 03
55 04 0A 0C 20 55 6E 69 76 65 72 73 69 74 C3 83
C2 A9 20 64 65 20 52 6F 75 65 6E 20 4E 6F 72 6D
61 6E 64 69 65 31 16 30 14 06 03 55 04 03 0C 0D
75 6E 69 76 2D 72 6F 75 65 6E 2E 66 72 31 1F 30
1D 06 09 2A 86 48 86 F7 0D 01 09 01 16 10 73 69
40 75 6E 69 76 2D 72 6F 75 65 6E 2E 66 72 30 1E
17 0D 32 30 30 34 30 36 31 30 31 30 31 32 5A 17
0D 33 30 30 34 30 34 31 30 31 30 31 32 5A 30 7E
31 0B 30 09 06 03 55 04 06 13 02 46 52 31 0B 30
09 06 03 55 04 08 0C 02 37 36 31 29 30 27 06 03
55 04 0A 0C 20 55 6E 69 76 65 72 73 69 74 C3 83
90 00 : Normal processing.
00B0010000
> 00 B0 01 00 00
< C2 A9 20 64 65 20 52 6F 75 65 6E 20 4E 6F 72 6D
61 6E 64 69 65 31 16 30 14 06 03 55 04 03 0C 0D
75 6E 69 76 2D 72 6F 75 65 6E 2E 66 72 31 1F 30
1D 06 09 2A 86 48 86 F7 0D 01 09 01 16 10 73 69
40 75 6E 69 76 2D 72 6F 75 65 6E 2E 66 72 30 59
30 13 06 07 2A 86 48 CE 3D 02 01 06 08 2A 86 48
CE 3D 03 01 07 03 42 00 04 93 B6 D5 D9 B9 42 CF
7E 6B 04 FB 65 B7 D9 A2 3E 9F FA 5F 88 9D 1A F2
08 6A F1 4B 54 08 E6 60 DE 11 8D CA 00 DE 82 5A
2A 62 3A FD 6F DF C0 15 D1 F9 80 94 77 7F 49 80
F2 F1 0A 41 26 0B F6 AF A1 A3 66 30 64 30 1D 06
03 55 1D 0E 04 16 04 14 26 B9 B2 78 0E 7F DC E1
0A 67 11 1E 56 9A A7 DC 2B 92 38 9F 30 1F 06 03
55 1D 23 04 18 30 16 80 14 23 65 6A 02 C5 64 E3
8A 32 81 76 D5 94 9E 48 7A 27 DB 37 4E 30 12 06
03 55 1D 13 01 01 FF 04 08 30 06 01 01 FF 02 01
90 00 : Normal processing.
00B0020000
> 00 B0 02 00 00
< 03 30 0E 06 03 55 1D 0F 01 01 FF 04 04 03 02 01
86 30 0A 06 08 2A 86 48 CE 3D 04 03 02 03 49 00
30 46 02 21 00 FC 0A 75 A8 A2 E1 3A 28 18 8D 71
B4 70 B8 83 3A C3 02 1F EA C5 D7 C0 75 A4 43 60
```

```
AC EA 5C C6 87 02 21 00 90 E9 87 E6 D0 5D 67 2C
10 7F 73 75 CE 7F DF 73 A8 03 9C C6 2A FC F7 DA
B2 67 5C 35 0A 87 80 88 90 00 : Normal processing.
```

Esup-Nfc-Tag permet également cette validation via ISO 7816 en passant ces mêmes APDU (pas de validation des clefs donc par rapport à la validation en Desfire) :

Délégation d'applications Mifare Desfire EV2 (implémentation en cours)

Présentation

Cette dernière partie correspond à la fonctionnalité la plus avancée que propose le projet de Carte Étudiante Européenne en rapport à la technologie Desfire EV2.

Elle propose en effet de mettre en oeuvre la technique DAM (Delegated Application Management ou DMA pour "Délégation de Management d'Application") introduite sur la puce NXP DESFIRE EV2.

L'objectif est :

- de permettre à un établissement accueillant un étudiant extérieur de positionner une nouvelle application Desfire sur la carte de l'étudiant,
- ce en accord avec l'étudiant et son établissement d'origine
- et sans pour autant connaitre au préalable les paramètres techniques/sécurité de la carte, c'est à dire sans connaitre (et sans avoir recours à) la master-key de la carte donc.

Un cas d'usage est de proposer l'application Desfire de contrôle d'accès sur la carte institutionnelle d'un utilisateur extérieur accueilli dans l'établissement.

Dit autrement, cela peut en fait "rendre compatible" une carte extérieure Desfire EV2 avec des applications sécurisées de l'établissement, applications telles que le contrôle d'accès ou encore le paiement Izly.

Mise en oeuvre

Chargement des DAM keys

Pour que les cartes que vous éditez puissent effectivement héberger des applications d'établissements partenaires, il faut que celles-ci présentent les 3 "DAM keys" dans l'application Master (000000).

Pour cela, vous devez configurer l'écriture de ces clefs.

1. Une clef de diversification propre à la carte est demandée par esup-nfc-tag-server auprès d'esup-sgc
2. esup-sgc la génère cette clef aléatoirement et la persiste en base (afin de pouvoir la retrouver lors d'une éventuelle procédure d'écriture d'une DAM) ; chaque carte dispose de sa propre clef de diversification
3. esup-nfc-tag-server calcule les clefs DAM diversifiées en utilisant cette clef de diversification avec comme entrée le CSN (...) de la carte (comme proposé au niveau des spécifications du projet ESC)
4. esup-nfc-tag-serve encodent ces 2 clefs DAM dans la carte (application Master)

Configuration esup-nfc-tag-server

Pour écrire les damKeys, il faut dans le fichier applicationContext-desfire.xml au niveau de votre org.esupportail.nfctag.beans.DesfireTag représentant votre Tag à écrire ajouter une propriété damKeysTagWriteApi faisant référence à bean de type DamKeysTagWriteApi, ce brena étant lui-même à définir.

```
...
<bean id="damKeysTagWriteRestWs" class="org.esupportail.nfctag.service.api.impl.DamKeysTagWriteRestWs">
  <property name="createDamKeysFromCsnUrlTemplate" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/createDamDiversBaseKey?csn={0}"/>
  <property name="damKeysFromCsnUrlTemplate" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/getDamDiversBaseKey?csn={0}"/>
  <property name="resetDamKeysUrlTemplate" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/resetDamDiversBaseKey?csn={0}"/>
</bean>
...
<bean id="desfireComueTagEsupSgc" class="org.esupportail.nfctag.beans.DesfireTag" p:formatBeforeWrite="false" p:
keyStart="00000000000000000000000000000000" p:keyTypeStart="AES" p:keyFinish="00000000000000000000000000000000"
p:keyTypeFinish="AES" p:keyVersionFinish="00" p:damKeysTagWriteApi-ref="damKeysTagWriteRestWs">
....
</bean>
...
```

Configuration esup-sgc

Le cardType envoyé à ESCR doit être revu. En chargeant des DAM keys sur vos cartes, il doit maintenant passer à 4.

Pour ce faire, vous pouvez paramétrer votre bean europeanStudentCardService (dans applicationContext-crous.xml) ainsi :

```
        <bean id="europeanStudentCardService" class="org.esupportail.sgc.services.esc.ApiEscrService">
                <property name="enable" value="true"/>
                <property name="webUrl" value="https://api.europeanstudentcard.eu/v1" />
                <property name="key" value="clef-a-recuperer-auprès-d-escr" />
                <property name="restTemplate" ref="restTemplate" />
                <property name="countryCode" value="FR"/>
                <property name="picInstitutionCode" value="le-code-pic-de-letablissement"/>
                <!--
                Type of cards. Possibles values are :
                1 - passive card, with no electronic
                2 - Smartcard without European common data zone
                3 - Smartcard with European common data zone
                4 - Smartcard on which application may be installed by service providers
                 -->
                 <property name="cardType" value="2"/>
        <property name="cardTypes">
          <map>
             <entry key="2021-03-25 10:41:00" value="3"/>
                        <entry key="2021-07-07 16:25:00" value="4"/>
          </map>
        </property>
        </bean>
```

Création d'une application Mifare Desfire sur une carte d'un établissement partenaire

L'implémentation correspondant à cette partie est en cours.