MajecSTIC 2010
Bordeaux, France, du 13 au 15 octobre 2010

Perspectives d'utilisation du serveur web embarqué dans la carte à puce Java Card 3

Nassima KAMEL, Jean louis LANET, Julien Igouchi CARTIGNY, Matthieu BARREAUD

Université de Limoges, Laboratoire XLIM, Equipe SSD, 83 rue d'Isle, 87000 Limoges - France.

Contact : {nassima.kamel, julien.cartigny, jean-louis.lanet}@xlim.fr,
matthieu.barreaud@etu.unilim.fr

Résumé
Depuis son apparition la carte à puce a très vite été adoptée grâce à sa portabilité et une authentification robuste. La Java Card est la plate forme dédiée à de tels objets de faibles ressources avec comme but de masquer la complexité sous-jacente du matériel. La dernière version est la Java Card 3 qui se rapproche de J2ME par rapport aux précédentes versions. La principale nouveauté de cette spécification est l'intégration d'un nouveau modèle de programmation : les servlets qui permettent l'utilisation des protocoles standards HTTP/ HTTPS offrant ainsi un accès plus sécurisé, plus simple et plus rapide face au protocole par APDU (*Application Protocol Data Unit*) [12] utilisé dans les précédentes versions. Dans cet article, nous présentons cette nouvelle version de Java Card et nous montrons l'apport de l'intégration du web dans la carte à puce à travers des cas d'utilisation. Nous présentons particulièrement une application que nous avons développé « Disque Virtuel » qui offre un accès à un très grand espace de stockage distant, dont les fichiers sont cryptés/décryptés à la volée par la carte.

Abstract
Since its introduction the Smart Card was quickly adopted thanks to its portability and a more robust authentication. The Java Card platform is dedicated such objects with limited resources; in order to hide the underlying complexity of the material. The latest version is the Java Card 3, which approximates J2ME more than its previous versions. The main novelty in this specification is the inclusion of a new model of programming: the servlets using standard internet protocols HTTP/HTTPS providing, simpler, quick and a more secured access to deal with APDU (Application Protocol Data Unit) [12] protocol used in the previous versions. In this paper we present a new version of Java Card and we show through use cases, the advantages of integrating the Web into the smart card. We present our application "Virtual Drive" that offers access to a large area of remote storage, including files, encrypted/decrypted on fly by a smart card.

Mots-clés : Java Card 3, serveur web, carte à puce.

Keywords : Java Card 3, web server, smart card.

1. Introduction

L'intégration du web dans le monde la carte est une idée initiée par OMA (Open Mobil Alliance), organisation qui regroupe les principaux opérateurs de la téléphonie mobile. Elle annonce la spécification SCWS (Smart Card Web Serveur) qui définit l'interface d'un serveur HTTP embarqué dans la carte à puce [6]. Cette spécification est destinée aux cartes SIM et permet une connexion locale entre le navigateur internet et le terminal. Ainsi elle offre aux operateurs de téléphonie mobile la possibilité d'offrir à leurs clients un grand nombre de services utilisant le protocole HTTP et l'administration distante à travers le serveur web embarqué. Cette technologie n'est pas liée à la Java Card 3 qui définit également un serveur web embarqué. La spécification de l'OMA est plus simple, dissociée de Java Card et a pour but de fournir un «

simple » serveur web embarqué. Elle correspond à des cartes qui n'ont pas les performances suffisantes pour supporter la norme Java Card 3.
La spécification Java Card 3.0, comporte deux éditions, la *Classic Edition* et la *Connected Edition*. La *Connected Edition* est compatible avec la spécification OMA-SCWS, qu'elle étend considérablement, en particulier au niveau des fonctionnalités disponibles au niveau du conteneur Java Card 3.0. Cette spécification introduit une plus grande flexibilité, ainsi que de nombreuses fonctions de sécurité supplémentaires, allant de SSL à un système de permissions donnant accès aux diverses fonctions.
La principale nouveauté dans la Java Card 3 est l'intégration de certains standards web, notamment HTTP. Elle permet le déploiement simultané d'applications traditionnelles de cartes à puce et d'applications web (à l'aide d'un nouveau modèle de programmation hérité de J2ME : les servlets [8]. Ces applications web fournissent ainsi des services aux autres entités sur le réseau IP. Elles sont accessibles en utilisant les protocoles Internet standard tels que HTTP/ HTTPS pour permettre une facilité d'intégration avec les infrastructures existantes des services Internet et avec les applications utilisateurs. De plus, de nouvelles API sont supportées telles que les Threads et String. Des fonctionnalités de sécurité ont également été rajoutées, telles que : la vérification obligatoire du *bytecode*, l'isolation des applications par un firewall, communication sécurisée, les annotations, etc.
La première partie de cet article présente les principales caractéristiques de la Java Card3 (architecture physique et logique, mécanismes de sécurité supportés). La deuxième partie montre à travers quelques cas d'utilisation l'apport de l'intégration d'un serveur web dans la carte, nous présentons ensuite un exemple d'application Java Card 3 où nous exploitons différents outils offerts par cette nouvelle technologie pour effectuer des transferts de fichiers cryptés via la carte. Le but de cette application est de pouvoir récupérer à distance et en toute sécurité (grâce à la fonction de cryptage/décryptage implémentée) des fichiers stockés sur un serveur distant.

2. Présentation de la Java Card 3

Le Java Card est un sous ensemble du langage Java destiné aux objets à faibles ressources comme les cartes à puce [12]. Il s'agit d'une plate-forme portable (basée sur une machine virtuelle JCVM : Java Card Virtual Machine), sécurisée et multi-applicative.
La version Java Card 3.0.1 est la spécification la plus récente de la plateforme Java Card de Sun Microsystems. Elle conserve les principes des précédentes versions (exécution d'applications multiples, stricte isolation des applications, …) et permet d'exécuter les applications développées pour les versions précédentes, tout en apportant des évolutions majeures. La spécification de la plate-forme Java Card 3 est disponible en deux éditions distinctes [9]: la Java Card 3 *Classic Edition* et la Java Card 3 *Connected Edition*.
La spécification Java Card 3 *Classic Edition* décrit une évolution légère de l'actuelle plate-forme Java Card 2.2.2 et cible des marchés à faible valeur ajoutée, en privilégiant la continuité avec les plates-formes précédentes.
La Java Card 3 *Connected Edition*, à laquelle on s'intéresse dans notre projet, est conçue pour la nouvelle génération de cartes à puce, et s'adresse donc au marché de plus haut de gamme, en particulier les cartes SIM les plus puissantes et certains marchés d'authentification. Avec cette nouvelle technologie, la carte à puce peut communiquer avec le protocole TCP/IP comme tout autre élément du réseau internet. Ainsi, elle sera capable de fournir ou d'accéder à des services du réseau internet. Dans la suite de ce papier nous nous intéressons à cette version de la plateforme et nous la désignons simplement par Java Card3.

2.1. Architecture de la Java Card 3

Les cartes à puce Java Card 3 sont capables de recevoir et d'exécuter des applications après que celles-ci soient délivrées au client (on parle de post-issuance). Pour cela, la carte accueille chaque application Java dans un conteneur qui lui fournit des services et gère son cycle de vie.
Deux types de conteneurs sont définis :

- Le conteneur d'applet qui gère les applets classiques (applets supportées dans la Java

Card2.2) et étendus durant leur cycle de vie à travers des Applet API.
- Le conteneur de servlet qui contient des applications web et gère leur cycle de vie grâce à la Servlet API [2].

Ces conteneurs peuvent accueillir trois modèles d'applications:

Les Applets classiques :

- La communication se fait via le protocole APDU (norme ISO 7816-4).
- Utilisation d'API compatibles avec la précédente version.

Les Applets étendus :

- La communication se fait aussi via le protocole APDU. Permet l'utilisation des API des applets classiques mais aussi de nouvelles API tels que les Threads, Strings, et GCF (*Generic Connection Framework*).

Les Servlets :

- La communication se fait via le protocole HTTP ou HTTPS.
- Basées sur la version spécification des Servlets 2.4 [2].

La figure ci-dessous résume les différentes modèles d'applications et leur intégration dans les conteneurs de la Java Card 3 :

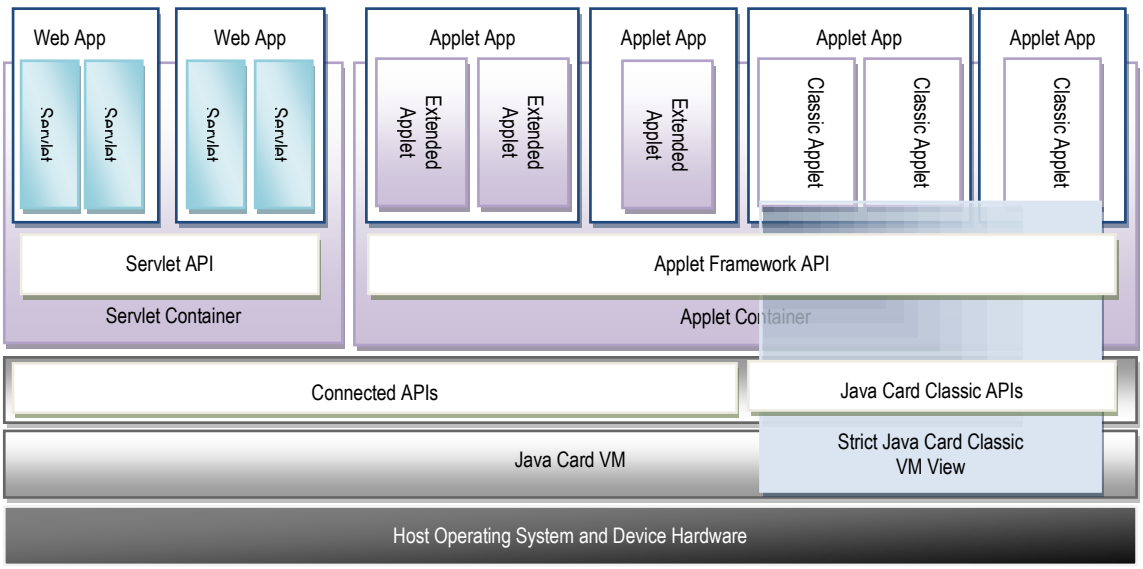

FIG.1 - Architecture de la plateforme Java Card 3 *Connected Edition* [7].

2.2. La pile de protocoles et les connections physiques

La plate-forme Java Card 3, est destinée à fonctionner sur une grande variété de cartes à puce et de dispositifs sécurisés avec des ressources limitées. La figure2 illustre la pile de protocoles de la plate-forme Java Card 3.

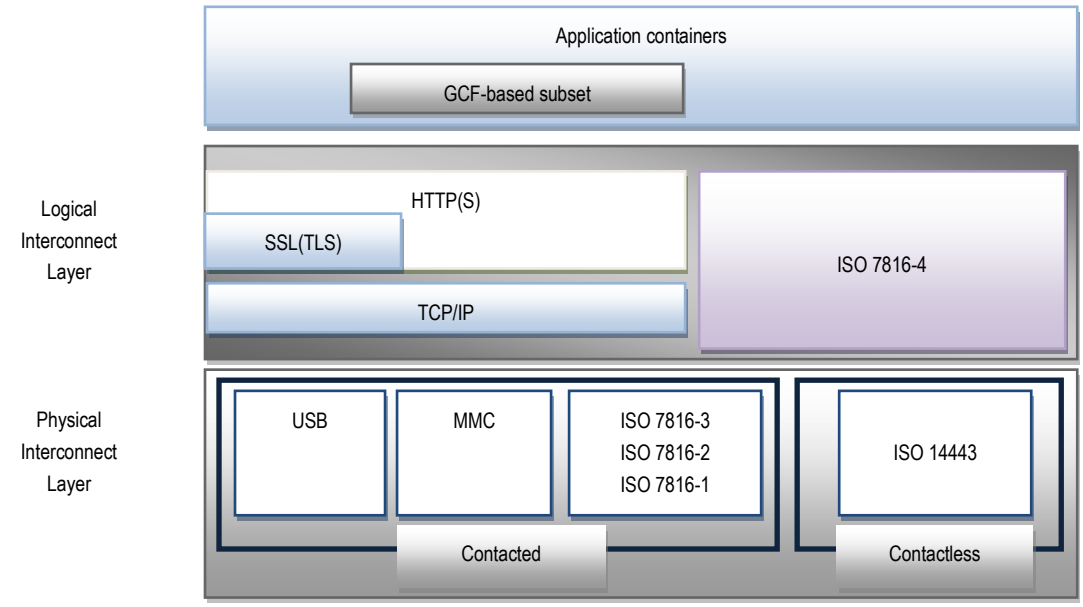

FIG.2 - Pile protocolaire et type de connexions dans la Java Card 3 *Connected Edition* [7]

Connectivité physique

La plate-forme Java Card *Connected Edition* est conçue pour des cartes à puce possédant des interfaces physiques à haut débit. Les protocoles vers des interfaces d'E/S additionnelles ont été intégrés, comme :

- Universal Serial Bus (USB) MultiMediaCard (MMC) ISO 7816-3.
- ISO 14443 (contactless)

Le protocole de communication par APDU basé sur la norme ISO 7816-4 est aussi supporté.

Connectivité Logique

La mise en œuvre de la plate-forme Java Card 3 doit fournir aux applications une interface réseau logique qui prend en charge les protocoles réseau suivants :

- Internet Protocol (IP) Transmission Control Protocol (TCP).
- Universal Datagram Protocol (UDP).
- Transport Layer Security (TLS).
- Hypertext Transfer Protocol (HTTP).
- Secure Hypertext Transfer Protocol (HTTP sur TLS).

2.3. La sécurité dans Java Card 3

La disponibilité d'un serveur web dans une carte à puce permet l'accès aux services de la carte à travers un protocole standard du monde internet et facilite la communication avec l'utilisateur. Du point de vue de la sécurité, la disponibilité de TLS permet la sécurisation de bout-en-bout entre la carte et son correspondant. De plus, la carte à puce offre les services classiques de sécurité : une gestion fine des utilisateurs, de leur authentification et du contrôle d'accès aux ressources. La plateforme Java Card 3, outre les dispositifs classiques d'isolation offre de nouveaux mécanismes de sécurité :

Mécanismes de sécurité déjà existants dans la précédente version :

- Récupération de mémoire: Un mécanisme automatique appelé ramasse-miettes (*Garbage Collector*) permet de libérer les zones mémoires des objets volatiles (objets à durée de vie

limitée stockés dans une mémoire volatile) non référencés depuis une longue durée. Pour des objets persistants (stockés dans une mémoire non-volatile) ce mécanisme peut être appelé à la demande.
- L'isolation des objets des applications grâce à un pare-feu (*firewall*).

Nouveau mécanismes de sécurité :
- Vérification de bytecode obligatoire: ce contrôle est maintenant systématique lors du chargement des applets sur la plate-forme.
- Gestion de données volatiles: certains objets peuvent être conservés en RAM, ce qui peut rendre certaines attaques plus complexes.
- Un contrôle fin de l'accès (*Access Controller*) aux services de la plateforme grâce à un moniteur de référence.
- Une communication sécurisée basée sur TLS.
- Une authentification des utilisateurs pour l'accès aux ressources web.

3. Cas d'utilisation

L'intégration d'un serveur web embarqué introduit plusieurs nouvelles problématiques au domaine des cartes à puce vis-à-vis des évolutions des téléphones mobiles qui permettent d'accéder à internet et tous les services offerts sur ce réseau. On peut se demander l'utilité même d'embarquer un serveur web dans la carte à puce. Dans ce paragraphe, nous montrons à travers des cas d'utilisation l'apport de cette nouvelle plateforme au niveau facticités d'accès et d'utilisation et au niveau sécurité.

3.1. Accès rapide et simple aux services offerts par la carte

L'utilisation des protocoles standards d'internet offre une plus grande flexibilité d'utilisation et une plus grande aisance d'administration pour les émetteurs de la carte (industriels et opérateurs).

L'utilisateur peut désormais lire le contenu de sa carte ou faire des transactions bancaires ou commerciales en toute sécurité à partir de n'importe quel terminal contenant un navigateur web [7]. L'émetteur de la carte peut installer plusieurs services accessibles localement sur la carte via un navigateur web comme tout autre service d'internet. L'accès à internet ou à un service disponible sur la carte est transparent pour l'utilisateur. L'utilisateur peut télécharger les applications qu'il souhaite en accédant à une page d'accueil qui lui propose une liste de services et des contenus sur internet. Ces applications peuvent également être configurées selon ses besoins. En demandant d'installer une application, le propriétaire de la carte rempli un formulaire et l'application installée sera configurée selon ses besoins indiqués dans le formulaire.

Du coté de l'émetteur de la carte, l'utilisation d'un serveur web embarqué et du protocole HTTP(S) standard permet une administration à distance des applications installées sur la carte. En effet, un administrateur peut effectuer un suivi des applications, faire des mises à jour ou des installations, à condition que la carte soit connectée à un réseau internet.

3.2. Amélioration de l'interface d'accès aux services offerts par l'émetteur

L'utilisateur souhaite avoir une interface d'accès aux services offerts par la carte et qui soit conviviale et facilement personnalisable. L'utilisation des standards du web permet d'avoir une expérience utilisateur continue et équivalente à celle qu'on trouve lorsqu'on navigue sur internet. Ces interfaces sont offertes par les lecteurs de carte tels que les téléphones mobiles et sont différents d'un appareil à l'autre. L'introduction d'un nouveau modèle d'application complètement dédié au mode web (servlet) offre aux opérateurs ou émetteurs de la carte la possibilité de concevoir des interfaces conviviales et standardisées pour tous les lecteurs à condition que celui-ci dispose d'un navigateur internet. Dans le cas d'une carte SIM, par exemple, une gestion du carnet d'adresse par une interface est facilement réalisable. De plus la forte connectivité de la carte permet d'envisager un enrichissement de ce genre d'applications.

3.3. Sécurité de connexion d'un terminal distant à une application critique

Une carte à puce Java Card 3 est une carte à forte connectivité avec une possibilité d'accéder à

internet via un réseau intermédiaire, exemple: réseau 2G/3G pour les cartes SIM. L'aspect sécurité est également renforcé. D'une part la carte à puce apporte son aspect *tamper-resistance*. Elle enferme des données chiffrées que le microprocesseur embarqué sera en mesure de traiter en fonction des instructions fournies par le lecteur de carte. Afin de garantir la sécurité de ces opérations, plusieurs dispositifs de protection ont été apportés à la carte à puce [5]. Les matériaux constituant le corps de la carte visent à faire échouer les attaques chimiques d'extraction du micromodule. De plus, tous les composants sont sur le même silicium; le microprocesseur et ses capteurs sont enrobés dans une résine, ce qui rend difficile la pose de sonde pour espionner les bus internes.
D'autre part, l'aspect web de la Java Card 3 apporte l'utilisation du protocole TSL ainsi qu'une gestion fine des utilisateurs (définition des rôles et des permissions), de leur authentification et de l'accès aux différentes ressources de la carte. S'ajoute à cela d'autres aspect sécuritaires qui ont été renforcés et rajoutés dans la Java Card (Vérifieur de *bytecode*, gestion de données volatiles, récupération de mémoire).

4. Sécurité des application Web

Les mécanismes mis en place peuvent être contournés par des attaques exploitant les vulnérabilités applicatives [6]. Des failles dans le développement des applications web peuvent ouvrir des portes à différents types d'attaques web, tels que l'injection de données malicieuses dans une page web (XSS, CSRF) [1,11], le vol d'identifiants de session ou l'accès non autorisé à des ressources [8]. Un attaquant peut alors réussir à effectuer du vol d'information, troubler le bon fonctionnement d'un service, accéder à des ressources non autorisées, modifier des données, etc. Ainsi, une bonne conception et de bonnes pratiques de développement de l'application web s'imposent.
Une application web dans Java est une archive possédant l'extension .war déployée et instanciée dans le conteneur web. Elle doit respecter une structure de fichier la composant (les servlets, ressources statiques, descripteurs de déploiement). Pour sécuriser une application web une méthodologie de développement consiste à mettre en place des contremesures pour éviter les vulnérabilités citées précédemment, parmi ces contremesures **:**

- Valider et filtrer les entrées : Ce mécanisme consiste à valider la longueur, le type et la syntaxe de toutes les entrées saisies.
- Echapper certaines données : Consiste à protéger certains caractères afin qu'ils gardent les sens souhaités. Un code malicieux peut être injecté à plusieurs endroits (exemple dans les attributs communs de HTML, dans des attributs JavaScript, etc.). Des mesures de prévention sont nécessaires à différents niveaux pour éviter qu'un code malicieux puisse s'exécuter [11].
- Protéger la session de l'utilisateur par un jeton de session qui doit être régénéré à chaque nouvelle authentification [3].
- Associer un timeout à chaque session au-delà duquel une déconnexion automatique est effectuée si l'utilisateur est inactif [1].
- Eviter les messages d'erreurs détaillées contenant des informations de débogage ou des informations relatives au chemin.
- Utiliser SSL pour toutes les connexions authentifiées et la transmission des données sensibles, tel que les données d'authentification, numéro de carte bancaire, Etc.
- Etc.

5. Exemple d'application : « le disque virtuel »

Dans le but de tester cette nouvelle plateforme et les différents dispositifs qu'elle intègre, nous avons développé une des premières applications Java Card 3. Notre application, le Disque Virtuel, est un gestionnaire de fichiers sur carte à puce JavaCard3. Il permet de récupérer et d'envoyer des fichiers sur des serveurs FTP distants. Ces fichiers peuvent être cryptés à la volée par la carte ce qui garantit une sécurité maximale en cas de compromission du serveur web. La carte contient les clés de décryptage des fichiers ainsi que l'adresse où se trouvent ces fichiers.

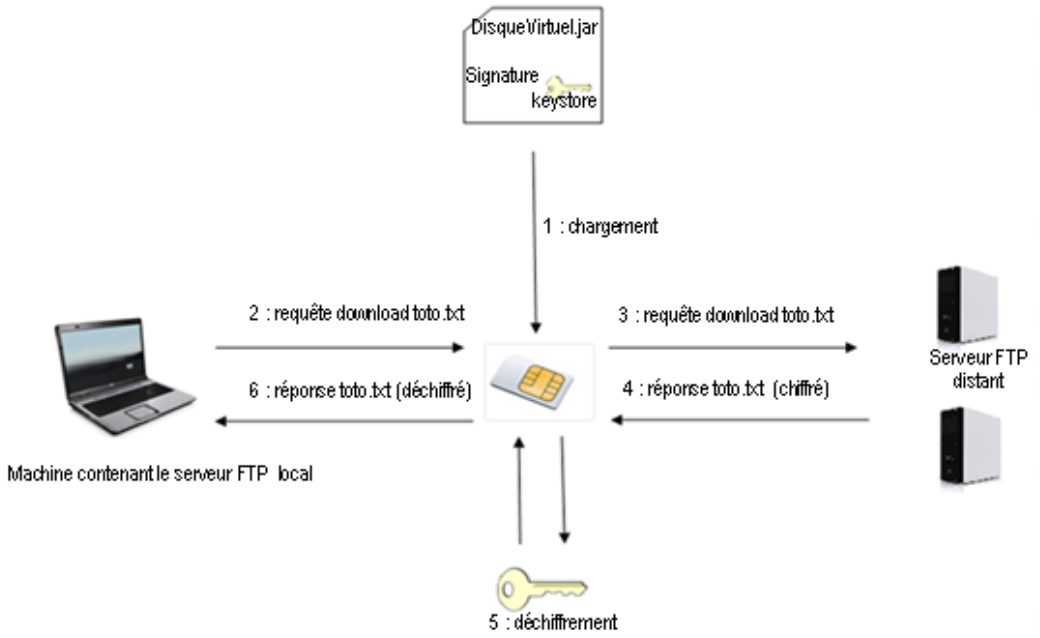

FIG.3 – Téléchargement d'un fichier via la carte à puce Java Card 3.

La figure ci-dessus illustre le mécanisme de transfert de fichiers via notre application. Avant d'être chargée dans une carte à puce, notre application doit être signée avec une clé lui permettant d'avoir des autorisations spécifiques (1) : par exemple, accepter les connexions extérieures. La signature permet donc d'appartenir à un domaine de protection spécifique.

Le mécanisme de transfert de fichiers est très simple. L'utilisateur doit posséder un serveur FTP sur sa machine pour pouvoir envoyer ou récupérer des fichiers. Par exemple pour le téléchargement d'un fichier (2), la carte récupère le fichier sur un serveur FTP distant (3,4), le déchiffre à la volée (5) puis l'envoi sur le serveur FTP local (6). Le mécanisme d'uploade est simplement l'inverse du téléchargement. La carte récupère le fichier sur le serveur FTP local, le chiffre à la volée puis l'envoi sur un serveur FTP distant.

L'avantage de cette application est d'avoir un outil permettant de manipuler un grand nombre de fichiers stocké à distance d'où son nom Disque Virtuel. Elle permet de garantir l'intégrité des fichiers. En cas de compromission du serveur Web où sont stockés nos fichiers, un attaquant ne pourra pas les utiliser sans la clé de décryptage associée pour chaque fichier. On peut ainsi voir ce disque virtuel comme un disque de stockage où les données ne seraient visibles que par le propriétaire de la clé. L'utilisateur peut récupérer ses données à partir de n'importe quel terminal distant à condition qu'il dispose d'un serveur FTP local.

Notre application utilise des mécanismes de sécurité comme une authentification par code PIN ou encore les annotations. On a rendu cette application intuitive afin de montrer les capacités des cartes à puce Java Card 3 par rapport aux anciennes.

A travers cette application nous avons pu constater la facilité de développement en Java Card 3 comparé à sa précédente version Java Card 2.2. Java Card 3 se rapproche beaucoup du langage Java ME ce qui en fait un outil de développement plus simple à adapter pour les développeurs Java. Les développeurs d'applications Java Card 2.2 sont tenus d'avoir des connaissances spécifiques à ce langage. Les transactions laborieuses à mettre en place en JavaCard2.2, ont grandement été simplifiées en Java Card 3 avec l'utilisation d'annotations. De plus, contrairement à la Java Card 2.2 on dispose d'un *Garbage collector* pour supprimer les objets inutilisés en RAM. Le développement a aussi été simplifié par la prise en charge de nouveaux types comme le Integer et String.

6. Conclusion

La Java Card 3 est une grande évolution de la plate forme Java Card. L'ouverture de la carte aux standards du web HTTP/HTTPS offre un accès plus simple et plus rapide au service qu'elle offre, la sécurité a été également renforcée par une gestion plus fine des utilisateurs et une vérification des applications avant leurs chargement (vérificateur de *bytecode* obligatoire) et pendant leurs exécution (pare feu, isolation de contexte d'exécution, ramasse miette, etc.). D'autre partt cette plateforme offre aux développeurs un moyen de programmation plus simple à adapter qui se rapproche beaucoup de Java ME.

Dans cet article nous résumons à travers un ensemble de cas d'utilisation les avantages de l'embarquement d'un serveur web dans la carte à puce. Notre application « Disque Virtuel » est un exemple d'application pouvant être implémentée dans la carte grâce à cette nouvelle plateforme. Cette application consiste à effectuer des transferts de fichiers via la carte qui va les crypter avant de les transférer au serveur distant où ils seront stockés. L'avantage de cette application est la possibilité d'un accès sécurisé (les fichiers sont cryptés et décryptés par la carte) à l'ensemble des fichiers stockés sur le serveur distant.

Toutefois, le développement des applications web nécessite d'adopter des mesures afin de garantir leurs sécurité et de prévenir certaines attaques. En effet les attaques sur les applications web sont très répandus (XSS, CSRF, etc.) et leurs conséquences peuvent être très graves. La carte manipule des données très sensibles. Une méthodologie du développement des applications web est nécessaire afin d'éviter qu'elle soit vulnérables aux attaques web.

Bibliographie

1. Cross-Site Request Forgery (CSRF), http://www.owasp.org/index.php/Cross-Site_Request_Forgery
2. 2. Java™ Servlet Specification, Version 2.4, http://jcp.org/aboutJava/communityprocess/final/jsr154/index.html
3. Mitja Kolšek, Session Fixation Vulnerability in Web-based Applications. ACROS Security. 2007.
4. OMA: Smartcard Web Server Requirements Approved Version 1.0 – 21 Apr 2008.
5. OMA: Smartcard-Web-Server Approved Version 1.0 – 21 Apr 2008.
6. OWASP Top 10 for JAVA EE. 2007.
7. Runtime Environement specification, Java Card Platform, Version 3.0.1 Connected Edition.
8. Servlet specification, Java Card Platform, Version 3.0.1 Connected Edition.
9. The Java Card™ 3 Platform, White Paper, August 2008.
10. Wolfgang Rankl and Wolfgang Effing. Smart Card Handbook.
11. XSS (Cross Site Scripting) Prevention Cheat Sheet. http://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet#XSS_Prevention_Rules
12. Z. Chen. Java Card Technology for Smart Cards. Addison, Wesley, 2000.