ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní

Bc. Monika Andršová

OPTIMALIZACE DATOVÉ KOMUNIKACE V ODBAVOVACÍCH SYSTÉMECH V DOPRAVĚ

Diplomová práce

2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
d ě k a n
Konviktská 20, 110 00 Praha 1

K614..... Ústav aplikované informatiky v dopravě

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Monika Andršová

Kód studijního programu a studijní obor studenta:

N 3710 – ID – Inženýrská informatika v dopravě a spojích

Název tématu (česky): **Optimalizace datové komunikace v odbavovacích systémech v dopravě**

Název tématu (anglicky): Optimization of Data transmitted in Fare control systems

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Porovnání technologií ID karet s vazbou na využitelnost v dopravě: Mifare Standart, Desfire, EV1, EV2.
- Formulace konkrétního problému z oblasti datové komunikace.
- Optimalizace aplikací - velikost, datová struktura: porovnání struktury při využití MAD různých verzí, rozložení datového prostoru pro multiaplikační platformu.
- Optimalizace přístupových algoritmů uložených v SAM - rychlost čtení resp. zápisu do souborů v jednotlivých aplikacích.
- Návrh aplikace pro bonusový systém slev ve vazbě na tarifní plán v oblastním dopravním systému.

Rozsah grafických prací: dle charakteru diplomové práce

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: RANKL, W, EFFING, W. Smart Card Handbook : Third Edition. [s.l.] : WILEY, c2003. 1088 s. ISBN 0-470-85668-8.

RANKL, W. Smart Card Applications: Design Models for using and programming smart cards, New York, John Wiley & Sons, 2007.

Vedoucí diplomové práce: **Ing. Jana Kaliková, Ph.D.**

Datum zadání diplomové práce: **30. června 2014**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **30. listopadu 2015**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

L. S.

doc. Dr. Ing. Tomáš Brandejský
vedoucí
Ústavu aplikované informatiky v dopravě

prof. Dr. Ing. Miroslav Svítek, dr. h. c.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

Bc. Monika Andršová
jméno a podpis studenta

V Praze dne ..... 15. června 2015

Poděkování

Ráda bych poděkovala Ing. Janě Kalikové, Ph.D. a Ing Radkovi Holému za odborné vedení a poskytnutí cenných rad a připomínek v průběhu zpracovávání mé diplomové práce. Velmi děkuji svým rodičům, mému příteli, celé rodině a přátelům za trvalou motivaci, podporu a to, že při mně vždy stáli, a to jak po celou dobu mého studia, tak v jeho cílové rovince.

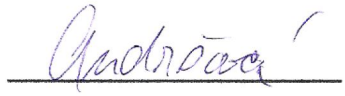
Čestné prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na ČVUT V Praze, Fakultě dopravní.

Prohlašuji, že jsem předloženou práci na téma **Optimalizace datové komunikace v odbavovacích systémech v dopravě** vypracovala samostatně a veškeré použité informační zdroje, ze kterých jsem při tom čerpala, jsou uvedeny na konci této práce v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č.121/2000 Sb., o právu autorském, právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 30. 11. 2015



Monika Andršová

ABSTRAKT

Tato diplomová práce se zabývá technologií bezkontaktních čipových karet MIFARE v oblasti odbavovacích systémů v dopravě, konkrétně vývojovými řadami MIFARE Classic a MIFARE DESFire. Hlavním cílem je na základě znalostí základních principů a mechanismů získaných v teoretické části provést optimalizaci datové komunikace probíhající mezi kartou a čtečkou. Optimalizace je prováděna návrhem vhodné struktury uložených dat v paměti a specifikací bezpečnostních mechanismů zaručujících požadovanou úroveň zabezpečení.

Klíčová slova:

MIFARE Classic, MIFARE DESFire, NXP Semiconductors, bezkontaktní čipová karta, zabezpečení, aplikace, odbavovací systémy. ISO/IEC 14443

ABSTRACT

This Diploma Thesis deals with the technology of contactless IC card MIFARE in the area of Fare control systems, specifically with development series MIFARE Classic and MIFARE DESFire. The main aim is to execute optimization of data communication between card and reader device based on knowledge of basic principles and mechanisms acquired in theoretical part. The optimization is performed by designing appropriate data structure stored in a card memory and specification of security mechanisms which guarantee the required security level.

Key words:

MIFARE Classic, MIFARE DESFire, NXP Semiconductors, Contactless Smart Cards, security, application, transport fare collection systems, ISO/IEC 14443A

Obsah

Seznam použitých zkratek

3DES	Triple DES – trojitý DES
ADV	Application Directory Version – verze MAD
AES	Advanced Encryption Standard – standard pokročilého šifrování dat
AID	Application IDentifier – identifikátor aplikace
APDU	Application Protocol Data Unit – datová jednotka aplikačního protokolu, (komunikační jednotka mezi čtečkou a kartou)
ASCII	American Standard Code for Information Interchange – americký standardní kód pro výměnu informací
ATS	Answer to Select – příkaz: „odpověď na SELECT“
BCC	Block Check Character – kontrolní znak bloku
CBC	Cipher Block Changing – algoritmus řetězení šifrovacích bloků
CID	Card IDentifier – identifikátor karty
CMAC	Cipher-based Message Authentication Code – šifrovaný autentizační kód zprávy
CPU	Central Processing Unit – centrální procesorová jednotka
CRC	Cyclic Redundancy Check/Code – cyklický redundantní součet, kontrolní součet
CSV	Comma-Separated Values – hodnoty oddělené čárkami
ČNI	Český normalizační institut
DA	MAD available – dostupnost MAD
DES	Data Encrypted Standard – standard pro šifrování dat
DESELECT	příkaz: „zrušení výběru“
EAL4+	Evaluation Assurance Level 4+ – úroveň jistoty ohodnocení 4+ (dle mezinárodního standardu Common Criteria ISO/IEC 15408)
EAL5+	Evaluation Assurance Level 5+ – úroveň jistoty ohodnocení 5+ (dle mezinárodního standardu Common Criteria ISO/IEC 15408)
EEPROM	Electrically Erasable Programmable Read-Only Memory – elektricky mazatelná semipermeabilní paměť typu ŘOM-RAM

GPB	General Purpose Byte – byte obecného použití
HALT	stav karty: „vypnutí“
LFSR	Linear Feedback Shift Register – posuvný registr s lineární zpětnou vazbou
LUID	Logické číslo karty
MA	Multi-application Card – multiaplikační karta
MAC	Message Authentication Code – autentizační kód zprávy
MAD	MIFARE Application Directory – adresářová aplikační struktura MIFARE čipů
MF3ICD40	označení první řady karet MIFARE DESFire
NITS	National Institute of Standards and Technology – Národní institut standardů a technologie
PIN	Personal Identification Number – osobní identifikační číslo
PPS	Protocol and Parameter Selection – příkaz: „výběru protokolu a parametrů“
QR kód	Quick Response – kód rychlé reakce
RATS	Request for Answer to Select – příkaz: “žádost o odpověď na SELECT“
REQA	Request Answer – příkaz: „žádost o odpověď“
RF	Radio Frequency – rádiová frekvence
RFID	Radio Frequency Identification – radiofrekvenční systém identifikace
RFU	Reserved for Future Use – vyhrazeno pro použití v budoucnu
SAK	Select Acknowledge – příkaz “potvrzení výběru“
SAM	Security Access Module – bezpečnostní přístupový modul, modul poskytující bezpečné uložiště klíčů a provádějící kryptografické operace umístěný na terminálu
SELECT	příkaz: “výběr“
SIM karta	Subscriber Identity Module – účastnická identifikační karta
UID	Unique Identifier– jedinečný identifikátor
WUPA	Wake Up – příkaz „probuď se“ ze stavu HALT

Úvod

Datová komunikace je v dnešní době běžným způsobem přenosu informací, a ne jinak tomu je v dopravě. Jak dopravci, tak samotní cestující stále častěji přistupují k možnosti bezkontaktního odbavení za využití technologie bezkontaktních čipových karet. Karty jsou využívány nejenom jako elektronické peněženky, které přinášejí zrychlení procesu platby a omezení přímého kontaktu řidiče s penězi v hotovosti. Slouží také jako nosiče jízdních dokladů a důležitých dat týkajících se držitele a jeho nároku na slevu či jinou službu. Jejich přínosem je celkové zrychlení a zefektivnění procesu odbavení, podpora automatizace a flexibility v rámci nabízených služeb i funkčnosti integrovaných dopravních systémů, zvýšení uživatelské přívětivosti a vytvoření lepšího mechanismu zabezpečení proti falšování jízdních dokladů, krádežím hotovosti a podobně.

Přestože již samotné používání bezkontaktních karet přineslo velký pokrok v procesech odbavování cestujících, snaha o další jejich zefektivňování je neustále na pořadu dne. Jedna z cest, jak toho dosáhnout, je optimalizovat datovou komunikaci probíhající mezi kartou a odbavovacím terminálem. Aby mohla být takováto optimalizace úspěšně provedena, je zapotřebí plně porozumět celému optimalizovanému procesu – od znalosti základních prvků a funkcí, jejich vzájemného propojení a nastavení jednotlivých parametrů, přes pochopení průběhu vnitřních pochodů, až po definování vlivů vnějších faktorů. Teprve poté, co je proces podrobně znám, mohou v něm být provedeny úpravy jednotlivých prvků, vazeb a parametrů tak, aby výsledkem byly požadované vlastnosti.

Hlavním úkolem této práce je ukázat na příkladu bezkontaktní multiaplikační karty konkrétního typu proces optimalizace datové komunikace mezi kartou a terminálem pro účely využití v oblasti odbavení cestujících. Pro tuto demonstraci byly zvoleny celosvětově nejčastěji používané čipové karty MIFARE, konkrétně zástupci jejich nejstarší a nejnovější řady MIFARE Classic a MIFARE DESFire. Tyto řady byly zvoleny pro názorné ukázání síly technologického vývoje, jeho přínosu pro reálné využití v každodenním životě a předvedení rozdílných možnosti aplikace optimalizačních procedur na procesy navržené dle odlišných základních principů.

Teoretická část nejprve ve stručnosti popisuje účel odbavovacích systémů v dopravě a specifikuje v nich úlohu bezkontaktní čipové karty jako nosiče jízdního dokladu, poté se již plně orientuje na její konkretizaci a podrobný popis MIFARE technologie. Nejprve uvádí obecné vlastnosti této technologie, následně konkretizuje vlastnosti a vyzdvihuje rozdíly řešených řad MIFARE Classic a MIFARE DESFire.

Stejně jako v jiných odvětvích i v oblasti bezkontaktních čipových karet je klíčem k úspěchu neustálý technologický vývoj reagující na aktuální možnosti a požadavky trhu, kontinuální

zlepšování starých postupů, vytváření a přijímání principů a postupů nových. Technologie MIFARE je vystavěna jak na standardech mezinárodně uznávaných, podporujících interoperabilitu zařízení různých výrobců, tak na standardech vlastních, odlišující produkt od konkurence a tím mu dávající přidanou hodnotu. Progresivní vývoj je u ní názorně vidět na přechodu od původně používaného proprietárního přenosového protokolu MIFARE a s ním spojeného kryptografického algoritmu CRYPTO1 k mezinárodním standardům ISO/IEC 14443-4, ISO/IEC 7816-4 a šifrovacím algoritmům DES, 3DES a AES aplikovaným v nejnovějších řadách. Druhou zásadní změnou, kterou tato technologie prodělala je přetvoření pevně dané blokové struktury vnitřní paměti EEPROM ve flexibilní souborový systém umožňující efektivnější návrh aplikací uložených na kartě.

Praktická část této práce ukazuje již konkrétní postup optimalizace datových struktur na kartách MIFARE Classic a MIFARE DESFire. V úvodu této části jsou stanoveny zásady tvorby optimální struktury aplikací, kterými je potřeba se řídit, aby bylo dosaženo požadovaných výsledků při nasazení multiaplikační karty do reálného provozu. Dále je dle těchto principů navrženo šest aplikací pro využití během procesů poskytování služeb v oblasti odbavovacích systémů v dopravě. Struktura těchto aplikací je primárně vytvořena pro parametry paměti EEPROM karet MIFARE DESFire, následně je upravena pro možnosti karet MIFARE Classic. Na zřetel je brána kromě samotné struktury dat také potřeba zabezpečení dle úrovně jejich citlivosti.

V závěru jsou diskutována fakta, společně s jejich konečnými důsledky, zjištěná v průběhu zpracovávání této práce a stanoveny návrhy na využití získaných poznatků v praxi.

I. TEORETICKÁ ČÁST

1 Odbavovací systémy v dopravě

Odbavovací systémy jsou používány v prostorech, kam je přístup zákazníků zpoplatněn dle daného ceníku – jedná se mimo jiné o muzea, zoologické zahrady či prostředky osobní veřejné dopravy. V posledním zmíněném případě přeprava cestujících z výchozího bodu do cílového je podmíněna zaplacením jízdného. Zaplacení jízdného ale není jediná součást odbavení cestujících.

1.1 Odbavení cestujících

Z hlediska procesního je odbavení cestujících množina procesů (posloupnost událostí) probíhajících nad odbavovacím systémem. Do této množiny spadá:

- tvorba tarifů, jízdních řádů, ceníků apod.,
- výběr spojení,
- zadání požadavku na jízdní doklad, rezervaci atp.,
- platba,
- výdej jízdního dokladu,
- ověření jízdního dokladu,
- informování cestujících během cesty,
- zpracování dat z prodeje.

Konkrétní průběh procesu odbavení je ovlivněn příslušným tarifem, způsobem platby a nosičem jízdního dokladu. [1]

Z hlediska systémového bereme tuto problematiku jako jeden veliký celek – systém odbavení cestujících. Přestože existuje mnoho různých definic systému, pro účely této práce je plně dostačující uvést jednoduchou definici systému S:

„S := {množina prvků a jejich funkcí, množina vazeb mezi nimi a parametrů na nich, množina procesů na i v systému a jeho výsledné (cílové) chování}“ [2]

Pokud je tedy systém definován takto, na určité rozlišovací úrovni tvoří množinu prvků odbavovacího systému Back Office, Front Office, prvky komunikační infrastruktury, prvek nosiče jízdního dokladu a rozhraní. Tyto prvky mají své funkce – vytváření ceníků, prodej jízdenek, informování cestujících apod. – a jsou vzájemně spojeny vazbami určitých parametrů – cena jízdenky je dána ceníkem, cestující je informován o cenách jednotlivých cest, apod. Tento systém pokrývá procesy odbavení cestujících a jako jeho cílové chování může být uvedeno mimo jiné přemístění cestujících z bodu A do bodu B za odpovídající cenu. Systém

má také své okolí, do něj v tomto případě patří například systémy rozúčtování tržeb, věrnostní systémy či systémy pro informování cestujících. [1]

1.2 Jízdní doklad

Potvrzením o zaplacení jízdného a tudíž i nároku na přepravu je pro cestujícího jízdní doklad. Jízdní doklady rozlišujeme podle druhu jejich nosiče a podle toho, zda jsou čitelné přímo nebo přístrojově. Kombinací těchto aspektů získáváme následující dělení.

- Papírový jízdní doklad:
  - přímo čitelný
    – přímo vytištěny informace o zakoupeném jízdném,
    – pro zabránění zneužití či vytvoření falsifikátu jsou použity fyzické prvky dokladu: vodotisk, hologram, gilošové prvky apod.;
  - strojově čitelný
    – na dokladu je vytištěn QR, čárový či jiný strojově čitelný kód, který obsahuje informace o zakoupeném jízdném,
    – uložené informace jsou zde chráněny použitím šifrovacích kódů, kontrolních součtů apod.
- Elektronický jízdní doklad:
  – strojově čitelný – informace o zakoupeném jízdném jsou uloženy v digitální podobě,
  – používanými paměťovými médii jsou:
  - bezkontaktní čipová karta
    – pro ochranu uložených informací jsou použity kryptografické klíče a přístup k jednotlivým aplikacím na kartě je omezen na základě definovaných přístupových práv;
  - mobilní telefon používající rozhraní NFC
    – uložené informace jsou také zabezpečeny použitím kryptografických klíčů a omezením přístupu na základě definovaných přístupových práv k jednotlivým aplikacím uloženým v paměti nebo SIM telefonu;
  - paměť mobilního telefonu, SMS/MMS kód elektronické jízdenky
    – pro zabezpečení informací bývá v tomto případě využita metoda kontrolních součtů nebo šifrování dat, další možností je zabezpečení skrz webovou aplikaci provozovatele služby.

[1]

Tato práce se dále zaměří na bezpečnostní aspekty bezkontaktních čipových karet pro účely odbavení cestujících.

2 Bezkontaktní čipové karty v odbavovacích systémech

Bezkontaktní čipová karta je používána odbavení cestujících, kdy informace o zakoupeném jízdném jsou ukládány/zapsány na kartu bez přímého dotyku se zařízením k tomu určeným a stejně tak probíhá i jejich získávání/čtení z karty. Tato karta je obecně tvořena čipem zalisovaným společně s jeho anténou v plastovém těle. Zalisovaný čip uchovává zapsané informace a musí být schopen bezkontaktní komunikace. Anténa čipu je tvořena cívkou s určitým počtem a velikostí závitů, tyto její parametry udávají kmitočet, při kterém probíhá přenos dat a karta je v aktivním stavu.

Čtečka – zařízení určené pro zápis a čtení informací na a z karty – vytváří kolem sebe elektromagnetické pole určitého kmitočtu. Pokud kmitočet pole čtečky odpovídá kmitočtu antény karty a ta se nachází v tomto poli, může mezi nimi probíhat komunikace.

Setkáváme se s mnoha různými technologiemi využívaných v oblasti bezkontaktních čipových karet, jako jednoho z jejich nejvíce rozšířených zástupců můžeme jmenovat například karty se zabudovaným RFID čipem. V tomto čipu je natrvalo uloženo jeho UID, zatím co ostatní data v něm mohou být opakovaně přepisována. Na obdobném principu jako FRID technologie fungují i jiné velmi rozšířené karty, jejichž čip navíc umožňuje šifrování zapsaných dat a fungování více aplikací na jedné kartě současně. Takovéto čipy obsahují například karty MIFARE společnosti NXP, které jsou často využívány nejenom v odbavovacích systémech. [1]

2.1 Technologie karet MIFARE

Bezkontaktní čipové karty MIFARE celosvětově poskytují více jak 40 různorodých, různě náročných, aplikací z oblasti prokázání identity či nároku na určitou službu, uložení specifických dat, peněžních transakcí apod. Jedná se mimo jiné o identifikační, přístupové, věrnostní, zákaznické nebo parkovací karty.

Již nějaký čas neplatí, že jedna karta poskytuje právě jednu službu, technologie MIFARE upravila strukturu paměti a vznikla multiaplikační karta poskytující držiteli několik aplikací najednou. Dále tato technologie přináší vysokou přenosovou rychlost 106-848 kbit/sek, a bezpečné uložení i bezkontaktní přenos dat použitím šifrovacích algoritmů, což vede k rychlému a bezpečnému provedení transakce bez potřeby přímého dotyku karty se čtecím terminálem, avšak ve vzdálenosti maximálně 100 mm od něj. Vedle bezkontaktního přenosu dat zde probíhá i bezkontaktní přenos energie, díky čemuž je karta napájena elektromagnetickým polem čtečky a nepotřebuje tak vlastní zdroj energie.

Karty MIFARE jsou vyvíjeny déle jak dvě desetiletí. Za tu dobu bylo postupně vytvořeno několik řad: MIFARE Classic, Plus, DESFire, Ultralight a SmartMX. Všechny tyto řady se řídí mezinárodními standardy ISO/IEC 7810 a ISO/IEC 14443 A, zároveň jsou spolu navzájem kompatibilní. [3]

2.1.1 MIFARE podle standardů

Technologie MIFARE je vystavěna na základě několika standardů specializujících se na bezkontaktní čipové karty. Využívá jak svých vlastních standardů vytvořených speciálně pro karty společností NXP, tak standardů mezinárodně uznávaných podporujících kompatibilitu s cizími zařízeními.

2.1.1.1 ISO/IEC 7810

Rozměry a další **fyzické vlastnosti** karet MIFARE se řídí standardem ISO/IEC 7810. Podle této normy, která udává čtyři základní kategorie dle rozměrů, karty MIFARE společně s většinou identifikačních a platebních karet spadají do kategorie ID-1, která stanovuje rozměry karty na 85,60 mm X 53,98 mm. Jejich tloušťka je, stejně jako u ostatních kategorií, 0,76 mm. Viz Obr. 1. [4]

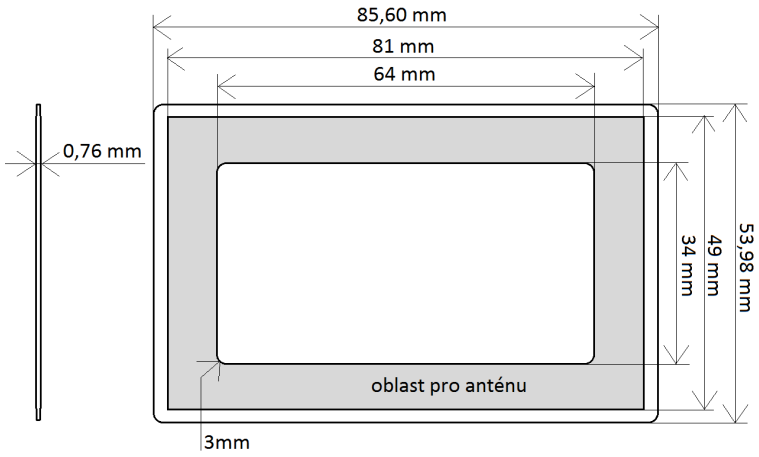
2.1.1.2 ISO/IEC 14443

V oblasti **komunikace a přenosu dat** jsou karty MIFARE v souladu se standardem pro identifikační karty, bezkontaktní karty s integrovanými obvody a karty s vazbou na blízko ISO/IEC 14443. Ten rozlišuje podle komunikačního rozhraní dva typy karet, A a B. Produkty MIFARE jsou navrženy pro rozhraní A [5], jehož důležité body jsou popsány níže.

ISO/IEC 14443 se skládá ze čtyř částí, avšak všechny karty MIFARE jsou kompatibilní jen s prvními třemi částmi, čtvrtou část podporují v současné době pouze řady MIFARE DESFire, MIFARE Plus třetí úrovně zabezpečení a karty NXP s dvojitým či trojitým rozhraním (karty SmartMX apod.). Jednotlivé části jsou:

- Část 1: Fyzikální charakteristiky.
- Část 2: Radiofrekvenční výkonové a signální rozhraní.
- Část 3: Inicializace a antikolize.
- Část 4: Protokol přenosu. [5]

První část, **ISO/IEC 14443-1**, definuje vlastnosti a umístění antény čipu uvnitř karty. Oblast, kde je anténa vedena a de facto tvoří cívku, je ohraničena dvěma obdélníky. Rozměry vnitřního obdélníku jsou 64x34 mm s poloměrem zaoblení rohů 3 mm. Vnější obdélník má rozměry 81x49 mm. Názorně je toto ukázáno na Obr. 1. [6] [5]



Obr. 1: Bezkontaktní čipová karta – rozměry dle ISO/IEC 7810 [4], oblast pro umístění antény dle ISO/IEC 14443-1 [6].

Podle druhé části, I**SO/IEC 14443-2,** jsou nastaveny parametry obousměrné komunikace mezi kartou a čtečkou. Tento dialog probíhá v radiofrekvenčním poli čtečky s frekvencí 13.56±0,007 MHz a intenzitou 1,5-7,5 A/m. Karta musí být schopna v takovémto poli nepřetržitě pracovat. Pokud v poli není žádná karta, čtečka je ve stavu „spánku“. Modulace a kódování dat během jejich přenosu rychlostí 106 kbit/s je u karet typu A prováděna 100% ASK modulací modifikovanou Millerovým kódováním ve směru od čtečky ke kartě a zátěžovou modulací podpořenou OOK modulací a kódováním Manchester ve směru opačném. [6]

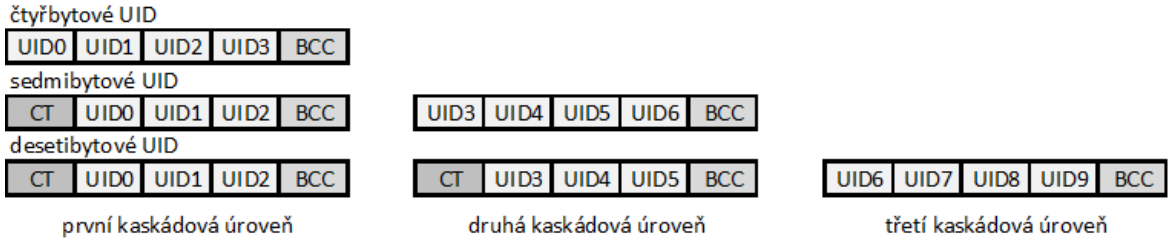
Třetí část. **ISO/IEC 14443-3**, popisuje proces aktivace karty čtečkou zahrnující nejprve průběh antikolizní smyčky a následný výběr právě jedné karty. Tento proces je navržený tak, aby fungoval nezávisle na počtu přítomných karet v poli nebo počtu různých aplikacích na kartě/kartách. [7]

Jak je zmíněno výše, pokud se v poli čtečky karta nevyskytuje, čtečka je v neaktivním módu, když se karta přiblíží, čtečka se „probudí“ a může započít komunikaci. Sled událostí během úvodního dialogu probíhá následovně:

1. Aktivace karty radiofrekvenčním polem čtečky.
2. Karta v klidu vyčkává na obdržení příkazu od čtečky.
3. Přenos příkazu od čtečky – **REQA**.
4. Přenos odpovědi od karty – **ATQA**.

Jelikož v poli čtečky se může najednou vyskytovat více karet – jedná se o okolí do vzdálenosti 10 cm –, všechny tyto karty obdrží REQA a každá z nich odpoví ATQA. Čtečka zpracuje pouze první obdrženou odpověď, která je pro ní důkazem, že se v poli vyskytuje minimálně jedna karta. Aby zajistila spojení pouze s jednou kartou, spouští antikolizní smyčku. [5]

Antikolizní smyčka funguje na principu získávání celého jedinečného UID karty. Standard ISO/IEC 14443-3 definuje tři různé velikosti UID používané během procesů antikolizní smyčky a výběru karty – čtyř, sedmi nebo desetibytový identifikátor. Jako první byly používány nejmenší UID, vzhledem k omezenému počtu použitelných kombinací – přibližně 3,7 miliard – se začaly využívat také sedmibytové UID. Podle citovaného zdroje karty s desetibytovým UID zatím používány nejsou (rok 2013), ale jmenovaný standard vyžaduje, aby čtečky byly schopné s nimi pracovat. Struktura těchto identifikátorů je názorně zobrazena na Obr. 2, kde UIDX znázorňuje X-tý identifikační byte, BCC kontrolní znak odpovídajícího bloku a CT kaskádový tag značící kaskádovou úroveň následujícího bloku. [8]



Obr. 2: Struktura UID dle ISO/IEC 14443

UIDX: X-tý identifikační byte, BBC: kontrolní byte daného bloku, CT: kaskádový tag [8]

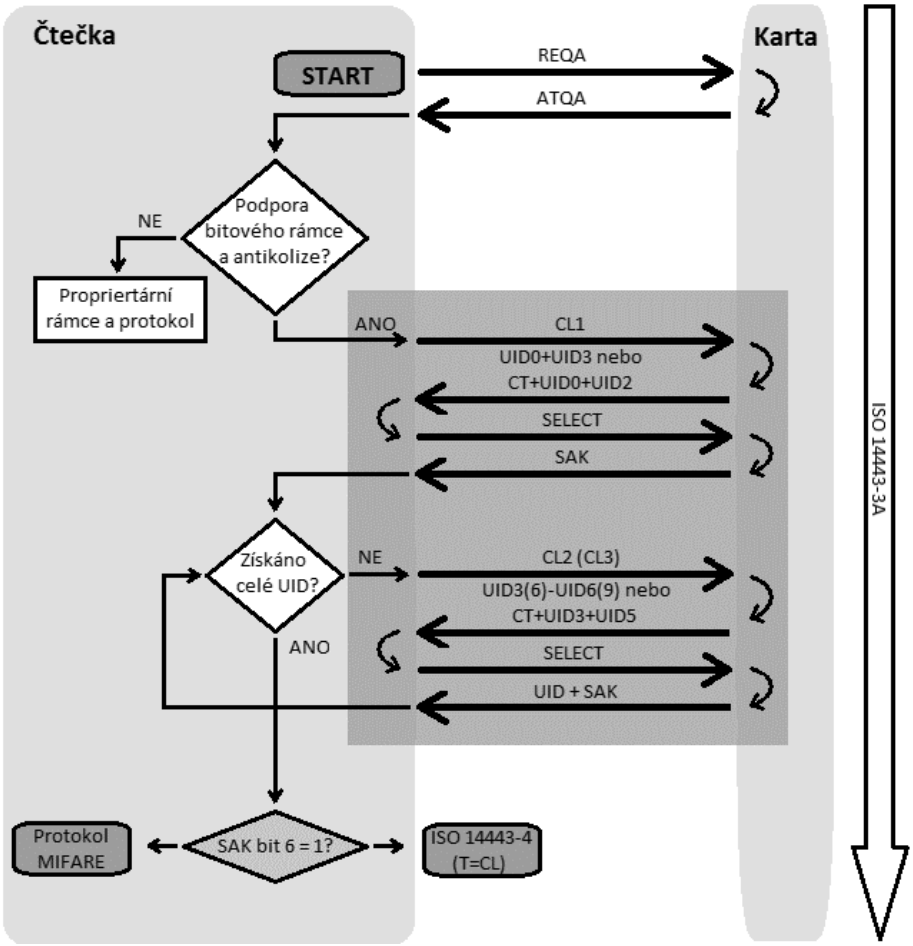
Vrátíme-li se k antikolizní smyčce, čtečka nejprve vyšle příkaz CL1, na který všechny karty v poli odpoví byty z první kaskádové úrovně. Když odpovídá více jak jedna karta, dojde ke kolizi odpovědí. Čtečka zaznamená pozici prvního bitu, kde ke kolizi došlo a zaznamená přijatou část UID – bity předcházející kolizní bit. Poté vyšle příkaz SELECT obsahující přijatou část UID. Karty tento příkaz zachytí a zkontrolují přijatou část UID. Karty, jejichž UID odpovídá přijatému, odpoví čtečce zbytkem UID. Takto se pokračuje až do doby, kdy čtečka

a) obdrží kaskádový tag (0x88) – znak toho, že obdržené UID není kompletní – a vyšle příkaz CL2, aby karty pokračovaly v posílání dalších bytů druhé, případně třetí, kaskádové úrovně,
b) získá pouze jednu odpověď s kompletním UID.

Po získání kompletního UID čtečka ještě jednou vyšle povel SELECT obsahující celé UID a CRC kód. Karta s odpovídajícím UID odpoví příkazem SAK a zároveň přejde do aktivního módu – v případě karet podporující ISO/IEC 14443-4 se tím rozumí správné nastavení přenosového protokolu.

Na základě šestého bitu obdrženého SAK čtečka zjistí, jestli daná karta je typu podporujícího ISO/IEC 14443-4 nebo není. Jestliže na této pozici je “0“, čtečka ví, že komunikuje s kartou podporující MIFARE Protokol. Poté, na základě bitů na ostatních pozicích SAK odpovědi, určí, zda se jedná o kartu MIFARE Classic 1K či 4K, MIFARE Ultralight

či Ultralight C, MIFARE Mini nebo MIFARE Plus první či druhé úrovně zabezpečení [7]. Pokud je na pozici šestého bitu SAK "1", jedná se o kartu podporující ISO/IEC 1444-4. Celý tento proces je znázorněn na Obr. 3. [5][8]



Obr. 3: Proces započetí komunikace čtečky a karty – vyslání prvního dotazu, antikolizní smyčka a zjištění protokolu následné komunikace; UIDX: X-tý identifikační byte, BBC: kontrolní byte daného bloku, CT: kaskádový tag [5]

Jelikož to hlavní, o co při navázání kontaktu s kartou systému (čtečce) jde, je získat určité informace, je zapotřebí, aby čtečka navázala komunikaci s kartou, která tyto informace opravdu má. Obecně jde o to, že terminál nejprve musí postupně navázat kontakt s každou kartou přítomnou v poli a zjistit, jaké aplikace nese. Kartu, u které objeví pouze cizí aplikace, vyřadí z výběru. Když nalezne kartu s danou aplikací, zaznamená ji a pokračuje v prohledávání ostatních karet. Když objeví v poli více karet s danou aplikací, systém buď tyto karty upřednostní, nebo transakci zamítá.

Uvědomme si, že se jedná v podstatě o kulovitý prostor poloměru 10 cm, tudíž v něm nebude přítomno příliš mnoho karet, natož karet se stejnou aplikací. Názorným příkladem je případ několika karet v jedné peněžence. Pokud jsou v peněžence pouze karty s různými aplikacemi, je možné přiložit celou peněženku ke čtečce a proces proběhne v pořádku. Ovšem, pokud

v ní jsou karty se stejnými aplikacemi nebo nastane jiný neočekávaný problém, proces je předčasně ukončen a je lepší odpovídající kartu z peněženky vyndat a proces zopakovat pouze s ní.

Konkrétní postupy fungování tohoto procesu pro karty podporující ISO/IEC 14443-4 a naopak pro karty ho nepodporující jsou uvedeny v odpovídajících pasážích níže. Některé typy karet podporují obě varianty a automaticky se zapojují do procesu ve stavu odpovídajícím aktuálnímu režimu fungování čtečky. [7]

Čtvrtá část, **ISO/IEC 14443-4**, popisuje asynchronní blokově orientovaný přenosový protokol T=CL pracující na principu poloviční duplexu a definuje aktivační a deaktivační sekvenci karty. Stanovuje příkazy, jejich parametry a pravidla použití, velikosti a strukturu datových rámců, časový průběh přenosu apod. [6]

Příkladem příkazů, které tento protokol používá, jsou:

RATS – identifikuje kartu čtečce jako typ podporující ISO/IEC,

PPS – umožňuje individuální nastavení přenosové rychlosti mezi kartou a čtečkou,
– karty MIFARE DESFire umožňují nastavení různé přenosové rychlosti ve směru od čtečky ke kartě a od karty ke čtečce

DESELLECT
– uvádí kartu do stavu HALT – „zastavuje kartu“ [9]

Proces výběru jedné karty obsahující určitou aplikaci začíná nastavením parametru N = 0 označujícího počet nalezených karet s požadovanou aplikací. Poté čtečka naváže komunikaci s první kartou – viz ISO/IEC 14443-3. Čtečka vyšle příkaz RATS pro ujištění se, že karta podporuje ISO/IEC 14443-4 – očekávána je odpověď ATS. Pokud tato odpověď nedorazí, čtečka kartu reaktivuje a nechá ji příkazem DESELECT přejít do stavu HALT, čímž ji vyřadí z výběru. Když čtečka obdrží ATS, může, pokud to situace vyžaduje, změnit parametry komunikace vysláním příkazu PPS. Poté si vyžádá od karty seznam aplikací, které obsahuje – viz kapitola 2.1.2.2, adresář aplikací MAD3. Když danou aplikaci v seznamu objeví, použije požadované přístupové klíče, aby došlo k autentizaci – více viz 2.1.3.1. Pokud aplikace na seznamu není nebo proces autentizace je neúspěšný, karta je také vyřazena z výběru a přepnuta do stavu HALT. Když naopak oprávnění k přístupu k aplikaci je potvrzeno, navýší se parametr N = N + 1. Pokud je poté parametr N = 1, je použit příkaz DESELECT a započat výběr další karty v poli. Jestliže jsou zkontrolovány všechny karty v daném okolí a N je stále nulové, žádná karta danou aplikaci neobsahuje, a tudíž požadovaný proces nemůže proběhnout. V případě, že N = 1, dojde k reaktivaci karty obsahující danou aplikaci, opětovnému použití příkazu RATS, případně i PPS, proběhne i opětovná autentizace

a aplikace je spuštěna. Pokud v nějakém okamžiku tohoto celého procesu nastane N > 1, nastává chyba a transakce je zamítnuta – v poli je více karet obsahujících danou aplikaci.

Některé aplikace vyžadují pracovat s informacemi z více karet najednou. Tehdy ISO/IEC 14443-4 povoluje současnou komunikaci čtečky s několika kartami zároveň za použití jejich jedinečného UID pro aktivaci a adresaci jednotlivých karet během transakce – karty nejsou „vypínány“. Přesto, že toto standard povoluje, je doporučováno komunikovat s kartami postupně, tedy aktivovat jednu kartu, provést potřebné úkony, správně ji deaktivovat, vybrat druhou kartu (aktivovat, provést úkony, deaktivovat) – takto postupně vybrat všechny potřebné karty – a zase se třeba vrátit k dříve použitým kartám.

Pro správný průběh procesů by se karta neměla dostat mimo pracovní pole čtečky. Zároveň je doporučováno, aby v aplikaci – na kartě či v terminálu – byly nastaveny obnovovací a zálohovací procedury pro případ přerušení transakce. [7]

2.1.1.3 ISO/IEC 7816-4

Dalším standardem zbývajícím se identifikačními kartami a kartami s integrovanými obvody je ISO/IEC 7816. Tento standard se, stejně jako ISO/IEC 14443, dělí na čtyři hlavní části zabývající se fyzickými a elektronickými vlastnostmi a přenosovým protokolem karet. Karty MIFARE DESFire dodržují jeho čtvrtou část s podtitulem „Organizace, zabezpečení a příkazy pro přenos dat“. [10][11]

ISO/IEC 7816-4 stanovuje:

a) obsah párů *příkaz–odpověď* vyměňovaných na rozhraní mezi kartou a čtečkou,
b) prostředky získávání datových prvků a objektů na kartě,
c) strukturu a obsah historických bytů pro popsání provozních charakteristik karty,
d) struktury pro aplikace a data na kartě, jak jsou viděny na rozhraní při zpracování příkazů,
e) metody přístupu k souborům a datům na kartě,
f) bezpečnostní architekturu definující přístupová práva k souborům a datům na kartě,
g) prostředky a mechanismy identifikace a adresování aplikací na kartě,
h) metody bezpečného posílání zpráv,
i) metody přístupu k algoritmům zpracovávaných kartou – nepopisuje tyto algoritmy.

Tento standard nabízí různé možnosti a obecné doporučení, jak daný problém řešit, ale neuvádí přesné vnitřní provedení, nepopisuje algoritmy, nespecifikuje aplikace ani neuvádí přesné bezpečnostní politiky. [11]

MIFARE DESFire konkrétně z něj přebírá:

1. strukturu APDU zpráv a
2. příkazy:
   - SELECT FILE,
   - READ BINARY,
   - UPDATE BINARY,
   - READ RECORDS,
   - APPEND RECORD,
   - GET CHALLENGE,
   - INTERNAL AUTHENTICATE,
   - EXTERNAL AUTHENTICATE. [10]

2.1.1.4 Standard MIFARE

Standard MIFARE je podporován kartami řady MIFARE Classik 1k, MIFARE Classic 4k, MIFARE Ultralight a MIFARE Plus první a druhé úrovně zabezpečení. Tyto karty po aktivaci nekomunikují se čtečkou podle přenosového protokolu, nýbrž aplikace sama spustí vlastní příkazy – například příkazy autentizace. [7]

Příkazy autentizace **Authentication with key A** nebo **Authentication with key B** jsou používány při procesu prokázání oprávnění k přístupu k určité aplikaci založeném právě na principu znalosti klíčů, které jsou buď veřejně známé, nebo naopak tajné – dle daných bezpečnostních požadavků na aplikaci – více viz kapitola 2.2.1.1.

Základní příkazy používané pro samotnou práci s daty jsou **Read** a **Write**, které přečtou, respektive zapíší data do paměti. Tyto příkazy mohou být aplikovány v jakémkoli typu bloku. Další příkazy jsou používané pouze při práci s hodnotovými bloky (viz kapitola 2.2.1.1). **Increment** a **Decrement** zvýší, případně sníží hodnotu uloženou v hodnotovém bloku a výsledek uloží do vnitřního datového registru. Příkaz **Transfer** poté zapíše obsah tohoto registru do hodnotového bloku, naopak příkaz **Restore** přečte obsah bloku a uloží ho do vnitřního registru. [9]

Při výběru karty s požadovanou aplikací dle standardu MIFARE systém nejprve nastaví parametr N = 0 značící počet nalezených karet nesoucích požadovanou aplikaci – shodně s ISO/IEC 14443-4 – a terminál naváže komunikaci s jednou z karet přítomných v poli – za použití antikolizní smyčky, viz ISO/IEC 1444-3 v kapitole 2.1.1.2. V případě, že karta používá adresář aplikací MAD – viz kap. 2.1.2.1 –, přečte ho a zkontroluje, zda je na kartě přítomna daná aplikace. Pokud karta nepodporuje MAD, spustí příkaz autentizace k požadované aplikaci. Když na kartě aplikace není (nebyla nalezena v MAD, případně

autentizace k aplikaci nebyla úspěšná), systém kartu reaktivuje – použije příkaz **WUPA** a **SELECT** – a převede do stavu HALT. V opačném případě zvýší hodnotu N na N = N + 1 a spustí rozhodovací proces, zda N = 1. Pokud tomu tak je, systém převede kartu do stavu HALT. Když je karta převedena do stavu HALT – nezávisle na tom, zda aplikaci obsahuje či ne – naváže čtečka komunikaci s další kartou přítomnou v poli a postupuje, jak bylo právě popsáno. Pokud N = 1 a v poli se již nevyskytuje žádná další karta, dojde k reaktivaci karty obsahující danou aplikaci a tato aplikace je spuštěna. V případě, že N > 1, je vyslána chybová hláška a transakce je zamítnuta – v oblasti je více jak jedna karta s danou aplikací. Pokud zůstalo N = 0, v okolí se nevyskytuje karta poskytující danou službu. [7] Blokové schéma tohoto procesu je zobrazeno v příloze A.

2.1.2 Adresář aplikací MAD

Karty jsou užívány pro různé aplikace (prokázání totožnosti, nároku na benefit, nákup jízdenky apod.), přičemž každá aplikace vyžaduje odlišné informace, které jsou zapsány na kartu (identifikační údaje, potvrzení nároku a specifikace benefitu, informace o zaplacení). Pro správné a efektivní přečtení a vyhodnocení uložených dat systémem je nutné, aby data byla uložena ve správném formátu a v definované struktuře. Navíc by data měla být chráněna proti úmyslnému i neúmyslnému poškození, zničení či zneužití.

První karty byly nosiči pouze jedné aplikace, tudíž v celé EEPROM byla uložena, s výjimkou základních informací o kartě, pouze data této aplikace, pro které byla volně k dispozici celá její předdefinovaná struktura. S přibývajícími oblastmi využití se toto řešení ukázalo poněkud nepraktické a začalo se uvažovat o tom, že by jedna karta mohla poskytovat více aplikací najednou. Na trhu se objevily multiaplikační karty. Zde už uložení všech dat dohromady nebylo vhodné, protože by bylo zaprvé velice obtížné se v nich orientovat a určit, která data patří ke které aplikaci, a zadruhé nastavení přístupových práv by bylo neflexibilní. Byla tedy určena pravidla, podle kterých se daná struktura paměti začala rozdělovat mezi jednotlivé aplikace tak, aby jejich data byla od sebe jasně odlišena.

Jednotnou strukturu datových záznamů v adresáři aplikací na MIFARE kartě zavádí Standard MIFARE Application Directory – **MAD**. Nejen že zajišťuje správný výběr konkrétní aplikace na jedné kartě, navíc umožňuje výběr správné aplikace, když je v blízkosti čtečky více karet s různými aplikacemi – například, když máme v peněžence zároveň studentskou kartu a zákaznickou kartu nějakého dopravce, nemusíme požadovanou kartu vyndávat, ale celou peněženku přiložíme k terminálu.

Podle typu MAD můžeme karty dělit na:

- monoaplikační karta bez položek v adresáři,
- monoaplikační karta obsahující položky v adresáři,
- multiaplikační karta s položkami adresáře.

V současnosti jsou používány tři verze MAD:

- MAD1 – karty s pamětí rozdělenou do 16 sektorů – MIFARE Classic 1K,
- MAD2 – karty s pamětí > 1 kB –.MAFARE Classic 4K, MIFAREPro, MIFAREProX,
  – plně kompatibilní s MAD1,
- MAD3 – specifikuje užití registrovaných identifikátorů aplikace v rámci MIFARE DESFire

Jedná se v podstatě o adresář (seznam) aplikací uložených na dané kartě, jehož prohledáním systém – čtečka – jednoduše zjistí, zda hledaná aplikace na kartě je či není.

2.1.2.1 MAD sektor

Adresář aplikací je v na kartách pracujících podle MAD1 uložen v sektoru 0x00, takzvaném **MAD sektoru,** obsahujícím jedinečný identifikátor aplikace (dále AID) a adresu uložení všech aplikací na kartě. V případě MAD2 se jedná o dva MAD sektory 0x00 a 0x10. Poslední verze, MAD3, na to jde trošku jinak, karta automaticky na vyžádání vytváří seznam aplikací aktuálně uložených na kartě reprezentovaných jejich MAD3 AID.

MAD sektor, stejně jako jiné sektory (viz kap. 2.2.1.1), se dělí na čtyři bloky po 16 bytech. V těchto bytech jsou zapsány AID aplikací uložených na kartě, jeden informační byte a jeden CRC byte. Blok tři je zavaděčem sektoru určujícím přístupová práva k němu. Jeho struktura pro sektor 0x00 je zobrazena na Obr. 4 a pro sektor 0x10 na Obr. 5.

[12]

	byte 15	byte 14	byte 13	byte 12	byte 11	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
Blok 0	B	l	o	k	V	ý	r	o	b	c	e	-	k	ó	d	
Blok 1	AID sek.0x07		AID sek.0x06		AID sek.0x05		AID sek.0x04		AID sek.0x03		AID sek.0x02		AID sek.0x01		info	CRC
Blok 2	AID sek.0x$F		AID sek.0x$E		AID sek.0x$D		AID sek.0x$C		AID sek.0x$B		AID sek.0x&A		AID sek.0x09		AID sek.0x08	
Blok 3	Z	a	v	á	d	ě	c	í	B	l	o	k	0	x	0	0

Obr. 4: Struktura MAD sektoru 0x00 [12]

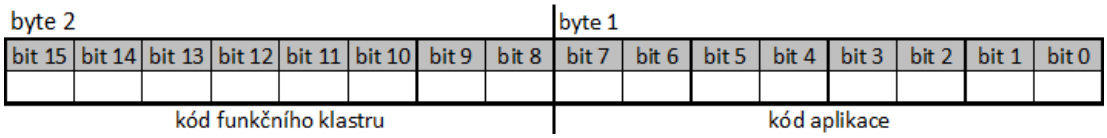
	byte 15	byte 14	byte 13	byte 12	byte 11	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
Blok 0	AID sek.0x17		AID sek.0x16		AID sek.0x15		AID sek.0x14		AID sek.0x13		AID sek.0x12		AID sek.0x11		info	CRC
Blok 1	AID sek.0x$1F		AID sek.0x$1E		AID sek.0x$1D		AID sek.0x$1C		AID sek.0x&1B		AID se. 0x&1A		AID se. 0x19		AID sek.0x18	
Blok 2	AID sek.0x27		AID sek.0x26		AID sek.0x25		AID sek.0x24		AID sek.0x23		AID sek.0x22		AID sek.0x21		AID sek.0x20	
Blok 3	Z	a	v	á	d	ě	c	í	B	l	o	k	0	x	1	0

Obr. 5: Struktura MAD sektoru 0x10 [12]

Identifikátor aplikace AID

Každá aplikace je v rámci karty označena jedinečným AID, díky kterému ji můžeme přesně identifikovat, specifikovat její typ i určit sektor, ve kterém je uložena.

Pro MAD1 a MAD2 je to dvoubytový identifikátor rozdělen na dvě části. Prvních osm bitů (0–7) určuje **kód aplikace**. Druhých osm bitů (8–15) zaplňuje **kód funkčního klastru**, kam daná aplikace patří – druh aplikace –, díky němu se mohou aplikace jednoduše třídit. Tyto kódy jsou již dané a při tvorbě nové aplikace by měla být vybrána skupina nejlépe charakterizující aplikaci. Struktura dvoubytového AID je zobrazena v Obr. 6, výběr některých funkčních klastrů v Tab. 1.



;Obr. 6: Struktura AID pro karty podporující MAD1 a MAD2 [12]

Tab. 1: Kódy vybraných funkčních klastrů [12]

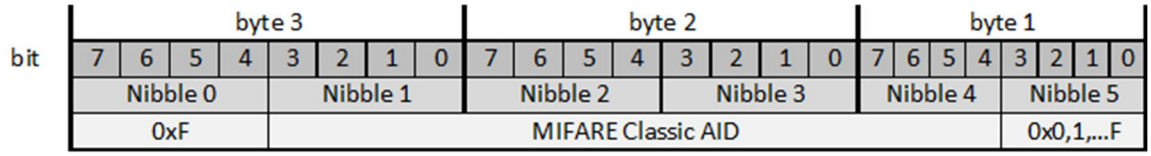
kód klastru	Funkce - druh aplikace
0x00	Administrativní kód
0x08	Aerolinky
0x10	Železniční doprava
0x18	Městská doprava
0x19	České železnice
0x20	Autobusová doprava
0x21	Multimodální tranzit

Pokud je kód funkčního klastru 0x00, jedná se o speciální **administrativní kód** daného sektoru a celé AID označuje jeho specifický stav:

- 0x00 00 sektor je volný,
- 0x00 01 závada sektoru – např. přístupové klíče jsou neznámé,
- 0x00 02 sektor je rezervovaný,
- 0x00 03 sektor obsahuje další informace o adresáři – RFU,
- 0x00 04 sektor obsahuje informace o držiteli katy v ASCII formátu,
- 0x00 05 sektor je nepoužitelný kvůli velikosti paměti.

V případě standardu MAD3 je AID definováno ve chvíli vytváření aplikace a zabírá tři byty paměti. Seznam aktuálně instalovaných aplikací na kartě si tvoří samotná karta, a tedy si vydavatel nemusí vést jejich evidenci. K získání seznamu se použije příkaz GetApplicationAIDs, který by měl být povolený Master klíčem karty bez autentizace žadatele.

Při jeho tvorbě může být využita struktura AID podle MAD1 a MAD2. V tom případě dané tři byty rozdělíme do šesti malých čtyřbitových částí, takzvaných „nibblů“. Nibbl nula označuje použití MAD1 (MAD2) struktury uložením hodnoty 0xF. V následujících čtyřech nibblech (1–4) je zapsáno 16bitové MIFARE Classic AID. V posledním nibblu (část 5) jsou čtyři bity, které mohou být libovolně zaplněny – tak, aby celkové AID bylo jedinečné. Díky tomuto nibblu můžeme z jediného MIFARE Classic AID získat 16 unikátních MIFARE DESFire AID. Tříbytové AID je naznačeno v Obr. 7.



Obr. 7: Struktura AID pro karty podporující MAD3 [12]

MAD3 AID identifikátor 0xFF FF FF je vyhrazen pro informace vydavatele karty:

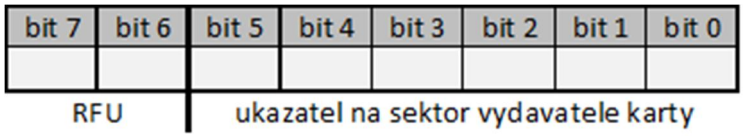
- soubor 0x0 – hodnotový soubor volně přístupný pro příkaz GetValue,
  – zapsáno: 0 x 00 00 03 – určuje verzi MAD3;
- soubor 0x1 – standardní datový soubor volně přístupný pro čtení,
  – kontaktní informace o držiteli karty zapsány v prostém textu v jednoduchém souborovém formátu CSV;
- soubor 0x2 – standardní datový soubor volně přístupný pro čtení,
  – kontaktní informace o vydavateli karty zapsány v prostém textu v jednoduchém souborovém formátu CSV;
- soubor 0x3 až 0XF
  – RFU,
  – neměly by být použity v této aplikaci,
  – měly by být ignorovány softwarem ve čtečce.

CRC byte

V MAD sektoru je na pozici nultého bytu bloku 0 (bloku 1 v sektoru 0x10) zapsán cyklicky redundantní osmibitový kód. Jedná se o tzv. CRC byte zajišťující kontrolu integrity bloků daného sektoru, která by měla být kontrolována pokaždé, když je MAD čten. CRC bity jsou počítány i kontrolovány CRC koprocesorem zabudovaným v čipu. Výpočet i kontrola kódu jsou prováděny na základě informačního bitu a bytů ID1 až ID$F, případně informačního bytu a bytů ID$11 až ID$27. Je důležité, aby byl vždy zpracován nejprve nižší byte a poté až vyšší. Pro správný výpočet by také měl být koprocesor pokaždé nejprve restartován, aby v něm nezůstávaly nějaké staré hodnoty.

Informační byte a vydavatel karty

Na pozici vedle CRC bytu je uložen informační byte, jehož první čtyři bity ve verzi MAD1 ukazují na sektor vydavatele karty. V případě MAD2 na tento sektor ukazuje prvních 6 bitů. Zbylé bity jsou RFU. Pokud na kartě sektor vydavatele není, použije se kód 0x00. Jeho struktura je znázorněna na Obr. 8.

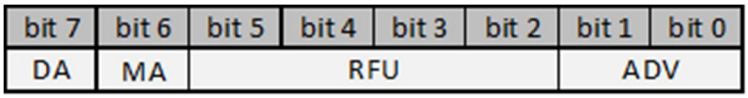


Obr. 8: Struktura informačního bytu v MAD sektoru [12]

Nastavení použití standardu MAD

Poslední ze čtyř přístupových bytů zaváděcího bloku sektoru 0x00 (byte devět) je tzv. **byte obecného použití (GPB)**, který umožňuje nastavení standardu MAD na dané kartě.

Když se na jeho strukturu podíváme odzadu, tak bit sedm (**DA**) udává, zda je na kartě MAD aktivní (1_b), či ne (0_b). Šestý bit (**MA**) určuje, zda se jedná o kartu s jednou (1_b), či více (0_b) aplikacemi. Následující čtyři bity jsou prozatím nevyužity a slouží jako RFU. Bity jedna a nula (**ADV**) označují použitou verzi MAD, MAD1 je značena 01_b a MAD2 10_b, viz Obr. 9.



Obr. 9: Struktura GPB bytu – DA: aktivita MAD, MA: jedna či více aplikací, RFU: zatím nevyužité místo, ADV: typ verze MAD na kartě [12]

Na kartách používajících MAD2 je v sektoru 0x10 GPB byte nastaven na hodnotu 0x00 značící RFU.

Přístupová práva k MAD sektoru

Přístupová práva k jednotlivým blokům sektoru MAD jsou definována v takzvaných **přístupových bitech** (6-8) třetího bloku sektoru. Je zde určeno, jaké klíče je potřeba znát, aby čtečka mohla přečíst či zapsat data z/do odpovídajícího bloku. [9]

Klíč A, tzv. **klíč pro čtení sektoru**, je uložen v prvních pěti bytech zavaděče sektoru. Je doporučováno, aby byl nastaven jako veřejný klíč, což uživatelům umožní bez problémů zjistit AID všech aplikací na kartě. Přestože to vede k myšlence, že tento klíč nemusí být vůbec nastaven, je doporučeno používání přednastaveného kódu **0xa0a1a2a3a4a5**.

Klíč B, tzv. **klíč pro zápis do sektoru**, je naprogramován výrobcem karty a měl by být tajný – řádně zabezpečen proti zneužití –, aby se zabránilo neoprávněnému přepsání

či smazání důležitých dat v sektoru. Neoprávněná manipulace s MAD sektorem může vést k tomu, že aplikace na kartě nabude nalezena, protože:

a) v adresáři je uvedena špatná adresa jejího uložení,
b) v adresáři vůbec není uvedeno dané AID nebo
c) v adresáři je zapsané AID aplikace, která na kartě vůbec není uložena.

Více viz kap. 2.2.1.1.

2.1.2.2 Prohledávání adresáře MAD

Jak bylo řečeno, adresář MAD usnadňuje systému orientaci ve struktuře karty při hledání konkrétní aplikace či jen zjišťování, jaké aplikace daná karta nese. V případě verzí MAD1 a MAD2 je postup v podstatě stejný, zatímco MAD3 přichází s jiným, jednodušším přístupem.

Hledání aplikace s MAD1/MAD2

Požadovaná transakce na kartách MIFARE Classic a dalších, podporujících dané verze MAD, začíná prohledáním adresáře aplikací dané karty uloženém v MAD sektoru. Pro přečtení daného adresáře musí žadatel (systém) znát klíč A sektoru 0x00, poté může postupně prohledávat jeho první a druhý blok, kde jsou uloženy AID – hledaná aplikace může využívat i více sektorů, proto je dobré projít celý adresář. Z GPB bytu systém zjistí, zda karta používá MAD verzi jedna nebo dva. V případě používání verze MAD2 systém navíc prohledá sektor 0x10, ke kterému také potřebuje znát odpovídající klíč A. Tímto způsobem zjistí, zda hledaná aplikace je v sektorech 0x01 až 0x0F, případně v 0x11 až 0x27.

Po zjištění adresy sektoru obsahující požadovanou aplikaci a při znalosti přístupového klíče (A nebo B dle požadavku) může systém s ní začít pracovat. Struktura odpovídajícího sektoru (sektoru aplikace) vypadá podle toho, jak ji vytvoří sw aplikace – jak aplikace vyžaduje.

Hledání aplikace s MAD3

Pro získání informace o aplikacích přítomných na kartě MIFARE DESFire stačí vyslat příkaz **GetApplicatiolDs**. Jako odpověď žadatel obdrží automaticky generovaný výpis všech aplikací na kartě. Projitím tohoto seznamu se zjistí, zda se požadovaná aplikace na kartě vyskytuje či ne.

2.1.2.3 Sektory vydavatele a uživatele karty

Ne všechny sektory obsahují aplikace. Kromě sektorů MAD a zatím prázdných nebo rezervovaných sektorů jsou na kartě i takové, které nesou specifické informace o vydavateli karty nebo jeho uživateli.

Sektor vydavatele karty

Sektor vydavatele karty obsahuje informace o subjektu odpovědném za vydání karty a údržbu její i jejího MAD. Jedná se především o identifikační a kontaktní informace. Na polohu tohoto sektoru ukazuje informační byte v MAD sektoru.

Pokud je v paměti karty stále volný alespoň jeden sektor (v případě karet MIFARE DESFire dostačující prostor), je možné se obrátit na vydavatele karty a nechat si tam přidat novou aplikaci – pokud její velikost nepřekročí velikost volné paměti. Žadatel musí vyplnit registrační formulář týkající se jak zadavatele/majitele aplikace (identifikační a kontaktní informace), tak aplikace samotné (popis, požadavky uložení, apod.).

Informace o držiteli karty

Informace o držiteli kary jsou ukládány do sektorů s administrativním kódem 0x00 nebo 0x04 (viz kap. 2.1.2.1 – AID). Na jedné kartě může být více takovýchto sektorů.

Sektor držitele karty v případě MAD1 obsahuje čtyři bloky po 16 bytech, z čehož tři jsou bloky pro zápis dat. Na kartách s verzí MAD2 může blok obsahovat i 16 bloků po 16 bytech (viz kap. 2.2.1.1). Na kartách MIFARE DESFire s verzí MAD3 sektor může být volitelně veliký (viz kap. 2.2.2.1).

Jednotlivé zapsané informace na sebe plynule navazují, zároveň jsou od sebe odděleny kódem 0x00 značícím ukončení předchozí informace. Zápis je prováděn v ASCII formátu a začíná od bytu nula bloku nula, kam se zapisují úvodní parametry první zapisované informace. Nejnižších šest bitů tohoto bytu určuje počet použitých bytů na zápis zahrnující i ukončovací byte 0x00. Horní dva bity specifikují typ zapsané informace:

- 00 – příjmení,
- 01 – křestní jméno,
- 10 – pohlaví: 0x6D je zadáno pro muže, 0x66 pro ženy,
- 11 – další data.

Hned v následujícím bytu za „ukončujícím“ bytem může být uložena další posloupnost začínající bytem s úvodními parametry následující informace.

Kódem 0x00 jsou značeny také prázdné byty. Pokud naopak je v sektoru nedostatek místa, zápis může pokračovat v dalším sektoru označeném administrativním kódem 0x00 nebo 0x04.

Příklad zápisu v tomto sektoru je uveden na Obr. 10. [12]

bit 7 ……. bit 0	bit 7 ……. bit 0	…….	bit 7 ……. bit 0	bit 7 ……. bit 0	
byte n	byte n-1	…….	byte 1	byte 0	
ukončující byte 0x00	poslední znak		znak 1	typ info	délka <n>

Obr. 10: Struktura zápisu dat do sektoru držitele karty [12]

2.1.3 Zabezpečení dat

Zabezpečení uložených dat na kartě a celkové komunikace s terminálem je důležité pro zamezení odposlechnutí soukromých či jinak citlivých dat, jejich úmyslnému i neúmyslnému přepsání či smazání, ale také zabránění vydávání se za někoho jiného nebo získání informací, služeb či benefitů, na které nemá útočník nárok.

Karty MIFARE toto zajišťují použitím šifrovacích algoritmů. První generace využívaly vlastní proudovou šifru CRYPTO1, která ovšem byla prolomena a společnost NXP byla nucena přejít k bezpečnějším algoritmům – DES, 3DES a AES. Šifrovací klíče zde umožňují nastavení různé úrovně bezpečnosti (různá přístupová práva) ke každému sektoru, u karet MIFARE DESFire dokonce ke každému bloku zvlášť.

Navázání kontaktu s kartou, zjištění jejího UID a v případě karet DESFire (používajících MAD3) i získání seznamu aplikací na ní uložených probíhá bez šifrování, tedy v holém textu bez potřeby autentizovat se. U karet podporujících MAD1 nebo MAD2 bývají v MAD sektorech použity šifrovací klíče, ale většinou klíč A je veřejně známý, aby terminál jednoduše přečetl, jaké aplikace karta obsahuje. Avšak ve chvíli, kdy se systém chce dostat k soukromým datům uživatele nebo k aplikacím uloženým v dalších sektorech, je potřeba, aby prokázal, že na to má právo – autentizoval se. [3][9][10]

2.1.3.1 Tříprůchodová autentizace

Systémy MIFARE využívají vzájemnou tříprůchodovou autentizaci – založenou na standardu ISO/IEC DIS 9798-2 –, kdy se ověřuje jak identita a oprávnění systému, tak identita karty. Identita obou stran se ověřuje proto, aby se zajistilo, že spolu skutečně komunikují ti, za koho se strany vydávají.

Tento proces se dá popsat v pěti hlavních bodech:

1. Čtečka určí sektor, ke kterému chce získat přístup, a odpovídající typ přístupového klíče: A nebo B dle typu přístupu, který chce/může získat – z hlediska bezpečnosti podmínky pro různý typ přístupu jsou většinou různé, např. ne každý, kdo smí data číst, je může i upravovat.
2. Karta z odpovídajícího zavaděče sektoru přečte podmínky přístupu a přístupový klíč, případně klíče (podle výzvy čtečky), poté vygeneruje náhodně číslo, které:
   a) nezašifruje (CRYPTO1),
   b) zašifruje (DES, 3DES či AES)
   a odešle jako výzvu pro čtečku – „první průchod“.

3. Po obdržení výzvy čtečka vypočítá odpověď (na přijaté číslo použije požadovaný tajný klíč a další vstup podle odpovídajícího algoritmu), připojí k tomu vlastní náhodně vygenerované číslo/výzvu pro kartu, zašifruje to a odešle zpět – „druhý průchod“.
4. Karta ověří přijatou odpověď porovnáním s vlastním výsledkem výpočtu – také použije tajný klíč na své vygenerované číslo. Poté vypočítá odpověď na výzvu od čtečky a odešle ji – „třetí průchod“.
5. Čtečka porovná obdrženou odpověď karty s vlastním výsledkem použití klíče na totéž vygenerované číslo.

Pokud všechny kontroly odpovědí (na straně karty i čtečky) dopadnou dobře, je ověřeno, že obě strany znají potřebný šifrovací klíč a čtečka s kartou spolu mohou bezpečně komunikovat – dokud zůstanou šifrovací klíče utajeny. Všimněme si rozdílu, že v případě použití algoritmu CRYPTO1 je komunikace šifrovaná až po prvním průchodu, zatímco ostatní algoritmy zasílají šifrovanou již první výzvu, čímž je zvýšena bezpečnost proti odposlechu třetí stranou. [9][10]

2.1.3.2 CRYPTO1

CRYPTO1 je šifrovací algoritmus vytvořený konkrétně pro účely karet MIFARE, který nebyl nikdy oficiálně zveřejněn. Přesto bylo zveřejněno již několik prací, například [13] odhalujících jeho strukturu a především jeho slabá místa umožňující útočníkovi číst data na kartě, klonovat kartu či ji navrátit do minulého stavu.

CRYPTO1 patří mezi proudové šifry [13]. To jsou šifry, které postupně implementují – bit po bitu – na originální holý text pseudonáhodně generovaný proud bitů tvořený na základě šifrovacího klíče a šifrovacího algoritmu. Tento postup by měl zaručit, pokud jsou bity tvořeny dostatečně náhodně, že při opakovaném vstupu stejného datového proudu je vytvořen jiný zašifrovaný výstupní proud. Proudové šifry se navíc řadí mezi šifry symetrické, tudíž pro zašifrování dat i jejich zpětné rozšifrování používají stejný tajný šifrovací klíč, který by pro bezpečnou komunikaci měly znát pouze oprávněné – komunikující – strany. [14]

Základem algoritmu CRYPTO1 je pseudonáhodný generátor proudu bitů realizovaný 48bitovým LFSR – posuvným registrem s lineární zpětnou vazbou používajícím generující polynom *g(x)* a nelineární filtrovací funkci *f*.

$$g(x) = x^{48} + x^{43} + x^{39} + x^{38} + x^{36} + x^{34} + x^{33} + x^{31} + x^{29} + x^{24} + x^{23} + x^{21} + x^{19} + x^{13} + x^{9} + x^{7} + x^{6} + x^{5} + 1 \quad (1)$$

Při každém taktu hodin jsou bity v registru posunuty o jeden bit doleva, přičemž bit nejvíce vlevo je vyřazen a přes polynom *g(x)* je počítán zpětný bit. Na tento bit a na právě příchozí vstupní bit LFSR je vzápětí použita logická funkce XOR a výsledek této operace je zapsán

zprava do LFSR – všechny bity se posunou doleva. Pro účel šifrování jsou určité bity použity jako vstup do funkce f, které bity to jsou, však nebylo zveřejněno.

CRYPTO1 se chtěl prosadit jako rychlý bezpečný krypto-systém. Prvního cíle dosáhl, avšak co se týče druhého, brzy se ukázalo, že mu je na míle vzdálený. Karty pracující s tímto algoritmem jsou sice stále vydávané, pokud tam ale není navíc podpora dalšího – bezpečného – algoritmu, nejsou určeny pro operace s citlivými daty.

Během několika let zkoumání bezpečnostních aspektů CRTYPTO1 bylo zjištěno, že útočník je schopen několika různými způsoby obnovit část nebo celý tajný klíč. Jako příklad zde budou naznačeny tři cesty popsané v práci [13].

První možností je využití znalosti, že do filtrovací funkce f jsou brány pouze liché bity. Díky tomu lze rozdělit 48 bitů LFSR na liché a sudé a pracovat s nimi odděleně a efektivněji, případně je i kombinovat, až je nakonec možné obnovit ty stavy LFSR, které vytváří proud klíče. Tento potup se dá popsat jako invertování filtrovací funkce *f*.

Druhou slabinou těchto karet pracujících s CRYPTO1 – především MIFARE Classic – je, že během procesu autentizace používají pseudonáhodný generátor čísel uložený na kartě, který používá 16bitový LFSR s generujícím polynomem $x^{16} + x^{14} + x^{13} + x^{11} + 1$. Hodnoty jím tvořené, tedy i výzvy vysílané kartou, jsou plně předvídatelné – zcela závisí na době uběhnuté mezi započetím napájení karty (jejího vstupu do pole čtečky) a počátku komunikace. V případě tohoto systému na čtečce jsou hodnoty updatovány při každém vyvolání čtečky. Při znalosti těchto okolností je útočník schopen odhalit šifrovací klíče používané během autentizace. Dosáhne toho například díky další znalosti, a to že první výzvu karta vysílá v holém textu a čtečka na ni odpovídá šifrovanou hodnotou. Při vícenásobném opakování této výzvy je schopen šifru rozluštit.

Do třetice zde můžeme uvést způsob, jakým útočníci mohou zneužít skuliny v procesu autentizace. V případě, že karta neodpoví na výzvu čtečky – viz krok 4. v kap. 2.1.3.1 – čtečka se zachová buď:

a) po vypršení určitého času pošle v zašifrované podobě příkaz HALT nebo
b) pokračuje v komunikaci, jako by k autentizaci došlo, – pošle v zašifrované podobě příkaz READ.

Při znalosti tohoto chování a bytových kódů příkazů HALT a READ útočník může schválně zařídit, aby karta neodpověděla, počká na reakci čtečky a z obdrženého příkazu obnoví šifrovací klíč. [13]

2.1.3.3 DES & 3DES

Algoritmus **DES** se také řadí mezi symetrické šifry, avšak ne proudové nýbrž blokové. Jejich základním principem je šifrování neaplikované na jednotlivé bity, nýbrž vždy na celé jejich bloky určité délky. Na tyto bloky jsou pak aplikována vždy stejná transformační pravidla daná šifrovacím klíčem. To znamená, že jeden a tentýž vstup je zašifrován pokaždé (při opakovaném vstupu) stejně, což není zrovna nejbezpečnější přístup, proto se často navíc používají další algoritmy minimalizující negativní vlivy tohoto postupu. Jedním z nich je například algoritmus řetězení šifrovacích bloků CBC, který funguje na principu ovlivnění nezašifrovaného bloku předešlým již zašifrovaným blokem. Pro první blok je generován inicializační vektor. Na ovlivňující se bloky je použita logická bitová operace XOR. Výsledkem je proud na sobě vzájemně závislých zašifrovaných bloků.

Originální text je tedy na začátku algoritmu DES rozkouskován do bloků o velikosti 64 bitů. Každý blok nejprve projde prvotní přestavbou, kdy je pořadí jeho bitů permutačně zpřeházeno, a poté je rozdělen na dvě 32bitové části L a R. Následně probíhá cyklus 16 nelineárních transformací, kdy je v každém cyklu použit jeden ze šestnácti 48bitových subklíčů vytvořených z hlavního klíče – jedná se o tzv. substitučně permutační Feistelovu síť. Výstupem jsou dva 32bitové bloky, které jsou spojeny zpět do jednoho 64bitového bloku, jehož bity jsou zpětně přestaveny. Proces dešifrování probíhá obdobně, pouze pořadí subklíčů je otočeno.

Jako hlavní klíč je většinou použit 56bitový blok. Někdy je použito 64 bitů, avšak každý osmý bit slouží pro kontrolu parity a tedy opravdu efektivně použitých je opět pouze 56 bitů. Tato délka je považována za hlavní slabinu celého algoritmu, jelikož technický pokrok v rychlosti práce výpočetní techniky umožňuje velmi rychlé prohledání prostoru možných kombinací tohoto klíče – pouze 72 biliard možností – a tedy prolomení algoritmu hrubou silou během relativně krátké doby.

Přestože DES byl prolomen, princip, na kterém pracoval, je stále považován za bezpečný. Roku 1990 byl aplikován tento algoritmus třikrát se třemi různými klíči délky 56 bitů – vznikl algoritmus **3DES** s kombinovaným klíčem celkové délky 168 bitů. Tento systém je sice možné teoreticky prolomit, prakticky je ale zatím bezpečný – v jeho designu nebyly nalezeny žádné závažné díry a prolomení hrubou silou by s dnešní technikou trvalo v řádu desítek let.

Dnes se nejčastěji používají dvě varianty tohoto algoritmu, kdy ta první, odpovídající výše zmíněnému použití tří různých klíčů, se nazývá **3K3DES**. Druhá varianta, o něco méně bezpečná, je značena **2K3DES** a používá pouze dva různé klíče K1 a K2, třetí klíč K3 je shodný s klíčem K1 a tudíž je efektivní velikost kombinovaného klíče pouze 112 bitů, přesto byla dlouhou dobu považována za bezpečnou. [14]

Roku 2011 se objevila zpráva, že zabezpečení karet MIFARE DESFire používajících 2K3DES bylo prolomeno. Vědcům z německé univerzity Bochum se podařilo metodou postranních kanálů úplně obnovit 112bitový master klíč. [15]

2.1.3.4 AES

Roku 2001 vyhlásil americký Národní institut standardů a technologie NITS výherce soutěže na nový kryptografický šifrovací standard AES. Byl jím modifikovaný algoritmus Rijndeal, který převyšoval 3DES softwarově i hardwarově – rychlejší a efektivnější.

AES je bloková šifra používající kombinaci substitučně permutační sítě s operacemi Galiosova pole – podobné RSA modulo aritmetickým operacím. Velikost bloků 128 bitů a šifrovacího klíče 128, 192 nebo 256 bitů zajišťuje vysokou odolnost proti útokům hrubou silou.

Přestože zatím nebyly nalezeny žádné hrubé nedostatky v designu algoritmu, ani AES, stejně jako ostatní algoritmy, není 100% bezpečný. V podstatě je to vždy otázka peněz a času vynaložených útočníkem na zjištění klíče. Můžeme také říci, že úroveň bezpečnosti šifry je dána především délkou použitého klíče, čím je delší, tím je potřeba vynaložit větší úsilí na její prolomení. V případě standardu AES jsou náklady na prolomení poměrně, většinou nadmíru, vysoké. [14]

Autoři článku [14] roku 2010 počítali, že pokud by útočník použil systém, který hledá správnou kombinaci klíče rychlostí jedna miliarda klíčů za sekundu, což je mimochodem 1 000krát rychlejší než nejrychlejší osobní počítač z roku 2004, potřeboval by 10^{22} let na prohledání všech 2^{128} možných kombinací nejmenšího 128bitového klíče AES. Také odhadovali, že americká vojenská zpravodajská služba bude mít technické a ekonomické prostředky na prolomení tohoto algoritmu přibližně za 30–40 let. Avšak s příchodem rychlejších kvantových počítačů o rok později se musí počítat s tím, že tyto odhady budou překonány. Článek [16] o hrozbě kvantových počítačů vůči šifrovacím algoritmům s tajným klíčem uvádí, že pro ně *dosud známé* kvantové útoky, s výjimkou použití hrubého výpočtu všech možných kombinací, nepředstavují hrozbu. Pro případ použití hrubé síly uvádí, že zvětšení šifrovacího klíče je dostatečnou ochranou. Tedy v případě AES je úroveň zabezpečení proti kvantovým počítačům při použití 256bitového klíče v podstatě shodná, jako zabezpečení 128bitového klíče proti osobním počítačům, které je naznačeno výše.

2.1.4 Rychlost zpracování operací

Neustálá snaha o zrychlování a zefektivňování procesů probíhajících ve většině oblastí lidského bytí je typickým znakem moderní doby, i proto je pro systémy pracující s aplikacemi uloženými na kartě jedním z nejdůležitějších faktorů ohodnocení efektivity nasazení konkrétní

aplikace do provozu rychlost zpracování požadovaných operací. Proto tento faktor často patří mezi základní parametry návrhu aplikace, přičemž operacemi, které mají být prováděny, jsou především zápis a čtení dat.

Rychlost zpracování operace v aplikaci v podstatě závisí na způsobu zabezpečení této aplikace. Z pohledu použitého šifrovacího algoritmu je evidentní, že při použití delších či vícero klíčů, bude proces operace probíhat déle, než při použití klíčů kratších. Rychlost zpracování ale neovlivňují pouze šifrovací algoritmy, významný vliv má také podepisování provedených změn v aplikaci. Více o důvodu a principu použití digitálních podpisů bude uvedeno v kapitole 4.1.1.2, prozatím uvedeme jen, že se jedná o způsob zabezpečení integrity datové struktury a používají se tři základní typy podpisů:

- podpis aplikace,
- podpis souboru,
- podpis řádku.

V podstatě základním pravidlem je, že čím podrobnější podpis používáme – čím menší položku podepisujeme – tím delší čas na danou operaci potřebujeme. Je tomu tak proto, že když používáme podpis celé aplikace, kontrolujeme integritu jednoho velkého celku, kdežto v případě podepisování souboru podepisujeme soubor a k tomu zároveň celou aplikaci, do které patří. Stejně tomu je i u podpisu řádku, kdy je nejprve kontrolována integrita daného řádku – konkrétní změny –, následně integrita souboru, v němž je měněná položka uložena, a nakonec i integrita celé aplikace. Hodnoty časů zpracování operací čtení a zápisu dat do a z aplikace v závislosti na použitém typu digitálního podpisu, které byly získány během experimentálního měření Výpočetního Informačního centra VIC ČVUT, jsou uvedeny v Tab. 2. Samozřejmě hodnoty pro čtení dat jsou menší oproti hodnotám pro zápis, jelikož při čtení neprovádíme žádnou změnu dat trvající určitý čas, jak tomu je ve druhém případě.

Tab. 2: Hodnoty rychlostí čtení a zápisu dat z a na kartu MIFARE DESFire při použití různého typu digitálního podpisu získané dle experimentálního měření VIC ČVUT [17]

Typ podpisu	Zpracování [ms]	
	čtení	zápis
podpis aplikace	380	600
podpis souboru	430	720
podpis řádku	580	900

[17]

2.2 MIFARE Classic & MIFARE DESFire

V této kapitole budou porovnány dvě nejrozšířenější řady karet MIFARE, vůbec první řada versus řada o 12 let mladší, podporující již technologii smart karet. Výsledný pohled by měl ukázat, která řada je vhodnější pro použití v oblasti moderních procesů odbavení cestujících.

2.2.1 MIFARE Classic

První řada bezkontaktních čipových karet MIFARE, MIFARE Classic, byla uvedena na trh již roku 1994. Tyto karty se rychle rozšířily po celém světě a přesto, že se objevily bezpečnostní chyby v používaném šifrovacím protokolu, jsou v mnoha odvětvích používány dodnes, i když už jen pro aplikace nevyžadující zabezpečení dat. Oblibu mezi zákazníky si získala také díky příznivému poměru cena/výkon. K dostání jsou verze s 1kB a 4kB datovou pamětí, mírně se lišící v její struktuře (viz níže). Byla vyráběna i tzv. verze ‘‘mini“ s 320B pamětí, která dnes již v nabídce není. [3]

Tyto multiaplikační karty, stejně jako ostatní řady, dodržují standard ISO/IEC 7810 a první tři části ISO/IEC 14443A. Po navázání komunikace se čtečkou pokračují v práci s daty dle vlastního standardu MIFARE. Pro zabezpečení uložených dat a probíhající komunikace byl pro ně vytvořen vlastní šifrovací standard CRYPTO1, u kterého ale bylo po několika letech prokázáno, že zdaleka neposkytuje avizovanou ochranu. Vyhledávání aplikací na kartě je podporováno standardem MAD verze jedna či dvě. Více k těmto standardům je uvedeno v kapitolách 2.1.1, 2.1.2 a 2.1.3.2.

Pro svou identifikaci používaly nejdříve čtyřbytové NUID, ale se vzrůstajícím zájmem zákazníků – tím pádem zvýšenou spotřebou karet – a požadavkem na jedinečnost těchto identifikátorů začaly být vyráběny karty se sedmibytovým UID. Tyto identifikátory jsou používány především při výběru karty a navazování komunikace karta-čtečka (viz kap. 2.1.1.2)

Operace probíhající na kartě zprostředkovává logická jednotka. Bezkontaktní přenos dat je prováděn rychlostí 106 kbit/sek, což zaručuje poměrně rychlé provádění jednotlivých transakcí. Například transakce prodeje jízdenky bývá provedena za méně jak 100 ms. Pro zabezpečení spolehlivého přenosu dat – jejich integrity – je implementováno několik kontrolních mechanismů:

- 16 kontrolních CRC bitů v bloku,
- paritní bity pro každý byte,
- kontrola bitového počítání,
- bitové kódování pro rozlišení „1“, 0“ a „žádná informace“,
- monitorování kanálu – analýza sekvence protokolu a bitového toku. [9]

2.2.1.1 EEPROM MIFARE Classic

Data jsou na kartě ukládána do paměti EEPROM, která může být velikosti 1 kB nebo 4 kB, až na dobu 10 let a mohou být přepisována až 100 000krát. [9]

Rozdíl mezi dvěma vyráběnými typy MIFARE Classic karet není pouze ve velikosti EEPROM, ale také v její struktuře. 1kB (1 024B) paměť je rozdělena do 16 sektorů obsahujících čtyři bloky, zatímco 4kB (4 096B) paměť obsahuje 40 sektorů, z nichž 32 sektorů se skládá ze čtyř bloků a osm sektorů z 16bloků. Všechny bloky mají velikost 16 B. Struktura obou typů pamětí je zobrazena v příloze B.

Díky dané struktuře je velikost i pozice sektorů pevně daná. Uvnitř sektorů jsou ukládány buď specifické informace, nebo jednotlivé aplikace (v jednom sektoru maximálně jedna aplikace, avšak jedna aplikace může zabírat více sektorů). Díky tomu, že pro každý sektor mohou být nastaveny podmínky přístupu zvlášť, i přístup k informacím, popřípadě k aplikacím, v jednotlivých sektorech může být nastaven na požadovanou úroveň nezávisle na ostatních.

Přístupové podmínky sektoru

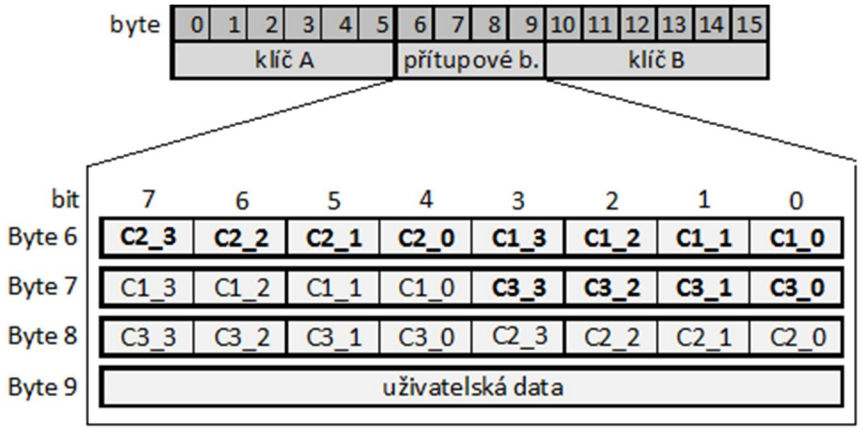
Přístupové podmínky k danému sektoru, ke každému jeho bloku, jsou definovány v jeho posledním bloku, tzv. **zavaděči sektoru**. Zde jsou uloženy dva šestibytové přístupové klíče A a B, tři přístupové byty a jeden byte vyhrazen pro uživatelská data – viz Obr. 11. V případě, že není definován klíč B, jsou byty jemu vyhrazené poskytnuty také pro uživatelská data.

Přístupové byty nesou informaci o tom, jaké operace jsou v jednotlivých blocích sektoru povoleny (read, write, increment, decrement, …) a za jakých podmínek mohou být prováděny, respektive jaké klíče pro to musí systém (žadatel) znát – A, B, A|B. Aby mohl systém skutečně dané příkazy provádět, musí se těmi klíči nejprve řádně autentizovat (viz. 2.1.3.1). V případě, že není uveden žádný klíč, daný příkaz nelze provést nikdy.

Přístupové podmínky bloku jsou zadány třemi bity (C1_X, C2_X a C3_X, kde $0 \leq X \leq 3$), které jsou uloženy jednou v neinvertované podobě a jednou v podobě invertované – zabírají šest bitů z 24. X v označení přístupových bitů specifikuje bloky, pro které dané podmínky platí. Ve čtyřblokových sektorech mohou být definovány pro každý blok různé podmínky, zatímco v sektoru se 16 bloky jsou vytvořeny pětice bloků se stejnými přístupovými bity a zavaděč sektoru, jako jediný blok, je má nastaven samostatně (viz Tab. 3).
Při každém přístupu k paměti vnitřní logika systému ověří formát přístupových podmínek, v případě, že je detekováno formátové ohrožení, celý sektor je nenávratně zablokován.

Přístupové klíče jsou šifrovacími klíči algoritmu CRYPTO1 (viz 2.1.3.2). **Klíč A** musí být tajný a vždy stanoven – je povinný –, zatímco **klíč B** je volitelný, a tedy nemusí být tajný, případně nemusí být použit vůbec. Když je zavaděč sektoru čten, byty tajného klíče vrací logickou

nulu – nemohou být tedy přečteny. Když klíč B není tajný, může být přečten. Aby se zabránilo narušení nových karet, musí být poprvé autentizovány předdefinovaným klíčem A.



Obr. 11: Struktura zavaděče sektoru a přístupových bytů – C1_X, C2_X, C3_X: přístupové bity bloku skupiny X; tučně zobrazené jsou invertované podoby bitů [9]

Tab. 3: Specifikace označení přístupových bitů C1_X, C2_x a C3_X [9]

X	Číslo bloku	
	Sektor se 4 bloky	Sektor s 16 bloky
0	0	0-4
1	1	5-9
2	2	10-14
3	3	15

MAD sektor

Karty MIFARE Classic mohou podporovat adresář aplikací MAD. Pokud tomu tak je, na 1kB kartě je jeden MAD sektor (0x00) a na 4kB kartě jsou tyto sektory dva (0x00 a 0x10). Více viz kap. 2.1.2.

Blok výrobce

Nultý blok nultého sektoru (0x00) se nazývá blok výrobce karty. Jeho první čtyři byty (případně sedm bytů) jsou zaplněny identifikátorem karty NUID (případně UID), ve zbytku bloku jsou uložena data o výrobci. Tento blok je naprogramován během výrobního testu a je chráněn proti přepsání.

Datové bloky

Ostatní bloky na kartě jsou **datové bloky**, které mohou být nastaveny jako **blok pro čtení/zápis** nebo jako **hodnotový blok**. Do prvního typu bloku jsou zapisována data v ASCII formátu a jsou zde podporovány pouze příkazy read a write (viz kap. 2.1.1.3). Hodnotové bloky jsou specializované na efektivní práci s čísly – využitelné například pro funkci elektronické peněženky apod. Mají pevný datový formát podporující detekci a opravu chyb

a zálohovací management. Tento speciální formát zaznamenává hodnotu do prvních čtyř bytů bloku (0–3), v následujících čtyřech bytech (4–7) ukládá její invertovanou podobu a následně ještě jednou neinvertovanou podobu (bity 8–11). V posledních čtyřech bytech je čtyřikrát (dvakrát v neinvertovaném a dvakrát v invertovaném tvaru) uložena adresa bloku obsahujícího zálohu zapsané hodnoty – pokud je použit management zálohování. V hodnotovém formátu jsou využívány příkazy read, write, increment, decrement, restore a transfer (viz kapitola 2.1.1.3) – data uložena v adresových bytech mohou být měněna pouze příkazem write.

byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	hodnota				**hodnota**				hodnota				adr	**adr**	adr	**adr**

Obr. 12: Hodnotový blok – „hodnota“: aktuální uložená hodnota, „adr“: adresa bloku obsahujícího zálohu uložené hodnoty; tučně zobrazené jsou invertované tvary [9]

[9]

2.2.1.2 Zabezpečení MAFARE Classic

Při shrnutí zabezpečovacích mechanismů čipové karty předkládaných výrobcem, získáme následující výčet:

- výrobcem nastavené sedmibytové UID nebo čtyřbytové NUID pro každé zařízení,
- podpora náhodného ID,
- vzájemná tříprůchodová autentizace a
- individuální sada dvou přístupových klíčů pro každý sektor podporující multiaplikační prostředí s klíčovou hierarchií. [9]

Na první pohled tento výčet vypadá dobře, ovšem když si uvědomíme, že většina uvedených bodů z velké části stojí a zároveň i padá na již dávno prolomeném šifrovacím standardu CRYPTO1 a na generátoru – nenáhodně – náhodných čísel (viz kap. 2.1.3.2), je zřejmé, že karty MIFARE Classic nemohou být používány pro účely práce s citlivými daty.

2.2.2 MIFARE DESFire

Roku 2002 byla na trh uvedena nová řada bezpečnostních multiaplikačních smart karet MIFARE DESFire (MF3ICD40), které jako první disponovaly vlastní CPU jednotkou. Silné vlastnosti tohoto produktu výrobce zakotvil už do jeho názvu. „DES“ mělo ukázat na vysokou úroveň zabezpečení zajištěnou šifrovacími algoritmy DES a 3DES, zatímco „Fire“ byl akronymem pro – „Fast Innovative Reliable and Enhanced” – rychlý inovativní spolehlivý vylepšený systém pracující v prostředí bezkontaktních aplikací. [3][15]

V roce 2006 se objevila řada MIFARE DESFire EV1, která jako první pro své zabezpečení navíc použila 128bitový AES. Tato řada po pěti letech plně nahradila svého předchůdce (MF3ICD40), který byl po prolomení jeho 2K3DES protokolu označen za nespolehlivý – viz kap. 2.1.3.3. Brzy si vydobyla jednu z předních pozic na trhu bezkontaktních čipových karet, kterou si drží dodnes. [3][15]

V roce 2013 byla vydána prozatím nejnovější verze MIFARE DESFire EV2 přinášející několik funkčních inovací, jako například pozměněnou strukturu EEPROM, možnost vzájemné kooperace nahraných aplikací na jedné kartě – sdílení souborů mezi aplikacemi – nebo vytvoření úplně nových multiaplikačních funkcí MISmartApp podporujících prostředí více provozovatelů a více služeb na kartě pro účely nových chytrých obchodních modelů. Tato verze zatím ve světě není tak hojně rozšířena jako její předchůdce. [3]

MIFARE DESFire EV1 je dle standardu Common Criteria certifikovaným produktem AEL4+, její nástupce EV2 dokonce AEL5+ a v plánu je dosažení úrovní vyšších. Dále jsou obě verze kompatibilní s mezinárodními standardy ISO/IEC 7810 a ISO/IEC 14443A a používají některé příkazy standardu ISO/IEC 7816-4 a kryptografické standardy DES, 3DES a AES. Přístup k aplikacím uloženým na kartě je podporován vlastním standardem MAD3. Více viz kapitoly 2.1.1,2.1.2, 2.1.3.3 a 2.1.3.4. [10] [18]

Karty MIFARE DESFire jsou jednoznačně identifikovatelné podle jejich sedmibytového UID, navíc umožňují použití náhodně generovaného tříbytového ID. Poskytují vysokorychlostní přenos dat rychlostmi 106 kbit/s, 212 kbit/s, 424 kbit/s a 848 kbit/s a integritu přenášených dat zajišťují implementací:

- CRC bitů,
- paritních bitů,
- bitového kódování a
- bitového součtu. [10]

2.2.2.1 EEPROM MIFARE DESFire

Karty MIFARE DESFire používají pro uložení dat, stejně jako ostatní produkty MIFARE, energeticky nezávislou paměť EEPROM. Ta je pro tyto karty dostupná ve velikostech 2 kB (2048 B), 4 kB (4096 B) a 8 kB (8192 B) a uchovává údaje na dobu až 10 let. Oproti MIFARE Classic umožňuje přepisování dat až 500 000krát, tedy pětkrát častěji.

Struktura paměti je pro všechny její velikosti stejná a v podstatě pro každou kartu jiná. Určitá část paměti je vždy vyhrazena pro výrobce karty. Tato část je naprogramovaná během procesu výroby a je chráněna proti přepsání. Jsou v ní uložena důležitá data, jako například UID karty. Zbylá část paměti je vytvořena jako flexibilní souborový systém, což znamená, že zde jsou

zapisovány jednotlivé aplikace, které jsou tvořeny soubory nastavitelné velikosti dle jejich potřeby – velikost je omezena pouze volným místem v paměti (celková kapacita-využitá kapacita).

V případě řady EV1 byly na souborový systém aplikací kladeny dvě omezující podmínky. Může být vytvořeno maximálně 28 různých aplikací a každá tato aplikace smí obsahovat nanejvýš 32 souborů. S příchodem MIFARE DESFire EV2 hranice počtu aplikací padla. Nyní může být vytvořen jakýkoli počet aplikací obsahujících však stále maximálně 32 souborů. Jediným omezením počtu aplikací na kartě je velikost samotné paměti.

Každá aplikace je v rámci karty jednoznačně identifikovatelná svým jedinečným tříbytovým AID – více viz kap. 2.1.2.1. Soubory aplikace mohou být tvořeny v průběhu

a) inicializace (výroby) karty,
b) personalizace karty
   – procesu, kdy se z ní stává konkrétní karta pro konkrétní účel, aplikaci, případně pro konkrétního držitele,
c) provozu karty (během jejího využívání držitelem)
   – když se zjistí potřeba nebo možnost přidat další soubor, novou funkci aplikace apod.

Během vytváření nového souboru je pevně stanovena jeho velikost. Pokud se časem ukáže jako nedostatečná, musí být vytvořen nový, navazující, soubor. Zároveň je určen i jeho typ. Je stanoveno pět typů souborů:

- standardní datový,
- zálohovací,
- hodnotový se zálohováním,
- lineárně zaznamenávající se zálohováním a
- cyklicky zaznamenávající se zálohováním.

V jedné aplikaci je možné kombinovat soubory různých typů nezávisle na tom, zda podporují zálohování či ne. Zálohování dat zaručuje integritu dat na aplikační úrovni. Když soubor, případně celá aplikace, přestane být používán nebo se stane zastaralým, může být natrvalo zrušen.

Každá aplikace EV1 může používat až 14 subklíčů odvozených od hlavního master klíče karty. Každý její soubor tak může použít jedinečnou kombinaci dostupných subklíčů a díky tomu získat úroveň zabezpečení odpovídající citlivosti uložených dat. Délka klíče určuje šifrovací algoritmus, který má být použit (56 bitů: DES, 112 bitů: 2K3DES, 168 bitů: 3K3DES, 128 bitů: AES). V případě EV2 je možné pro jednu aplikaci použít až 16 sad klíčů vytvořených technikou několikanásobného rolování, navíc pro zabezpečení přístupu k souboru může být

vyžadována znalost kombinace až osmi klíčů. V případě zjištění ohrožení klíčů zasahuje samoopravný mechanismus.

Příkazy, které mají vliv na strukturu paměti, například příkazy pro vytvoření aplikací nebo změnu klíčů, automaticky aktivují mechanismus vrácení změn, který chrání strukturu před poškozením. Pro práci s daty v souborech používá karta jak vlastní MIFARE DESFire EV1 (EV2) příkazy – Read Data, Write Data, Get Value a další – i některé příkazy standardu ISO/IEC 7816-4 (viz kap. 2.1.1.3).

[3][10] [18]

2.2.2.2 Zabezpečení MIFARE DESFire

Bezpečné uložení a přenos dat zaručuje MIFARE DESFire EV1 a EV2 použitím následujících mechanismů:

- jedinečné sedmibitové UID pro každé zařízení
  - zaručuje jednoznačnou identifikaci zařízení,
  - pevně přiřazeno k zařízení již při výrobě a chráněno proti přepsání – ochrana proti vytvoření falzifikátu,
  - možnost použití pro odvození subklíčů pro každý vytvořený tiket, čímž je tvořen efektivní nástroj proti jejich klonování a je zajištěna vyšší bezpečnost originálního klíče,
  - použití metody rolování klíčů (EV2),
  - samoopravný mechanismus klíčů (EV2);
- možnost generování náhodného tříbitového NUID;
- vzájemná tříprůchodová autentizace dle ISO/IEC DIS 9798-2 a ISO/IEC 7816-4
  - ověření identity karty i čtečky před započetím práce s daty;
- hardwarové šifrování DES, 3DES a AES;
- jeden master klíč pro kartu a až 14 klíčů / 16 sad klíčů pro každou aplikaci
  - délka klíčů: 56 b pro DES, 112 b pro 2K3DES, 128 b pro AES a 168 b pro 3K3DES;
- možnost nastavení různé úrovně zabezpečení komunikace pro každý soubor zvlášť;
- přenos holých dat
  - možné pouze v režimu zpětné kompatibility s modelem MF3ICD40;
- přenos holých dat s (šifrovaným) kontrolním součtem (MAC)
  - většinou využíván osmibitový CMAC,
  - v režimu zpětné kompatibility s modelem MF3ICD40 je používán MAC;
- přenos šifrovaných dat;

  - nejprve je přes celý proud vypočítán kontrolní součet CRC, který je k datům přidán, společně s nimi zašifrován a následně je výsledek přenesen,
  - většinou počítán 32bitový CRC kód,
  - v režimu zpětné kompatibility s modelem MF3ICD40 je použit 16bitový CRC kód;
- šifrování dat v RF kanálu;
- zálohovací management;
- kontrola integrity dat;
- hw senzory výjimečných stavů.

[10] [18]

2.2.3 Classic vs. DESFire

Z informací uvedených v předchozích kapitolách se dá vyčíst, že karty MIFARE DESFire oproti svému předchůdci poskytují větší a především flexibilní prostor pro uložení dat a aplikací, který tak může být efektivněji využit. V případě pevně daných velikostí sektorů se aplikace snažila těmto sektorům co nejvíce přizpůsobit – objemově se napasovat do daného prostoru tak, aby nezůstalo příliš mnoho nevyužitých bytů – zatímco s flexibilním systémem se sektory přizpůsobují aplikaci a paměť může být využita do posledního bitu. Navíc uložená data mohou být vícekrát přepsána, což podporuje aplikace, u kterých dochází k časté změně položek – např. elektronická peněženka kde dochází k frekventovanému pohybu peněz. Vyšší přenosové rychlosti umožňují rychlejší zpracování transakcí a propracovanější zabezpečovací mechanismy zaručují bezpečné uložení i přenos dat přesně dle jejich požadavků. Na druhou stranu MIFARE Classic 4K je schopná celkově poskytnout více různých aplikací – při využití alespoň 29 ze 40 sektorů – než řada MIFARE DESFire EV1, která jich poskytne maximálně 28. MIFARE DESFire EV2 ovšem poskytuje již neomezený počet aplikací. Navíc pro aplikace nevyžadující zabezpečení dat může být vhodnou volbou nejenom díky dobrému poměru cena/výkon. Přehledné porovnání vybraných parametrů je uvedeno v Tab. 4.

Při výběru konkrétního typu karty je nejdůležitější si nejprve uvědomit, jaké jsou na ni kladeny nároky – jaké aplikace a jaká data bude nosit a spravovat – a to především z hlediska citlivosti údajů a potřeby je nějak chránit. V případě využití v prostředí odbavovacích systémů je potřeba pracovat s poměrně citlivými daty, která potřebují být zabezpečena, aby se zabránilo neoprávněným transakcím, jako například padělání jízdních dokladů, upravování kritických dat – navyšování zůstatku v elektronické peněžence – či užívání benefitů, na které osoba nemá nárok – nahrání průkazu na slevu.

Tab. 4: Porovnání vybraných parametrů bezkontaktních čipových karet MIFARE Classic a MIFARE DESFire [9][10][18]

Vlastnosti produktu	MIFARE Classic™		MIFARE DESFire™		
	1 kB	4 kB	2 kB	4 kB	8K
	MF1 IC S50	MF1 IC S70	MF3 IC D21	MF3 IC D41	MF3 IC D81
Paměť – EEPROM					
Velikost [byte]	1024	4096	2048	4096	8192
Počet možných přepsání [cykly]	100 000		500 000		
Uchování dat [roky]	10				
Struktura	16 sektorů po 64 bytech	32 sektorů po 64 bytech + 8 sektorů po 256 bytech	flexibilní systém souborů (EV1 až 28 aplikací, EV2 neomezeně aplikací – obě max 32s.)		
Radiofrekvenční prostředí					
Frekvence [MHz]	13,56				
Pracovní vzdálenost [mm]	až 100				
Soulad s ISO/IEC 14443A	ano - až do 3. části		ano - až do 4. části		
Přenosová rychlost [kbit/s]	106		106, …, 848		
Antikolizní proces	bitový				
Zabezpečení					
UID [byte]	4 NUID, 7 UID		7 UID		
Generátor náhodných čísel	ano - plně předvídatelný		ano		
Proces autentizace	trojitá vzájemná autentizace		trojitá vzájemná autentizace, autentizace dle ISO/IEC 7816-4		
Přístupové klíče	2 klíče pro sektor		pro aplikaci: EV1: 14 klíčů, EV2 16 sad klíčů, + master klíč ap.		
Podmínky přístupu pro	sektor		soubor		
Zabezpečení komunikace	šifrovaná data		holá data, data + CMAC, šifrovaná data		
Kryptografický standard	CRYPTO1 (48bitový klíč)		DES, 2K3DES, 3K3DES, AES (délka klíče: 56, 112, 168, 128 bitů)		
Podpora ISO 7816-4 APDU	ne		ano		
Common Criteria certifikace	ne		EV1: EAL4+, EV2: EAL5+, příprava na vyšší úrovně		
Systém proti roztržení dat	pro hodnotové bloky		ano		

II. PRAKTICKÁ ČÁST

Struktura a zabezpečení aplikací na MIFARE Classic a DESFire

V praktické části se budeme zabývat návrhem struktury a zabezpečením několika aplikací podporujících procesy odbavení cestujících, případně určité přidané služby, za účelem jejich optimalizace. Během návrhu struktur jednotlivých aplikací je nutné dbát na několik důležitých faktorů zároveň, samozřejmě hlavním kritériem je to, aby daná aplikace skutečně dělala to, co má.

Aby byla zabezpečena správná funkce aplikace, prvním krokem jejího návrhu by měla být úvaha nad tím, co se od ní požaduje, jaké vstupy – vstupní informace – aplikace vyžaduje a jaké výstupy chceme získat. Důležité je pracovat pouze s daty, které jsou nezbytně nutné. Nadbytečná data zabírají užitečný prostor karty a mohou být zdrojem chyb.

Když jsou specifikovány jednotlivé položky aplikace (vstupy a výstupy), specifikujeme formát, ve kterém mají být zapsány – jeho specifikací zabráníme zapsání chybných dat a tvorbě formátových chyb během zpracování. Aplikace poté bude schopna data adekvátně zpracovat, případně je uložit na jim určené místo, kde je později může přečíst a zkontrolovat. Když se stanou data neaktuální, může je přepsat či smazat a jejich místo využít pro data nová.

Dalším důležitým aspektem, který by se neměl opomenout, je velikost jednotlivých položek. Ty by z důvodu efektivního využití paměti měly zabírat jen nezbytně nutný prostor, avšak zároveň nesmí dojít k tomu, že by některá informace, z důvodu nedostatečného místa, nebyla zapsána. Navíc se jedná o důležitý parametr pro výběr typu karty – respektive velikosti EEPROM. Když budou známy velikosti jednotlivých aplikací, které chceme na kartě mít uloženy, budeme vědět, jak veliká paměť karty je zapotřebí.

Nemělo by se opomenout určení výchozích hodnot (některých) zvolených položek, které specifikují základní nastavení aplikace.

Jak již bylo několikrát zmíněno v teoretické části, je velice důležité zajistit bezpečnost uložených dat, a to jak proti jejich přečtení neoprávněným objektem, tak především proti jejich neoprávněnému zapsání či změně. Na druhou stranu se musí myslet na to, že čím více bezpečnostních opatření je použito, tím prostorově nákladnější, výpočetně složitější a tedy i časově náročnější dané operace jsou. Proto se musíme pečlivě zamyslet nad rozdělením přístupových práv jednotlivých položek v aplikaci, to jak editačních (zápisových), tak práv pro čtení. Dalším bezpečnostním opatřením je zajištění a kontrola integrity dat, které je zajišťováno CRC kontrolními součty, paritními bity a digitálními podpisy hash funkcemi.

3 Role subjektů vůči kartě

Ještě před úvahou nad samotnými aplikacemi je důležité stanovit role jednotlivých subjektů pracujících s kartou během jejího životního cyklu. Tyto role mají dané určité funkce, které jsou oprávněny a schopny vykonávat, zároveň disponují znalostí prokazující jejich oprávnění – v případě aplikací se jedná mimo jiné o znalost přístupových klíčů. Poté již při konkrétní implementace stačí přiřadit konkrétním subjektům (jednotlivým dopravcům, firmám, uživatelům atp.) konkrétní role dle míry jejich zapojení, čímž se specifikují jejich práva a možnosti práce s kartou. Jednu roli může reprezentovat více subjektů a zároveň jeden subjekt může nabývat více rolí.

Výrobce karty

Výrobce vyrábí surovou kartu – polotovar – ve které je již implementován konkrétní čip určující typ karty (Classic, Ultralight, DESFire, …) – je stanovena velikost a struktura EEPROM, typ šifrování apod. Na kartu nenahrává žádná data ani aplikace.

Vydavatel karty

Vydavatel kupuje od výrobce prázdné karty konkrétního typu a stává se v podstatě „vlastníkem prostoru“ EEPROM – je majitelem master klíče karty. S tímto prostorem má několik možností:

- celý prostor využije pro vlastní účely – nahraje na kartu pouze své aplikace, jejichž je správcem (viz níže),
- nahraje na kartu vlastní aplikace a zbylý prostor poskytne zájemcům[1],
- poskytne celý prostor karty jednomu zájemci, který, jako jediný, bude mít na dané kartě aplikace – vydavatel karty v podstatě pouze přeprodá – nebo
- poskytne celý prostor karty několika různým zájemcům.

Nahrání cizí (zájemcovy) aplikace na kartu vydavatele může být provedeno jedním z následujících postupů:

- Vydavatel vyhradí zájemci potřebný prostor pro aplikaci, kterou mu tam sám nahraje, a
  - dále již nemá k aplikaci přístup – nezná přístupové klíče
    -> správcem této aplikace je pouze zájemce,
  - nadále má k aplikaci přístup a může měnit její data
    -> správcem aplikace je pouze vydavatel karty, zájemce je její majitel, ale nespravuje ji,
    -> správcem aplikace je jak vydavatel karty, tak zájemce.

[1] Zájemce – subjekt mající zájem o umístění vlastní aplikace na kartu vydavatele karty (vydavatel aplikace – viz níže)

- Vydavatel vyhradí zájemci požadovanou část paměti, počet sektorů, jejich AID, přenechá mu šifrovací klíče a nastavení přístupových práv – respektive nechá zájemce, aby si je sám vytvořil. Poté si zájemce (případně jím zvolený správce aplikace) sám nahraje aplikaci na kartu a spravuje ji, vydavatel karty k ní nemá přístup.
  -> Správcem aplikace je zájemce či jim zvolený subjekt.

Výrobce aplikace

Výrobce aplikace vymyslí a vytvoří aplikaci – její strukturu, logiku, funkce atd.

Vydavatel aplikace

Vydavatel provozuje aplikaci sám, nebo ji přeprodává subjektům, kteří ji provozují – vydavatel vlastní nebo nevlastní od výrobce autorská práva k aplikaci (v druhém případě ji dává k dispozici třetí straně – nevlastní ji).

Správce aplikace

Správce aplikace se stará o celou aplikaci, což zahrnuje několik funkcí:

- nahrání aplikace na kartu, aktualizace verzí,
- nahrání *důležitých dat aplikace*,
- zápis, změna a odstranění *dat, se kterými aplikace pracuje*,
- blokace a odblokování souborů aplikace, případně celé aplikace,
- správa klíčů, určení přístupových práv k aplikaci a
- řešení problémů s aplikací.

Organizace využívající aplikaci

a) aktivně

Organizace využívající aplikaci aktivně je subjekt, který pracuje s aplikací, využívá ji pro svou interní nebo externí činnost, služby zákazníkům, apod. Její pravomoci vůči aplikaci jsou:

- čtení zapsaných dat v aplikaci,
- zápis, změna a odstranění *dat, se kterými aplikace pracuje,*
- podpis zapsaných dat, provedených činností, v aplikaci.

b) pasivně

Organizace využívající aplikaci pasivně poskytne potřebné údaje správci aplikace, který je přidá do systému a zapíše na kartu. Tato organizace poté data z aplikace pouze čte, ale sama do ní nic nezapisuje ani v ní nic nemění. Jedná se například o aplikaci zaměstnaneckých slev, kdy na kartu musí být správcem zapsáno, že držitel karty je zaměstnancem dané organizace, ta poté během provádění služby z karty pouze přečte, že tomu tak je – nic tam nemění – a slevu poskytne.

Pravomoci této organizace vůči aplikaci jsou:

- čtení zapsaných dat v aplikaci.

Organizace akceptující aplikaci

Organizace akceptující aplikaci ji využívá pro své účely, ale žádná data jí neposkytuje, nevkládá, ani je jinak nemění. Má právo pouze přečíst stanovená data uložená v aplikaci. Jako příklad si můžeme uvést aplikaci slev na základě věku uživatele – v aplikaci je uloženo jeho datum narození – případně jeho nárok na určitou slevu –, organizace ho přečte a odvodí, zda má uživatel nárok na určitou slevu (dětskou, ZTP apod.).

Pravomoci této organizace vůči aplikaci jsou:

- čtení zapsaných dat v aplikaci.

Organizace poskytující data pro aplikaci

Tato organizace nevyužívá aplikaci pro sebe či své účely, ale pouze poskytuje důležitá data, se kterými pracuje. Tato data může organizace poskytovat

- **aktivně**, kdy má právo je do aplikace nahrát sama (např. škola má oprávnění nahrát na kartu nárok na studentskou slevu),
  - její pravomoci vůči aplikaci jsou:
    - čtení zapsaných dat v aplikaci – data důležitá pro ověření totožnosti uživatele karty,
    - zápis, změna a odstranění *dat, se kterými aplikace pracuje* a
    - podpis zapsaných dat, provedených činností, v aplikaci;
- **pasivně**, kdy je do aplikace nahrává správce (např. student přinese papírové potvrzení o studiu a správce ho nahraje)
  - organizace nemá žádné pravomoci vůči aplikaci.

Uživatel karty

Uživatel zakoupí kartu od vydavatele karty nebo organizace patřící do stejné transportní sítě a poté smí využívat aktivované aplikace, které jsou na ní nahrané. Pokud některá aplikace nahraná na kartě není aktivována, musí o to uživatel požádat správce aplikace – většinou je zapotřebí nahrát určitá data pro aktivaci aplikace.

Pravomoci uživatele karty vůči aplikaci mohou být:

- čtení zapsaných dat v aplikaci a
- zápis, změna a odstranění *dat, se kterými aplikace pracuje*.

Jako shrnutí této kapitoly a pro účely použití v dalších kapitolách, zde uvedeme, které role mohou být oprávněny údaje na kartě/v aplikaci číst a které by je mohly editovat – v případě znalosti klíčů –:

- **právo číst**
    - správce,
    - organizace využívající aplikaci aktivně,
    - organizace využívající aplikaci pasivně,
    - organizace akceptující aplikaci,
    - organizace poskytující data pro aplikaci aktivně a
    - uživatel;
- **právo editovat**
    - správce,
    - organizace využívající aplikaci aktivně,
    - organizace poskytující data pro aplikaci aktivně a
    - uživatel.

4 Aplikace na kartě

Služby, které by měla multiaplikační karta v prostředí odbavovacích systémů poskytovat, se především týkají prodeje **jízdního dokladu**, a to jak jednotlivých jízdenek, tak dlouhodobých kuponů. K zaplacení dokladu můžeme kromě hotovosti použít platbu **elektronickou peněženkou** nahranou na kartě. Dále by neměly chybět aplikace nesoucí informace o nároku držitele na různorodé **slevy** nebo jeho zařazení do **věrnostního programu** zapojených organizací. Moderní karta by neměla opomenout aplikaci podporující stále více ve městech se rozmáhající **bike sharing**. Samozřejmě zde také nesmí chybět aplikace **personalizace karty**, kde jsou uloženy informace o kartě samotné i o jejím držiteli.

Jelikož chceme, aby využití karty bylo co nejširší, subjekty akceptující její aplikace nemusí být pouze dopravní společnosti. Přijmout platbu z elektronické peněženky, ověřit nárok na slevu z integrované slevové aplikace či vést držitele jako věrnostního zákazníka může kterýkoli subjekt zařazený do systému.

4.1 Procesní model využití multiaplikační karty

Ve chvíli, kdy jsou stanoveny aplikace, které má karta nést, je výhodné vytvořit procesní model životního cyklu karty. Procesní model graficky zobrazuje průběh požadovaného procesu, kdy jsou na určité pozorovací úrovni názorně specifikovány veškeré jeho vstupy a výstupy, probíhající subprocesy a subjekty vykonávající ve vzájemné interakci jednotlivé činnosti. Správně vytvořený procesní model ulehčí přípravnou fázi tvorby aplikací, kdy je zapotřebí přesně a celistvě specifikovat požadavky na jednotlivé aplikace. Základní otázkou je, co má daná aplikace provádět a jaké jsou její hlavní výstupy, kvůli kterým je vytvořena. Které vstupy jsou zapotřebí, aby byla schopna v požadované kvalitě výstupy poskytnout. Jaké subjekty mohou vstupovat a ovlivňovat aplikaci – poskytovat vstupy, provádět jednotlivé činnosti, získat výstupy – a jaká oprávnění by tedy měla získat. Z dobrého modelu jsou tyto jednotlivé složky dobře viditelné.

V příloze D je zobrazen procesní model využití multiaplikační karty obsahující aplikace, které budou popsány v následujících kapitolách. Z důvodu přehlednosti a prostornosti je model vytvořen na vyšší pozorovací úrovni, než by byla potřeba pro specifikaci jednotlivých aplikací na úrovni uložených dat.

4.2 Struktura a zabezpečení vytvářených aplikací

Základní struktura aplikace, kterou zde budeme vytvářet, bude řešena pro MIFARE DESFire, což nám umožňuje rozdělit aplikaci do jednotlivých souborů velikostí vyhovujících požadovaným datům – položkám – a specifikovaných přístupových práv a nemusíme

být omezeni (řízeni) velikostí jednotlivých bloků, jak je tomu u MIFARE Classic. Po vytvoření této struktury pro určitou aplikaci, stanovíme změny a nutná opatření, které by musely být provedeny v případě použití karty typu MIFARE Classic.

4.2.1 MIFARE DESFire soubor

Struktura MIFARE Classic aplikací zapisovaných do sektorů, jak bylo popsáno v kapitole 2.2.1.1, je v podstatě pevně daná (pevně dané počty a velikosti bloků). Oproti tomu o struktuře aplikací v MIFARE DESFire bylo v kapitole 2.2.2.1 uvedeno, že je prakticky pro každou aplikaci různá. Na kartě MIFARE DESFire může být uloženo až 28 aplikací, v případě EV2 jich může být neomezeně. Každá aplikace obsahuje maximálně 32 souborů volitelné velikosti. Žádné jiné parametry pro jejich strukturu nejsou pevně specifikovány. Pro vytvoření určitého pořádku v našich aplikacích si vytvoříme základní strukturu souborů, která se v konkrétních případech může trochu lišit. Dále zde uvedeme, jaká konkrétní bezpečnostní opatření budou aplikována na soubory a aplikace.

4.2.1.1 Základní struktura

Základní struktura souboru je tvořena dvěma částmi, částí nešifrovanou a částí šifrovanou. **Nešifrovaná data** jsou veřejně přístupná pro čtení. Jedná se o data definující parametry, která jsou důležitá pro práci s daty uloženými v šifrované části stejného souboru. Pro účel našich aplikací jsou v této části uloženy položky:

- verze souboru
  – počítáme maximálně s 32 verzemi, proto vyhrazujeme pro tuto položku 5 b;
- status souboru
  – používáme čtyři základní statusy souboru, proto vyhrazujeme pro tuto položku 2 b:
    - 0_{10} **v pořádku** – použitelný pro verzi použitou ve čtečce,
    - 1_{10} **zrušen** – nadále nefunkční,
    - 2_{10} **předem přidělený** – pro použití konkrétními uživateli,
    - 3_{10} **nedostupný** – nepoužitelný pro verzi používanou čtečkou;
- typ šifrování souboru – šifrované části
  – počítáme maximálně s 16 způsoby šifrování, proto vyhrazujeme pro tuto položku 4 b:
    - 0_{10} nešifrováno,
    - 1_{10} soukromý algoritmus vydavatele aplikace,
    - 2_{10} symetrický algoritmus DES-CBC,
    - 3_{10} symetrický algoritmus 2k3DES-CBC,
    - 4_{10} symetrický algoritmus 2k3DES-CBC,
    - 5_{10} symetrický algoritmus AES128,

- 6_{10}-9_{10} specifický algoritmus pro danou síť,
- 10_{10}-15_{10} RFU;

- typ podpisu souboru
  – pokud soubor nepodepisujeme, nemusí být uvedeno,
  – počítáme maximálně se 16 způsoby šifrování, proto vyhrazujeme pro tuto položku 4 b:
    - 0_{10} nepodepsáno,
    - 1_{10} soukromý algoritmus vydavatele aplikace,
    - 2_{10} bloková šifra DES-CBC-MAC8,
    - 3_{10} bloková šifra 2K3DES-CBC-MAC8,
    - 4_{10} bloková šifra 3K3DES-CBC-MAC8,
    - 5_{10} hash funkce MD5,
    - 6_{10} hash funkce SHA-1,
    - 7_{10} hash funkce SHA-2,
    - 8_{10} hash funkce HMAC,
    - 9_{10}-12_{10} specifický algoritmus pro danou síť,
    - 13_{10}-15_{10} RFU;
- RFU prostor
  – velikost tohoto prostoru je dána dle realistického posouzení budoucí potřeby během návrhu aplikace;
- mohou zde být uložena i jiná data, jimž je při návrhu aplikace vyhrazena potřebná část paměti.

Jelikož jsou tato data důležitá pro ochranu šifrované části souboru, mohou být editována pouze správcem aplikace. Například kdyby typ šifrování neoprávněná osoba změnila z určitého způsobu šifrování na „nešifrováno“, získala by snadnější přístup k datům, ke kterým se ve skutečnosti nemá dostat.

V **šifrované části** souborů jsou uváděny již konkrétní datové položky důležité pro správnou funkci aplikace. Jejich formát je většinou Binary, jehož velikost je vždy odvozena od potřeby konkrétní položky. Datumové či časové položky jsou uloženy jako DateStamp nebo TimeReal s danou velikostí 14 b nebo 32 b. Ve vhodných situacích je použit formát Boolean. Práva upravovat jednotlivé položky mají subjekty dle citlivosti a typu jednotlivých dat (specifikováno v jednotlivých souborech níže). Většinu automaticky generovaných dat, jako je aktuální čas, různé identifikátory, zápis do hodnotového souboru, digitální podpis a další, zapisuje SAM modul čtecího zařízení – terminálu.

Ne všechny soubory využívají obě zmíněné části. Hodnotové soubory a soubory Sign nesoucí kryptografické klíče a přístupová práva (viz níže) jsou zašifrované celé. Toto opatření je zavedeno kvůli zvýšení jejich bezpečnosti.

4.2.1.2 Zabezpečení souborů

Optimální **zabezpečení jednotlivých aplikací** na MIFARE DESFire je pro každou z nich zajištěno možností výběru nejvhodnějšího kryptografického algoritmu – DES, 2K3DES, 3K3DES, AES128 nebo specifický pro danou transportní síť – a kombinace přístupových klíčů pro jednotlivé její soubory. V případě karet EV1 může být pro aplikaci použito až 14 subklíčů vytvořených z master klíče karty a jeden master klíč aplikace, EV2 je schopná používat až 16 sad klíčů, při jejichž tvorbě a obnově využívá podporu infrastruktury terminálů.

V našich aplikacích jsou použity algoritmy 2k3DES, 3K3DES a AES128 a pro každý soubor v základu dva klíče s tím, že jeden – klíč A – je prvotně určený pro zápis a druhý – klíč B – pro čtení souboru. Při skutečné realizaci ale vždy záleží na konkrétních souborech a jejich specifických požadavcích. V některých případech je použit jen jeden klíč, jindy naopak více klíčů. Musíme také počítat s omezeným počtem klíčů na aplikaci. Finální rozdělení klíčů je dáno přístupovými právy uloženými v položce Access. Kromě těchto klíčů k souborům používáme ještě pro celou aplikaci jeden RFU klíč, který musí subjekt znát, když chce využít RFU prostor – toto smí správně provádět pouze správce aplikace. Současně s RFU klíčem je nutné znát i klíč pro zápis konkrétního souboru, ve kterém chceme RFU prostor využít – musí použít jejich kombinaci. V aplikaci mohou být pro různé účely použity i jiné specifické klíče.

Tyto klíče společně s master klíčem aplikace a přístupovými právy jednotlivých souborů jsou uloženy vždy v posledním souboru aplikace označeným *Sign*. Soubor Sign je celý zašifrovaný a právo měnit v něm jakékoli údaje má pouze správce aplikace. Na kartě MIFARE Classic jsou klíče a přístupové byty uloženy v posledním bloku – zavaděči sektoru.

Dalším způsobem jak zabezpečit aplikace na kartě je **zajištění a kontrola integrity uložených dat**. Integrity dat je docíleno ve chvíli, když data jsou přesná, jejich obsah je zaručený a zabezpečen proti neautorizovaným změnám. K jejímu zabezpečení přispívají mimo jiné paritní bity, CRC kontrolní součty, digitální podpisy a různé hash funkce.

V principu jde o to, že na daná data použijeme například určitou hash funkci, ta vytvoří svou jedinečnou reprezentaci, jakoby otisk, který je odeslán společně s daty. Na přijatá data druhá strana použije stejnou hash funkci jako odesílatel, svou získanou hodnotu porovná s hodnotou otisku přijatého se zprávou. Pokud během přenosu nedošlo ke změně dat, otisky se shodují. Jestliže data byla změněna, vznikne odlišný otisk od přijatého.

Společně s hash funkcí lze použít i metodu digitálního podpisu subjektu měnícího data. V tomto případě je každá provedená změna podepsána, čímž je jednoznačně určen subjekt odpovědný za změnu. Tento podpis je tvořen tak, že je nezaměnitelně spojen se subjektem

a zároveň se vždy dopočítává tak, aby nebyla rozbita integrita dat. Na takto podepsaný datový soubor je použita hash funkce.

Podepisovány mohou být různě velké celky, díky tomu si můžeme zvolit, zda podepíšeme:

- celou aplikaci,
- jednotlivé soubory nebo
- každý řádek zvlášť.

Každá z těchto možností má své výhody i nevýhody a výrobce aplikace se musí rozhodnout, kterou variantu pro jaká data a aplikace použije. Podepsání větších celků neumožňuje například zpětné dohledání jednotlivých změn (podpis je na konci celku), které můžeme provést při podepisování jednotlivých řádků. Na druhou stranu podepisování celých aplikací nebo souborů zabere daleko méně času než při použití u řádků, přičemž rychlost transakce je jedním z jejich základních parametrů (viz kapitola 2.1.4).

4.2.2 Aplikace na MIFARE DESFire

V této kapitole bude postupně popsáno šest aplikací navržených pro bezkontaktní čipové karty MIFARE DESFire s ohledem na optimální poměr bezpečnost/rychlost. U každé aplikace je nejprve krátce specifikována požadovaná funkce, pro kterou je navržena a poté podrobněji všechny její soubory. Struktura souborů je názorně zobrazena tabulkou uvádějící názvy jednotlivých položek, jejich formát, velikost, výchozí hodnotu (pokud je specifikována), typ editace definující subjekty oprávněné danou položku přepisovat a jejich stručný popis. Sloupec „Typ souboru“ určuje, která část souboru je či není šifrovaná. Výchozí hodnota jednotlivých položek je specifikována ve sloupci „Default“ a může v podstatě nabývat jedné ze tří podob:

- \- – výchozí hodnota není daná,
- hodnota – výchozí hodnota je daná zapsanou hodnotou,
- * – výchozí hodnota je daná neznámou hodnotou.

Jednotlivé role subjektů jsou v tabulkách zapsány ve zkratkách:

- SA – správce aplikace,
- OVyuzAA – organizace využívající aplikaci aktivně,
- OPoskDA – organizace poskytující data pro aplikaci aktivně,
- U – uživatel,
- SAM – SAM modul.

U každého sektoru a následně i pro celou aplikaci je uvedena celková velikost, kterou bude zapotřebí na kartě vyhradit. Součet velikostí všech aplikací specifikuje typ paměti EEPROM, která má být použita.

Po závěrečném souhrnu specifických požadavků všech aplikací na strukturu, zabezpečení, celkové jejich funkce a velikost EEPROM budeme schopni určit nejvhodnější typ použité multiaplikační karty

4.2.2.1 0xF00041 – Personalizace karty

Aplikace 0xF00041 s názvem „Personalizace karty“ ukládá základní informace o vydavateli a držiteli karty. Informace o vydavateli karty ho umožňují identifikovat, dohledat a kontaktovat, což může být vyžadováno pro nahrání nových aplikací na kartu, reklamaci či jiné důvody. Identifikace držitele karty je důležitá pro většinu subjektů pracujících s aplikacemi uloženými na kartě. Zároveň mnoho aplikací s informacemi o držiteli karty pracuje a je pro ně výhodnější, když spolupracují s aplikací personalizace a získají potřebná data od ní, místo toho, aby data získala sama a uchovávala je zvlášť. Informace o kartě samotné, jejím UID (hardwarové identifikační číslo), výrobci, doby životnosti apod. jsou uloženy v oddělené části paměti, kam byly zapsány během procesu výroby.

Tato aplikace je celkově tvořena čtyřmi soubory. V prvním souboru jsou zapsány informace o vydavateli karty a její speciální logické číslo, v dalších dvou souborech jsou uloženy základní informace o držiteli a jeho nároku na základní slevy týkající se jeho osobnosti (věku, školní docházky, zdravotního stavu) a posledním souborem je soubor Sign.

Soubor 0 – CardPublisherInfo

- číslo souboru 0
- název souboru CardPublisherInfo
- typ souboru standardní datový soubor
- popis souboru:

Soubor CardPublisherInfo ukládá identifikátor vydavatele karty a jeho transportní sítě, díky kterým by mohl být v online systému dohledán. Navíc však obsahuje i jeho konkrétní kontaktní informace, jako je jeho jméno, adresa, telefonní a faxové číslo pro případ, že by nebylo možné dané informace online dohledat. Navíc je zde uloženo logické číslo karty (LUID), které je za účelem zvýšení bezpečnosti vytvořeno a používáno jako identifikátor karty namísto hardwarového identifikátoru uloženého v oddělené části paměti. Společně s LUID je zde uložena verze klíče, kterým je tento identifikátor zašifrován, a podepsaná hodnota LUID.

LUID karty, jehož struktura je zobrazena na Obr. 13, je vytvořeno na základě specifikace normy ČSN ISO/IEC 7812. LUID je tvořeno 18 čísly rozdělenými do tří částí:

- **Identifikační číslo vydavatele (INN)**
  – definováno standardem a národním normalizačním institutem (v České republice ČNI),
  – struktura pro všechny karty stejná:

- Oblast zařazení (MII)
  – 1 číslice,
  – pro ČNI – 9;
- Kód země
  – 2–4 číslice,
  – dle ČSN EN ISO 3166-1: CZ – 203;
- Kód vydavatele
  – 4 číslice,
  – přidělen na základě žádosti o přidělení čísla vydavatele karet ČNI.

- **Identifikace individuálního účtu**
  - Kód výrobce
    – 2 číslice, nemusí být využito
    – definuje výrobce karet (např. konkrétního dopravce), který kartu „vyrobí“, neboli aktivuje;
  - Pořadové číslo karty
    – 7 číslic,
    – pořadové číslo „vyrobené“ aktivované karty (začíná se od 0000001).
- **Kontrolní číslice**
  – kontrolní číslice vypočtena ze všech předchozích číslic (PAN) pomocí Luhnova vzorce pro kontrolní číslici modulo-10 dle normy ČSN ISO/IEC 7812-1.

LOGICKÉ ČÍSLO KARTY																	
Identifikační číslo vydavatele (IIN)								Identifikace individuálního účtu									Kontrolní číslice
MII	kod země			kód vydavatele				kód výrobce		pořadové číslo karty							KČ
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
9	2	0	3	V	V	V	V	D	D	C	C	C	C	C	C	C	K

Obr. 13: Struktura Logického čísla karty dle ČSN ISO/IEC 7812 – MII: oblast zařazení, VVVV: čtyřmístný kód vydavatele, DD: dvoumístný kód výrobce, CCCCCCC: sedmimístné pořadové číslo karty, K: kontrolní číslice [19]

[19]

Soubor CardPublisherInfo je volně přístupný pro čtení, ale upravovat v něm data smí pouze správce aplikace – v tomto případě sám vydavatel karty. Je zde použit podpis souboru. Tab. 5 zobrazuje jeho vytvořenou strukturu.

Tab. 5: Aplikace 0x0F00041 – Personalizace karty: soubor CardPublisherInfo

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
PublisherID	Binary	24	0	SA	ID vydavatele karty	šifrovaný
PublNetworkID	Binary	24	0	SA	ID transportní sítě vydav. karty	
Name	Binary	200	-	SA	Jméno vydavatele karty	
Address	Binary	500	-	SA	Adresa vydavatele karty	
PhoneNum	Binary	104	-	SA	Telefon vydavatele karty	
FaxNum	Binary	104	-	SA	Fax vydavatele karty	
CardLogicUID	BDC	72	-	SA	LUID	
SignatureVersion	Binary	8	1	SA	Verze klíče ECDSA[2]	
SignedUID	Binary	448		SA	Podepsané LUID privát. klíčem	
Sign	Binary	64	-	SAM	Podpis souboru	
Reserve	Binary	36	-	SA	RFU	

Velikost souboru: 1 608 b = 201 B

[2] Klíč ECDSA (Elliptic Curve Digital Signature Algorithm) – typ digitálního podpisu za použití eliptických křivek

Soubor 1 – CardHolderInfo

- číslo souboru 1
- název souboru CardHolderInfo
- typ souboru standardní datový soubor
- popis souboru:

Soubor CardHolderInfo ukládá základní údaje držitele karty, kterými jsou křestní jméno, příjmení, datum narození, pohlaví a jedinečné identifikační číslo držitele v rámci systému. Systémem se zde myslí soubor všech subjektů (firem, organizací apod.), které akceptují či jinak využívají alespoň jednu aplikaci na dané kartě. Uživateli tak stačí jedno identifikační číslo pro svoji registraci u subjektů, které spolu jinak nemají nic společného. Ne všechny zmíněné údaje, které jsou editovatelné pouze správcem aplikace, musí být povinně vyplněny – záleží na typu karty, který si uživatel zvolí (každý typ má své specifické podmínky využití). Jelikož se jedná o veřejně přístupnou informaci, zvolený typ karty je uložen v položce CardHolderType v nešifrované části souboru. Tato položka nabývá následujících hodnot:

- 0_{10} – anonymní,
- 1_{10} – personalizovaná,
- 2_{10} – přenosná,
- 3_{10} – nepřenosná nepersonalizovaná,
- 4_{10} – náhradní karta,
- 5_{10} – zaměstnanecká graficky personalizovaná,
- 6_{10}-7_{10} – RFU.

Pohlaví je určeno v položce Sex dle následující logiky:

- 0_{10} – není zadáno,
- 1_{10} – muž,
- 2_{10} – žena,
- 3_{10} – není použito.

Jelikož se jedná o uložiště důležitých identifikátorů uživatele, je zde při každé změně dat použit princip podpisu souboru. Vytvořená struktura je zobrazena v Tab. 6.

Tab. 6: Aplikace 0x0F00041 – Personalizace karty: soubor CardHolderInfo

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
CardHolderType	Binary	3	1	SA	Typ karty dle držitele	
Reserve	Binary	10	-	SA	RFU	
FirstName	Binary	160	-	SA	Jméno držitele	šifrovaný
Surname	Binary	240	-	SA	Příjmení držitele	
BirthDate	DateStamp	14	-	SA	Datum narození	
Sex	Binary	2	0	SA	Pohlaví držitele	
ID	Binary	80	-	SAM	ID držitele	
Sign	Binary	64	-	SAM	Podpis souboru	
Reserve	Binary	132	-	SA	RFU	

Velikost souboru: 720 b = 90 B

Soubor 2 – CardHolderDiscount

- číslo souboru 2
- název souboru CardHolderDiscount
- typ souboru standardní datový soubor
- popis souboru:

V souboru CardHolderDiscount se zaznamenávají informace o nároku držitele karty na slevu vztahující se k jeho stavu – zda je dítětem či seniorem, žákem či studentem školy nebo zdravotně postiženou osobou. Kromě potvrzení či zamítnutí nároku zapsaného booleovskou hodnotou je u každé slevy zaznamenáno období její platnosti. Změnu dat o nároku na slevu v tomto souboru může provádět správce aplikace, organizace využívající aplikaci aktivně nebo oprávněná organizace poskytující data aplikaci, přičemž na její potvrzení musí podepsat celou aplikaci.

V případě slev vztahujících se k věku držitele karty – dětská, senior – by mohlo být namítnuto, že je sleva na kartě zapsána zbytečně, protože ve chvíli, kdy je na kartě uloženo jeho datum narození, systém může jednoduše dopočítat jeho věk a určit, zda na slevu má či nemá nárok. Tato úvaha je sice správná, ale proces dopočítávání věku je výpočetně (tedy i časově) pro systém náročnější než pouhé přečtení položky potvrzující či vyvracející nárok.

Struktura souboru CardHolderDiscount jezobrazena v Tab. 7.

Tab. 7: Aplikace 0x0F00041 – Personalizace karty: soubor CardHolderDiscount

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Reserve	Binary	13	-	SA	RFU	
ChildD1	Boolean	1	0	SA OVyuzAA	Dětská sleva (0-5 let)	šifrovaný
EndChildD1	DateStamp	14	-	SA OVyuzAA	Konec platnosti Dětské slevy (0-5 let)	
ChildD2	Boolean	1	0	SA OVyuzAA	Dětská sleva (6-10 let)	
EndChildD2	DateStamp	14	-	SA OVyuzAA	Konec platnosti Dětské slevy (6-10 let)	

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
65+D	Boolean	1	0	SA OVyuzAA	Sleva pro lidi starší 65 let	šifrovaný
End65+D	DateStamp	14	-	SA OVyuzAA	Konec platnosti Slevy pro lidi starší 65let	
70+	Binary	1	0	SA OVyuzAA	Sleva pro lidi starší 70let	
PupilD	Boolean	1	0	OVyuzAA OPoskytDA	Žákovská sleva (6-15 let)	
StartPupilD	DateStamp	14	-	OVyuzAA OPoskytDA	Začátek platnosti Žákovské slevy	
EndPupilD	DateStamp	14	-	OVyuzAA OPoskytDA	Konec platnosti Žákovské slevy	
StudentD	Boolean	1	0	OVyuzAA OPoskytDA	Studentská sleva (16-26 let)	
StartStudentD	DateStamp	14	-	OVyuzAA OPoskytDA	Začátek platnosti Studentské slevy	
EndStudentD	DateStamp	14	-	OVyuzAA OPoskytDA	Konec platnosti Studentské slevy	
DisabledD	Boolean	1	0	OVyuzAA OPoskytDA	Sleva pro zdravotně postižené	
StartDisabledD	DateStamp	14	-	OVyuzAA OPoskytDA	Začátek platnosti Slevy pro zdravotně postižené	
EndDisabledD	DateStamp	14	-	OVyuzAA OPoskytDA	Konec platnosti Slevy pro zdravotně postižené	
Reserve	Binary	59	-	SA	RFU	

Velikost souboru: 216 b = 27 B

Soubor 3 – Sign

- číslo souboru 3
- název souboru Sign
- typ souboru zálohovací soubor
- popis souboru:

Jak bylo zmíněno v úvodu kapitoly, posledním souborem aplikace je soubor Sign, který obsahuje master klíč celé aplikace, klíče pro čtení a zápis do jednotlivých souborů, zápisový klíč RFU prostoru a přístupová práva ke všem souborům aplikace. Kromě těchto položek je zde pro tuto aplikaci uložena také booleovská položka FreezeCard, která v případě zjištění nebezpečných aktivit zajistí blokaci aplikace, což povede k automatickému zablokování celé karty. Změny v tomto souboru smí provádět pouze správce aplikace. (Viz Tab. 8.)

Jelikož soubor 0 je celý volně přístupný pro čtení, je vytvořen pouze klíč A pro zápis dat. Druhou změnou oproti zavedenému systému zápisu dvou klíčů pro jednu aplikaci je využití jednoho čtecího klíče B pro soubory CardHolderInfo a CardHolderDiscount. Je to způsobeno tím, že uložená data jsou obdobné informativní úrovně a mohou být přístupná stejným subjektům. Využití dvou různých klíčů by bylo zbytečné.

Tab. 8: Aplikace 0x0F00041 – Personalizace karty: soubor Sign

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
MasterKeyAp	Binary	128	*	SA	Master klíč aplikace	šifrovaný
CardSignA	Binary	128	*	SA	Klíč A souboru 0	
HolderSignA	Binary	128	*	SA	Klíč A souboru 1	
HolderDiscSignA	Binary	128	*	SA	Klíč A souboru 2	
HolderSignB	Binary	128	*	SA	Klíč B souboru 1 a 2	
SignSignA	Binary	128	*	SA	Klíč A souboru 3	
SignSignB	Binary	128	*	SA	Klíč B souboru 3	
RFUSign	Binary	128	*	SA	Klíč RFU prostoru	
FreezeCard	Boolean	1	0	SA	Blokace aplikace	
Access	Binary	24	*	SA	Přístupová práva souborů	

Velikost souboru: 1 049 b = 131,125 B

Celková velikost aplikace „Personalizace karty“ je **3 593 b = 449,1 B**.

4.2.2.2 0xF88342 – Elektronická peněženka

Jako druhou zde uvedeme aplikaci „Elektronická peněženka“ s AID 0xF88342. S touto aplikací má uživatel možnost mít na kartě uložený určitý peněžitý obnos, který může využít při online a off-line platbě u různých subjektů akceptujících tuto aplikaci místo platby v hotovosti či klasickou kreditní nebo debetní kartou.

Uživatel může využívat jak debetních, tak kreditních transakcí. To znamená, že peníze na kartu může nahrávat buď předem (libovolnou částku), nebo až po skončení daného období, kdy s kartou libovolně platil, (minimálně částku vyrovnávající záporný stav konta). Platba může být provedena jak za služby spojené s jinou aplikací na kartě (například platba jízdního dokladu), tak za jakékoli jiné, s kartou nesouvisející služby (platba v restauraci akceptující kartu). Vyrovnání mezi jednotlivými subjekty přijímajícími či nahrávajícími peníze z/na kartu probíhá v rámci clearingu daného systému. Na kartě je uloženo několik posledních transakcí provedených peněženkou pro případ nutnosti dohledání platební historie a hlídání správné návaznosti jejich dat.

Aplikace je rozdělena do šesti souborů. V prvním a druhém souboru jsou uloženy obecné parametry peněženky, jejich rozdělením do dvou souborů získáme možnost nastavení různých přístupových práv dle citlivosti uložených parametrů. Dále zde máme jeden hodnotový soubor, ve kterém je uložena aktuální hodnota uložených peněz. Následují soubory CreditLog a DebetLog, ve kterých jsou specifikovány parametry provedených kreditních a debetních transakcí. Posledním souborem je soubor Sign.

Soubor 0 – BasicParameters

- číslo souboru 0
- název souboru BasicParameters
- typ souboru standardní datový soubor
- popis souboru:

První soubor, BasicParameters, obsahuje základní parametry aplikace specifikující v podstatě mantinely jejího použití, které smí měnit pouze správce aplikace. Mezi tyto parametry patří identifikátor vydavatele aplikace a jeho transportní sítě, datum vypršení platnosti nahrané verze, použitá měna a maximální povolená hodnota, která může být v peněžence uložena. Dále je zde uvedena hodnota vratné zálohy, specifikován den splacení využitého kreditu z předchozího měsíce a vyhrazen prostor pro RFU. Tato struktura je zobrazena v Tab. 9. Změna uložených dat je potvrzena podpisem aplikace.

Tab. 9: Aplikace 0xF88342 – Elektronická peněženka: soubor BasicParameters

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Reserve	Binary	13	-	SA	RFU	
ApPublisher	Binary	24	-	SA	Vydavatel aplikace	šifrovaný
ApPublNetwork	Binary	24	-	SA	ID transportní sítě vydavatele aplikace	
ExpirationDay	DateStamp	14	-	SA	Expirace platnosti EP	
Currency	Binary	10	203	SA	Měna	
MaxValue	Binary	16	5000	SA	Maximální hodnota	
ReturnDeposit	Binary	16	100	SA	Vratná záloha	
DayCreditBack	Binary	5	1	SA	Den v měsíci, kdy musí být stav kreditu 0	
Reserve	Binary	35	-	SA	RFU	

Velikost souboru: 168 b = 21 B

Soubor 1 – ExtendedParameters

- číslo souboru 1
- název souboru ExtendedParameters
- typ souboru standardní datový soubor
- popis souboru:

Takzvané rozšiřující parametry jsou uloženy v souboru ExtendedParameters a mohou být editovány kromě správce aplikace také samotným jejím uživatelem prostřednictvím k tomu určených terminálů. Jedná se o data, jejichž nastavením si uživatel může specifikovat základní parametry prováděných transakcí a definovat tak uživatelské zabezpečení. Lze si nastavit maximální povolenou hodnotu kreditu, který uživatel za specifikované období může využít – čímž si vytvoří pojistku, aby nebyl příliš v mínusu. Vhodné je specifikovat limit jednorázové či denní platby udávající částku, která může být zaplacena během jedné transakce, popřípadě během jednoho dne. Dále pak uživatel může určit povolený počet operací provedených za den a maximální hodnotu jedné platby bez nutnosti použití PIN.

Změna samotného bezpečnostního PINu je umožněna také. Konkrétní struktura tohoto souboru je zobrazena v Tab. 10. Jestliže změny provádí uživatel přes terminál, celý soubor je po konečném potvrzení podepsán SAM modulem daného zařízení. V případě, že změny provádí správce aplikace, podepisována je pouze celá aplikace.

Tab. 10: Aplikace 0xF88342 – Elektronická peněženka: soubor ExtendedParameters

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
MaxCredit	Binary	16	0	SA / U	Max hodnota povoleného kreditu	šifrovaný
MaxOnePay	Binary	16	0	SA / U	Max hodnota jedné platby	
MaxDayOper	Binary	8	0	SA / U	Max počet operací za den	
MaxADay	Binary	16	0	SA / U	Max hodnota denní platby	
MaxNoPINOnePay	Binary	16	500	SA / U	Max hodnota jedné platby bez PINu	
PIN	Binary	20	1234	SA / U	PIN pro platbu	
LastChange Time	TimeReal	32	-	SAM	Datum a čas provedení poslední změny parametrů	
Sign	Binary	64	-	SAM	Podpis SAM modulu provádějícího zařízení	
Reserve	Binary	28	-	SA	RFU	

Velikost souboru: 240 b = 30 B

Soubor 2 – ValueEP

- číslo souboru 2
- název souboru ValueEP
- typ souboru hodnotový soubor se zálohováním
- popis souboru:

Hodnotový soubor ValueEP, zobrazený v Tab. 11, zaznamenává aktuálně uloženou hodnotu peněz v aplikaci. Díky použití speciálního typu souboru je uložená hodnota řádně zálohována a chráněna proti neoprávněným manipulacím. Změny tohoto souboru jsou podepisovány systémovým podpisem SAM modulu provádějícím transakci.

Tab. 11: Aplikace 0xF88342 – Elektronická peněženka: soubor ValueEP

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
ValueEP	Binary	24	0	SAM	Aktuální hodnota EP	šifrovaný
SystemSign	Binary	64	-	SAM	Systémový podpis souboru	

Velikost souboru: 88 b = 11 B

Soubor 3 – CreditLog

- číslo souboru 3
- název souboru CreditLog
- typ souboru cyklicky zaznamenávající soubor se zálohováním – 10 záznamů
- popis souboru:

Soubor CreditLog zaznamenává posledních 10 kreditních transakcí provedených peněženkou. Pro každou transakci je zapsáno datum a čas provedení, její pořadové číslo během daného dne, hodnota elektronické peněženky před transakcí, hodnota převedených peněz během transakce a typ transakce určující, zda se jedná o nahrání peněz na kartu či platbu – hodnota se v souboru ValueEP přičítá či odečítá. Tato data slouží pro kontrolu návaznosti transakcí. Dále jsou zde zapsána identifikační čísla SAM modulu a zařízení provádějícího operaci, díky jimž je subjekt aktivující transakci dohledatelný. Každá takováto operace je navíc podepsána provádějící organizací – jejím SAM modulem. Položky v tomto souboru mohou být přepisovány správcem aplikace nebo organizací využívající aplikaci aktivně.

Struktura tohoto souboru je zobrazena v Tab. 12.

Tab. 12: Aplikace 0xF88342 – Elektronická peněženka: soubor CreditLog

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
CreditDateTime	TimeReal	32	-	SAM	Datum a čas kredit. trans.	šifrovaný
OrdNumTrans	Binary	8	0	SAM	Pořadové číslo transakce	
LastCreditValue	Binary	16	0	SAM	Hodnota EP před trans.	
TransType	Binary	1	0	OVyuzAA	Typ transakce: 0_{10} – platba 1_{10} – nahrání peněz	
CreditValue	Binary	16	0	OVyuzAA	Hodnota kreditní trans.	
ChangeDevice	Binary	64	-	SAM	Číslo zařízení prov. trans.	
SAMNum	Binary	80	-	SAM	Číslo SAM modulu, kde byla transakce provedena	
Signature	Binary	64	-	SAM	Podpis řádku	
Reserve	Binary	46	-	SA	RFU	

Jelikož se jedná o cyklicky zaznamenávající soubor se zálohováním, položky týkající se jedné transakce – řádky 6–13 – jsou v paměti ukládány jako jeden záznam. S každou další transakcí je vytvořen nový záznam. V paměti je vymezen prostor pro daný počet záznamů – v našem případě 10 záznamů. Ve chvíli, kdy jsou zaplněny všechny záznamy a je prováděna další operace, její data jsou ukládána na místo nejstaršího zapsaného záznamu – jsou cyklicky přepisována. Při počítání velikosti celého souboru musíme vzít v úvahu prostor vyhrazený pro všechny záznamy – řádky 6–13 započítáme desetkrát. Jelikož se jedná o vysoce citlivá data, je zde podepisován každý záznam – používá se podpis řádků.

Velikost souboru: 2 880 b = 360 B

Soubor 4 – DebetLog

- číslo souboru 4
- název souboru DebetLog
- typ souboru cyklicky zaznamenávající soubor se zálohováním – 10 záznamů
- popis souboru:

Debetní transakce elektronické peněženky jsou zapisovány do souboru DebetLog, kde může být v jeden okamžik zapsáno až 10 posledních záznamů. Struktura tohoto souboru je v podstatě shodná se strukturou CreditLog s jedinou výjimkou, že debetní jsou pouze transakce platby, nikoli nahrání peněz na kartu. Z toho důvodu soubor neobsahuje položku TransType. V Tab. 13 se můžete podívat na celou strukturu DebetLog.

Tab. 13: Aplikace 0xF88342 – Elektronická peněženka: soubor DebetLog

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
DebetDateTime	TimeReal	32	-	SAM	Datum a čas debet.transakce	šifrovaný
OrdNumTrans	Binary	8	0	SAM	Pořadové číslo transakce	
LastDebetValue	Binary	16	0	SAM	Hodnota EP před trans.	
DebetValue	Binary	16	0	OVyuzAA	Hodnota debetní trans.	
ChangeDevice	Binary	64	-	SAM	Číslo zařízení prov. trans.	
NumberSAM	Binary	80	-	SAM	Číslo SAM modulu, kde byla transakce provedena	
Signature	Binary	64	-	SAM	Podpis řádku	
Reserve	Binary	40	-	SA	RFU	

Jak bylo napsáno, strukturou odpovídá DebetLog souboru předchozímu, proto i zde musíme počítat s dostatečným prostorem pro všechny záznamy – řádky 6–12 počítáme desetkrát.

Velikost souboru: 2 864 b = 358 B

Soubor 5 – Sign

- číslo souboru 5
- název souboru Sign
- typ souboru zálohovací soubor
- popis souboru:

Soubor Sign v této aplikaci kromě přístupových klíčů a práv obsahuje také položku pro zablokování aplikace v případě, že dojde k neoprávněné manipulaci s aplikací (změna údajů, neoprávněný převod peněz apod.) – booleovská hodnota "0" značí, že je aplikace v pořádku, neblokovaná. Díky této položce je informace o blokaci dostupná přímo na kartě, díky čemuž jsou i off-line systémy bez aktualizovaného black-listu informováni. Soubor ukládá také číslo aktuálního dne v roce a počet provedených operací během daného dne. Tyto údaje pomáhají kontrole návaznosti transakcí a celistvosti a správnosti informací o nich. Celá struktura souboru Sign je ukázána v Tab. 14.

Každému souboru jsou zde přiřazeny dva šifrovací klíče A a B. Jedinou výjimkou zde jsou soubory BasicParameters a ExtendedParameters, které sdílejí klíč B. V obou případech se jedná o soubor ukládající důležité parametry aplikace, které mají být přístupné k přečtení stejnými subjekty. Přestože soubory CreditLog a DebetLog také obsahují velice podobná data, jedná se přeci jen o poměrně citlivá data a z hlediska bezpečnosti je lepší, když budou mít klíče oddělené.

Tab. 14: Aplikace 0xF88342 – Elektronická peněženka: soubor Sign

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
MasterKeyA	Binary	156	*	SA	Master klíč aplikace	šifrovaný
BasicParSignA	Binary	156	*	SA	Klíč A souboru 0	
ExtParSignA	Binary	156	*	SA	Klíč A souboru 1	
ParSignB	Binary	156	*	SA	Klíč B souboru 0 a 1	
ValueSignA	Binary	156	*	SA	Klíč A souboru 2	
ValueSignB	Binary	156	*	SA	Klíč B souboru 2	
CreditLogSignA	Binary	156	*	SA	Klíč A souboru 3	
CreditLogSignB	Binary	156	*	SA	Klíč B souboru 3	
DebetLogSignA	Binary	156	*	SA	Klíč A souboru 4	
DebetLogSignB	Binary	156	*	SA	Klíč B souboru 4	

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
SignSignA	Binary	156	*	SA	Klíč A souboru 5	
SignSignB	Binary	156	*	SA	Klíč B souboru 5	
RFUSign	Binary	156	*	SA	Klíč RFU prostoru	
FreezeEP	Boolean	1	0	SAM	Blokace EP	
DayNo	Binary	5	0	SAM	Číslo kalendářního dne	
SumPerDay	Binary	8	0	SAM	Počet transakcí za den	
Access	Binary	36	*	SA	Přístupová práva soub.	

Velikost souboru: 2 234 b = 279,25 B

Celková velikost aplikace „Elektronická peněženka“ je **8 474 b = 1 059,3 B**.

4.2.2.3 0xF12173 – Jízdní doklad

Aplikace 0xF12173 nazvaná „Jízdní doklad“ ukládá v elektronické podobě uživatelem zakoupené aktuální jízdní doklady. Naše aplikace vydává hned několik druhů jízdních dokladů, jsou jimi jízdenky z výchozí stanice do cílové, jízdní kupony zónové, kupony síťové a kilometrický pas. U každé z těchto možností může držitel karty navolit několik parametrů, kterými specifikuje mimo jiné počet cestujících a jejich různorodé nároky na slevu, počet zakoupených jízd nebo období, po které je doklad platný, a požadované přidané služby, jako jsou místenka, přeprava zavazadla či zvířete. Aplikace poskytuje možnost mít zároveň na kartě nahráno více jízdních dokladů nejenom různého duhu, ale také druhu stejného, přičemž jednotlivé doklady mohou mít odlišné parametry.

Z celkových 20 souborů aplikace jich 17 ukládá jízdní doklady, jeden zaznamenává služby dokoupené k jízdenkám a dva soubory, BasicParameters a Sign, nesou data o aplikaci jako celku. Mezi 17 soubory jízdních dokladů je pět souborů jízdenek, pět souborů kuponů zónových a pět kuponů síťových. Poslední dva soubory ukládají údaje kilometrického pasu.

Soubor 0 – BasicParameters

- číslo souboru 0
- název souboru BasicParameters
- typ souboru standardní datový soubor
- popis souboru:

V souboru BasicParameters jsou uloženy základní parametry aplikace, mezi které patří identifikátory vydavatele aplikace a jeho transportní sítě společně s datem vypršení platnosti aktuálně nahrané verze. Jeho struktura je zobrazena v Tab. 15.

Data má právo přepisovat pouze správce aplikace, který při zápisu podpisuje celou aplikaci.

Tab. 15: Aplikace 0xF12173 – Jízdní doklad: soubor BasicParameters

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Reserve	Binary	13	-	SA	RFU	
ApPublisher	Binary	24	0	SA	Vydavatel aplikace	šifrovaný
ApPublNetwork	Binary	24	-	SA	ID transportní sítě vydavatele aplikace	
ExpirationDay	DateStamp	14	-	SA	Datum vypršení platnosti	
Reserve	Binary	26	-	SA	RFU	

Velikost souboru: 112 b = 14 B

Soubor 1–5 – OriginDestinationTicket

- číslo souboru 1–5
- název souboru OriginDestinationTicket
- typ souboru standardní datový soubor
- popis souboru:

První zmíněná pětice souborů ukládá jízdní doklad typu jízdenka z výchozí stanice do cílové. Všechny tyto soubory mají naprosto shodnou strukturu a jsou nazývány OriginDestinationTicket. Každý soubor uchovává jízdní doklad – jízdenku – s parametry definovanými uživatelem, přičemž jedním z těchto parametrů je i počet dokladů stejných parametrů, popřípadě období, po které doklad platí. Tudíž slouží jako zásobník totožných jízdenek. Díky tomu, že takovýchto souborů zde máme uloženo pět, můžeme mít na kartě současně uloženo až pět dokladů typu jízdenka z výchozí do cílové stanice různých parametrů, přičemž každý tento typ tam může být nahrán vícekrát. Na stejném principu jsou navrženy také jízdní doklady druhu kupon zónový a kupon síťový popsané v následujících částech kapitoly.

Jelikož tedy víme, že všech pět souborů má totožnou strukturu, budeme zde popisovat pouze jeden tento soubor. Popis je doplněn názorným zobrazením struktury v Tab. 16. Můžeme říci, že šifrovaná část souboru, respektive dokladu, má tři části. V první části jsou ukládány jeho základní parametry určující dopravce (dopravní společnost), jeho transportní síť a konkrétního řidiče, případně prodejce na pokladně, který prodal jízdní doklad, a společně s tím i identifikátor prodejního místa. Dále je zde uloženo identifikační číslo (ID) dokladu, datum a čas prodeje. Tyto údaje uživatel nijak neovlivňuje (kromě výběru dopravní společnosti, se kterou pojede).

Do druhé částí dokladu jsou zapisovány parametry, které definuje cestující, nebo se týkají samotné jízdy. Cestující nejprve volí cestu buď specifikací výchozí a cílové stanice, kde pro větší konkretizaci může určit i průjezdovou stanici, nebo zvolením ID konkrétní trasy dle číselníku. Cestující pokračuje volbou jednosměrné nebo zpáteční jízdenky, vozové třídy – pokud takovou možnost dopravní prostředek umožňuje – a počtu osob, pro které je určena. Jedna jízdenka může být koupena až pro 31 cestujících, přičemž každému cestujícímu lze přiřadit slevu, na kterou má nárok. Je zde zohledněn i fakt, že jeden cestující může mít současně nárok na více slev (například zákaznická a žákovská). Toho je dosaženo na základě principu, že každá sleva je procentuální srážkou ze základní ceny pro jednoho dospělého cestujícího. Doklad ukládá jak počet všech cestujících, tak celkový počet jednotlivých slev, na které mají cestující nárok – pokud má jeden cestující nárok na více slev, je zahrnut v obou položkách –, výsledná sleva je dopočtena součtem jednotlivých slev. Nárok

držitele karty na požadovanou slevu ověřuje systém sám v aplikacích „Personalizace karty“ a „Sleva“, nárok ostatních cestujících ověřuje buď prodejce sám, nebo také systém z jejich multiaplikačních karet. Kupující pokračuje volbou platnosti dokladu.

Naše aplikace umožňuje prodej jak jednorázových jízdenek zakoupených u řidiče, jejichž platnost začíná okamžikem nástupu/koupě, tak jízdenek zakoupených na jiných prodejních místech, kde čas začátku platnosti je zapsán prodavačem. Navíc, jak bylo řečeno, podporuje koupi až 15 jízd stejných parametrů najednou nebo jízdenky platné určitý počet dnů – cestující může po dané trase v daném období jet neomezeně krát. Při koupi několika jízdenek stejných parametrů je počet jízdenek zaznamenán do položky NumOfBoughtJourneys, odkud jsou při každé využité cestě postupně odečítány. Počet platných dnů cestující volí z možností: víkend, po-pá, sedmidenní, 30denní, 90denní, 180denní a roční.

V položce TicketExpiration je zapsán konec platnosti jízdenky. V případě koupě jedné jízdenky je zapsán datum a čas konce jízdy. Při koupi několika jízd je zapsáno datum, do kterého musejí být jízdy využity, jinak propadnou, – stanovuje dopravce ve svém přepravním řádu, například jeden rok od zakoupení. Pokud je zakoupen doklad na určitý počet dnů, je zde zapsána půlnoc posledního dne platnosti.

V souboru jsou specifikována také omezení pro doklad, týkající se jeho platnosti (například, že platí pouze v pracovní dny, během akademického roku apod.) a typu dopravních prostředků, které mohou být pro cestu použity (autobus, vlak, prostředky MHD apod.). Poté je zapsána cena odpovídající parametrům dokladu.

Třetí část dokladu slouží k záznamu odbavení dokladu. Když cestující jízdenku skutečně použije, je do dokladu zaznamenáno identifikační číslo SAM modulu zařízení, které doklad odbavilo. Podle tohoto identifikátoru je možné dohledat zařízení, ve kterém budou uchovány detailní informace o daném odbavení – datum, čas, apod. Tyto informace jsou důležité pro případné řešení reklamací dokladu. V paměti karty je takto uchováváno posledních pět záznamů o odbavení, starší záznamy jsou uloženy v systému a dohledatelné v něm.

Při jakékoli změně záznamů je celý soubor podepsán. Změny smí dělat správce aplikace (v nešifrované části) a organizace využívající aplikaci aktivně (v šifrované části).

Tab. 16: Aplikace 0xF12173 – Jízdní doklad: soubor OriginDestinationTicket

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
ContractProvID	Binary	24	-	OVyuzAA	Kód dopravní spol.	šifrovaný
NetworkID	Binary	24	-	OVyuzAA	ID transportní sítě dopr.spol.	
SaleAgent	Binary	16	-	OVyuzAA	ID prodavače dokladu	
SaleDevice	Binary	24	-	SAM	Číslo prodejního místa	
TicketID	Binary	24	0	SAM	ID dokladu	
TimeOfPurchase	TimeReal	32	-	SAM	Čas a den nákupu dokladu	
NumOfBoughtJourneys	Binary	4	0	OVyuzAA	Počet zakoupených jízd	
NumOfValidDays	Binary	3	0	OVyuzAA	Počet platných dnů	
Origin	Binary	32	-	OVyuzAA	ID výchozí stanice	
ThroughStation	Binary	32	-	OVyuzAA	ID průjezdná stanice	
Destination	Binary	32	-	OVyuzAA	ID cílové stanice	
NumOfLine	Binary	24	-	OVyuzAA	ID zakoupené trasy	
ReturnTicket	Boolean	1	0	OVyuzAA	Zpáteční jízdenka	
VehiclClassRestr	Binary	2	0	OVyuzAA	Povolená vozová třída	
NumPers	Binary	5	0	OVyuzAA	Počet všech cestujících	
NumOfChildD	Binary	5	0	OVyuzAA	Počet uplat. slev ChildD	
NumOf65+D	Binary	5	0	OVyuzAA	Počet uplat. slev 65+	
NumOf70+D	Binary	5	0	OVyuzAA	Počet uplat. slev 70+	
NumOfPupilD	Binary	5	0	OVyuzAA	Počet uplat. slev PupilD	
NumOfStudentD	Binary	5	0	OVyuzAA	Počet uplat.slev Student	

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
NumOfDisabledD	Binary	5	0	OVyuzAA	Počet uplat.slev Disabl.	
NumOfEmplD	Binary	5	0	OVyuzAA	Počet uplat. slev Empl.	
NumOfCustomD	Binary	5	0	OVyuzAA	Počet uplat. slev Cust.	
NumOfSpecD	Binary	5	0	OVyuzAA	Počet SpecialDiscount	
StartOfJourney	TimeReal	32	-	OVyuzAA SAM	Začátek cesty (den, čas)	
TicketExpiration	TimeReal	32	-	SAM	Konec platnosti dokladu	
ValidRestrictDays	Binary	16	0	OVyuzAA	Omezení platnosti dokl.	
TransMeansRestr	Binary	16	0	OVyuzAA	Povolené dopravní pros.	šifrovaný
Price	Binary	16	0	SAM	Cena dokladu	
SAMUsedTicket1	Binary	80	-	SAM	Číslo odbav. SAM mod.	
SAMUsedTicket2	Binary	80	-	SAM	Číslo odbav. SAM mod.	
SAMUsedTicket3	Binary	80	-	SAM	Číslo odbav. SAM mod.	
SAMUsedTicket4	Binary	80	-	SAM	Číslo odbav. SAM mod.	
SAMUsedTicket5	Binary	80	-	SAM	Číslo odbav. SAM mod.	
Signature	Binary	64	-	SAM	Podpis souboru	
Reserve	Binary	76	-	SA	RFU	

Velikost souboru: 1 000 b = 125 B

Při výpočtu celkové velikosti aplikace nesmíme zapomenout počítat tento soubor pětkrát, případně tolikrát, kolik těchto dokladů bude aplikace poskytovat.

Soubor 6–10 – Coupons_Zones

- číslo souboru 6–10
- název souboru Coupons_Zones
- typ souboru standardní datový soubor
- popis souboru:

V souborech pod názvem Coupons_Zones jsou ukládány jízdní doklady typu kupon zónový. Stejně jako u předchozího typu dokladu i zde bude dostatečné popsat pouze jeden z pěti totožných souborů. Navíc tento doklad má velice podobnou strukturu jako doklad typu jízdenka či kupon síťový, proto se během popisu budeme hodně odkazovat na strukturu souborů OriginDestinationTicket a zaměříme se především na popis odlišných vlastností. Celá struktura je následně zobrazena v Tab. 17.

I zde můžeme šifrovanou část souboru účelově rozdělit na tři části, přičemž první a třetí část jsou strukturou i účelem totožné s první a třetí částí dokladu popsaného u předchozí funkce. Rozdíly oproti struktuře dokladu jízdenka se týkají v podstatě pouze několika málo položek ve druhé části, především specifikací oblasti použití. Nekupuje se zde konkrétní trasa z bodu A do bodu B, ale celá oblast, ve které je doklad platný, proto zde místo položek výchozí, průjezdové a cílové trasy či jejího ID je uložen počet zakoupených zón, který může být doplněn o jejich přesný výčet. Specifikuje se také pokaždé doba platnosti kuponu, která může být: 30 min, 90 min, 3 hod, 1 den, 30 dnů, 90 dnů, 270 dnů a 1 rok. Stejným způsobem jako při koupi jízdenek je zde specifikován počet cestujících, jejich nárok na slevu a vozová třída, pro kterou je doklad zakoupen.

Může zde být zakoupeno také až 31 kuponů stejných parametrů, které se při každém využití odečítají z položky *NumOfCoupons*, tudíž když jsou vyčerpány, v položce je zapsána nula. Lze specifikovat počet cestujících i jejich slevy, vozovou třídu, typy dopravních prostředků i dny pro které je doklad platný.

Začátek doby platnosti může být specifikovaný prodejcem při vytváření dokladu nebo v případě kuponů na kratší časové období je zapsán v době odbavení zařízením, u kterého byl doklad cestujícím validován. V případě krátkodobých kuponů, platných většinou jen pro menší počet zón bez jejich specifikace, je zaznamenáno také ID nástupní zóny pro účely kontroly využití dokladu pro správný počet zón. Konec platnosti dokladu je dopočítáván dle jeho parametrů buď při jeho koupi, nebo validaci.

Při změně záznamů je podepisován celý soubor. Změny smí dělat správce aplikace a organizace využívající aplikaci aktivně.

Tab. 17: Aplikace 0xF12173 – Jízdní doklad: soubor Coupons_Zones

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
ContractProvID	Binary	24	-	OVyuzAA	Kód dopravní společnosti	šifrovaný
NetworkID	Binary	24	-	OVyuzAA	ID transportní sítě prod. dokl.	
SaleAgent	Binary	16	-	OVyuzAA	ID prodavače dokladu	
SaleDevice	Binary	24	-	SAM	Číslo prodejního místa	
CouponID	Binary	24	0	SAM	ID dokladu	
TimeOfPurchase	TimeReal	32	-	SAM	Čas a den nákupu dokl.	
NumOfCoupons	Binary	5	0	OVyuzAA	Počet koupených kuponů	
NumOfZones	Binary	4	0	OVyuzAA	Počet zakoupených zón	
ZonesIDs	Binary	120	-	OVyuzAA	ID zakoupených zón	
ValidityPeriod	Binary	3	-	OVyuzAA	Doba platnosti	
NumPers	Binary	5	0	OVyuzAA	Počet všech cestujících	
NumOfChildD	Binary	5	0	OVyuzAA	Počet uplat. slev ChildD2	
NumOf65+D	Binary	5	0	OVyuzAA	Počet uplat. slev 65+	
NumOf70+D	Binary	5	0	OVyuzAA	Počet uplat. slev 70+	
NumOfPupilD	Binary	5	0	OVyuzAA	Počet uplat. slev PupilD	
NumOfStudentD	Binary	5	0	OVyuzAA	Počet uplat.slev StudentD	
NumOfDisabledD	Binary	5	0	OVyuzAA	Počet uplat.slev Disabled	
NumOfEmplD	Binary	5	0	OVyuzAA	Počet uplat. slev Empl.	
NumOfCustomD	Binary	5	0	OVyuzAA	Počet uplat. slev Cust.	
NumOfSpecD	Binary	5	0	OVyuzAA	Počet SpecialDiscount	
VehiclClassRestr	Binary	2	0	OVyuzAA	Povolená vozová třída	

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
FromStation	Binary	32	-	SAM	ID nástupní zóny	šifrovaný
StartOfJourney	TimeReal	32	-	OVyuzAA SAM	Začátek platnosti (den a čas)	
TicketExpiration	TimeReal	32	-	SAM	Konec platnosti kuponu	
ValidRestrictDays	Binary	16	0	OVyuzAA	Omezení platnosti dokl.	
TransMeansRestr	Binary	16	0	OVyuzAA	Povolené dop.prostředky	
Price	Binary	16	0	OVyuzAA	Cena dokladu	
SAMUsedTicket1	Binary	80	-	SAM	Číslo odbav. SAM modulu	
SAMUsedTicket2	Binary	80	-	SAM	Číslo odbav. SAM modulu	
SAMUsedTicket3	Binary	80	-	SAM	Číslo odbav. SAM modulu	
SAMUsedTicket4	Binary	80	-	SAM	Číslo odbav. SAM modulu	
SAMUsedTicket5	Binary	80	-	SAM	Číslo odbav. SAM modulu	
Signature	Binary	64	-	SAM	Podpis souboru	
Reserve	Binary	80	-	SA	RFU	

Velikost souboru: 1040 b = 130 B

Při výpočtu celkové velikosti aplikace nesmíme zapomenout počítat tento soubor pětkrát, případně tolikrát, kolik těchto dokladů bude aplikace poskytovat.

Soubor 11–15 – Coupons_Network

- číslo souboru 11–15
- název souboru Coupons_Network
- typ souboru standardní datový soubor
- popis souboru

Kupony síťové jsou nahrávány do souborů Coupons_Network. Jejich náležitosti jsou velmi obdobné s kupony zónovými, a tedy i jízdenkami, s tím rozdílem, že jejich platnost není pro prostorové zóny ani pro jednotlivé trasy, nýbrž pro celou síť definovanou dopravcem. Proto i struktura souborů je velice obdobná a zde budou uvedeny především rozdíly, díky nimž jsou tyto doklady odlišné od ostatních. Celková struktura souboru je zobrazena v Tab. 18.

Kupon opět můžeme rozdělit na tři hlavní části, přičemž první část specifikuje prodejce jízdního dokladu a třetí uchovává záznamy o odbavení dokladu.

Změny ve druhé části vycházejí z odlišného popisu platnosti kuponu. Oblast platnosti je specifikována identifikačním číslem konkrétní zakoupené sítě, zatímco časová platnost je volena dle následujících možností použití kuponu:

- jednotlivé dny
  – cestující smí libovolně cestovat v dané síti v X zvolených dnech během období určité délky – například v průběhu 10 dnů od začátku validace smí v pěti libovolných dnech cestovat; začátek daného období – validace dokladu – je stanoven prodávajícím při koupi dokladu nebo cestujícím u odbavovacího terminálu první den použití; cestující může zvolit:
  - 1 den – platnost 1 rok od koupě,
  - 5 dnů v 10 dnech,
  - 10 dnů ve 30 dnech,
  - 30 dnů ve 180 dnech;
- dny v celku
  – cestující smí ve specifikovaném období libovolně cestovat v dané síti – začátek období je nastaven prodávajícím při koupi dokladu nebo cestujícím u odbavovacího terminálu první den použití; cestující může zvolit:
  - víkend,
  - 90 dnů,
  - 180 dnů,
  - 1 rok.

Pro specifikaci těchto variant je použita nejprve položka TypeOfCoupon, kde je specifikováno použití pro jednotlivé dny nebo dny v celku. V prvním případě je do položky NumOfBoughtD zapsán počet zakoupených dnů, čímž je i specifikován počet dnů období platnosti – při první validaci je zapsán den začátku do položky StartOfJourney a dopočítán den konce platnosti do položky TicketExpiration. Cestující v tomto případě musí provést opakovanou validaci kuponu každý den cestování, během této validace je snížen počet zakoupených dnů o jeden a zároveň do položky DayOfUse je uloženo aktuální datum. Díky zaevidovanému dni použití je zajištěna možnost kontroly správného použití – při kontrole má revizor potvrzeno, že kupon byl daný den použit.

V případě volby typu kuponu pro „dny v celku“ je v položce ValidityPeriod označena jedna ze čtyř možností použití. Poté je během validace, která je provedena pouze při koupi nebo v den první cesty, zároveň označen i začátek a konec období platnosti.

Ostatní nastavení dokladu – počet cestujících, nárok na slevu a omezení platnosti – je shodné jako v kuponech zónových. Při změně záznamů je podepsán celý soubor. Změny smí dělat správce aplikace a organizace využívající aplikaci aktivně.

Tab. 18: Aplikace 0xF12173 – Jízdní doklad: soubor Coupons_Network

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
ContractProvID	Binary	24	-	OVyuzAA	Kód prodejce dokladu	šifrovaný
ProvNetworkID	Binary	24	-	OVyuzAA	ID transportní sítě prodejce dokladu	
SaleAgent	Binary	16	-	OVyuzAA	ID prodavače dokladu	
SaleDevice	Binary	24	-	SAM	Číslo prodejního místa	
CouponID	Binary	24	0	SAM	ID dokladu	
TimeOfPurchase	TimeReal	32	-	SAM	Čas a den nákupu dokl.	
NumOfCoupons	Binary	5	0	OVyuzAA	Počet koupených kup.	
NetworkID	Binary	16	-	OVyuzAA	ID zakoupené sítě	
TypeOfCoupon	Binary	1	0	OVyuzAA	Typ kuponu	
NumOfBoughtD	Binary	5	0	OVyuzAA	Počet koupených dnů	
DayOfUse	DateStamp	14	-	SAM	Den použití kuponu	
ValidityPeriod	Binary	2	0	OVyuzAA	Doba platnosti	
NumPers	Binary	5	0	OVyuzAA	Počet všech cestujících	
NumOfChildD	Binary	5	0	OVyuzAA	Počet uplat.slev ChildD2	
NumOf65+D	Binary	5	0	OVyuzAA	Počet uplat. slev 65+	
NumOf70+D	Binary	5	0	OVyuzAA	Počet uplat. slev 70+	
NumOfPupilD	Binary	5	0	OVyuzAA	Počet uplat. slev PupilD	
NumOfStudentD	Binary	5	0	OVyuzAA	Počet uplat.slevStudentD	
NumOfDisabledD	Binary	5	0	OVyuzAA	Počet slev Disabled	

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
NumOfEmplD	Binary	5	0	OVyuzAA	Počet uplat. slev Empl.	šifrovaný
NumOfCustomD	Binary	5	0	OVyuzAA	Počet uplat. slev Cust.	
NumOfSpecD	Binary	5	0	OVyuzAA	Počet SpecialDiscount	
VehiclClassRestr	Binary	2	0	OVyuzAA	Povolená vozová třída	
StartOfJourney	DateStamp	14	-	OVyuzAA SAM	Den začátku platnosti	
TicketExpiration	DateStamp	14	-	SAM	Konec platnosti kuponu	
ValidRestrictDays	Binary	16	0	OVyuzAA	Omezení platnosti dokl.	
TransMeansRestr	Binary	16	0	OVyuzAA	Povolené dop.prostředk	
Price	Binary	16	0	OVyuzAA	Cena dokladu	
SAMUsedTicket1	Binary	80	-	SAM	Číslo odbav. SAM mod.	
SAMUsedTicket2	Binary	80	-	SAM	Číslo odbav. SAM mod.	
SAMUsedTicket3	Binary	80	-	SAM	Číslo odbav. SAM mod.	
SAMUsedTicket4	Binary	80	-	SAM	Číslo odbav. SAM mod.	
SAMUsedTicket5	Binary	80	-	SAM	Číslo odbav. SAM mod.	
Signature	Binary	64	-	SAM	Podpis souboru	
Reserve	Binary	85	-	SA	RFU	

Velikost souboru: 888 b = 111 B

Při výpočtu celkové velikosti aplikace nesmíme zapomenout počítat tento soubor pětkrát, případně tolikrát, kolik těchto dokladů bude aplikace poskytovat.

Soubor 16 – PrepaidKms

- číslo souboru 16
- název souboru PrepaidKms
- typ souboru standardní datový soubor
- popis souboru:

Soubor 16 společně se souborem 17 ukládá informace kilometrického pasu poskytovaného aplikací. Zatímco první soubor ukládá obecné nastavení a informace o jeho validaci, soubor 17 uchovává hodnotu nahraných – neprojetých – kilometrů v pasu. Tato funkce je dostupná

v celé dopravní síti dopravců zapojených do systému kilometrického pasu – většinou jen pár dopravců.

Princip poskytování služby spočívá v možnosti zakoupení určitého počtu kilometrů – omezen minimální počet přepravním řádem kilometrického pasu, například musí se nahrát minimálně 500 km –, které je možné výhodně využít při cestách na delší vzdálenosti. Při každé cestě je odečítán počet ujetých kilometrů. Počet kilometrů, který je odečten pro jednoho cestujícího za jednu cestu, je omezen dle přepravního řádu kilometrického pasu – například minimálně 100 km a maximálně 500 km za cestu. To dělá z této služby nevýhodnou v případě cestování na krátké trasy, ale výhodnou při cestování na trasy delší.

Cestující je povinen při nástupu do vozidla se vždy přihlásit s kartou do systému – vozidla musí disponovat terminály přizpůsobenými aplikaci – a při výstupu se zase odhlásit. Během odhlašování systém dopočítá podle interních informací počet ujetých kilometrů, které odečte ze zásobníku kilometrů. Na jeden kilometrický pas může zároveň cestovat více cestujících, ale kilometry jsou odečítány pro každého zvlášť. To znamená, že pokud spolu cestující ujedou 60 km, neodečte se ze zásobníku 120 kilometrů, ale 200 – v případě, že dopravce stanovil minimální denní limit na 100 km. I zde cestující mohou využít nároku na slevu, i když ne na všechny slevy dostupné u dokladů předešlých – typ dokladu to neumožňuje. Dostupné slevy jsou pouze dětská, pro seniory, studentská a speciální sleva nabízená dopravcem.

Strukturu dokladu, která je zobrazena v Tab. 19, můžeme opět rozdělit do více částí. V první části je uložen jedinečný identifikátor dokladu, který je neměnný, a údaje poslední provedené transakce nabití kilometrů – identifikátory specifikující prodejce, místo, čas, počet zakoupených kilometrů. Tyto údaje slouží ke zpětné kontrole údajů. Během každého nahrání je specifikováno datum, do kdy musí být kilometry vyčerpány – dáno přepravním řádem, například 1 rok od nákupu. Pokud by kilometry vyčerpány nebyly, došlo by k jejich propadnutí. Při nahrání kilometrůve chvíli, kdy není zásobník prázdný, se datum propadnutí posouvá.

V druhé části dokladu se ukládají data o jeho využití. Při nástupu na začátku cesty je karta přiložena k terminálu, který zobrazí držiteli karty aktuální stav zásobníku kilometrů. Pokud je v zásobníku méně kilometrů než limitní hodnota určená systémem, terminál kartu nevaliduje. Pokud je v zásobníku více kilometrů, cestující sám musí vědět, zda mu budou na cestu stačit, případně musí stav peněženky navýšit. Pokud je přeci jen využito více kilometrů, než jich má držitel předplaceno, hodnota zásobníku počítá do mínusu a držitel musí tyto kilometry zaplatit (případně i s přirážkou).

Po kontrole použitelných kilometrů v peněžence uživatel na terminálu navolí parametry jeho cesty – počet cestujících, jednotlivé nároky na slevu a vozovou třídu. Terminál tyto údaje

zapíše na kartu. Zároveň na ni nahraje identifikátor vozu, ve kterém se cestující nachází, a do svého systému naopak identifikátor karty.

Revizor při kontrole cestujících z dokladu zjistí, že je validován, kontrolou položky CoachID, která je vyplněna pouze během přepravy. Pokud je v aplikaci již limitní stav nahraných kilometrů, řeší situaci s cestujícím dle přepravního řádu – pokud cestující hodlá vystoupit včas, pouze ho zkontroluje. Na konci cesty držitel kartu opět přiloží k terminálu, který ji odhlásí ze systému a na kartě vymaže položku CoachID. Uživatel je opět informován o aktuálním stavu zásobníku.

Změny v dokladu smí provádět správce aplikace, organizace využívající aplikaci aktivně a uživatel. Při změně záznamů je podepisován celý soubor.

Tab. 19: Aplikace 0xF12173 – Jízdní doklad: soubor PrepaidKms

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
TicketID	Binary	24	0	SAM	ID dokladu	šifrovaný
LastContractProviderID	Binary	24	-	OVyuzAA	Kód dopravce, který naposledy prodal km	
LastProviderNetworkID	Binary	24	-	OVyuzAA	ID transportní sítě dopr., který naposledy prod. km	
LastCharge_SaleAgent	Binary	16	-	OVyuzAA	Pokladník, který naposledy prodal km	
LastCharge_SaleDevice	Binary	24	-	OVyuzAA	Číslo posledního prodejního místa	
LastChargeTime	TimeReal	32	-	SAM	Čas a den posledního nákupu km	
LastChargeKm	Binary	16	0	OVyuzAA	Počet km naposledy zakoupených	
TicketExpiration	DateStamp	32	-	SAM	Konec platnosti kuponu	

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
CoachID	Binary	24	-	SAM	ID vozu	
NumPers	Binary	5	0	U	Počet všech cestujících	
NumOfChildD	Binary	5	0	U	Počet uplat. slev ChildD2	
NumOf65+D	Binary	5	0	U	Počet uplat. slev 65+	
NumOfStudentD	Binary	5	0	U	Počet uplat.slevStudentD	šifrovaný
NumOfSpecD	Binary	5	0	U	Počet SpecialDiscount	
UsedVehicleClass	Binary	1	0	U	Použitá vozová třída	
Signature	Binary	64	-	SAM	Podpis souboru	
Reserve	Binary	86	-	SA	RFU	

Velikost souboru: 416 b = 52 B

Soubor 17 – VeluePrepaidKms

- číslo souboru 17
- název souboru ValuePrepaidKm
- typ souboru hodnotový soubor se zálohováním
- popis souboru:

Soubor ValuePrepaidKms zaznamenává aktuálně uloženou hodnotu zakoupených nevyužitých kilometrů kilometrického pasu definovaného v souboru PrepaidKms. Jeho struktura je zobrazena v Tab. 20.

Při koupi kilometrů se hodnota NumPrepaidKms zvyšuje o jejich počet, při jízdě na tento doklad je počet ujetých kilometrů odčten. Kilometry se odečítají v závislosti na přepočítávacím koeficientu použité slevy. Změny tohoto souboru jsou podepisovány systémovým podpisem SAM modulu provádějícím transakci.

Tab. 20: Aplikace 0xF12173 – Jízdní doklad: soubor ValuePrepaidKms

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
NumPrepaidKm	Binary	16	0	SAM	Aktuální počet předplac. km	šifrovaný
SystemSign	Binary	64	-	SAM	Systémový podpis	

Velikost souboru: 80 b = 10 B

Soubor 18 – AddService

- číslo souboru 18
- název souboru AddService
- typ souboru standardní datový soubor
- popis souboru:

Ke každému jízdnímu dokladu je možné přikoupit určitou přidanou službu. Tato možnost je zprostředkována souborem AddService. Naše aplikace nabízí jako přidanou službu přikoupení dokladu pro zvíře, platba za nadměrné zavazadlo a rezervace místenky. Strukturu souboru 18, zobrazenou v Tab. 21, můžeme rozdělit do čtyř částí.

V první části jsou uloženy položky zaznamenávající přidanou službu „lístek pro zvíře“. Struktura této části je velmi jednoduchá, skládá se celkem z 11položek, z nichž deset specifikuje jízdní doklad, ke kterému je lístek zakoupen, a jedenáctá položka je vyhrazena pro podpis změn – podpis řádku. Takto k jednomu jízdnímu dokladu může být zakoupeno více lístků. S vypršením platnosti dokladu je položka uvolněna.

V druhé části je na stejném principu zapisováno zavazadlo přikoupené k jízdnímu dokladu.

Třetí část souboru ukládá autobusové místenky, které mohou být zakoupeny z logických důvodů pouze k dokladům specifikujícím konkrétní linku, den a čas. Kromě položky specifikující jízdní doklad je u každé rezervace zapsáno číslo rezervovaného sedadla, pokud je na jízdní doklad odbaveno více cestujících, až deset z nich může mít rezervované místo. Jakákoli změna zápisu v oblasti místenka pro autobus je podepsána.

Místenky pro přepravu vlakem jsou ukládány ve čtvrté části. Zde pro přesnou specifikaci místenky je kromě identifikátoru jízdního dokladu zapsán datum cesty a číslo konkrétního vlaku, jeho vozu a sedačky. Takto konkrétním popisem může být zakoupeno i lehátko ve vozu, které má v něm svou specifickou polohu. Jakákoli změna zápisu v oblasti místenka pro vlak je podepsána.

Změny v tomto souboru smí provádět kromě správce aplikace také organizace využívající aplikaci aktivně i sám uživatel prostřednictvím terminálů k tomu určených.

Tab. 21: Aplikace 0xF12173 – Jízdní doklad: soubor AddService

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
Animal1	Binary	24	-	OVyuzAA U	ID přidruženého jízdního dokladu služby zvíře (1)	šifrovaný
Animal2	Binary	24	-	OVyuzAAU	ID přidruženého jízdního dokladu služby zvíře (2)	
…	…	…	…	…	…	
Animal10	Binary	24	-	OVyuzAAU	ID přidruženého jízdního dokladu služby zvíře (10)	
SignatureAn	Binary	64	-	SAM	Podpis služeb – zvíře	
Luggage1	Binary	24	-	OVyuzAAU	ID přidruženého jízdního dokl. služby zavazadlo (1)	
Luggage2	Binary	24	-	OVyuzAAU	ID přidruženého jízdního dokl. služby zavazadlo (2)	
…	…	…	…	…	…	
Luggage10	Binary	24	-	OVyuzAAU	ID přidruženého jízdního dokl. služby zavazadlo(10)	
SignatureLug	Binary	64	-	SAM	Podpis služeb – zavazadlo	
BusTicketSeatReservation1	Binary	24	0	OVyuzAAU	ID přidruženého dokladu služby místenka – bus (1)	
BusSeatReservation1	Binary	7	-	OVyuzAAU	Rezervované sedadlo – bus (1)	
…	…	…	…	…	…	
BusTicketSeatReservation10	Binary	24	0	OVyuzAAU	ID přidruženého dokladu služby místenka – bus (10)	

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
BusSeatReservation10	Binary	7	0	OVyuzAAU	Rezervované sedadlo – bus (10)	
SignatureBus	Binary	64	-	SAM	Podpis – místenka, bus	
TrainTicketSeatReservation1	Binary	24	0	OVyuzAAU	ID přidruženého dokladu služby místenka – vlak (1)	
DateSeatReservation1	Datestamp	14	-	OVyuzAAU	Specifikovaný den cesty(1)	
TrainTrainReservation1	Binary	36	-	OVyuzAAU	Číslo vlaku, ke kterému se rezervace vztahuje (1)	
TrainCoachReservation1	Binary	10	-	OVyuzAAU	Číslo vozu, ke kterému se rezervace vztahuje (1)	
TrainSeatReservation1	Binary	7	-	OVyuzAAU	Rezervované sedadlo – vlak (1)	šifrovaný
…	…	…	…	…	…	
TrainTicketSeatReservation10	Binary	24	0	OVyuzAAU	ID přidruženého dokladu služby místenka – vlak(10)	
DateSeatReservation10	Datestamp	14	-	OVyuzAAU	Specifikovaný den cesty (10)	
TrainTrainReservation10	Binary	36	-	OVyuzAAU	Číslo vlaku, ke kterému se rezervace vztahuje (10)	
TrainCoachReservation10	Binary	10	-	OVyuzAAU	Číslo vozu, ke kterému se rezervace vztahuje (10)	
TrainSeatReservation10	Binary	7	-	OVyuzAAU	Rezervované sedadlo – vlak (10)	
SignatureTrain	Binary	64	-	SAM	Podpis – místenka, vlak	
Reserve	Binary	76	-	SA	RFU	

Velikost souboru: 2 056 b = 257 B

Soubor 19 – Sign

- číslo souboru 19
- název souboru Sign
- typ souboru zálohovací soubor
- popis souboru:

Stejně jako u předchozích dvou aplikací i tato v souboru Sign obsahuje kromě ‚základních položek také položku FreezeCard, která v případě neoprávněné manipulace s aplikací – nahrání falzifikovaného jízdního dokladu, změna časových údajů platnosti apod. – může zajistit její blokaci. Pro udržení vysoké bezpečnosti souboru jsou pro každý typ souboru použity dva různé klíče – soubory 1–5, 6–11 a 11–15 mají v rámci pětice klíče shodné. Viz Tab. 22.

Tab. 22: Aplikace 0xF12173 – Jízdní doklad: soubor Sign

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
MasterKey	Binary	156	*	SA	Master klíč Aplikace	
BasicParSignA	Binary	156	*	SA	Klíč A bloku 0	
BasicParSignB	Binary	156	*	SA	Klíč B bloku 0	
TicketSignA	Binary	156	*	SA	Klíč A bloků 1–5	
CoupZonesSignA	Binary	156	*	SA	Klíč A bloků 6–10	
CoupNetwSignA	Binary	156	*	SA	Klíč A bloků 11–15	
TicketCoupSignB	Binary	156	*	SA	Klíč B bloků 11–15	
PrepaidKMsSignA	Binary	156	*	SA	Klíč A souboru 16	
ValPrepKmSignA	Binary	156	*	SA	Klíč A souboru 17	šifrovaný
PrepaidKmSignB	Binary	156	*	SA	Klíč B souboru 16–17	
AddServiceSignA	Binary	156	*	SA	Klíč A souboru 18	
AddServiceSignB	Binary	156	*	SA	Klíč B souboru 18	
SignSignA	Binary	156	*	SA	Klíč A souboru 19	
SignSignB	Binary	156	*	SA	Klíč B souboru 19	
RFUSign	Binary	156	*	SA	Klíč pro využití RFU prostoru	
FreezeApp	Boolean	1	0	SA	Blokace aplikace	
Access	Binary	120	*	SA	Přístupová práva souborů	

Velikost souboru: 2 461 b = 307,625 B

Celková velikost aplikace „Jízdní doklad“ je **19 765 b = 2 470,6 B**.

4.2.2.4 0xF02354 – Sleva

Aplikace „Sleva“ s přiděleným AID 0xF02354 slouží k potvrzení nároku držitele karty na určitou slevu. Uložené slevy se dělí do tří hlavních skupin. První jsou zaměstnanecké slevy, kde je počítáno i s tím, že uživatel může být zaměstnancem několika – až pěti – zaměstnavatelů zároveň. Druhou skupinou jsou slevy zákaznické, kdy je počítáno nejen s tím, že uživatel využívá zákaznickou slevu u více obchodníků – subjektů –, ale také s tím, že jeden subjekt poskytuje různé zákaznické slevy, a je tedy potřeba specifikovat přesný její typ, na který uživatel má nárok – jako například České dráhy, které poskytují zákaznické slevy IN25 a IN50. Naše aplikace počítá s maximálně deseti současně potvrzovanými zákaznickými slevami. Třetí skupinou jsou speciální akční slevy na určitou komoditu či službu.

Struktura této aplikace je tvořena souborem nesoucím základní parametry aplikace, třemi slevovými soubory s téměř stejnou strukturou zápisu a souborem Sign. Pro každou jednotlivou slevu je specifikován její identifikátor a období platnosti a její pravost je potvrzena podpisem zapisujícího subjektu – aplikuje se podpis řádků.

Při pohledu na strukturu souborů slev vidíme, že v něm má být uloženo vždy X shodných záznamů – jednotlivých slev. Proto si můžeme myslet, že nejvhodnější volbou je cyklicky zaznamenávající typ souboru, ve kterém budou neplatné záznamy postupně nahrazovány záznamy novými. Musíme si však uvědomit, že záznamy jsou zde přepisovány vždy od nejstaršího – nejdříve zapsaného –, zatímco v případě platnosti slev není pravidlem, že expirace dříve zapsané slevy nutně nastane dříve, jak expirace slevy zapsané později. Pokud bychom použily cyklicky se přepisující datové soubory, mohlo by se stát, že bude přepsána ještě platná sleva na „první“ nejstarší pozici místo již neplatné slevy zapsané na „čtvrté“ nejstarší pozici. Z tohoto důvodu slevové soubory v naší aplikaci volíme jako standardní datové soubory. Nová sleva zde bude smět být zapsána pouze v případě, že existuje buď volný záznam – žádná sleva zde nebyla předtím zapsána, případně byla vymazána –, nebo vypršela doba platnosti dříve zapsané slevy – zjistí se z odpovídající položky.

Soubor 0 – BasicParameters

- číslo souboru 0
- název souboru BasicParameters
- typ souboru standardní datový soubor
- popis souboru:

V souboru BasicParameters jsou uloženy základní informace o aplikaci, identifikační číslo jejího vydavatele a jeho transportní sítě a datum vypršení platnosti aktuálně použité verze

aplikace – Tab. 23. Tento soubor spravuje pouze správce aplikace, který při zápisu podepisuje celou aplikaci.

Tab. 23: Aplikace 0xF02354 – Sleva: soubor BasicParameters

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Reserve	Binary	13	-	SA	RFU	
ApPublisher	Binary	24	0	SA	Vydavatel aplikace	šifrovaný
ApPublNetwork	Binary	24	-	SA	ID transportní sítě vydavatele aplikace	
ExpirationDay	DateStamp	14	-	SA	Vypršení platnosti ap.	
Reserve	Binary	26	-	SA	RFU	

Velikost souboru: 112 b = 14 B

Soubor 1 – EmployeeDiscount

- číslo souboru 1
- název souboru EmployeeDiscount
- typ souboru standardní datový soubor
- popis souboru:

Do souboru EmployeeDiscount jsou zapisovány identifikační čísla zaměstnavatelů a období, po které má držitel karty nárok na využívání jejich zaměstnanecké slevy – většinou je to začátek a konec zaměstnaneckého poměru. Pokud má zaměstnanec sjednaný úvazek na dobu neurčitou, je sleva nastavována na platnost dvou let, pokud zaměstnanecký stav stále trvá, držitel si ji nechá pouze prodloužit. Když je zaměstnanecký poměr ukončen, nárok by měl být z aplikace smazán.

Jelikož slevy může zapisovat mnoho subjektů, je třeba, aby byla potvrzena jejich autenticita. Proto každá sleva (řádek) je podepsána svým zapisovatelem. Zapisovatelem může být správce aplikace nebo organizace využívající aplikaci aktivně.

Struktura souboru nesoucího tuto slevu je zobrazena v Tab. 24.

Tab. 24: Aplikace 0xF02354 – Sleva: soubor EmployeeDiscount

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
Employer1ID	Binary	12	0	OVyuzAA SA	ID zaměstnavatele č. 1	šifrrovaný
StartEmp1	DateStamp	14	0	OVyuzAA SA	Začátek zaměstnaneckého poměru č. 1	
EndEmp1	DateStamp	14	0	OVyuzAA SA	Konec zaměstnaneckého poměru č. 1	
SignatureE1	Binary	64	-	SAM	Podpis řádku (zam. č. 1)	
Employer2ID	Binary	12	0	OVyuzAA SA	ID zaměstnavatele č. 2	
StartEmp2	DateStamp	14	0	OVyuzAA SA	Začátek zaměstnaneckého poměru č. 2	
EndEmp2	DateStamp	14	0	OVyuzAA SA	Konec zaměstnaneckého poměru č. 2	
SignatureE2	Binary	64	-	SAM	Podpis řádku (zam. č. 2)	
…	…	…	…	…	…	
Employer5ID	Binary	12	0	OVyuzAA SA	ID zaměstnavatele č. 5	
StartEmp5	DateStamp	14	0	OVyuzAA SA	Začátek zaměstnaneckého poměru č. 5	
EndEmp5	DateStamp	14	0	OVyuzAA SA	Konec zaměstnaneckého poměru č. 5	
SignatureE5	Binary	64	-	SAM	Podpis řádku (zam. č. 5)	
Reserve	Binary	32	-	SA	RFU	

Velikost souboru: 576 b = 72 B

Soubor 2 – CustomerDiscount

- číslo souboru 2
- název souboru CustomerDiscount
- typ souboru standardní datový soubor
- popis souboru:

Ve třetím souboru, CustomerDiscount, může být zaznamenáno v jeden okamžik až deset subjektů (obchodů), u kterých je držitel karty veden jako stálý zákazník a má nárok na zákaznickou slevu – na konkrétní její typ. Je zde zapsáno identifikační číslo daného subjektu, identifikační číslo typu slevy, začátek a konec období nároku na slevu a podpis subjektu, který nárok zaznamenal. V Tab. 25 je ukázána struktura zápisu těchto údajů. Tyto zápisy může provádět správce aplikace nebo organizace využívající aplikaci aktivně.

Tab. 25: Aplikace 0xF02354 – Sleva: soubor CustomerDiscount

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
Vendor1ID	Binary	12	0	OVyuzASA	ID obchodníka č. 1	šifrovaný
TypeOfCustD	Binary	4	1	OVyuzA A SA	Typ zákaznické slevy č. 1	
StartCustD1	DateStamp	14	-	OVyuzASA	Datum začátku platnosti zákaznické slevy č. 1	
EndCustD1	DateStamp	14	-	OVyuzASA	Datum vypršení platnosti zákaznické slevy č. 1	
SignatureV1	Binary	64	-	SAM	Podpis řádku (obch. č. 1)	
Vendor2ID	Binary	12	0	OVyuzASA	ID obchodníka č. 2	
TypeOfCustD	Binary	4	1	OVyuzASA	Typ zákaznické slevy č. 2	

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
StartCustD2	DateStamp	14	-	OVyuzASA	Datum začátku platnosti zákaznické slevy č. 2	šifrovaný
EndCustD2	DateStamp	14	-	OVyuzASA	Datum vypršení platnosti zákaznické slevy č. 2	
SignatureV2	Binary	64	-	SAM	Podpis řádku (obch. č. 1)	
…	…	…	…	…	…	
Vendor10ID	Binary	12	0	OVyuzASA	ID obchodníka č. 10	
TypeOfCustD	Binary	4	1	OVyuzASA	Typ zákaznické slevy č. 10	
StartCustD10	DateStamp	14	-	OVyuzASA	Datum začátku platnosti zákaznické slevy č. 10	
EndCustD10	DateStamp	14	-	OVyuzASA	Datum vypršení platnosti zákaznické slevy č. 10	
SignatureV10	Binary	64	-	SAM	Podpis řádku (obch. č. 10)	
Reserve	Binary	48	-	SA	RFU	

Velikost souboru: 1 152 b = 144 B

Soubor 3 – SpecialDiscount

- číslo souboru 3
- název souboru SpecialDiscount
- typ souboru standardní datový soubor
- popis souboru:

Čtvrtým souborem je SpecialDiscount, kde jsou uloženy speciální jednorázové slevy. Pro každou slevu je zde zapsán její identifikátor, který musí být jedinečný v rámci všech subjektů využívajících tuto aplikaci a období, kdy je tato sleva aktivní – struktura ukázána v Tab. 26. Při zápisu nové slevy zapisující subjekt podepíše celý soubor. Tím získáme rychlejší zpracování transakce na úkor ztráty detailního přehledu, kdo kterou slevu zapsal (ten zde ovšem není tak důležitý). Sleva může být platná buď pro jedno použití, nebo pro neomezený počet použití v období své platnosti. V prvním případě je při jejím využití uživatelem vymazána systémem ze souboru. Zápis údajů v tomto souboru může provádět správce aplikace nebo organizace využívající aplikaci aktivně.

Tab. 26: Aplikace 0xF02354 – Sleva: soubor SpecialDiscount

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
Discount1ID	Binary	24	0	OVyuzAA SA	ID slevy č. 1	šifrovaný
StartD1Validity	DateStamp	14	-	OVyuzAA SA	Začátek platnosti slevy č. 1	
EndD1Validity	DateStamp	14	-	OVyuzAA SA	Datum vypršení platnosti slevy č. 1	
Discount2ID	Binary	24	0	OVyuzAA SA	ID slevy č. 2	
StartD2Validity	DateStamp	14	-	OVyuzAA SA	Začátek platnosti slevy č. 2	
EndD2Validity	DateStamp	14	-	OVyuzAA SA	Datum vypršení platnosti slevy č. 2	
…	…	…	…	…	…	
Discount10ID	Binary	24	0	OVyuzAA SA	ID slevy č. 10	
StartD10Validity	DateStamp	14	-	OVyuzAA SA	Začátek platnosti slevy č. 10	
EndD10Validity	DateStamp	14	-	OVyuzAA SA	Datum vypršení platnosti slevy č. 10	
Signature	Binary	64	-	SAM	Podpis souboru	
Reserve	Binary	24	-	SA	RFU	

Velikost souboru: 632 b = 79 B

Soubor 4 – Sign

- číslo souboru 4
- název souboru Sign
- typ souboru zálohovací soubor
- popis souboru:

Poslední soubor obsahuje základní položky souboru Sign – jelikož se nejedná o aplikaci s tak důležitými údaji jako je „Personalizace karty“, „Elektronická peněženka“ či „Jízdní doklad“, není zde uložena položka blokující celou aplikaci – blokace jednotlivých souborů je dostačující. Tab. 27 zobrazuje strukturu tohoto souboru.

Tab. 27: Aplikace 0xF02354 – Sleva: soubor Sign

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
MasterKeyA	Binary	112	*	SA	Master klíč Aplikace	šifrovaný
BasicParSignA	Binary	112	*	SA	Klíč A souboru 0	
EmployeeDSignA	Binary	112	*	SA	Klíč A souboru 1	
EmployeeDSignB	Binary	112	*	SA	Klíč B souboru 1	
CustomerDSignA	Binary	112	*	SA	Klíč A souboru 2	
CustomerDSignB	Binary	112	*	SA	Klíč B souboru 2	
SpecialDSignA	Binary	112	*	SA	Klíč A souboru 3	
SpecialDSignB	Binary	112	*	SA	Klíč B souboru 3	
SignSignA	Binary	112	*	SA	Klíč A souboru 4	
SignSignB	Binary	112	*	SA	Klíč B souboru 4	
RFUSign	Binary	112	*	SA	Klíč RFU prostoru	
Access	Binary	30	*	SA	Přístupová práva souborů	

Velikost souboru: 1 374 b = 171,75 B

Celková velikost aplikace „Sleva“ je **3 846 b = 480,8 B**.

4.2.2.5 0xF02975 – Věrnostní program

„Věrnostní program“ je aplikace s AID 0xF02975 zaznamenávající počet uživatelem nasbíraných věrnostních bodů v rámci zapojených subjektů – organizací. Nasbírané body uživatel může využít pro získání určitých benefitů specifikovaných konkrétním programem – slevy, dárky apod.

Specifikem tohoto programu je, že všechny zapojené subjekty zapisují věrnostní body do společného uložiště – body od různých organizací mají stejnou hodnotu a sčítají se dohromady, stejně tak mohou být i u všech využity. Navíc až 15 libovolných subjektů, zapojených do poskytování tohoto programu, má právo na vytvoření vlastního věrnostního programu. Do něj zapisuje body pouze vlastnící subjekt – zapisuje si ty stejné body, které zapsal do společného věrnostního programu.

Struktura této aplikace je tvořena čtyřmi soubory, prvním ukládajícím základní informace o aplikaci a programu, druhým hodnotovým zaznamenávajícím aktuální hodnotu věrnostního programu, třetím uchovávajícím věrnostní programy jednotlivých subjektů a čtvrtým souborem Sign.

Soubor 0 – BasicParameters

- číslo souboru 0
- název souboru BasicParameters
- typ souboru standardní datový soubor
- popis souboru:

První soubor BasicParameters nese informace o základních vlastnostech aplikace, kterými jsou identifikátory vydavatele aplikace a jeho transportní sítě a datum vypršení platnosti aktuálně používané verze. Dále zde jsou uložena data pro specifikaci konkrétního věrnostního programu, do kterého je uživatel zapojen – kód typu programu a kód zařazení držitele do programu. Nakonec, kromě položky RFU, je zde ještě uveden datum a čas poslední provedené změny hodnoty konta, což dopomáhá zpětné kontrole zápisu. Názorně se na strukturu souboru BasicParameters můžete podívat v Tab. 28.

Změny v tomto souboru musejí být vždy potvrzeny podpisem aplikace vytvořeným odpovědným subjektem – správce aplikace nebo organizace využívající aplikaci aktivně.

Tab. 28: Aplikace 0xF02975 – Věrnostní program: soubor BasicParameters

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
ApPublisher	Binary	24	0	SA	ID vydavatel aplikace	šifrovaný
ApPublNetwork	Binary	24	-	SA	ID transportní sítě vydavatele aplikace	
ExpirationDay	DateStamp	14	-	SA	Datum expirace VP	
LoyalityType	Binary	8	0	OVyuzAA	Kód typu VP	
LoyalTypeUser	Binary	8	0	OVyuzAA	Kód zařazení do VP	
LastBonus	Binary	24	0	OVyuzAA	Hodnota naposled zapsaných bodů	
LastBonusTime	TimeReal	32	-	SAM	Datum a čas poslední změny bodů VP	
Signature	Binary	64	-	SAM	Podpis souboru	
Reserve	Binary	34	-	SA	RFU	

Velikost souboru: 256 b = 32 B

Soubor 1 – ValueLoyality

- číslo souboru 1
- název souboru ValueLoyality
- typ souboru hodnotový soubor se zálohou
- popis souboru:

Aktuální hodnota konta společného věrnostního programu je uložena v hodnotovém souboru ValueLoyality, jehož struktura je ukázána v Tab. 29. Změny tohoto souboru jsou podepisovány systémovým podpisem SAM modulu provádějícím transakci.

Tab. 29: Aplikace 0xF02975 – Věrnostní program: soubor ValueLoyality

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
ValueLoyality	Binary	24	0	SAM	Aktuální hodnota VP	šifrovaný
SystemSign	Binary	64	-	SAM	Systémový podpis souboru	

Velikost souboru: 88 b = 11 B

Soubor 2 – PartlyLoyality

- číslo souboru 2
- název souboru PartlyLoyality
- typ souboru standardní datový soubor
- popis souboru:

Hodnoty věrnostních programů jednotlivých subjektů jsou ukládány v souboru PartlyLoyality. V jednu chvíli zde může být vedeno až 15 programů, přičemž každý program je zapsán svým identifikačním číslem – respektive číslem subjektu poskytujícím tento program. Dále je u něj zaznamenána aktuální hodnota bodů, datum a čas posledního zápisu a podpisem je potvrzena jeho autenticita. Provádět změny zde může subjekt vedený jako organizace využívající aplikaci aktivně, případně správce aplikace.

Datová struktura souboru PartlyLoyality je zobrazena v následující tabulce Tab. 30.

Tab. 30: Aplikace 0xF02975 – Věrnostní program: soubor PartlyLoyality

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ soubor
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Sign	Binary	4	1	SA	Použitý typ podpisu	
Reserve	Binary	9	-	SA	RFU	
Provider1ID	Binary	12	0	SA OVyuzAA	ID poskytovatele VP1	šifrovaný
LoyalityP1	Binary	24	0	OVyuzAA	Body VP1	
TimeLastBonusP1	TimeReal	32	-	SAM	Datum a čas poslední změny bodů VP1	
SignatureP1	Binary	64	-	SAM	Podpis řádku VP1	
Provider2ID	Binary	12	0	SA OVyuzAA	ID poskytovatele VP2	
LoyalityP2	Binary	24	0	OVyuzAA	Body VP2	
TimeLastBonusP2	TimeReal	32	-	SAM	Datum a čas poslední změny bodů VP2	
SignatureP2	Binary	64	-	SAM	Podpis řádku VP2	
…	…	…	…	…	…	
Provider15ID	Binary	12	0	SA OVyuzAA	ID poskytovatele VP15	
LoyalityP15	Binary	24	0	OVyuzAA	Body VP15	
TimeLastBonusP15	TimeReal	32	-	SAM	Datum a čas poslední změny bodů VP15	
SignatureP15	Binary	64	-	SAM	Podpis řádku VP15	
Reserve	Binary	36	-	SA	RFU	

Velikost souboru: 2 040 b = 255 B

Soubor 3 – Sign

- číslo souboru 3
- název souboru Sign
- typ souboru zálohovací soubor
- popis soubor:

Poslední soubor této aplikace obsahuje základní položky souboru Sign – jelikož se, stejně jako v předchozí aplikaci, nejedná o aplikaci s tak důležitými údaji jako je „Personalizace karty“, „Elektronická peněženka“ či „Jízdní doklad“, není zde uložena položka blokující aplikaci. Struktura tohoto souboru je zobrazena v Tab. 21.

Pro zabezpečení přístupu ke každému souboru jsou použity vždy dva klíče.

Tab. 31: Aplikace 0xF02975 – Věrnostní program: soubor Sign

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
MasterKey	Binary	112	*	SA	Master klíč Aplikace	
BasicParSignA	Binary	112	*	SA	Klíč A souboru 0	
BasicParSignB	Binary	112	*	SA	Klíč B souboru 0	
ValueLoyalSignA	Binary	112	*	SA	Klíč A souboru 1	
ValueLoyalSignB	Binary	112	*	SA	Klíč B souboru 1	
PartlyLoyalSignA	Binary	112	*	SA	Klíč A souboru 2	šifrovaný
PartlyLoyalSignB	Binary	112	*	SA	Klíč B souboru 2	
SignSignA	Binary	112	*	SA	Klíč A souboru 3	
SignSignB	Binary	112	*	SA	Klíč B souboru 3	
RFUSign	Binary	112	*	SA	Klíč RFU prostoru	
Access	Binary	24	*	SA	Přístupová práva souborů	

Velikost souboru: 1 144 b = 143 B

Celková velikost aplikace „Věrnostní program“ je **3 528 b = 441 B**.

4.2.2.6 0xF12686 – Bike sharing

Poslední zde uvedenou je aplikace 0xF12686 “Bike sharing” podporující systém sdílení jízdních kol na určitém území. Tato aplikace obsahuje pouze dva soubory, první nese informace aktuální výpůjčky a druhý je soubor Sign.

Když si uživatel chce vypůjčit kolo, musí být nejprve zaregistrován v systému bike sharingu, kdy je jeho identifikační číslo (z aplikace „Personalizace karty“) zapsáno do databáze systému a na kartě je aktivována odpovídající aplikace. Poté uživatel může kdykoli použít svou kartu pro odemčení bezpečnostního zámku kola – systém uživatele identifikuje podle jeho ID a na kartu zapíše ID zapůjčeného kola společně s časem zapůjčení. Při vrácení kola uživatel opět použije kartu, systém odečte dobu zapůjčení a dopočítá částku, která bude stržena z aplikace „Elektronická peněženka“. Částka se může také odvíjet od nároku na slevu, účasti uživatele ve věrnostním programu či dalších parametrech daných systémem.

Tato aplikace může pracovat jak v online, tak offline systému. V případě práce v offline systému je platba provedena kreditní transakcí.

Soubor 0 – BikeSharing

- číslo souboru 0
- název souboru BikeSharing
- typ souboru standardní datový soubor
- popis souboru:

V souboru BikeSharing jsou uloženy základní parametry aplikace – identifikační číslo vydavatele aplikace a jeho transportní sítě, datum vypršení platnosti nahrané verze aplikace, jedinečné identifikační číslo právě zapůjčeného kola, datum a čas jeho zapůjčení a položka FreeTime. Do položky FreeTime je zapsán čas, po který má uživatel zapůjčení kola zdarma – například prvních patnáct minut. Další údaje, jako například konec vypůjčení, cena, historie výpůjček a podobně, by zde byly nadbytečné. Struktura tohoto souboru je zobrazena v Tab. 32.

Zápis do tohoto souboru provádí správce aplikace a organizace využívající aplikaci aktivně. Vzhledem k typu aplikace zde není zapotřebí podepisovat změny, zabezpečení použitím šifrování dat je dostačující.

Tab. 32: Aplikace 0xF12686 – Bike sharing: soubor BikeSharing

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
Version	Binary	5	0	SA	Verze	nešifr.
State	Binary	2	1	SA	Status souboru	
Encrypt	Binary	4	1	SA	Použitý typ šifrování s.	
Reserve	Binary	13	-	SA	RFU	
ApPublisher	Binary	24	0	SA	Vydavatel aplikace	šifrovaný
ApPublNetwork	Binary	24	-	SA	ID transportní sítě vydavatele aplikace	
ExpirationDay	DateStamp	14	-	SA	Datum expirace aplikace	
BikeID	Binary	20	-	OVyuzAA	ID půjčeného kola	
StartCycling	TimeReal	32	-	SAM	Den a čas zapůjčení k.	
FreeTime	Binary	8	15	OVyuzAA	Čas [min] zapůjčení free	
Reserve	Binary	38	-	SA	RFU	

Velikost souboru: 184 b = 23 B

Soubor 1 – Sign

- číslo souboru 1
- název souboru Sign
- typ souboru zálohovací soubor
- popis souboru:

Poslední soubor má základní strukturu souboru Sign (viz Tab. 33). Přístupové klíče ke každému souboru jsou různé.

Tab. 33: Aplikace 0xF12686 – Bike sharing: soubor Sign

Název	Formát	Velikost [b]	Default [dec]	Typ editace	Popis	Typ souboru
MasterKeyA	Binary	112	*	SA	Master klíč aplikace	
BikeShSignA	Binary	112	*	SA	Klíč A souboru 0	
BikeShSignB	Binary	112	*	SA	Klíč B souboru 0	
SignSignA	Binary	112	*	SA	Klíč A souboru 1	šifrovaný
SignSignB	Binary	112	*	SA	Klíč B souboru 1	
RFUSign	Binary	112	*	SA	Klíč RFU prostoru	
Access	Binary	12	*	SA	Přístupová práva souborů	

Velikost souboru: 684 b = 85,5 B

Celková velikost aplikace „Bike sharing“ je **868 b = 108,5 B**.

4.2.2.7 Shrnutí

V předchozích kapitolách jsme vytvořili strukturu šesti aplikací pro karty MIFARE DESFire. Vzniklé aplikace jsou celkově tvořeny 41 soubory a zabírají prostor 4 922 B, podrobněji v Tab. 34.

Tab. 34: Velikosti aplikací na MIFARE DESFire

Aplikace			
AID	**Název**	**Souborů**	**Velikost [B]**
0xF00041	Personalizace karty	4	449
0xF88342	Elektronická peněženka	6	1 059
0xF12173	Jízdní doklad	20	2 471
0xF02354	Sleva	5	481
0xF02975	Věrnostní program	4	441
0xF12686	Bike sharing	2	109
SUM		**41**	**5 010**

Pokud se máme rozhodnout, která ze dvou vývojových řad MIFARE DESFire je pro navržené aplikace výhodnější, musíme se podívat, které jejich rozdílné znaky jsou v této otázce důležité. Kapitola 2.2.2 ukázala pro nás tři základní rozdíly. Zaprvé, struktura paměti EEPROM umožňuje ve starší verzi vytvoření ‘’pouhých” 32 aplikací, zatímco ve verzi novější není jejich počet omezen. V obou případech jedna aplikace smí obsahovat maximálně 32 souborů. Vzhledem k tomu, že náš zájem se týká pouze šesti aplikací s maximálně 20 soubory, jsou si řady v této oblasti rovny.

Druhým rozdílem je počet klíčů, které mohou být pro aplikaci vytvořeny. Zde umožňuje novější řada použití více samotných klíčů i jejich kombinací, což by podpořilo vyšší zabezpečení dat. Na druhou stranu při samotném návrhu jsme se počtem klíčů vždy vešli do počtu, který pokryjí i možnosti EV1.

Nakonec byl v dané kapitole zmíněn fakt, že MIFARE DESFire EV2 podporuje jako první sdílení souborů mezi aplikacemi. Jelikož naše aplikace nesou data, která jsou využitelná ostatními aplikacemi – sleva pro jízdní doklady, elektronická peněženka pro bike sharing, apod. –, byla by tato novinka jistě využita. Starší řada sdílení souborů nepodporuje, potřeba sdílení dat zde ale může být vyřešena prostřednictvím vhodného systému uvnitř čtecího terminálu. Pokud systém bude oprávněn přistupovat ke všem potřebným aplikacím, získat z nich potřebná data, sám s nimi pracovat a případně je zapsat zpět na kartu, bude dostačujícím, práce s aplikacemi bude také probíhat dle očekávání, možná výpočetně trochu pomaleji.

Na základě těchto poznatků můžeme říci, že pro využití našich aplikací nebude markantním rozdílem, zda budou zapsány na kartách EV1 nebo EV2. Samozřejmě novější verze bude mít v některých ohledech lepší parametry, avšak lze očekávat, že bude také nákladnější. Při výběru konečného řešení se nesmí zapomenout na poměr cena/výkon.

Vzhledem k velikosti paměti EEPROM, která je u karet MIFARE DESFire k dostání ve třech základních velikostech 2, 4 a 8 kB, bychom pro naše aplikace volili tu největší, tedy přesně 8 192 B EEPROM. Poté by nám na kartě zbýval nevyužitý prostor přes 3 kB. Tento prostor by mohl být využit pro případné rozšíření našich aplikací (zvýšení počtu jízdních dokladů, slevových nabídek apod.) nebo nabídnut na využití dalším aplikacím. Dokud nikdo neprojeví o daný prostor zájem, bude veden jako RFU.

V případě okolností vyžadujících použití 4kB karty bychom byli nuceni zmenšit aplikacemi nárokovaný prostor. Toho bychom mohli dosáhnout vypuštěním některých z vytvořených aplikací, kde by v rozhodování o výběru konkrétní aplikace hrálo důležitou roli kromě její velikosti také definování hlavního účelu využití konečného produktu. S ohledem na požadované využití naší karty v odbavovacích systémech, jsou aplikace, které by na kartě

jednoznačně měly zůstat, „Personalizace karty“ a „Jízdní doklad“, zbylé čtyři by teoreticky mohly být vynechány.

Vzhledem k malé velikosti, přestože s odbavovacími systémy nemá mnoho společného, aplikace „Bike sharing“ nemá mít velký vliv na velikost výsledného prostoru a na kartě může být zanechána. Naopak současné vynechání aplikací „Sleva“ a „Věrnostní program“ již zajišťuje požadované zmenšení. Druhou možností, pokud bychom na kartě vyžadovali slevovou a bonusovou aplikaci, je odstranění aplikace „Elektronické peněženky“, v jehož důsledkuby se za jízdní doklady muselo platit v hotovosti nebo jinou kartou. Tato možnost se pro moderní společnost nejeví jako optimální.

Druhou cestou k uspoření nárokovaného prostoru aplikacemi je, namísto odstranění celých aplikací, jejich nezbytné zmenšení provedené tak, abychom je mohli pohodlně nahrát na kartu, a zároveň jsme zachovali jejich funkčnost. Při výběru této možnosti bychom původní varianty aplikací nechtěli takzvaně vyhodit do koše, proto by bylo vhodným přístupem vytvoření nových verzí. Tím bychom získali dvě téměř identické varianty, jednu pro 4kB a druhou pro 8kB kartu.

Rozdíl, kvůli kterému nemohou být naše aplikace nahrány v původní navržené podobě na 4kB kartu, je přes 900 B, což vyžaduje více nezbytných úprav. Uvolnění potřebného prostoru můžeme provést na několika místech. Jsou jimi aplikace, ve kterých je ukládáno více shodných položek z důvodu nabídnutí velké variability uživateli. Například v aplikaci „Jízdní doklad“ je možnost nahrát až pět různých dokladů od každého ze tří různých druhů (jízdenky, zónové a síťové kupony) zároveň. Kdybychom tuto možnost omezili například na tři doklady každého druhu – tedy devět dokladů celkem – získali bychom užitečný prostor. Druhou možností jsou aplikace „Sleva“ a „Věrnostní program“, kde se nabízí omezení počtu jednotlivých slev či poskytovatelů vlastních věrnostních programů. Třetí možností, kde by se teoreticky dalo docílit zmenšení zabíraného prostoru, jsou soubory Sign. Tam by mohla být přehodnocena volba jednotlivých klíčů – změnou typu šifrování by se zmenšily velikosti klíčů –, případně příslušných přístupových práv – použitím méně klíčů a jejich vyšší kombinovatelnosti. V tomto případě bych ale apelovala na skutečně důkladné zhodnocení všech možností, aby nedošlo k nebezpečnému snížení zabezpečení aplikací.

4.2.3 Aplikace na MIFARE Classic

Cílem této kapitoly je jednoduchá demonstrace rozdílů (ne)praktičnosti použití blokové struktury MIFARE Classic oproti flexibilnímu souborovému systému MIFARE DESFire, proto zde nebudou uvedeny stejně detailní návrhy jednotlivých aplikací, jako tomu bylo v předchozí kapitole. Vystačíme si s ukázkami, kolik by vytvořené aplikace využily sektorů, v jakém

rozložení a zda se vůbec všechny najednou dají do pevně dané struktury takzvaně napasovat. Bude zde také ukázán důsledek použití staršího způsobu zabezpečení dat.

4.2.3.1 Proces přetvoření struktury DESFire na Classic

Vzhledem k tomu, že dostupné velikosti pamětí EEPROM karet MIFARE Classic jsou pouze 1kB a 4kB, je jasné, že pro naše účely musíme volit větší z těchto variant, přičemž jsme nuceni zmenšit celkovou velikost aplikací tak, aby se všechny do daného prostoru vešly. K prvnímu „ořezání“ velikosti dojde v důsledku dvou základních rozdílů MIFARE Classic oproti MIFARE EEPROM. Prvním rozdílem je již zmíněná pevně daná struktura paměti, tím druhým je způsob zabezpečení dat.

Struktura sektorů karty je tvořena datovými bloky a právě jedním zavaděčem sektoru, který slouží pro uložení bezpečnostních klíčů a přístupových práv příslušného sektoru. Tyto bloky jsou ekvivalentem souborů Sign v našich aplikacích. Když k tomu připočteme, že je zde místo kryptografických algoritmů DES, 3DES a AES použit jediný algoritmus CRYPTO1 s pevně danou délkou klíčů, zjistíme, že ve struktuře aplikací můžeme vynechat položky Encrypt, určující použitý typ šifrování, a všechny soubory Sign. Protože již od začátku vidíme, že místa na kartě nemáme na zbyt a zároveň víme, že použití pevně dané struktury lehce dojde k ne vždy přesnému zaplnění celého prostoru – data délkově nepasují do bloků –, vynecháme při přenášení také veškeré položky uchovávajících RFU prostor, které pro funkci aplikací nejsou důležité.

Po takto provedeném zmenšení celkové velikosti aplikací – vynechání položek Encrypt, Reserve a souborů Sign –, se dostáváme na velikost 3 080 B, což je na první pohled pro naše účely více než dostačující. Musíme si ale uvědomit, že toto je velikost dat, která smíme zapsat, pouze do datových bloků, a že velikosti využitých zavaděčů sektoru budeme muset teprve připočítat – jejich počet bude roven počtu využitých sektorů.

S převodem dat začneme tak, že ke každému vytvořenému souboru na kartě DESFire (kromě souboru Sign) dopočítáme jeho velikost po odebrání položek Encrypt a Reserve, počítáme v bytech. Získanou hodnotu vydělíme číslem 16 a výsledek zaokrouhlíme nahoru. Takto získáme počet bloků potřebných pro zápis dat z odpovídajícího souboru – 16 reprezentuje velikost jednoho bloku. Po zjištění, kolik bloků který soubor (aplikace) potřebuje, začíná ta náročnější část, rozdělení bloků do jednotlivých sektorů aplikace tak, aby bylo zajištěno optimální rozložení přístupových práv.

V tomto okamžiku je dobré si připomenout následující fakta. Aplikace převádíme na 4kB MIFARE Classic kartu, u které víme, že obsahuje 32 4blokových a osm 16blokových sektorů. Navíc jeden 4blokový sektor poskytuje pouze dva místo tří datových bloků nebo v případě

podpory MAD adresářů jsou celé dva sektory vyhrazeny pro uložení AID aplikací. Rozdíl mezi 4blokovými a 16blokovými sektory je kromě velikosti také v možnosti nastavení přístupových práv. Jelikož oba typy obsahují shodně pouze jeden zavaděč sektoru, ve kterém jsou uloženy maximálně dva šifrovací klíče a tři přístupové byty, je možné nastavit maximálně čtyři úrovně zabezpečení. 4blokový sektor má tedy zaručenou možnost nastavení různých přístupových práv pro každý svůj blok, kdežto práva přístupu v 16blokovém sektoru jsou nastavena pro tři pětice bloků a zavaděč sektoru zvlášť (viz kapitola 2.2.1.1).

To, že většina aplikací je velikostně větší než jednotlivé sektory, neznamená veliký problém. Použijeme pro jejich zápis více sektorů, přičemž můžeme pro jednu aplikaci použít kombinaci obou typů sektorů. Při rozdělování jednotlivých bloků nesmíme zapomenout na to, že do jednoho sektoru uložíme maximálně 3, v případě většího sektoru 15 bloků – jeden blok je vyhrazen zavaděči sektoru. Zároveň se snažíme o to, aby bloky jednoho souboru byly v sektorech uloženy pohromadě, přičemž uložení bloků různých souborů do jednoho sektoru nemusí být problém, pokud správně rozdělíme přístupová práva. Na co bychom měli myslet je to, že hodnotové bloky by měly mít vzhledem ke své specifičnosti vlastní přístupová práva. Proto není vhodné je ukládat do 16blokových sektorů, kde by musely sdílet práva s dalšími čtyřmi bloky – pokud by tam nebylo uloženo více hodnotových bloků se stejnými přístupovými právy.

Celý proces přiřazování je v podstatě o tom, že se vytvářejí různé varianty aplikací tak, aby vyhovovaly jednotlivým požadavkům, a poté jsou hledány jejich vhodné kombinace tak, aby se všechny aplikace na kartu vešly – aby nedošlo k překročení parametrů karty, tedy použití více jak 32 4blokových a osmi 16blokových sektorů. Když najdeme optimální kombinaci, spočítáme počet sektorů aplikace, který nám udává počet zavaděčů sektoru, dopočítáme velikost prostoru, který zabírají, a přičteme ji k velikosti aplikace. Získáme tak velikost dat zapsaných v aplikaci – uvědomme si, že v sektorech nemusí být data zapsaná ve všech blocích, přesto i bloky prázdné jsou součástí aplikace, v podstatě uchovávají RFU prostor. Skutečná velikost aplikace je přesně daná počtem a typem sektorů nesoucí její data.

4.2.3.2 Přenesení aplikací na strukturu MIFARE Classic

Během „přenášení“ aplikací vytvořených v kapitole 4.1.2 ze struktury DESFire na strukturu Classic se získalo konečných 240 bloků, které byly rozřazeny do 30 4blokových sektorů a osmi 16blokových sektorů.

Rozložení všech původně získaných 231 datových bloků do jednotlivých sektorů dle potřeb aplikací se ukázalo neproveditelné – nebyla nalezena vhodná kombinace struktur všech aplikací tak, aby mohly být současně zapsány na kartu. Ukázalo se, že datových bloků je příliš

mnoho. Aby se dosáhlo optimálního rozložení, bylo zapotřebí provést změny velikostí některých aplikací a snížit tak počet bloků. Nejzásadnější změna byla provedena v aplikaci „Jízdní doklad“, protože jednotlivé jízdní doklady vyžadovaly příliš mnoho prostoru tak, že nebylo možné je vhodně zkombinovat. Proto byl snížen počet dostupných dokladů typu jízdenka, kupon zónový a kupon síťový. Z původních pěti dokladů pro každý typ byly ponechány pro jízdenku a síťový kupon čtyři doklady a pro zónový kupon pouze tři, což výrazně snížilo počet potřebných bloků. Dále, ve stejné aplikaci, bylo pro původní soubor AddService vytvořeno místo 16 pouze 15 bloků, díky čemuž mohl být celý soubor uložen do jednoho 16blokového sektoru. Pro docílení tohoto stačilo zmenšit soubor o necelých šest bytů, čehož může být lehce dosaženo například snížením počtu položek Animal nebo Luggage o pouhé dvě. Totéž bylo potřeba provést v aplikaci „Věrnostní program“ s původním souborem PartLoyality. Zmenšením souboru o necelých devět bytů – stačilo změnit počet subjektů poskytující vlastní věrnostní program z 15 na 14 – vzniklo 15 bloků, které byly uloženy do pěti 4blokových sektorů. Ani jedna z provedených změn nemá zásadní vliv na danou aplikaci.

Výsledné rozdělení aplikací do jednotlivých bloků a posléze i sektorů je názorně uvedeno v Tab. 35. Tato tabulka zjednodušeně ukazuje jak původní DESFire strukturu každé aplikace, tak její strukturu na MIFARE Classic. Aplikace je uvedena svým názvem a jedinečným AID pro obě struktury. Dále pak jsou vypsány všechny její soubory společně s velikostmi v bytech. Vedle této hodnoty je uvedena přepočítaná velikost bez položek Encrypt a Reserve. V případě souboru Sign je uvedena již velikost vypočítaná na základě počtu přidělených bloků. V posledních dvou sloupcích je uveden počet přiřazených bloků každému souboru – v levém sloupci jsou bloky přiřazené do 4blokového sektoru, v pravém ty přiřazené 16blokovému sektoru, navíc v závorce je uvedeno označení sektoru, do kterého bloky patří, toto označení je pouze orientační v rámci dané aplikace. Nakonec je u každé aplikace uveden počet využitých sektorů.

Tab. 35: Rozložení aplikací na MIFARE Classic a MIFARE DESFire

Aplikace					
MIFARE DESFire - soubory			MIFARE Classic – bloky		
č.soub.	**Název**	**Velikost [B]**	**4blokový sektor**	**16blokový sek.**	
Personalizace karty					
0xF00041			**0x0004**		
0	CardPublisherInfo	201,0	130,9	9 (s1, s2, s3)	
1	CardholderInfo	90,0	63,8	4 (s4, s5)	
2	CardHolderDiscount	27,0	17,5	2 (s5)	
3	Sign	131,1	80,0	5 (s1–5)	
SUM		**449,1**	**292,1**	**5 sektorů**	**0 sektorů**
Elektronická peněženka					
0xF88342			**0x8834**		
0	BasicParameters	21,0	14,5		1 (s1)
1	ExtendedParameters	30,0	24,9		2 (s1)
2	ValueEP	11,0	11,0	1 (s1)	
3	CreditLog	360,0	352,6	23 (s1–8)	
4	DebetLog	358,0	351,4		22 (s1, s2)
5	Sign	279,3	160,0	8 (s1–8)	2 (s1, s2)
SUM		**1 059,3**	**914,4**	**8 sektorů**	**2 sektory**
Jízdní doklad					
0xF12173			**0x1217**		
0	BasicParameters	14,0	13,9	1 (s1)	
1-5	OriginDestinationTicket	625,0	455,5		29 (s1, s2)
6-10	Coupons_Zones	650,0	355,1	3 (s3)	20 (s3, s4)
11-15	Coupons_Network	555,0	395,0		25 (s4, s5)
16	PrepaidKms	52,0	50,4	4 (s1, s2)	
17	ValuePrepaidKms	10,0	10,0	1 (s1)	
18	AddService	257,0	245,9		15 (s6)
19	Sign	307.6	144,0	3 (s1–3)	6 (s1–6)
SUM		**2 470,6**	**1669,8**	**3 sektory**	**6 sektorů**
Sleva					
0xF02354			**0x0235**		
0	BasicParameters	14,0	8,6	1 (s1)	
1	EmployeeDiscount	72,0	66,4	5 (s1, s2)	
2	CustomerDiscount	144,0	136,4	9 (s3-5)	
3	SpecialDiscount	79,0	74,4	5 (s6, s7)	
4	Sign	171,8	112,0	7 (s1–7)	
SUM		**480,8**	**397,8**	**7 sektorů**	**0 sektorů**
Věrnostní program					
0xF02975			**0x0297**		
0	BasicParameters	32,0	26,1	2 (s1)	
1	ValueLoyality	11,0	11,0	1 (s1)	
2	PartlyLoyality	255,0	248,9	15 (s2-6)	
3	Sign	143,0	96,0	6 s(1–6)	
SUM		**441,0**	**382,0**	**6 sektorů**	**0 sektorů**
Bike sharing					
0xF12686			**0x1268**		
0	BikeSharing	23,0	16,1	2 (s1)	
1	Sign	85,5	16,0	1 (s1)	
SUM		**108,5**	**32,1**	**1 sektor**	**0 sektorů**
SUM		**5009,3**	**3688,1**	**30 sektorů**	**8 sektorů**

4.2.3.3 Shrnutí

Po zjištění výsledného počtu a typu potřebných sektorů pro jednotlivé aplikace – 30 4blokových a osm 16blokových sektorů – bylo možné dopočíst skutečnou velikost paměti, kterou na kartě zabírají. Námi vytvořené aplikace zabírají na kartě 3 968 B z celkových 4 096 B, přičemž tam zbývají nevyužity pouhé dva sektory, v případě použití adresářů MAD2 je karta plně využita. Velikosti všech aplikací jsou uvedeny v Tab. 36. Tabulky Tab. 35 a Tab. 36 potvrzují to, že využitím blokové struktury paměti aplikace zabírá více prostoru, než skutečně pro uložená data potřebuje (3 968 B oproti 3 688 B). Pravdou sice je, že se nejedná o markantní rozdíl, který navíc může být použit pro RFU prostor, jde ale o to, jaká je struktura tohoto prostoru, zda je tvořen jednotlivými byty náhodně „roztroušenými v aplikaci“, což by bylo velmi obtížně využitelné, nebo jejich většími shluky, pro které by se v budoucnu našlo vhodné uplatnění.

Tab. 36: Velikosti aplikací na MIFARE Classic

Aplikace				
AID	Název	Sektorů		Velikost [B]
		4blokový	16blokový	
0x0004	Personalizace karty	5	0	320
0x8834	Elektronická peněženka	8	2	1024
0x1217	Jízdní doklad	3	6	1728
0x0235	Sleva	7	0	448
0x0297	Věrnostní program	6	0	384
0x1268	Bike sharing	1	0	64
SUM		**30**	**8**	**3968**

Získaný výsledek se sice na první pohled může zdát uspokojivý – aplikace se nám na kartu v podstatě přesně vešly, stačilo upravit pouze pár parametrů. Musíme si však uvědomit, že jsme „přenos“ řešili pouze na úrovni celých souborů, nikoli na úrovni zapsaných dat (položek). Při tomto přístupu bylo jednoduché říci, že soubor takto jde rozdělit a přidělená přístupová práva jsou dostačující, ale takto jednoduše se k tomu přistupovat nedá. Skutečný přenos musí být řešen na úrovni jednotlivých položek! Jedině tak budeme schopni posoudit, jak přesně se mají – a hlavně mohou – soubory rozdělit. Důkazem toho jsou například jízdní doklady, které mají velikosti v průměru 7,5 B, což není moc přívětivá velikost v případech, když chceme od sebe jednotlivé doklady oddělit. V případě menšího sektoru se musí dělit na mnoho částí, v případě sektoru většího musí sdílet s někým část přístupových práv nebo vytváří nevyužité byty – zaplní jeden a půl pětice se shodnými přístupovými právy – půl pětice je nevyužito.

Zároveň z takto nepodrobného pohledu nemáme jasnou představu o tom, jak jsou položky zabezpečeny. Tedy, popravdě řečeno, my to víme – v teoretické části to bylo mnohokrát

zmíněno. Data uchovávaná na kartách MIFARE Classic v podstatě nejsou vůbec chráněna, jelikož je zde používán pouze kryptografický algoritmus CRYPTO1, který byl již před několika lety prolomen a neposkytuje uživatelům v podstatě žádnou skutečnou ochranu. Proto v této úloze nemá cenu podrobněji řešit jednotlivé možnosti rozložení práv, klíčů a tvorbu podpisů ve struktuře.

V podstatě všechny vytvořené aplikace pracují s daty, která by mohla být v důsledku zneužita pro neoprávněné obohacení držitele karty, subjektu akceptujícího či využívajícího její služby, nebo třetí strany – nahrání na kartu neoprávněný nárok na slevu, bonusové body, falešné jízdní doklady, změny údajů elektronické peněženky, apod. Proto pro jejich uložení nejsou vhodné karty MIFARE Classic, které nezajišťují dostatečné zabezpečení.

5 Závěr

V teoretické části této práce byl čtenář seznámen se základními principy datové komunikace probíhající mezi bezkontaktními čipovými kartami MIFARE a odbavovacím terminálem. Byla zde popsána struktura paměti karty a pravidla ukládání dat v ní. Byl vysvětlen princip vyhledávání a zabezpečení dat uložených na kartě, přenášených během komunikace i právě zapisovaných do paměti. Na závěr byl zmíněn vliv tohoto zabezpečení na rychlost provádění transakcí, která je důležitým parametrem většiny komerčních aplikací.

Praktická část práce se zabývala optimalizací komunikace karta–terminál za účelem zvýšení efektivity procesů odbavovacích systémů v dopravě. Optimalizace byla prováděna na základě vhodného uspořádání datové struktury nahraných aplikací a nastavení odpovídající úrovně zabezpečení dle citlivosti uložených dat vzhledem k nejlepšímu poměru zabezpečení/rychlost.

Na začátku byla vyzdvihnuta důležitost přípravné fáze, kdy řešitel musí předem co nejpodrobněji a nejpřesněji stanovit a utřídit požadované funkce každé aplikace, jejich vstupní a výstupní data i role jednotlivých subjektů vstupujících do procesu. Právě na kvalitě provedení tohoto kroku často závisí úspěch celého snažení, jelikož právě na základě těchto údajů je tvořena výsledná datová struktura. Poté proběhl samotný proces návrhu jednotlivých aplikací a to jak pro strukturu nejnovější řady MIFARE DESFire, tak pro možnosti využití na struktuře první řady MIFARE Classic.

Ve výsledku bylo navrženo šest samostatných aplikací, konkrétně se jednalo o aplikace Personalizace karty, Elektronická peněženka, Jízdní doklad, Sleva, Věrnostní program a Bike sharing. Při návrhu byly specifikovány jejich jednotlivé funkce, které mnohokrát nemají význam pouze pro aplikaci samotnou, ale podporují také procesy ostatních aplikací uložených na kartě – platba jízdního dokladu elektronickou peněženkou, ověření nároku na slevu při platbě. Poté byla navržena taková struktura dat a jejich způsob zabezpečení, aby výsledné aplikace měly co možná nejširší záběr využití při schopnosti poskytovat kvalitní služby, které jsou schopny splnit různé kombinace přání uživatele karty.

V průběhu zpracování této práce vyplynulo, že karty MIFARE Classic využívají pro uchovávání dat pevně danou strukturu 1kB nebo 4kB paměti EEPROM dělené do sektorů tvořených 16bytovými bloky s přesně specifikovanými možnostmi zabezpečení. V důsledku této struktury je velmi obtížné vytvořit skutečně efektivní aplikaci, která plní požadované funkce s využitím pouze nezbytně nutného datového prostředí a má zajištěno zabezpečení adekvátní jejímu využití. Je tomu tak proto, že aby mohla být aplikace optimalizována, je potřeba přizpůsobit prostor a zabezpečení jejím požadavkům a ne naopak, aby se aplikace přizpůsobovala striktně definovanému prostředí.

V případě, že by byl pevně stanoven požadavek na efektivní využití prostoru MIFARE Classic aplikacemi, muselo by se společně s pečlivým stanovením veškerých datových položek řešit také jejich přesné umístění v rámci sektoru tak, aby byly uloženy v bloku společně s dalšími položkami stejných nároků na zabezpečení, a zároveň aby jednotlivé bloky byly vyplněny co nejefektivněji – obsahovaly minimum prázdných bytů. Navíc by měly být vyplněny nejlépe všechny bloky sektoru, aby nezbylo příliš nevyužitého prostoru, jelikož bloky jednoho sektoru mohou být použity pouze pro jednu aplikaci. Z těchto požadavků je evidentní, že optimalizace dat na MIFARE Classic vyžaduje velice přesné počítání velikostí jednotlivých položek a následné jejich vzájemné kombinování pro nalezení vhodného uspořádání.

Druhým, ještě více přesvědčujícím argumentem proti snaze optimalizovat datovou komunikaci karet MIFARE Classic je celý jejich zabezpečovací mechanismus, který stojí a zároveň okamžitě padá na již několik let prolomeném šifrovacím standardu CRYPTO1. Ten tím pádem neplní svou funkci a veškerá data, se kterými tyto karty pracují, jsou v podstatě nezabezpečena. Optimalizace nezabezpečené datové komunikace v dnešní době nemá téměř smysl. Jediné vhodné využití těchto bezkontaktních čipových karet by bylo fungování jako obyčejný nezabezpečený datový nosič, jehož výhodou by mohla být relativně vyšší přenosová rychlost zápisu a čtení dat v případě, že se rozhodne vůbec nezapojovat již prolomené šifrovací mechanizmy.

V případě karet MIFARE DESFire je vidět značný pokrok od dob vytvoření první řady a snaha co nejvíce přizpůsobit vlastnosti karet efektivnímu využití a požadavkům trhu. Nahrazení blokové struktury flexibilním systémem souborů umožňuje vytvoření vlastních aplikací přesně dle definovaných požadavků na prostor, přičemž nevyužité byty zde vznikají pouze jako předem reservovaný prostor pro účely využití v budoucnu – zavedení nových funkcí, parametrů či služeb. Nejen nová struktura, ale také modernější, prozatím dostatečně bezpečné kryptografické algoritmy 3DES a AES jsou tím hlavním mechanismem podporujícím vhodné zabezpečení dat na těchto kartách. Na základě zvoleného algoritmu jsou pro aplikaci vytvořeny kryptografické klíče, kterými jsou jeden master klíč aplikace a až 14 subklíčů master klíče karty pro řadu MIFARE DESFire EV1. V případě nejnovější řady MIFARE DESFire EV2 může být vytvořeno až 16 setů klíčů pro aplikaci. Poté lze libovolnou kombinaci klíčů aplikace použít pro zabezpečení kteréhokoli jejího souboru. Tyto bezpečnostní mechanismy společně s použitím principů CRC bytů a digitálních podpisů jsou nastaveny tak, aby zajistily požadovanou úroveň zabezpečení dat uložených v aplikaci.

Výsledkem takovýchto vlastností bezkontaktních čipových karet, respektive jejich EEPROM, je vysoká podpora optimalizace procesu datové komunikace mezi nimi a terminálem

za účelem dosažení požadovaných funkcí definované úrovně zabezpečení při minimalizaci využitého datového prostoru a doby zpracování.

Poznatky vycházející z této práce jasně ukazují, že doba používání technologie MIFARE Classic pro účely aplikací pracujících s citlivými daty je již nějaký čas za námi, navíc její parametry nejsou vhodně přizpůsobeny pro optimalizaci datové komunikace. Proto se trh, stejně jako její výrobce, rychle přeorientovává na jejího mladšího a mnohem flexibilnějšího nástupce, řadu MIFARE DESFire.

Tato práce může být základem vytvoření skutečných aplikací implementovaných na bezkontaktní čipové karty MIFARE DESFire 8kB zavedených do reálného provozu odbavovacích systémů či jiných odvětví. Jednotlivé aplikace mohou být zavedeny do provozu ve formátu zde navrženém, ale nebrání se ani jednotlivým úpravám dle přesných požadavků systému, ve kterém budou operovat.

Věřím, že zde vytvořené struktury jednotlivých aplikací budou inspirací či dobrým pomocníkem dalším studentům a vědcům při návrhu vlastních aplikací v prostředí technologií bezkontaktních čipových karet.

Použitá literatura a internetové zdroje

[1] Borka, J.: *"Bezkontaktní technologie v odbavovacích systémech."* Vědeckotechnický sborník ČD [online]. 2012. č. 34. [cit. 21. 9. 2015]. Dostupné z http://vtsb.cd.cz/VTS/CLANKY/vts34/3401.pdf

[2] Vlčková, V.: *Kudy tudy systémovým inženýrstvím*. Praha: České vysoké učení technické v Praze, Fakulta dopravní, 2010. 105 s. ISBN 978-80-01-05447-5

[3] MIFARE [online]. [cit. 25. 9. 2015]. Dostupné z: https://www.mifare.net/en/

[4] ISO [online]: *ISO/IEC 7810: 2003. Identification cards – Physical characteristics*, Abstract. [cit. 26. 9. 2015]. Dostupné z: http://www.iso.org/iso/catalogue_detail?csnumber=31432

[5] NXP Semiconductors: AN10833, *MIFARE Type Identification Procedure*. Rev. 3.5 – 27. března 2014. 14s. Dostupné z:

[6] OPEN PCD [online*]: ISO/IEC 14443: Identification cards – Contactless integrated circuit(s) cards – Proximity cards*, Part 1 – 4. [cit. 26. 9. 2015]. Dostupné z: http://www.openpcd.org/ISO14443#Mifare_Classichttp://www.nxp.com/documents/application_note/AN10833.pdf

[7] NXP Semiconductors: AN10834, *MIFARE ISO/IEC 14443 PICC Selection*. Rev. 3.0 – 26. června 2009. 17s. Dostupné z: http://www.nxp.com/documents/application_note/130830.pdf

[8] NXP Semiconductors: AN10927, *MIFARE and handling of UIDs*. Rev. 3.1 – 2. října 2013. 21s. Dostupné z: http://www.nxp.com/documents/application_note/AN10927.pdf

[9] NXP Semiconductors: *MF1S50yyX, MIFARE Classic 1K - Mainstream contactless smart card IC for fast and easy solution development*. Rev. 3.0 – 2. května 2011. 39s. Dostupné z: http://www.nxp.com/documents/data_sheet/MF1S50YYX.pdf

[10] NXP Semiconductors: *MF3ICDx21_41_81, MIFARE DESFire EV1 contactless multi-application IC*. Rev. 3.1 – 21. prosince 2010. 18s. Dostupné z: http://www.nxp.com/documents/short_data_sheet/MF3ICDX21_41_81_SDS.pdf

[11] ISO [online]: *ISO/IEC 7816-4: 2003. Identification cards – Integrated circuits cards – Part 4: Organization, security and commands for interchange,* Abstract. [cit. 9. 10. 2015]. Dostupné z: http://www.iso.org/iso/catalogue_detail.htm?csnumber=54550

[12] NXP Semiconductors: AN10787, *MIFARE Application Directory (MAD)*. Rev. 7.1 – 16. ledna 2013. 24s. Dostupné z: http://www.nxp.com/documents/application_note/AN10787.pdf

[13] Garcia, F. D., G. de Koning Gans; R. Muijrers; P. van Rossum, R. Verdult; R. W. Schreur; B. Jacobs: *Dismantling MIFARE Classic.* [online]. 13th European Symposium on Research in Computer Security. 2008. LNCS, Springer. Str. 98-114. [cit. 8.10.2015]. Dostupné z: http://www.cs.ru.nl/~flaviog/publications/Dismantling.Mifare.pdf

[14] Hamdan. O. Alanazi, B. B. Zaidan, A. A. Zaidan, Hamid A. Jalab, M. Shabbir a Y. Al-Nabhani: *New Comparative Study Between DES, 3DES and AES within Nine Factors*. Journal Of Computing, Volume 2, ISSUE 3, pp. 152-157, MARCH 2010.

[15] Oswald, D. & Ch. Paar: *Breaking MIFARE DESFire MF31CD40: Power Analysis and Templates in the Real World. – Extended Version.* [online]. Německo. Ruth – University Bochum. Horst Görtz Institute for IT Security. 2011. 19s. [cit. 13. 10. 2015]. Dostupné z: http://www.iritech.com/index.html

[16] Majot, A., & R. Yampolskiy,: *Global catastrophic risk and security implications of quantum computers.* [online]. Futures. 2015. 10s. [cit. 8.10.2015]. Dostupné z: http://dx.doi.org/10.1016/j.futures.2015.02.006

[17] Holý, R. - Scherks, J. - Kalika, M.: *Application of ID cards - security components In: 6th International Conference on Signal Processing and Communication Systems.* New Jersey: IEEE, 2012, p. 1-250. ISBN 978-1-4673-2391-8.

[18] NXP Semiconductors: *Introduction MIFARE DESFire EV2, Q3 2014.* 2014. [cit. 25.11.2015]. Dostupné z: http://elearning.nxp.com/pluginfile.php/13413/mod_resource/content/5/MIFARE%20DESFire%20EV2%20product%20launch%20presentation%20general%20V2%209.pdf

[19] ČSN SO/IEC 7812, *Identifikační karty – Identifikace vydavatelů karet.* Praha: Český normalizační institut, 2002.

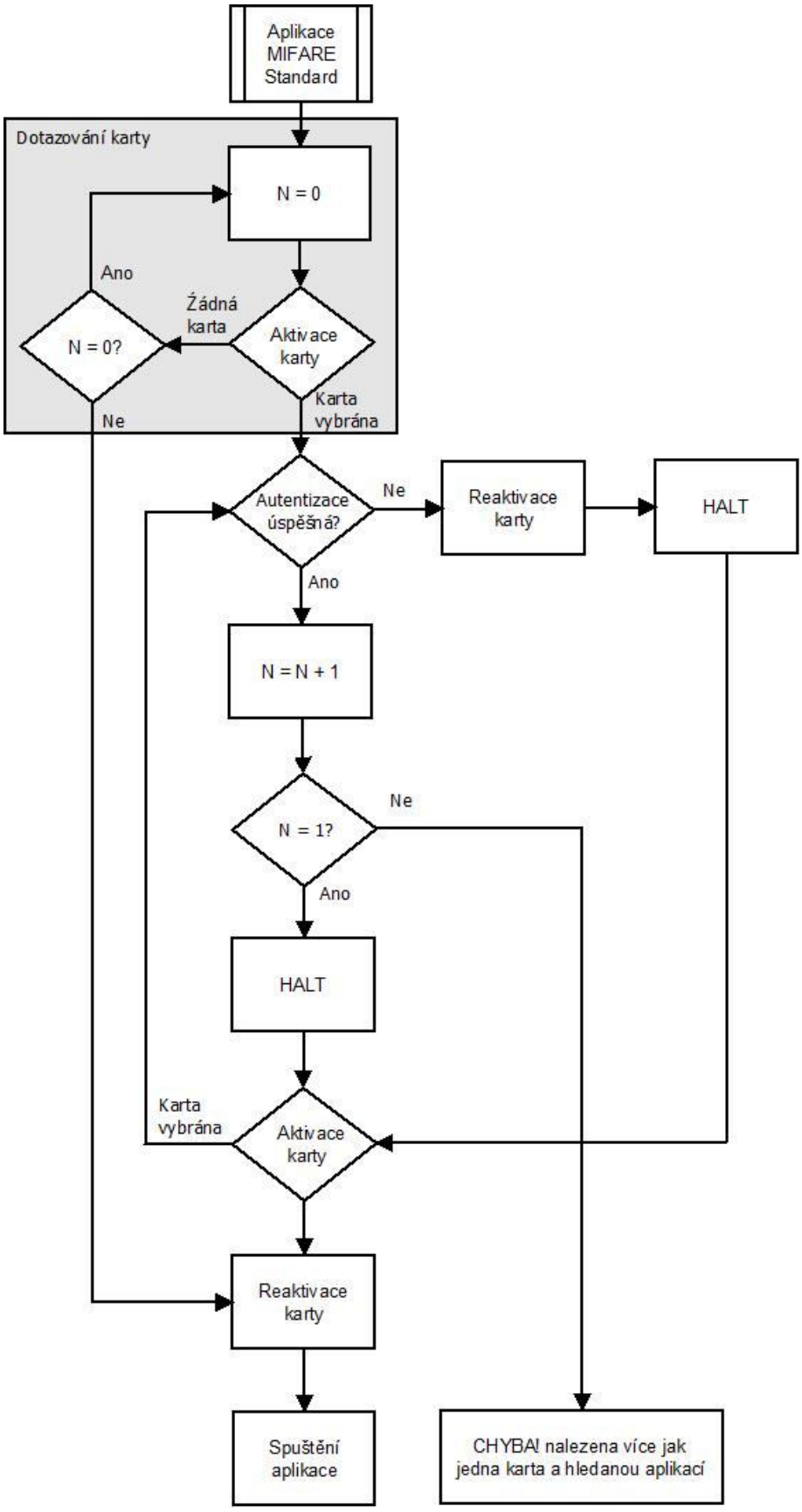
Seznam obrázků

Seznam tabulek

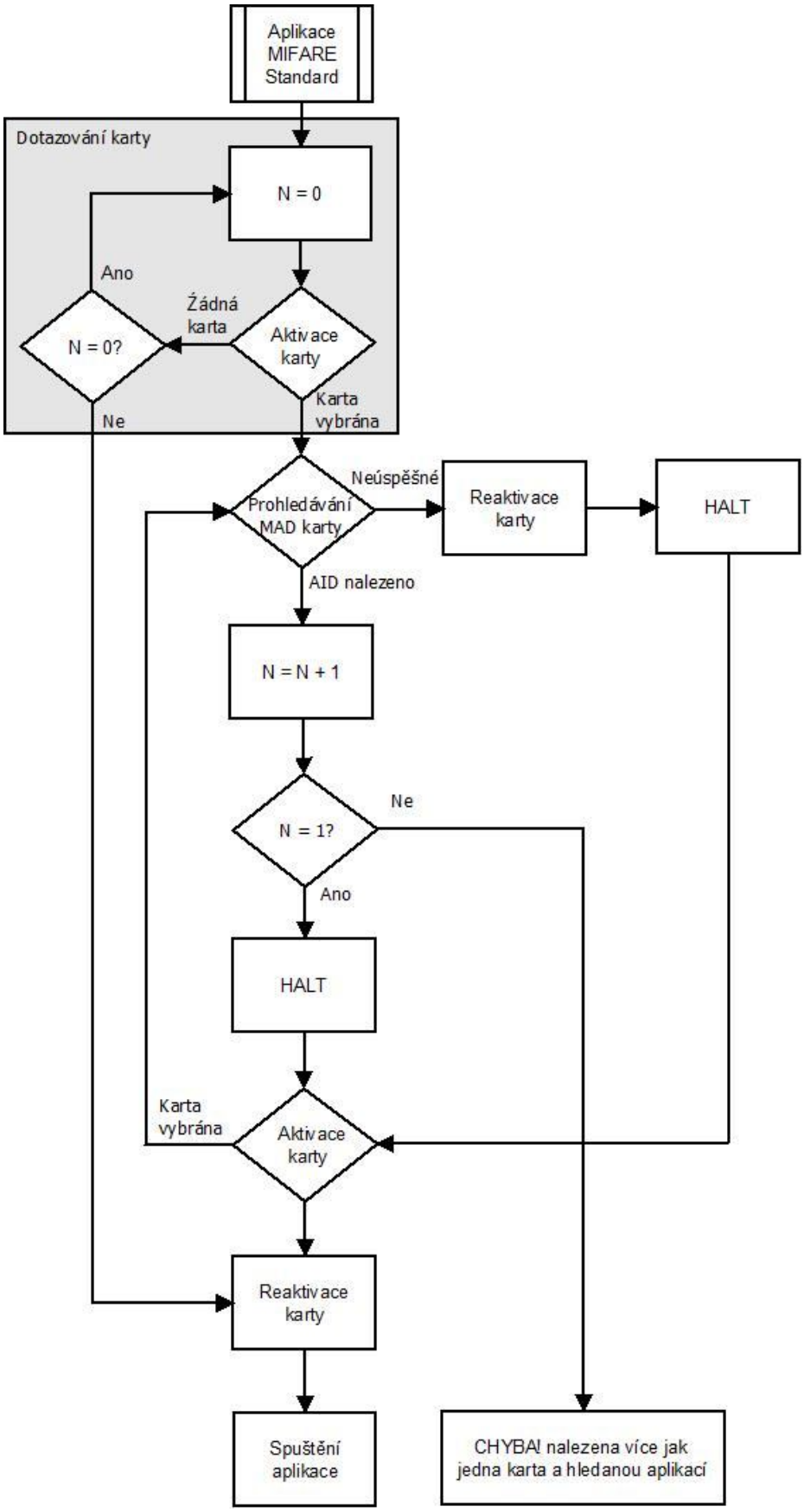
Seznam příloh

příloha A

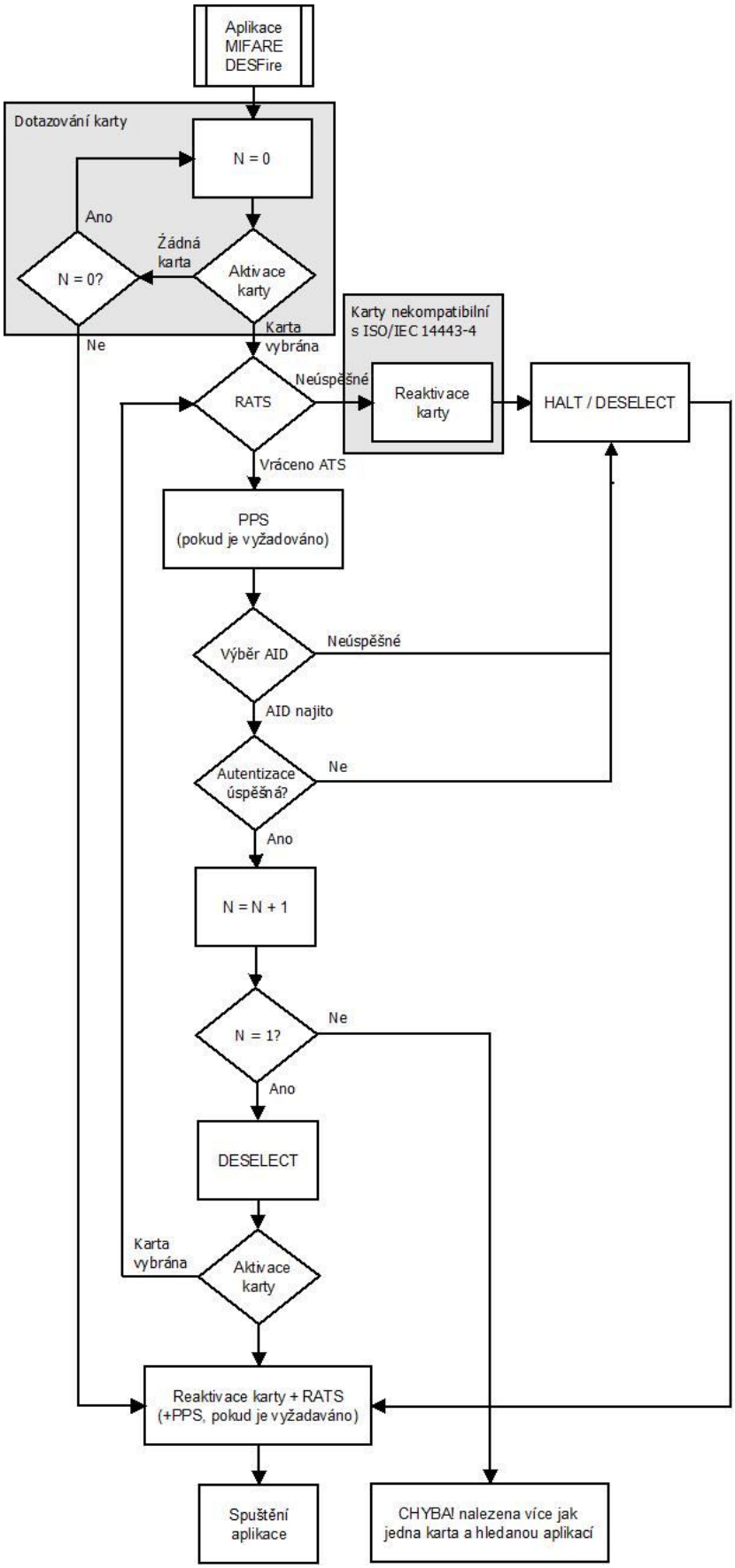
Proces výběru karty MIFARE



A 1: Proces výběru jedné MIFARE Classic karty obsahující určitou aplikaci bez integrovaného adresáře MAD [7]



A 2: Proces výběru jedné MIFARE Classic karty obsahující určitou aplikaci s využitím adresáře MAD [7]



A 3: Proces výběru jedné MIFARE DESFire karty obsahující určitou aplikaci [7]

příloha B

Struktura paměti MIFARE Classic

		Číslo bytu uvnitř bloku															
Sektor	Blok	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	popis
15	3	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 15
	2																	Data
	1																	Data
	0																	Data
14	3	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 14
	2																	Data
	1																	Data
	0																	Data
.																		.
.																		.
.																		.
.																		.
.																		.
1	3	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 1
	2																	Data
	1																	Data
	0																	Data
0	3	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 0
MAD	2																	Data
sektor	1																	Data
0x00	0																	Blok výrobce

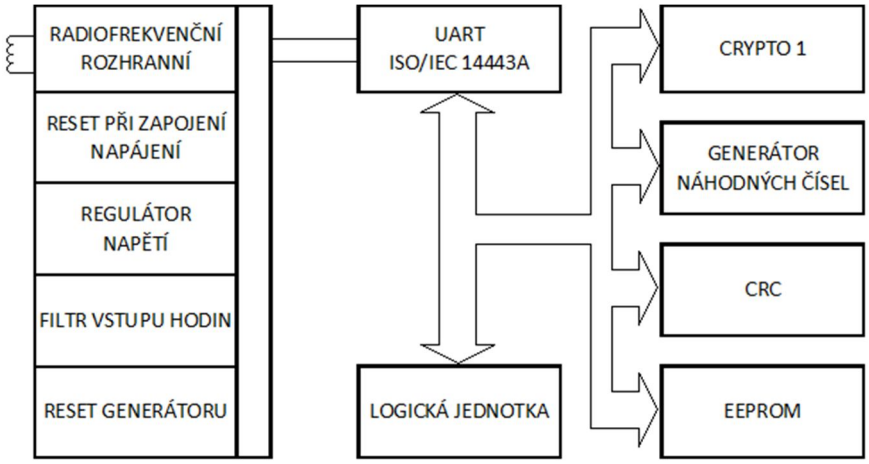
B 1: Struktura paměti EEPROM karty MIFARE Classic 1K [9]

		Číslo bytu uvnitř bloku															
Sektor	Blok	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	popis
39	15	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 39
	14																	Data
	13																	Data
	.																	.
	.																	.
	.																	.
	2																	Data
	1																	Data
	0																	Data
.																		.
.																		.
.																		.
.																		.
32	15	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 32
	14																	Data
	13																	Data
	.																	.
	.																	.
	.																	.
	2																	Data
	1																	Data
	0																	Data
31	3	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 31
	2																	Data
	1																	Data
	0																	Data
.																		.
.																		.
.																		.
.																		.
.																		.
16	3	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 16
MAD	2																	Data
sektor	1																	Data
0x01	0																	Data
.																		.
.																		.
.																		.
.																		.
.																		.
0	3	Klíč A						Přístupové b				Klíč B						Zavaděč sektoru 0
MAD	2																	Data
sektor	1																	Data
0x00	0																	Blok výrobce

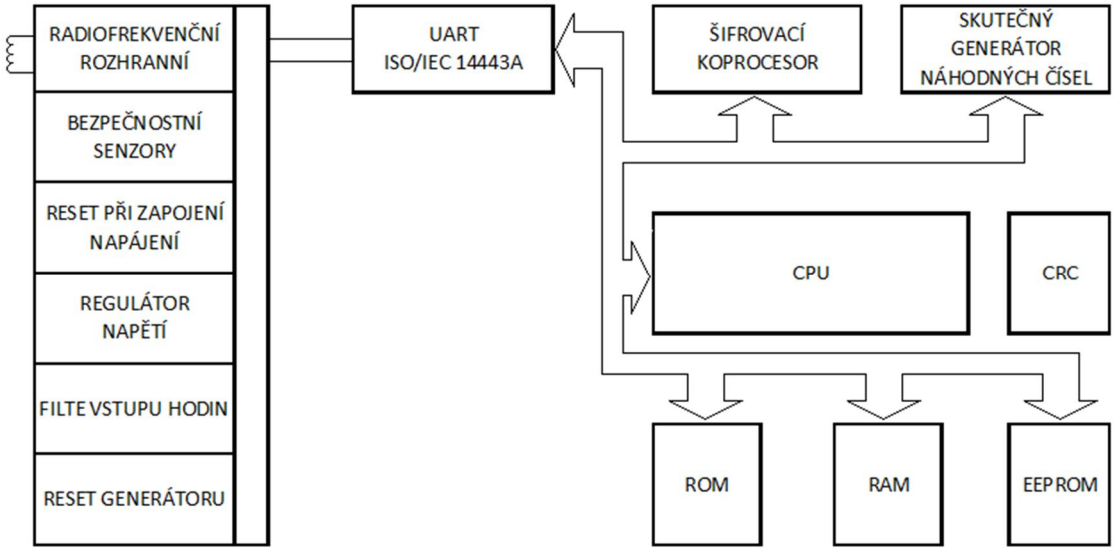
B 2: : Struktura paměti EEPROM karty MIFARE Classic 4K [9]

příloha C

Bloková schémata karet MIFARE



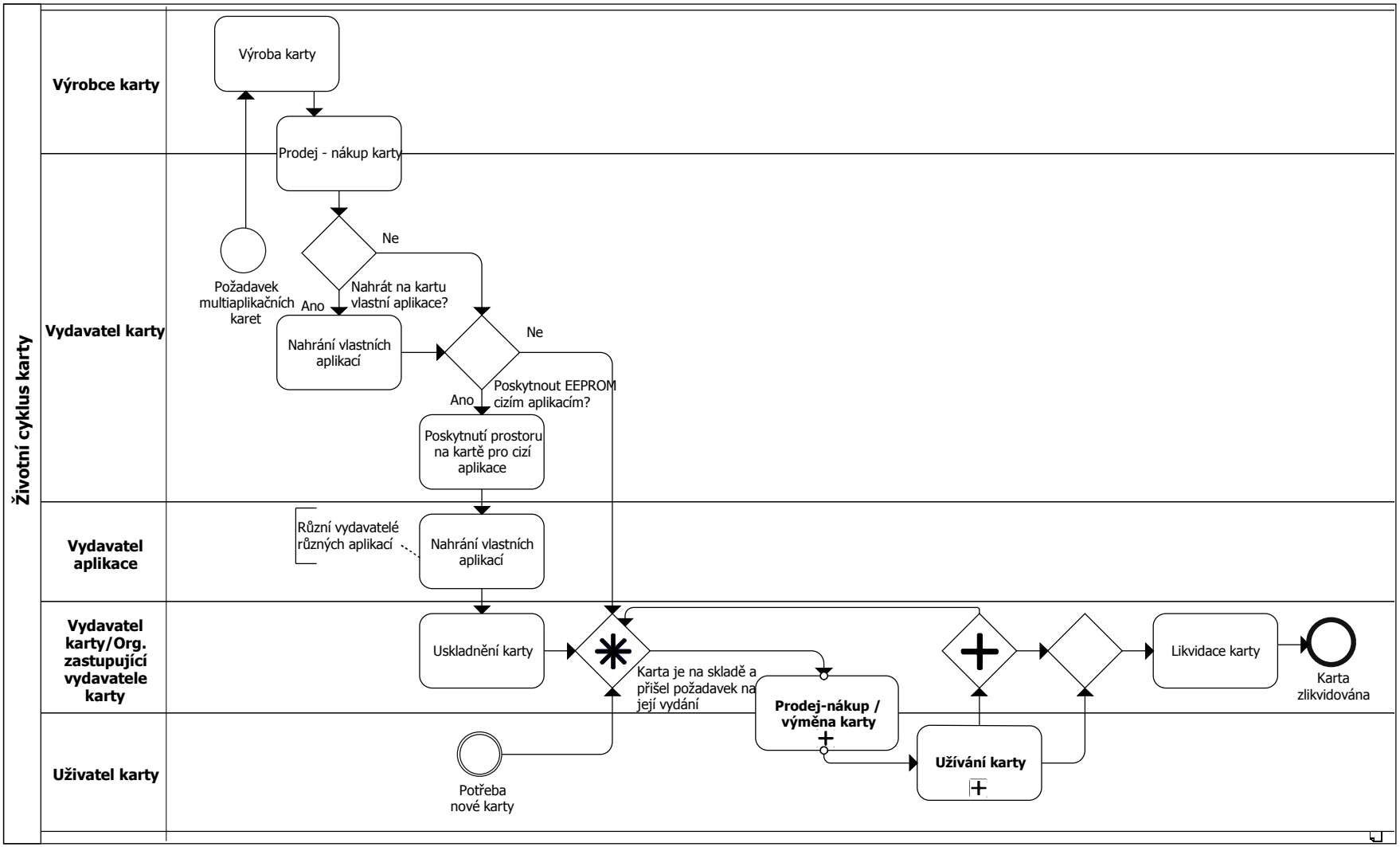
C 1: Blokové schéma čipu MIFARE Classic [9]



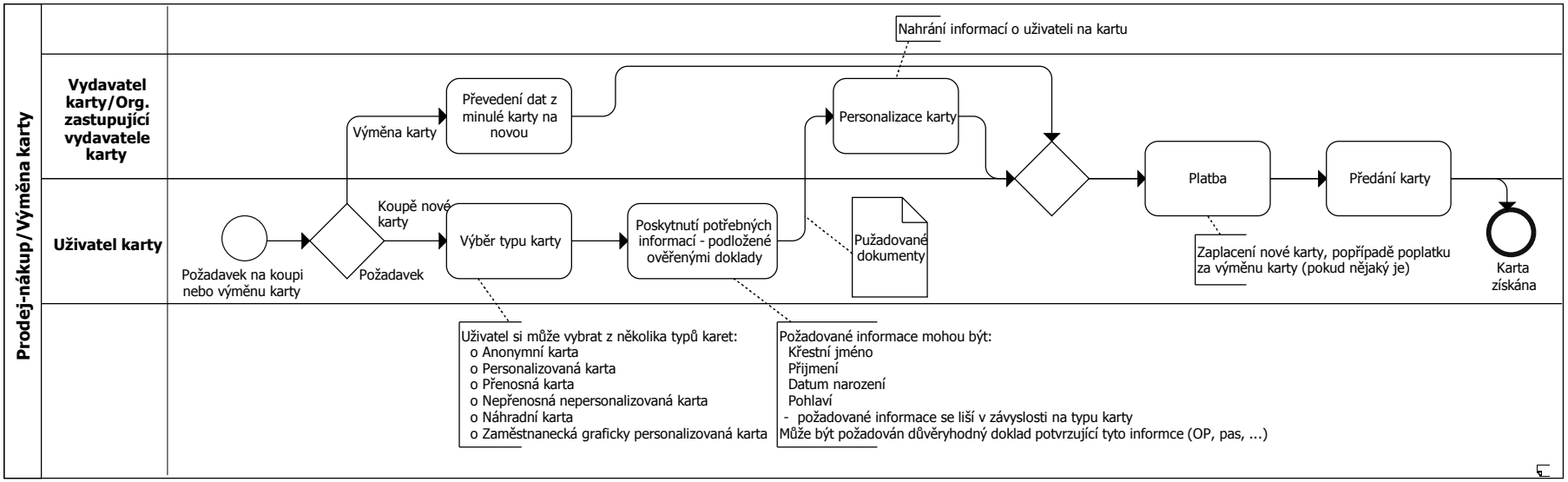
C 2: Blokové schéma čipu MIFARE DESFire [10]

příloha D

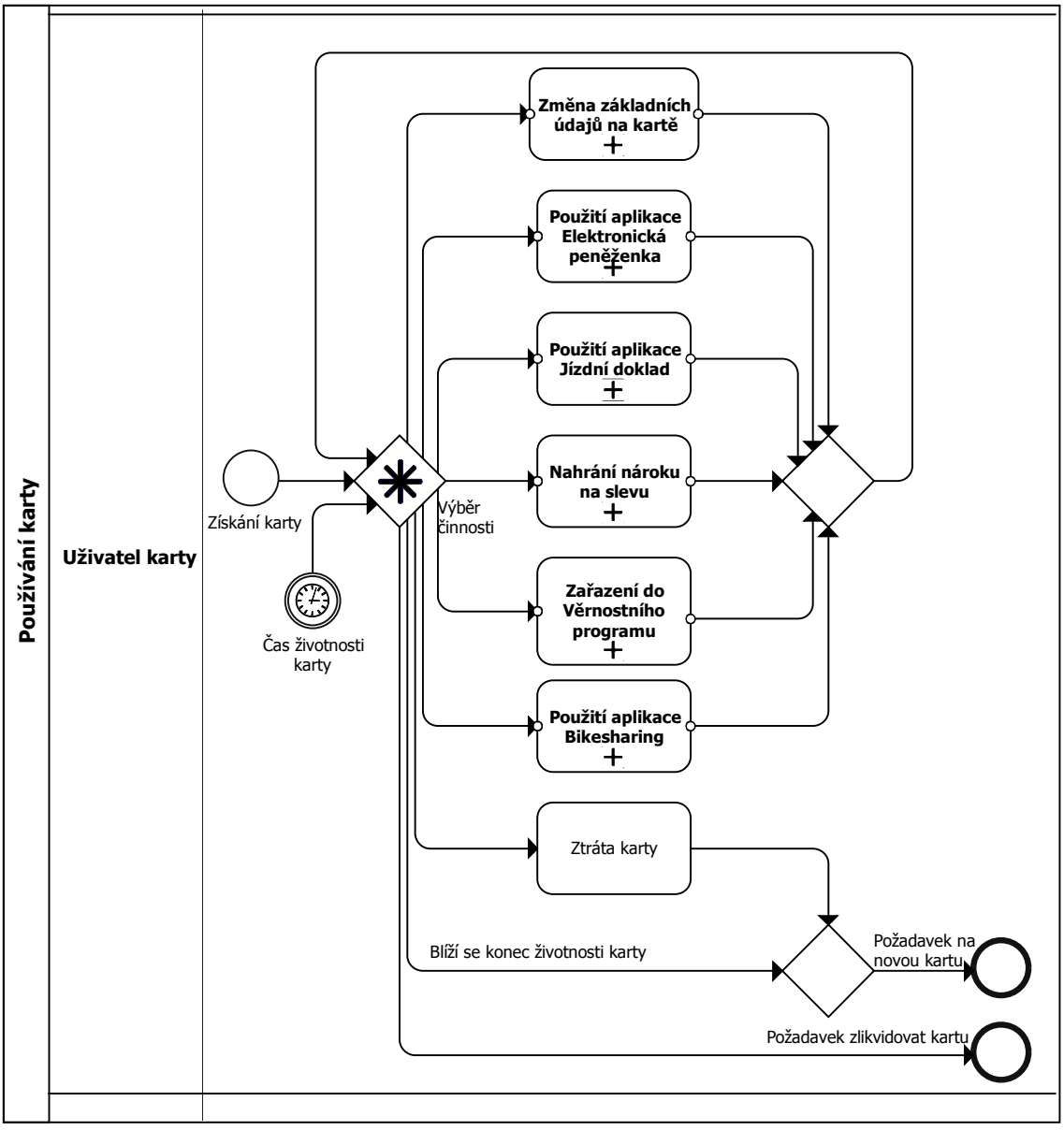
Procesní model využití multiaplikační karty



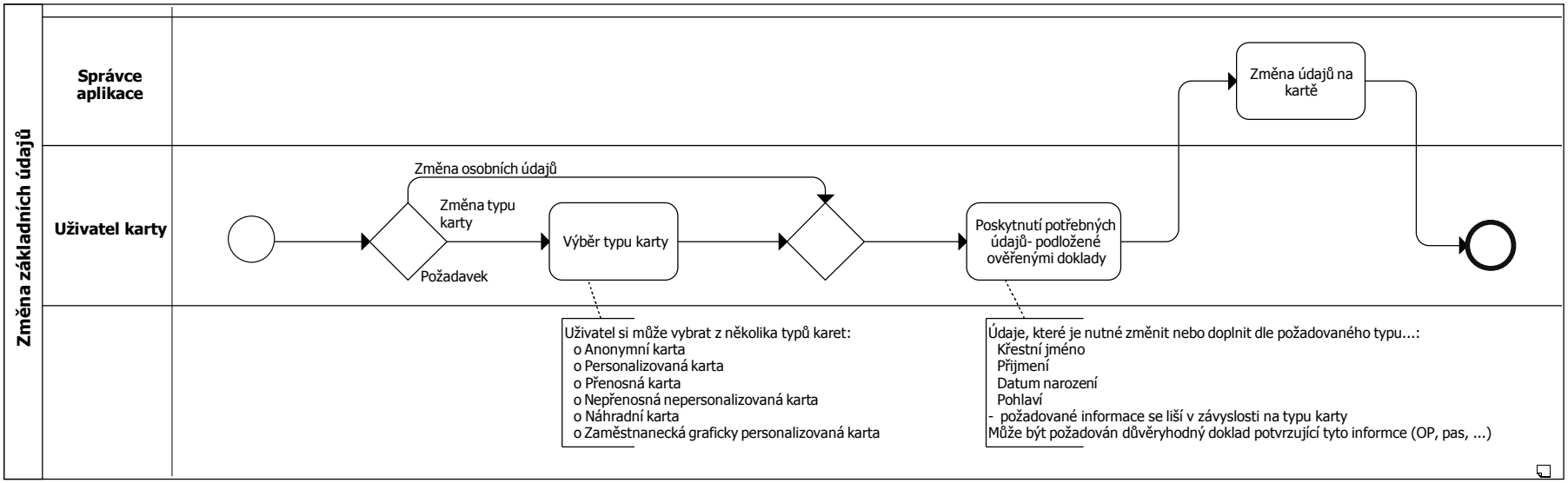
D1: Procesní model: Životní cyklus multiaplikační karty



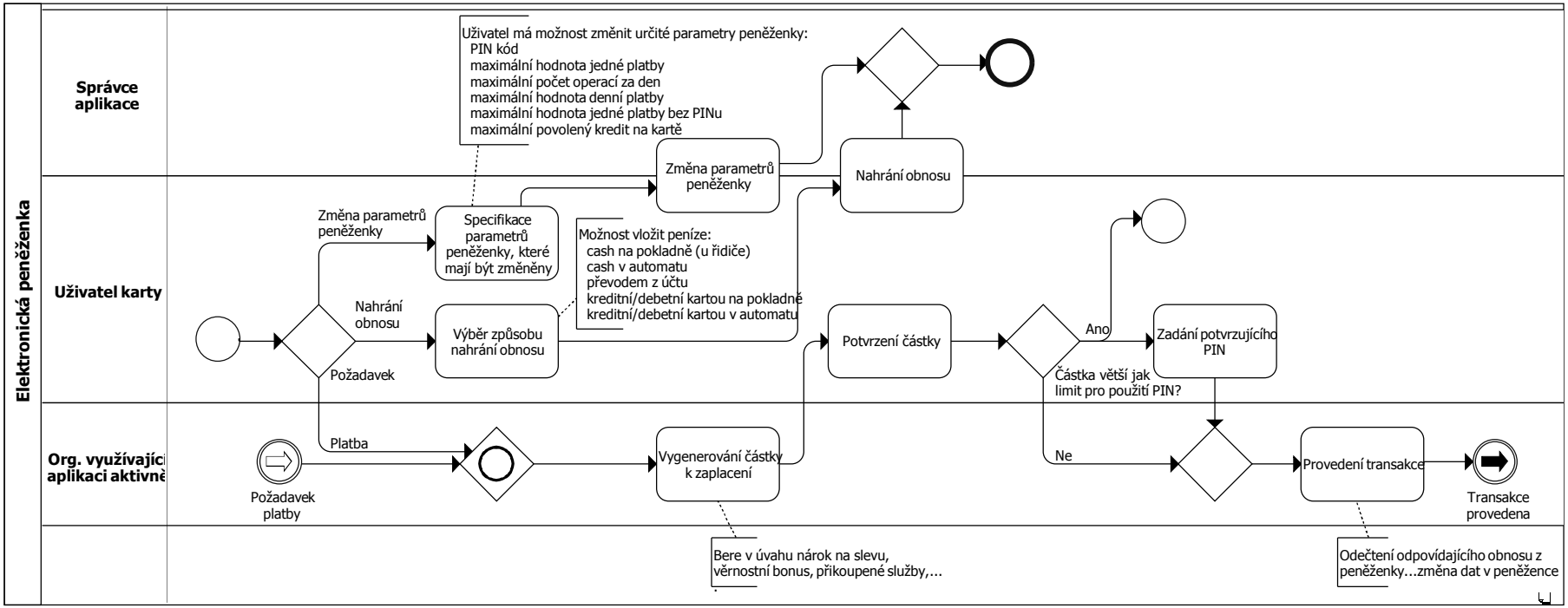
D2: Procesní model: Prodej-nákup/Výměna karty



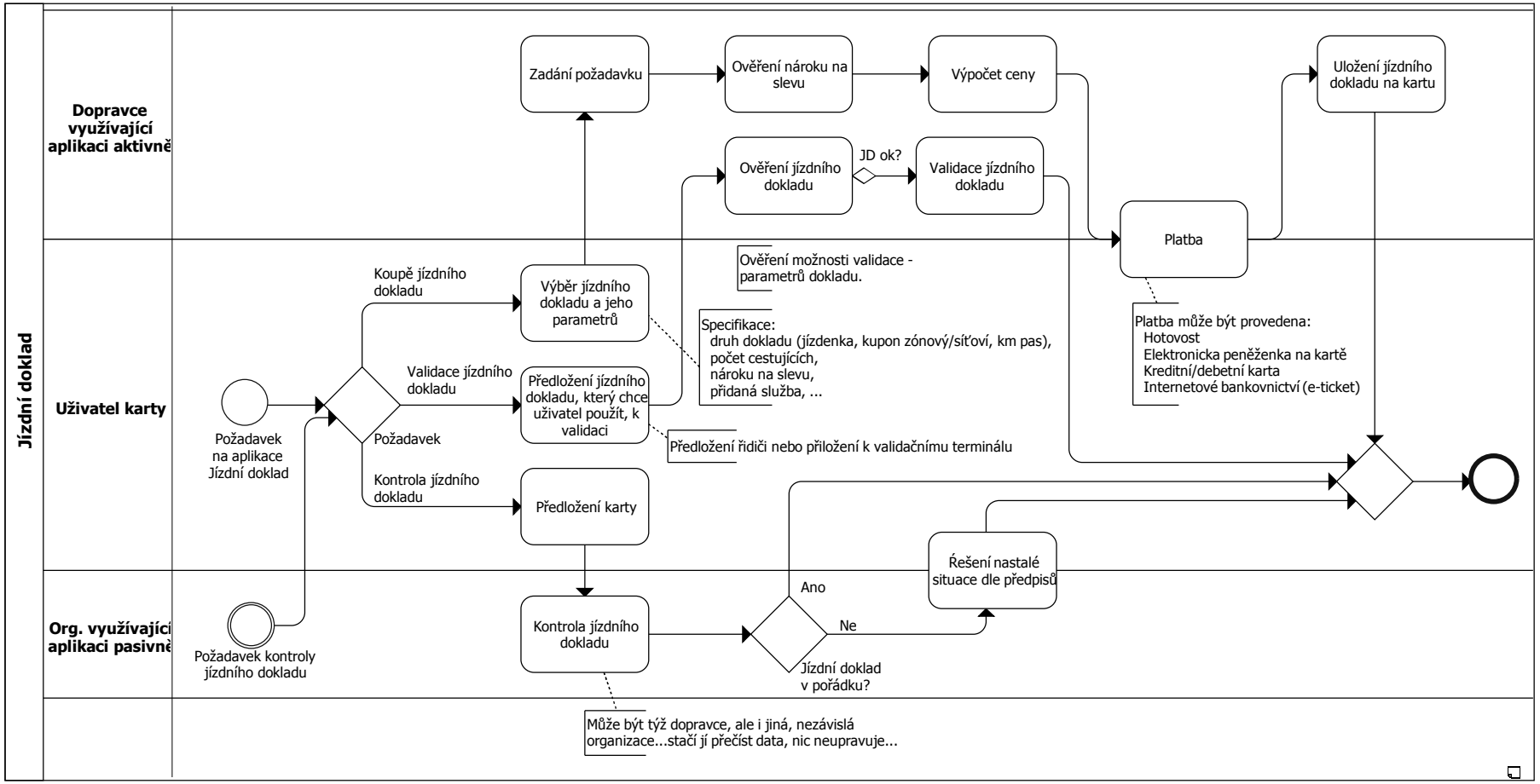
D3: Procesní model: Užívání karty



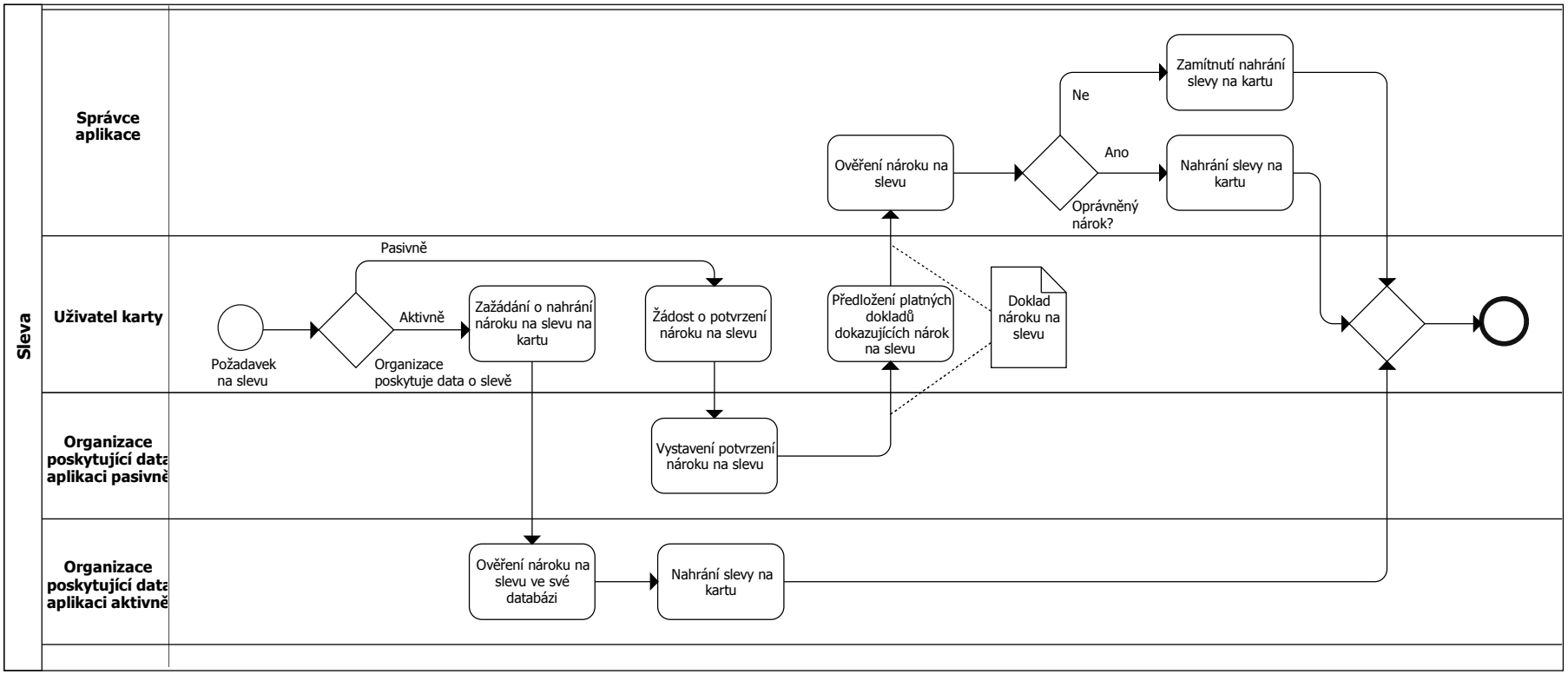
D4: Procesní model: Změna základních údajů na kartě



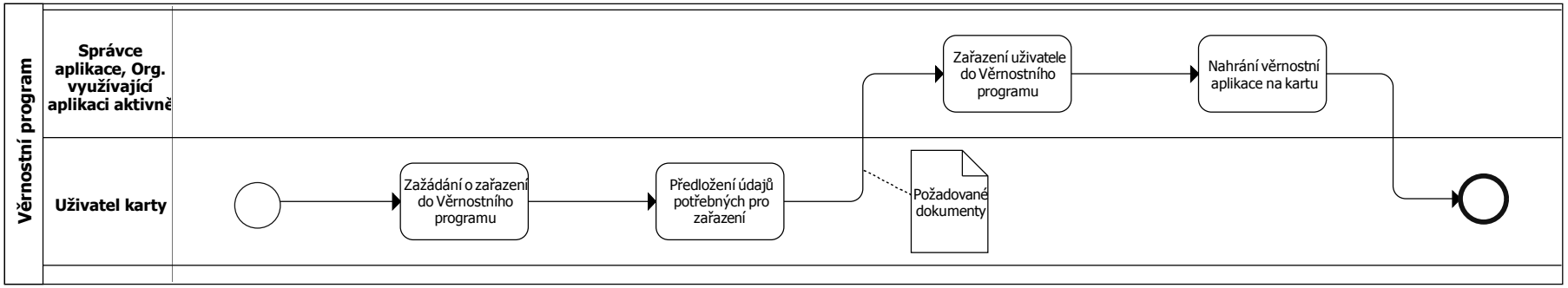
D5: Procesní model: Použití aplikace Elektronická peněženka



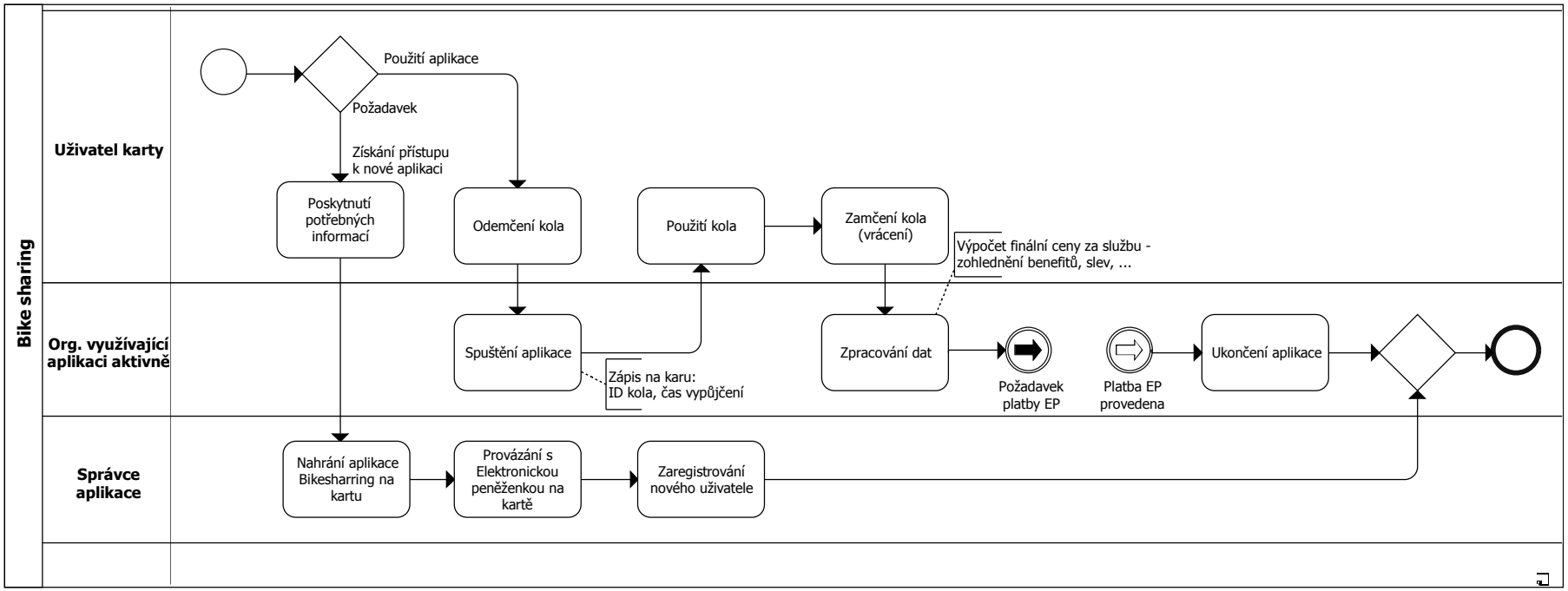
D6: Procesní model: Použití aplikace Jízdní doklad



D7: Procesní model: Nahrání nároku na slevu



D8: Procesní model: Zařazení do věrnostního programu



D9: Procesní model: Použití aplikace BikeSharing