R0ANNE

Zeste de savoir

Anatomie d'un titre de transport : du billet à la carte à puce

2 juillet 2021

Table des matières

Introduction

Vous avez toujours rêvé de comprendre les informations absconses présentes sur votre billet de train? Bienvenue dans le monde de la billettique.

Dans cet article, nous allons explorer en profondeur les informations présentes sur différents supports répandus (billet papier, ticket de métro, carte à puce, application sur smartphone sans contact) et les techniques que ceux-ci tendent à utiliser, tout en vulgarisant quelques notions de sécurité électronique et matérielle. Le degré de complexité dépendra du système examiné.

À noter que cet article ne vous sera pas utile pour frauder, tous les titres de transports dont il est question ici étant protégés par des signatures cryptographiques, ou bien par des sommes de contrôle propriétaires (sauf le ticket de métro, qu'on est en train de faire disparaître).

1. Le billet cartonné « classique »

Commençons par nous intéresser au billet cartonné «classique», mesurant environ 20 x 8 cm, que vous pouvez imprimer sur les bornes libre-service.

/opt/zds/data/contents-public/anatomie-dun-tit

FIGURE 1.1. – Face recto d'un billet IATA classique. *Source: Wikimedia Commons / Domaine public*

Tout d'abord, un peu de contexte historique: toute entreprise faisant rouler des trains et vendant des billets informatiquement doit s'équiper d'un système de réservation (dans le jargon du voyage, GDS). Les trois principaux GDS dans le monde s'appellent Amadeus, Sabre et Travelport. Ils sont tous issus du monde du transport aérien (logique, il y a beaucoup plus de compagnies aériennes que ferroviaires).

Le GDS de la SNCF, Résarail, est directement basé sur le GDS Sabre de la compagnie aérienne American Airlines, développé dès la fin des années 1950, qui a été réécrit en plusieurs langages ensuite, dont une licence a été achetée par la SNCF en 1988, et dont le déploiement de la version adaptée par les soins de la SNCF a commencé avec grand fracas en 1993 (dans le cadre de ce qui est appelé en interne le projet SOCRATE , et qui fut largement relaté par la presse de l'époque [1]footnote:1). Par rapport à l'ancien système de réservation, le nouveau système introduit notamment le yield management (pratique consistant à faire varier le prix du billet selon le nombre de places restantes, et déjà répandue depuis plus longtemps dans l'hôtellerie et l'aviation).

Pourquoi est-ce que je vous explique tout ça? Parce que c'est ce qui explique aussi que la SNCF utilise des systèmes d'impression de billets directement issus du monde aérien, qui utilise ses propres normes.

En effet, le billet cartonné vendu par la SNCF est appelé, selon ses propres termes, billet au «format IATA» [3]footnote:4 (l'IATA étant l'association mondiale des compagnies aériennes, un peu comme la GSMA pour les opérateurs mobiles, et par ailleurs une organisation qui occupe diverses fonctions comme attribuer des codes utilisés commercialement à chaque aéroport). L'IATA édicte, en effet, des spécifications techniques pour les machines à impression de billets cartonnés à bande magnétique, dont elle vend le seul accès à plusieurs milliers de dollars l'unité. Cela comprend notamment:

— La norme ATB-2 (ATB signifie *Automated Ticketing and Boarding Pass Printer*, soit imprimante automatique pour billetterie et titres d'embarquement), qui spécifie comment fonctionnent lesdites imprimantes,
— Le format PECTAB (*Parametric Table*, soit table des paramètres) qui est un format de fichier de configuration permettant d'indiquer à quelle position imprimer du texte sur lesdites imprimantes,
— La norme BCBP (*Bar-Coding Boarding Pass*, soit titre d'embarquement à codes-barres) qui fournit des consignes générales sur la manière dont doit être encodé un code-barre sur un billet d'avion, et préconise notamment d'utiliser des codes-barres de type PDF417 ou Aztec pour cela.

Maintenant que nous savons pourquoi la SNCF appelle le billet IATA comme tel, intéressons-nous à ce qui est écrit dessus.

1. [2]footnote:1 «La SNCF débauche SOCRATE», 13 janvier 1993, l'Humanité

FIGURE 1.2. – Recto-verso d'un billet IATA. *Source: Wikimedia Commons / Domaine public*

Peu de documents publics évoquent la structure des informations sur le billet SNCF, mais il y en a quand même un: la norme RCT2 (*Rail Combined Ticket 2*) [4]footnote:2.

Il s'agit d'un document récent tentant d'harmoniser l'aspect des titres de transport, dont la rédaction a été commandée par le Règlement (UE) n° 454/2011 de la Commission du 5 mai 2011 [5]footnote:3.

Si le billet cartonné actuel ne suit pas exactement ce standard, la section C.2 de ce document, clairement rédigée par la SNCF (plusieurs gallicismes sont présents de même qu'un logo sur les billets), nous donne du grain à moudre et nous fournit la structure globale des données encodées dans le code-barre PDF417 (que nous verrons après).

On y apprend également qu'il existe un document partagé entre les opérateurs ferroviaires nommé «International Rail Transport Committee (CIT) - *CIT's CIV ticketing manual, Appendix 5 : CIV Ticket Manual*»

Ce document, qui n'est pas public, donne les caractéristiques d'impression du billet de train européen. Par exemple, le type de papier:

> All travel documents described in B.6 are designed to be printed on blank paper coupons with security background and issued electronically. The international standard for security paper and the characteristics of blank paper coupons to be used as travel documents compliant with TAP (types and quality of paper, security features integrated in the body paper, mandatory reference colours, copyright, dimensions for paper tickets, etc) are described in the "CIV Ticket Manual (GTT-CIV)" of CIT. They cannot be reproduced here both for avoidance of frauds and for intellectual property reasons. They will be referenced in the following as "CIT coupons". Information on the procurement of blank coupons can be found in following chapter 9.
>
> *Annexe B.53: «Direct fulfilment application guide» des TAP TSI* .

On voit d'ailleurs une mention «© CIT 1996» en bleu clair en bas à gauche de chaque billet. Le CIT étant une association mondiale de compagnies ferroviaires, l'IATA du rail.

Voici une version annotée du recto-verso du billet montré plus haut:

2. [6]footnote:2 Aussi aussi appelée annexe B.6 : «Réservation électronique de places assises/couchettes et production électronique de documents» des TAP TSI (Telematics Applications for Passenger Services - Technical Specification for Interoperability) de l'European Railway Agency.

3. [7]footnote:3 Nommé en entier: «Règlement (UE) n° 454/2011 de la Commission du 5 mai 2011 relatif à la spécification technique d'interopérabilité concernant le sous-système «applications télématiques au service des voyageurs» du système ferroviaire transeuropéen.»

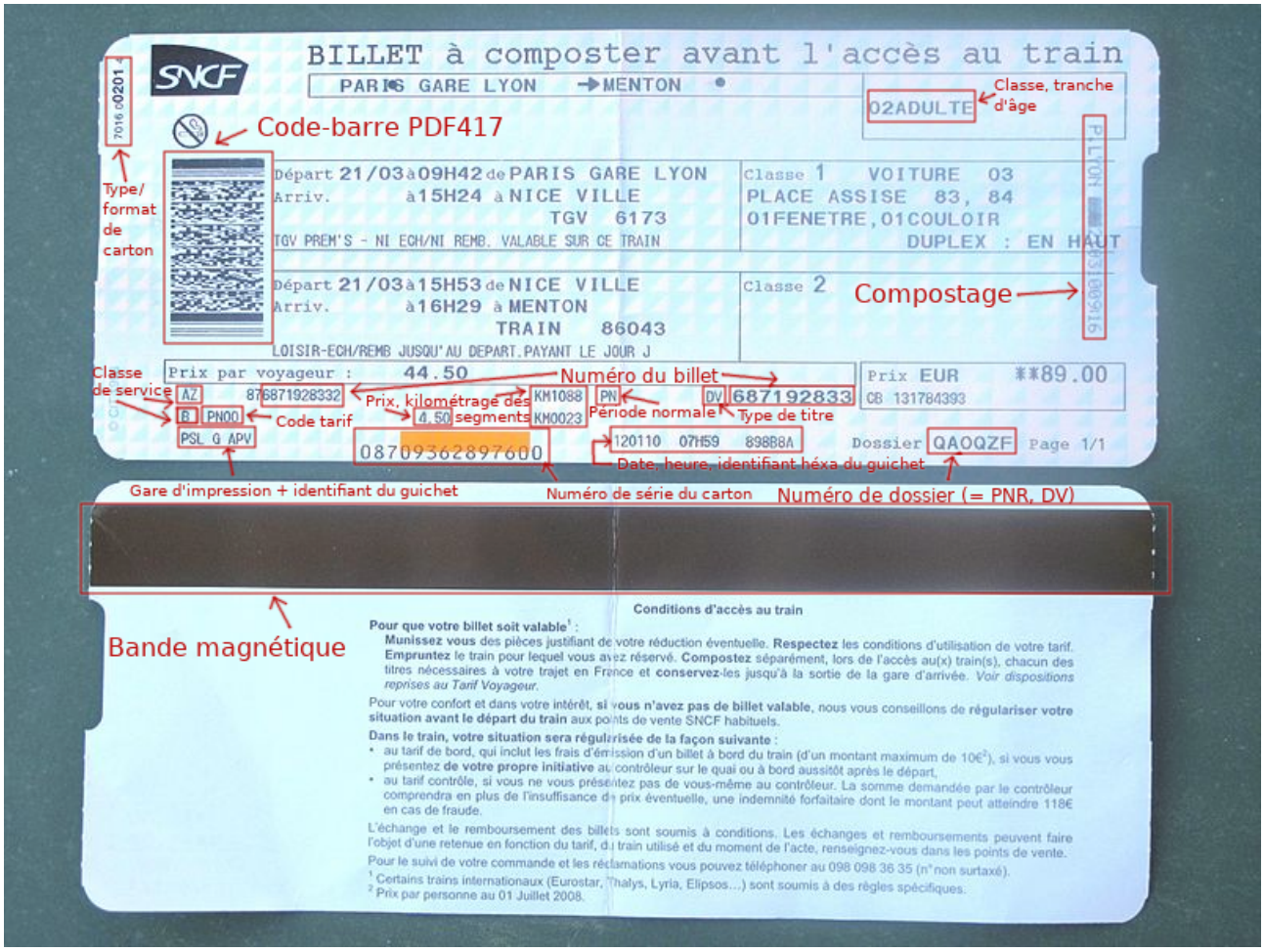


FIGURE 1.3. – Le billet montré plus haut, annoté notamment grâce à l'article «Titre de transports: billets papier» de l'excellent blog de Quentin Godfroy , que je vous invite à consulter pour une annotation plus détaillée.

S'il y a par exemple deux classes de service, c'est qu'elles sont chacune sur la ligne d'un des segments du voyage, le voyage en question étant composé de deux segments (Paris-Nice et Nice-Menton, il y a au maximum deux segments par billet). On peut constater par là même que les informations sont formatées différemment pour le segment TGV et le segment TER.

Le PNR («Passenger Name Record »), un identifiant de six lettres pour les trajets grandes lignes, couplé à votre nom, permet de récupérer ce que la SNCF appelle votre DV («Dossier Voyage») qui contient toutes les informations relatives à votre voyage, dont votre billet. Saisir votre référence PNR sur le site ou sur l'application de la SNCF permet d'ailleurs de récupérer une copie de votre billet, à montrer sur votre téléphone ou à imprimer, et les saisir sur le portail WiFi des TGV permet de vous identifier et d'accéder à Internet.

1.1. Le code-barre PDF417

Si le type de code-barre 2D que vous avez l'occasion de croiser le plus souvent dans la rue est le QR Code , le format utilisé par les billets cartonnés pour être lisibles à la machine est le PDF417 . Ce format, créé par les Américains au début des années 1990, normalisé par l'ISO

et puis adopté par l'IATA, comme je l'ai mentionné plus haut, se trouve en haut à gauche du billet.

Très compact, il se lit très bien au laser, mais se prend très mal en photo (au contraire du QR Code). Comme la plupart des codes-barres 2D, il dispose de plusieurs types d'encodages, de sous-sections visuelles, et de mécanismes de détection d'erreurs avec correction (et il est complexe , c'est pourquoi nous ne nous attarderons pas trop sur son décodage).

Je me suis amusé à scanner une dizaine de billets cartonnés afin de les comparer. N'ayant trouvé aucun outil libre permettant de décoder des photos de mes PDF417 avec la définition de mes scans, j'ai utilisé cet outil en ligne (le nombre de scans par adresse IP est limité), puis j'ai soigneusement rangé les résultats du décodage dans un fichier JSON à l'aide d'un script de ma conception.

Voici les champs contenus dans un code-barre de billet cartonné classique, d'après le document TAP TSI B.6 cité plus haut:

A list of all the elements to be integrated in the PDF-417 barcode:

(All elements are in **Alphanumerical** format (A. .Z,0. .9))

	Element_name	Size	Element_Description
Decoding Info	ID_format	1	Defines type of barcode/ticket/key etc... - Default value="e"
	Code pectab	1	Code, used for ATB-printers - Default value="R"
	Ticket code	2	Code, indicating what kind of ticket is in the barcode 6
	PNRReference of the booking		
	TCN-code	9	Issue booking number
	Specimen-flag	1	1=real ticket, 0=specimen
	Barcode Version		For decryption purposes - which elements can be found
	Number	1	where in which format
	Sequence number	2	xy: ticket x out of y tickets
	Non-used digits	10	for future use
Ticket Info	Traveler type	2	frequent traveler / ...
	Number of adults	2	00 - 99
	Number of childs	2	00 - 99
	Year (last digit)	1	e.g. 2007 -> '7'
	Emission day	3	Sequence number (1/1=1, 2/1=2, ..)
	Begin validity day	3	Sequence number (1/1=1, 2/1=2, ..)
	End validity day	3	Sequence number (1/1=1, 2/1=2, ..)
Segment 1	Departure station	5	5 digit Alphanumerical encoding e.g. FRPNO
	Arrival station	5	5 digit Alphanumerical encoding
	Train number	6	6 characters (or 5 + 1 blanc)
	Security code	4	Specific code for a train - antifraud
	Departure date	3	Sequence number (1/1=1, 2/1=2, ...)
	Coach number	3	Alphanumerical - 3 digits
	Seat/bed number	3	Alphanumerical - 3 digits
	Class of transport	1	1 =first class, 2=second class
	Tariff code	4	4 blancs = full fare ticket
	Class of service	2	defining extra services or conditions (non exchangable, ...)

Segment 2	Departure station	5	5 digit Alphanumerical encoding e.g. FRPNO
	Arrival station	5	5 digit Alphanumerical encoding
	Train number	6	6 characters (or 5 + 1 blanc)
	Security code	4	Specific code for a train - antifraud
	Departure date	3	Sequence number (1/1=1, 2/1=2, ...)
	Coach number	3	Alphanumerical - 3 digits
	Seat/bed number	3	Alphanumerical - 3 digits
	Class of transport	1	1 =first class, 2=second class
	Tariff code	4	4 blancs = full fare ticket
	Class of service	2	defining extra services or conditions (non exchangable, ...)
	TOTAL SIZE	85	(Characters) for a one-segment trip
	TOTAL SIZE	121	(Characters) for a two-segment trip

1. Le billet cartonné «classique»

Voici le contenu décodé du billet recto-verso que nous avons pris pour exemple plus haut:

```
eEDVQAOQZF6871928331111100000000000  02000012      FRPLYFRNIC06173
 9982080003083184  AZFRNICFRXMT86043     080      2PN00B
```

Nous allons maintenant utiliser un script Python sommaire qui permettra de le décoder, d'après les instructions de la SNCF:

Contenu masqué n°1

Voici les résultats produits par ce script:

```
ID_format => 'e'
Code pectab => 'E'
Ticket code => 'DV'
PNR => 'QAOQZF'
TCN-code => '687192833'
Specimen-flag => '1'
Barcode Version Number => '1'
Sequence number => '11'
Non-used digits => '0000000000'
Traveler type => '  '
Number of adults => '02'
Number of childs => '00'
Year (last digit) => '0'
Emission day => '012'
Begin validity day => '   '
End validity day => '   '
Departure station 1 => 'FRPLY'
Arrival station 1 => 'FRNIC'
Train number 1 => '06173 '
Security code 1 => '9982'
Departure date 1 => '080'
Coach number 1 => '003'
Seat/bed number 1 => '083'
Class of travel 1 => '1'
Tariff code 1 => '84  '
Class of service 1 => 'AZ'
Departure station 2 => 'FRNIC'
Arrival station 2 => 'FRXMT'
Train number 2 => '86043 '
Security code 2 => '    '
Departure date 2 => '080'
Coach number 2 => '   '
Seat/bed number 2 => '   '
Class of travel 2 => '2'
```

```
Tariff code 2 => 'PN00'
Class of service 2 => 'B '
```

1.1.0.1. Que remarquons-nous ?

> Une description plus détaillée du contenu décodé est disponible dans l'article «Codes barres et billets SNCF» du blog de Quentin Godfroy , que je vous invite également à lire.

Nous avons donc en premier lieu des codes gares sous la forme de «FRXMT» pour Menton ou «FRPLY» pour Paris-Lyon qui semblent n'être utilisés que par la SNCF (il semble néanmoins déjà y avoir eu des tentatives de compilation - attention, il existe plusieurs référentiels de codes gares au sein même de la SNCF).

Nous remarquons aussi ceci:

— Le texte désigne le numéro du billet par l'acronyme TCN: «Ticket Control Number»

— Il y a un champ «Security code» qui est la (courte) somme de contrôle propriétaire de quatre chiffres pour chaque segment de trajet du billet, et qui fait que vous n'avez probablement pas envie de frauder!

— La première lettre du contenu décodé est importante pour savoir si on a scanné un billet classique, un e-billet ou autre. La deuxième lettre, dans le cas du billet classique, est un format d'impression et la description du document TAP TSI B.6 fait directement référence aux normes PECTAB et ATB de l'IATA que nous avons vues plus haut.

— Pêle-mêle, on trouve un dernier chiffre de l'année, un numéro de jour au sein de l'année, une période de validité, le fait qu'il s'agisse de la page «1/1» (donc «11»), et d'autres informations plus classiques que l'on trouve aussi imprimées sur le papier.

Y a-t-il d'autres gabarits de billets cartons? Oui, en plus du billet IATA, on trouve également le terme de billet ISO, qui désigne les billets sans bande magnétique émis par les distributeurs régionaux[8]footnote:5:

4. [9]footnote:4

Les Tarifs voyageurs - mars 2017 - SNCF (PDF)

5. [10]footnote:5 Il s'agit probablement d'une référence à la norme ISO/CEI 7810 , qui définit les dimensions des cartes d'identité, cartes bancaires, etc.

FIGURE 1.4. – Billet émis par un DBR (Distributeur de Billets Régionaux), au «format ISO». *Source: Wikimedia Commons / Domaine public*

FIGURE 1.5. – Avant le système actuel, un billet «informatisé» émis par l'ancien système de réservation SNCF (1980). *Source: Wikimedia Commons / CC-BY-SA Wuyouyuan*

FIGURE 1.6. – Encore plus ancien: le billet au format Edmonson (1979). *Source: Wikimedia Commons / CC-BY-SA Wuyouyuan*

2. Le billet à imprimer chez vous : l'e-billet

Au cours des années 2000, la SNCF a progressivement déployé la vente de billets en ligne. Le «e-billet» dispose de conditions d'usage différentes du billet classique (il est nominatif et incessible). Depuis 2019, il s'agit du type de billet vendu presque systématiquement sur les trajets grandes lignes. Il n'a pas à être composté, son code-barre peut être présenté indifféremment sur papier, smartphone ou billet cartonné; en effet, il existe aussi sous forme de billet cartonné (tel qu'on peut l'imprimer à une borne):

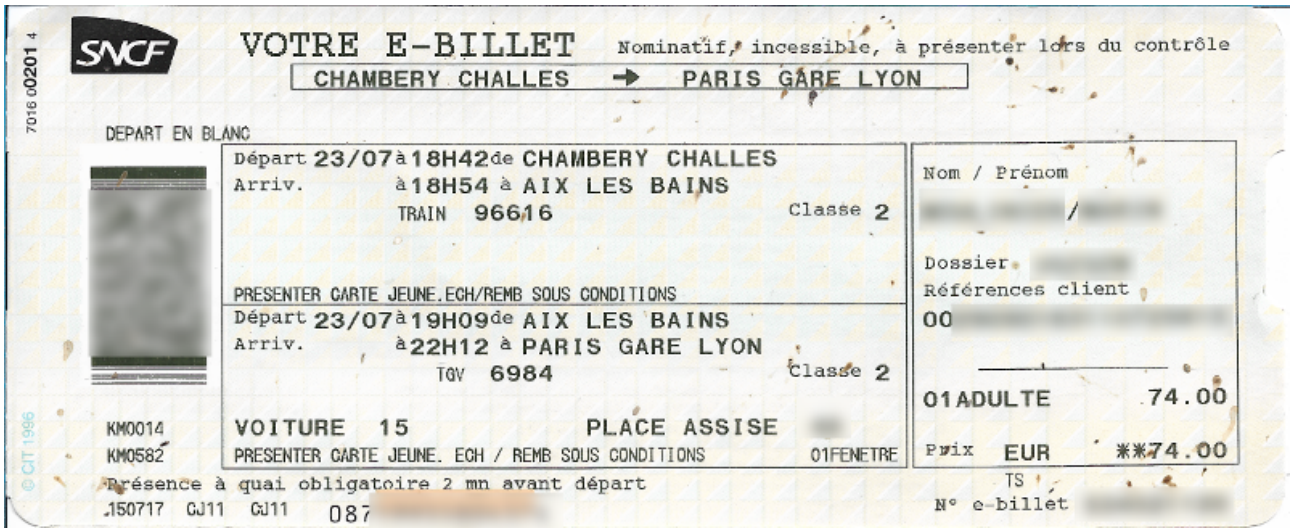


FIGURE 2.7. – Exemple d'e-billet grandes lignes imprimé sur une borne, avec code-barre PDF417.

On remarque qu'il n'est pas exactement formé et mis en page comme le billet classique.

Il semblerait que des versions plus anciennes se conformaient strictement aux exemples de mise en page donnés dans le document TAP TSI B.6:

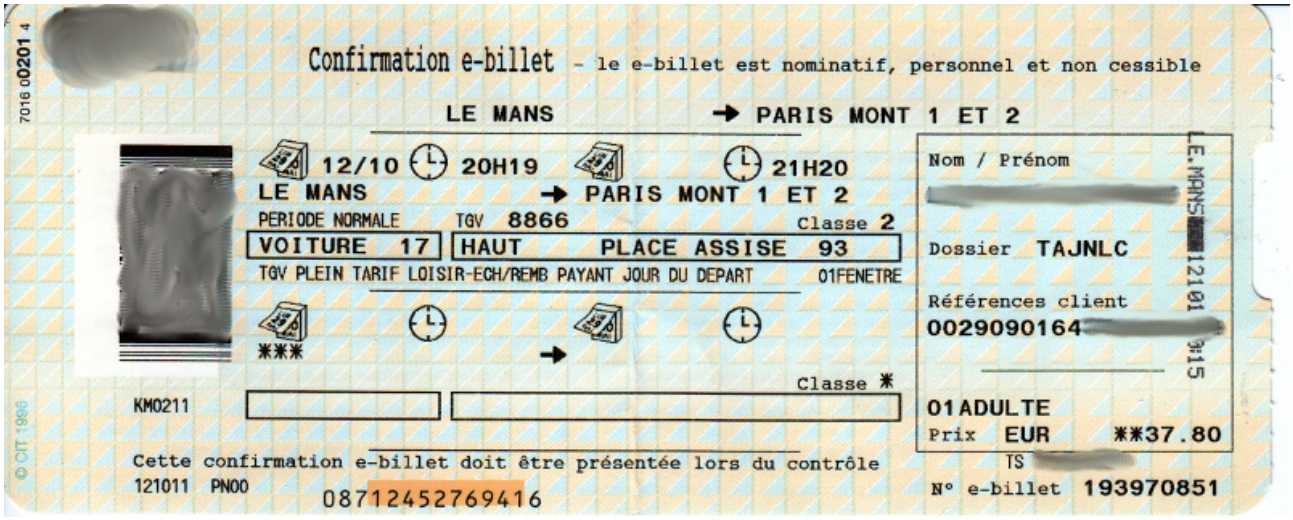


FIGURE 2.8. – *Source: tarification.blogpost.com*

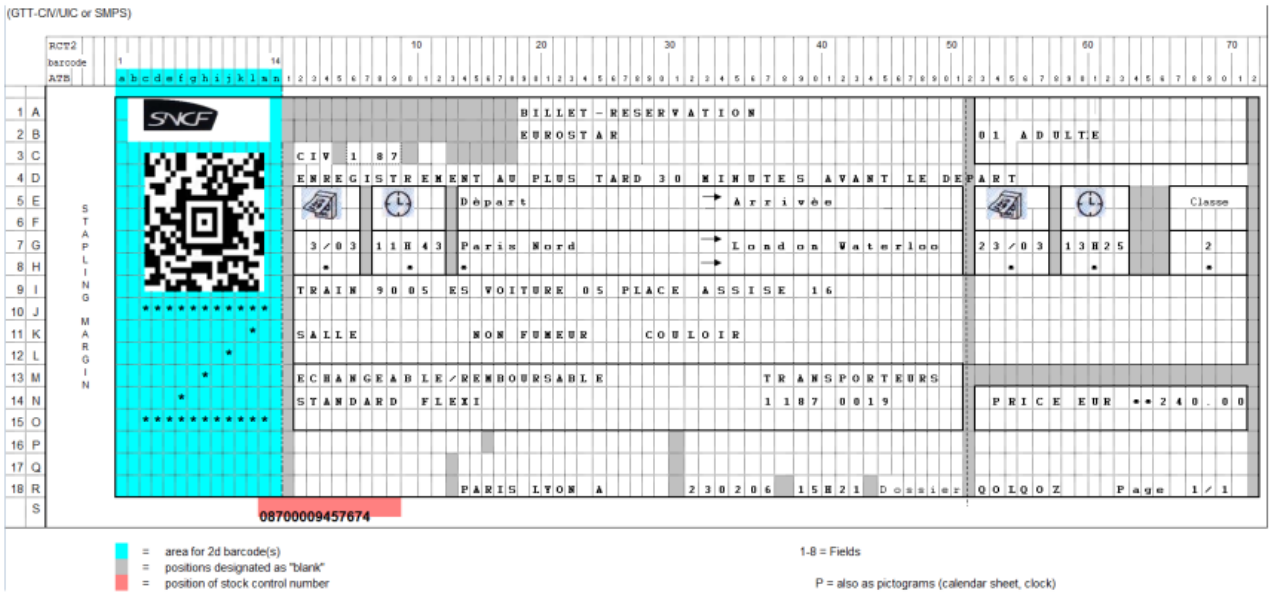


FIGURE 2.9. – Exemple donné en page 125 de la spécification en question.

Bien sûr, l'e-billet apparaît le plus souvent sous sa forme dématérialisée, ou imprimée à l'imprimante grand public, que sous forme cartonnée. Alors, seul le code-barre compte pour le contrôle (ou pour le portique d'accès). Dans ce cas-là, pas de code-barre PDF417, seulement un code Aztec très proche du QR Code sur la forme. Étant délivrés notamment au format PDF, vous en trouverez de nombreux spécimens sur Internet .

Des opération de décodage sur des billets portant sur des trajets semblables permettent d'observer que les données présentes sont à peu près les mêmes entre codes PDF417 et Aztec, néanmoins le format des données pour les e-billets (qui peuvent aussi être encodées en PDF147 quand imprimées sur borne) diffère de celui des billets classiques (elles commenceront par la lettre «i» et utiliseront des champs de tailles différentes[11]footnote:1). Un code Aztec se décode facilement avec le logiciel Zxing (pensez à laisser un peu de blanc autour).

Le format des e-billets TER seul (dont des spécimens sont aussi sur Internet) n'est pas non plus celui des e-billets grandes lignes. Les e-billets TER seuls comprennent notamment un code Aztec beaucoup plus gros, et pour cause, il y a une longue signature binaire avant les données texte! Comme relevé dans cet article , le code-barre d'un e-billet contient systématiquement votre date de naissance.

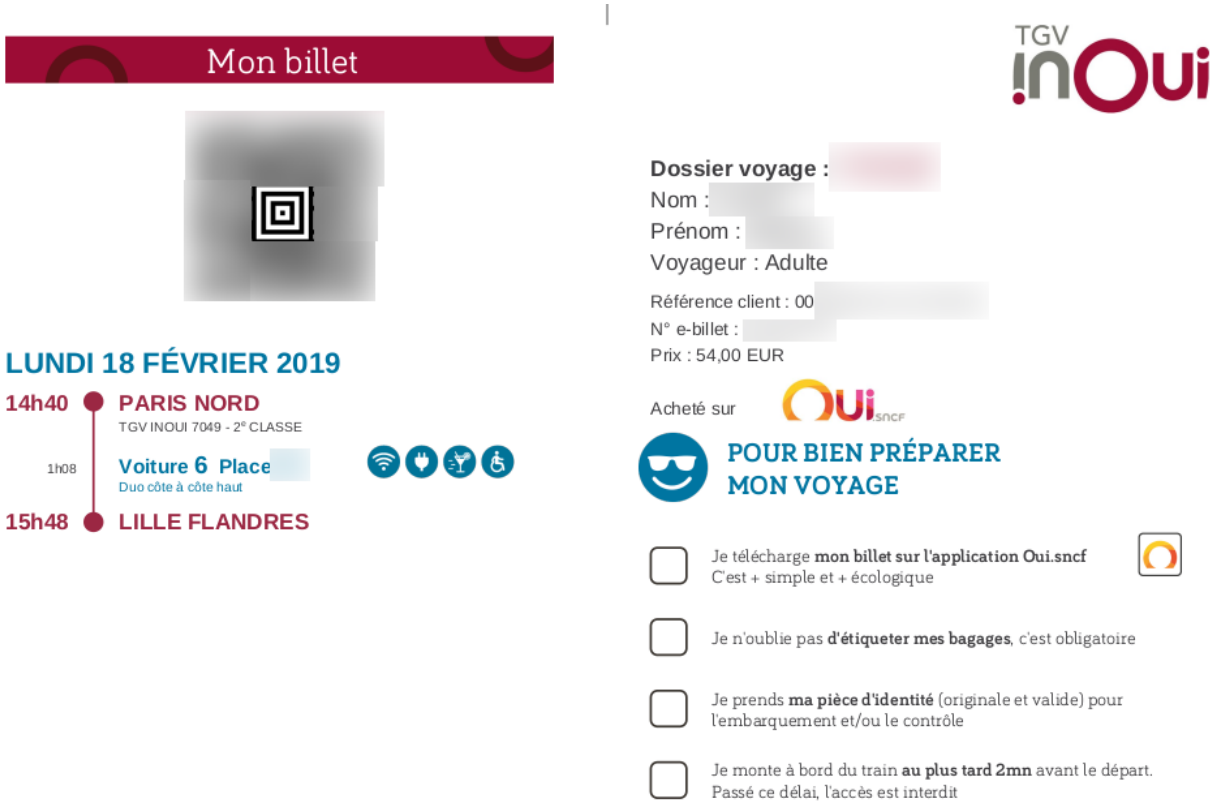


FIGURE 2.10. – Un spécimen d'e-billet grandes lignes au format A4 (la partie inférieure de la feuille est informationnelle ou publicitaire).

1. [12]footnote:1 «Codes barres et billets SNCF», section «Confirmation e-billet», blog de Quentin Godfroy, 22 juin 2012

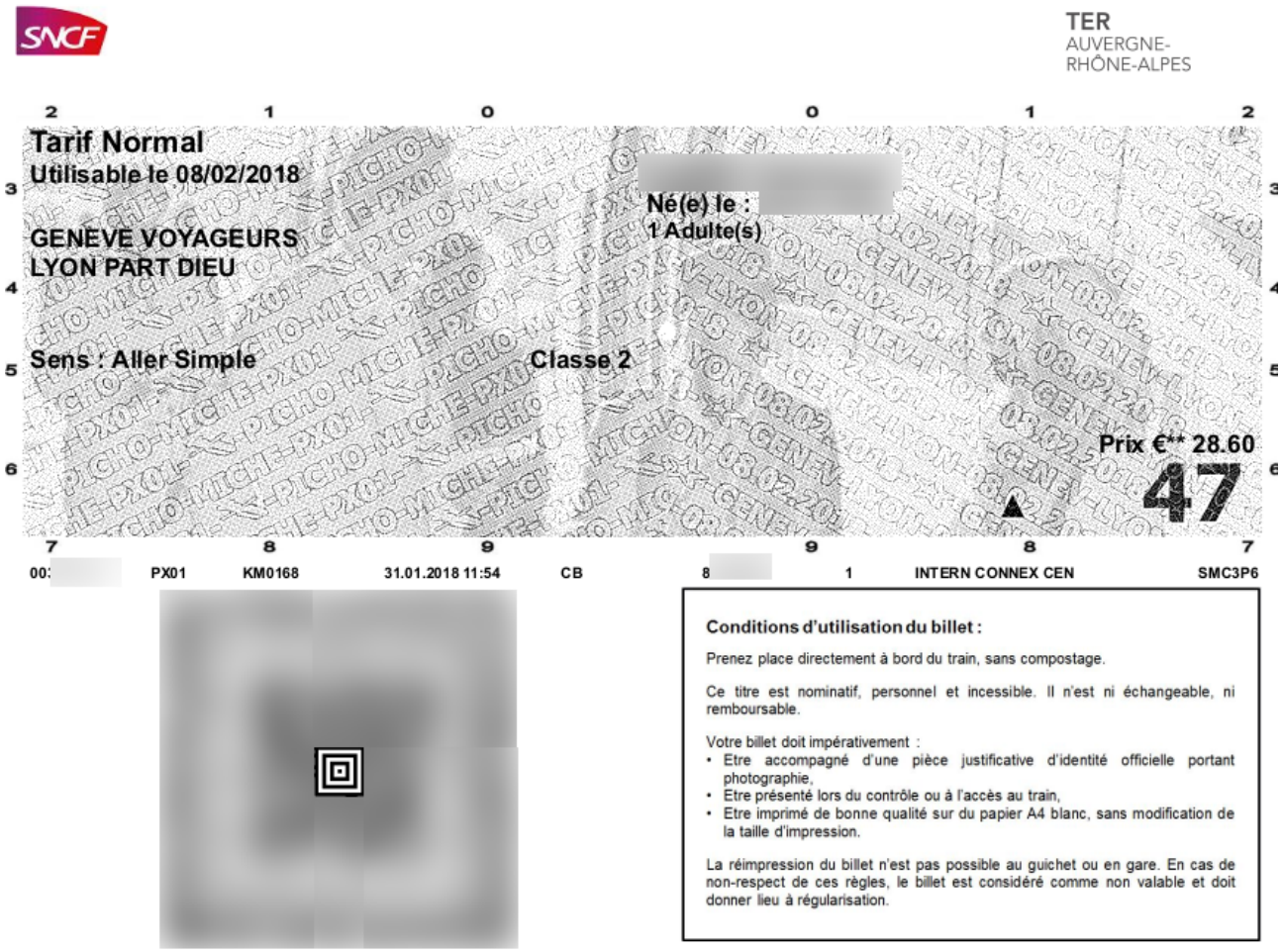


FIGURE 2.11. – Spécimen d'e-billet TER, qui dispose par ailleurs d'un filigrane.

3. Le ticket de métro : la bande magnétique

Contrairement au billet de train dont le principal moyen de validation est le code-barre 2D, le ticket de métro ou de RER, plus compact, est validé grâce à sa bande magnétique .

La bande magnétique, qui est le support utilisé par les cassettes audio, les VHS ou encore pour stocker les premiers programmes informatiques, a été inventée en 1928 (soit bien après le métro, mais dites-vous bien que ça devait être la blockchain de l'époque !). Son principe revient à poser des fines particules métalliques sur une bande plastifiée, dont on va changer le sens d'aimantation (la polarité) pour coder un 0 ou bien un 1.

FIGURE 3.12. – Comme à l'époque!

Peu capacitaire (pas trop la place pour mettre un long code de sécurité), peu résistante (si vous les laissez des mois dans votre portefeuille à côté d'objets métalliques, les tickets de métro peuvent avoir tendance à se démagnétiser et à devenir illisibles !), adossée à un support polluant qu'on jette un peu trop sur la voie publique, elle n'a pas le vent en poupe et les grandes métropoles européennes comptent pour beaucoup supprimer le ticket de métro d'ici quelques années, pour les remplacer par des cartes à puces (évoquées plus bas) à usage occasionnel et temporaire.

En dehors du domaine des transports, lente à lire mais peu coûteuse, la bande magnétique reste utilisée pour stocker de grands volumes de données dont on a peu besoin, comme des sauvegardes. C'est ce que font par exemple certains hôpitaux, ou encore des services comme Amazon Glacier .

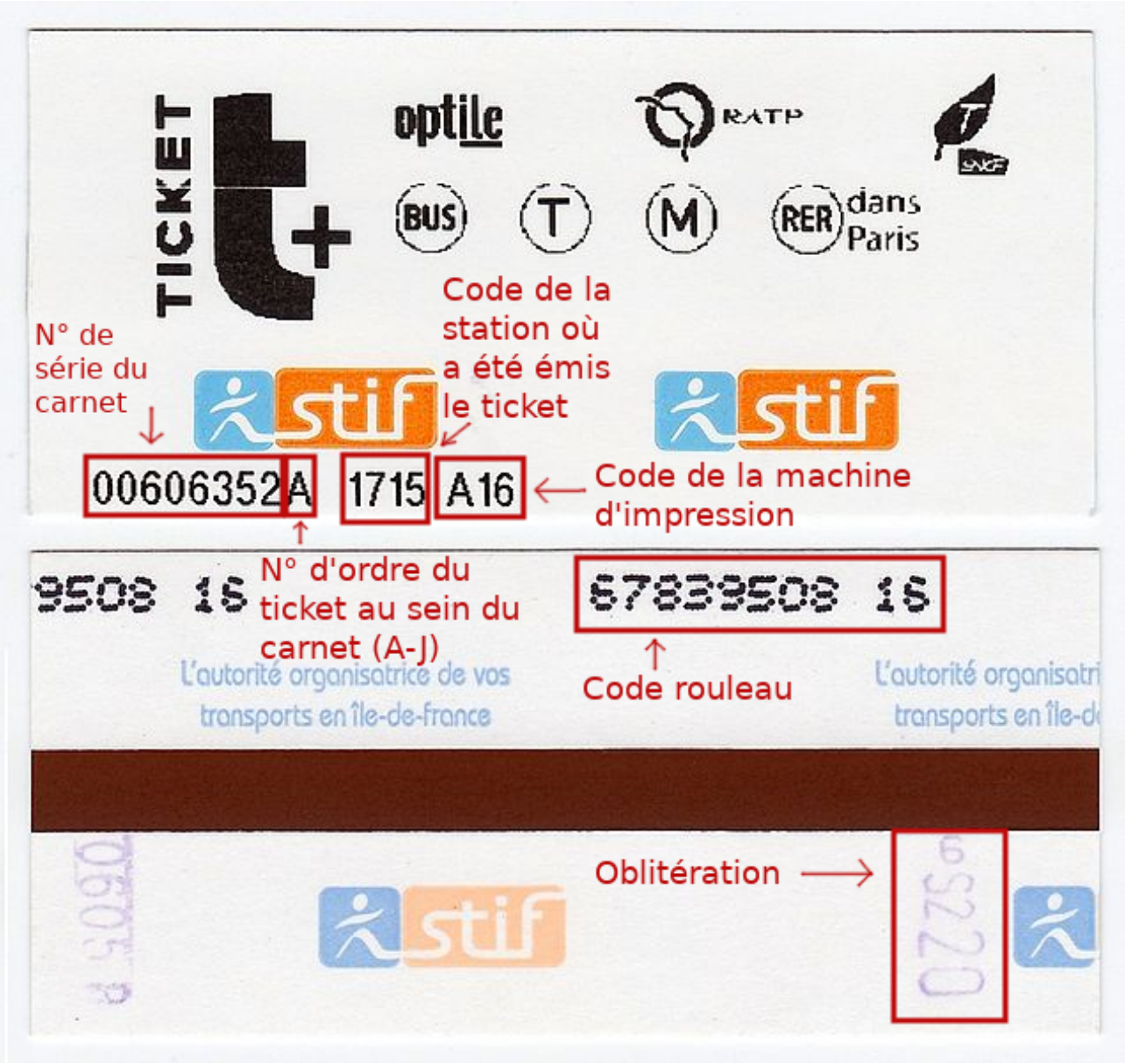


FIGURE 3.13. – Vous trouverez ci-dessus un ticket de métro francilien annoté par mes soins. Pour plus de détails, je vous invite à consulter la page dédiée du site Symbioz .

Concernant la bande magnétique, les données sont stockées en utilisant un codage F2F (fréquence/double-fréquence) : cela signifie qu'on écrit une alternance de «0» et de «1» magnétiques sur la bande en boucle, mais qu'un «bit 1» d'origine correspondra à une alternance deux fois plus courte qu'un «bit 0» d'origine, par exemple.

Cet article de Damien Cauquil explique comment il est possible de la lire en pervertissant un vieux lecteur de cassettes audio. Si en connaître plus sur le formatage et l'encodage des données vous intéresse, je vous en propose une synthèse dans le bloc dépliant ci-dessous.

Contenu masqué n°2

4. La carte à puce : la technologie Calypso

Le billet de train papier est, en théorie, protégé de la fraude par le fait que sa signature cryptographique empêche sa falsification, et contre la réutilisation par le fait qu'il n'est valable que pour un trajet donné (comme le billet TGV) ou sur une période limitée (de 1 à 7 jours pour le billet TER, toujours 1 jour pour l'e-billet).

Quand il ne s'agit pas d'un e-billet, il doit toujours être composté (opération censée être irréversible de détérioration du support papier), et poinçonné lors de son éventuel contrôle par le contrôleur. Un appareil de validation peut aussi se souvenir d'avoir vu plusieurs fois le même billet.

Mais un abonnement est censé être utilisable un nombre illimité de fois. Comment faire pour éviter que différents usagers ne se le partagent? Il y a bien le fait qu'il soit nominatif, mais un valideur électronique ne vérifie ni votre visage ni une carte d'identité. Si on utilisait un simple code-barre comme on le fait pour les billets individuels, il serait trop facile de le copier.

4.1. La carte à puce, une invention française

Qu'ont en commun votre carte bleue, votre carte de transports, votre badge de chambre d'hôtel, votre carte vitale, et votre carte de décodeur TV, sans oublier votre carte SIM? Il s'agit de systèmes semblables, entourés de bouts de plastiques plus ou moins larges, des cartes à puces.

La carte à puce est un système électronique aux circuits minuscules. Il est alimenté par le contact avec le système qui effectue la lecture (ou par le contact proche, dans le cas du système sans-contact). Elle dispose d'un microprocesseur très lent et d'une petite mémoire (mis ensemble, on appelle ça un microcontrôleur). Souvent, elle dispose de son problème système d'exploitation très basique qui sait interpréter des applications utilisant une version simplifiée du bytecode Java, implémentée matériellement.

La carte à puce dispose d'**une** grande caractéristique: l'unicité. Une information dite «secrète» qui est stockée dans une carte à puce ne doit pas en sortir. Elle ne doit donc pas être duplicable ni être communiquée directement.

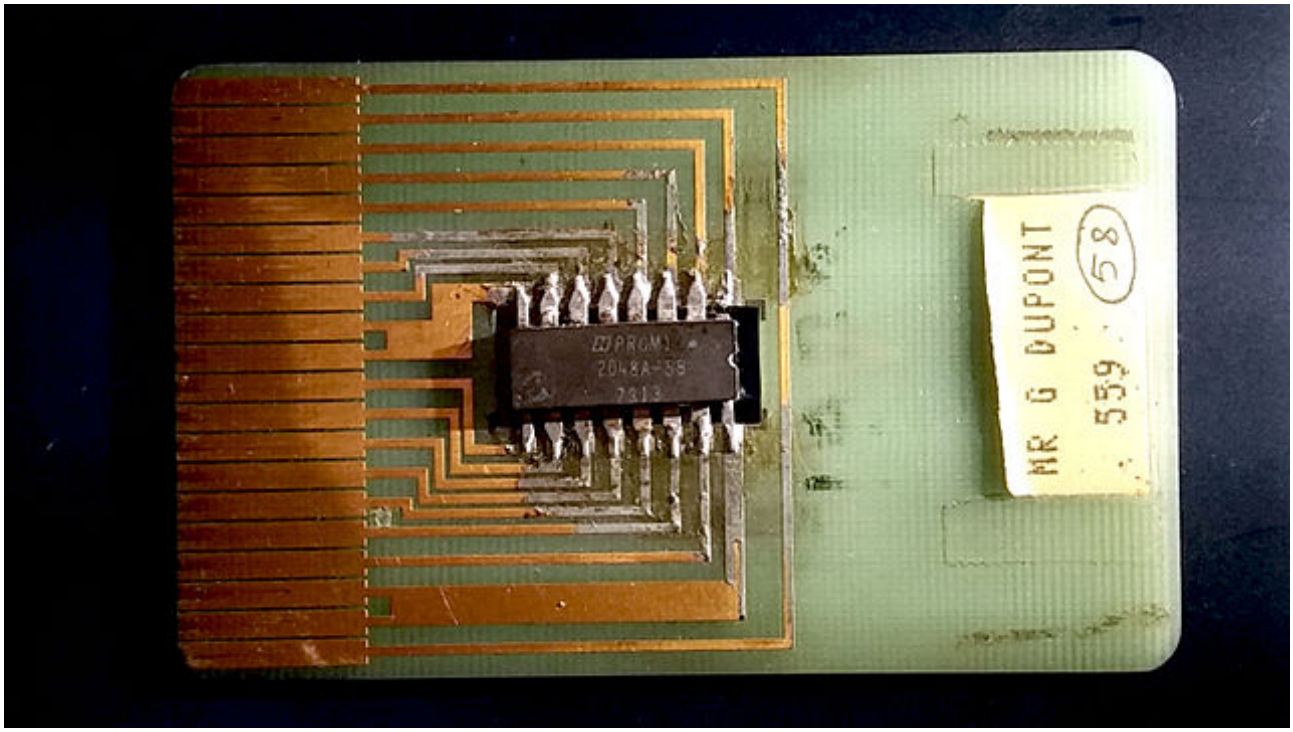


FIGURE 4.14. – Un des premiers prototypes de carte à puce, avant sa miniaturisation. (Wikipédia)

Tout ce que la carte à puce doit savoir faire, c'est fournir une preuve cryptographique qu'elle dispose d'une information. Elle peut communiquer des informations publiques, comme le fait que vous disposez d'un abonnement ou les cinq dernières stations où vous l'avez validé, sans restrictions, mais elle doit pouvoir fournir au valideur une preuve qu'elle est légitime à transmettre une information: cette information sera accompagnée d'un dérivé cryptographique (hash) composé à la fois de l'information publique, et de la clef secrète de votre carte à puce qui est unique. La clef secrète, connue seulement par votre carte et par le module de sécurité du valideur, est imprimée à la fabrication et n'est jamais communiquée directement.

Cela marche aussi dans l'autre sens: lorsque le distributeur de billets charge un abonnement sur votre carte, il prouve qu'il est légitime à charger cette information en utilisant le même mécanisme. Ainsi, l'abonnement ne sera chargé qu'à un seul endroit.

Le gros enjeu de la carte à puce, c'est donc de ne pas laisser fuiter la clef secrète pour ne pas pouvoir être dupliquée. Cela a conduit, depuis son invention dans les années 60, à des jeux du chat et de la souris où une guerre a été menée durant des années. Notamment, pour casser les cartes des décodeurs TV (il fallait insérer, dans votre décodeur, une carte de droits TV que vous receviez par la poste et qui était renouvelée régulièrement). Cela se fait de moins en moins depuis qu'on utilise des cryptoprocesseurs directement intégrés aux décodeurs.

De nombreux moyens ont été employés dans ce jeu du chat et de la souris: microscopes très précis et très chers, attaques par faute (techniques visant, par exemple, à introduire une variation microscopique dans l'alimentation électrique dans l'espoir d'inverser un bit) en sont les moyens les plus courants, qui ont tous connu diverses contre-mesures.

> Vous l'avez compris, des secteurs entiers de l'économie reposent sur la carte à puce. Si les cartes à puces sont cassées, les abonnements de transports peuvent être dupliqués à l'infini, les abonnements de téléphonie aussi au détail qu'ils sont plus détectables (selon les conditions), les cartes bleues peuvent éventuellement être dupliquées et un paiement réémis, bref tout s'écroule. Des économies parallèles peuvent se développer très vite, comme on l'a vu avec les décodeurs TV pirates. Gemalto, un gros fabricant de cartes à puces français, a lui-même fait partie du CAC 40.

En ce qui concerne les transports en particulier, il existe deux grands systèmes répandus:

— Les cartes basées sur la technologie MIFARE, créée à l'origine par la société néerlandaise Philips (maintenant NXP Semiconductors), très répandues dans divers pays et notamment anglo-saxons (l'Oyster Card londonienne, etc.). Les cartes basées sur MIFARE ont connu divers soucis de sécurité dus notamment à une cryptographie faible (je parle de celle utilisée dans le cadre des réseaux de transports, et pas de celle utilisée probablement par votre badge d'immeuble qui est une variante nommée MIFARE Classic et ne dispose pratiquement d'aucune protection contre la copie).

— Les cartes basées sur la technologie Calypso, plus récente, créée par la RATP et un conglomérat d'industriels. Elle est pour sa part réputée comme étant éprouvée et utilisant des procédés technologiques solides. Elle est également utilisée par une grande majorité de villes et de régions (passes Navigo, Técély, OùRA, Pass'Pass, KorriGo…) françaises. Le passe Navigo succède à la carte Orange, qui était basée sur une simple bande magnétique, un support peu sécurisé, et qui revient dans le meilleur des cas à utiliser un code-barre lisible à distance.

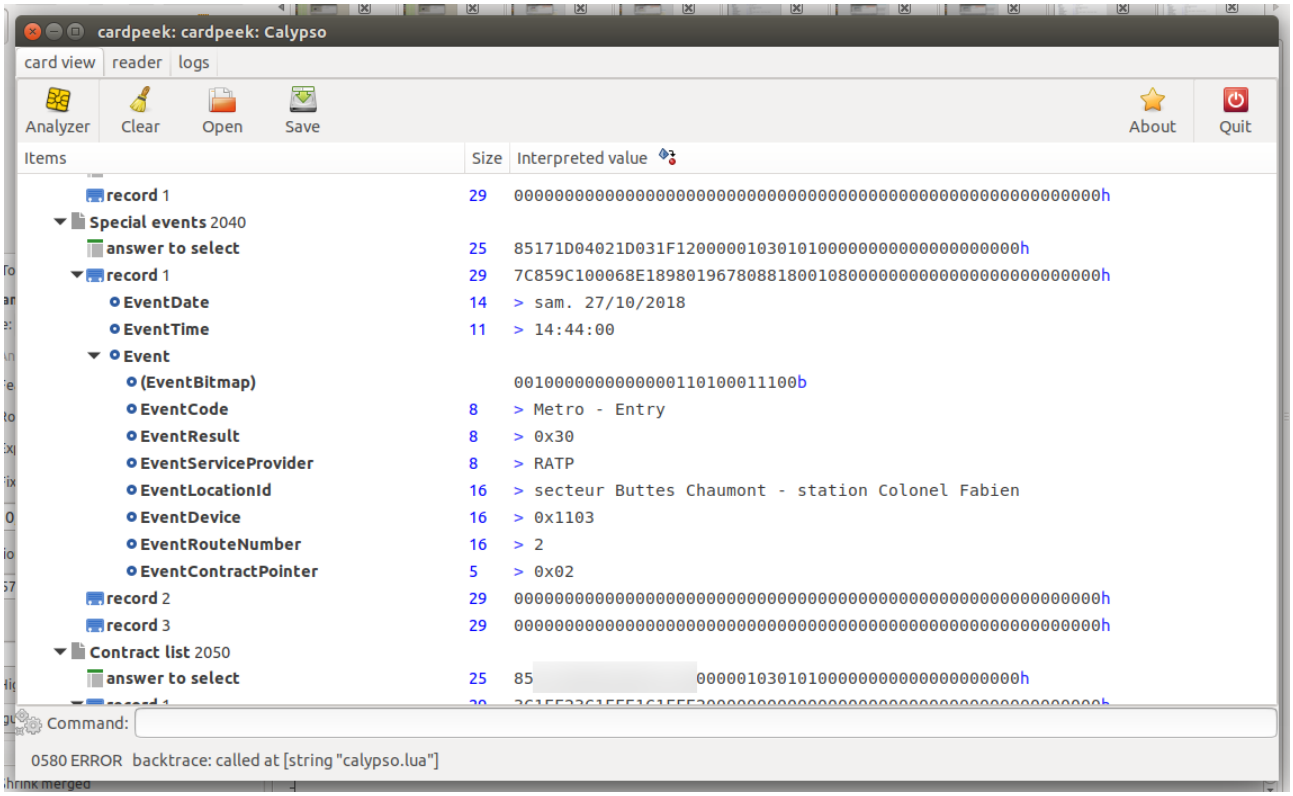


FIGURE 4.15. – Les informations publiques présentes sur l'essentiel des cartes à puces du marché (de la carte Vitale à la carte SIM, ici un validation contenu dans une passe Navigo) peuvent être lues à l'aide du logiciel libre Cardpeek (pas la clef secrète, bien entendu !), sur tout ordinateur équipé d'un lecteur de cartes à puces.

Typiquement, sur une carte de transports, vous trouverez de stocké: vos dernières validations, tous vos titres (billets, tickets, abonnements, abonnements vélo libre-service), votre nom et date de naissance, votre numéro de contrat, votre photo, la date d'expiration de votre carte, et bien sûr le pays et l'identifiant de votre opérateur de transports. En présence de plusieurs opérateurs de transports dans votre région, il existe une hiérarchie de clefs publiques/privées leur permettant d'émettre ou de valider des titres.

Pourquoi valider votre passe même si vous avez un abonnement en cours de validité? D'abord parce que votre opérateur de transports doit mesurer le nombre de validations pour pouvoir prévoir les flux et organiser les transports dans votre région. Autrement, cela revient à composter.

En théorie, les mécanismes cryptographiques impliqués dans l'implémentation de la carte à puce font qu'un abonnement frauduleux ne doit pas pouvoir être écrit. Il reste qu'un abonnement qui a été contracté et légitiment écrit peut être suspendu, notamment en cas de défaut de paiement , ce qui nécessite que les validateurs puissent connaître certains bons ou mauvais abonnements.

Les valideurs présents dans les bus sont rarement connectés à Internet ou autre réseau; en général, ils sont reliés au réseau lorsque le bus arrive au dépôt , soit toutes les 24 ou 48 heures en pratique. Cela explique que la prise ou la résiliation d'un abonnement ne soit pas forcément immédiate.

Pour l'anecdote, les industriels qui fabriquent les cartes à puces sont appelés les encarteurs (en voilà un joli mot de français !).

4.2. Le rechargement

À défaut de pouvoir analyser ce qui se passe à l'intérieur de la carte à puce (qui est en principe un mini-ordinateur fonctionnant avec du Java), nous pouvons facilement capturer ce qui se passe lorsque nous rechargeons notre passe par Internet.

Lorsque notre passe est rechargé, il est fait des lectures sur les parties publiques de la carte comme le fait Cardpeek, ainsi que des écritures signées cryptographiquement. Ces écritures ne sont pas censées pouvoir être rejouées.

Dans le jargon du monde des cartes à puces, une requête/réponse échangée avec une carte à puce est appelée un APDU (*Application Protocol Data Unit*). Le protocole qui permet de faire communiquer votre PC et votre carte à puce avec un lecteur de cartes à puces standard (par exemple USB) est appelé PC/SC (pour «*PC smartcard*»).

Quand j'ai voulu voir comment ça fonctionnait en 2018, cela fonctionnait de la manière suivante: l'utilisateur télécharge et exécute un programme léger en Java sur son ordinateur (avant, c'était un applet Java directement dans le navigateur mais avec les navigateurs récents, ce n'est plus possible).

Le navigateur va faire une suite de requêtes/réponses HTTP vers le serveur de l'autorité de transports afin d'obtenir des APDU brutes (encodées en hexadécimal) à envoyer à notre carte à puce, lesquelles seront relayées au programme Java via une Websocket en local. Ensuite, le programme Java transmettra les données à la carte à puce de manière tout à fait standard, avant de retransmettre les données dans l'autre sens (la réponse repart vers la Websocket locale puis va, toujours encodée en hexadécimal, dans la prochaine requête envoyée aux serveurs de l'autorité de transports).

Un rechargement en ligne, c'est donc comme un branchement sur un guichet à distance: le lecteur de cartes est de votre côté, mais l'intelligence qui interprète et envoie les APDU brutes est située chez l'autorité de transports.

À noter que les spécifications techniques de Calypso sont partiellement publiques, certaines étant disponibles sur inscription ou réservées aux membres du consortium.

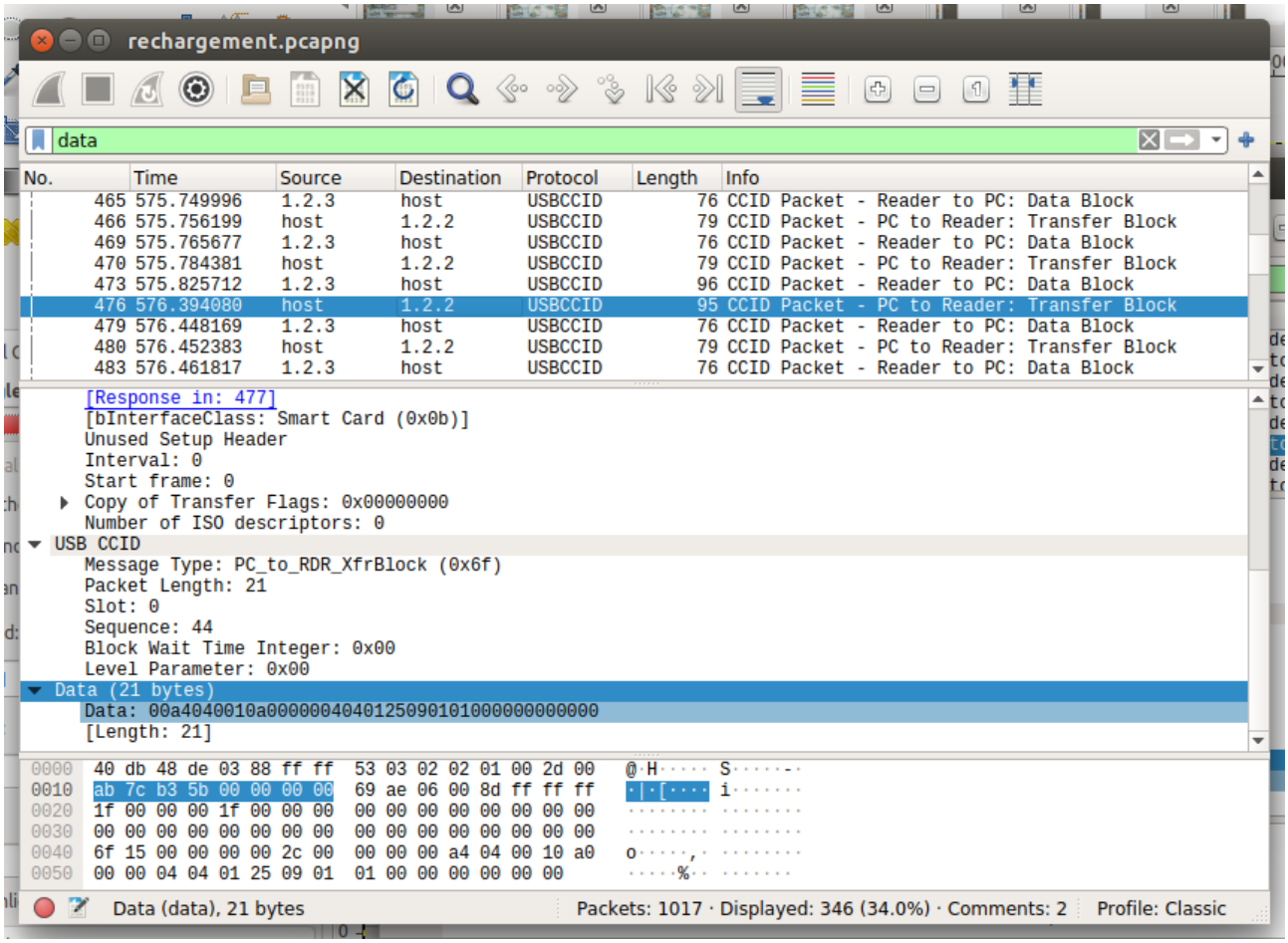


FIGURE 4.16. – Lors du rechargement d'une carte basée sur Calypso, les données échangées sur USB peuvent facilement être capturées avec l'analyseur réseau Wireshark (sous Linux, il faut activer le module de capture USB via la commande `sudo modprobe usbmon`).

FIGURE 4.17. – Extrait d'échanges HTTP issus d'un rechargement de passe Navigo, journalisés au format HAR . Comme on peut le voir, des APDU bruts sont transmis en hexadécimal (et dans la requête HTTP – où se trouvent les APDU réponses de la carte – et dans la réponse HTTP, où se trouvent les APDU requêtes de la borne distante).

4.3. La validation sans contact

Le terme «RFID » désigne un ensemble de normes permettant de créer des puces électriques qui ont la possibilité de s'activer *sans contact*, mais à très courte distance. Ces puces sont appelées «tags», et un «tag passif» est une puce qui a la possibilité de s'activer par la seule puissance des ondes électromagnétiques émises par un lecteur proche, à convertir en courant (le lecteur de tags doit donc émettre un signal puissant pour permettre au tag de s'alimenter).

Développée dans les années 1980, cette technologie est utilisée aussi bien pour les antivols dans les magasins de vêtements (vous passez par des portiques RFID) que pour tous types de carte à puces **sans contact**, ou que pour votre badge d'immeuble.

Le terme «NFC » (*Near-field Communication*) désigne un sous-ensemble de normes RFID, qui peuvent être utilisées sur des distances d'environ 4 cm, en général sur la fréquence de 13,56 MHz, ainsi que l'écosystème protocolaire associé. Une grande partie des smartphones actuels sont compatibles NFC; cela vous permet aussi bien (à certains endroits et avec certains modèles) de charger des tickets sur votre smartphone que, dans certains cas de figure, de cloner un badge d'immeuble individuel (ceux-ci utilisant fortuitement une technologie très proche et étant très peu protégés, au contraire de la carte de transports ou du badge «passe-partout» du facteur).

Attention, il existe plusieurs variantes de NFC. Le protocole le plus fréquemment utilisé pour la radiocommunication se nomme ISO/IEC 14443 , mais les cartes Calypso utilisent une version—légèrement modifiée—de la variante appelée «type B» de cette norme. Tandis que Google ne recommande de supporter sur Android que la variante dite «type A». C'est pourquoi les valideurs Navigo doivent probablement supporter au moins ces deux variantes, type B modifié pour vous permettre de valider avec une carte, et type A pour vous permettre de valider avec un smartphone (comme nous le verrons plus loin).

5. Le titre sur smartphone : l'e-ticket NFC

Néanmoins, une carte de transports doit être amenée jusqu'à un distributeur, ou au mieux un lecteur de cartes pour PC, pour que son contenu puisse être rechargé ou lu, et doit être trimballée lors de vos déplacements, ce qui est peu pratique. En effet, à la fin des années 2010, tout le monde avait un smartphone, et l'industrie des transports s'est dit: pourquoi le ticket de bus ou l'abonnement ne pourrait pas être acheté sur smartphone, payé sur smartphone, et validé en passant le smartphone sur le valideur – une grande partie des smartphones, à cette époque, étant compatibles NFC?

Comment? En fait, il y a deux grands moyens d'effectuer une transaction sans contact, et de stocker les secrets afférents à votre abonnement (je vous rappelle qu'il faut stocker une clef secrète qui ne doit pas sortir de votre téléphone si le téléphone remplace la carte à puce, mais aussi exécuter un morceau de logiciel qui doit pourtant à la fois connaître votre clef secrète ET les titres ou abonnements que l'autorité de transports indique, avec ses clefs, que vous avez légitiment achetés; un téléphone étant par nature bidouillable, il faut un second processeur isolé – un peu comme une carte à puce? – qui ne soit pas accessible directement par le premier):

— Soit utiliser directement la carte SIM, qui est elle-même une carte à puce – donc un mini-ordinateur qui sait exécuter des applications écrites dans un Java simplifié – mais aussi, chez certains opérateurs seulement, un tag NFC passif (votre carte SIM vous permet donc, dans ces cas-là, de stocker et valider votre titre de transport même lorsque votre téléphone est éteint !). Une carte SIM pouvant tout à fait accueillir des applications supplémentaires (appelées «cardlet»), dont le code doit dûment être signé avec les clefs de votre opérateur de téléphonie pour être accepté et installé par votre téléphone bien entendu. Dans cette configuration, l'application du téléphone installe directement le cardlet sur votre carte SIM, puis lorsqu'il faudra charger votre titre la carte SIM communiquera de manière sécurisée avec l'infrastructure distante de l'opérateur de transports (par le biais d'Internet).

— Soit utiliser ce qu'on appelle le cryptoprocesseur du téléphone: selon ses variantes techniques, on peut l'appeler «trustzone» (chez ARM/Qualcomm), «TPM» pour «*Trusted Platform Module*» chez Microsoft ou encore «*Secure Enclave*» chez Apple, ou de manière plus générique, «Secure Element». C'est une puce isolée rattachée au microprocesseur, mais sur laquelle ne peuvent être installées que des applications dûment signées avec les clefs du fondeur du microprocesseur ou du vendeur du téléphone (donc vérifiées minutieusement par celui-ci dans leur bienveillance), qui dispose de sa propre mémoire et qui n'interfère jamais avec les applications présentes sur le processeur principal du téléphone (appelé aussi «AP» pour «*Application Processor*». Le module NFC du téléphone sera alors utilisé pour valider le titre.

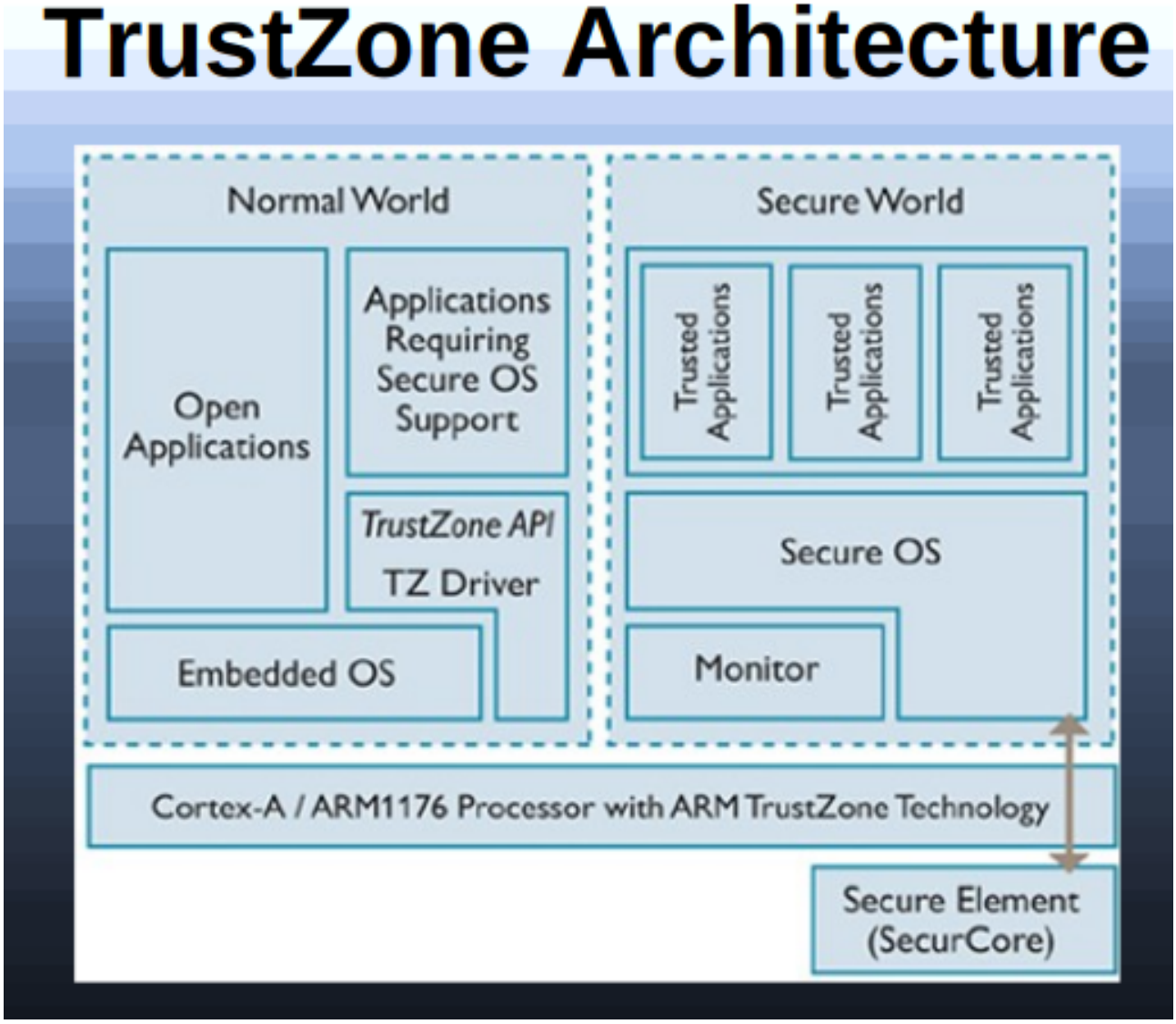


FIGURE 5.18. – Schéma de l'architecture sécurisée d'un processeur ARM (source: ARM).

Avec le service qui existe depuis 2019 en Île-de-France, ou un peu avant à Strasbourg, la solution qui a été choisie repose sur les deux options à la fois: le support billetique peut être soit une carte SIM Orange (seulement Orange, car le développement a été fait avec eux et les cartes SIM des autres opérateurs ne supportent pas toujours NFC), ou la trustzone d'un téléphone compatible NFC Samsung, pas trop ancien non plus.

Les téléphones Apple étant exclus de l'expérimentation, Apple ne voulant pas ouvrir l'interface NFC de ses téléphones, ou pas sans s'arroger d'une commission qui porte probablement sur les transactions effectuées par l'application [13]footnote:1 (ce qui a peu de chances d'être accepté politiquement et économiquement pour plusieurs raisons).

La technologie utilisée pour le ticket dématérialisé dans cette région a été développée par Wizway, une co-entreprise de la RATP, la SNCF, Orange et Gemalto, en collaboration avec DéjàMobile, une start-up caennaise spécialisée dans la billetique mobile.

1. [14]footnote:1 Voir aussi ici et là

5.1. Analyse technique des applications ViaNavigo et Ticket Sans Contact

Acheter et valider son ticket sur smartphone avec cette technologie demande d'installer deux applications: celle de l'opérateur de transports, et celle de Wizway, appelée «Ticket Sans Contact».

Après une rapide analyse, on s'aperçoit que diverses vérifications, somme toute classiques, sont effectuées par l'application de l'opérateur de transports, pour limiter le risque que l'application fonctionne sur un téléphone compromis ou soit analysable trop simplement.

Il s'agit de mesures *anti-tampering* très génériques (détection du root, de l'utilisation d'un émulateur…). Par contre, je ne vous donnerai pas trop de détails car j'irais probablement au-delà des jalons de l'article L122–6-1 du Code de la propriété intellectuelle. Je vais donc me contenter de décrire des mesures assez génériques pour que l'on puisse considérer qu'elles auront des chances correctes d'être en place sur une application que l'on souhaite protéger (que ce soit du bancaire ou autre).

Également, nous sommes en présence de *certificate pinning*, c'est-à-dire que l'empreinte du certificat TLS des serveurs HTTPS du développeur, avec lesquels l'application communique, est écrite en dur dans l'application afin de compliquer l'interception des flux par un tiers.

Ces obstacles à l'analyse de l'application sont néanmoins facilement contournables en modifiant le *bytecode* de l'application. On peut également s'apercevoir que les deux applications vérifient les signatures de leurs fichiers APK, en récupérant par exemple une empreinte de clef publique émise par un serveur distant au démarrage – une vérification qu'il est également possible d'altérer.

L'intention de ces mesures supplémentaires n'est ainsi pas de créer des barrières «physiquement inviolables» comme peut l'être le cryptoprocesseur, mais d'ajouter des obstacles techniquement contournables par un chercheur ou un analyste à l'ingérence technique par une autre application, ou la modification trop simple de l'application par un individu mal intentionné mais peu compétent.
Elles n'empêchent pas le cardlet (ou le *trustlet*, l'équivalent du cardlet au sein de la trustzone), situé à un niveau de système différent de l'application mobile, de communiquer directement ses preuves cryptographiques au serveur distant de l'opérateur de transports, sans que l'application n'ait à les comprendre ou à pouvoir les générer.

Si ces premières mesures sont contournables dans un laps de temps relativement bref par un ingénieur expérimenté, il s'avère également que dans le cas où la carte SIM est utilisée pour support des tickets, celle-ci vérifie si l'application Android qui souhaite communiquer avec le cardlet fraîchement installé est connue et autorisée, encore une fois par l'empreinte du certificat de son producteur (il y a un système de liste blanche sur la carte SIM). Cette dernière contre-mesure empêcher d'utiliser une application «Ticket Sans Contact» qui soit à la fois modifiée et fonctionnelle sur le téléphone sans modifier également certaines des bibliothèques systèmes d'Android, fournies dans le cadre du framework Open Mobile API de la SIMAlliance (un groupement d'industriels qui définit des normes et services en rapport avec les cartes SIM).

Néanmoins, si falsifier l'application qui communique avec le cardlet est dispendieux en temps (le cardlet en lui-même est dûment protégé et signé, «de toute façon»), une analyse plus partielle de l'application peut être effectuée sans contourner toutes ces mesures.

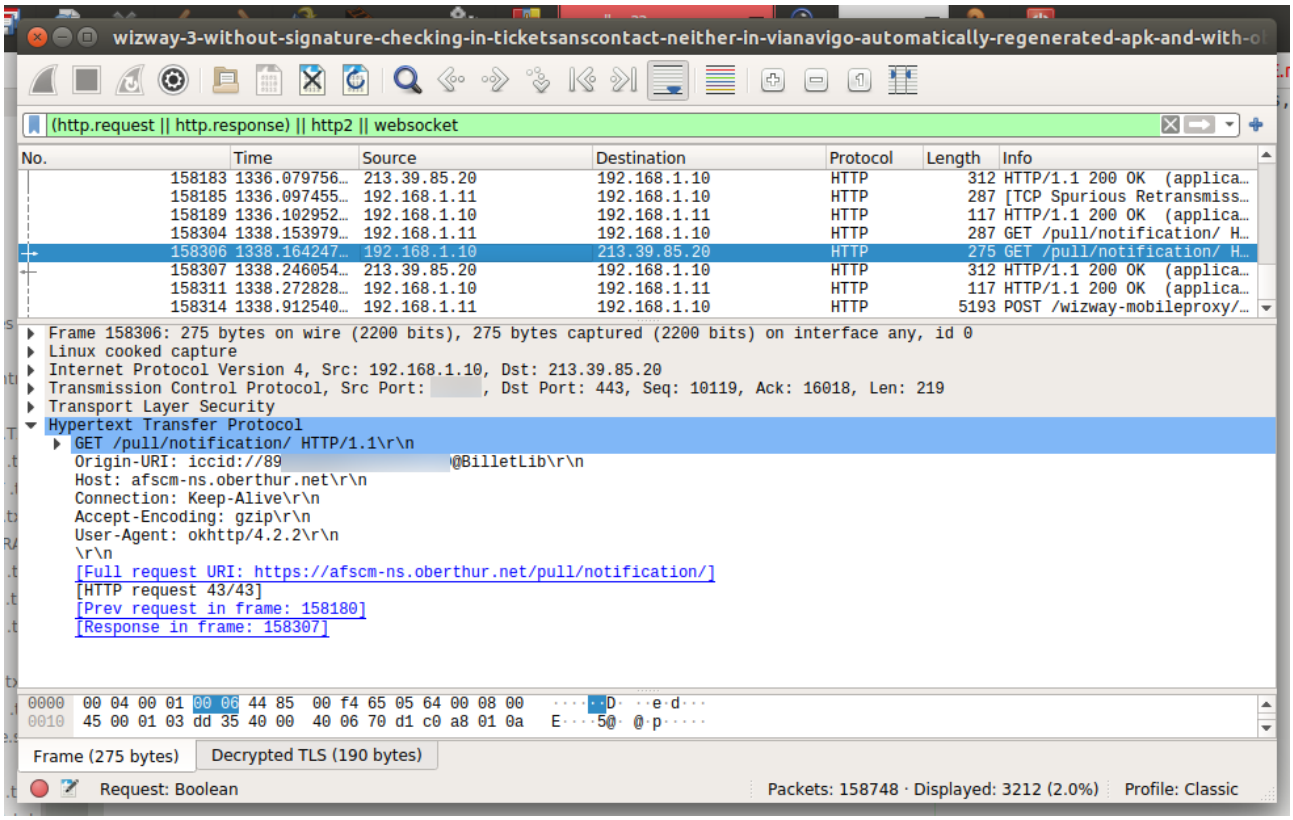


FIGURE 5.19. – Une capture issue de l'application partiellement modifiée. On remarque que l'ICCID (le numéro de série de notre carte SIM), ici floutée est transmise de manière partiellement claire dans chaque requête dirigée vers les serveurs d'Oberthur (la société connue pour éditer le passe Navigo), a priori dans l'optique d'obtenir des notifications – comme de nouveaux titres à charger sur le cardlet?

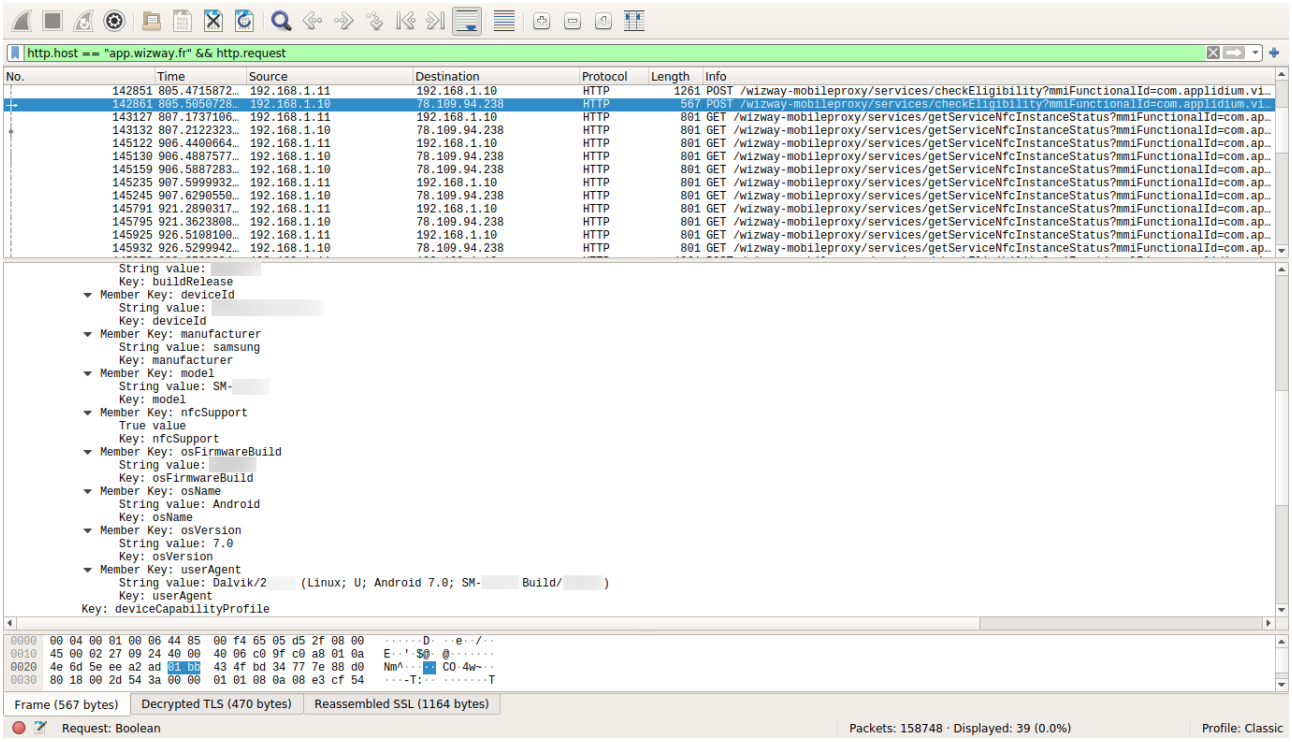


FIGURE 5.20. – Des informations très détaillées sur votre téléphone, dont l'identifiant unique de votre appareil ou toutes informations propres à caractériser sa version, sont envoyées aux serveurs de WizWay. Cela aurait pu être évité.

Conclusion

En conclusion, nous avons vu que les différents titres de transport sont de plus en plus sécurisés, et que la friction pour les utiliser se réduit grâce au post-paiement ou encore la validation sans contact. Et ça tombe bien, les réseaux de transports auront toujours besoin d'être financés (ce que l'on peut aussi faire sans titres de transport ni abonnements, saviez-vous que la ville d'Aubagne a ouvert dès 2014 un tramway entièrement gratuit à l'usage ?)).

Bonne route, et n'oubliez pas que votre billet doit être composté, ou votre carte OùRA, validée avant de monter à bord!

Contenu masqué

Contenu masqué n°1

```
#!/usr/bin/python3
#-*- encoding: Utf-8 -*-
from io import StringIO


```

Contenu masqué

```
class TicketField:

    def __init__(self, name, size, description):

        self.name = name
        self.size = size
        self.description = description

fields = [
    TicketField('ID_format', 1, 
        'Defines type of barcode/ticket/key etc... - Default value="e"'
        ),
    TicketField('Code pectab', 1,
        'Code, used for ATB-printers - Default value="R"'),
    TicketField('Ticket code', 2,
        'Code, indicating what kind of ticket is in the barcode 6'
        ),
    TicketField('PNR', 6, 'Reference of the booking'),
    TicketField('TCN-code', 9, 'Issue booking number'),
    TicketField('Specimen-flag', 1, '1=real ticket, 0=specimen'),
    TicketField('Barcode Version Number', 1, 
        'For decryption purposes - which elements can be found where in which 
        ),
    TicketField('Sequence number', 2,
        'xy: ticket x out of y tickets'),
    TicketField('Non-used digits', 10, 'for future use'),
    TicketField('Traveler type', 2, 'frequent traveler / ...'),
    TicketField('Number of adults', 2, '00 - 99'),
    TicketField('Number of childs', 2, '00 - 99'),
    TicketField('Year (last digit)', 1, 'e.g. 2007 -> \'7\''),
    TicketField('Emission day', 3,
        'Sequence number (1/1=1, 2/1=2, ..)'),
    TicketField('Begin validity day', 3,
        'Sequence number (1/1=1, 2/1=2, ..)'),
    TicketField('End validity day', 3,
        'Sequence number (1/1=1, 2/1=2, ..)'),
    TicketField('Departure station 1', 5,
        '5 digit Alphanumerical encoding e.g. FRPNO'),
    TicketField('Arrival station 1', 5,
        '5 digit Alphanumerical encoding'),
    TicketField('Train number 1', 6,
        '6 characters (or 5 + 1 blanc)'),

    TicketField('Security code 1', 4,
        'Specific code for a train - antifraud'),
    TicketField('Departure date 1', 3,
        'Sequence number (1/1=1, 2/1=2, ...)'),
    TicketField('Coach number 1', 3, 'Alphanumerical - 3 digits'),
    TicketField('Seat/bed number 1', 3,
        'Alphanumerical - 3 digits'),
```

Contenu masqué

```
    TicketField('Class of travel 1', 1,
        '1 =first class, 2=second class'),
    TicketField('Tariff code 1', 4, '4 blancs = full fare ticket'),
    TicketField('Class of service 1', 2,
        'defining extra services or conditions (non exchangable, ...)'
        ),
    TicketField('Departure station 2', 5,
        '5 digit Alphanumerical encoding e.g. FRPNO'),
    TicketField('Arrival station 2', 5,
        '5 digit Alphanumerical encoding'),
    TicketField('Train number 2', 6,
        '6 characters (or 5 + 1 blanc)'),
    TicketField('Security code 2', 4,
        'Specific code for a train - antifraud'),
    TicketField('Departure date 2', 3,
        'Sequence number (1/1=1, 2/1=2, ...)'),
    TicketField('Coach number 2', 3, 'Alphanumerical - 3 digits'),
    TicketField('Seat/bed number 2', 3,
        'Alphanumerical - 3 digits'),
    TicketField('Class of travel 2', 1,
        '1 =first class, 2=second class'),
    TicketField('Tariff code 2', 4, '4 blancs = full fare ticket'),
    TicketField('Class of service 2', 2,
        'defining extra services or conditions (non exchangable, ...)'
        ),
]



ticket_text = StringIO(
    '''eEDVQAOQZF6871928331111000000000  02000012      FRPLYFRNIC06173
 9982080003083184  AZFRNICFRXMT86043     080      2PN00B '''.repla
    ce('\n',
    ''))

for field in fields:
    field_value = ticket_text.read(field.size)

    print(field.name, '=>', repr(field_value))

print()
print('Remaining text =>', repr(ticket_text.read()))

print()
```

Retourner au texte.

Contenu masqué n°2

Il en décrit le début: sont stockées au moins la station de validation (ou rien si le ticket est neuf), un type de ticket et un bit de parité. Le bit de parité est la forme la plus simple de somme de contrôle; elle consiste à additionner tous les bits présents dans un message et à inscrire «1» ou «0» selon si la somme est paire ou impaire. Elle est censée permettre de détecter les dégradations involontaires (parfois).

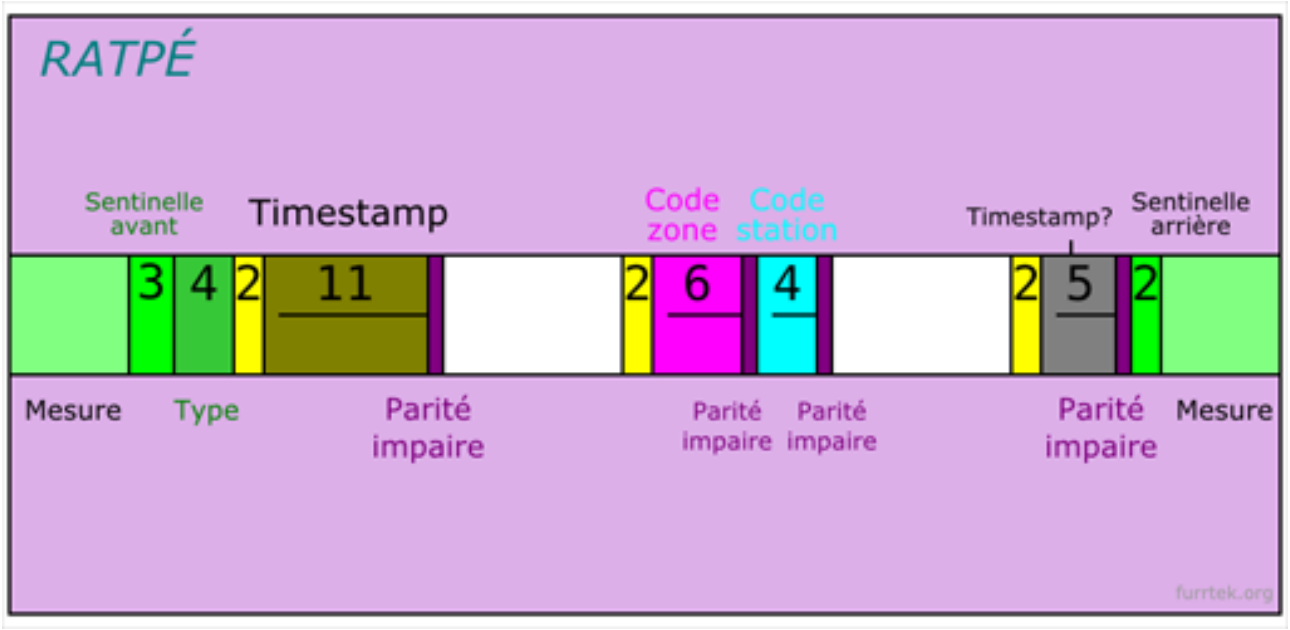
La durée d'utilisation des titres après validation sur le réseau parisien étant limitée, la suite du ticket contient à minima un horodatage de la validation.

Pour citer l'article:

> J'ai donc entrepris de décoder les premiers bits significatifs communs aux trois tickets, et voici ce qui en ressort :
>
> ```
> Ticket neuf:
> 11111111 00100 11111 11111 11111 0
>
> Ticket #1:
> 11111111 00100 11111 00110 01001 0
>
> Ticket #2:
> 11111111 00100 11111 00110 01110 1
> ```
>
> Les espaces sont de mon fait 🍊 . J'ai pu déduire le rôle de chacun (ou du moins tenter de deviner) à partir des informations décodées :
>
> — La première série de bits à 1 sert probablement au système de lecture à calculer la vitesse de balayage, une sorte de padding
> — La séquence suivante de 5 bits correspond probablement au type de ticket. Je n'ai pas pu prendre d'autres tickets pour corroborer cette hypothèse
> — La deuxième séquence de 5 bits semble être un marqueur
> — Les troisième et quatrième séquences semblent être un code de station (présent sur les tickets compostés)
>
> Si on regarde de plus près le codage, on peut voir que sur le premier ticket validé les troisième et quatrième séquences de 5 bits correspondent aux valeurs 69 et 6E. Ou plus précisément aux valeurs 6 et 9, et 6 et 14. Cela ressemble bien aux codes de stations connus de Paris, tels que décrit sur Wikipédia . Cela signifie qu'un de mes tickets a été validé à Saint-Lazare même (code station 0609), l'autre à la station ayant pour code 0614 (inconnu sur Wikipédia 🍊). Le dernier bit que j'ai repéré semble être un bit de parité.

Le blogueur Furrtek est allé plus loin et a produit un schéma partiel des 64 bits (8 octets) de données utiles présentes sur la bande magnétique, que je reproduis ci-dessous:

Contenu masqué



Retourner au texte.

Liste des abréviations

GDS Acronyme de « Global Distribution System ». 2

IATA Acronyme de « International Air Transport Association ». 2, 3, 5, 8

Sabre À l'origine, acronyme de « Semi-automated Business Research Environment ». 2

SOCRATE Acronyme de « Système offrant à la clientèle des réservations d'affaires et de tourisme en Europe ». 2