



18.06.2014

Julkinen

HSL Matkakortin ja Android-
kirjaston kuvaus kehittäjille

Versio:
1.01

sivu:
1 (19)

HSL Matkakortin ja Android- kirjaston kuvaus kehittäjille

Sisällysluettelo

1	Yleistä.....	5
1.1	Tiedostojen kuvaustapa ja tietojen formaatit.....	5
2	HSL:n matkakortin sisältö.....	6
2.1	ApplicationInformation.....	6
2.2	PeriodPass.....	6
2.3	StoredValue.....	8
2.4	eTicket.....	8
2.5	History.....	9
3	HSL:n kertakortin sisältö.....	10
3.1	Kertalippusovellus.....	10
3.1.1	Kertakortin sarjanumeron muodostus.....	12
4	Taulukoita.....	12
4.1	Vyöhykkeet.....	12
4.2	Myyntilaitetyypit.....	12
5	Android kirjasto HSL korteille.....	13
5.1	com.hsl.cardproduct.TravelCard.....	13
5.1.1	com.hsl.cardproduct.TravelCard.History.....	14
5.2	com.hsl.cardproduct.SingleTicket.....	15
5.3	com.hsl.cardproduct. eTicket.....	15
5.4	com.hsl.util. ValidityAreaMappings.....	16
5.5	com.hsl.util. Convert.....	16
5.6	com.hsl.examples.CardOperations.....	17
5.6.1	readTravelCardData.....	17
5.6.2	readSingleCardData.....	17
5.6.3	getTravelCardStrings.....	17
5.6.4	getSingleTicketStrings.....	17
5.6.5	getTravelCardPeriod1Validity.....	17
5.6.6	getTravelCardValue.....	17
5.6.7	getETicketValidity.....	17

5.6.8 getTravelCardHistory..... 18

Termit ja lyhenteet

BCD	Binary Coded Decimal. Kortin kuvauksissa käytetty tietotyyppi, kts. kpl 1.1
Datestamp	Päiväys. Kortin kuvauksissa käytetty tietotyyppi, kts. kpl 1.1
LSB	Least significant bit. Kuvaa tavun bittien tulkintajärjestystä
MIFARE DESFire	NXP:n etäkorttityyppi
MIFARE Ultralight	NXP:n etäluettava muistikorttityyppi
MSB	Most significant bit. Kuvaa tavun bittien tulkintajärjestystä
NFC	Near Field Communications
NXP	NXP Semiconductors. Elektroniikkavalmistaja, joka omistaa ja lisensoi MIFARE-tuoteperheen tuotteita. (ent. Philips Semiconductors)
Timestamp	Aikaleima. Kortin kuvauksissa käytetty tietotyyppi, kts. kpl 1.1
UINT	Positiivinen kokonaisluku. Kortin kuvauksissa käytetty tietotyyppi, kts. kpl 1.1

1 Yleistä

Tämä dokumentti kuvaa HSL:n matkakorteilla olevan korttisovelluksen sisältöä sovelluskehittäjille, jotka haluavat luoda sovelluksia sekä palveluita, joissa hyödynnetään matkakortin tietoja.

HSL:n matkakortti on NXP:n Mifare DESFire tuotealustalle määritelty korttisovellus. Mifare DESFire korttien käsittelyyn ja rakenteeseen liittyviä tietoja ei kuvata tässä dokumentissa.

HSL:n Matkakorttisovellus koostuu useista tiedostoista, joista matkustajille suunnattujen sovellusten ja palveluiden kannalta tarpeelliset tiedostot kuvataan tässä dokumentissa.

1.1 Tiedostojen kuvaustapa ja tietojen formaatit

Tiedostojen sisältö jakaantuu vaihtelevan mittaisiin bittikenttiin ja tiedostot kuvataan dokumentissa alla olevan esimerkin mukaisesti:

Tietokenttä	Offset tavu	Offset bitti	Pituus (bittejä)	Tyyppi	Kuvaus
Field1	0	0	3	UINT	...
Field2	0	3	8	BCD	...
<i>reserved</i>	1	2	2	-	varattu tila
Field3	1	5	11	Timestamp	...
Pituus: 3 tavua			(24 bittiä)		

Taulukko 1: Tiedoston sisällön esimerkkitaulukko

Tietojen sijainti tiedostossa kerrotaan taulukossa offset-tavuilla ja offset-tavujen sisällä offset-biteillä. Offset-tavu kertoo tiedoston alusta lähtien lasketun tavun ja offset-bitti kertoo monennestako offset-tavun bitistä tietokenttä alkaa. Tietokenttien sisällä bittejä tulkitaan järjestyksessä MSB...LSB.

Esimerkki yllä kuvatun tiedoston sisällöstä kun tiedostosta luettu sisältö on **"50C253h"**:

Offset-tavu 0 (50h)								Offset-tavu 1 (C2h)								Offset-tavu 2 (53h)							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	1	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1
field1 = 2				field2 = 86 (BCD)				res.				Field3 = 595 (= "09:00")											

Taulukko 2: Esimerkkitiedoston sisältö

Tietokenttien sisältöä kuvataan taulukoissa seuraavilla termeillä:

UINT	kokonaisluku, maksimiarvo vaihtelee tietokentän käyttämän bittimäärän mukaan
Datestamp	Aikaleima, jonka arvo on päivien lukumäärä laskettuna 1.1.1997 alkaen (EN 1545-1, DateStamp)
Timestamp	Aikaleima, jonka arvo on minuuttien lukumäärä alkaen vuorokauden vaihteesta 00:00 (EN 1545-1, TimeStamp)
BCD	BCD-koodaus, jossa yksi yksinumeroinen luku (0-9) koodataan 4:llä bitillä

2 HSL:n matkakortin sisältö

HSL:n Matkakorttisovelluksen tiedostoista seuraavat ovat kuvattuina tässä dokumentissa:

Tiedoston nimi	Id nro	Pituus (tavuja)	Tyyppi	Kuvaus
ApplicationInformation	8	11	Standard	HSL:n korttisovelluksen tunniste ja versiotiedot
PeriodPass	1	32	Backup	Kausituotteiden voimassaolo-, lataus- ja käyttötiedot
StoredValue	2	12	Backup	Matkakortin arvokukkaro
eTicket	3	26	Backup	Arvolipun tiedot
History	4	96	Cyclic[8]	Historiatiedosto kahdeksalle viimeiselle kortin käyttötapahtumalle

Taulukko 3: Matkakortin tiedostot

Seuraavissa kappaleissa kuvataan tiedostojen sisältöä sekä tietojen käyttötarkoituksia.

2.1 ApplicationInformation

ApplicationInformation –tiedosto sisältää HSL:n korttisovelluksen tunniste- ja versiotietoja.

Tietokenttä	Offset tavu	Offset bitti	Pituus (bitteinä)	Tyyppi	Kuvaus
Sovellustiedot:					
ApplicationVersion	0	0	4	UINT	Korttisovelluksen versiotunnus (=1)
<i>reserved</i>	0	4	4	-	varattu tila
ApplicationInstanceld	1	0	72	BCD[18]	Matkakortin numero. 18 numeroa BCD-koodatuna
PlatformType	10	0	3	UINT	Korttialustan tyyppi 0=NXP DESFire 4kB
SecurityLevel	10	3	1	UINT	Määrittelee matkakortin ja lukijan välisen kommunikoinnin turvatasoa 0=Avoin 1=MAC-suojattu (kts. DESFire korttispesifikaatio)
<i>reserved</i>	10	2	5	-	varattu tila
Pituus: 11 tavua (88 bittiä)					

Taulukko 4: Application information –tiedosto

2.2 PeriodPass

PeriodPass-tiedosto sisältää matkakortin kausituotteiden voimassaolo-, lataus- ja käyttötiedot.

Kortille voidaan sijoittaa kaksi kautta, joiden voimassaoloaika voidaan asettaa erikseen. Uuden kauden voimassaolo voi alkaa aikaisintaan edellisen kauden viimeistä voimassaolopäivää seuraavana päivänä. Kausi on voimassa päättymispäivän sekä seuraavan vuorokauden puolelle käyttöhetkellä käytössä olevan liikennöintivuorokauden

vaihtumiseen asti. Tällä hetkellä liikennöintivuorokausi vaihtuu klo 04:30. Mikäli päättymispäiväksi on asetettu arvo 0, on kausituote voimassa jatkuvasti. Kausituotteilla voi olla myös muita tuotekohtaisia voimassaolorajoituksia, kuin mitä kortilla on asetettu (esim. ei voimassa yöliikenteessä).

Kortilla kumpi tahansa kausista (1 tai 2) voi olla voimassa ja toinen kausi voi olla vanhentunut tai olla tulossa voimaan seuraavaksi.

Kun kausien voimassaoloa tutkitaan, pitää huomioida jatkuvasti voimassaolevien kausien tutkinta ennen tai jälkeen päivämäärämuunnoksen. Ennen muunnosta arvo on 0 tai muunnoksen jälkeen 1.1.1997.

Tietokenttä	Offset tavu	Offset bitti	Pituus (bitteinä)	Tyyppi	Kuvaus
Kausituote 1:					
ProductCode1	0	0	14	UINT	Kausituotteen 1 tunnus 0-9999.
ValidityAreaType1	1	6	1	UINT	Voimassaoloalueen tyyppi 0=Vyöhyke, 1=Ajoneuvo
ValidityArea1	1	7	4	UINT	Voimassaoloalue (kts. kpl 4.1)
PeriodStartDate1	2	3	14	Datestamp	Voimaantulopäivä
PeriodEndDate1	4	1	14	Datestamp	Voimassaolon päättymispäivä
<i>reserved</i>	5	7	1	-	varattu tila
Kausituote 2:					
ProductCode2	6	0	14	UINT	Kausituotteen 2 tunnus 0-9999.
ValidityAreaType2	7	6	1	UINT	Voimassaoloalueen tyyppi 0=Vyöhyke, 1=Ajoneuvo
ValidityArea2	7	7	4	UINT	Voimassaoloalue (kts. kpl 4.1)
PeriodStartDate2	8	3	14	Datestamp	Voimaantulopäivä
PeriodEndDate2	10	1	14	Datestamp	Voimassaolon päättymispäivä
<i>reserved</i>	11	7	1	-	varattu tila
Edellinen lataus:					
ProductCode	12	0	14	UINT	Ladatun tuotteen tunnus. 0-9999.
LoadingDate	13	6	14	Datestamp	Latauksen päivämäärä
LoadingTime	15	4	11	Timestamp	Latauksen kellonaika
LoadedPeriodLength	16	7	9	UINT	Ladatun kauden pituus päivinä
PriceOfPeriod	18	0	20	UINT	Ladatun kauden hinta sentteinä
LoadingOrganisationID	20	4	14	UINT	Lataajan organisaatiotunnus 0-9999
LoadingDeviceNumber	22	2	14	UINT	Latauslaitteen tunnus 0-9999
Edellinen käyttö:					
BoardingDate	24	0	14	Datestamp	Nousupäivä
BoardingTime	25	6	11	Timestamp	Nousuaika
BoardingVehicle	27	1	14	UINT	Ajoneuvon tunnus 0-9999
BoardingLocation-NumberType	28	7	2	UINT	Nousupaikkatyyppi 0=varattu, 1=Linjan numero, 2=Junanumero, 3=Laitenumero
BoardingLocationNumber	29	1	14	UINT	Nousupaikan tunnus 0-9999
BoardingDirection	30	7	1	UINT	Noususuunta
BoardingArea	31	0	4	UINT	Nousualue (kts. kpl 4.1)
<i>reserved</i>	31	4	4	-	varattu tila

Pituus: 32 tavua (256 bittiä)

Taulukko 5: PeriodPass –tiedosto

2.3 StoredValue

StoredValue –tiedosto sisältää matkakortin arvon ja arvon lataustiedot.

HSL:n järjestelmän kaikki matkakortit sisältävät arvokukkaron. Mikäli kortille on talletettu arvoa, sitä voidaan käyttää arvolippujen ostamiseen itsepalvelulukijoilla tai automaateilla sekä kertalippujen maksamiseen kuljettajalaitteella.

Tietokenttä	Offset tavu	Offset bitti	Pituus (bitteinä)	Tyyppi	Kuvaus
Arvo:					
ValueCounter	0	0	20	UINT	Kortin arvo sentteinä
Edellinen arvon lataus:					
LoadingDate	2	4	14	Datestamp	Latauksen päivämäärä
LoadingTime	4	2	11	Timestamp	Latauksen kellonaika
LoadedValue	5	5	20	UINT	Ladattu arvo sentteinä
LoadingOrganisationID	8	1	14	UINT	Lataajan organisaatiotunnus 0-9999
LoadingDeviceNumber	9	7	14	UINT	Latauslaitteen tunnus 0-9999
<i>reserved</i>	11	5	3	-	varattu tila
Pituus: 12 tavua (96 bittiä)					

Taulukko 6: StoredValue –tiedosto

2.4 eTicket

eTicket –tiedosto sisältää matkakortin arvolipun tiedot.

Arvolla ostetun arvolipun tiedot talletetaan kortille arvolipuksi, jonka tiedot on jaettu myynti-, voimassaolo- sekä nousutietoihin.

Voimassaolotiedot päivitetään myynnin tai ensimmäisen leimauksen yhteydessä ja nousutiedot jokaisen matkustuksen yhteydessä. Jos voimassaolotiedot asetetaan myynnin yhteydessä, voimassaolon alkuajaksi asetetaan myyntiajankohta. Mikäli lipun myynti eli arvon veloitus tapahtuu itsepalvelulukijalla matkustuksen yhteydessä, päivitetään myynti-, voimassaolo- ja nousutiedot samalla kertaa. Nousutiedot päivitetään myös vaihdon yhteydessä.

Tietokenttä	Offset tavu	Offset bitti	Pituus (bitteinä)	Tyyppi	Kuvaus
Myyntitiedot:					
ProductCode	0	0	14	UINT	Lipputuotteen tunnus 0-9999
Child	1	6	1	UINT	0=Aikuinen tai muu asiakasryhmä, 1=Lapsi
LanguageCode	1	7	2	UINT	Kielikoodi 0=Suomi, 1=Ruotsi, 2=Englanti

ValidityLengthType	2	1	2	UINT	Voimassaolon pituuden tyyppi 0=Minuutteja, 1=Tunteja, 2=Vuorokausia, 3=Päiviä
ValidityLength	2	3	8	UINT	Voimassaolon pituus
ValidityAreaType	3	3	1	UINT	Voimassaoloalueen tyyppi 0=Vyöhyke, 1=Ajoneuvo
ValidityArea	3	4	4	UINT	Voimassaoloalue (kts. kpl 4.1)
SaleDate	4	0	14	Datestamp	Myyntin päivämäärä
SaleTime	5	6	5	UINT	Myyntitunti 0-23
SaleDeviceType	6	3	3	UINT	Myyntilaitteen tyyppi (kts. kpl 4.2)
SaleDeviceNumber	6	6	14	UINT	Myyntilaitteen tunnus 0-9999
TicketFare	8	4	14	UINT	Lipun hinta sentteinä 0 - 9999
GroupSize	10	2	5	UINT	Ryhmän koko 1-31
SaleStatus	10	7	1	UINT	1=Voimassaolotiedot asetettu
Voimassaolotiedot:					
ValidityStartDate	11	0	14	Datestamp	Voimaantulon päivämäärä
ValidityStartTime	12	6	11	Timestamp	Voimaantulon kellonaika
ValidityEndDate	14	1	14	Datestamp	Voimassaolon päättymispäivä
ValidityEndTime	15	7	11	Timestamp	Voimassaolon päättymisaika
<i>reserved</i>	17	2	5	-	varattu tila
ValidityStatus	17	7	1	UINT	1=Voimassaolo alkanut
Nousutiedot:					
BoardingDate	18	0	14	Datestamp	Nousupäivä
BoardingTime	19	6	11	Timestamp	Nousuaika
BoardingVehicle	21	1	14	UINT	Ajoneuvon tunnus 0-9999
BoardingLocation- NumberType	22	7	2	UINT	Nousupaikkatyyppi 0=varattu, 1=Linjan numero, 2=Junanumero, 3=Laitenumero
BoardingLocationNumber	23	1	14	UINT	Nousupaikan tunnus 0-9999
BoardingDirection	24	7	1	UINT	Noususuunta
BoardingArea	25	0	4	UINT	Nousualue (kts. kpl 4.1)
<i>reserved</i>	25	4	4	-	varattu tila
Pituus: 26 tavua (208 bittiä)					

Taulukko 7: eTicket-tiedosto

2.5 History

Historiatiedosto on syklinen tiedosto johon voidaan tallettaa seitsemän kortin käyttötapahtumaa. Tiedoston pituus on kahdeksan tapahtuman mittainen, mutta yksi tietue on varattuna tiedon kirjoituksen varmennukseen, joka on DESFire kortin sisäinen toiminto.

Historiatiedostoon talletetaan kausituotteen käyttö matkustukseen sekä kortin arvosta tehdyt veloitukset (= arvolipun osto tai kertalipun maksu kuljettajapäätteellä). Arvolipun käyttöä tai vaihtomatkestusta ei talleteta historiatietoihin eikä kauden tai arvon latausta talleteta historiatietoihin.

Kausikortin leimauksessa ja arvon veloituksen tapahtuessa nousun yhteydessä talletetaan historiatietoihin nousutapahtuman päivämäärä, aika sekä vaihtoajan päättymispäivä ja aika.

Mikäli arvolla Maksetaan kertalippu tai ostetaan myöhemmin ensimmäisestä leimauksesta voimaan tuleva arvolippu, talletetaan nousupäivän ja ajan tilalle myyntihetken aikatiedot. Tällöin vaihtoajan päättymisaikatietoja ei talleteta.

Tietokenttä	Offset tavu	Offset bitti	Pituus (bitteinä)	Tyyppi	Kuvaus
Historiatapahtuma[8]:					
TransactionType	0	0	1	UINT	0=Kauden leimaus, 1=Arvon veloitus
BoardingDate	0	1	14	Datestamp	Nousupäivä
BoardingTime	1	7	11	Timestamp	Nousuaika
TransferEndDate	3	2	14	Datestamp	Vaihto-oikeuden päättymispäivä
TransferEndTime	5	0	11	Timestamp	Vaihto-oikeuden päättymisaika
TicketFare	6	3	14	UINT	Lipun hinta sentteinä 0 - 9999
GroupSize	8	1	5	UINT	Ryhmän koko 1-31
RemainingValue	8	6	20	UINT	Arvokukkaron saldo veloituksen jälkeen. Asetetaan 0-arvoon kausikortin leimaustietojen yhteydessä.
<i>reserved</i>	11	2	6	-	varattu tila
Pituus: 8*12 tavua			(8*96 bittiä)		

Taulukko 8: History-tiedosto

3 HSL:n kertakortin sisältö

HSL:n kertakorttisovellus mahdollistaa erilaisten lipputuotteiden kuten matkailijakorttien, ennakkoon myytävien kerta- ja matkailijalippujen lippujen sekä automaattista myytävien kertalippujen toteutuksen.

Tämä sovellus on määritelty Mifare Ultralight- tuotealustalle. Mifare Ultralight korttien käsittelyyn ja rakenteeseen liittyviä tietoja ei kuvata tässä dokumentissa. Alla kuvattu Kertakortin sisältö alkaa kortin ensimmäisestä datalohkosta (4).

3.1 Kertalippusovellus

Kertalippusovellus muodostuu kahdesta päälohkosta. Sovelluksen perustiedot (Sovellustiedot) sisältävät sovelluksen ja tuotteenomistajan tunnistetiedot sekä versionhallintatiedot. Ostetun lipun tiedot talletetaan sähköisenä lippuna (eTicket) sovellusrakenteeseen.

Voimassaolotiedot päivitetään myynnin tai ensimmäisen leimauksen yhteydessä ja nousutiedot jokaisen matkustuksen yhteydessä. Jos voimassaolotiedot asetetaan myynnin yhteydessä, voimassaolon alkuajaksi asetetaan myyntiajankohta.

Tietokenttä	Offset tavu	Offset bitti	Pituus (bitteinä)	Tyyppi	Kuvaus
Sovellustiedot:					
ApplicationVersion	0	0	4	UINT	Korttisovelluksen versiotunnus (=1)
<i>reserved</i>	0	4	4	-	varattu tila

ApplicationInstanceid	1	0	44	BCD[11]	Osa kertakortin numerosta. Kertakortin sarjanumeron muodostus on kuvattu kappaleessa 3.1.1
PlatformType	6	4	3	UINT	Korttialustan tyyppi 1=NXP Mifare Ultralight
<i>reserved</i>	6	7	1	-	varattu tila
Myyntitiedot:					
ProductCode	7	0	14	UINT	Lipputuotteen tunnus 0-9999
Child	8	6	1	UINT	0=Aikuinen tai muu asiakasryhmä, 1=Lapsi
LanguageCode	8	7	2	UINT	Kielikoodi 0=Suomi, 1=Ruotsi, 2=Englanti
ValidityLengthType	9	1	2	UINT	Voimassaolon pituuden tyyppi 0=Minuutteja, 1=Tunteja, 2=Vuorokausia, 3=Päiviä
ValidityLength	9	3	8	UINT	Voimassaolon pituus
ValidityAreaType	10	3	1	UINT	Voimassaoloalueen tyyppi 0=Vyöhyke, 1=Ajoneuvo
ValidityArea	10	4	4	UINT	Voimassaoloalue (kts. kpl 4.1)
SaleDate	11	0	14	Datestamp	Myyntiin päivämäärä
SaleTime	12	6	5	UINT	Myyntitunti 0-23
SaleDeviceType	13	3	3	UINT	Myyntilaitteen tyyppi (kts. kpl 4.2)
SaleDeviceNumber	13	6	14	UINT	Myyntilaitteen tunnus 0-9999
TicketFare	15	4	14	UINT	Lipun hinta sentteinä 0 - 9999
GroupSize	17	2	5	UINT	Ryhmän koko 1-31
SaleStatus	17	7	1	UINT	1=Voimassaolotiedot asetettu
<i>reserved</i>	18	0	48	-	varattu
Voimassaolotiedot:					
ValidityStartDate	24	0	14	Datestamp	Voimaantulon päivämäärä
ValidityStartTime	25	6	11	Timestamp	Voimaantulon kellonaika
ValidityEndDate	27	1	14	Datestamp	Voimassaolon päättymispäivä
ValidityEndTime	28	7	11	Timestamp	Voimassaolon päättymisaika
<i>reserved</i>	30	2	78	-	varattu tila
Nousutiedot:					
BoardingDate	40	0	14	Datestamp	Nousupäivä
BoardingTime	41	6	11	Timestamp	Nousuaika
BoardingVehicle	43	1	14	UINT	Ajoneuvon tunnus 0-9999
BoardingLocation-NumberType	44	7	2	UINT	Nousupaikkatyyppi 0=Portin numero, 1=Linjan numero, 2=Junanumero, 3=Laitenumero
BoardingLocationNumber	45	1	14	UINT	Nousupaikan tunnus 0-9999
BoardingDirection	46	7	1	UINT	Noususuunta
BoardingArea	47	0	4	UINT	Nousualue (kts. kpl 4.1)
<i>reserved</i>	47	4	4	-	varattu tila
Pituus: 48 tavua (384 bittiä)					

Taulukko 9: eTicket-tiedosto

3.1.1 Kertakortin sarjanumeron muodostus

Kortin numero muodostetaan kertakortin sarjanumerosta (UID) sekä kertalippusovelluksen ApplicationInstanceId:stä seuraavasti:

1. Kortinumeron alkuun otetaan ApplicationInstanceId:n BCD-luvun 10 ensimmäistä numeroa.
2. Edellisten perään lisätään 7 numeroa laskettuna kortin sarjanumerosta:
Kortin sarjanumeron seitsemästä tavusta (SN0-SN6) tehdään kolme XOR operaatiota (AA=SN1 XOR SN4, BB=SN2 XOR SN5, CC=SN3 XOR SN6), joiden tulostavut yhdistetään yhdeksi kolmetavuiseksi luvuksi AABBCCh. Luvusta nollataan ylin bitti (MSB) ja tulos muutetaan seitsemännumeroiseksi desimaaliluvuksi, joka lisätään kortin numeroon
3. Kortin numeron loppuun lisätään vielä viimeinen ApplicationInstanceId:n BCD luku. Kortin sarjanumerosta

Esimerkki kortinnumeron muodostuksesta:

1. ApplicationInstanceId (BCD[11]): 0x12, 0x34, 0x56, 0x78, 0x90, 0x1 ("12345678901")
2. UID (SN0-SN6): 0x04, 0xA1, 0xB2, 0xC3, 0x4D, 0x5E, 0x6F
3. XOR: AA=0xA1 xor 0x4D = 0xEC, BB=0xEC, CC=0xAC
4. XOR tulos: 0xECECAC
5. XOR tuloksesta ylin bitti pois: 0x6CECAC (desimaalilukuna "7138476")
6. Kortinnumero "1234567890 7138476 1"

4 Taulukoita

4.1 Vyöhykkeet

Arvo	Selite
00	Ei määritelty
01	Helsinki
02	Espoo
03	varattu
04	Vantaa
05	Seutu (HEL+ESP-VAN)
06	Kirkkonummi
07	Vihti
08	Nurmijärvi
09	Kerava-Sipoo
10	varattu (Sipoo)
11-13	varattu
14	Lähialue (Lähiseutu2, ESP+VAN+KIR+KER+SIP)
15	Koko alue (Lähiseutu3, HEL+ESP+VAN+KIR+KER+SIP)

4.2 Myyntilaitetyypit

Arvo	Selite
0	Myyntilaitte (palvelupisteet)
1	Kuljettajapäätte
2	Itsepalvelulukija
3	Automaatti
4	Palvelin
5	HSL:n pienlaitteet

6	Ulkopuolisten järjestelmien laitteet
7	varattu

5 Android kirjasto HSL korteille

Android laitteille suunnattu java-kirjasto *HSL-card library* tarjoaa valmiin toteutuksen HSL-matkakortin tietojen purkamiseen valmiiksi käytettävään muotoon sen jälkeen kun tiedot on luettu matka- tai kertakortilta.

Kirjasto sisältää myös esimerkkejä siitä, kuinka tiedot puretaan ja miten kirjastoa käytetään tietoja hyödynnettäessä.

Kirjasto sisältää seuraavat java-paketit ja luokat

Paketin nimi	Luokat	Kuvaus
com.hsl.cardproducts	TravelCard	Luokka HSL:n matkakortin sisällön esittämiseksi.
com.hsl.cardproducts	TravelCard.History	Luokka HSL:n matkakortin tapahtumahistorian esittämiseksi.
com.hsl.cardproducts com.hsl.cardproducts	SingleTicket eTicket	Luokka HSL:n kertakortin sisällön esittämiseen HSL-kertalipun sisältöluokka, jota käytetään kertakortin tietojen ja matkakortin arvolipun tietojen tallennukseen.
com.hsl.util	ValidityAreaMappings	Apuluokka, jolla voi hakea kortin voimassaoloalueelle tai -ajoneuvolle valmiit tekstit.
com.hsl.util	Convert	Apuluokka, jota käytetään kirjatun sisäisesti mm. päivämäärämuunnoksissa.
com.hsl.examples	CardOperations	Esimerkkiluokka, joka sisältää esimerkkejä matka- ja kertakortin tietojen luvusta sekä TravelCard ja SingleTicket -luokkien luonnista ja käytöstä.

Taulukko 10: Android-kirjaston paketit ja luokat

Seuraavissa kappaleissa listataan luokat ja niiden metodit, mutta luokista on tarjolla myös *javadoc*-dokumentaatio. Lisäksi *com.hsl.examples.CardOperations*-luokan lähdekoodin esimerkit esittelevät kirjaston käyttöä.

5.1 com.hsl.cardproduct.TravelCard

Luokka HSL:n matkakortin sisällön esittämiseen.

Kun TravelCard-instanssi luodaan, annetaan sille parametrina matkakortilta luetut tiedostot sellaisenaan. TravelCard purkaa tiedot sisäisiin muuttujiin, jotka ovat luettavissa tietokenttiä vastaavilla get-kutsuilla. Get-kutsujen nimet vastaavat kappaleessa 2 kuvattujen kortin tietokenttien nimiä. Esimerkiksi matkakortin kausituote 1:n tuotetunnus *ProductCode1* saadaan kutsulla *getProductCode1()* jne.

Poikkeuksena matkakortilta saataviin tietoihin on arvolippu, joka vastaa tietorakenteeltaan kertakortin sisältöä ja siksi sen esittämiseen käytetään kertakorttien kanssa yhteistä eTicket-luokkaa. Matkakortin arvolipun sisältö kysytään *getValueTicket()*-kutsulla, joka palauttaa eTicket-luokan instanssin. Tältä instanssilta voidaan edelleen kysyä eTicket-tiedostossa kuvattuja kenttiä get-kutsuilla, esim. *getValueTicket().getProductCode()* jne.

Constructor	Parametrit
TravelCard (byte[] appInfoBytes, byte[] periodPassBytes, byte[] storedValueBytes, byte[] eTicketBytes, byte[] historyBytes)	appInfoBytes – ApplicationInformation-tiedoston sisältö periodPassBytes – PeriodPass-tiedoston sisältö. storedValueBytes – StoredValue-tiedoston sisältö eTicketBytes – eTicket-tiedoston sisältö historyBytes – history-tiedoston sisältö

Taulukko 11: TravelCard-luokan luonti

Metodi	Parametrit	Paluarvo
getApplicationInstanceId		String
getApplicationVersion		byte
getBoardingArea		byte
getBoardingDate		Date
getBoardingDirection		byte
getBoardingLocationNum		short
getBoardingLocationNumType		byte
getBoardingVehicle		short
getErrorStatus		int
getHistory		TravelCard.History[]
getHistoryLen		byte
getLoadedPeriodLength		short
getLoadedPeriodPrice		int
getLoadedPeriodProduct		short
getLoadedValue		int
getLoadingDate		Date
getLoadingDeviceNumber		short
getLoadingOrganizationID		short
getLoadingTime		byte
getPeriodEndDate1		Date
getPeriodEndDate2		Date
getPeriodLength1		short
getPeriodLength2		short
getPeriodLoadingDate		Date
getPeriodLoadingDeviceNumber		short
getPeriodLoadingOrganisation		short
getPeriodStartDate1		Date
getPeriodStartDate2		Date
getPlatformType		byte
getProductCode1		short
getProductCode2		short
getValidityArea1		byte
getValidityArea2		byte
getValidityAreaType1		byte
getValidityAreaType2		byte
getValueCounter		int
getValueTicket		eTicket

Taulukko 12: TravelCard-luokan metodit

5.1.1 com.hsl.cardproduct.TravelCard.History

Luokka HSL:n matkakortin historiatietojen esittämiseksi.

TravelCard-luokka luo sisäisesti taulukon, jossa on jokaista kortin historiatiedoissa olevaa tapahtumaa kohden yksi History-luokan instanssi. TravelCard luokka palauttaa taulukon *getHistory()*-kutsulla.

Constructor	Parametrit
TravelCard.History()	

Taulukko 13: *TravelCard.History*-luokan luonti

Metodi	Parametrit	Paluuarvo
<code>getTransactionType</code>		<code>int</code>
<code>getTransactionDateTime</code>		<code>Date</code>
<code>getGroupSize</code>		<code>int</code>
<code>getPrice</code>		<code>int</code>

Taulukko 14: *TravelCard.History* -luokan metodit

5.2 com.hsl.cardproduct.SingleTicket

Luokka HSL:n kertakortin sisällön esittämiseen.

Kun `SingleTicket`-instanssi luodaan, annetaan sille parametrina kertakortilta kaikki luettu data jaettuna kahteen osaan. `SingleTicket` purkaa tiedot sisäisiin muuttujiin siten, että Sovellustiedot puretaan suoraan `SingleTicket` instanssista get-kutsuilla luettaviksi (`getApplicationInstanceId()`, `getApplicationVersion()` ja `getPlatformType()`). Muut tiedot `SingleTicket` antaa luomalleen `eTicket`-luokan instanssille, josta tiedot ovat luettavissa get-kutsuilla, esim. `getValueTicket().getProductCode()` jne.

Constructor	Parametrit
<code>SingleTicket (byte[] appInfoBytes, byte[] eTicketBytes)</code>	<code>appInfoBytes</code> – kertakortin 23 ensimmäistä tavua kortin alusta asti luettuna. <code>eTicketBytes</code> – kertakortin lipputiedot (tavut 24-64, yht. 41 tavua).

Taulukko 15: *SingleTicket* -luokan luonti

Metodi	Parametrit	Paluuarvo
<code>getApplicationInstanceId</code>		<code>String</code>
<code>getApplicationVersion</code>		<code>byte</code>
<code>getPlatformType</code>		<code>byte</code>
<code>getValueTicket</code>		<code>eTicket</code>

Taulukko 16: *SingleTicket* -luokan metodit

5.3 com.hsl.cardproduct.eTicket

Luokka HSL:n kertalipun sisällön esittämiseen.

`eTicket`-instanssin luo `SingleTicket`- tai `TravelCard`-luokka, joilta `eTicket`-instanssi pyydetään `getValueTicket()`-kutsulla.

Constructor	Parametrit
<code>eTicket (byte[] eTicketData, boolean isSingleTicket)</code>	<code>eTicketData</code> – matkakortilta tai kertakortilta luettu kertalippudata. <code>isSingleTicket</code> – onko tiedot luettu kertakortilta.

Taulukko 17: *eTicket* -luokan luonti

Metodi	Parametrit	Paluuarvo
getBoardingArea		byte
getBoardingDate		Date
getBoardingDirection		byte
getBoardingLocationNum		short
getBoardingLocationNumType		byte
getBoardingVehicle		short
getChild		boolean
getGroupSize		byte
getLanguageCode		byte
getProductCode		short
getSaleDate		Date
getSaleStatus		byte
getSaleTime		byte
getValidityArea		byte
getValidityAreaType		byte
getValidityEndDate		Date
getValidityLength		byte
getValidityLengthType		byte
getValidityStartDate		Date
getValidityStatus		byte

Taulukko 18: eTicket-luokan metodit

5.4 com.hsl.util. ValidityAreaMappings

Apuluokka, jolla voi hakea lipputiedoista luettua numeerista vyöhyke- tai ajoneuvotietoa vastaavan tekstin.

Kirjastossa tekstit ovat tallennettuna suomen-, ruotsin- ja englanninkielisinä Android-resursseina kielikohtaisissa *strings.xml*-tiedostoissa. Android kielilokalisoinnista voi lukea lisää [Android developer-sivustolta](#).

Constructor	Parametrit
ValidityAreaMappings (android.content.Context app_context)	app_context – Android sovelluksen sovelluskonteksti, jota tarvitaan tekstiresurssien lukemiseksi.

Taulukko 19: ValidityAreaMappings-luokan luonti

Metodi	Parametrit	Paluuarvo
getValidityArea	byte areaType byte areaCode	String

5.5 com.hsl.util. Convert

Apuluokka, joka sisältää kirjaston sisäisesti käyttämiä kutsuja päivämäärä- ja heksamerkkijonomuunnoksiin.

5.6 com.hsl.examples.CardOperations

Esimerkkiluokka, joka sisältää lähdekoodiesimerkkejä matka- ja kertakorttien tietojen lukemiseen, TravelCard ja SingleTicket luokkien luomiseen ja tietojen näyttämiseen korttiluokista.

5.6.1 readTravelCardData

```
TravelCard readTravelCardData (android.nfc.tech.IsoDep isoCard)
```

Esimerkki siitä, kuinka matkakortin tiedostot luetaan matkakortilta ja kuinka luetuista tiedoista luodaan TravelCard-instanssi.

5.6.2 readSingleCardData

```
SingleTicket readSingleCardData (android.nfc.tech.MifareUltralight ulCard)
```

Esimerkki kuinka kertakortin tiedot luetaan kortilta ja kuinka luetuista tiedoista luodaan SingleTicket-instanssi.

5.6.3 getTravelCardStrings

```
getTravelCardStrings(TravelCard card, Context app_context)
```

Lukee matkakortin tiedot ja palauttaa kortin sarjanumeron, sekä 1 kausituotteen, arvon, arvolipun ja historiatiedot tekstinä. Kutsu koostaa CardOperations-luokan kutsujen *getTravelCardPeriod1Validity*, *getTravelCardValueString*, *getETicketValidityString* ja *getTravelCardHistoryString* palauttamia merkkijonoja.

5.6.4 getSingleTicketStrings

```
getSingleTicketStrings(SingleTicket singleTicket, Context app_context)
```

Lukee kertakortin tiedot ja palauttaa kortin sarjanumeron sekä voimassaolotiedot tekstinä. Kutsu käyttää CardOperations-luokan kutsua *getETicketValidityString* voimassaolotietojen hakemiseen.

5.6.5 getTravelCardPeriod1Validity

```
getTravelCardPeriod1Validity(TravelCard card, Context app_context)
```

Lukee matkakortin ensimmäisen kausituotteen tiedot ja palauttaa kauden voimassaolotiedoista muodostetun tekstin.

5.6.6 getTravelCardValue

```
getTravelCardValue(TravelCard card)
```

Lukee matkakortin arvotiedot ja palauttaa siitä muodostetun tekstin.

5.6.7 getETicketValidity

```
getETicketValidity(TravelCard card, Context app_context)
```

Lukee matkakortin arvolipun tai kertakortin lipputiedot ja palauttaa voimassaolotiedoista muodostetun tekstin.

5.6.8 `getTravelCardHistory`

`getTravelCardHistory` (TravelCard card)

Lukee matkakortin historiatiedot ja palauttaa niistä muodostetun tekstin.

6 Historia

Versio	Päiväys	Kuvaus
1.00	18.6.2014	Ensimmäinen versio
1.01	12.1.2015	Tekstien muotoilupäivitys (Luottamuksellinen → Julkinen)