UNIVERZITET SINGIDUNUM
Fakultet za poslovnu informatiku
Danijelova 32
11000 Beograd

Magistarski rad

P A M E T N E
K A R T I C E

Mentor:
Prof. dr Mladen Veinović

Kandidat:
Miroslav Pavlović

Beograd 2007

Sadržaj

Sadržaj **1**
Sažetak **10**
Abstract **10**
Uvod **11**
Indeks slika **13**
I deo – Uvod. Osnovna znanja 13
II deo – Sigurnost pametne kartice 13
III deo – Standardi 14
IV deo – JAVA CARD tehnologija 15
V deo – Sigurna autentifikacija sa pametnim karticama 15
Indeks tabela **16**
I deo – Uvod. Osnovna znanja 16
II deo – Sigurnost pametne kartice 16
III deo – Standardi 16
IV deo – JAVA CARD tehnologija 17
V deo – Sigurna autentifikacija sa pametnim karticama 17
Deo I **18**
Uvod **18**
1. Osnovna znanja **19**
1.1. Tipovi kartica 19
1.2. Električna i fizička svojstva pametne kartice 21
1.2.1. Formati i dimenzije 21
1.2.2. Čip 21
1.2.3. Kontaktna površina 23
1.2.4. CPU 24
1.2.5. Koprocesor 24
1.2.6. Memorija 24
1.3. Komunikacioni model 26
1.3.1. CAD uređaj i aplikacija terminala 26
1.3.2. Komunikacioni model sistema sa karticama 26
1.3.3. APDU protokol 27
zaglavlje 27
telo 27
1.3.4. TPDU protokol 28
1.3.5. ATR 29
1.3.6. PTS 29
1.4. Operativni sistem kartice 30
1.5. Životni ciklus kartice 32
1.6. Sistem kartice 33
1.7. Standardi i specifikacije 34
1.7.1. ISO 7816 standard 34
1.7.2. GSM 35

1.7.3. EMV ... 35
1.7.4. Open Platform ... 35
1.7.5. OpenCard Framework ... 35
1.7.6. PC/SC ... 36
Deo II ... 37
2. Sigurnost pametne kartice ... 37
2.1. Klasifikacija napada i napadača ... 38
2.1.1. Klasifikacije napada ... 38
2.1.2. Posledice napada i klasifikacija napadača ... 40
2.2. Napadi i mehanizmi odbrane u fazi razvoja ... 41
2.2.1. Razvoj mikrokontrolera pametne kartice ... 42
2.2.1.1. Zaštita: dizajnerske mere ... 42
2.2.1.2. Zaštita: jedinstven broj čipa ... 42
2.2.2. Razvoj operativnog sistema pametne kartice ... 43
2.2.2.1. Zaštita: Principi razvoja ... 43
2.2.2.2. Zaštita: Deljenje znanja ... 43
2.3. Napadi i mehanizmi odbrane u fazi proizvodnje ... 44
2.3.1.1. Zaštita: autentifikacija u završnim koracima ... 44
2.4. Napadi i mehanizmi odbrane u fazi korišćenja kartice ... 44
2.4.1. Napadi na fizičkom nivou ... 45
2.4.2. Statička analiza mikrokontrolera pametne kartice ... 46
2.4.2.1. Zaštita: tehnologija proizvodnje poluprovodnika ... 46
2.4.2.2. Zaštita: dizajn čipa ... 47
2.4.2.3. Zaštita: lažne strukture ... 47
2.4.2.4. Zaštita: magistrale (magistrale) čipa ... 47
2.4.2.5. Zaštita: dizajn memorije ... 48
2.4.2.6. Zaštita: zaštitni slojevi ... 48
2.4.2.7. Napad i odbrana: čitanje sadržaja izbrisive memorije ... 48
2.4.2.8. Zaštita: kodiranje memorije ... 48
2.4.2.9. Napad i odbrana: šifrovano čuvanje PIN-a ... 49
2.4.3. Dinamička analiza mikrokontrolera pametne kartice ... 49
2.4.3.1. Zaštita: nadziranje pasivnog sloja ... 49
2.4.3.2. Zaštita: nadziranje naponskog nivoa ... 49
2.4.3.3. Zaštita: nadziranje frekvencije takta ... 50
2.4.3.4. Zaštita: nadziranje temperature ... 50
2.4.3.5. Zaštita: kodiranje magistrala ... 50
2.4.3.6. Zaštita: nepovratno prespajanje iz testnog načina rada u korisnički način rada ... 51
2.4.4. Dinamička analiza i mehanizmi odbrane: prisluškivanje memorijskih magistrala mikrokontrolera ... 52
2.4.5. Dinamička analiza i mehanizmi odbrane: merenje potrošnje struje CPU-a ... 53
2.4.6. Analiza i mehanizmi odbrane: merenje elektromagnetskog zračenja CPU-a. 55
2.4.7. Manipulacija mikrokontrolera pametne kartice ... 55
2.4.7.1. Manipulacija i mehanizmi odbrane: promene sadržaja memorije mikrokontrolera pametne kartice ... 55
2.4.8. Napadi na logičkom nivou ... 56

2.4.8.1. Analiza: određivanje skupa naredbi pametne kartice 56
2.4.8.2. Napad: prisluškivanje prenosa podataka 57
2.4.8.3. Napad i odbrana: uklanjanje napajanja 58
2.4.8.4. Napad i odbrana: strujna analiza za vreme upoređivanja PIN-a 58
2.4.8.5. Napad i odbrana: vremenska analiza za vreme upoređivanja PIN-a 59
2.4.8.6. Zaštita: šifarski algoritmi sa odsutnošću šuma 59
2.4.8.7. Manipulacija: diferencijalna analiza grešaka (DFA) 60
2.4.9. Zaštitni elementi operativnog sistema pametne kartice 62
2.4.9.1. Zaštita: hardverski i softverski testovi 62
2.4.9.2. Zaštita: slojevitost operativnog sistema 62
2.4.9.3. Zaštita: kontrola prenosa podataka 62
2.4.9.4. Zaštita: kontrolna suma memorijskog sadržaja 62
2.4.9.5. Zaštita: ukalupljivanje aplikacija 63
2.4.9.6. Zaštita: maskiranje stvarnih aktivnosti operativnog sistema 63
2.4.9.7. Zaštita: objektno-orijentisana kontrola pristupa 63
2.4.9.8. Napad i odbrana: generatori slučajnih brojeva 63
2.4.10. Zaštitni elementi aplikacije pametne kartice 64
2.4.10.1. Zaštita: jednostavnost mehanizama zaštite 64
2.4.10.2. Zaštita: konzervativne dozvole pristupa 64
2.4.10.3. Zaštita: mašina stanja 64
2.4.10.4. Zaštita: dvostruka kontrola pristupa 65
2.4.10.5. Zaštita: siguran prenos podataka 65
2.4.10.6. Zaštita: funkcije oporavka od greške 65
2.4.10.7. Zaštita: autentifikacija 65
2.4.10.8. Zaštita: on-line ponašanje 66
2.4.10.9. Zaštita: crne liste 66
2.4.10.10. Napad i odbrana: virusi i trojanski konji 66
2.4.10.11. Napad i odbrana: prostor tajnog ključa 67
Deo III 69
3. Standardi 69
3.1. Standard ISO 7816 69
3.1.1. ISO 7816-1 69
3.1.2. ISO 7816-2 72
3.1.3. ISO 7816-3.1 73
3.1.3.1. Napon i struja napajanja 74
3.1.3.2. I/O Signal 74
3.1.3.3. Signal vremenskog vođenja 75
3.1.3.4. Reset signal 75
3.1.4. ISO 7816 3.2 76
3.1.5. ISO 7816 3.2.a Spajanje i aktiviranje kontakata 76
3.1.6. ISO 7816 3.2.b Reset kartice 76
3.1.7. ISO 7816 3.2.c Deaktivacija kontakata 76
3.1.8. ISO 7816 3.3. Odgovor na reset 77
3.1.8.1. ISO 7816 3.3.a Odgovor na reset kod asinhronog prenosa 77
3.1.8.2. Izgled znakova 77
3.1.8.3. Detekcija grešaka i ponavljanje znaka 78

3.1.8.4. Odgovor na reset ... 78
3.1.8.5. Inicijalni znak TS ... 78
3.1.8.6. Znak formata T0 ... 80
3.1.8.7. Znaci interfejsa TAi, TBi, TCi, TDi ... 81
TA1 ... 82
TB1 ... 83
TC1 ... 83
3.1.8.8. Historical znaci ... 84
3.1.8.9. Znak provere TCK ... 84
3.1.9. ISO 7816 3.4 Selekcija parametara protokola (PPS) ... 84
3.1.9.1. Protokoli za prenos podataka ... 87
3.1.10. ISO 7816 3.5. Komunikacioni protokol T=0 ... 88
3.1.10.1. T=0 sekvenca naredbe i odgovora ... 89
3.1.11. Komunikacioni protokol T=1 ... 93
3.1.11.1. Struktura bloka ... 95
3.1.11.2. Prolog polje ... 95
Adresa čvora (NAD) ... 96
Kontrolni bajt protokola (PCB) ... 96
Polje dužine (LEN) ... 96
3.1.11.3. Polje informacije ... 96
3.1.11.4. Epilog polje ... 97
3.1.11.5. Primer komunikacije protokolom T=1 ... 97
3.1.11.6. Vremena čekanja ... 98
Vreme čekanja na znak (CWT) ... 98
Vreme čekanja na blok(BWT) ... 99
Čuvano vreme bloka (BGT) ... 99
3.1.11.7. Detekcija i rukovanje greškama ... 99
3.1.12. ISO 7816-4 ... 101
3.1.12.1. Struktura datoteka ... 101
Glavna datoteka (MF) ... 102
Datoteka direktorijuma (DF) ... 102
Elementarna datoteka (EF) ... 102
Radni EF ... 102
Interni EF ... 102
3.1.12.2. Identifikator aplikacije (AID) ... 104
3.1.12.3 Strukuture elementarnih datoteka ... 104
Transparentna struktura ... 105
Linearna fiksna struktura ... 105
Linearna varijabilna (promenljiva) struktura ... 105
Ciklična struktura zapisa ... 105
Struktura izvršne datoteke ... 106
Struktura datoteke baze podataka ... 106
3.1.12.4. Struktura poruka u komunikaciji kartice i čitača (APDU) ... 106
3.1.12.5. APDU naredbe ... 106
3.1.12.6. APDU odgovora ... 107
3.1.12.7. Osnovne naredbe ... 108

ERASE BINARY 108
VERIFY 108
EXTERNAL AUTENTIFICATION 108
INTERNAL AUTENTIFICATION 108
SELECT FILE 108
READ BINARY 108
READ RECORD 109
GET RESPONSE 109
WRITE BINARY 109
WRITE RECORD 109
3.1.13. Drugi komunikacioni protokoli 109
3.1.13.1. Protokol 'T=14' 109
3.1.13.2. USB protokol prenosa 109
3.2. EMV Standard 110
3.2.1. Prednosti EMV standarda 111
3.2.1.1. Povećana sigurnost i smanjena zloupotreba 111
3.2.1.2. Učinak procesiranja stalno rastućeg broja transakcija 112
3.2.1.3. Interoperabilnost različitih platnih aplikacija i dodatnih usluga 112
3.2.2. Hronologija EMV 113
3.2.3. Sigurnost i rukovanje ključevima kod pametnih kartica 113
3.2.3.1. Statička autentifikacija podataka 113
3.2.3.2. Dinamička autentifikacija podataka 115
3.2.3.3. Enkripcija ličnog identifikacionog broja (PIN) 116
3.2.3.4. Aplikativni kriptogram i autentifikacija izdavača 117
3.2.3.4.1. Generisanje aplikativnog kriptograma 117
3.2.3.4.2. Autentifikacija izdavača 118
3.2.3.5. Sigurno slanje poruka 118
3.2.3.5.1. Formati poruka 118
3.2.3.5.2. Sigurno slanje poruka u svrhu zaštite integriteta i autentifikacije 119
Polje podataka naredbe 119
Izračunavanje MAC-a 120
3.2.3.5.3. Sigurno slanje poruka za osiguranje poverljivosti 121
Polje podataka naredbe 121
3.2.3.6. Javni ključ sertifikatora 121
3.2.3.6.1. Životni ciklus javnog ključa sertifikatora 122
Normalni životni ciklus javnog ključa sertifikatora 122
Planiranje 122
Generisanje 122
Distribucija 123
Upotreba 123
Poništavanje 124
Kompromitovanje para javnih ključeva za autorizaciju sertifikata 124
Detektovanje 125
Procena 125
Odluka 125
Poništavanje (ubrzano) 126

3.2.3.6.2. Vremenski odnosi ... 126
Uvodjenje ključa ... 127
Ukidanje ključa ... 128
3.2.3.7. Sigurnost terminala i zahtevi za rukovanje ključevima ... 128
3.2.3.7.1. Sigurnosni zahtevi ... 128
Sigurni uređaji (engl.Tamper-Evident Devices) ... 128
Fizička sigurnost ... 129
Logička sigurnost ... 129
Tastatura za unos identifikacionog broja (engl.PIN Pad) ... 130
3.2.3.7.2. Zahtevi za rukovanje ključem ... 131
Uvođenje javnih ključeva sertifikatora u terminal ... 131
Spremanje javnih ključeva sertifikatora u terminal ... 131
Ukidanje javnog ključa sertifikatora ... 133
3.3. Global platform ... 133
3.4. Standardizacija pametnih kartica u šifarskim sistemima ... 135
3.4.1. Mogućnosti pametnih kartica u šifarskim sistemima ... 135
3.4.2. Digitalni sertifikat ... 136
3.4.2.1. Standard X.509 ... 136
Deo IV ... 137
4. JAVA Card tehnologija ... 137
4.1. Pregled arhitekture ... 137
4.2. Java Card jezički podskup ... 138
4.3. Java Card virtuelna mašina ... 139
4.3.1. CAP datoteke i izvedene datoteke ... 140
4.3.2. Java Card konvertor ... 141
4.3.3. Java Card interpreter ... 141
4.4. Java Card instaler i instalacija programa izvan kartice ... 142
4.5. Java Card izvršna okolina ... 143
4.5.1. Vreme života Java Card izvršne okoline ... 145
4.5.2. Funkcionisanje JCRE za vreme CAD sesije ... 146
5.5.3. Svojstva Java Card izvršne okoline ... 146
4.6. Java Card API ... 147
4.6.1. java.lang paket ... 147
4.6.2. javacard.framework paket ... 148
4.6.3. javacard.security paket ... 148
4.6.3.1. Paket `java.security` ... 149
4.6.3.2. Paket `java.security.cert` ... 151
4.6.3.3. Paket `java.security.interfaces` ... 154
4.6.3.4. Paket `java.security.spec` ... 154
4.6.4. javacardx.crypto paket ... 155
4.7. Sertifikacioni putevi ... 156
4.8. Java™ Secure Socket Extension ... 158
4.9. Java™ Authentication and Authorization Service ... 158
4.10. Java™ Generic Security Service ... 158
4.11. Java™ Cryptography Extension ... 158
4.11.1. Paket `javax.crypto` ... 158

4.11.2. Paket `javax.crypto.interfaces` ... 161
4.11.3. Paket `javax.crypto.spec` ... 161
4.12. Java Card Appleti ... 161
4.13. Dogovor vezan za pridruživanje imena paketu i appletu ... 161
4.14. Razvojni proces appleta ... 162
4.15. Instalacija appleta ... 164
4.15.1. ROM appleti ... 164
4.15.2. Predizdati i postizdati appleti ... 164
4.15.3. Instalacija postizdatih appleta ... 164
4.15.4. Oporavak od greške za vreme instalacije appleta ... 165
4.15.5. Instalaciona ograničenja ... 165
4.16. Sigurnosne osobine Java Card platforme ... 166
4.16.1. Sigurnosne osobine Java jezika ... 166
4.16.2. Dodatne sigurnosne osobine Java Card platforme ... 166
4.16.3. Sigurnost appleta ... 168
Deo V ... 169
5. Sigurna autentifikacija ... 169
5.1. Osnove računarske sigurnosti ... 169
5.1.1. Sigurnosni incidenti ... 169
5.1.2. Sigurnosni zahtevi ... 170
5.1.3. Kriptografija ... 170
5.1.3.1. Simetrična kriptografija ... 171
5.1.3.2. Asimetrična kriptografija ... 172
5.1.4. Sažetak poruke ... 173
5.1.5. Elektronski potpis ... 173
5.1.5.1. Svojeručni potpis ... 173
5.1.5.2. Potpis elektronskih informacija ... 173
5.1.5.3. Razlike između elektronskog i digitalnog potpisa ... 174
5.1.6. Digitalni potpis ... 174
5.1.6.1. Provera digitalnog potpisa ... 174
5.1.7. Digitalni sertifikat ... 176
5.1.7.1. X.509 sertifikat ... 176
5.1.7.2. Formati zapisa X.509 sertifikata ... 179
5.1.8. Protokoli za sigurnu razmenu podataka ... 180
5.1.8.1. SSL protokol ... 181
SSL protokol nad podacima ... 182
SSL protokol promene specifikacije algoritma ... 182
SSL protokol upozorenja ... 182
SSL protokol rukovanja ... 183
Poruka početka klijenta (engl. ClientHello) ... 183
Poruka početka servera (engl. ServerHello) ... 184
Sertifikat servera ... 184
Poruka razmene ključeva servera (engl. ServerKeyExchange) ... 184
Poruka zahteva server za sertifikatom klijenta ... 184
Poruka servera o završetku inicijalne komunikacije (engl. ServerHelloDone) ... 184

Sertifikat klijenta ... 184
Razmena ključeva klijenta ... 184
Provera sertifikata ... 184
Poruka promene specifikacije algoritma ... 185
Poruka završetka (engl. Finished) ... 185
5.2. Infrastruktura javnog ključa ... 185
5.2.1. Sertifikacioni centar ... 188
5.2.1.1. Vrhovni sertifikacioni centar ... 188
5.2.1.2. Ulančani sertifikacioni centar ... 189
5.3. Autentifikacija ... 189
5.3.1. X.509 autentifikacioni protokoli ... 190
5.3.1.1. X.509 protokol sa jednom porukom ... 190
5.3.1.2. X.509 protokol sa dve poruke ... 191
5.3.1.3. X.509 protokol sa tri poruke ... 192
Zaključak ... 193
Pravci razvoja ... 193
Literatura ... 195

Sažetak

U ovom magistarskom radu kroz pet poglavlja opisane su osnovne karakteristike pametne kartice kroz osnovna znanja, sigurnost i klasifikacije napada i napadača, posledica i mehanizama odbrane kroz sve faze: razvoja, proizvodnje i korišćenja kartice. U trećem poglavlju opisani su standardi koji unificirano opisuju njihove osobenosti i nešto čega se svi moraju pridržavati. Razvoj aplikacija se javio usled zahteva na sve široj upotrebi kartica, a JAVA CARD tehnologija omogućila je da se programi pisani u Java programskom jeziku izvode na pametnim karticama i drugim uređajima ograničenih sredstava. Ova tehnologija opisana je u četvrtom poglavlju. I konačno podaci koji se razmenjuju mogu lako biti vidljivi svima prisutnim u istom mrežnom okruženju. Samim time nameće se više pitanja vezanih za računarsku sigurnost. Najvažnija pitanja su sigurna autentifikacija i tajnost informacija koje se razmenjuju. U petom poglavlju opisana je osnova računarske sigurnosti kroz sažetak poruke, elektronski i digitalni potpis, sertifikat, infrastrukturu javnog ključa i autentifikaciju preko X.509 protokola.

Abstract

In this master work through five chapters the basic characteristics of smart card were described in basic knowledge, security and classification of aggression and aggressor, consequences and mechanism of defence through all phases: development, production and usage a card. In the third chapter are described the standards which describe on an unified way their individualities and something that everybody has to abide by. Development of applications appeared because of request for using the cards more and more, but JAVA CARD technology enabled the programmes written in JAVA programme language to do on smart card and other devices in limited means. This technology was described in the fourth chapter. Finally the data which are exchanged can be visible to everybody in the same enviroment. In this way there are more questions connected with computing security. The most important questions are authentification and secrecy of informations which are enchanged. In the fifth chapter is described the base of computing security through resume of the message, electronic and digital signature, certificate, infrastruction of public key and authentification over X.509 protocols

Uvod

Tehnologija ulazi u sva područja ljudske delatnosti, stvara novi soj ljudi i poslova, ona je uzrok evolucije društva. Jedan izum objedinjuje i simbolizuje navedene promene, računar.

Prvi računari izvodili su nekoliko računskih operacija, a današnji računari izvode 3D animacije, izvršavajući pritom bilione operacija brzinom svetlosti. Razvoj poluprovodničke industrije omogućio je da se veličina računara sa dimenzijama jedne sobe spusti na površinu od nekoliko kvadratnih milimetara. Time je stvoren prostor za razvoj novih tehnologija. Jedna od njih je i pametna kartica.

Najveće finansijske organizacije, Europay, Mastercard i Visa udružuju svoje napore u cilju primene pametne kartice u finansijskoj industriji. Mobilna telefonija kroz GSM specifikaciju otvara novo veliko područe primene pametne kartice. Telekomunikaciona i finansijska industrija predstavljaju najveće sisteme kartica, ali one ne određuju esencijalna svojstva pametne kartice, one ih samo koriste. Pametna kartica poseduje neograničene mogućnosti, njene najveće prednosti su veličina, prenosivost, višekratna upotrebljivost, raznolikost funkcionalnosti i sigurnost.

Sigurnost predstavlja najvažnije svojstvo pametne kartice. Današnje društvo evoluira u informaciono društvo. Tehnologija postaje alat u službi informacije, a informacija znanje, moć i novac. Sigurno čuvanje informacija i zaštita protoka informacija postaju nužnost i neophodnost. Kao idealno rešenje javlja se pametna kartica. Pametna kartica čini visok nivo sigurnosti informacije dostupnu svakom, na siguran način čuva informacije i omogućuje da se procesiranje podataka izvodi u sigurnom okruženju.

Pametna kartica pruža podacima i programima sigurno okruženje. Kad količina napora da se pročitaju podaci sa pametne kartice ne bi bila tako velika, kartica bi bila ništa više nego disketa sa drugačijom ulogom.

Praktično je nemoguće sagraditi potpuno siguran sistem, pa čak i potpuno sigurnu pametnu karticu. Ako napor utrošen na napad dostigne dovoljno visok nivo, moguće je prodreti u bilo koji sistem i na njemu izvršiti neovlašćene promene. Svaki potencijalni napadač svesno ili nesvesno procenjuje dobit koju mu donosi uspešan napad. Vreme, novac i napor uloženi na provalu u sistem moraju biti vredni dobijene nagrade. Ako nagrada nije dostojna uloženog napora do napada nikad neće doći.

Sigurnost pametne kartice obuhvata četiri komponente. Prva komponenta je telo kartice. Telo kartice implementira mnoštvo sigurnosnih obeležja, ne samo mašinski-čitljivih i time podvrgnutih kontroli uređaja, već i vizuelno-čitljivih i time podvrgnutih kontroli čoveka. Npr. to su holografi, reljefna obeležja, ultraljubičasti tekst, laserska graviranja, itd. Tehnike implementacije sigurnosnih obeležja zajedničke su za sve tipove kartica, nisu specifičnost pametnih kartica. Preostale komponente – hardver čipa, operativni sistem i aplikacija – štite podatke i programe mikrokontrolera kartice.

Kao i na drugim tehničkim područijima i kod pametnih kartica postoje standardi koji opisuju njihove osobenosti. Nedostatak standarda prvih godina razvoja se ogleda u nekompatibilnosti kartica pojedinih proizvođača mada se situacija poslednjih godina poboljšava.

Postoji čitav niz standarda koji se odnose na pametne kartice. Oni koji se odnose na njihove fizičke karakteristike su generalno usaglašeniji od onih za softversku arhitekturu. ISO 7816 pod punim imenom ***I**ntegrated **C**ircuit **C**ards with **E**lectrical contacts* je međunarodni standard za mikroprocesorske kartice sa kontaktima. To je ujedno najšire prihvaćen standard vezan uz pametne kartice. Osim njega vrlo su rašireni i **ETSI** (engl.***E**uropean **T**elecommunications **S**tandards **I**nstitute*) standardi koji određuju **SIM** (engl.***S**ubscriber **I**dentity **M**odule*) kartice u mobilnim telefonima.

Usled zahteva na sve široj primeni pametnih kartica javila se potreba za bržim razvojem aplikacija kartica. Brži razvoj aplikacija kartica ne sme ugroziti temeljni razlog korišćenja pametnih kartica, sigurno procesiranje. To je za razvojne programere pametnih kartica najvažnije pitanje, pitanje sigurnosti.

Java Card tehnologija omogućuje da se programi pisani u Java programskom jeziku izvode na pametnim karticama i drugim uređajima ograničenih sredstava. Nasleđujući sigurnosne osobine Java jezika, istovremeno pruža još jedno svojstvo, nezavisnost aplikacije kartice od operativnog sistema pametne kartice. Dodatno svojstvo je i mogućnost učitavanja aplikacija na kartici u korisničkoj fazi životnog ciklusa pametne kartice. Time pametna kartica postaje aktivni deo mrežne arhitekture.

Sigurnost se mora uzeti u obzir sa tačke gledišta ukupnog, opšteg sistema. Java Card platforma je izgrađena na vrhu kartične platforme, koju sačinjava kartični hardver, izvorni operativni sistem, terminal s kojim kartica komunicira i sistem u pozadini.

Komunikacija, pristup informacijama i obavljanje različitih vrsta transakcija u mrežnom okruženju postala je svakodnevica. Podaci koji se razmenjuju mogu lako biti vidljivi svima prisutnim u istom mrežnom okruženju. Samim time nameće se više pitanja vezanih za računarsku sigurnost. Najvažnija pitanja su sigurna autentifikacija i tajnost informacija koje se razmenjuju. Najčešći oblik autentifikacije je autentifikacija gde se korisnik predstavlja korisničkim imenom i lozinkom. Takva autentifikacija međutim ne garantuje veliku sigurnost.

Opis aspektata sigurnosti, operativnih sistema, standarda i nekih od načina bezbedne autentifikacije prožeti su kroz ovaj magistarski rad, govoreći kako o sigurnosti tako i slabostima i potencijalnim pretnjama koje su neizostavne u jednom ozbiljnom sistemu "...koji vredi da se napadne...".

Indeks slika

I deo – Uvod. Osnovna znanja (18-36)

Slika 1.1. Tipovi kartica
Slika 1.2. Fizički izgled pametne kartice
Slika 1.3. Formati i pripadajuće dimenzije pametnih kartica
Slika 1.4. Struktura modula i proces umetanja u telo kartice
Slika 1.5. Šematski prikaz mikrokontrolera pametne kartice
Slika 1.6. Kontaktna površina pametne kartice
Slika 1.7. Fizičke veličine memorijskih ćelija
Slika 1.8. Mikrokontroler pametne kartice površine 21 mm2, 32kB ROM-a, 16kB EEPROM-a, 1280 bajtova RAM-a. Dva neoznačena područija sa leve strane čine numerički koprocesor i skup perifernih komponenti. Na krajevima se vidi 5 spojnih mesta za ostvarivanje kontakata sa modulom čipa
Slika 1.9. Komunikacioni model pametne kartice
Slika 1.10. Struktura APDU naredbe
Slika 1.11. Struktura APDU odgovora
Slika 1.12. Mogući slučajevi APDU komunikacije
Slika 1.13. OSI komunikacioni model između terminala i pametne kartice
Slika 1.14. Inicijalni prenos podataka između terminala i pametne kartice
Slika 1.15. Sistem dadoteka na kartici
Slika 1.16. Strukture elementarnih datoteka
Slika 1.17. Životni ciklus pametne kartice prema ISO 10202-1

II deo – Sigurnost pametne kartice (37-67)

Slika 2.1 Klasifikacija sigurnosnih komponenti pametne kartice
Slika 2.2 Klasifikacija napada na pametnu karticu
Slika 2.3 Klasifikacija tipova napada
Slika 2.4 Klasifikacija napada u odnosu na vreme odvijanja napada
Slika 2.5 Klasifikacija mogućih tipova napadača
Slika 2.6 Klasifikacija napada na mikrokontroler kartice na fizičkom nivou
Slika 2.7 Fotografija poluprovodničke strukture i ljudske kose
Slika 2.8 Uporedni prikaz klasične i kodirane memorije
Slika 2.9 Kodiranje magistrala mikrokontrolera pametne kartice između memorije i CPU-a prikazano 8-bitnom magistrlom (
Slika 2.10 Pojednostavljeni prikaz potrošnje struje mikrokontrolera pametne kartice. Potrošnja struje zavisi od izvedenih mašinskih instrukcija
Slika 2.11 Alat sa kojim se komunikacija terminal-kartica vrši eksterno izvan čitača terminala. Kontakti koji ulaze u terminal vide se desno, a pločica za analizu nalazi se levo
Slika 2.12 Primer bučnog šifarskog algoritma

III deo – Standardi (69-136)

Slika 3.1. a) ID-1 format b) ID-000 format
Slika 3.2. Savijanje kartice i mašina za testiranje kartica

Slika 3.3. Ispitivanje otpornosti na statički elektricitet
Slika 3.4. ISO 7816-2 konektor
Slika 3.5. Izgled znakova
Slika 3.6. Detekcija grešaka i ponavljanje znaka
Slika 3.7. i 3.8. Konvencija o redosledu bitova i njihova interpretacija
Slika 3.9. Kompletan dijagram odgovora na reset
Slika 3.10. Dijagram stanja PPS procedure po 7816-3 standardu
Slika 3.11 Osnovna struktura i elementi PPS-a
Slika 3.12. Tipična PPS procedura
Slika 3.13. Klasifikacija korišćenih protokola za komunikaciju sa karticama
Slika 3.14. Struktura naredbe u T=0 protokolu
Slika 3.15. Prenos bitova putem U/I interfejsa koji koristi T=0 protokol
Slika 3.16. Prenos podataka sa čitača na karticu
Slika 3.17. Prenos podataka sa kartice na čitač
Slika 3.18. Dijagram toka T=0 protokola
Slika 3.19. OSI model komunikacije kartice i čitača
Slika 3.20. Struktura T=1 bloka za prenos podataka
Slika 3.21. Čitač šalje pojedinačne I blokove, kartica odgovara takođe I blokovima. Vidi se stanje bitova [N,M]
Slika 3.22. Čitač šalje ulančane blokove a kartica odgovara R blokovima, sve do poslednjeg bloka u nizu
Slika 3.23. Dijagram komunikacije između kartice i čitača pri kojoj je došlo do greške
Slika 3.24. Stablo datoteka prema ISO/IEC 7816-4 standardu
Slika 3.25. Moguće organizacije stabla sa datotekama
Slika 3.26. DF ime u sprezi sa AID-om
Slika 3.27. Strukture elementarnih datoteka
Slika 3.28. Struktura APDU naredbe
Slika 3.29. Struktura APDU-a odgovora
Slika 3.30. Informacija o greški iz statusnih znaka
Slika 3.31. Dijagram statičke autentifikacije podataka
Slika 3.32. Dijagram dinamičke autentifikacije podataka
Slika 3.33. Format 1 polja podataka naredbe u svrhu zaštite inegriteta i autentifikacije
Slika 3.34. Format 2 polja podataka naredbe u svrhu zaštite inegriteta i autentifikacije
Slika 3.35. Format 1 šifrovanih objekata podataka u polju podataka naredbe
Slika 3.36. Format 2 polja podataka naredbe za sigurno slanje poruka
Slika 3.37. Distribucija javnog ključa za autorizaciju sertifikata
Slika 3.38. Distribucija javnog ključa
Slika 3.40. Legenda oznaka dijagrama
Slika 3.41. Dijagram ukidanja ključa
Slika 3.42. Global platform specifikacija
Slika 3.43. Stanja aplikacije prema *Open Platform* arhitekturi

IV deo – JAVA CARD tehnologija (137-168)

Slika 4.1. Java Card virtuelna mašina
Slika 4.2. Konverzija paketa
Slika 4.3. Java Card instaler i van kartični instalacioni program
Slika 4.4. JCRE arhitektura
Slika 4.5. APDU I/O komunikacija
Slika 4.6. Aplikativni identifikator
Slika 4.7. Razvojni proces appleta

V deo – Sigurna autentifikacija sa pametnim karticama (169-192)

Slika 5.1. Komunikacija preko nesigurnog kanala
Slika 5.2. Digitalni potpis i provera potpisa
Slika 5.3. *PEM* format sertifikata sa dodatnim tekstom
Slika 5.4. SSL sloj u TCP/IP mrežnom okruženju
Slika 5.5. SSL protokol rukovanja
Slika 5.6. Osnovna arhitektura infrastrukture javnog ključa
Slika 5.7. X.509 protokol s jednom porukom
Slika 5.8. X.509 protokol sa dve poruke
Slika 5.9. X.509 protokol sa tri poruke

Indeks tabela

I deo – Uvod. Osnovna znanja (18-36)

II deo – Sigurnost pametne kartice (37-67)

III deo – Standardi (69-136)

Tabela 3.1. Kontakti po ISO 7816-2 standardu
Tabela 3.2. Skraćenice i značenje skraćenica
Tabela 3.3. Naponi napajanja po ISO 7816-3
Tabela 3.4. Električne vrednosti I/O signala po ISO 7816-3 (Čitač mora imati mogućnost rada sa obe vrednosti. Visok nivo se osigurava pull-up otpornikom (preporučena vrednost 20kΩ)
Tabela 3.5. Električne vrednosti CLK signala po ISO 7816-3
Tabela 3.6. Električne vrednosti RST signala po ISO 7816-3
Tabela 3.7. Kodiranje formatirajućeg znaka T0
Tabela 3.8. Kodiranje formatirajućeg znaka TDi
Tabela 3.9. Kodiranje TA1
Tabela 3.10. Interpretacija F1
Tabela 3.11. Interpretacija D1
Tabela 3.12. Kodiranje znaka TC1
Tabela 3.13. Elementi podataka PPS-a i njihovo značenje
Tabela 3.14. Kodiranje znaka PPS0
Tabela 3.15. Kodiranje znaka PPS1
Tabela 3.16. Protokoli za prenos podataka premo ISO/IEC 7816-3
Tabela 3.17. Objašnjenje skraćenica T=0 protokola
Tabela 3.18. Vreme potrebno za prenos podataka kod nekih tipičnih naredbi T=0 protokola. (N: naredba, O: odgovor)
Tabela 3.19. Značenja bajtova sw1 i sw2 definisana u dodatku standarda
Tabela 3.20. Vremena prenosa podataka za neke tipične naredbe T=1 protokola, frekvenciju sistemskog sata od 3.5712MHz, vrednošću delitelja od 372, XOR koda za detekciju grešaka, 2 stop bita i 8 bajtova podatka po naredbi (N=naredba,O=odgovor)
Tabela 3.21. Kodiranje polja NAD
Tabela 3.22. Kodiranje polja PCB za I blok
Tabela 3.23. T=1 Stepeni rukovanja greškama
Tabela 3.24. Rezervisane vrednosti identifikatora u najvažnijim standardima kartica
Tabela 3.25. Kodiranje RID-a
Tabela 3.26. Polja u APDU naredbe
Tabela 3.27. Preporučeni minimalni set podataka za generisanje kriptograma
Tabela 3.28. Minimalni set elemenata podataka vezanih za javni ključ za autorizaciju sertifikata koji se spremaju u terminal
Tabela 3.29. Sadržaj sertifikata X.509

IV deo – JAVA CARD tehnologija (137-168)

Tabela 4.1. Podržane i nepodržane osobine Jave
Tabela 4.2. Prikaz algoritama i uloga koju izvršavaju
Tabela 4.3. Skup operacija koji definiše Java 2 SDK Security API
Tabela 4.4. Značajni interfejsi java.security paketa
Tabela 4.5. Značajne funkcije java.security paketa
Tabela 4.6. Interfejs java.security.cert paketa
Tabela 4.7. Klase java.security.cert paketa
Tabela 4.8. Funkcije java.security.cert paketa
Tabela 4.9. Niz algoritama u različitim režimima rada sa više načina dopune
Tabela 4.10. Interfejs za rad sa sertifikacionim putevima
Tabela 4.11. Klase za rad sa sertifikacionim putevima
Tabela 4.12. Funkcije za rad sa sertifikacionim putevima
Tabela 4.13. Interfejs paketa javax.crypto
Tabela 4.14. Značajne klase paketa javax.crypto
Tabela 4.15. Značajne funkcije paketa javax.crypto

V deo – Sigurna autentifikacija sa pametnim karticama (169-192)

Tabela 5.1. Trajanje generisanja ključeva i potpisa kao i provere potpisa za *RSA* i *DSA* algoritme
Tabela 5.2. Sadržaj *X.509* sertifikata
Tabela 5.3. Polja *X.500* prepoznatljivog imena

Deo I

Uvod

Tehnologija danas više nego ikad određuje društvo u kojem živimo. Ulazi u sva područja ljudske delatnosti, stvara novi soj ljudi i poslova, ona je uzrok evolucije društva. Jedan izum objedinjuje i simbolizuje navedene promene, računar.

Prvi računari izvodili su nekoliko računskih operacija, a današnji računari izvode 3D animacije, izvršavajući pritom bilione operacija brzinom svetlosti. Razvoj poluprovodničke industrije omogućio je da se veličina računara sa dimenzijama jedne sobe spusti na površinu od nekoliko kvadratnih milimetara. Time je stvoren prostor za razvoj novih tehnologija. Jedna od njih je i pametna kartica.

Začetnikom pametne kartice smatra se francuski novinar Roland Moreno koji je 1974. registrovao idejni patenat pametne kartice. Zbog toga i ne čudi što je prva primena pametne kartice ostvarena u Francuskoj. Telefonska industrija, odnosno francuski PTT, 1984. izdaje prvu telefonsku karticu. Danas se ona koristi u javnoj telefoniji u više od 50% zemalja širom sveta. Najveće finansijske organizacije, Europay, Mastercard i Visa udružuju svoje napore u cilju primene pametne kartice u finansijskoj industriji. Rezultat je EMV specifikacija izdata 1994. Mobilna telefonija izdaje 1998. GSM specifikaciju koja otvara novo veliko područe primene pametne kartice. Telekomunikaciona i finansijska industrija predstavljaju najveće sisteme kartica, ali one ne određuju esencijalna svojstva pametne kartice, one ih samo koriste. Pametna kartica poseduje neograničene mogućnosti, njene najveće prednosti su veličina, prenosivost, višekratna upotrebljivost, raznolikost funkcionalnosti i sigurnost.

Upravo sigurnost predstavlja najvažnije svojstvo pametne kartice. Današnje društvo evoluira u informaciono društvo. Tehnologija postaje alat u službi informacije, a informacija znanje, moć i novac. Sigurno čuvanje informacija i zaštita protoka informacija postaju nužnost i neophodnost. Kao idealno rešenje javlja se pametna kartica. Pametna kartica čini visok nivo sigurnosti informacije dostupnu svakom, na siguran način čuva informacije i omogućuje da se procesiranje podataka izvodi u sigurnom okruženju.

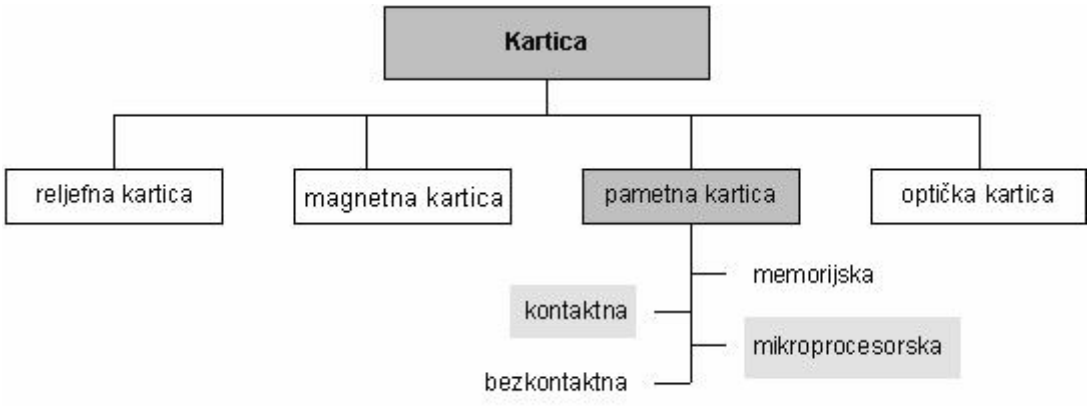
Budući građanin postaće najveći korisnik sistema kartica. Pametna kartica ga identifikuje, čuva njegovo finansijsko i zdrastveno stanje. Potencijalna opasnost prikupljanja informacija na jedan medijum u cilju jednostavnijeg posmatranja pojedinca i upravljanja građanima, korisnicima sistema kartica deluje zastrašujuće. Ali dobro osmišljenom eksploatacijom tehničkih mogućnosti kartica pojedinac zadržava svoju samostalnost i privatnost.

1. Osnovna znanja

1.1. Tipovi kartica

Pametna kartica (engl. *Smart Card*) je najmlađi član porodice identifikacionih kartica. Zbog potrebe da se sa postojećih infrakstruktura kartica ostvari prelaz na novu tehnologiju kartica, često pametne kartice objedinjuju obeležja magnetnih ili reljefnih kartica. Ali bez obzira da li pametna kartica sadrži magnetnu traku ili reljefne podatke njeno esencijalno obeležje je čip.

Pametne kartice se mogu razvrstati u nekoliko grupa. Zavisno od čipu razlikuju se memorijske kartice i mikroprocesorske kartice. Uzimajući u obzir prenos podataka i mehanizam pristupa razlikuju se kontaktne i bezkontaktne kartice.



Slika 1.1. Tipovi kartica

Memorijska kartica ima ugrađen čip s memorijom i neprogramibilnom logikom, ne sadrži mikroprocesor. Usled nepostojanja mikroprocesora, neprogramibilna logika procesira podatke. Ona omogućuje direktan pristup memoriji i podržava nekoliko neprogramibilnih naredbi. Memorija kartice obično čuva do 4K podataka. Tipičan predstavnik memorijskih kartica je telefonska kartica. Posledica neprogramibilnosti je da telefonska kartica postaje neupotrebljiva nakon što se potroši predefinisani kredit. Prednost memorijskih kartica je jednostavna tehnologija, što rezultuje niskom cenom. Sačuvani podaci se štite sigurnosnom logikom i zaštitom memorijskih ćelija. Ali i usled takvog nivoa zaštite, memorijske kartice se mogu relativno lako krivotvoriti.

Kao što i sam naziv govori, mikroprocesorske kartice sadrže mikroprocesor. Mikroprocesor značajno podiže nivo sigurnosti. Omogućava ugradnju šifarskih algoritama i primenu širokog skupa zaštitnih mehanizama. Funkcionalnost mikroprocesorskih kartica ograničena je samo veličinom njihove memorije i snagom procesiranja. Koriste se u aplikacijama koje zahtevaju sigurnost i privatnost podataka,

npr. kontrola pristupa, aplikacije bankovnih sistema, aplikacije bežičnih telekomunikacionih sistema.

Uopšte, naziv pametna kartica odnosi se i na memorijske i na mikroprocesorske kartice, ali zbog "inteligencije" koju pruža ugrađeni procesor često se naziv pametna kartica veže samo za mikroprocesorske kartice. Naziv čip kartica obuhvata i memorijske i mikroprocesorske kartice.

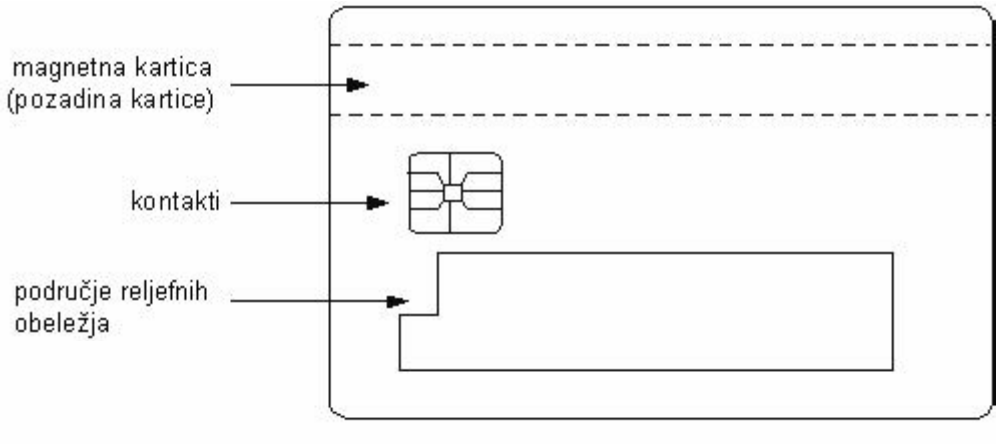
Kontaktne kartice komuniciraju sa spoljnim svetom preko serijskog komunikacionog porta 8-polne priključnice (*slika 1.6*). Priključnica ostvaruje mehanički i električni kontakt sa uređajem za prihvatanje kartice – CAD (engl. *Card Acceptance Device*) uređaj (npr. CAD uređaj može biti čitač kartice). Kartica ne poseduje sopstveni izvor napajanja i nema mogućnost generisanja unutrašnjeg signalnog takta.

Zbog toga što kontaktne kartice zahtevaju tačno definisani položaj i orijentaciju kartice prilikom stavljanja u CAD uređaj, u situacijama gde su potrebne brze transakcije upotrebljavaju se bezkontaktne kartice. Sistem javnog transporta je idealan za primenu bezkontaktnih kartica. Bezkontaktne kartice vrše komunikaciju sa spoljnim svetom preko antene ugrađene u telo kartice. Napajanje se izvodi internom baterijom ili elektromagnetnom indukcijom kroz ugrađenu atenu. Podaci se do CAD uređaja prenose elektromagnetnim poljem. Visoko frekventno elektromagnetno polje ima dvostruku ulogu, ulogu prenosa energije elektromagnetnom indukcijom i ulogu prenosa podataka faznom ili amplitudskom modulacijom.

Prednosti bezkontaktnih kartica su u tome što je čip u potpunosti integrisan u telo kartice i ne ostvaruje mehanički kontakt sa okolinom. To znači da ne postoji mogućnost naponskog i strujnog udara i da nema trošenja kontaktnih površina učestalim korišćenjem kartice. Time se produžuje životni vek kartice. Robusnost bezkontaktnih kartica i robusnost odgovarajućih CAD uređaja čini takve sisteme kartica idealnim za industrijska okruženja.

Bezkontaktne kartice imaju i svojih nedostatka. Da bi se mogli razmenjivati podaci sa CAD uređajem moraju se nalaziti unutar određene razdaljine. Zbog mogućeg kratkog vremena transakcije prenosi se samo ograničena količina podataka. Povećana je osetljivost na savijanja. Moguća su oštećenja kartice usled utiskivanja reljefnih obeležja. Postoji potencijalna opasnost da se bez znanja nosioca kartice izvede transakcija sa karticom ili da se presretnu poslati podaci. Možda i trenutno najvažniji nedostatak je što su bezkontaktne kartice skuplje od kontaktnih kartica.

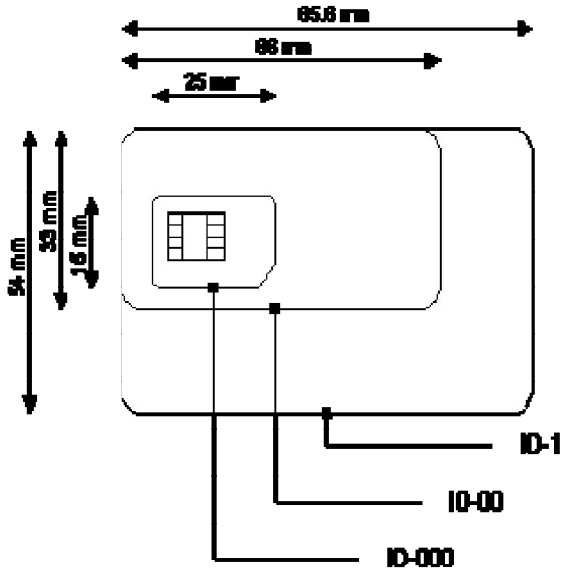
Standardi vezani uz bezkontaktne kartice najvećim delom su još u izradi i to je dodatni razlog češće implementacije kontaktnih kartica unutar sistema kartica. Ta činjenica usmerava dalja razmatranja primarno na kontaktne kartice, iako je većina primenjiva i na bezkontaktne kartice.



Slika 1.2. Fizički izgled pametne kartice

1.2. Električna i fizička svojstva pametne kartice

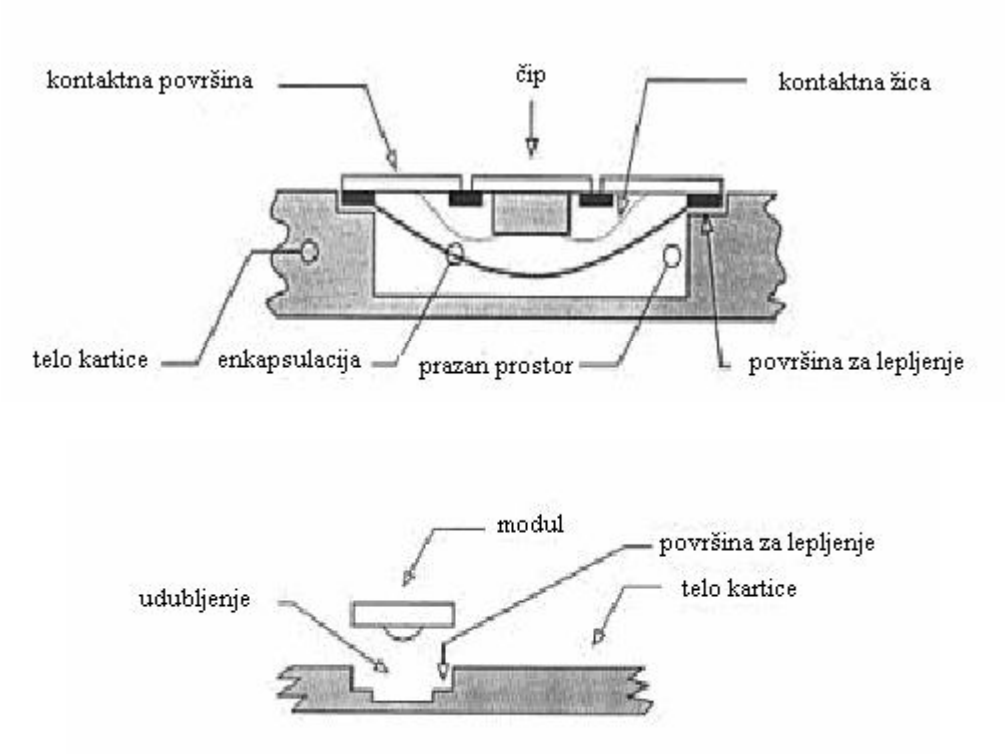
1.2.1. Formati i dimenzije



Slika 1.3. Formati i pripadajuće dimenzije pametnih kartica

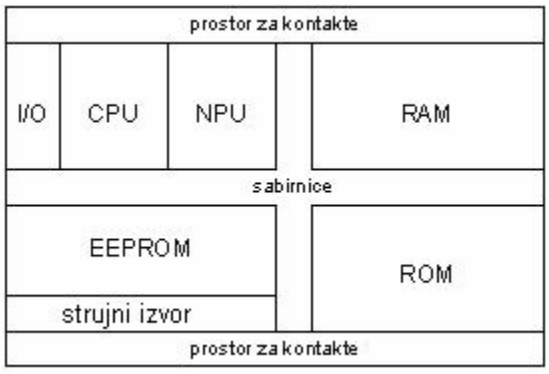
1.2.2. Čip

Čip je najvažniji element pametne kartice. Za razliku od magnetne trake čip je krhk i lomljiv i ne može se jednostavno nalepiti na površinu kartice. Umesto toga čip se stavlja u neku vrstu kućišta da se zaštiti od spoljnih uticaja svakodnevnog korišćenja kartice. To kućište se naziva *modul čipa*. Struktura modula i proces umetanja u telo kartice prikazani su na *slici 1.3*.



Slika 1.4. Struktura modula i proces umetanja u telo kartice

Deo površine modula služi kao 8-polna kontaktna površina preko koje pametna kartica, odnosno njen mikrokontroler komunicira sa spoljnim svetom. Tipična struktura mikrokontrolera pametne kartice prikazana je na *slici 1.5*. Funkcionalno on u potpunosti predstavlja mali računar.



Slika 1.5. Šematski prikaz mikrokontrolera pametne kartice

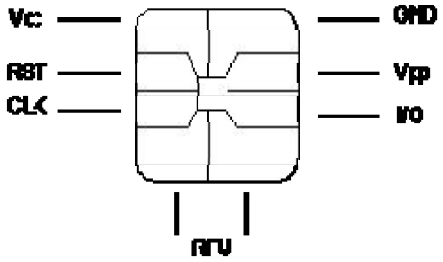
Mikrokontroler pametne kartice dizajnira se i razvija specijalno za tu svrhu. Poluprovodničke komponente koje se pritom koriste nisu standardne, masovno dostupne poluprovodničke komponente. To proizilazi iz nekoliko razloga. Za sada se čipovi rade od silicijuma koji je neelastičan i lomljiv. Da se izbegne oštećenje čipa prilikom savijanja

kartice, mikrokontroler je ograničen na veličinu od 25 mm^2 i prati oblik četvorougla. To zahteva optimizaciju komponenti mikrokontrolera s obzirom na njihovu veličinu, položaj i povezivanje sa drugim komponentama. Osim toga na mikrokontroleru je neophodna prisutnost aktivnih i pasivnih zaštitnih elemenata što dodatno komplikuje dizajn mikrokontrolera. Sigurnosni princip nedostupnosti korišćenih komponenti dodatno otežava analizu i falsifikovanje mikrokontrolera kartica.

1.2.3. Kontaktna površina

Pametna kartica ima osam kontaktnih tačaka (*slika 1.6*). Dimenzije i lokacije kontakata opisuje standard ISO 7816, 2. deo.

- Na V_{cc} kontakt se dovodi napajanje. Naponski nivo je 3V (Volti) ili 5V, sa maksimalnom devijacijom od 10%. Pametne kartice mobilnih telefona, SIM moduli, imaju napajanje od 3V.
- RST kontakt služi za slanje reset signala mikroprocesoru, to je tzv. *topli reset*. *Hladni reset* se dobija prekidanjem signala napajanja. Izvlačenje i ponovno umetanje kartice u CAD uređaj rezultuje hladnim reset-om.
- Mikroprocesor pametne kartice ne poseduje mogućnost generisanja signala takta. Na CLK kontakt se dovodi spoljeni signal takta iz koga se derivacijom dobija unutrašnji takt.
- GND kontakt se koristi kao referentna naponska vrednost, najčešće je to nulti naponski nivo.
- V_{pp} kontakt je opcionalan i koristi se samo kod starih tipova kartica. Takvi tipovi kartica zahtevali su korišćenje dva programska naponska nivoa. Niži naponski nivo označavao je pasivno stanje, a viši naponski nivo se koristio prilikom pisanja u EEPROM memoriju. Današnje pametne kartice ne koriste taj kontakt jer moderni mikrokontroleri koriste integrisani strujni izvor.
- I/O kontakt se koristi za prenos podataka i naredbi između pametne kartice i spoljnog sveta. Komunikacija je dvosmerna u iz jednog od oba pravca (naizmenični prenos u oba smera, ali ne istovremeno).
- RFU kontakti su rezervisani za buduća proširenja.



Slika 1.6. Kontaktna površina pametne kartice

1.2.4. CPU

Centralna procesorska jedinica (**CPU** *–engl. Cental processor unit*) kod većine današnjih čipova kartica je 8-bitni procesor baziran na CISC (engl. *Complex Instruction Set Computing*) arhitekturi procesora Motorola 6805 ili Intel 8051 sa signalom takta frekvencije 5MHz i 16-bitnim memorijskim adresiranjem. Snažnije kartice implementiraju množitelje takta (2x, 4x, 8x). Time je moguće podići radnu frekvenciju do 40MHz.

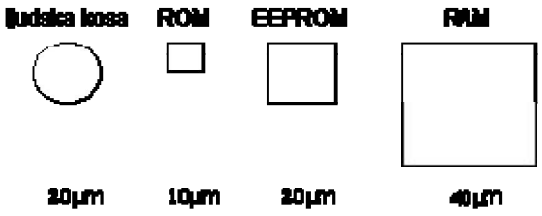
Noviji čipovi kartica implementiraju 16-bitne i 32-bitne procesore. Razvoj 32-bitnih procesora sa 20MHz, 3V i 40mA (kompatibilnost sa PDA (engl. *Personal Digital Assistant*) uređajima) proizašao je iz zahteva za sve složenijom primenom aplikacija na karticama. Ali to neće u potpunosti izbaciti 8-bitni procesore. Njihova cena, mala energetska potrošnja i mala površina čipa biće temeljni razlozi buduće proizvodnje takvih jeftinih čipova kartica.

1.2.5. Koprocesor

Aplikacije koje koriste pametne kartice često zahtevaju primenu šifarskih algoritama. Ograničena procesna moć CPU jedinice usko je grlo zahtevnih šifarskih operacija. Šifarski koprocesor je specijalno dizajniran za korišćenje računskih operacija asimetričnih šifarskih algoritama, implementira operacije eksponenta i modula velikih brojeva na nivou poluprovodničkih elemenata. Time je omogućena i generacija kartica 1024-bitnih RSA ključeva. Negativna strana ugradnje dodatne NPU jedinice je povećanje cene čipa.

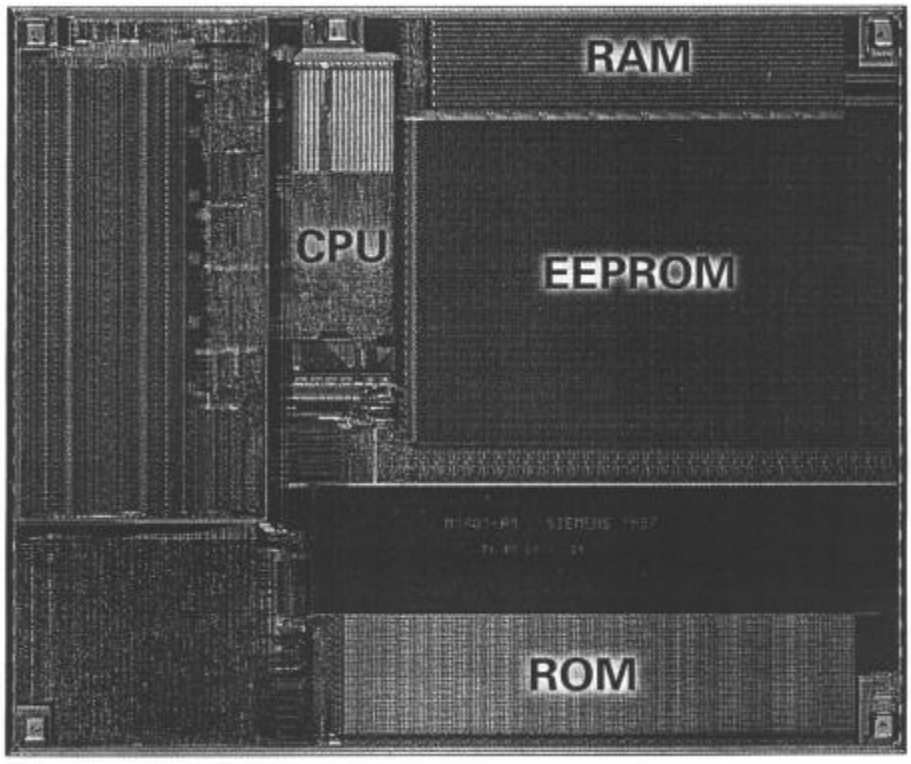
1.2.6. Memorija

Pametne kartice obično sadrže tri tipa memorije: perzistentnu (stalnu) ispisnu memoriju, perzistentnu upisno-ispisnu memoriju i neperzistentnu upisno-ispisnu memoriju. ROM (engl. *Read-Only Memory*), EEPROM (engl. *Electrical Erasable Programmable Read-Only Memory*) i RAM (engl. *Random Access Memory*) su najčešće korišćene memorije. Korišteni promer pojedinačnih tipova memorije zavisi od područija primene pametne kartice. Ali, s obzirom da je RAM ćelija otprilike 16 puta veća od ROM ćelije, zbog ograničene površine čipa, uvek je prisutan zahtev za što manjom količinom RAM memorije.



Slika 1.7. Fizičke veličine memorijskih ćelija

- ***ROM memorija*** se koristi za čuvanje nepromenjivog programskog kôda. Sadržava kôd operativnog sistema i nepromenljive podatke korisničkih aplikacija. Proces pisanja stalnih programskih komponenti u ispisnu neprogramrljivu memoriju čipa naziva se maskiranje i dešava se u fazi proizvodnje kartice. Nakon te faze nemoguće je upisivanje podataka, zapisani podaci se mogu samo čitati. ROM memorija zadržava svoje stanje i bez prisutnosti napajanja.
- EEPROM memorija takođe, kao i ROM, zadržava svoje stanje kad nema napajanja. Sadržaj memorije se može menjati za vreme normalnog korišćenja kartice. Zbog toga se koristi za čuvanje podataka i korisničkih aplikacija, pa funkcionalno odgovara tvrdom disku personalnog računara (PC). Važni električni parametri EEPROM memorije su broj ciklusa pisanja, vreme zadržavanja podataka i vreme pristupa. EEPROM koji koriste pametne katice može izvesti barem 100 000 ciklusa pisanja i zadržati podatke duže od 10 godina. Čitanje EEPROM-a je jednako brzo kao i čitanje RAM-a, ali je zato pisanje u EEPROM 1000 puta sporije od pisanja u RAM.
- RAM memorija se koristi kao privremeni radni prostor za čuvanje i modifikaciju podataka. Uklanjanjem napajanja gubi se informacija o sadržaju memorije.



Slika 1.8. Mikrokontroler pametne kartice površine 21 mm2, 32kB ROM-a, 16kB EEPROM-a, 1280 bajtova RAM-a. Dva neoznačena područija sa leve strane čine numerički koprocesor i skup perifernih komponenti. Na krajevima se vidi 5 spojnih mesta za ostvarivanje kontakata sa modulom čipa

1.3. Komunikacioni model

1.3.1. CAD uređaj i aplikacija terminala

Pametna kartica se ubacuje u uređaj za prihvatanje kartice – CAD (engl. *Card Acceptance Device*) uređaj. Dve su vrste CAD uređaja: *čitač kartce* (engl. *Card Reader*) i *terminal*.

Pametna kartica komunicira sa računarom putem čitača kartica. Čitač je povezan sa računarom preko serijskog, paralelnog ili USB porta. On najčešće vrši detekciju grešaka nastalu usled korišćenja neodgovarajućeg transportnog protokola i nema sposobnost procesiranja podataka. Može integrisati tastaturu za direktni unos podataka.

Terminali su računari koja imaju čitač kartica integrisan kao vlastitu komponentu. Primer terminala je bankovni ATM (engl. *Automatic Teller Machine*) terminal. Osim funkcionalnosti čitača, terminali poseduju mogućnost procesiranja podataka. Npr. ATM terminal može umanjiti količinu kredita na kartici. Poseduju vizuelnu zaštitu (kameru instaliranu pored terminala) i sigurnosnu tastaturu, tzv. PIN podlogu.

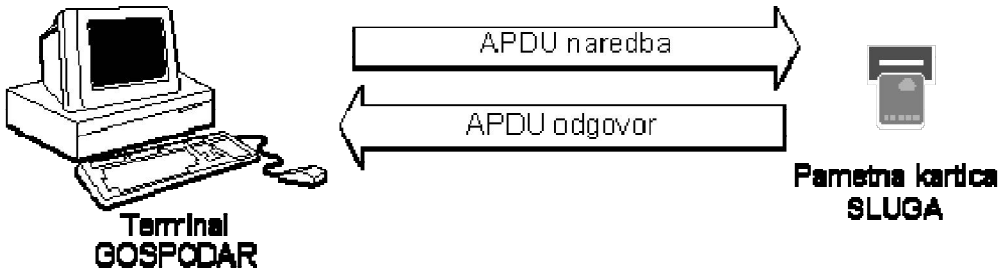
Pojmovi terminal i čitač kartica često se koriste naizmenično, zavisno od primera nad kojima se sprovode razmatranja. Funkcionalno se terminal i personalni računar (PC) sa čitačem kartice ne razlikuju, osim što je terminal robusniji i specijalizovaniji. Prema tome zamena pojmova računar i terminal najčešće ne utiče na sprovedena razmatranja.

Aplikacije koje komuniciraju sa pametnom karticom, nezavisno od toga da li se nalaze na terminalu ili PC-u, nazivaju se aplikacije terminala. Aplikacija terminala upravlja procesom komunikacije.

1.3.2. Komunikacioni model sistema sa karticama

Komunikacioni kanal između pametne kartice i glavnog računara podržava obosmernu komunikaciju, što znači prenos podataka u oba smera, ali ne u isto vreme.

Dva računara međusobno komuniciraju razmenjujući pakete. Počka sa podacima paketi se konstruišu na temelju nekog protokola, kao što je TCP/IP. Na sličan način i pametna kartica razgovara sa terminalom, pritom koristeći vlastiti format paketa sa podacima – APDU (engl. *Application Protocol Data Units*) . APDU nosi naredbu ili odgovor, odnosno razlikujemo APDU naredbu ili APDU odgovor. Sistem sa karticama temelji se na gospodar-sluga (engl. *Master-Slave*) komunikacionom modelu. Pametna kartica uvek preuzima pasivnu ulogu sluge, čeka od terminala APDU naredbu i šalje odgovarajući APDU odgovor. U komunikacionom kanalu razmenjuju se APDU naredbe terminala i APDU odgovori pametne kartice.



Slika 1.9. Komunikacioni model pametne kartice

1.3.3. APDU protokol

APDU protokol, specifikovan u ISO 7816-4, je protokol na nivou aplikacija između pametne kartice i terminala. APDU poruke definisane u ISO 7816-4 obuhvaćene su sa dve strukture: prvu strukturu koristi aplikacija na strani terminala za slanje naredbi kartici, a drugu kartica za slanje povratnih odgovora aplikaciji terminala. Dotični formati poruka se nazivaju APDU naredba (C-APDU) i APDU odgovor (R-APDU). Njihova struktura je prikazana na *slikama 1.10. i 1.11.*

zaglavlje				telo		
CLA	INS	P1	P2	Lc	Polje sa podacima	Le

Slika 1.10. Struktura APDU naredbe

telo	statusna reč	
Polje sa podacima	SW1	SW2

Slika 1.11. Struktura APDU odgovora

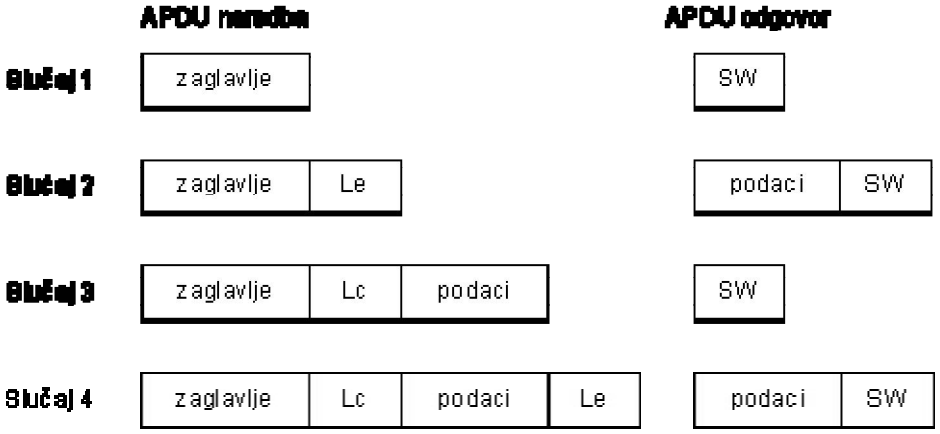
Zaglavlje APDU naredbe se sastoji od 4 bajta: CLA (oznaka klase kojoj pripada instrukcija), INS (oznaka instrukcije), P1 i P2 (parametri instrukcije). Bajt klase identifikuje kategoriju APDU naredbe i APDU odgovora. Instrukcijski bajt određuje instrukciju naredbe. Parametri P1 i P2 se koriste kao proširenja funkcionalnosti instrukcije.

Nakon zaglavlja APDU naredbe sledi telo APDU naredbe. Telo je opcionalno i varijabilne dužine. Lc polje određuje dužinu polja sa podacima (broj bajtova). Polje sa podacima sadrži podatke koji se šalju kartici za izvršavanje instrukcije specifikovane u zaglavlju. Poslednji bajt APDU naredbe je Le polje. Le polje određuje koliki broj bajtova očekuje aplikacija terminala u sledećem APDU odgovoru.

Kartica šalje APDU odgovor koji se sastoji od opcionalnog tela i obaveznog završnog dela. Telo obuhvata polje sa podacima čija je dužina određena Le poljem odgovarajuće APDU naredbe. Završni deo se sastoji od dva polja, SW1 i SW2, koja zajedno čine statusnu riječ. Statusna reč opisuje u kojem je radnom stanju kartica nakon

što izvrši instrukciju APDU naredbe. Npr. statusna reč "0x9000" znači da je kartica uspešno i u potpunosti izvršila naredbu.

Iz činjenice da C-APDU i R-APDU uvek dolaze u paru i da je polje sa podacima opcionalno u obe APDU strukture, proizlaze četiri moguća slučaja APDU komunikacije *(slika 1.12).*



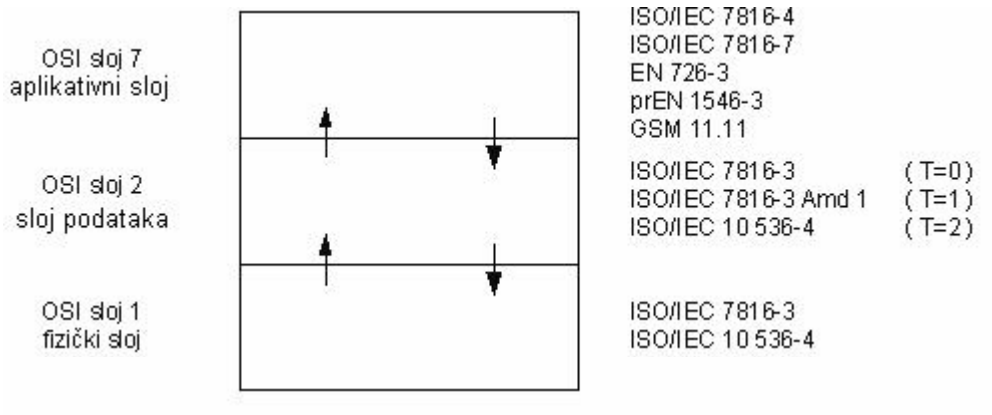
Slika 1.12. Mogući slučajevi APDU komunikacije

1.3.4. TPDU protokol

TPDU (engl. ***T****ransmission* ***P****rotocol* ***D****ata* ***U****nit*) protokol je protokol nižeg nivoa – transportni protokol definisan u ISO 7816-3 standardu. Strukture podataka koje izmenjuju terminal i kartice nazivaju se TPDU jedinice (engl. ***U****nits*). Sada se može povući paralela sa ISO OSI referentnim modelom gde APDU protokol predstavlja OSI sloj 7, TPDU protokol OSI sloj 2. Ispod njega se nalazi fizički sloj, žice i porta kojim putuju bitovi. Razdvajanjem komunikacionog modela sistema sa karticama na tri sloja, želja je postići nezavisnost aplikativnog protokola od fizičkog načina prenosa podataka. Ponašanje i interakcija navedenih slojeva definisana je kroz nekoliko međunarodnih standarda (*slika 1.13*).

Trenutno su u sistemima sa karticama najraširenija dva tipa transportnih protokola, T=0 protokol i T=1 protokol. Oba protokola su asinhrona i obosmerna. T=0 protokol je bajt-orijentisan što znači da je najmanja prenesena jedinica bajt. Nasuprot tome, T=1 protokol je blok-orijentisan, najmanja prenesena jedinica je blok, odnosno niz bajtova. T=0 protokol se smatra zastarelim zbog nepostojanja jasnog odvajanja gore navedenih komunikacionih slojeva. T=1 protokol jasno razdvaja slojeve ukalupljujući APDU poruke u TPDU jedinice. Time se omogućuje ulančavanje blokova i šifrovanje sadržaja APDU poruka. Fizički sloj za prenos podataka dobija svu potrebnu informaciju od TPDU zaglavlja, a transportni sloj prosleđuje šifrovani APDU aplikacionom sloju. Sada se na nivou operativnog sistema, odnosno aplikativnog sloja vrši dešifrovanje i tumačenje APDU poruke.

Dvosmerni transportni protokol T=2 trenutno je u fazi izrade. Temelji se na T=1 protokolu.



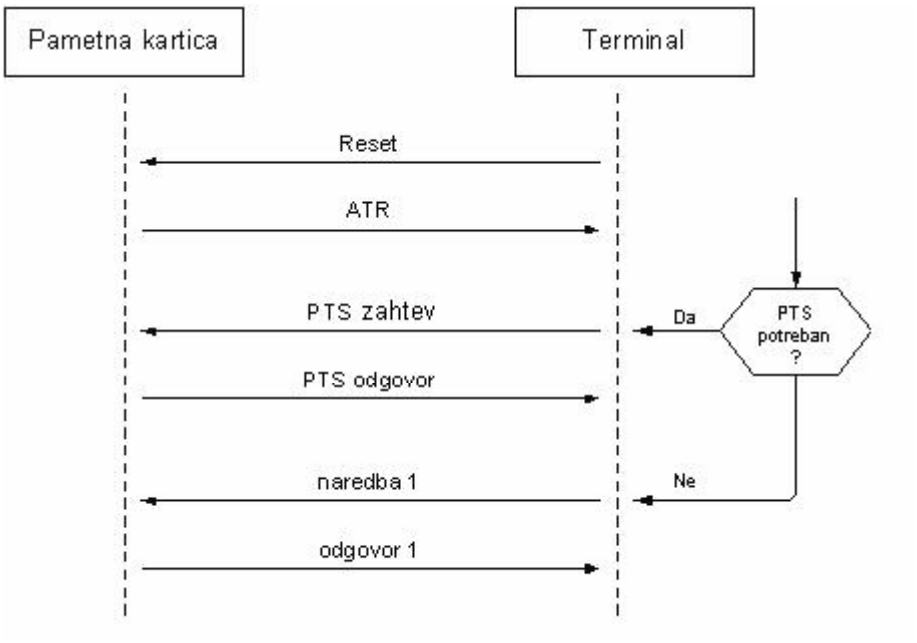
Slika 1.13. OSI komunikacioni model između terminala i pametne kartice

1.3.5. ATR

Nakon što se na kontaktnu površinu pametne kartice dovede signal napajanja, signal takta i reset signal, kartica preko I/O linije šalje terminalu ATR (engl. ***A**nswer **T**o **R**eset*) poruku. ATR porukom saopštava terminalu vrednosti parametara potrebnih za uspostavljanje međusobne komunikacije. Dužina ATR poruke je maksimalno 33 bajta. Sadrži informacije o podržanom tipu prenosnog protokola, o brzini prenosa podataka, parametre hardvera kartice i druge slične informacije koje terminal mora znati da bi mogao uspostaviti pravilnu komunikaciju sa datom karticom.

1.3.6. PTS

Prijemom ATR poruke terminal dobija sve potrebne informacije za uspostavljanje komunikacionog puta između njega i pametne kartice. Ako terminal želi promeniti jedan ili više parametara komunikacionog protokola, šalje kartici PTS poruku. PTS (engl. ***P**rotocol **T**ype **S**election*) poruka se mora poslati neposredno nakon ATR poruke. Nakon ATR poruke moguće je poslati samo jednu PTS poruku. Terminal na taj način može promeniti parametre komunikacionog protokola, naravno, samo one koje kartica dozvoljava.



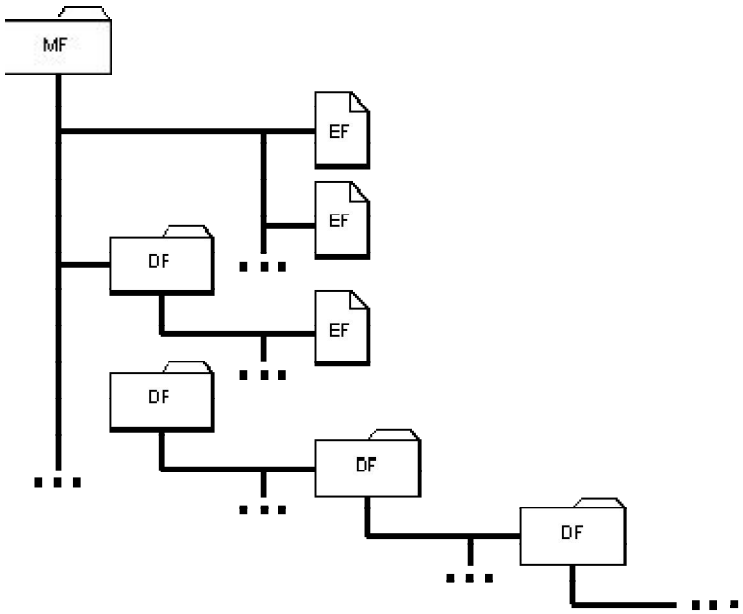
Slika 1.14. Inicijalni prenos podataka između terminala i pametne kartice

1.4. Operativni sistem kartice

Operativni sistemi pametnih kartica imaju malu sličnost sa operativnim sistemima personalnih računara (UNIX, DOS). Između ostalog ne uključuju korisnički interfejs i nemaju mogućnost pristupa spoljnim jedinicama za čuvanje podataka. Njihova osnovna uloga obuhvata prenos podataka, kontrolu izvršavanja naredbi, upravljanje datotekama, upravljanje i izvršavanje šifarskih algoritama. Optimizuju sigurnost izvršavanja programa na kartici i zaštitu kontrole pristupa podacima. Pružaju standardizovani skup instrukcija za izgradnju korisničkih aplikacija.

Standard ISO 7816-4 predstavlja temelj većine današnjih operativnih sistema pametnih kartica. Njegova najbitnija karakteristika je orijentacija datoteka. Kao posledica toga korisnička aplikacija predstavlja se datotekom sa podacima, a naredbe za pristup aplikactivnim podacima definiše operativni sistem. Upravo zbog te karakteristike, odnosno nedovoljno jasnog razdvajanja operativnog sistema i korisničke aplikacije, razvila se potreba za novim tehnologijama. Jedna od tih tehnologija je i Java Card tehnologija.

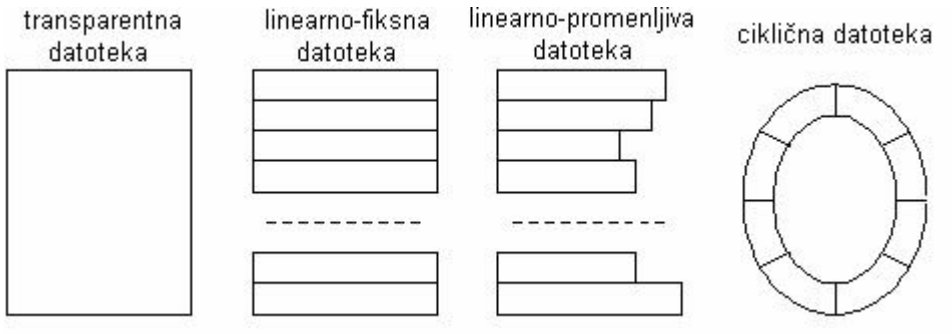
Sistem sa datotekama kartice temeljen na ISO 7816-4 ima hijerarhijsku strukturu prikazanu *slikom 1.15*. ISO 7816-4 definiše tri tipa datoteka. Najviši nivo se zove *glavna datoteka – MF (*engl. ***M****aster* ***F****ile)*, ispod nje je nekoliko slojeva *namenskih datoteka – DF (*engl. ***D****edicated* ***F****ile)*, i konačno sloj *elementarnih datoteka – EF (*engl. ***E****lementary* ***F****ile)*. Svaka datoteka je određena svojim 2-bajtnim identifikatorom – FID-om *(*engl. ***F****ile* ***I****dentifier)*. Pomoću FID-a se jednoznačno selektuje datoteka. Svaka naredba nad datotekom zateva prethodnu selekciju datoteke.



Slika 1.15. Sistem dadoteka na kartici

Unutrašnju strukturu svake datoteke čini zaglavlje i telo datoteke. Zaglavlje sadrži dozvole pristupa i informacije o strukturi tela datoteke. Glavna datoteka i namenska datoteka u svom telu sadrže zaglavlja svih namenskih i elementarnih datoteka njima neposredno podređenih unutar hijerarhijske strukture datoteka. Elementarna datoteka je datoteka podataka, njeno telo nosi podatke. Pristup do elementarne datoteke, odnosno do njenih podataka, mora ići kroz datoteke roditelja. Na taj način formira se logički kanal do elementarne datoteke.

S obzirom na strukturu tela elementarne datoteke razlikuju se četiri vrste EF datoteka (*slika 1.16.*). Strukturu EF datoteke ne određuje korisnička aplikacija već operativni sistem kartice.



Slika 1.16. Strukture elementarnih datoteka

S obzirom na to ko koristi sačuvane podatake razlikuju se radne i interne elementarne datoteke. Radni EF čuva podatke namenjene spoljnom svetu, razmenjuju se

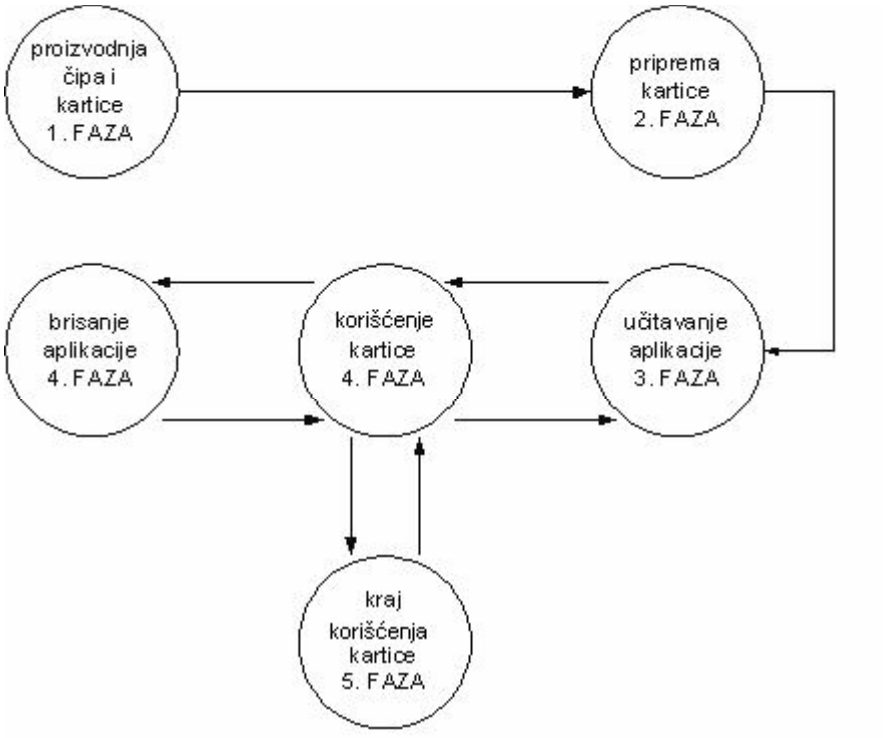
u komunikaciji sa terminalom. Podatke internog EF-a isključivo koristi operativni sistem kartice (tajni ključevi, PIN). U bilo kom slučaju datoteka mora biti selektovana pre izvođenja operacije nad njom. To je ekvivalentno otvaranju datoteke.

Sumirajući strukturu datoteka i kontrolu pristupa podacima, podaci na kartici mogu biti zaštićeni individualno, podešavajući uslove pristupa u zaglavlju svake datoteke, i hijerarhijski, grupišući datoteke ispod jedne namenske datoteke sa postavljenim uslovima pristupa.

1.5. Životni ciklus kartice

Standard ISO 10202-1 razlikuje 5 glavnih faza u životnom ciklusu kartice. Prelazi između faza su tačno specifikovani. Svi proizvodni procesi i transportni putevi između različitih firmi moraju biti osigurani fizičkim ili šifarskim načinom zaštite. Sledi kratki prikaz tipičnih aktivnosti u pojedinim životnim fazama:

Faze životnog ciklusa kartice	**Tipične aktivnosti**
Faza 1: Proizvodnja čipa i proizvodnja kartice	· Dizajn čipa
	· Generacija operativnog sistema kartice
	· Proizvodnja čipa i modula
	· Proizvodnja tela kartice
	· Ugrađivanje modula u telo kartice
Faza 2: Priprema kartice	· Završavanje operativnog sistema kartice
Faza 3: Priprema aplikacije	· Inicijalizacija aplikacije/a
	· Personalizacija (individualizacija) aplikacije/a – električni i optički
Faza 4: Korišćenje kartice	· Aktiviranje aplikacije
	· Deaktiviranje aplikacije
Faza 5: Kraj korišćenja kartice	· Deaktiviranje aplikacije
	· Deaktiviranje kartice



Slika 1.17. Životni ciklus pametne kartice prema ISO 10202-1

1.6. Sistem kartice

Sistem kartice je distribuisani sistem. Sastoji se od dva dela. Prvi deo je pozadinski sistem koji se oslanja na računare, čitače i terminale. Drugi deo sistema kartice predstavljaju pametne kartice i programi na kartici.

Većina programske podrške izvodi se na strani pozadinskog sistema, uključujući sistemsku programsku podršku i korisničke aplikacije. Programska podrška pozadinskog sistema prepoznaje specifičnu karticu i upravlja komunikacijom između korisničke aplikacije i date kartice. Takođe ona servisira infrakstrukturu sistema kartice. Izdaje i personalizuje kartice, održava sigurnost sistema, upravlja ključevima.

Korisnička aplikacija se naziva aplikacija terminala. Najčešće je pisana u programskom jeziku višeg nivoa (Java, C, C++). Ona implementira funkcije koje sarađuju sa određenom karticom ili aplikacijom na kartici. Npr. bankovna ATM aplikacija implementira funkcije autentifikacije korisnika i procesiranja transakcija, pruža korisnički interfejs koji omogućuje jednostavno i intuitivno korišćenje bankovnih usluga. Sa pojavom novih arhitektura kartica, zasnovanih na operativnim sistemima koji dopuštaju učitavanje koda u korisničkoj fazi kartice (Java Card, Multos), svaki umreženi personalni računar sa čitačem kartica pretvara se u terminal pozadinskog sistema, a Java applet u aplikaciju terminala.

Programi na kartici se definišu kao programi koji se u potpunosti izvode na kartici. To uključuje sistemske programe i korisničke aplikacije na kartici. Sistemski programi na kartici obuhvataju operativni sistem kartice i uslužne servise kartice koji upravljaju memorijom, I/O komunikacijom, strukturom datoteka, osiguravaju integritet i sigurnost podataka na kartici. Aplikacije na kartici sadrže podatke i funkcije za procesiranje podataka. Npr. kartična aplikacija e-novčanika čuva iznos računa i implementira funkcije za ažuriranje računa. Zavisno od složenosti primene sistema kartice, prisutnost aplikacije na kartici je nepotrebna ako se sva funkcionalnost sistema kartice može realizovati pozivima instrukcija podržanih od operativnog sistema kartice. Programi na kartici implementiraju se u mašinskom jeziku mikroprocesora kartice. Novije arhitekture kartica omogućuju implementaciju aplikacija kartice u programskim jezicima višeg nivoa.

Razvoj sistema kartice zahteva infrakstrukturu i odgovarajuću programsku podršku pozadinskog sistema i njegovu implementaciju i integraciju sa programima kartice. Time su obuhvaćeni proizvođači operativnih sistema kartice, proizvođači terminala, razvojni programeri aplikacija (i kartice i terminala), izdavači kartica. Sve te strane teško je pronaći u jednoj organizaciji. To je rezultovalo da iz nužnosti međusobne saradnje proizađu standardi i udruženja kao što su Open Platform i OpenCard Framework. Stvoren je okvir za razvoj i operativnu upotrebu sistema kartica temeljenih na komponentama različitih dobavljača.

1.7. Standardi i specifikacije

Standardi će više biti obrađeni u trećeg poglavlju. Ovde su navedene neke od vrsta standarda i njihove glavne komponente koje pokrivaju delimičnu ili potpunu upotrebljivost (unificiranost) pametnih kartica u svetu.

1.7.1. ISO 7816 standard

***I**nternational **O**rganization for **S**tandardization* izdala je standard ISO 7816 *«Identification cards – Integrated circuit cards with contacts»*. On definiše najvažnije karakteristike kontaktnih čip kartica:

1. deo – fizičke karakterisitike,
2. deo – dimenzije i položaj kontakata,
3. deo – električni signali i transportni protokoli,
4. deo – međuindustrijske naredbe,
5. deo – identifikator aplikacije,
6. deo – međuindustrijski elementi podataka,
7. deo – međuindustrijske SCQL naredbe.

1.7.2. GSM

***E**uropean **T**elecommunications **S**tandards **I**nstitute (ETSI)* je izdao standard koji pokriva korišćenje pametnih kartica u javnim i mobilnim telefonskim sistemima. GSM standardi su specifikacija međunarodnog zemaljskog mobilnog telefonskog sistema. Značajni su :

GSM 11.11 – specifikacija SIM modula,
GSM 11.14 – specifikacija alata za razvoj aplikacija SIM modula,
GSM 03.48 – sigurnosni mehanizmi alata za razvoj aplikacija SIM modula,
GSM 03.19 – SIM API za Java Card platformu. SIM API je proširenje Java Card 2.1 API.

1.7.3. EMV

***E**uropay, **M**asterCard* i ***V**isa* definisali su EMV specifikaciju. Ona je nastala kao proširenje ISO 7816 standarda u smeru specifičnih potreba finansijske industrije. Zadnja verzija specifikacije, EMV 96 verzija 3.3.1, izdata je u maju 1998 i sastoji se iz tri dela:

EMV ´96 Integrated Circuit Card specifikacija,
EMV ´96 Integrated Circuit Card Terminal specifikacija,
EMV ´96 Integrated Circuit Card Application specifikacija.

1.7.4. Open Platform

Open Platform standard definiše integrisano okruženje za razvoj i operativnu upotrebu višenamenskih sistema kartice. Open Platform standard sadrži specifikaciju kartice i specifikaciju terminala. Specifikacija kartice definiše komunikaciju kartice sa terminalom i management aplikacije kartice. Specifikacija terminala definiše arhitekturu aplikacije terminala. Terminal mora zadovoljiti ISO i EMV standarde.

Open Platform specifikacije u početku je razvijala Visa. Danas su one deo *Global Platform* organizacije.

1.7.5. OpenCard Framework

***O**pen**C**ard **F**ramework (OCF)*, inicijalno je izašao iz IBM-a, vlasništvo je OpenCard konzorcijuma. OpenCard konzorcijum obuhvata sve važnije učesnike industrije kartica. OCF definiše standardni interfejs za pisanje aplikacija terminala, interfejs za komunikaciju sa čitačima i aplikacijama kartica. Arhitektura OCF modela deli funkcionalnost sistema kartica između proizvođača terminala kartica, proizvođača operativnih sistema kartica i izdavača kartica. Cilj je postići međusobnu nezavisnost navedenih strana.

OCF je implementiran u programskom jeziku Java zbog želje da se pametna kartica postavi kao deo globalne mrežne arhitekture.

1.7.6. PC/SC

PC/SC specifikacija vlasništvo je *PC/SC WorkGroup*, industrijskog konzorcijuma koji obuhvata sve važnije učesnike industrije kartica. PC/SC definiše arhitekturu namenjenu korišćenju pametnih kartica u domenu personalnih računara.

U toj arhitekturi aplikacija terminala se gradi iznad PC/SC servisa i PC/SC managera. Servisi enkapsuliraju funkcionalnosti i specifičnosti pojedinih pametnih kartica i omogućuju da se kartici pristupi kroz programerski interfejs višeg nivoa, a PC/SC manager povezuje operativni sistem personalnog računara sa CAD uređajem. Njih najčešće osigurava proizvođač kartica.

OCF i PC/SC poseduju slične koncepte. Na Windows platformama OCF pristupa CAD uređaju kroz instalirani PC/SC manager.

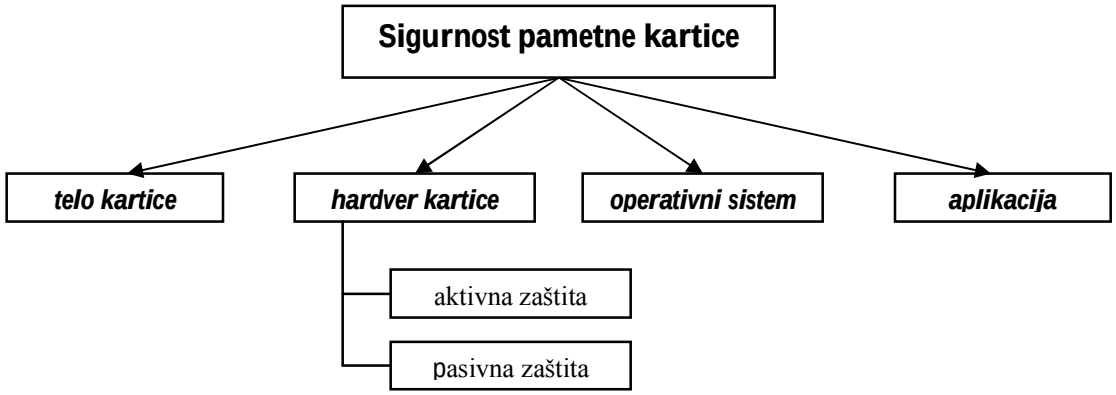
Deo II

2. Sigurnost pametne kartice

Pametna kartica pruža podacima i programima sigurno okruženje. Kad količina napora da se pročitaju podaci sa pametne kartice ne bi bila tako velika, kartica bi bila ništa više nego disketa sa drugačijom ulogom.

Praktično je nemoguće sagraditi potpuno siguran sistem, pa čak i potpuno sigurnu pametnu karticu. Ako napor utrošen na napad dostigne dovoljno visok nivo, moguće je prodreti u bilo koji sistem i na njemu izvršiti neovlašćene promene. Svaki potencijalni napadač svesno ili nesvesno procenjuje dobit koju mu donosi uspešan napad. Vreme, novac i napor uloženi na provalu u sistem moraju biti vredni dobijene nagrade. Ako nagrada nije dostojna uloženog napora do napada nikad neće doći.

Sigurnost pametne kartice obuhvata četiri komponente. Prva komponenta je telo kartice. Telo kartice implementira mnoštvo sigurnosnih obeležja, ne samo mašinski-čitljivih i time podvrgnutih kontroli uređaja, već i vizuelno-čitljivih i time podvrgnutih kontroli čoveka. Npr. to su holografi, reljefna obeležja, ultraljubičasti tekst, laserska graviranja, itd. Tehnike implementacije sigurnosnih obeležja zajedničke su za sve tipove kartica, nisu specifičnost pametnih kartica. Preostale komponente – hardver čipa, operativni sistem i aplikacija – štite podatke i programe mikrokontrolera kartice.



Slika 2.1 Klasifikacija sigurnosnih komponenti pametne kartice

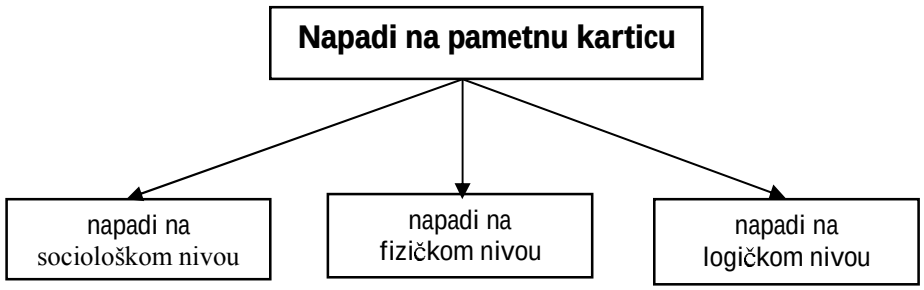
Prisutnost navedenih komponenti i ispravnost rada njihovih obrambenih mehanizama obezbeđuje sigurnost pametne kartice. Ako kartica nije predmet ljudske verifikacije tada nije nužna komponenta tela kartice. Naravno, preostale tri komponente su neophodne u fizičkoj i logičkoj zaštiti kartice. Šta više, ako bilo koja od njih zataji ili ne zadovolji predviđen nivo zaštite, pametna kartica postaje nesigurna. Komponente kartice stvaraju lanac sigurnosti pametne kartice. Ako se prekine jedna karika puca celo lanac.

2.1. Klasifikacija napada i napadača

Osnovni problem svih informacionih sistema predstavlja jednostavnost kopiranja i množenja uspešnih napada. Zbog toga su izuzetno važne klasifikacije napada i napadača u cilju što jasnije procene potencijalnih napada i potrebnih mera u sprečavanju dotičnih napada. Iskustvo govori da je lakše zaštiti sistem od poznatog nego od nepoznatog tipa napada.

2.1.1. Klasifikacije napada

Napadi na pametnu karticu, s obzirom na nivo na kome su izvedeni, mogu se podeliti na tri različita tipa napada: napadi na sociološkom nivou, napadi na fizičkom nivou i napadi na logičkom nivou.



Slika 2.2 Klasifikacija napada na pametnu karticu

Često se u praksi navedeni tipovi napada isprepliću. Napad na fizičkom nivou često je rezultat pripreme za napad na logičkom nivou (npr. napad *diferencijalnom analizom grešaka*).

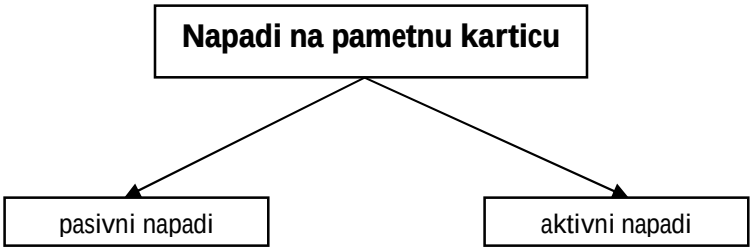
Napadi na sociološkom nivou su primarno usmereni na ljude koji rade sa pametnim karticama. To mogu biti kreatori čipa, zaposleni u poluprovodničkoj industriji, dizajner softvera ili u kasnijem životnom ciklusu kartice nosilac kartice. Ti napadi se samo delomično mogu rešiti tehničkim merama, osnovna zaštita se sprovodi organizacionim merama. Zaštita od tuđih pogleda prilikom ukucavanja PIN-a sprovodi se postavljanjem tastature unutar vizuelnih štitova. Atraktivnost napada usmerenog na programere kartica znatno se smanjuje javnim objavljivanjem korišćenih procedura ili

sprovođenjem evaluacije programskog kôda kod nepristranog ocenjivačkog tela. U tom slučaju se sigurnost temelji samo na tajnosti korišćenih ključeva, time znanje razvojnog programera prestaje biti meta napada.

Napadi na fizičkom nivou obično zahtevaju tehničku opremu jer je potrebno uspostaviti fizički pristup hardveru mikrokontrolera kartice. Napadi mogu biti *statički*, kad napajanje nije prisutno na mikrokontroleru, ili *dinamički*, kad se napad izvršava za vreme rada mikrokontrolera. Statički fizički napad ne nameće vremenska ograničenja napadaču, on svoj posao može raditi u miru. Za razliku od statičkog, dinamički napad zahteva sofisticiranu opremu za merenje podataka sa brzim odzivom na izmerene podatke.

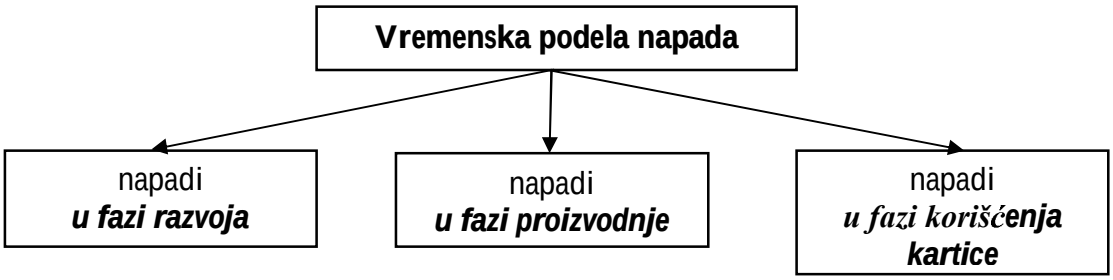
Najuspešniji napadi na pametne kartice izvode se na logičkom nivou. Oni su plod ljudskog razmišljanja i mašinskih operacija. Ova kategorija napada uključuje klasičnu kriptoanalizu, napade koji koriste poznate greške operativnih sistema kartice i trojanske konje unutar izvršnog programskog teksta aplikacija kartice.

Sve gore navedene napade možemo podeliti na *pasivne* i *aktivne*. Prilikom pasivnog napada napadač analizira šifrovani tekst ili kriptografski protokol, sprovodi merenja nad mikrokontrolerom, ali pritom ne vrši nikakve promene. Za razliku od pasivnog napada, aktivnim napadom se vrše promene, napadač se neovlašćeno upliće u prenos podataka i rad mikrokontrolera.



Slika 2.3 Klasifikacija tipova napada

Napadi se mogu klasifikovati obzirom na vremensko usklađivanje sa fazama životnog ciklusa pametne kartice. Pojednostavljenjem faze životnog ciklusa pametne kartice može se napraviti podela na tri vremenska intervala: (a) razvoj, (b) proizvodnju, (c) korišćenje kartice.



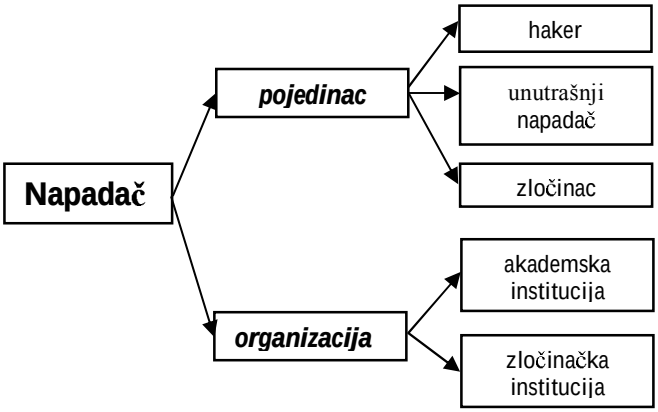
Slika 2.4 Klasifikacija napada u odnosu na vreme odvijanja napada

Napadi za vreme faze razvoja vezani su uz dizajn sistema, razvoj čipa, razvoj operativnog sistema i generaciju aplikacija. Termin "proizvodnja" se odnosi na sve procese od proizvodnje poluprovodničkih štampanih pločica do personalizacije kartica i slanja kartica do krajnjih korisnika. Korišćenje kartice počinje kad vlasnik kartice dobije dodeljenu karticu.

2.1.2. Posledice napada i klasifikacija napadača

Da bi se mogla proceniti snaga i slabost napada na sigurnost pametne kartice važno je poznavati moguće tipove napadača. Dva su osnovna tipa motivacije za napad. Prvi je pohlepa, odnosno novac, a drugi želja za statusom i slavom unutar određene "scene". Sa stanovišta sistemskog operatera svejedno je da li je napad došao od lopova ili obrazovanog čoveka (naučnika). Oba rezultuju značajnim finansijskim gubitkom i narušenom reputacijom sigurnosti sistema. U najgorem slučaju sistem se gasi, sve kartice se blokiraju i izdaju se nove kartice imune na napad. Takav proces kod velikih sistema sa nekoliko miliona kartica u upotrebi može potrajati duže od pola godine.

Slika 2.5 prikazuje klasifikaciju različitih tipova napadača baziranu na prethodnom razmatranju. Sva četiri tipa napadača mogu biti jednako opasni po sistem kartica, a razlikuju se po svojim sposobnostima i mogućnostima.



Slika 2.5 Klasifikacija mogućih tipova napadača

Tipičan haker ima srednji nivo znanja o sistemu, kreativne ideje i grupu sličnih njemu prijatelja. Obično ne poseduje veliku količinu opreme i finansijski je ograničen. Ako primeni odgovarajući pristup može Internet kampanjom doći do značajne procesorske snage.

Unutrašnji napadač ima pristup hardverskim i softverskim komponentama. Poseduje zavidnu količinu sistemskog znanja i upoznat je sa slabim tačkama sistema. Unutrašnji napadač teško može ostati anoniman zbog malog broja potencijalnih unutrašnjih napadača.

Treću grupu potencijalnih individualnih napadača čine *zločinci*. Oni obično ne poseduju visok nivo tehničkog znanja, ali ulažu veliku količinu energije u postizanje ličnog dobitka (najčešće finansijskog).

Ne sme se zanemariti da potencijalni izvor napada predstavljaju i *akademske institucije* sa svim svojim studentima i profesorima. Oni obično nemaju specijalistička znanja unutrašnjih napadača, ali zato poseduju sumu vrednosti generalno korisnog znanja. Imaju pristup velikoj bazi tehničke radne snage i odgovarajućim, tehnički adekvatno opremljenim laboratorijama.

Organizovane zločinačke organizacije predstavljaju potpuno drugačiji izvor napada na sisteme pametnih kartica. Na osnovu dostupnih izvora novca i kriminalnih sredstava stiču znanje i tehničku opremu potrebnu za uspešan napad.

2.2. Napadi i mehanizmi odbrane u fazi razvoja

Generalno se može reći, vezano za napade u fazi razvoja, da je pristup dotičnim uređajima i objektima izrazito težak i da je potreban visok nivo stručnog znanja. Time je atraktivnost napada značajno umanjena, iako je potencijalna opasnost uspešnog napada u

ovoj fazi i dalje prisutna zbog vrlo široke mogućnosti manipulacije hardverom i softverom.

2.2.1. Razvoj mikrokontrolera pametne kartice

Razvoj mikrokontrolera pametne kartice traje nekoliko meseci. U njemu učestvuje mali broj osoba koji rade u strogo obezbeđenim objektima. Sistemi koji učestvuju u razvoju čipa deo su nezavisne mreže fizički odvojene od ostatka sveta. Dizajn čipa i njegovi mehanizmi zaštite daju se na testiranje nezavisnim organizacijama. Nezavisne organizacije detektuju namerne ili nenamerne previde u sigurnosti sistema i time smanjuju mogućnost unutrašnjeg napada. Svakako, napadaču može biti atraktivno i znanje vezano za dizajnerske kriterijume i raspored komponenti čipa. Na taj način saznaje koji su zaštitni mehanizmimi i detektori prisutni na čipu. To je vrlo korisno znanje za kasnije napade zasnovane na fizičkoj analizi čipa.

2.2.1.1. Zaštita: dizajnerske mere

Mehanizmi zaštite od statičkih i dinamičkih napada moraju zaista da funkcionišu. Senzori i drugi elementi zaštite od male su koristi ako se lako onemogućuju ili ako ne funkcionišu u određenim okolnostima. Primer je senzor na čipu mikrokontrolera koji ima toliko malu površinu da se lako može uništiti iglom.

Vrlo je važno da se na čipu ne nalazi apsolutno ni jedan nedokumentovani mehanizam ili funkcija. Nedokumentovana obeležja često su nedovoljno istestirana i predstavljaju potencijalni generator grešaka. Prilikom evaluacije hardvera, jer nisu navedene u dokumentaciji, mogu se nenamerno prevideti i poslužiti za kasnije napade.

2.2.1.2. Zaštita: jedinstven broj čipa

WORM memorija, uz senzore i zaštitne slojeve, predstavlja hardverski sigurnosni element mikrokontrolera. Često se takva memorija naziva OTP memorija. Prilikom proizvodnje poluprovodničkog čipa u WORM memoriju se upisuje jedinstveni broj čipa. To znači da svaki čip postaje jedinstven, odnosno može se unutar nekog sistema na nedvosmislen način identifikovati pametna kartica. Broj čipa se koristi u derivaciji ključa. Isto tako broj čipa omogućuje generaciju "crne liste" kartica. Crna lista kartica se koristi za označavanje i uklanjanje sumnjivih kartica iz sistema.

Broj u originalnom čipu se ne može promeniti, ali se programibilnim mikrokontrolerom može imitirati orginalni čip. Iz tog razloga sigurnosne mere se ne mogu temeljiti na prisutnosti broja dotičnog čipa, već se jedinstveni broj čipa koristi kao baza za prave kriptografske sigurnosne algoritme. Na primer broj čipa se koristi prilikom derivacije tajnih ključeva koji se potom koriste prilikom upit-odgovor autentikacije.

2.2.2. Razvoj operativnog sistema pametne kartice

Računari za razvoj zahtevaju kompletno izolovanu mrežu, bez mogućnosti spoljnog pristupa. Razvojni alati, kao što su prevodilac i simulator, podvrgnuti su verifikaciji i testiranjima. Da se osigura ispravnost prevođenja često se koriste dva različita prevodioca. Zabranjeno je korišćenje softvera nepoznatog porekla.

2.2.2.1. Zaštita: Principi razvoja

Vreme razvoja operativnog sistema se može značajno skratiti uvođenjem dodatnih naredbi operativnog sistema u svrhu testiranja. Takve naredbe često mogu čitati vitalne delove memorije kartice zaobilazeći sigurnosne mehanizme. Ako se takve naredbe nepažljivošću ostave u operativnom sistemu mogu kasnije poslužiti napadaču za čitanje tajnih ključeva pametne kartice. Da se eliminiše mogućnost takvog napada principijelno se zabranjuje kreiranje naredbi testiranja, i svemogućih (globalnih) naredbi.

Programeri nikad ne smeju raditi sami na projektu. Time se smanjuje mogućnost unutrašnjeg napada. Periodično se provode izveštaji nad izvornim kodom što osigurava kvalitet koda i omogućava kontrolu razvojnog procesa.

Sa završetkom razvoja operativnog sistema kartice celi izvorni kôd i njegove funkcije se daju na testiranje nezavisnim telima kao deo evaluacije softvera kartice. Glavni razlog tog obično skupog i dugotrajnog procesa leži u otkrivanju programerskih grešaka i u otkrivanju mogućeg Trojanskog konja, sakrivenog od strane razvojnog programera, u operativnog sistemu.

2.2.2.2. Zaštita: Deljenje znanja

Ako nekoliko osoba radi na zadatku rezultat će biti značajno otporniji na napad zbog raznolikosti njihovih ideja i iskustava. Princip podeljenog znanja (deljenje tajni) suprostavljen je principu da svako zna sve o svemu. U razvoju sigurnosnih komponenti ne sme se dozvoliti da kompletno znanje o komponenti pripadne pojedincu, jer bi takav pojedinac bio meta potencijalnog napada.

Sigurnosni efekt dobijen deljenjem znanja može se videti po načinu na koji je izvedena poslednja faza implementacije operativnog sistema pametne kartice. U njoj se u EEPROM memoriju učitavaju tablice, programski kod i konfiguracioni podaci. Proizvođač kartica, odnosno proizvođač čipa koji dobija finalni ROM kôd za produkciju proizvodnih maski, nema kompletno znanje o operativnom sistemu jer mu nisu poznati

delovi operativnog sistema smešteni u EEPROM-u. Analizom ROM kôda ne može u potpunosti otkriti sigurnosne mehanizme i funkcije operativnog sistema.

2.3. Napadi i mehanizmi odbrane u fazi proizvodnje

Napadi u fazi proizvodnje čipa kartice su tipično unutrašnji napadi zbog zatvorenosti proizvodne okoline. Svaki pristup se strogo kontroliše i beleži. Usprkos tome moguće je izvesti vrlo efikasne i tehnički zanimljive napade ako se ne preduzmu određene sigurnosne mere.

2.3.1.1. Zaštita: autentifikacija u završnim koracima

Već u ranoj fazi proizvodnje mikrokontroler čipa se imenuje brojem čipa i zaštićuje transportnim kôdom. Transportni kôd je u današnjim operativnim sistemima specifičan za pojedini čip, a autentikacija čipa nezaobilazan deo svakog pristupa procesu finalizacije čipa. Iako to podiže troškove i trajanje finalizacije i zahteva sigurnosni modul za svaki deo hardvera koji deli u tom procesu, time se značajno podiže nivo sigurnosti.

Mogući tip napada u procesu finalizacije bi bio opstrukcija pametnih kartica lažnim čipovima koji u potpunosti imitiraju originalne čipove, ali dodatno uključuju i svemoćne (globalne) naredbe za čitanje memorije. Takav tip napada može se prikazati na primeru pametne kartice za digitalno potpisivanje. Napadač pre inicijalizacije kartice zameni izvornu pametnu karticu lažnom pametnom karticom. Lažna kartica se inicijalizuje izvornim podacima i kasnije personalizuje. Takva kartica ima sve funkcije originalne pametne kartice. Njen mikrokontroler generiše PKI ključeve koristeći podatke dobijene procesom inicijalizacije i personalizacije. Nakon toga napadač ponovno mora doći u posed lažne pametne kartice. Korišćenjem globalnih naredbi lažne kartice saznaje se tajni ključ. Napadač sad poseduje sve podatke potrebne za izradu duplikata kartica. Pošto je pripadni javni ključ potpisan od strane CA centra, što jemči orginalnost dotičnog PKI para ključeva, duplikati će biti prepoznati kao izvorne kartice.

Realno, takav napad je nemoguće realizovati zbog administrativnih mera koje sprečavaju da se čipovi i pametne kartice donose ili odnose iz ustanove u kojima se vrši finalizacija kartice. Zamena kartica ili čipova se dodatno onemogućava autentikacijom pametne kartice i sigurnosnog modula hardvera u svakom koraku finalizacije.

2.4. Napadi i mehanizmi odbrane u fazi korišćenja kartice

Pristup pametnoj kartici najlakše je ostvariti nakon što je ona izdata. Rezultat je, visoka verovatnost napada u fazi korišćenja pametne kartice.

Oduvek je bila prisutna zamisao samouništavajućeg mikrokontrolera kao efikasnog rešenja za sve tipove napada. Takvi sigurnosni moduli postoje i najčešće se

koriste u vojne svrhe. Kod pametnih kartica se ne dopuštaju takve sigurnosne mere iz nekoliko razloga. Pametna kartica nema svoj vlastiti, rezervni izvor energije, energiju crpi iz spoljnog izvora. Kad spoljno napajanje nije prisutno kartica ne može prepoznati potencijalni napad i ne postoji mogućnost bilo kakve vrste aktivnog obrambenog mehanizma. Odvojeno od toga, u nekim slučajevima postoje i zakonski razlozi koji onemogućuju primenu samouništenja kartica. Npr. postavlja se potanje ko će biti odgovoran za gubitke nastale zbog neispravnog samouništenja kartice usled nepredviđenih okolnosti? Zapravo, pravo samouništenje često nije potrebno, dovoljno je izbrisati tajne ključeve.

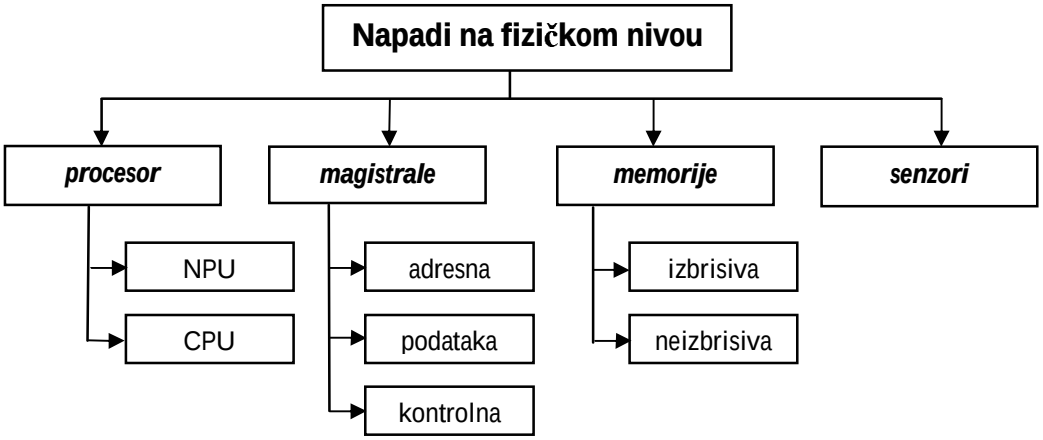
Pametna kartica teško može prepoznati da je napadnuta. Jednostavno ne postoji senzor koji bi rekao: "Napad! Sve izbriši!". Preniski naponski nivo ili povišena frekvencija takta mogu biti znak napada, ali isto tako se mogu pojaviti u normalnom radu usled nekih nepredviđenih spoljnih faktora, npr. uprljani kontakti stvore visoki kontaktni otpor što rezultuje niskim nivoom operativnog napona. Zbog toga, što je tako teško prepoznati stvarni napad, mehanizmi za automatsko blokiranje kartice i brisanje ključeva se najčešće ne koriste.

U nastavki su opisani i objašnjeni klasični napadi na sigurnost pametne kartice. Oni ne predstavljaju pretnju sigurnosti današnjih pametnih kartica jer su poznati već neko vreme, javni i protiv njih su postavljeni odgovarajući mehanizmi odbrane. To ne znači da ih treba ignorisati, naprotiv, razumevanje tih napada predstavlja bazu znanja za borbu protiv potencijalnih budućih napada i istovremeno sprečava ponavljanje prošlih grešaka.

Napadi se dele na napade usmerene direktno na hardver čipa i na napade izvedene na logičkom nivou, usmerene na sistem pametne kartice. Fizički napadi i analitičke metode mogu se podeliti na statičke i dinamičke napade, zavisno od toga da li prilikom napada čip miruje ili je aktivan.

2.4.1. Napadi na fizičkom nivou

Manipulacije na nivou poluprovodnika zahtevaju visok nivo stručnog znanja i sofisticiranu opremu. To umanjuje mogućnost potencijalnog napada na fizičkom nivou, ali ga ne uklanja. Za pretpostaviti je da postoji potencijalni napadač sa svom potrebnom opremom i sredstvima za takvu vrstu napada. Prema tome proizvođači kartica moraju ugraditi odgovarajuću zaštitu u sklopove kartice.



Slika 2.6 Klasifikacija napada na mikrokontroler kartice na fizičkom nivou

Da bi se izveo napad na fizičkom nivou potrebno je izvaditi modul čipa iz tela kartice. To se može učiniti oštrim nožem. Zatim se sa čipa uklanja zaštitni plastični sloj. Anderson i Kuhne su to učinili uz pomoć nitratne kiseline i infracrvene svetlosti. Nakon opisanog postupka čip je slobodan i još uvek u potpunosti operativan. Napadaču takav čip može izgledati nezaštićen poput otvorene knjige, ali to je samo privid. On još uvek mora savladati mnoštvo sigurnosnih mera da dođe do mogućnosti pristupa tajnim podacima.

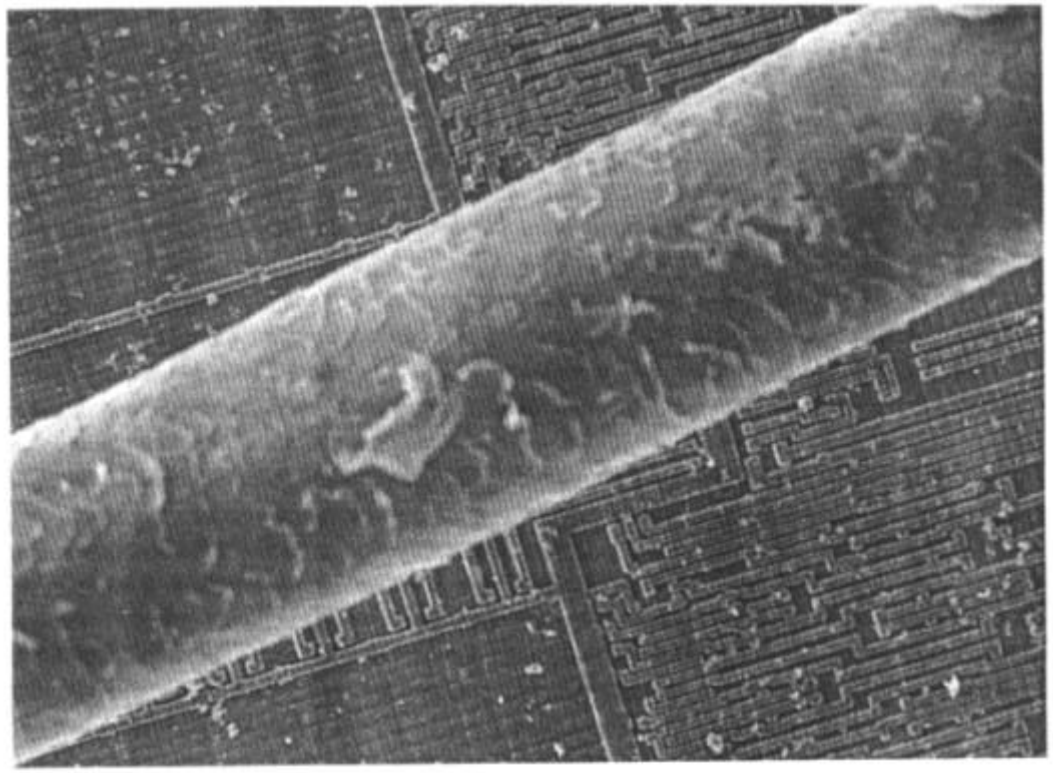
Sigurnosne mere implementirane u hardveru kartice mogu se podeliti na pasivne i aktivne elemente. Pasivni elementi su direktno bazirani na tehnikama poluprovodničke proizvodnje. Oni uključuju i sve procese koji štite memorijska područija i druge funkcionalne delove mikrokontrolera od različitih tipova analiza. Na poluprovodničkom čipu se nalazi i skup aktivnih elemenata koji upotpunjuju pasivnu zaštitu. Pojmom aktivne zaštite označava se integracija aktivnih elemenata, različitih tipova senzora, u silicijumski kristal. Te senzore prema potrebi poziva i proverava operativni sistem pametne kartice. Merenje i obrada signala pristiglih od senzora moguća je jedino ako je na čipu prisutno napajanje. Senzor sa svojstvom svetlosne osetljivosti, zadužen za otkrivanje optičke analize, neće reagovati ako se čip nalazi na predmetnom staklu optičkog mikroskopa bez prisustva napajanja i signala takta. Osim toga takav senzor je na površini čipa lako vizualno identifikovati, kapnuti na njega kap crnog mastila i na taj način neutralizovati njegovu zaštitnu funkciju.

U nastavku su objašnjeni najvažniji i najčešće korišćeni zaštitni mehanizmi mikrokontrolera pametne kartice.

2.4.2. Statička analiza mikrokontrolera pametne kartice

2.4.2.1. Zaštita: tehnologija proizvodnje poluprovodnika

Strukture čipa su toliko fine i malih dimenzija (tehnologije proizvodnje idu ispod 1 μm) da je praktično nemoguće analitičkim procedurama izvući iz njih informacije.



Slika 2.7 Fotografija poluprovodničke strukture i ljudske kose

2.4.2.2. Zaštita: dizajn čipa

U dizajnu čipa pametne kartice zabranjeno je korišćenje standardnih ćelija razvijenih za masovnu proizvodnju poluprovodničkih komponenti. Standardne ćelije omogućuju proizvođaču poluprovodnika da brzo proizvede široku paletu kvalitetnih čipova, ali istovremeno smanjuju nivo sigurnosti čipa pametne kartice. Njihov dizajn i funkcionalnost su javno poznati što svakako potencijalnom napadaču olakšava napad.

2.4.2.3. Zaštita: lažne strukture

Lažne strukture su elementi poluprovodnika koji nemaju stvarnu funkcionalnost, namenjene su da zbune napadača i navedu na pogrešan trag. Pripadajuća sigurnost oslanja se na tajnost njihovih lokacija. Lažne strukture se mogu posmatrati tako da se njihove promene detektuju i uzrokuju gašenje čipa. Njihov glavni nedostatak je što zauzimaju dodatni prostor na čipu.

2.4.2.4. Zaštita: magistrale čipa

Sve magistrale koje povezuju procesor sa tri različita tipa memorije su interne magistrale čipa, i ne izlaze izvan čipa. To znači da je direktan priključak na te magistrale nemoguć i stoga je nemoguće prisluškivanje i manipulisanje signalima magistrale u cilju

iščitavanja sadržaja memorije. Dodatna zaštita postiže se kodiranjem magistrala tako da je funkcija pojedinačne linije magistrale spolja neprepoznatljiva.

2.4.2.5. Zaštita: dizajn memorije

Sadržaj ROM memorije korišćene u industriji može se čitati bit po bit korišćenjem optičkog mikroskopa. Takav tip analize kod pametnih kartica se sprečava smeštanjem ROM kôda u donje slojeve čipa. Međutim, raznim metodama urezivanja može se ipak doći do ROM ćelija. Zato se kod mikrokontrolera pametnih kartica isključivo koristi specijalan, jonski ubačen ROM čiji se sadržaj ne može pročitati ni vidljivom ni ultraljubičastom svetlošću.

2.4.2.6. Zaštita: zaštitni slojevi

Informacijama skupljenim analizom električnog potencijala aktivnog čipa moguće je izvesti zaključke o trenutnom sadržaju RAM memorije. Zaštita se postiže smeštanjem provodnih metalnih slojeva na vrh dotičnih memorijskih ćelija. Ti slojevi se koriste u distribuciji napajanja, pa ako se uklone, npr. hemijskim urezivanjem, čip postaje neupotrebljiv. Takođe se prati njihov električni otpor tako da čip automatski prestaje sa radom ako dođe do oštećenja zaštitnog sloja. Ispod zaštitnih slojeva nalaze se i fototranzistori koji dodatno detektuju njihovo uklanjanje.

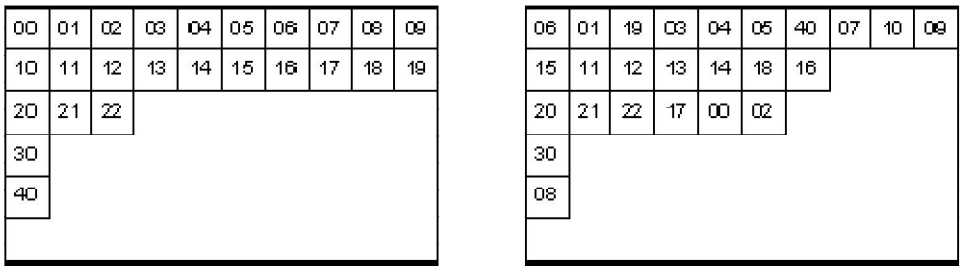
2.4.2.7. Napad i odbrana: čitanje sadržaja izbrisive memorije

Poznato je da sadržaj RAM memorije ne ostaje sačuvan kad se ukloni napajanje. Međutim, ako se memorijske ćelije ohlade ispod –60°C sadržaj RAM ćelija ostati će perzistentan (stalan) i kad se ukloni napajanje. Prema tome tajni ključevi ne bi se smeli nepotrebno držati u RAM memoriji, iza njihovog prisustva u RAM memoriji treba uraditi brisanje ili prepisivanje memorije novim vrednostima.

Čitanje sadržaja RAM ćelija je tehnički teško izvodljivo i zahteva sofisticiranu opremu. Ipak ono je moguće, ali treba znati da bi se ćelije mogle podvrgnuti takvoj analizi sa njih se mora ukloniti zaštitni pasivni sloj i ispod njega zaštitni metalni slojevi. Uklanjanje zaštitnih metalnih slojeva nepovratno uzrokuje uništavanje RAM ćelija jer je deo njihove funkcionalnosti ugrađen u te slojeve. Time se efikasno neutralizuje ovakav tip napada.

2.4.2.8. Zaštita: kodiranje memorije

Najčešća metoda zaštite memorije čipa mikrokontrolera je kodiranje memorije. Sigurnost ove tehnike temelji se na tajnosti primenjene mape kodiranja. Bez te informacije napadaču je izuzetno teško otkriti način adresiranja memorijskih ćelija. Ako se kodiranje izvodi programski, mapa kodiranja može biti specifična za pojedini čip pa čak i dinamička, menjajući se od sesije do sesije.



Slika 2.8 Uporedni prikaz klasične i kodirane memorije

2.4.2.9. Napad i odbrana: šifrovano čuvanje PIN-a

Napad, koji je naveden, demonstrativno prikazuje klasični neefikasni mehanizam odbrane. Pretpostavka je da napadač može uspešno pročitati sadržaj EEPROM memorije. Osnovna ideja odbrane je da se napadaču spreči saznanje vrednosti PIN-a, na način da se PIN šifruje jednosmernom funkcijom. Prilikom šifrovanja jednosmerna funkcija koristi ključ specifičan za karticu tako da referentni podaci za dva ista PIN-a budu različiti za različite kartice. Sada, ako je i moguće pročitati referentni podatak, čini se nemogućim saznati pravu vrednost PIN-a.

Ali, pametan napadač neće ni pokušati iz šifrovane vrednosti saznati pravu vrednost PIN-a. On zna da PIN najčešće čine četiri decimalne cifre što daje oko 10000 potencijalnih mogućnosti. Ako može čitati memoriju pametne kartice tada može i pročitati jednosmernu funkciju i privatni ključ. Nakon provedenih 5000 upoređivanja (srednja vrednost) šifrovanog ispitanog PIN-a i referentnog podatka saznaće pravu vrednost PIN-a. U ovom slučaju šifrovano čuvanje PIN-a ne predstavlja dovoljan nivo zaštite.

2.4.3. Dinamička analiza mikrokontrolera pametne kartice

2.4.3.1. Zaštita: nadziranje pasivnog sloja

Pasivni sloj predstavlja završnu fazu proizvodnog procesa i gornji silicijumski sloj mikrokontrolera. On sprečava oksidaciju i druge hemijske procese na površini čipa. Da bi bila moguća bilo kakva manipualcija čipom pasivni sloj se mora ukloniti. Senzori na čipu izvode merenja otpora ili kapacitativna merenja da ustanove da li je pasivni sloj još uvek prisutan. Ako pasivni sloj nije prisutan ili je oštećen, uzrokuju programski prekid ili čak i prekid rada kompletnog hardvera čipa. To predstavlja uspešnu zaštitu od bilo koje vrste dinamičke analize.

2.4.3.2. Zaštita: nadziranje naponskog nivoa

Nadziranje naponskog nivoa prisutno je u svakom mikrokontroleru pametne kartice. U slučaju da naponski nivo prekorači dopuštene gornje i donje naponske nivoe provodi se strogo definisano gašenje čipa. To pruža softveru pametne kartice sigurnost

stabilnog rada čipa. Kad nadziranje naponskog nivoa ne bi bilo prisutno, mogući bi bili npr. nekontrolirani skokovi unutar toka programa uzrokovani nestabilnom vrednošću programskog brojača. Može se jednostavno reći da rad čipa izvan specificiranog naponskog nivoa rezultuje nizom generisanih greški. Takvo ponašanje čipa predstavlja bazu za određivanje tajnih ključeva tehnikom diferencijalne analize grešaka – DFA .

Na čipu postoji još jedan senzor baziran na detektoru napona, osim gore navedenog koji prati naponski nivo. Ovaj senzor, nezavisno od reset signala, kad prvi put detektuje prisutnost napajanja postavi čip u definisano početno stanje. Time osigurava da čip ne izađe iz dozvoljenog skupa stanja.

2.4.3.3. Zaštita: nadziranje frekvencije takta

Brzinu procesiranja pametne kartice u potpunosti determinira (uspostavlja) spoljni signal takta. Teoretski to znači da se zaustavljanjem signala takta može zamrznuti rad mikrokontrolera i analizirati njegovo izvođenje korak po korak. To su savršeni preduslovi za merenje operativne potrošnje struje i električnog potencijala na površini čipa. Da bi se sprečili takvi napadi u čip se ugrađuje detektor frekvencije. On uklanja mogućnost spuštanja frekvencije takta ispod dozvoljenog nivoa. U većini specifikacija ta minimalna frekvencija takta iznosi 1 MHz. Zbog tehničkih razloga detektor se najčešće izvodi sa širokim opsegom tolerancije tako da se rad čipa zaustavlja ispod 600 kHz.

Da bi se na pravilan način mikrokontroler pametne kartice zaštitio od načina izvođenja u jednom koraku potrebno je zaštitnim slojevima osigurati ispravan rad detektora frekvencije.

2.4.3.4. Zaštita: nadziranje temperature

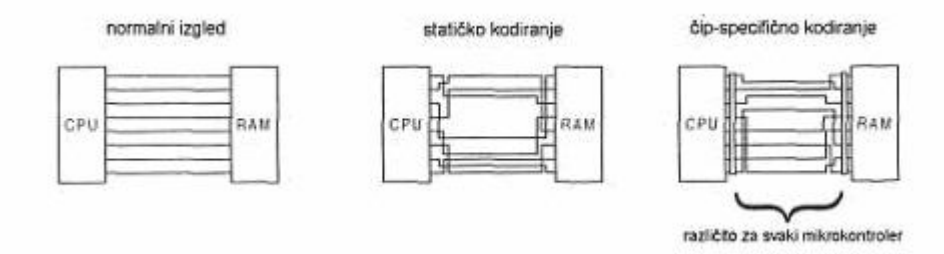
Temperaturni senzor nalazi se u nekim čipovima, iako je veliki znak pitanja nivo sigurnosti koja se dobija njegovim ugrađivanjem. Ako temperatura nakratko pređe maksimalnu dozvoljenu vrednost to neće uništiti čip niti se može reći da to predstavlja napad na mikrokontroler pametne kartice. Gašenje čipa u tim graničnim uslovima može rezultirati primetnim povećanjem procenta grešaka, a operatoru kartica ne pruža dodatnu sigurnost.

2.4.3.5. Zaštita: kodiranje magistrala

U mnoštvu mikrokontrolera kartica primenjuje se kodiranje internih magistrala. To znači da linije pojedinačnih magistrala ne leže jedna do druge u padajućem ili rastućem redosledu, već su nasumice raspoređene i zamenjene nekoliko puta, pa čak i rapoređene u nekoliko slojeva jedne ispod drugih. To predstavlja dodatnu prepreku potencijalnom napadaču jer ne zna koja linija magistrale adresira koji bit ili funkciju.

Kodiranje magistrala se u početku izvodilo statički, sa istom mapom kodiranja za svaki čip. To je predstavljalo potencijalnu slabu tačku jer je napadaču u nekom

vremenskom periodu bilo muguće otkriti mapu kodiranja i tu informaciju iskoristi u prisluškivanju magistrala na svim karticama unutar sistema. Nivo sigurnosti postignut tehnikom kodiranja magistrala povećao se korišćenjem kodirajuće mape specifične za pojedini čip. To nije postignuto proizvodnjom specifične maske za svaki čip jer je to trenutno tehnički nemoguće i preskupo, već generatorom slučajnih brojeva smeštenim tik iznad memorije. Najčešće je serijski broj čipa inicijalna vrednost generatora slučajnih brojeva. Takva tehnika kodiranja magistrala jednostavno se implementira u poluprovodničkoj tehnologiji. Dodatno se još može mapa kodiranja učiniti dinamičnijom ako se za svaku sesiju koristi varijabilna (promenljiva) inicijalizovana vrednost generatora slučajnih brojeva.



Slika 2.9 Kodiranje magistrala mikrokontrolera pametne kartice između memorije i CPU-a prikazano 8-bitnom magistralom

2.4.3.6. Zaštita: nepovratno prespajanje iz testnog načina rada u korisnički način rada

Svi mikrokontroleri pametnih kartica koriste test način rada (test mode) još prilikom provere čipova u procesu proizvodnje. Testiranje načina rada koristi se takođe prilikom izvršavanja test programa. Pritom su dozvoljeni direktni pristupi memoriji, što ne sme biti prisutno u kasnijoj korisničkoj fazi. Prema tome prespajanje iz korisničkog u test modu mora biti nepovratno.

Prespajanje se može izvesti na čipu korišćenjem silicijumskog osigurača. Na ispitnu tačku dovede se napon koji otapa silicijumski osigurač i na taj način izvrši hardversko prespajanje u korisnički režim rada. Normalno, to je ireverzibilan proces. Ipak, osigurač predstavlja relativno veliku strukturu na površini čipa pa ga je moguće mehanički premostiti nakon što se ukloni deo pasivnog sloja nad osiguračem. To bi stavilo mikrokontroler ponovo u testni režim rada i potencijalnom napadaču omogućilo iščitavanje kompletne memorije pametne kartice i proizvodnju njenih duplikata. Da bi se odbranila od takve vrste napada većina proizvođača koristi deo EEPROM memorije kao dodatni mehanizam prespajanja. Ako se u tom delu EEPROM-a locira određena nepromenjiva vrednost (WROM), čip je nepovratno prespojen u korisnički način rada. Sada ako se i premosti osigurač logička sklopka u EEPROM-u sprečava prespajanje u testni režim rada.

Sigurnost prespajanja iz test režima u korisnički režim rada može se dodatno povećati jednostavnim postupkom. Mikrokontroler čipa nalazi se na pločici sa testnim

dodatkom koji sadrži kontakte preko kojih se vrše testiranja čipa. Nakon završetka testiranja dodatak se jednostavno odseče od mikrokontrolera. U ovom slučaju nepotrebni su osigurač i logička sklopka u EEPROM-u jer delovi potrebni za testiranje više nisu prisutni. Sa današnjom tehnologijom nemoguće je uspostaviti ponovnu komunikaciju sa čipom kroz odsečene kontakte na rubu čipa.

2.4.4. Dinamička analiza i mehanizmi odbrane: prisluškivanje memorijskih magistrala mikrokontrolera

Da bi prisluškivanje magistrala između CPU-a i memorije mikrokontrolera bilo moguće mora se sa površine čipa ukloniti pasivni sloj. Pasivni sloj štiti čip od oksidacije, ali i od napada jer njegov integritet nadziru senzori na čipu. Prema Andersonu i Kuhnu on se može ukloniti hlorovodičnom kiselinom (HCl) (testitranja su vršili nad standardnim mikrokontrolerima, specijalni mikrokontroleri pametnih kartica štite integritet pasivnog sloja). Za selektivno otvaranje pasivnog sloja koristi se laserski rezač.

Ako je napadač uspešno uklonio celi ili samo jedan deo pasivnog sloja teoretski može mikro probnim iglama uspostaviti kontakte na magistrale. Uspešno izvedenom konekcijom može adresirati bilo koju ćeliju memorije i pročitati podatke, a da čip pritom ne mora biti aktivan. Uspešan napad ovom metodom rezultirao bi otkrivanjem svih tajnih podataka na čipu. Štaviše sa brzim logičkim analizatorom za vreme rada čipa mogla bi se snimiti kompletna komunikacija između memorije i CPU-a.

Izuzetno je teško na čipu ostvariti pojedinačni električni kontakt. Minimalni broj potrebnih kontakata za ovakav tip napada zavisi od broja postojećih magistrala, 16 adresnih linija, 8 podadresnih linija, 1-4 kontrolne linije, minimalno 25 linija. Čak i modernom mikrotehnologijom, zbog vrlo malih dimenzija poluprovodničkih struktura, nemoguće je ostvariti 25 istovetnih konekcija. Ipak, korišćenjem generatora za fokusiranje jonskih zraka (alat koji se koristi u poluprovodničkoj industriji) moguće je stvoriti provodljivu kontaktnu površinu za svaku liniju magistrale i zatim na nju dovesti vrh mikroprobne igle. Procedura je složena, a utrošeni napor ogroman.

Čak ako i napadač uspe ostvariti navedene konekcije, da bi mogao uspešno čitati podatke, još uvek mora utvrditi na koji su način magistrale kodirane.

Tehnologija prisluškivanja će u budućnosti doživeti značajna poboljšanja u odnosu na trenutne metode prisluškivanja današnjih mikrokontrolera. Ali, treba znati da se ona temelji na mikromehaničkoj tehnologiji i da će uvek zaostajati za poluprovodničkom tehnologijom koja se temelji na optičkim metodama. To znači, da čak i u budućnosti, ovakav tip napada neće moći značajno ugroziti sigurnost pametnih kartica.

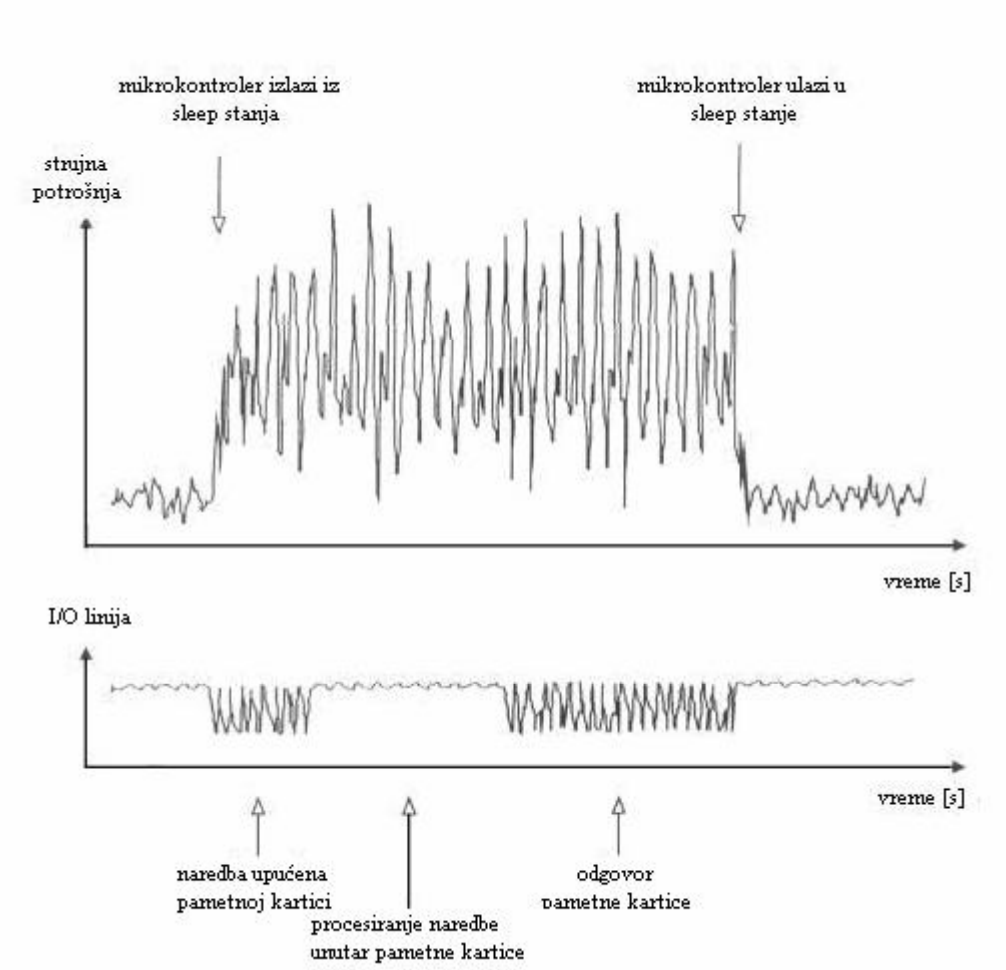
2.4.5. Dinamička analiza i mehanizmi odbrane: merenje potrošnje struje CPU-a

Procesor pametne kartice mora imati podednaku potrošnju struje za sve svoje hardverske instrukcije. Paul Kocher i njegovi saradnici su 1998. objavili članak sa temom jednostavne analize potrošnje – SPA i članak s temom diferencijalne analize potrošnje – DPA. Navedene analize se temelje na činjenici da se određeni deo tajne informacije može izvesti analizom potrošnje struje procesora dok izvodi instrukcije i procesira pripadajuće podatke.

Za vreme rada procesora, SPA analizom meri se pad napona otpornika serijski spojenog na V_{cc} kontakt. Merenje se izvodi brzim AD konvertorom, povezanim sa posebnim računarom. Zbog kompleksnosti internih procesa, prilikom izvođenja instrukcija, samo su kod jednostavno dizajniranih CPU-ova izmereni rezultati i njihove varijacije mogu dovesti u vezu sa procesiranim instrukcijama i podacima.

DPA analiza je znatno moćnija. Kod DPA analize meri se potrošnja struje mikrokontrolera dok procesira poznate podatke i zatim dok procesira nepoznate podatke. Merenja se ponavljaju nekoliko puta. Uzimaju se prosečne vrednosti da se eliminiše efekat šuma. Kad se sva merenja završe obrađuju se izmerene razlike. Iz dobijenih rezultata složenim algoritmom se izvlače zaključci vezani uz nepoznate podatke. Navedeni algoritam izuzetno je kompleksan, ali javnim objavljivanjem i detaljnim obrazlaganjem njegovog rada, postao je dostupan velikom broju potencijalnih napadača.

Potrošnja struje kod nekih mikrokontrolera zaista direktno zavisi od tipa izvedene instrukcije i podacima koje instrukcija procesira. Zbog toga, ako se ne sprovedu određene protivmere zaštite, navedeni tipovi analize predstavljaju ozbiljne oblike napada na sigurnost hardvera i softvera pametnih kartica. Protivmere koje se mogu preduzeti temelje se na hardverskim ili softverskim ješenjima. Najjednostavnije hardversko rešenje predstavlja naponski regulator sa brzim odzivom, smešten na čipu, čime strujna potrošnja mikrokontrolera postaje nezavisna od izvedene instrukcije i pripadajućih podataka. Delotvorno rešenje je i postavljanje na čip dodatnih generatora strujnog šuma. Tehnički najkomplikovanije rešenje je promena dizajna procesora mikrokontrolera tako da mikrokontroler zaista uvek ima konstantnu potrošnju struje.



Slika 2.10 Pojednostavljeni prikaz potrošnje struje mikrokontrolera pametne kartice. Potrošnja struje zavisi od izvedenih mašinskih instrukcija

Najjednostavniji softverski pristup je korišćenje samo onih mašinskih instrukcija čija se strujna potrošnja značajno ne razlikuje od nekog prosečnog nivoa. Dodatno rešenje predstavlja skup različitih procedura koje se koriste u izvođenju istih računskih operacija unutar šifarskih algoritama. Selekcija pojedine procedure obavlja se slučajnim odabirom. To značajno otežava prepoznavanje korelacije između poznatih i nepoznatih instrukcija ili procesiranih podataka. Prikupljanje podataka potrebnih za DPA analizu može se otežati i na način da se svi tajni ključevi zaštite brojačima. Brojači broje neuspešne pokušaje autentifikacije. Takođe, potrebno je blokirati slobodan pristup svim naredbama koje koriste tajne podatke kroz šifarske algoritme pametne kartice. Ako je njihova upotreba neophodna, pametna kartica pre izvršavanja takve naredbe mora provesti autentifikaciju terminala. Navedena ograničenja dodatno otežavaju prikupljanje relevantnih podataka potrebnih za sprovođenje analize.

2.4.6. Analiza i mehanizmi odbrane: merenje elektromagnetskog zračenja CPU-a

Teoretski je moguće iz merenja elektromagnetskog zračenja izvesti zaključke o unutrašnjim procesima mikrokontrolera pametne kartice, slično kao kod DPA analize. Ali ta tehnika je veoma komplikovana i zahteva poznavanje unutrašnje strukture poluprovodničkog uređaja, što najčešće nije dostupno. Značajna zaštita se postiže slaganjem nekoliko poluprovodničkih slojeva, jednih iznad drugih, tako da ako se i izmeri magnetno polje ne može se utvrditi koji sloj zapravo vodi pripadnu struju.

2.4.7. Manipulacija mikrokontrolera pametne kartice

2.4.7.1. Manipulacija i mehanizmi odbrane: promene sadržaja memorije mikrokontrolera pametne kartice

Možda i najočigledniji scenario napada predstavlja pokušaj direktnog čitanja memorijskog sadržaja mikrokontrolera. Takođe, podjednako snažan oblik napada su i namerne promene sadržaja memorije mikrokontrolera pametne kartice. Pritom se ne misli na razbacane slučajne greške u računskim procesima šifarskih algoritama, što se koristi kao temelj napada diferencijalnom analizom grešaka (DFA), već na ciljane promene određenih bitova ili bajtova u ROM ili EEPROM memoriji.

Promene se mogu izazvati izlaganjem modula X-zracima, što dovodi do promena na svim tipovima memorija. Ćelije EEPROM memorije prazne se izlaganjem ultraljubičastoj svetlosti i zauzimaju najniže energetsko stanje. Korišćenjem fokusiranih zraka svetlosti ili korišćenjem laserske svetlosti može se menjati sadržaj pojedinačnih memorijskih ćelija. Moguće promene memorijskog sadržaja podloga su neki od teoretskih uspešnih napada. Npr. moguća je manipulacija generatorom slučajnih brojeva tako da daje uvek iste vrednosti. To bi rezultiralo lažnom autentifikacijom terminala i pametne kartice na osnovu prisluškivanih vrednosti prethodnih autentifikacija.

Ako je poznata tačna lokacija DES ključa u EEPROM-u i ako je moguće modifikovati pojedinačne bitove EEPROM memorije, moguće je iskoristiti te uslove za izvođenje uspešnog napada. Napad se sastoji u postavljanju određenog bita DES ključa na 0 i zatim pozivanja naredbe koja koristi DES algoritam sa modifikovanim ključem. Ako povratni kôd javi grešku parnosti u ključu znači da je bit bio orginalno postavljen na 1. Ako greška nije prijavljena bit je orginalno postavljen na 0. Procedura se ponavlja za preostalih 55 bita i rezultuje otkrivanjem tajnog ključa.

Da bi napadač mogao ciljno menjati pojedinačne bitove mora poznavati fizičke adrese podataka i programskog koda unutar memorije. Mora takođe poznavati pripadajuću mapu kodiranja dotične memorije. Dodatno se svi podaci i programski blokovi značajni za pitanje sigurnosti štite kontrolnim sumama koje se proračunavaju prilikom svakog njihovog korišćenja. Napadač bi zbog toga morao stalno menjati i odgovarajuću kontrolnu sumu modifikovanih podataka. Da bi uopšte moglo doći do

navedenih manipulacija potrebno je neutralizovati sve zaštitne slojeve koji pokrivaju memoriju. Sva ta navedena razmatranja zajednički spuštaju nivo atraktivnosti ovakvog tipa napada na nulu.

2.4.8. Napadi na logičkom nivou

Glavni preduslov napada na sigurnost pametne kartice na logičkom nivou odnosi se na razumevanje komunikacionog i informacionog toka između terminala i pametne kartice. Važnije je shvatanje softverskih procesa od procesa koji se izvode na nivou hardvera.

2.4.8.1. Analiza: određivanje skupa naredbi pametne kartice

Instrukcijske klase i naredbe, podržane od pametne kartice, najčešće nisu javne, ali ih možemo odrediti jednostavnom procedurom pretraživanja. Procedura pretraživanja više je interesantna s obzirom na kompletno određivanje skupa naredbi nego što predstavlja napad na sigurnost pametne kartice.

Može se posmatrati opis procedure korišćene za određivanje skupa naredbi. Prvi korak je generisanje APDU naredbe i njeno slanje pametnoj kartici. Bajt klase u APDU bloku menja se od "00" do "FF". Kad pametna kartica vrati odgovor koji je različit od oznake za nepodržanu naredbu određen je prvi bajt i determinisana pripadajuća instrukcijska klasa (CLA bajt). Obično su podržane dve ili tri instrukcijske klase. Daljom procedurom za svaku instrukcijsku klasu na isti način određuje se podržani instrukcijski bajtovi (INS bajtovi).

Otkrivanje kompletnog skupa naredbi traje oko dva minuta. Prikazani algoritam može se dodatno ubrzati ako se koriste samo kodovi bajtova klasa koje dozvoljava ISO/IEC 7816-4 standard. Dodatno poboljšanje se postiže korišćenjem samo parnih vrednosti instrukcijskih bajtova jer su neparne vrednosti rezervisane za V_{pp} kontrolnu informaciju koja se danas više ne koristi.

Razlog efikasnosti ovog jednostavnog algoritma leži u tome što gotovo svi interpreteri naredbi operativnih sistema pametnih kartica proveravaju pristigle naredbe počinjući sa bajtom instrukcijske klase i redom nastavljajući dalju proveru bajtova. Čim se u tom rasporedu bajtova pojavi nevalidni bajt, proces provere se zaustavlja, generiše se odgovarajući odgovor i šalje terminalu.

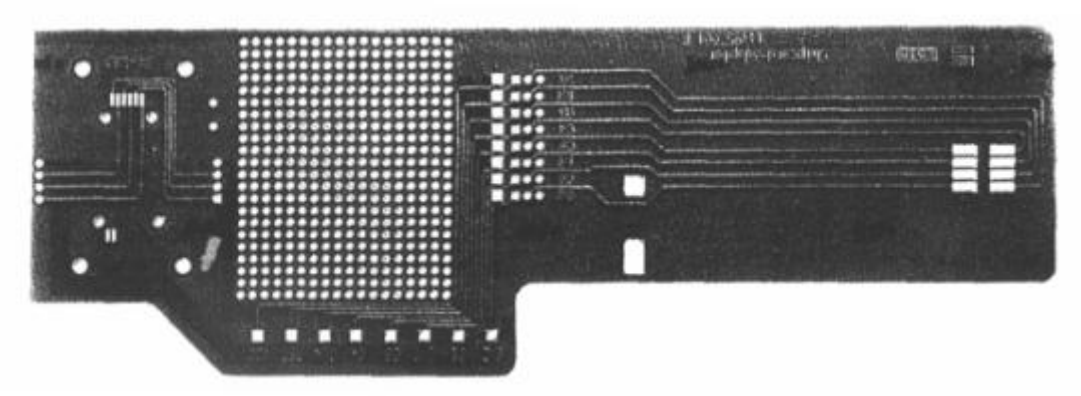
Pametne kartice koje nadziru sled naredbi globalnom mašinom stanja neće dopustiti opisanu proceduru pretraživanja podržanog skupa naredbi.

Prednost koju dobija napadač primenom ove metode možda se ne čini velikom jer skup naredbi obično nije tajan. Ali, na ovaj način napadač vrlo brzo može odrediti da li je proizvođač operativnog sistema ostavio u njemu nedokumentovanu naredbu.

2.4.8.2. Napad: prisluškivanje prenosa podataka

Neznatno promenjena pametna kartica može poslužiti u prisluškivanju prenosa podataka, pritom vršeći željene manipulacije nad podacima. Potrebna modifikacija kartice sastoji se od lepljenja posebnog lažnog kontakta na vrh I/O kontaktne površine modula pametne kartice tako da originalni I/O čitač terminala nije više povezan sa originalnim I/O kontaktom pametne kartice. Lažni kontakt je povezan sa brzim računarom. Programiranjem odgovarajućeg odziva računar može brisati ili umetati podatke u komunikacioni kanal između terminala i pametne kartice. Ako je računar dovoljno brz ni terminal ni kartica neće primetiti razliku između normalne i modifikovane komunikacije.

Očigledno je da se ovom metodom može značajno uticati na izvođenje sesije. Da li će napadač pomoću te metode izvući određenu korist i koliku, primarno zavisi od aplikacije kartice. Sigurnost dobro dizajniranog sistema neće biti narušena prisluškivanjem, brisanjem ili umetanjem podataka u komunikacionom kanalu.



Slika 2.11 Alat sa kojim se komunikacija terminal-kartica vrši eksterno izvan čitača terminala. Kontakti koji ulaze u terminal vide se desno, a pločica za analizu nalazi se levo.

Da bi se zaštitili od napada realizovanog ovom metodom terminali koriste čitače koji pre nego što uspostave kontakt sa pametnom karticom spuštaju zaslone oko njenog čipa. Na taj način zasloni seku sve žice koje idu od čipa. Drugi tip zaštite postiže se korišćenjem sigurne komunikacije ISO komunikacionog modela (zaštita APDU poruka), čime se na pouzdan način detektuju izvršene manipulacije.

Većina terminala je u praksi pod video nadzorom što otežava korišćenje modifikovanih pametnih kartica povezanih sa propratnim računarom. Može se zaključiti da iako teoretski ovakav tip napada izgleda privlačno i obećavajuće, u praksi ga je skoro nemoguće izvesti i povrh toga, ako se i sprovede, najčešće rezultuje neuspehom.

2.4.8.3. Napad i odbrana: uklanjanje napajanja

Uklanjanje napajanja sa čipa pametne kartice za vreme izvršavanja određene naredbe bio je donedavno uspešan tip napada kod mnoštva pametnih kartica. Pozadinu ovog napada čine ciklusi pisanja EEPROM stranica. Napad se može objasniti na primeru aplikacije e-novčanika, ako se iznos sačuvanog novca poveća pre nego što se ažurira dnevna datoteka, prilikom izvršavanja naredbe punjenja e-novčanika napadač će imati dobru šansu besplatno napuniti e-novčanik pametne kartice. Mora samo u pravom trenutku (preciznost na nivou milisekunde) prekinuti napajanje ili izvući karticu iz terminala. Iznos novca u e-novčaniku će biti promenjen, a u dnevnoj datoteci neće postojati zapis transakcije. U prošlosti je kod nekih jednostavnih sistema ovakav napad bio moguć.

Međutim apsolutna zaštita se ne može postići čak i savršenim sledom operacija pisanja. Npr. prilikom punjenja e-novčanika iz prethodnog primera, pre ciklusa pisanja interno se izvršava ciklus brisanja EEPROM stranice. Ako izbrisano stanje EEPROM-a odgovara maximalnom iznosu e-novčanika, što je najčešće istina, tada samo treba prekinuti napajanje u pravom trenutku i novčanik će biti napunjen. Taj trenutak je nakon što završi operacija brisanja, a ne započne operacija pisanja.

Dizajneri operativnih sistema, da bi sprečili takav tip napada, koriste efikasne protivmere, atomske procese. Karakteristika atomskih procesa je da su nedeljivi, izvršavaju se u potpunosti ili nikako. Oni predstavljaju potpunu zaštitu od gore opisanog napada.

2.4.8.4. Napad i odbrana: strujna analiza za vreme upoređivanja PIN-a

Napad na neku vrednost koja se poredi, kao što je PIN, može se izvesti kombinacijom fizičkih merenja i izmenama logičke vrednosti parametra koji se meri. Ovaj tip napada odnosi se na sve mehanizme koji šalju podatke pametnoj kartici na upoređivanje sa odgovarajućim vrednostima. Vrednosti brojača pokušaja se povećavaju zavisno od rezultata upoređivanja.

Napad se izvodi na principu merenja strujne potrošnje pametne kartice. Kad se kartici pošalje odgovarajuća naredba zajedno sa podatkom za upoređivanje, moguće je pre nego što kartica vrati odgovor iz strujnih merenja otkriti da li je brojač pokušaja povećan. Slanjem svih mogućih kombinacija poredbenog podatka i prekidanjem napajanja kartice svaki put kad se merenjem detektuje negativan rezultat upoređivanja, može se otkriti originalna vrednost sačuvana na kartici.

Dva su osnovna načina odbrane od ovakvog tipa napada. Najjednostavnija odbrana se postiže da se pre upoređivanja uvek poveća brojač pokušaja, a nakon upoređivanja se zavisno od rezultata smanji. Sada brojač pokušaja ostaje uvećan nezavisno od toga kad napadač prekine napajanje. Drugi način odbrane je složeniji. Brojač pokušaja se uvećava nakon negativne poredbe, a nakon pozitivne poredbe upisuje se u praznu EEPROM ćeliju. Oba ciklusa pisanja izvode se unutar procesa upoređivanja u

isto vreme tako da napadač ne može izvesti zaključke o rezultatu upoređivanja. Rezultat saznaje tek kad pristigne odgovor od kartice u obliku povratnog koda, ali tada je prekasno da se prekidom napajanja spreči ciklus pisanja u polje brojača pokušaja.

2.4.8.5. Napad i odbrana: vremenska analiza za vreme upoređivanja PIN-a

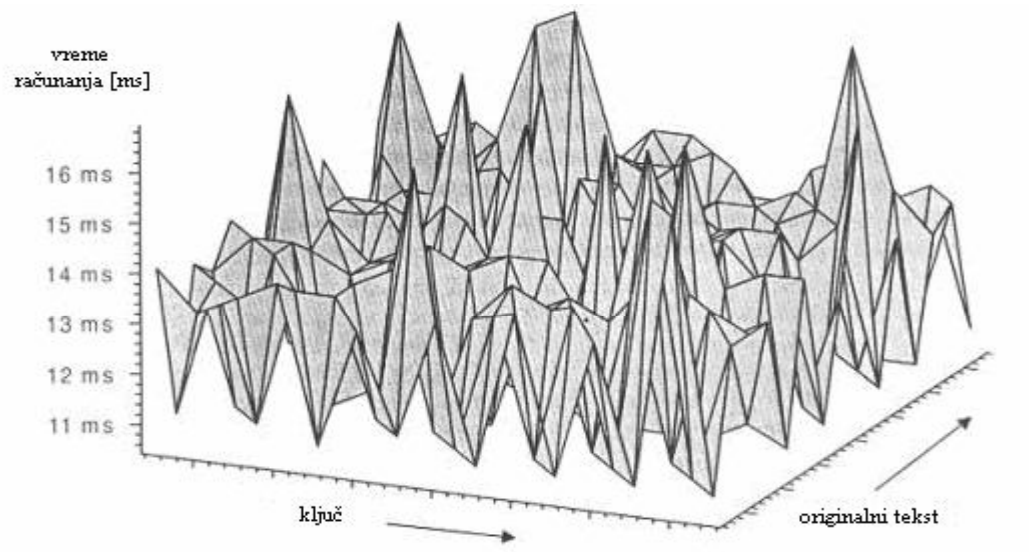
Programeri uvek ulažu velike količine napora na povećanje brzine izvođenja svojih programa. Činjenica da je vreme izvršavanja nekog procesa minimalno može postati podloga vrlo uspešnom napadu. Kad se pošalje PIN pametnoj kartici na upoređivanje, odgovarajuća programska procedura, zadužena za upoređivanje pristiglog PIN-a, sa vrednošću PIN-a sačuvanog na kartici, sprovodi upoređivanje bajt po bajt. Programer koji ne razmišlja o sigurnosti programiraće navedenu proceduru tako da prva razlika između poredbenih vrednosti rezultuje prekidom izvođenja. Kao posledica javljaju se male razlike u vremenu izvođenja upoređivanja PIN-ova. Mereći te razlike napadač na vrlo jednostavan način određuje tajni PIN.

Ovakav tip napada na pametne kartice bio je uspešan do pre nekoliko godina. Danas se procedure zadužene za procese upoređivanja konstruišu na način da se upoređuju sve cifre PIN-a, pa na taj način nema vremenske razlike između rezultata pozitivnog i negativnog upoređivanja.

2.4.8.6. Zaštita: šifarski algoritmi sa odsutnošću šuma

Sigurnost aplikacije pametne kartice temelji se na tajnosti ključeva korišćenih u šifarskim algoritmima. Terminal se mora autentifikovati uz pomoć tajnog ključa da bi mogao sa pametnom karticom izvršiti određene akcije. Autentifikacija terminala od strane pametne kartice predstavlja potencijalnu metu napada na njenu sigurnost. Pametna kartica autentifikuje terminal slanjem slučajnog broja, koji terminal šifruje i vraća kartici. Zatim pametna kartica izvodi istu enkripciju i upoređuje rezultat sa vrednošću dobijenom od terminala. Ako je rezultat upoređivanja pozitivan terminal je uspešno autentifikovan i prima odgovarajući povratni kod. Početna tačka napada je analiza vremena procesiranja od slanja šifrovanog broja kartici pa do primanja odgovarajućeg povratnog koda.

Nivo šuma šifarskog algoritma određuju vremenske razlike u trajanju procesa šifrovanja ili dešifrovanja zavisno od ulaznih vrednostima, originalnom tekstu i tajnom ključu. Što su te vremenske razlike veće, algoritam je bučniji.



Slika 2.12 Primer bučnog šifarskog algoritma

Članak o napadu DPA analizom, koga je Paul Kocher objavio 1998., došao je kao posledica daljeg istraživanja napada objavljenog još 1995 godine. Taj napad iz 1995. temeljio se na analizi bučnosti šifarskog algoritma, odnosno na varijacijama u vremenu procesiranja enkripcije različitih ulaznih vrednosti. Na osnovu izmerenog šuma, složenim algoritmom, javno objavljenim, otkrivao se tajni ključ.

Današnje pametne kartice koriste šifarske algoritme sa odsutnošću šuma, što znači da vreme trajanja procesa enkripcije i dekripcije ne zavisi od ulaznih vrednosti. Time se sprečava gore navedeni napad. Negativna posledica je da algoritmi sa odsutnošću šuma zahtevaju više programskog koda i da su uvek sporiji od svoje bučne verzije. Razlog tome je dizajn algoritma sa odsutnošću šuma, koji mora biti takav da je put kroz algoritam jednako dugačak za sve kombinacije originalnih podataka, šifrovanih podataka i tajnih ključeva. To znači da se za referentnu vrednost uzima najduži mogući put, a sve ostale kombinacije moraju se izravnati po toj vrednosti.

Dodatna sigurnost postiže se pridodavanjem brojača pokušaja svakom autentifikacijskom ključu, tako da se može izvršiti samo ograničeni broj neuspešnih autentifikacija. Kad brojač pokušaja dosegne maksimalnu vrednost blokira se svaki dalji pokušaj autentifikacije.

2.4.8.7. Manipulacija: diferencijalna analiza grešaka (DFA)

Istraživači Bell Communications Research laboratorije (Bellcore) su 1996. opisali teoretski model određivanja tajnog ključa asimetričnih šifarskih algoritama pomoću raspršenih hardverskih grešaka. Po njima je takav tip napada dobio ime Bellcore napad.

Dva meseca kasnije, Eli Biham i Adi Shamir objavili su ekstenziju Bellcore napada koja je obuhvatala i simetrične šifarske algoritme i nazvali je *diferencijalna analiza grešaka (DFA).*

Osnovni princip oba napada je relativno jednostavan. U prvom koraku odabere se blok podataka za enkripciju i šifruje se korišćenjem ključa koji predstavlja metu napada. Rezultat šifrovanja se sačuva. U sledećem procesu šifrovanja koristi se isti ulazni blok podataka, ali se pritom kartica izlaže visoko frekventnom zračenju koje promeni jedan bit korišćenog ključa. To rezultuje neispravno šifrovanim podacima. Proces se ponavlja nekoliko puta, a rezultati se čuvaju za dalju analizu. Iz sačuvanih rezultata se matematičkim putem može izračunati vrednosti tajnog ključa.

Snaga ovog napada leži u činjenici da nije potrebno poznavati tačnu lokaciju promenjenog bita tajnog ključa. Ako se istovremeno promeni više bitova tajnog ključa potrebno je više neispravno šifrovanih blokova. Biham i Shamir navode da je potrebno 200 šifrovanih blokova da se, jednobitnim promenama ključa, dođe do vrednosti tajnog DES ključa. Broj potrebnih blokova značajno ne poraste ako se umesto DES algoritma koristi DES-3 algoritam (168-bitni ključ).

U stvarnosti nije tako jednostavno izvesti navedeni napad. Napad podrazumeva promenu jednog bita ili što manji broj promenjenih bitova. Ako se celi mikrokontroler izloži zračenju, u većini slučajeva toliko će bitova biti promenjeno da će se u potpunosti prekinuti rad procesora. Umesto zračenja procesor se može ostaviti na produkciju izolovanih računskih grešaka uvođenjem specijalnih pikova u signal napajanja ili u signal takta. Ako filter navedenih ulaznih signala ne može neutralizovati pikove, oni mogu proizvesti željene greške.

Najjednostavniji način odbrane od Bellcore ili DFA napada je izvođenje šifarskog algoritma dvaput zaredom. Nakon toga sledi upoređivanje rezultata. Ako su rezultati identični, znači da nije postojao spoljni pokušaj promene bitova. Postupak upoređivanja podrazumeva da namerno izazvane raspršene greške neće promeniti isti bit dvaput zaredom. To je realna pretpostavka, jer sa mogućnošću ciljane promene specifičnog bita, mogući bi bili puno jednostavniji i brži napadi nego što je DFA napad. Problemi se mogu javiti zbog dodatnog vremena potrebnog za dvostruko izvođenje šifarskog algoritma. To se pre svega odnosi na vremenski zahtevne asimetrične šifarske algoritme kao što su RSA i DSA.

Još jedan uspešan način odbrane od DFA napada postiže se dodavanjem slučajnog broja, u obliku prefiksa ili sufiksa, na blok podataka namenjen za šifrovanje. Time se šifruju uvek različiti blokovi podataka što onemogućava DFA napad.

Nedugo nakon objavljivanja DFA napada zaštićeni su svi operativni sistemi pametnih kartica i pripadajuće aplikacije kartica tako da ni Bellcore ni DFA napad ne predstavljaju pretnju današnjim sistemima kartica.

2.4.9. Zaštitni elementi operativnog sistema pametne kartice

Zaštitni mehanizmi operativnig sistema pametne kartice temelje se na zaštitnim mehanizmima hardvera mikrokontrolera kartice. Tri osnovne komponente sigurnosnih mehanizama – hardver, operativni sistem, aplikacija – čine lanac sigurnosti pametne kartice. Najslabija karika određuje snagu celog lanca, ako se prekine jedna karika lanca puca celi lanac. Potencijalne slabosti operativnog sistema mogu narušiti kompletnu sigurnost pametne kartice.

2.4.9.1. Zaštita: hardverski i softverski testovi

Nakon inicijalizacije operativnog sistema potrebno je izvršiti barem najosnovnije hardverske testove. Nekoliko najosnovnijih testova predstavlja testiranje RAM memorije, provera kontrolnih suma važnijih delova ROM memorije, implicitno testiranje ATR signalom i jednostavni eksplicitni testovi CPU-a.

2.4.9.2. Zaštita: slojevitost operativnog sistema

Operativni sistem pametnih kartica dizajnira se prema OSI referentnom modelu sa tri karakteristična sloja: OSI sloj I – fizički sloj, OSI sloj II – sloj veze i OSI sloj VII – aplikativni sloj. Ponašanja i interakcije unutar i između slojeva specifikuj međunarodni standardi.

Takav dizajn operativnog sistema rezultuje minimizacijom grešaka unutar operativnog sistema. Razdvajanje slojeva onemogućava da se greška unutar jednog sloja proširi na ostale slojeve.

2.4.9.3. Zaštita: kontrola prenosa podataka

Kompletna komunikacija sa pametnom karticom, ujedno i jedini dozvoljeni komunikacioni kanal između pametne kartice i ostatka sveta, odvija se preko I/O čitača koji je pod kontrolom operativnog sistema. Kontrolom prenosa podataka na efikasan način štiti se memorija pametne kartice. Transportni manager operativnog sistema upravlja prenosnim protokolom i sprečava procesiranje neispravnih ulaznih vrednosti. On onemogućava da se manipulacijom prenesenih blokova utiče na proces prenosa podataka na način da se bez ispravne autorizacije pošalju terminalu podaci iz memorije.

2.4.9.4. Zaštita: kontrolna suma memorijskog sadržaja

Sva memorijska područja u EEPROM-u, bitna za ispravan rad operativnog sistema, moraju se zaštititi kontrolnim sumama. Prilikom svakog pristupa tako zaštićenom memorijskom području, odnosno pre izvršavanja sadržanog koda, proverava se nepromenljivost njegovog sadržaja. Takođe zahteva se da se struktura datoteka, odnosno zaglavlja datoteka, zaštite kontrolnim sumama. Taj zahtev je izuzetno važan

zbog objektno-orijentirane kontrole pristupa gde se uslovi pristupa datoteci čuvaju u zaglavlju dotične datoteke.

2.4.9.5. Zaštita: ukalupljivanje aplikacija

Neki operativni sistemi koriste u izolovanju različitih aplikacija postupak ukalupljivanja. Aplikacije i pripadajuće datoteke ukalupljuju se u posebne DF-ove. Nivo zaštite nije visok jer je izolacija postignuta na programskom nivou, ali takvom se zaštitom bar sprečava da posledice memorijske greške izađu van granica dotičnog DF-a.

Današnje pametne kartice ne sadrže memorijsku upravljačku jedinicu – MMU. Buduće pametne kartice sigurno će imati MMU jer će se time omogućiti operativnom sistemu da na nivou hardvera izoluje postojeće aplikacije.

2.4.9.6. Zaštita: maskiranje stvarnih aktivnosti operativnog sistema

Prilikom svakog upisivanja podataka u EEPROM potrebno je aktivirati strujni izvor. Time se povećava strujna potrošnja čipa, što se spolja vrlo lako detektuje jednostavnim merenjem. Operativni sistem pametne kartice ne sme dozvoliti da napadač na osnovu spoljnih merenja strujne potrošnje donosi ispravne zaključke o procesima koji se izvode unutar pametne kartice. Kad bi to bilo dozvoljeno napadač bi mogao na osnovu spoljnih merenja zaključiti kakav je rezultat upoređivanja PIN-a pre nego što se završi izvođenje date naredbe i time na vrlo jednostavan način, ponavljanjem navedenog procesa, doći do stvarne vrednosti sačuvanog PIN-a.

2.4.9.7. Zaštita: objektno-orijentisana kontrola pristupa

Nedostatak centralizovane kontrole pristupa je što se pojava softverske ili memorijske greške reflektuje na celoukupnu sigurnost pametne kartice. Moderni objekno-orijentisani sistemi datoteka, sa dozvolama pristupa sačuvanim u zaglavljima datoteka, izoluju uticaj memorijske greške na samo jednu datoteku. Time je sigurnost ostalih datoteka netaknuta. Raspodeljenom kontrolom pristupa značajno se podiže celoukupni nivo sigurnosti pametne kartice, ne samo s obzirom na moguće greške, već i s obzirom na potencijalne napade.

2.4.9.8. Napad i odbrana: generatori slučajnih brojeva

Pametna kartica generiše slučajne brojeve i koristi ih prilikom autentifikacije i individualizacije sesija. Time se postiže da se podaci dobijeni prisluškivanjem prethodnih sesija ne mogu uspešno primeniti na sledeću sesiju. Drugi tip napada bi bio da se pusti pametnoj kartici da generiše slučajne brojeve sve dok njihov sled ne postane predvidljiv. Još jedna mogućnost napada je da se uzastopnim potraživanjem slučajnih brojeva izazove ispad EEPROM memorije generatora slučajnih brojeva kartice (zbog ograničenog broja ciklusa pisanja EEPROM memorije), tako da generator počne generisati uvek isti slučajni broj.

Bilo koji od navedenih napada, ako je uspešno izveden, zaobići će autentifikaciju terminala od strane pametne kartice. Razlog je to da se kod današnjih pametnih kartica posvećuje velika pažnja dizajnu i zaštiti generatora slučajnih brojeva. Njegov rad se uvek pomno ispituje tako da današnji operativni sistemi u potpunosti onemogućuju izvođenje opisanih napada. Niz slučajnih brojeva toliko je dugačak da se nikad za vreme celog životnog ciklusa kartice neće pojaviti dva ista slučajna broja. Ako se izazove ispad EEPROM memorije, blokiraće se generisanje slučajnih brojeva i time sprečiti dalja autentifikacija.

2.4.10. Zaštitni elementi aplikacije pametne kartice

Zaštitni mehanizmi aplikacije temelje se i zavise od ispravnosti rada odgovarajućih zaštitnih mehanizama hardvera i operativnog sistema kartice. Ako je npr. moguće nekom analitičkom procedurom pročitati sadržaj EEPROM memorije, ni najsloženiji i najsigurniji šifarski algoritam neće biti od koristi, s obzirom da napadač može pročitati tajne ključeve direktno iz EEPROM memorije. Nezavisno od toga, aplikacija kartice mora biti tako dizajnirana da napad na pojedinačnu pametnu karticu ne dovodi u pitanje sigurnost celog sistema.

2.4.10.1. Zaštita: jednostavnost mehanizama zaštite

Jednostavni mehanizmi se lakše implementiraju, jednostavnije je proveriti ispravnost njihovog rada i efikasnost postignute zaštite. Izuzetno je pogrešna pretpostavka da se složenim mehanizmom uvek postiže viši nivo zaštite. Naprotiv, sa povećanjem nivoa kompleksnosti raste mogućnost greški i sigurnosnih propusta.

U osnovi, aplikacija bi morala uvek koristiti zaštitne mehanizme operativnog sistema. Pouzdanost njihovog rada je testirana i optimizovana. Optimizovanost je često bitna s obzirom na ograničene resurse pametne kartice. To ne znači da aplikacija ne sme imati svoje vlastite zaštitne mehanizme, već da se nadogradnja zaštite treba uvek temeljiti na mehanizmima već prisutnim u operativnom sistemu.

2.4.10.2. Zaštita: konzervativne dozvole pristupa

Pristup datotekama i naredbama mora biti maksimalno ograničen. Inicijalno treba zabraniti pristup, a prema potrebi dodeljivati dozvole. Time se smanjuje atraktivnost napada jer za svaku dodatnu informaciju napadač mora uložiti dodatni napor.

2.4.10.3. Zaštita: mašina stanja

Napad na pametnu karticu značajno se otežava ako se onemogući da se određena naredba izvodi u bilo koje vreme i u neograničenom broju pokušaja. Upravo se korišćenjem mašine stanja određuju dopušteni nizovi naredbi. Npr. autentifikacija se može prisilno nametnuti postavljanjem prefiksa na sve dozvoljene nizove naredbi, gde je

prefiks predstavljen nizom naredbi potrebnih za obostranu autentifikaciju terminala i pametne kartice.

2.4.10.4. Zaštita: dvostruka kontrola pristupa

Kontrola pristupa datotekama pametne kartice sprovodi se sa dva mehanizma. Dozvolama pristupa sačuvanim unutar datoteka i mašina stanja koja definiše dopušteni niz naredbi i dopuštene parametre tih naredbi. Sve druge naredbe i nedopuštene nizove podržanih naredbi mašina stanja blokira.

2.4.10.5. Zaštita: siguran prenos podataka

Često se prenos podataka odvija u potencijalno nesigurnoj okolini. Relativno jednostavnim tehničkim modifikacijama odnosa između terminala i kartice moguće je obrisati ili umetnuti željene podatke u normalan tok podataka između kartice i terminala. Da se spreči takav tip napada koristi se procedura sigurne komunikacije, ali treba imati na umu da je reč o dvostrukoj enkripciji, što smanjuje brzinu prenosa. Zbog toga se izbegava potpuna enkripcija svih podataka. Šifruju se tajni podaci, a za osiguranje integriteta podataka koristi se MAC procedura.

2.4.10.6. Zaštita: funkcije oporavka od greške

Sesija se može prekinuti zbog niza nedefinisanih razloga. Pritom se javlja potreba za informacijama koje mogu pomoći u rasvetljavanju i raščišćavanju nastale situacije. Te informacije su sadržane u aplikativnim dnevnim datotekama. Njih redovno, za vreme sesije, ažurira operativni sistem i u njih kopira trenutno stanje aplikacije. Dnevni podaci čuvaju se u cikličnoj datoteci. Prepunjavanjem ciklične datoteke pristigli najnoviji zapis briše najstariji zapis u datoteci.

Detaljne dnevne datoteke pametne kartice omogućuju određene funkcije oporavka od greške, kao što je automatska obnova na pređašnje stanje kartice u slučaju nedefinisanog prekida sesije.

2.4.10.7. Zaštita: autentifikacija

Sistemi kartca sa magnetnim karticama koriste jednosmernu autentifikaciju, i to samo u svrhu provere izvora magnetne kartice. Magnetna kartica je pasivni deo sistema i ne može proveriti indetitet terminala. Situacija se menja dolaskom pametnih kartica u sistemu kartica. Sada kartica može proveriti identitet terminala i aktivno sprečiti pokušaje neautorizovanog pristupa. Time se značajno podiže nivo sigurnosti celoukupnog sistema kartica.

Pametna kartica korišćenjem mogućnosti obostrane autentifikacije terminala i pametne kartice sprečava bilo kakvu analizu svog operativnog sistema ako on nije u tom trenutku aktivni deo kompletnog sistema kartice.

2.4.10.8. Zaštita: on-line ponašanje

Terminali sa ugrađenim sigurnosnim modulima mogu potpuno samostalno izvoditi aplikacije sa pametnim karticama. Periodično se uspostavlja direktna komunikacija sa pozadinskim sistemom da se učitaju potrebni podaci, kao što je npr. crna lista kartica ili novi skup autentifikacionih ključeva. Učestalost uspostavljanja te on-line veze raste sa kompleksnošću sistema i mogućnošću potencijalnih napada. Određivanje trenutka kada terminal treba uspostaviti on-line komunikaciju može se izvesti na više načina. Najjednostavniji način je da se postave periodični vremenski okidači u terminalima ili vremenski okidači koji se temelje na slučajnim varijablama (promenljivima). Sistemi kartica plaćanja za uspostavljanje on-line veze koriste brojač transakcija, sačuvan na pametnim karticama. Npr. nakon svake desete transakcije, dotične pametne kartice, izvrši se on-line konekcija i izvede obostrana autentifikacija pametne kartice i pozadinskog sistema. Na okidač izvedenog broja off-line transakcija dodatno se može postaviti da transakcija, čiji iznos prekorači određenu vrednost, takođe pobudi on-line vezu.

2.4.10.9. Zaštita: crne liste

Sistemi kartica moraju implementirati mehanizme zaštite koji će štiti sistem od mogućnosti pojave krivotvorenih (falsifikovanih) pametnih kartica. Takođe sistem mora zaštiti stvarne korisnike od mogućih posledica proizašlih krađom njihovih pametnih kartica.

Da bi se sprečilo korišćenje krivotvorenih ili ukradenih kartica potrebno je održavati listu dozvoljenih ili listu zabranjenih kartica. Liste se temelje na nekom jedinstvenom obeležju kartice, npr. na serijskom broju čipa kartice. Dizajn sistema u kojme sve što nije eksplicitno dozvoljeno - implicitno je zabranjeno, postiže najveći nivo zaštite. Takav sistem podrazumeva održavanje liste dozvoljenih kartica, tzv. bele liste.

Kod velikih sistema kartica bela lista postaje prevelika. Npr. za sistem sa 10 miliona pametnih kartica i 8-bajtnim serijskim brojem bela lista znači 80MB podataka, 80 MB podataka koje treba redovno ažurirati i on-line prenositi do terminala. To je razlog zašto veliki sistemi kartica održavaju crne liste, liste nedozvoljenih kartica. Crna lista za gore navedeni primer, uz podatak prosečne vrednosti 1% blokiranih kartica, iznosi 800kB podataka.

U on-line sistemima pametne kartice se proveravaju u realnom vremenu. Sistemi koji delimično ili u potpunosti rade off-line moraju što češće prenositi ažurirane crne liste do svojih terminala.

2.4.10.10. Napad i odbrana: virusi i trojanski konji

Sve do nedavno, računarski virusi su bili nepoznanica u svetu pametnih kartica. Do promene je došlo kad su moderni operativni sistemi pametnih kartica implementirali

mehanizme dinamičkog učitavanja programskog kôda. Takav programski kôd više nije pod kontrolom izdavača kartice jer se učitava u fazi korišćenja kartice. To je stvorilo potrebne uslove za pojavu računarskih virusa. Pod definicijom virusa podrazumeva se program koji može sam sebe umnožiti i proširiti na ostale računare. Ako takav program nema moć umnožavanja naziva se Trojanski konj. Zajedničko za oba programa je da pod određenim okolnostima mogu izvršiti neautorizovane akcije na računaru domaćina. U slučaju pametnih kartica te akcije bi mogle uključivati i slanje vrednosti tajnih ključeva izvan kartice.

Pametna kartica, za razliku od običnog personalnog računara, poseduje sigurnosne mehanizme blokiranja neautorizovanog izvođenja programa. Programski kôd se štiti digitalnim potpisom ili MAC-om. Operativni sistemi izoluju memorijska područja pojedinačnih aplikacija kartica, hardverski ili softverski, i time onemogućuju da slučajno učitani virus ili trojanski konj ugrozi funkcionalnost i sigurnost prisutnih aplikacija.

2.4.10.11. Napad i odbrana: prostor tajnog ključa

Jedan od mogućih tipova napada na šifarskom nivou je i detaljno pretraživanje prostora tajnog ključa. Prostor tajnog ključa se definiše kao broj mogućih vrednosti, odnosno kombinacija bitova, s obzirom na zadatu dužinu ključa, npr. 56 bitni ključ ima prostor veličine 2^{56} mogućih vrednosti. Preduslov je da napadač poseduje barem jedan šifarski par bloka (orginalni blok i njemu odgovarajući šifrovani blok) i da mu je poznat primenjeni šifarski algoritam. Šifrovanjem originalnog bloka sa svakom mogućom vrednošću unutar prostora ključa i upoređivanjem dobijenog rezultata sa drugim delom para, može se pronaći prava vrednost napadnutog ključa.

U sprečavanju ovog tipa napada koristi se nekoliko različitih pristupa. Najjednostavnija odbrana je da se toliko poveća prostor ključa da napadač sistemskom pretragom ne može doći do željenog rezultata u prihvatljivom vremenu. To je razlog što sve više aplikacija umesto DES algoritma koristi 3-DES algoritam. Prostor DES ključa je uz trenutno dostupnu procesorsku snagu postao premali.

Drugi način odbrane je da se protokol autentifikacije i komunikacije između kartice i terminala dizajnira na način da se šifarski parovi blokova uopšte ne pojavljuju. Osim toga, u praksi se najveći deo razmenjenih podataka ne šifruje, već se štiti MAC-om, a MAC je puno otporniji na napade izvedene sirovom procesorskom snagom.

Još jedan način odbrane je da pametna kartica pre procesa šifrovanja ili izračunavanja MAC-a generiše slučajni broj i smesti ga na početak originalnog bloka podataka. Rezultat je da u proces šifrovanja nikad ne ulazi isti blok podataka. Time je brutalna pretraga ključa otežana jer slučajni broj najčešće nije javan, on može biti tajna koju dele terminal i pametna kartica. Opisana procedura korišćenja slučajnog broja pruža dobru zaštitu od DFA i DPA napada, čak i kad je slučajni broj javno poznat.

Atraktivnost napada se smanjuje korišćenjem dinamičkih ključeva, različitih za svaku operaciju šifrovanja, ili korišćenjem sesijskih ključeva, različitih za svaku sesiju.

Ako napadač kojim slučajem uspe otkriti primenjeni ključ tu informaciju neće moći iskoristiti za buduće napade jer je već sledeća transakcija šifrovana drugim ključem.

Deo III

3. Standardi

Kao i na drugim tehničkim područijima i kod pametnih kartica postoje standardi koji opisuju njihove osobenosti. Nedostatak standarda prvih godina razvoja se ogleda u nekompatibilnosti kartica pojedinih proizvođača mada se situacija poslednjih godina poboljšava.

Postoji čitav niz standarda koji se odnose na pametne kartice. Oni koji se odnose na njihove fizičke karakteristike su generalno usaglašeniji od onih za softversku arhitekturu. ISO 7816 pod punim imenom ***I**ntegrated **C**ircuit **C**ards with **E**lectrical contacts* je međunarodni standard za mikroprocesorske kartice sa kontaktima. To je ujedno najšire prihvaćen standard vezan uz pametne kartice. Osim njega vrlo su rašireni i **ETSI** (engl.***E**uropean **T**elecommunications **S**tandards **I**nstitute*) standardi koji određuju **SIM** (engl.***S**ubscriber **I**dentity **M**odule*) kartice u mobilnim telefonima.

3.1. Standard ISO 7816

***I**nternational **O**rganization for **S**tandardization* izdala je standard ISO 7816 «*Identification cards – Integrated circuit cards with contacts*». On definiše najvažnije karakteristike kontaktnih čip kartica:

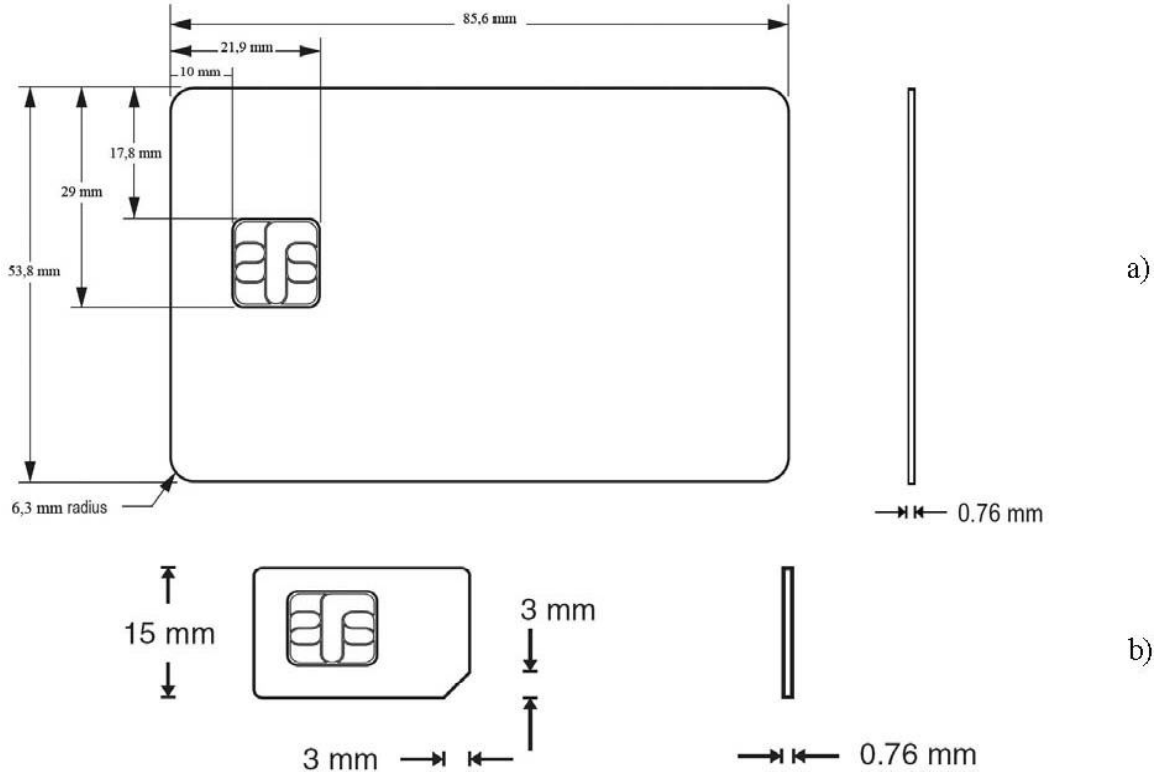
1. deo – fizičke karakterisitike,
2. deo – dimenzije i položaj kontakata,
3. deo – električni signali i transportni protokoli,
4. deo – međuindustrijske naredbe,
5. deo – identifikator aplikacije ,
6. deo – međuindustrijski elementi podataka,
7. deo – međindustrijske SCQL naredbe,
8. deo – međuindustrijske komande za sigurnosne operacije,
9. deo – međuindustrijske komande za upravljanje karticom,
10. deo – električni signali i odgovor na reset za sinhronizaciju kartice,
11. deo – lična verifikacija putem biometričkih metoda (u razvoju).

U nastavku će biti podrobnije opisani bitniji delovi.

3.1.1. ISO 7816-1

Prvi deo se odnosi na fizičke karakteristike pametnih kartica. On propisuje dimenzije i otpornost kartica na spoljne uticaje. Njime je definisana i otpornost kartica na

mehanička naprezanja kako bi se osiguralo da proizvedene kartice besprekorno rade tokom celog svog životnog veka. Proizvođači pametnih kartica su ti koji izbor materijala i proizvodnih procesa moraju prilagoditi kako bi kartice bile u skladu sa standardom. Fizička veličina pametne kartice po standardu 7816-1 temelji se na veličini identifikacione kartice ID–1 iz standarda ISO 7810.



Slika 3.1. a) ID-1 format b) ID-000 format

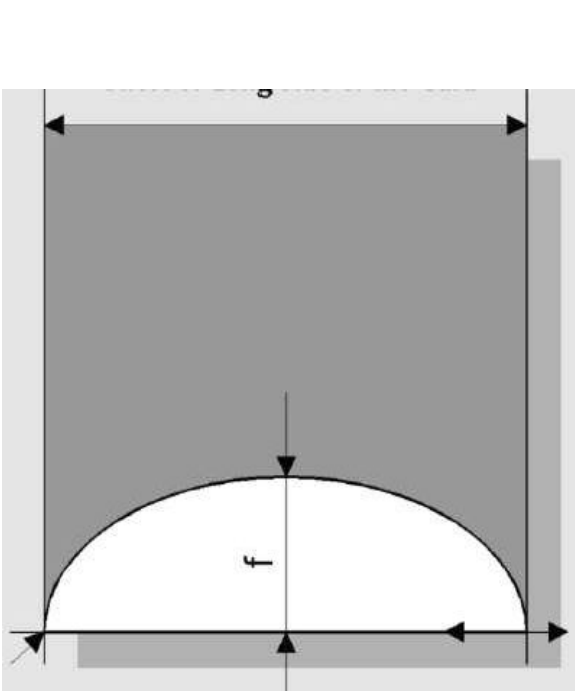
Dodatne karakteristike propisane ovim standardom su:

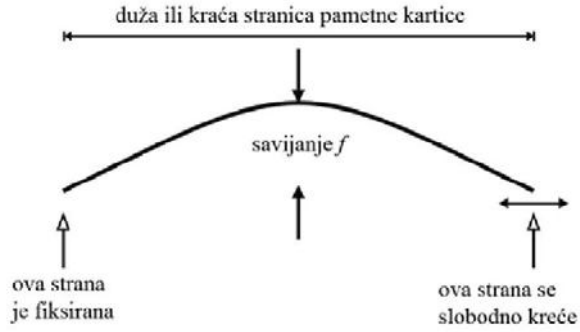
- Otpornost na ultra ljubičasto svetlo,
- Otpornost na X – zrake,
- Površinski profil kontakata,
- Mehanička čvrstoća kartice i kontakata,
- Električni otpor kontakata,
- Elektromagnetna interferencija između eventualne magnetne trake i čipa,
- Otpornost na elektromagnetna polja,
- Otpornost na statički elektricitet,
- Disipacija toplote.

U aneksu standarda su opisana i tri najčešće korišćena ispitna postupka od strane proizvođača:

- Savijanje kartice,
- Torzija kartice,
- Statički elektricitet.

Pri ispitivanju savijanja, kartica se savija uz f = 2 cm po dužoj osi odnosno f = 1 cm po kraćoj prema slici. Savijanje se obavlja 30 puta u minuti a preporučuje se da kartica izdrži 250 savijanja u svakom od četiri moguća smera.

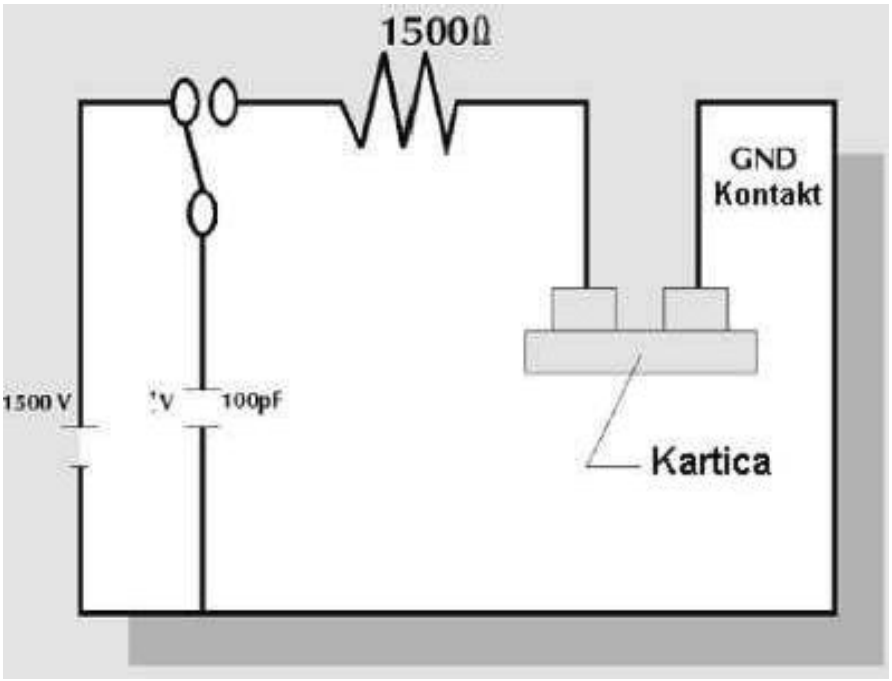






Slika 3.2. Savijanje kartice i mašina za testiranje kartica

Pri ispitivanju reakcije na torziju kartica se uvija za +/-15° po dužoj osi 30 puta u minuti. Mora izdržati 1000 uvijanja bez oštećenja čipa ili vidljivih pukotina (naprsnuća). Otpornost na statički elektricitet ispituje se postupkom prikazanim na slici 3.3. Napon testiranja je 1500 V, a kondenzator mora biti ispražnjen kroz svaki kontakt u oba smera. Nakon ispitivanja kartica mora normalno funkcionisati.

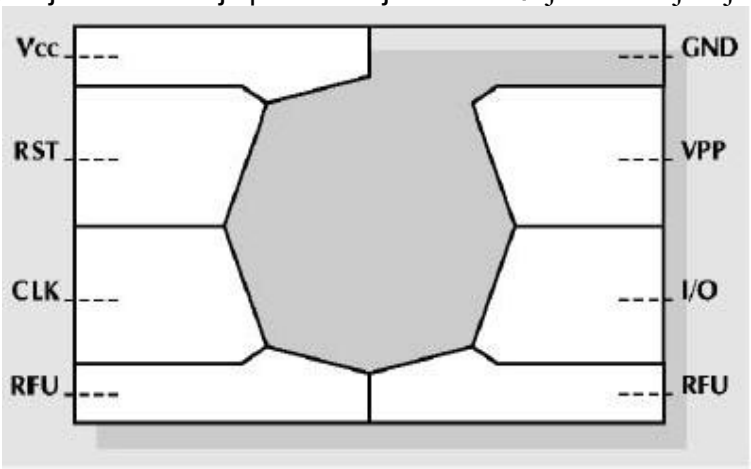


Slika 3.3. Ispitivanje otpornosti na statički elektricitet

Kartice moraju biti upotrebljive na temperaturama od –35 ºC do 50 ºC.

3.1.2. ISO 7816-2

U ovom delu standarda definisane su dimenzija i lokacija kontakata. Određeno je da ih ima osam, a njihova funkcija prikazana je na sledećoj slici i objašnjena u tabeli.



Slika 3.4. ISO 7816-2 konektor

Kontakt	Značenje
V_{cc}	Kontakt za napajanje kroz kojise struja dovodi mikroprocesoru na kartici.
RST	Kontakt za slanje *reset* signala mikroprocesoru.
CLK	Kontakt za signal vremenskog vođenja koji određuje brzinu komunikacije čitača i kartice. Najčešće frekvencije su 3.57 MHz i 4.92 MHz.
RFU	Rezervisano za buduću upotrebu (engl.*Reserved for Future Use*).
GND	Uzemljenje (engl.*Ground*).
V_{pp}	Kontakt za napajanje kartice dodatnom strujom u slučaju programiranja ili brisanja EEPROM-a. Više se ne koristi.
I/O	Kontakt za serijski prenos podataka (engl.*Input/Output*).

Tabela 3.1. Kontakti po ISO 7816-2 standardu

Standardom je propisano da konektori moraju biti na gornjoj strani kartice, pri čemu je zadnja strana (poleđina) ona koja sadrži magnetnu traku. Eventualna ispupčenja moraju se nalaziti na prednjoj strani kartice.

3.1.3. ISO 7816-3.1

Ovaj deo standarda specifikuje elektronske signale i komunikacione protokole pametnih kartica. Da bi se prikazale električne karakteristike signala potrebno je uvesti sledeće skraćenice:

Skraćenice	Značenje
V_{ih}	Ulazni napon visokog nivoa
V_{il}	Ulazni napon niskog nivoa
V_{cc}	Napon napajanja
V_{oh}	Izlazni napon visokog nivoa
V_{ol}	Izlazni napon niskog nivoa
t_r	Vreme rasta signala
t_f	Vreme pada signala

Tabela 3.2. Skraćenice i značenje skraćenica

3.1.3.1. Napon i struja napajanja

Standard propisuje tri klase kartica:

Klasa	Napon	Takt	Max. struja
A	5V ± 10%	1-5 Mhz	60 mA uz 5Mhz
B	3V ± 10%	1-5 Mhz	50 mA uz 4Mhz
C	1.8 ± 10%	1-5 Mhz	30 mA uz 4Mhz
Sve tri klase uz			0.5 mA

Tabela 3.3. Naponi napajanja po ISO 7816-3

Klase se mogu upotrebljavati pojedinačno ili u željenoj kombinaciji. Npr.ako kartica zadovoljava klasu A ili klasu B može se koristiti ili sa naponom od 5V ili sa naponom od 3V. Takođe standard nalaže da se kartica nesme oštetiti ako se pokuša koristiti sa pogrešnim iznosom napona.

3.1.3.2. I/O Signal

Za prenos podataka postoji samo jedan kontakt koji može biti korišćen kao ulazni (mod primanja) ili izlazni (mod slanja). Njegov signal može biti u dva moguća stanja:

- stanje visokog nivoa (stanje Z), koje se javlja u slučaju da su i kartica i čitač u stanju primanja ili usled slanja odgovarajućeg podatka,
- stanje niskog nivoa (stanje A), javlja se usled slanja odgovarajućeg podatka.

Veličina	Mogućnosti		Minimum	Maksimum	Jedinica
V_{ih}	ili	$I_{ih\ max}$= ± 500μA	2	V_{cc}	V
		$I_{ih\ max}$= ± 50μA	0.7 V_{cc}	V_{cc}	V
V_{il}	Iil max = 1mA		0	0.8	V
V_{oh}	ili	$I_{ol\ max}$= ± 100μA	2.4	V_{cc}	V
		I_{ol} max= ± 20μA	3.8	V_{cc}	V
V_{ol}	$I_{ol\ max}$ = 1mA		0	0.4	V
t_r, t_f	C_{in} = 30pF; C_{out} = 30pF			1	μs

Tabela 3.4. Električne vrednosti I/O signala po ISO 7816-3
(Čitačmora imati mogućnost rada sa obe vrednosti. Visok nivo se osigurava pull-up otpornikom (preporučena vrednost 20kΩ)

3.1.3.3. Signal vremenskog vođenja

Uobičajno je da mikroprocesori na pametnim karticama nemaju svoj signal vremenskog vođenja već da im on biva nametnut od strane čitača. Budući da je komunikacija kartice i čitača određena njegovom frekvencijom na kartice se dodaje delilac frekvencije kako bi se postigla brzina prenosa podataka od 9600 b/s.

Veličina	Mogućnosti		Minimum	Maksimum	Jedinica
V_{ih}	ili4	$I_{ih\ max}$ =±200 µA	2.4	V_{cc}	V
	ili	$I_{ih\ max}$ =±20 µA	0.7 V_{cc}	V_{cc}	V
	ili	$I_{ih\ max}$ =±10 µA	V_{cc} -0.7	V_{cc}	V
V_{il}	$I_{il\ max}$ = ± 200µA		0	V_{cc}	V
t_r, t_f	C_{in} = 30pF		9% perioda uz maksimum 0.5		µs

Tabela 3.5. Električne vrednosti CLK signala po ISO 7816-3

3.1.3.4. Reset signal

Signal za reset dolazi od čitača i služi pokretanju programa sadržanog na kartici. Standard specifikuje tri načina resetovanja. Interni reset, reset sa aktivnom niskim naponskim nivoom i sinhroni reset sa aktivnim visokim naponskim nivoom. Mikroprocesorske kartice najčešće koriste reset sa aktivnim niskim naponskim nivoom pri kojem mikroprocesor počinje izvršavati program od prve adrese kada reset signal skoči na visok naponski nivo. Sinhroni reset se češće koristi kod kartica u telekomunikacionim primenama.

Veličina	Mogućnosti		Minimum	Maksimum	Jedinica
V_{ih}	ili	$I_{ih\ max}$ =±200 µA	4	V_{cc}	V
		$I_{ih\ max}$ =±10 µA	V_{cc} -0.7	V_{cc}	V
V_{il}	$I_{il\ max}$ = ± 200 µA		0	0.6	V

Tabela 3.6. Električne vrednosti RST signala po ISO 7816-3

3.1.4. ISO 7816 3.2

Ovaj deo standarda opisuje redosled operacija prilikom komunikacije kartice sa čitačem:
- umetanje kartice u čitač i aktiviranje kontakata,
- reset kartice,
- odgovor kartice na reset,
- razmena informacija između kartice i čitača,
- deaktivacija kontakata i vađenje kartice.

3.1.5. ISO 7816 3.2.a Spajanje i aktiviranje kontakata

Aktivacija se sastoji od sledećih operacija:
- RST je u stanju *low,*
- V_{cc} je uključen,
- I/O je u stanju primanja,
- V_{PP} je u stanju *idle*,
- CLK daje prikladni i stabilni takt

3.1.6. ISO 7816 3.2.b Reset kartice

Na kraju procedure aktiviranja kontakata kartica je spremna za reset. Nakon što je na liniji CLK uspostavljen signal takta (T_0), I/O linija će biti postavljena u stanje visoke impedanse u roku od 200 ciklusa takta (t_2).

Odgovor na reset mora početi između 400 i 4000 ciklusa takta nakon uspostavljanja signala(t_1 nakon T_0), tokom kog vremena RST signal mora biti na niskom nivou. Ako se odgovor na reset ne dogodi u tom vremenskom okviru RST se podiže na visok nivo (T_1). Uz takvo stanje ponovo se očekuje odgovor na reset između 400 i 4000 ciklusa takta. Ukoliko odgovora nema kontakti se deaktiviraju od strane čitača (T_2).

3.1.7. ISO 7816 3.2.c Deaktivacija kontakata

Nakon završetka prenosa podataka ili nedostatka odgovora na reset sledi deaktivacija kontakata i vađenje kartice iz čitača. To se obavlja sledećim redosledom:
- spustiti RST na niski nivo,
- spustiti CLK na niski nivo,
- ugasiti V_{pp},
- stanje A na I/O,
- V_{cc} neaktivan.

3.1.8. ISO 7816 3.3. Odgovor na reset

Predviđeni su dva tipa prenosa podataka:

- Asinhroni prenos - karakteri se prenose asinhrono, svaki znak sadrži jedan bajt.
- Sinhroni prenos - prenose se nizovi bitova sinkronizovano sa signalom vremenskog vođenja.

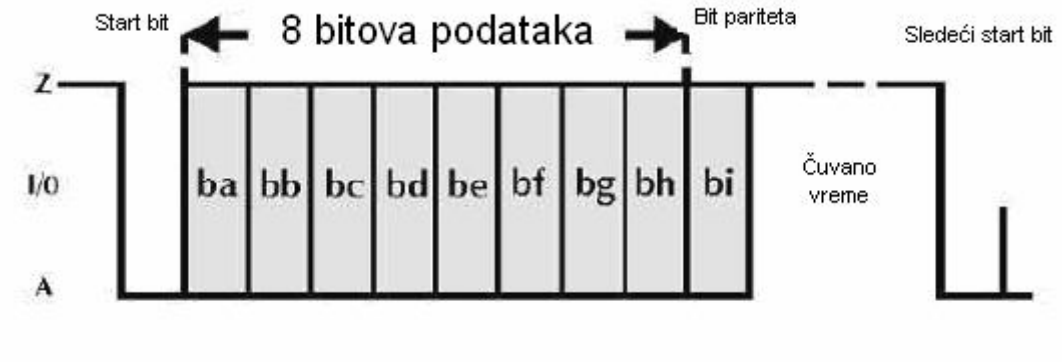
3.1.8.1. ISO 7816 3.3.a Odgovor na reset kod asinhronog prenosa

Trajanje bita na I/O liniji naziva se elementarnom vremenskom jedinicom (engl.***E**lementary **T**ime **U**nit - **ETU***). Za kartice koje koriste spoljni signal vremenskog vođenja postoji linearna veza između ETU-a korišćenog na I/O liniji i perioda tog signala. Inicijalno ETU iznosi 372/fi gde je fi inicijalna frekvencija signala na CLK-u za vreme odgovora na reset. Ona je između 1 i 5 Mhz.

3.1.8.2. Izgled znakova

Znak se sastoji od 10 uzastopnih bitova:

- početni bit (stanje A),
- 8 bitova informacije od **ba** do **bh,**
- bit **bi** za proveru pariteta.



Slika 3.5. Izgled znakova

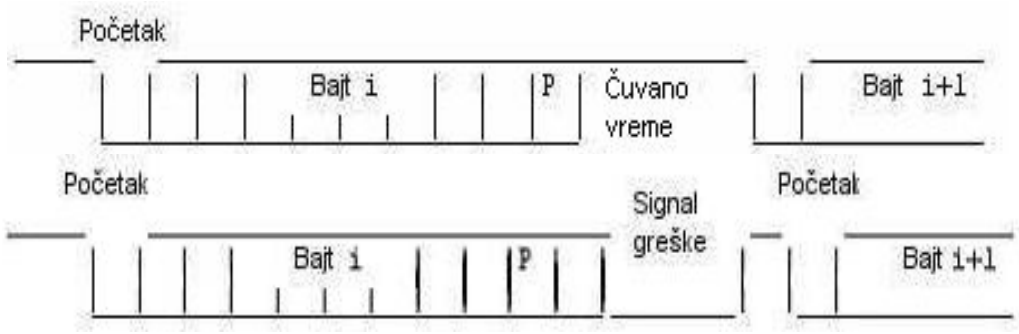
Budući da je prenos podataka asinhron potrebni su bitovi koji će označiti početak i kraj bloka informacionih bitova. Zbog toga znak započinje sa startnim bitom a završava "čuvanim vremenom" (engl. *guard time*) za vreme koga je I/O linija u stanju
Z. Za vreme odgovora na reset pauza vremenski razmak između dva uzastopna znaka nesme premašiti 9600 ETU. To je inicijalno vreme čekanja. Minimalno on mora iznositi barem 12 ETU što uključuje trajanje samog znaka (10 ± 0.2 ETU) + čuvano vreme.
Konvencija koja određuje naponski nivo koji će se smatrati logičkom jedinicom odnosno logičkom nulom i koja određuje hoće li **ba** ili **bi** biti bit sa najvećom težinom je

specifikovana u tzv. inicijalnom znaku TS koji se preuzima iz kartice tokom odgovora na reset. Paritet je tačan kada je broj jedinica u znaku paran.

3.1.8.3. Detekcija grešaka i ponavljanje znaka

Procedura ponavljanja znaka zavisi od tipa protokola. Ta procedura je obavezna za kartice koje koriste protokol T=0, a opcionalna je za čitač kao i za ostale kartice. Procedura je sledeća: čitač ispituje I/O liniju 11±0.2 ETU nakon početnog brida startnog bita.

- Ako je I/O u stanju Z, pretpostavlja se da je prenos bio uspešan,
- Ako je I/O u stanju A, pretpostavlja se da je prenos netačan. Prenos znaka se ponavlja nakon pauze od najmanje 2 ETU-a nakon detekcije signala greške.



Slika 3.6. Detekcija grešaka i ponavljanje znaka

3.1.8.4. Odgovor na reset

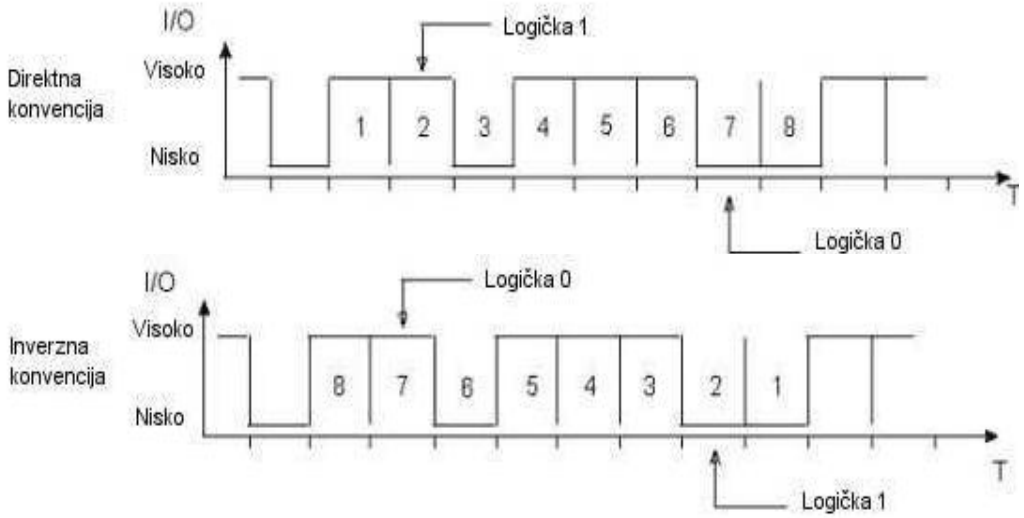
Sam odgovor na reset sastoji se od najviše 33 znaka (uključujući inicijalni) koji se dele u 5 grupa:

- inicijalni znak (TS),
- znak formata (TO),
- znakovi interfejsa (TA,TB,TC,TD),
- "istorijski" (historical) znakovi (T1,T2,TK),
- znak provere (TCK).

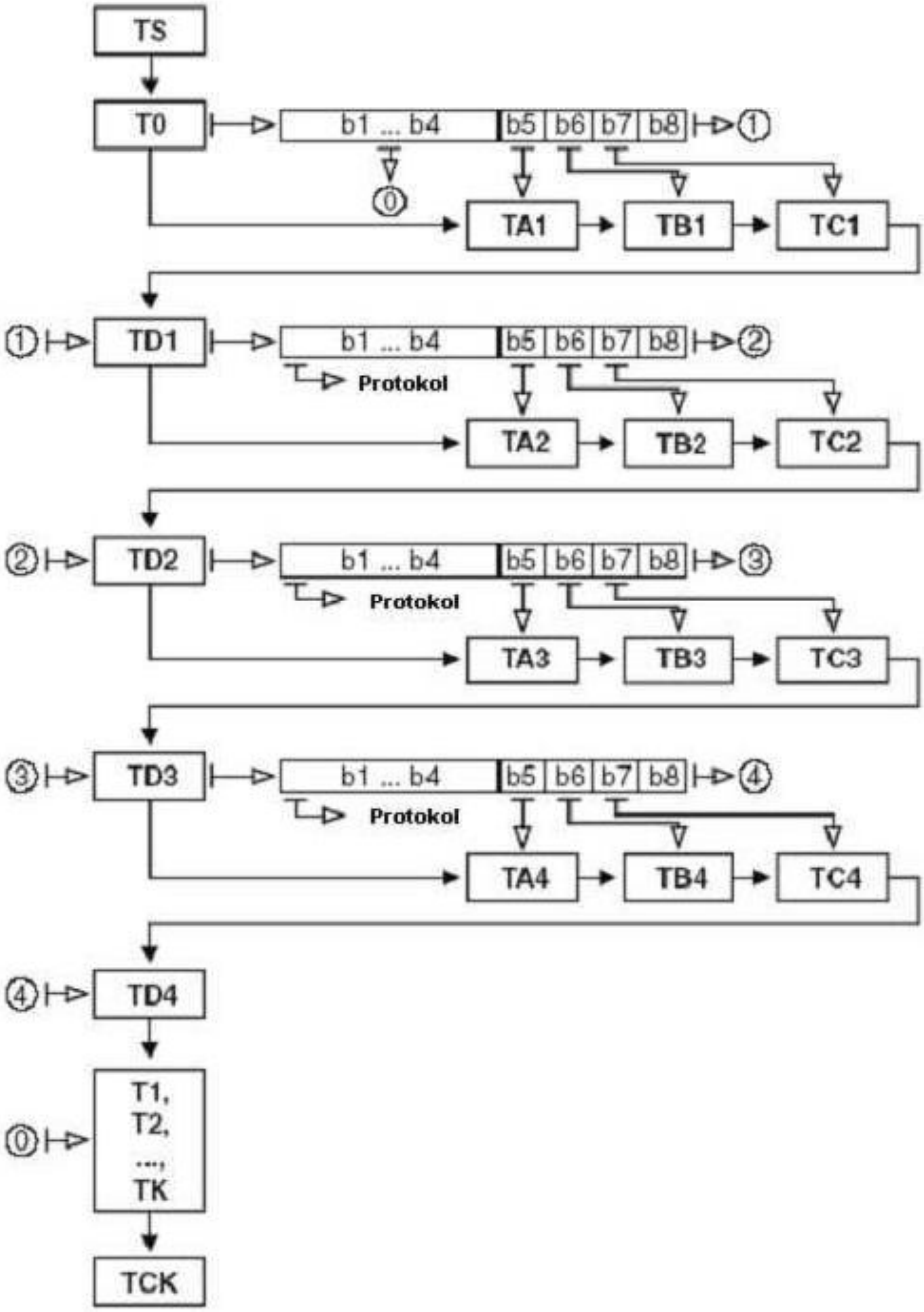
Bitovi u tim znakovima određuju razne parametre komunikacije kao i pojavljivanje samih znakova u odgovoru.

3.1.8.5. Inicijalni znak TS

Pomoću ovog znaka se određuje brzina prenosa podataka, i konvencija o redosledu bitova kao i njihovoj interpretaciji. Postoje dve konvencije direktna i inverzna. Kod inverzne konvencije je niski naponski nivo interpretiran kao logička jedinica, a prvi poslati bit ima najveću težinu. Slike 3.7 i 3.8 prikazuju te konvencije.



Slika 3.7 i 3.8. Konvencija o redosledu bitova i njihova interpretacija



Slika 3.9. Kompletan dijagram odgovora na reset

3.1.8.6. Znak formata T0

Gornja 4 bita ovog znaka govore hoće li znakovi TA1, TB2, TC1,TD1 biti poslati. Ta informacija se čita iz naponskog nivoa pojedinog bita npr. bit sa najvećom težinom b8 u stanju logičke jedinice označava prisutnost znaka TA1. Donja 4 bita određuju broj (0 do 15) historical znakova.

b8	b7	b6	b5	b4	b3	b2	b1	Značenje
...	...	...	...	X				Broj historical znakova
...	...	...	1	...	...	...	...	TA1 prisutan
...	...	1	...	...	...	...	...	TB1 prisutan
...	1	...	...	...	...	...	...	TC1 prisutan
1	...	...	...	...	...	...	...	TD1 prisutan

Tabela 3.7. Kodiranje formatirajućeg znaka T0

3.1.8.7. Znaci interfejsa TAi, TBi, TCi, TDi

TA_i, TB_i, TC_i (i = 1,2,3...) specifikuju parametre protokola. Oni po potrebi mogu biti izostavljeni jer su za sve parametre protokola predviđene default vrednosti. Takođe se mogu podeliti na globalne parametre koji se odnose na sve moguće protokole i na specifične parametre koji se odnose samo na neki određeni protokol. Neki od globalnih parametara odnose se samo na T=0 protokol (iz ranijih razloga), pa se mogu izostaviti ako se taj protokol ne koristi. TD_i znak se koristi kako bi se dobila informacija o pojavljivanju narednih znakova interfejsa npr. TD_1 daje informacije o pojavljivanju TA_2, TB_2 i TC_2 (upotrebljava se isto kodiranje kao kod T0).

b8	b7	b6	b5	b4	b3	b2	b1	Značenje
...	...	...	...	X				Broj komunikacionog protokola
...	...	...	1	...	...	...	...	TA(i+1) prisutan
...	...	1	...	...	...	...	...	TB(i+1) prisutan
...	1	...	...	...	...	...	...	TC(i+1) prisutan
1	...	...	...	...	...	...	...	TD(i+1) prisutan

Tabela 3.8. Kodiranje formatirajućeg znaka TDi

TA1

Ovaj znak definiše globalne parametre FI i DI. FI je delilac takta, a DI je faktor podešavanja brzine prenosa. Te veličine su potrebne kako bi se definisao ETU.

b8	b7	b6	b5	b4	b3	b2	b1	IFSC
		X			...			FI
		...			X			DI

Tabela 3.9. Kodiranje TA1

F	372	372	558	744	1116	1488	1860	RFU
FI	0000	0001	0010	0011	0100	0101	0110	0111
f_{max}	4 MHz	5 MHz	6 MHz	8 MHz	12 MHz	16 MHz	20 MHz	...
F	RFU	512	768	1024	1536	2048	RFU	RFU
FI	1000	1001	1010	1011	1100	1101	1110	1111
f_{max}	...	5 MHz	7.5 MHz	10 MHz	15 MHz	20 MHz	...	...

Tabela 3.10. Interpretacija F1

D	RFU	1	2	4	8	16	32	RFU
DI	0000	0001	0010	0011	0100	0101	0110	0111
D	12	20	RFU	RFU	RFU	RFU	RFU	RFU
DI	1000	1001	1010	1011	1100	1101	1110	1111

Tabela 3.11. Interpretacija D1

Tokom odgovora na reset podrazumeva se inicijalni ETU :

$$inicijalni\ ETU = \frac{372}{fi} s$$

Interval trajanja bita nakon odgovora na reset zove se radni ETU:

$$radni\ ETU = \frac{1}{D} \frac{F}{fs} s$$

TB1

Ovaj parametar se koristi kako bi se odredila naponski nivo potreban za programiranje memorije kartice. Ta vrednost bila je potrebna u prvim karticama koje su koristile EPROM umesto EEPROM memorije koja se koristi danas. Relativno visoki iznosi napona dolazili su na V_{pp} kontakt. Takve kartice više nisu u upotrebi pa je vrednost tog parametra 0.

TC1

Ovaj parametar se takođe svrstava u grupu globalnih. On kodira eventualno dodatno "čuvano vreme". To vreme se dodaje na trajanje stop bita. Vrednost N određuje broj ETU-a koji se dodaje, a ta vrednost se čita direktno iz TC1 bajta. Slučaj kada se on sastoji samo od jedinica (N=FF) je izuzetak. Interpretacija je data u tabeli 3.12.

b8	b7	b6	b5	b4	b3	b2	b1	Značenje
X								Dodatno "čuvano vreme" u rasponu od 0 – 254 ETU
1	1	1	1	1	1	1	1	X=255 & T=0; "čuvano vreme" = 2 ETU-a X=255 & T=1; "čuvano vreme" = 1 ETU

Tabela 3.12. Kodiranje znaka TC1

U praksi vrednost X=255 znači smanjenje "čuvanog vremena" kod protokola T=1 što ubrzava prenos podataka za 10%.

3.1.8.8. Historical znaci

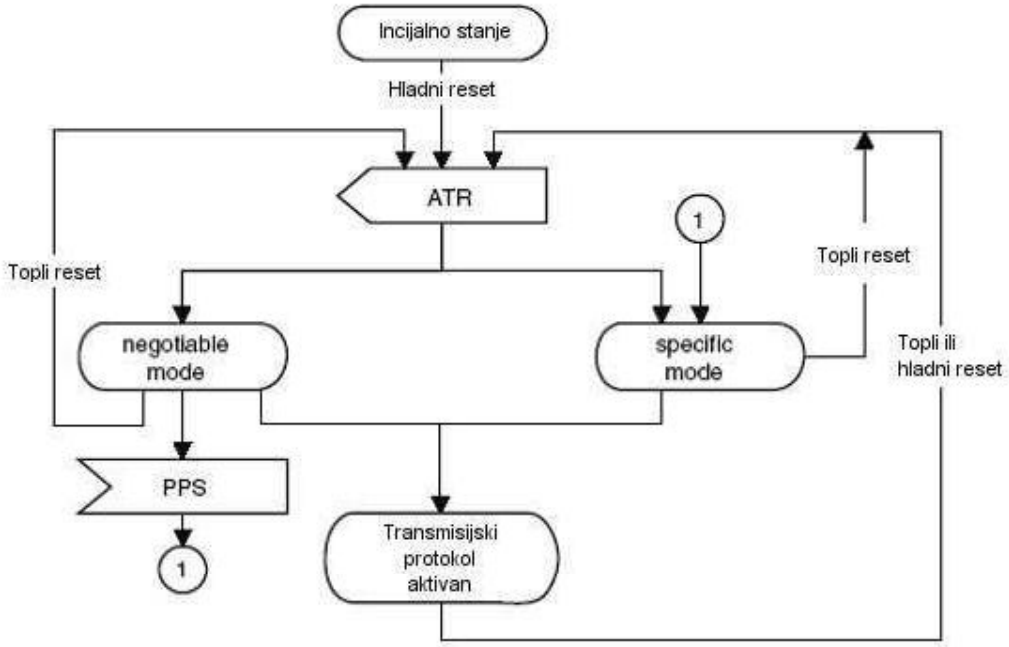
Historical znaci nisu obuhvaćeni standardom i iz tog razloga mogu sadržati širok raspon informacija zavisno od proizvođača pametnih kartica. Neki proizvođači su ih koristili za identifikaciju operativnog sistema i verziju memorije. Oni ne moraju biti prisutni u odgovoru na reset tako da se mogu u potpunosti izostaviti. U **ISO 7816-4** standardu postoji specifikacija za datoteku koja se koristi u odgovoru na reset kao dodatak historical znacima. Njena struktura nije određena standardom, ali ona može sadržati razne složene informacije vezane uz fajl sistem kartice.

3.1.8.9. Znak provere TCK

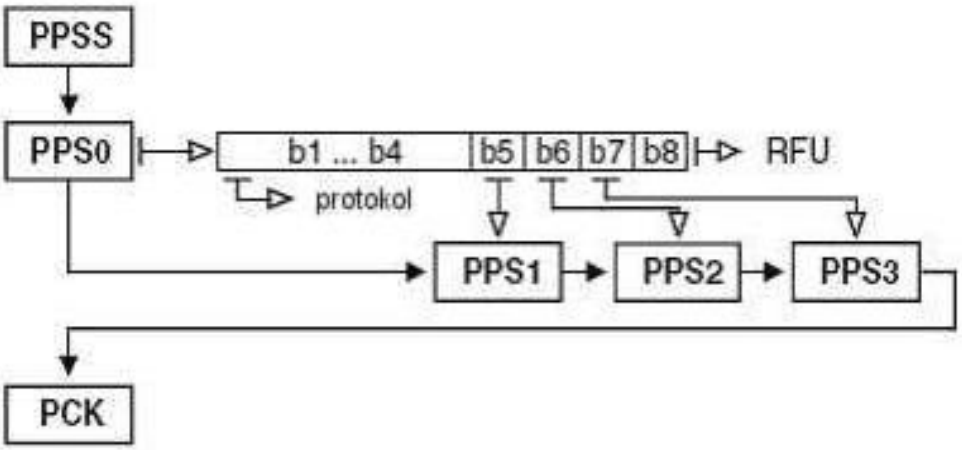
Vrednost ovog znaka mora biti takva da se operacijom ekskluzivno ili nad svim znacima od T0 do TCK dobije nula. Odgovor na reset gotov je 12 ETU-a nakon prvog brida poslednjeg znaka.

3.1.9. ISO 7816 3.4 Selekcija parametara protokola (PPS)

Tokom odgovora na reset pametna kartica specifikuje razne parametre prenosa podataka. Ako čitač želi promeniti neke od njih mora sprovesti selekciju parametara protokola (engl. ***P**rotocol **P**arameter **S**election*) PPS pre nego što počne razmena podataka. Ova procedura se do 1997 g. nazivala selekcija tipa protokola (engl.***P**rotocol **T**ype **S**election*). Ova procedura se može sprovesti na dva načina. Prema jednom (engl.***n**egotiable **m**ode*) standardne vrednosti F i D ostaju nepromenjene dok se PPS uspešno ne završi, a prema drugom (***s**pecific **m**ode*) PPS se izvršava koristeći vrednosti F i D specifikovane u ATR-u. Kartica indicira koji način podržava u znaku TA2. PPS se mora sprovesti odmah nakon odgovora na reset. Ako kartica prihvata zahteve iz PPS-a onda mora vratiti primljene PPS bajtove natrag čitaču. Standard zabranjuje ponovno slanje PPS podataka. U praksi PPS se koristi retko jer se obično parametri kartica i čitača unapred utvrde.



Slika 3.10. Dijagram stanja PPS procedure po 7816-3 standardu



Slika 3.11. Osnovna struktura i elementi PPS-a

Element	Namena
PPSS	Incijalni znak
PPS0	Znak formata
PPS1, PPS2,PPS3	Znakovi parametara
PCK	Znak za proveru

Tabela 3.13. Elementi podataka PPS-a i njihovo značenje

Prvi bajt poslat tokom ove procedure daje kartici do znanja da čitač započinje PPS. Uvek mora sadržati vrednost 'FF'. PPS0 je takođe obavezan a PPS1 i PPS2 su

opcionalni. PPS3 je rezervisan za buduću upotrebu. PCK je obavezan, ima isti smisao kao i ranije opisani TCK.

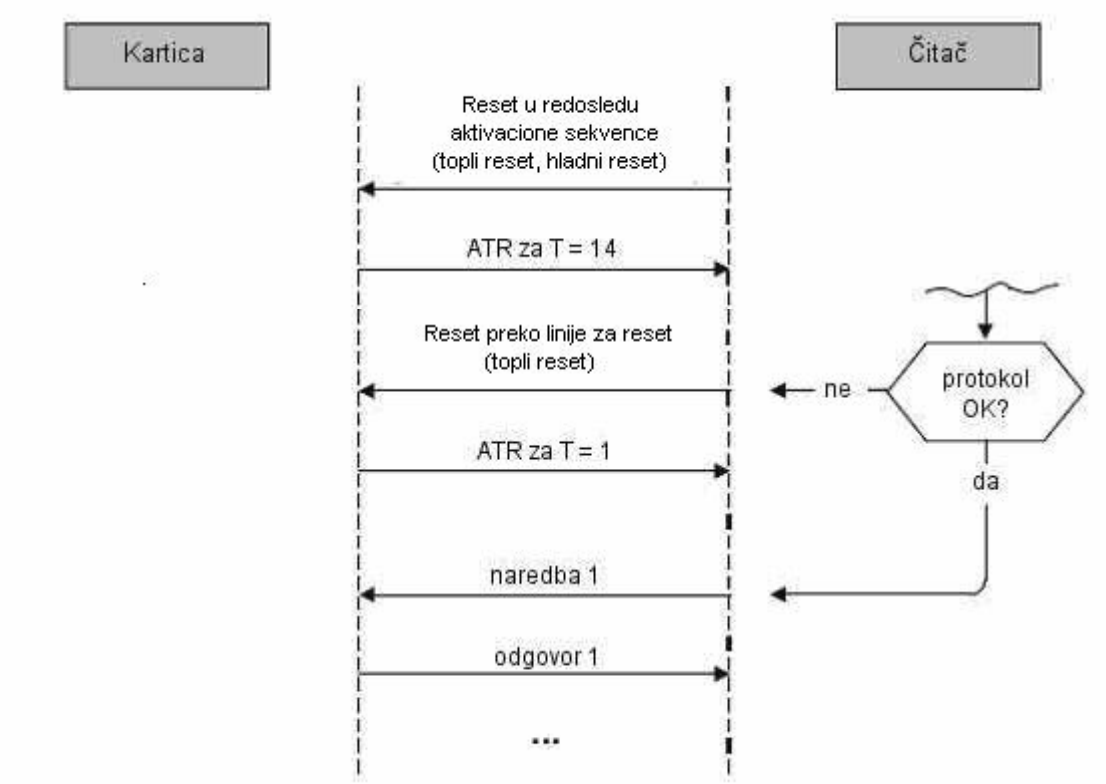
b8	b7	b6	b5	b4	b3	b2	b1	Značenje
...	...	...	...		X			Protokol koji će biti korišćen
...	...	...	1		...			PPS1 prisutan
...	...	1	...		...			PPS2 prisutan
...	1	...	...		...			PPS3 prisutan
0	...	...	...		...			Rezervisano za buduću upotrebu

Tabela 3.14. Kodiranje znaka PPS0

b8	b7	b6	b5	b4	b3	b2	b1	Značenje
X					...			FI
...					X			DI

Tabela 3.15. Kodiranje znaka PPS1

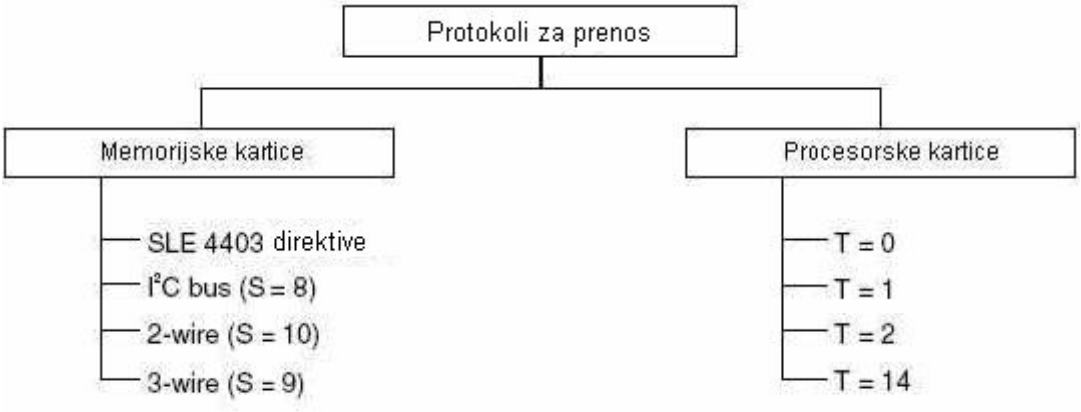
Ako kartica zna interpretirati PPS i uspešno modifikovati komunikacioni protokol ona vraća primljene PPS zakove čitaču. U suprotnom čeka reset. Najveći nedostatak ove procedure jeste to što je moguć veliki gubitak vremena pre uspostavljanja komunikacionog protokola. Ukoliko čitač nije u mogućnosti sprovesti PPS, a koristi neki posebni komunikacioni protokol nastaju problemi. Budući da čitači izvode ponovni reset ako ne prepoznaju ATR odlučeno je da kartice menjaju komunikacione parametre nakon svakog reseta (parametri se menjaju samo nakon reseta preko linije za reset (warm reset). Nakon uključenja napajanja na karticu (cold reset) ona se uvek ponaša identično).



Slika 3.12. Tipična PPS procedura

3.1.9.1. Protokoli za prenos podataka

Za vreme odgovora na reset u znaku TD1 niža četiri bita (16 mogućih protokola) diktiraju tip protokola za prenos podataka. Od izabranog protokola zavisi način na koji će se slati komande pametnoj kartici i kako će se njeni odgovori interpretirati. Od njih takođe zavisi postupak oporavka od eventualne greške pri komunikaciji.



Slika 3.13. Klasifikacija korišćenih protokola za komunikaciju sa karticama

Protokoli se obeležavaju sa 'T =' (engl.*transmission protocol*) i rednim brojem. Taj broj je upravo onaj kodiran u znaku TD1.

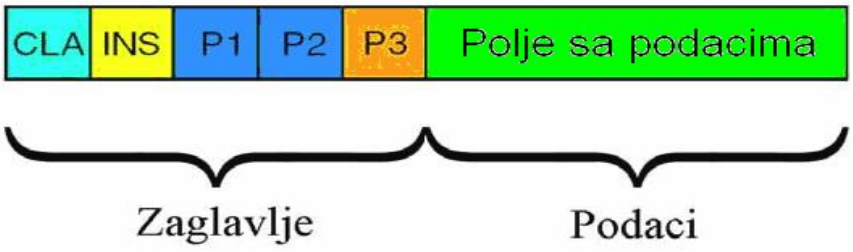
Protokol	Značenje
T=0	Asinhroni, half duplex, prenos podataka u bajtovima, specifikovan u ISO/IEC 7816-3
T=1	Asinhroni, half duplex, prenos podataka u blokovima, specifikovan u ISO/IEC 7816-3
T=2	Asinhroni, full duplex, prenos podataka u blokovima
T=3	Full duplex, još nije specifikovan
T=4	Asinhroni, half-duplex, prenos podataka u bajtovima, ekstenzija T=0 protokola, još nije specifikovan
T=5...T=13	Rezervisano za buduću upotrebu, još nije specifikovan
T=14	Za nacionalnu upotrebu (Nemačka), nije ISO standardizovan
T=15	Rezervisan za buduću upotrebu i još nespecifikovan

Tabela 3.16. Protokoli za prenos podataka premo ISO/IEC 7816-3

Trenutno su u najvećoj upotrebi protokoli T=0 i T=1.

3.1.10. ISO 7816 3.5. Komunikacioni protokol T=0

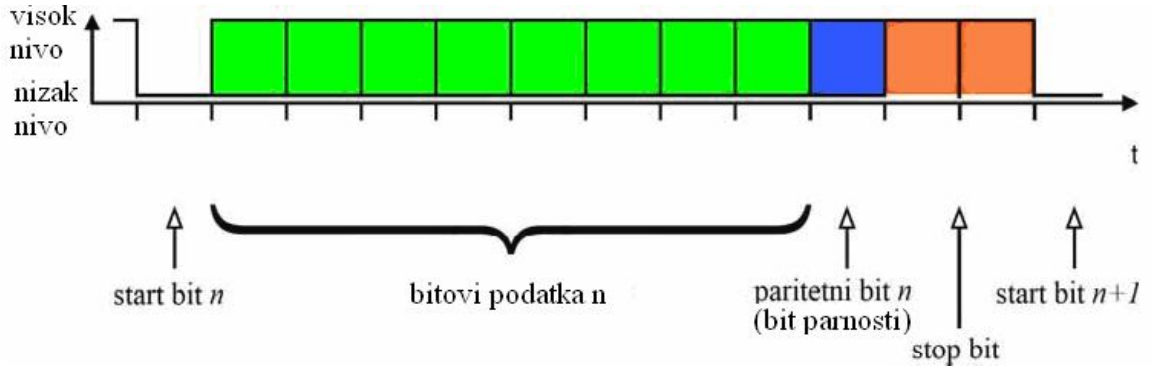
T=0 je bio jedini protokol inicijalno specifikovan u ISO 7816-3 standardu 1989. godine. S obzirom na snagu hardvera kojom se tada raspolagalo dizajniran je kako bi koristio što manje memorije i bio maksimalno jednostavan. Koristi se i u GSM karticama pa ga to čini najšire korišćenim od svih protokola za kartice. Njegove kompatibilne osobine specifikovane su u standardima GSM 11.11, TS 102.221 i EMV specifikacijama. Ovaj protokol je bajt orijentisan što znači da je najmanja prenosiva jedinica upravo jedan bajt. Naredba koja se prenosi sastoji se od zaglavlja (engl.*header*) i opcionalno polja sa podacima (engl.*data field*). Zaglavlje se se sastoji od bajta klase, bajta naredbe i tri bajta parametara.



Slika 3.14. Struktura naredbe u T=0 protokolu

Oznaka	Značenje
CLA	razred naredbe,
INS	identifikator naredbe (instrukcije),
P1, P2	ulazni parametri za naredbu,
P3	dužina, parametar koji određuje broj bajtova podataka (unutar naredbe) za prenos na (ili prenos od) kartice,
Polje sa podacima	Niz bajtova podataka poslatih u naredbi.

Tabela 3.14. Objašnjenje skraćenica T=0 protokola

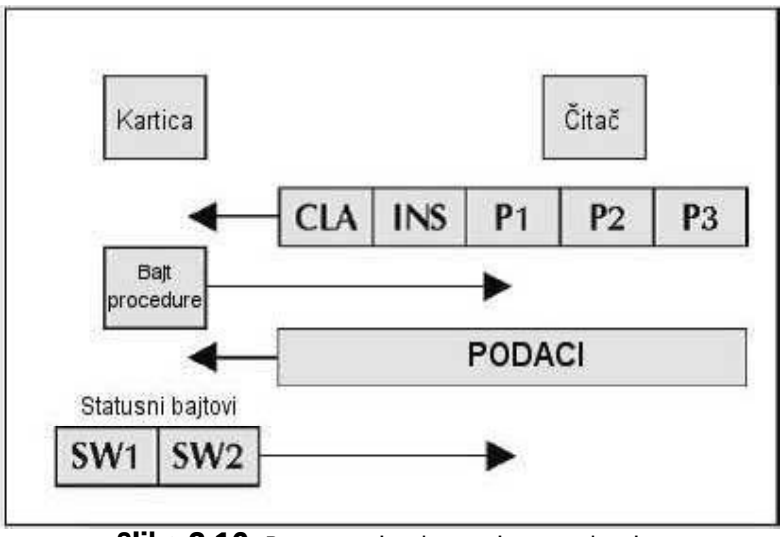


Slika 3.15. Prenos bitova putem U/I interfejsa koji koristi T=0 protokol

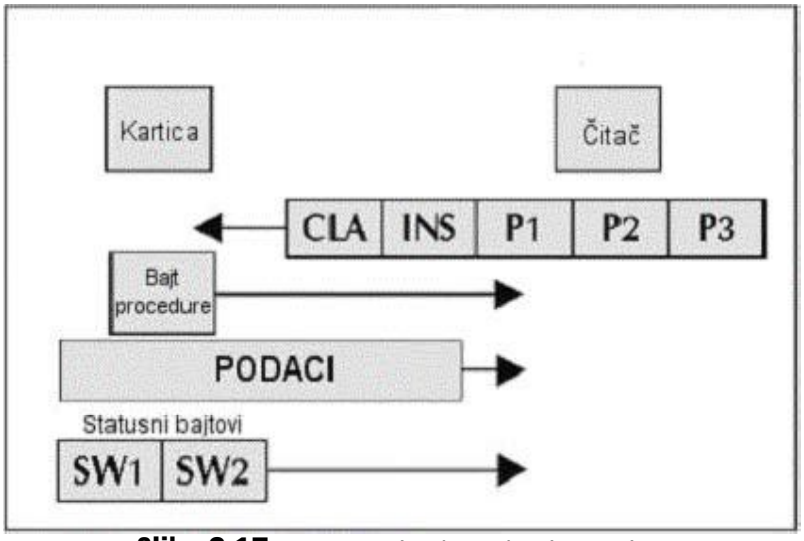
Ukoliko se dogodi greška pri prenosu podataka mora se odmah ponoviti netačan bajt, dok se kod protokola orijentisanih na prenos bloka podataka ponovo prenosi celi blok. Otkrivanje greške se bazira isključivo na proveri paritetnog bita (bit parnosti) koji je deo svakog poslatog bajta.

3.1.10.1. T=0 sekvenca naredbe i odgovora

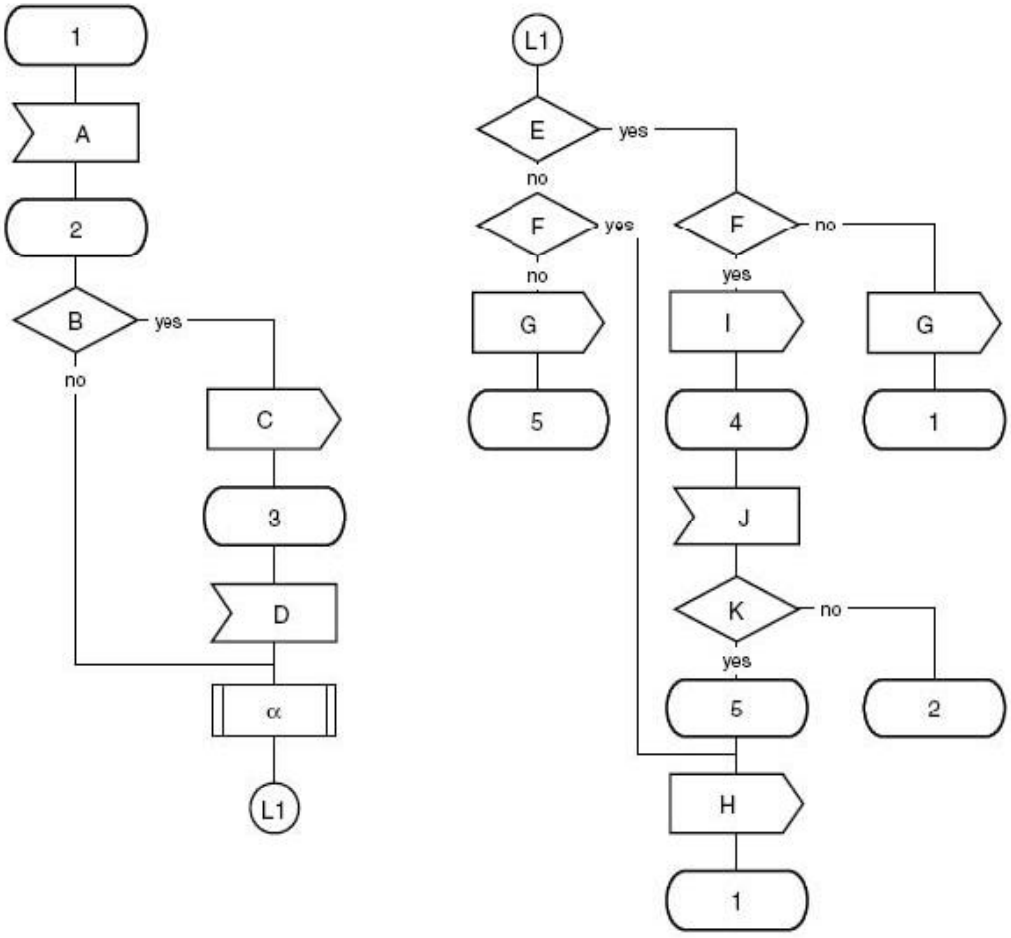
Za prikaz toka komunikacije između kartice i čitača pretpostavi se da čitač šalje kartici naredbu sa poljem podatka. Čitač će prvo poslati zaglavlje, i čekati da ono bude potvrđeno bajtom procedure (engl.*procedure byte*). Kad se to dogodi čitač šalje podatke dužine onoliko bajtova koliko je bilo naznačeno u bajtu P3. Sad je kartica primila celu naredbu, pa je može izvršiti i generisati odgovor. Odgovor se sastoji od dva statusna bajta SW1 i SW2 i eventualno od nekih dodatnih podataka što je naznačeno u bajtu SW2. Nakon što čitač primi statusne bajtove on "zna" da li želi da mu kartica pošalje još podataka. U slučaju da želi čitač šalje GET RESPONSE naredbu na koju kartica odgovara tim podacima. Time se zatvara ciklus jedne naredbe.



Slika 3.16. Prenos podataka sa čitača na karticu



Slika 3.17. Prenos podataka sa kartice na čitač



Slika 3.18. Dijagram toka T=0 protokola

α Izvršavanje naredbe
1 Početno stanje
2 Primljeno zaglavlje sa CLA, INS, P1, P2 & P3
3 Čekanje na deo sa podacima (P3 = broj bajtova)
4 Čekanje na naredbu (zaglavlje sa CLA, INS, P1, P2 & P3)
5 SW1 i SW2 poslati i GET RESPONSE primljen
A Primanje zaglavlja (5 bajtova)
B Ima li podataka (P3!=0)?
C Ima podataka; posalji PB (bajt procedure) čitaču
D Primi podatke (P3= broj bajtova)
E Da li je naredba sadržala podatke?
F Jesu li podaci za odgovor generisani?
G Pošalji SW1 i SW2
H Pošalji podatke za odgovor i SW1 + SW2
I Pošalji SW1 i SW2 (SW2 = količina podataka za odgovor)
J Primi zaglavlje naredbe
K Da li je primljena naredba GET RESPONSE?

T=0 protokol omogućava kartici primanje bajtova podataka jedan po jedan nakon primanja zaglavlja. To se ostvaruje tako da kartica kad PB čitaču pošalje invertovani bajt komande, posle kog čitač šalje jedan bajt podatka. Sledeći bajt podatka će biti poslat nakon sledećeg PB-a. To se nastavlja sve dok kartica ne primi sve podatke ili dok ne pošalje neinvertovani bajt komande kao PB. Između GSM 1.11 i ISO/IEC 7816-3 postoje dve nekompatibilnosti u specifikaciji T=0 protokola. Po GSM standardu , GET RESPONSE naredba se traži pomoću SW1= '9F' dok je ta vrednost kod ISO/IEC standarda '61'. Druga nekompatibilnost se odnosi na način na koji se podaci primaju nakon GET RESPONSE naredbe. Gore opisani postupak slanja pojedinačnih bajtova spada u GSM standard i uobičajno se koristi u većini primena širom sveta. Po ISO/IEC standardu ne postoji način da se zahtevaju bajtovi podataka jedan po jedan. Te nekompatibilnosti se lako rešavaju odgovarajućom programskom podrškom, bitna je svest o potrebi da one postoje! Jedna od bitnih stavki komunikacionog protokola je brzina prenosa. Slanje jednog bajta traje 12 ETU-a, što iznosi 1,25 ms kod frekvencije od 3.5712 Mhz pri deljenju iznosa sa 372.

Naredba	Korisnički podaci	Podaci protokola	Vreme prenosa
READ BINARY	N: 5 bajta O: 2+8 bajta		18.75 ms
UPDATE BINARY	N: 5 + 8 bajta O: 2 bajta		18.75 ms
ENCRYPT	N: 5 + 8 bajta O: 2+ 8 bajtova	N: 5 bajta R: 2 bajta	37.50 ms

Tabela 3.18. Vreme potrebno za prenos podataka kod nekih tipičnih naredbi T=0 protokola. (N: naredba, O: odgovor)

Na kraju svake komadne statusni SW1 i SW2 bajtovi govore o stanju kartice. Normalni završetak komande iznosi SW1-SW2=$90-$100. Definisane su i sledeće vrednosti za SW1:

- $6E Kartica ne podržava navedenu klasu naredbe,
- $6D Kod naredbe nije programiran ili nije ispravan,
- $6B Netačna referenca,
- $67 Netačna dužina,
- $6F Bez precizne dijagnostike.

SW1	SW2	Značenje
62	81	Vraćeni podaci možda nisu ispravni
62	82	Kraj datoteke dosegnut pre kraja čitanja
62	84	Odabrana datoteka nije ispravna
65	01	Memorijska greška. Problemi pri pisanju ili čitanju iz EEPROMA.
68	00	Funkcija traženja nije podržana od kartice
6A	00	Bajtovi P1 ili P2 su netačni
6A	80	Parametri u polju podatka su netačni
6A	82	Datoteka nije nađena
6A	83	Zapis nije nađen
6A	84	Nedovoljno memorije u zapisu ili datoteci
6A	87	Vrednost P3 nije u skladu sa vrednostima P1 i P2
6A	88	Referentni podaci nisu nađeni
6C	XX	Netačna dužina P3

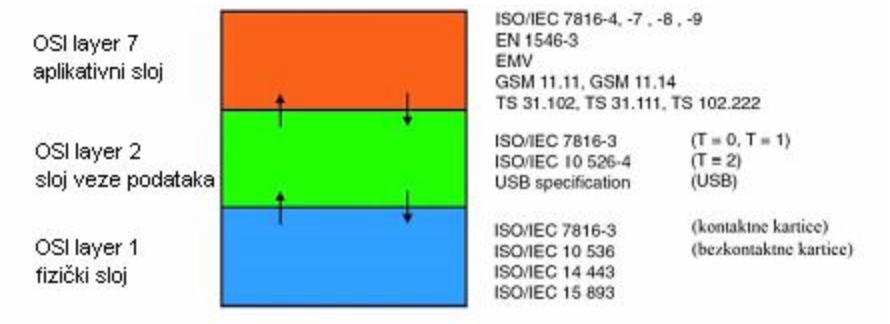
Tabela 3.19. Značenja bajtova sw1 i sw2 definisana u dodatku standarda

3.1.11. Komunikacioni protokol T=1

Ovaj protokol je asinhroni half-duplex blok orijentisan protokol. To znači da je najmanja prenosiva jedinica blok podataka. On postavlja blok podataka u svojevrsnu "kovertu" (envelopu) što omogućava:

- kontrolu toka,
- ulančavanje blokova,
- ispravljanje grešaka.

Po OSI modelu slojeva komunikacije on radi na 2., t.j. sloju podataka. Podela na slojeve omogućuje da podaci namenjeni višim slojevima (npr. aplikativnim) mogu biti preneseni potpuno transparentno na sloju podataka. Nije potrebno da se oni interpretiraju ili menjaju od strane slojeva koji nisu direktno uključeni. Sigurno slanje poruka zahteva pridržavanje podeli na slojeve. Samo se tako šifrovani korisnički podaci mogu propuštati kroz interfejs bez nekih komplikovanih procedura ili trikova. T=1 protokol je trenutno jedini međunarodni protokol za pametne kartice koji bez kompromisa može omogućiti siguran prenos podataka.On takođe uklanja ograničenje protokola T=0 prema kome je uvek čitač bio taj koji bi inicirao naredbu, a kartica bi odgovarala. Sada i kartica i čitač, u okvirima protokola, mogu inicirati naredbu.



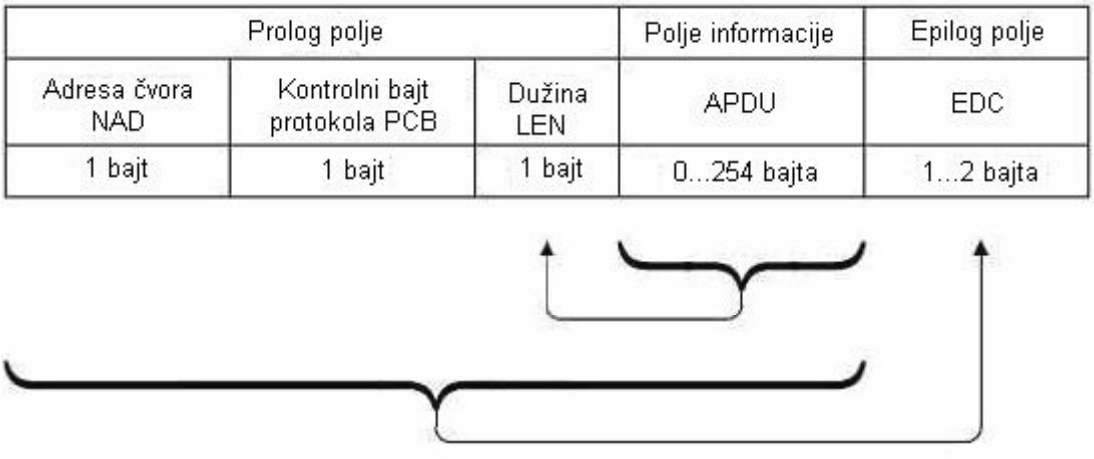
Slika 3.19. OSI model komunikacije kartice i čitača

Naredba	Korisnički podaci	Podaci protokola	Vreme prenosa
READ BINARY	N: 5 bajta O: 2+8 bajta	N: 4 bajta O: 4 bajta	28.75 ms
UPDATE BINARY	N: 5 + 8 bajta O: 2 bajta	N: 4 bajta O: 4 bajta	23.00 ms
ENCRYPT	N: 5 + 8 bajta O: 2+ 8 bajta	N: 4 bajta O: 4 bajta	38.75 ms

Tabela 3.20. Vremena prenosa podataka za neke tipične naredbe T=1 protokola, frekvenciju sistemskog sata od 3.5712MHz, vrednošću delitelja od 372, XOR koda za detekciju grešaka, 2 stop bita i 8 bajtova podatka po naredbi (N=naredba,O=odgovor)

3.1.11.1. Struktura bloka

Poslati blokovi se koriste u dve različite svrhe. Jedna je transparentno slanje aplikativnih podataka dok je druga slanje kontrolnih znakova protokola ili baratanje greškama u prenosu. Blok za slanje podataka se sastoji od prolog polja, polja informacije i epilog polja. Prolog i epilog polja su obavezna i moraju uvek biti postana. Polje informacije je opcionalno i sadrži podatke za aplikativni sloj, koji je ili APDU naredbe poslate kartici ili APDU odgovora kartice.



Slika 3.20. Struktura T=1 bloka za prenos podataka

Postoje tri tipa bloka u T=1: blok informacije, blok za potvrdu prijema i sistemski blok. Blokovi informacija (I blokovi) služe transparentnoj razmeni podataka aplikativnog nivoa. Blokovi za potvrdu prijema (R blokovi) koji ne sadrže polja sa podacima služe potvrđivanju ili negiranju prijema. Sistemski blokovi (S blokovi) služe za kontrolne informacije vezane uz sam protokol. Mogu sadržati polje informacije.

3.1.11.2. Prolog polje

Sadrži tri polja: adrese čvora (**NAD**), kontrolnog bajta protokola (**PCB**) i dužine (**LEN**). Dužine je tri bajta i sadrži osnovne kontrolne podatke i pokazivače za aktuelni blok.

Adresa čvora (NAD)

Sadrži adresu odredišta i izvorišta bloka. Svaka je kodirana koristeći tri bita. Ako se adresa ne koristi bitovi se postavljaju na 0.

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	Značenje
X	...	...	...	X	...	...	...	V_{pp} (ne koristi se)
...	X	X	X	...	...	...	...	DAD (adresa odredišta)
...	...	...	...	...	X	X	X	SAD (adresa izvora)

Tabela 3.21. Kodiranje polja NAD

Kontrolni bajt protokola (PCB)

Služi za kontrolu i nadgledanje protokola. Kodira tip bloka i neke dodatne informacije. Broj sekvence stanja kreće se između 0 i 1 i označava redni broj paketa. Bit M (engl.***M****ore*) služi pri ulančavanju prenosa i govori o tome da li sledi još blokova.

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	Značenje
0	...	...	...	...	...	...	...	Identifikator I bloka
...	N(S)	...	...	...	...	...	...	Broj sekvence slanja
...	...	X	...	...	...	...	...	Bit podatka M sekvence
...	...	...	X	X	X	X	X	Rezervisano

Tabela 3.22. Kodiranje polja PCB za I blok

Polje dužine (LEN)

Ovo polje sadrži dužinu polja informacije u heksadecimalnom obliku. Vrednost može biti od '00' do 'FE'. Kôd 'FF' je rezervisan za buduća proširenja i trenutno ne može biti korišćen.

3.1.11.3. Polje informacije

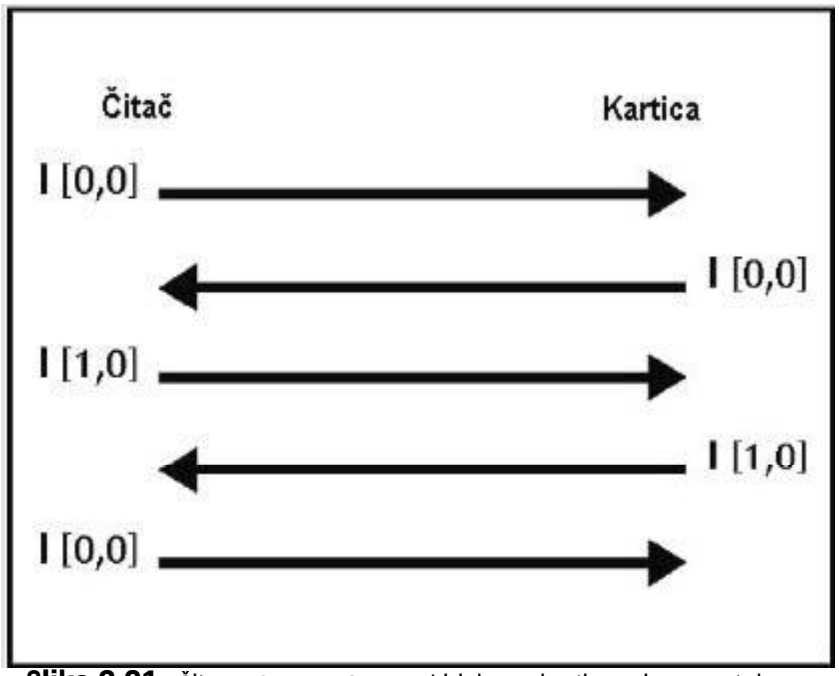
U I bloku ovo polje sluzi kao bafer za podatke aplikativnog sloja. Sadržaj ovog polja se šalje potpuno transparentno, t.j. protokol ga ne evaluira i interpretira. U **S** bloku ono sadrži podatke za protokol. Jedino se u ovom slučaju sadržaj ovog polja koristi u prenosnom sloju. Veličina ovog polja može biti od '00' do 'FE' bajtova. Podrazumevana vrednost je 32 bajta.

3.1.11.4. Epilog polje

Ovo polje se nalazi na kraju bloka. Sadrži kôd za detekciju grešaka izračunat na temelju svih prethodnih bajtova u bloku. Upotrebljava se ili **LRC** (engl.***L**ongitudinal **R**edundancy **C**heck*) ili **CRC** (engl.***C**ylic **R**edundancy **C**heck*). Izbor metode zavisi od odgovora na *reset*, a podrazumeva se da se koristi **LRC**. U tom slučaju polje **LRC** se dobija **XOR** operacijom nad svim bajtovima u bloku. Budući da je ta operacija brza i jednostavna može se izvoditi u "letu" i standardni je deo svih T=1 implementacija.

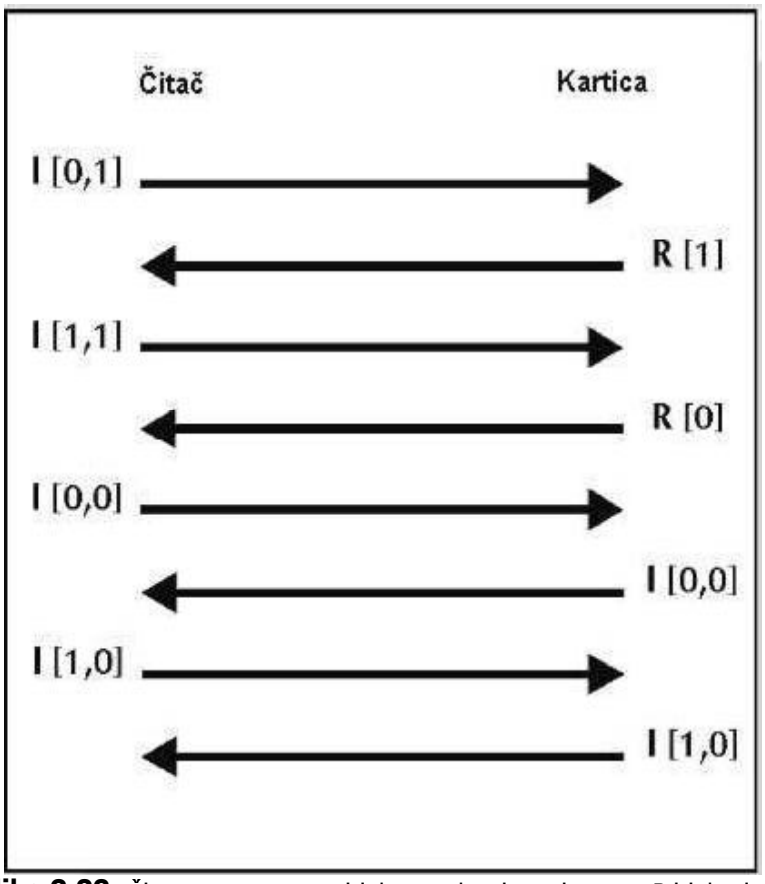
3.1.11.5. Primer komunikacije protokolom T=1

Slika prikazuje komunikaciju čitača (**IFD**) i kartice (**ICC**). Čitač šalje ***I*** blokove. Brojke u zagradi su iznosi bitova ***N***,*M*. Blokovi preneseni jedan po jedan mogu se potvrđivati ***I*** blokom dok se ulančani blokovi potvrđuju ***R*** blokom.



Slika 3.21. Čitač šalje pojedinačne I blokove, kartica odgovara takođe I blokovima. Vidi se stanje bitova [N,M]

Na sledećoj slici se vidi kako kartica na ulančane blokove odgovara ***R*** blokom. Vrednost u zagradi je broj sekvence slanja ***N***.



Slika 3.22. Čitač šalje ulančane blokove a kartica odgovara R blokovima. sve do poslednjeg bloka u nizu

3.1.11.6. Vremena čekanja

Definisano je nekoliko vremena čekanja kako bi se pošiljaocima (engl.***S**ender*) i primaocima (engl.***R**eceiver*) specifikovali intervali za razne akcije tokom prenosa podataka. Ona takođe predstavljaju način izbegavanja potpunog zastoja. Postoje podrazumevane vrednosti, ali se one mogu i modifikovati putem odgovora na *reset*.

Vreme čekanja na znak (CWT)

Definisano je kao maksimalni interval između dva znaka unutar bloka. Primalac aktivira brojač na svaki uzlazni brid koristeći **CWT** kao inicijalnu vrednost. Ako brojač dođe do nule pre nego što se registruje novi uzlazni brid primalac pretpostavlja da je blok u celosti primljen. Ovaj kriterijum se može uopšte koristiti za detekciju kraja bloka, ali to uveliko smanjuje brzinu prenosa jer se vreme prenosa svakog bloka uvećava za **CWT**. Prema tome je bolje kraj bloka odrediti brojeći primljene znake. **CWT** se računa na sledeći način (**CWI** se specifikuje u odgovoru na *reset*):

$$CWT = (2^{CWI} + 11) radni_ETU$$

Podrazumevana vrednost za CWI je 13, što daje sledeću vrednost:

$$CWT = (2^{13} + 11) radni_ETU = 8203\ radni_etu$$

Uz frekvenciju od 3.5712 Mhz i vrednost delitelja od 372 dobija se interval od 0.85 sekundi.

Vreme čekanja na blok(BWT)

Svrha vremena čekanja na blok je da se omogući prekid komunikacije ako kartica ne odgovara. Ono je definisano kao vremenski interval između uzlaznog brida poslednjeg bajta u poslatom bloku i uzlaznog brida prvog bajta kojeg kartica vraća. Ukoliko taj interval istekne čitač može pretpostaviti da sa karticom nešto nije u redu i preduzeti odgovarajuću akciju, npr. resetovati karticu. **BWT** se računa na sledeći način (**BWI** se specifikuje u odgovoru na reset):

$$BWT = 2^{BWI} \times 960 \times \frac{372}{f} s + radni_ETU$$

Uz pretpostavljenu vrednost BWI=4, 3.5712Mhz i vrednost delitelja 372 dobija se:

$$BWT = 2^{4} \times \frac{372}{3571200\ \mathrm{Hz}} s + 11\ radni_ETU = 2^{4} \times 0.1s + 11\ radni_ETU \approx 1.6s$$

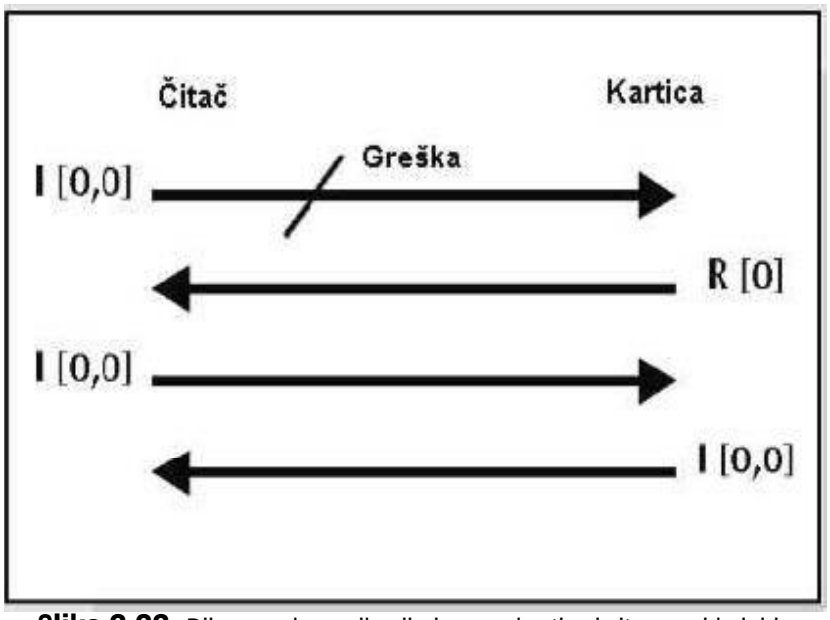
Ova vrednost predstavlja kompromis između normalne obrade naredbi i brze detekcije greške na kartici.

Čuvano vreme bloka (BGT)

Definisano je kao najmanji interval između uzlaznog brida poslednjeg bajta i uzlaznog brida prvog bajta u suprotnom smeru. Svrha mu je da osigura pošiljaocu vremenski interval u kojem se prebacuje iz slanja u primanje. Ima fiksno vreme od 22 ETU. Pri frekvenciji od 3.5712Mhz sa deliteljem iznosa 372 postiže se interval od otprilike 2.3 ms.

3.1.11.7. Detekcija i rukovanje greškama

T=1 protokol ima razrađenu detekciju i rukovanje graškama. Ako se detektuje neispravni blok protokol pokušava uspostavljanje oporavka od greške koristeći precizno definisane procedure. Sa gledišta čitača postoje tri sinhronizaciona stepena. Na prvom pošiljalac bloka sa greškom dobija kao potvrdu ***R*** blok sa naznakom greške i zahtevom za ponovno slanje. Tada pošiljalac mora ponovo poslati poslednji blok.



Slika 3.23. Dijagram komunikacije između kartice i čitača pri kojoj je došlo do greške

Ako se to pokaže nemogućim prelazi se na sledeći stepen. To znači da kartica dobija zahtev za ponovnom sinhronizacijom. Nakon odgovora na taj zahtev kartica i čitač postavljaju svoje brojače slanja i primanja na ***0***, što odgovara stanju protokola neposredno nakon odgovora na *reset*. Nakon toga sledi pokušaj ponovnog uspostavljanja komunikacije. Ova dva stepena tiču se samo sloja protokola. Nemaju uticaj na aplikaciju. Treći stepen ima uticaja na sve komunikacione slojeve u pametnoj kartici. Ako prva dva stepena ne urode plodom, terminal kartici šalje signal za *reset* preko odgovarajućeg kontakta. To znači gubitak svih podataka i stanja trenutne sesije. Nakon *reseta* komunikacija se ponovo uspostavlja od dna prema vrhu. Ako čak ni to ne uspe nakon tri pokušaja, čitač deaktivira karticu i javlja korisniku poruku o greški.

Stepen	Mehanizam
1	Ponoviti blok sa greškom
2	Resinhronizovati i ponoviti blok sa greškom
3	Resetovati karticu i ponovo uspostaviti komunikaciju

Tabela 3.23. T=1 Stepeni rukovanja greškama

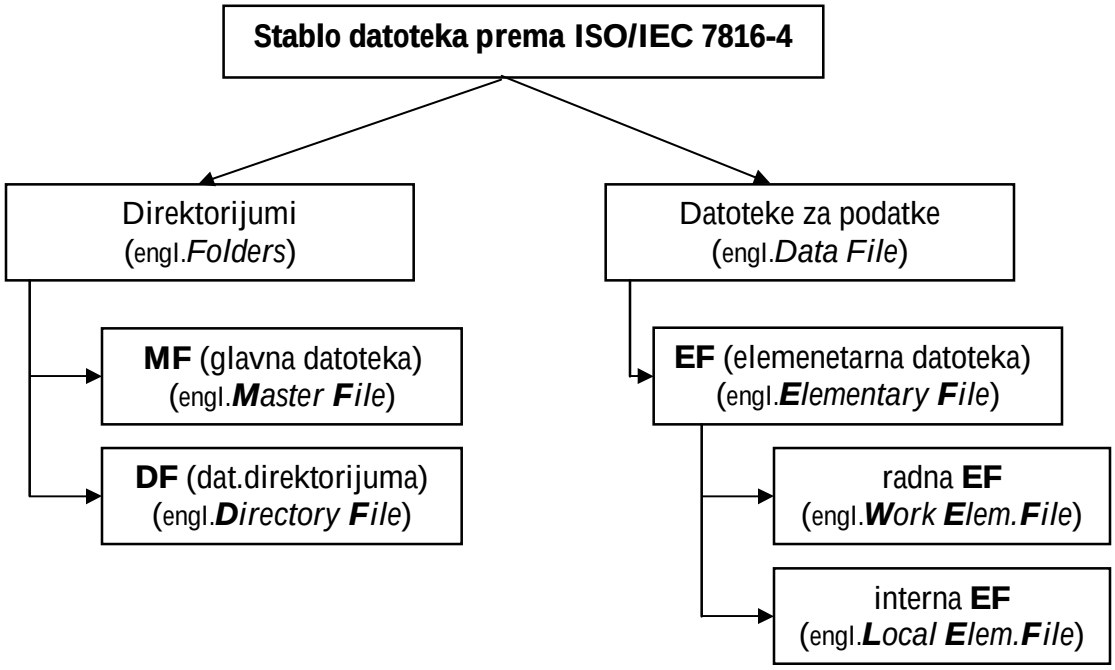
3.1.12. ISO 7816-4

Ovaj deo smart card standarda određuje:
- sadržaj poruka, naredbi i odgovora u komunikaciji kartice i čitača,
- strukturu i sadržaj *historical* znaka u odgovoru na *reset*,
- strukturu datoteka i podataka kako ih vidi čitač kada radi sa komandama za razmenu podatka,
- metode za pristup datotekama i podacima na kartici,
- metode za sigurno slanje poruka,
- metode za pristup algoritmima koje kartica sprovodi. Ne opisuje same algoritme.

On ne pokriva internu implementaciju unutar kartice ili spoljnog sveta. Takođe dozvoljava standardizaciju daljih naredbi kao i sigurnosnih arhitektura.

3.1.12.1. Struktura datoteka

Određene su tri kategorije datoteka:
- Master datoteka (**MF**)
- Datoteka direktorijuma (**DF**)
- Elementarna datoteka (**EF**)



Slika 3.24. Stablo datoteka prema ISO/IEC 7816-4 standardu

Glavna datoteka (MF)

Koren direktorijum (engl.*root*) se naziva i '*master datoteka*'. Prema standardu je njeno postojanje obavezno i implicitno je selektovana nakon *reseta* kartice. Ona sadrži sve ostale direktorijume i datoteke. To je zapravo posebni tip datoteke direktorijuma.

Datoteka direktorijuma (DF)

Datoteke direktorijuma mogu postojati hijerarhijski ispod master datoteka prema potrebi. One mogu sadržati druge datoteke direktorijuma. Broj nivoa takvog ugnježdenja nije ograničen iako zbog ograničene memorije na kartici retko postoji više od dva podnivoa.

Elementarna datoteka (EF)

Korisnički podaci potrebni aplikacijama nalaze se u '*elementarnim datotekama*' . One se mogu nalaziti odmah ispod master datoteke ili ispod datoteke direktorijuma. Uvek sadrže internu strukturu podataka kako bi se oni sačuvali u minimalnoj količini memorije i u logički optimizovanoj strukturi. Raspoređuju se u radne i interne.

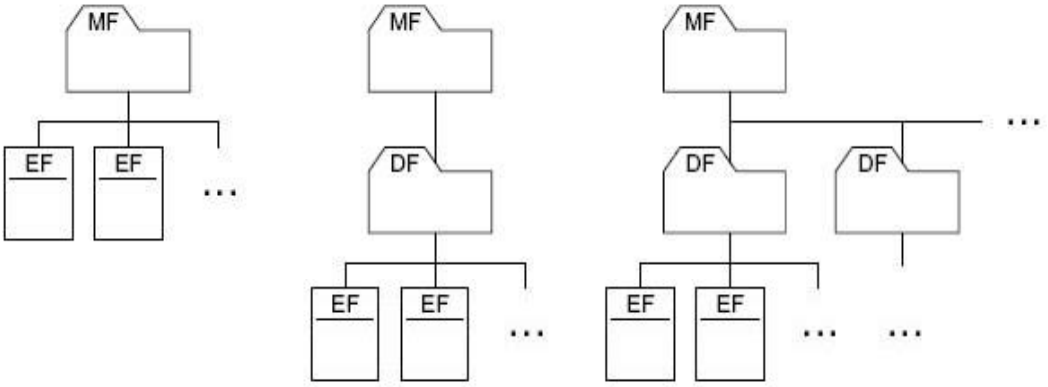
Radni EF

Svi podaci koji su namenjeni slanju ili primljeni od čitača moraju biti sačuvani u ovakvim datotekama. Te podatke ne koristi operativni sistem.

Interni EF

Ovde spadaju sistemske datoteke koje čuvaju podatke samog operativnog sistema, podatke za izvođenje aplikacija, tajne ključeve i programske kodove. Pristup njima posebno štiti operativni sistem, oni ne mogu biti označeni i njima se barata transparentno u odnosu na korisnika.

Radne datoteke mogu se grupisati zavisno od broju aplikacija na pojedinoj kartici. Ukoliko je na kartici samo jedna, datoteke se smeštaju direktno ispod *master* datoteke. Ukoliko aplikacija ima više, za svaku se može kreirati jedna datoteka direktorijuma i tako na odgovarajući način rasporediti aplikativne datoteke.



Slika 3.25. Moguće organizacije stabla sa datotekama

U novijim sistemima kartica koristi se logičko adresiranje datoteka umesto pristupa preko direktnih fizičkih adresa. Takav način adresiranja je proširiv i u skladu sa kriterijumima modernog programskog inženjerstva. Standard ISO/IEC 7816-4 propisuje upotrebu dvobajtnog identifikatora datoteke (**FID**) prilikom njene selekcije.

FID	Ime i svrha	Standard
'2F00'	Rezervisan za EF_{dir} datoteku koja se koristi za čuvanje identifikatora aplikacija	ISO/IEC 7816-4
'2F01'	Rezervisan za EF_{ATR} koja sadrži dodatke odgovoru na *reset*	ISO/IEC 7816-4
'3F00'	MF root direktorijum za sve datoteke na kartici	ISO/IEC 7816-4 GSM 11.11 TS 102.221 EMV
'3FFF'	Rezervisan za odabir datoteke korišćenjem naziva putanje	ISO/IEC 7816-4
'FFFF'	Rezervisano za buduću upotrebu	ISO/IEC 7816-4

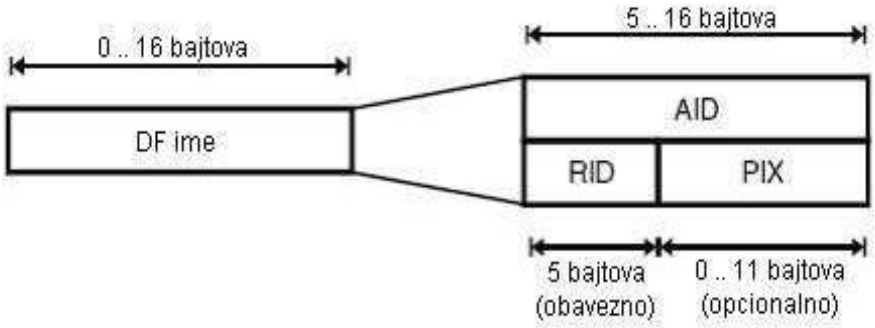
Tabela 3.24. Rezervisane vrednosti identifikatora u najvažnijim standardima kartica

Identifikatori u stablu moraju biti izabrani tako da se datoteke mogu jednoznačno odabrati. Zabranjeno je da dve datoteke unutar istog direktorijuma imaju isti identifikator. Pravila pri dodeljivanju identifikatora su:

1. Svi **DF** i **EF** unutar istog direktorijuma moraju imati različite identifikatore,
2. Ugnježđeni direktorijumi ne smeju imati iste identifikatore,
3. **EF** unutar nekog direktorijuma ne sme imati isti identifikator kao sledeći u hijerarhiji viši ili niži direktorijum.

DF se obično koristi kao skup datoteka koje koristi jedna aplikacija. U budućnosti bi adresni prostor **FID**-a od 2 bajta mogao postati premali. Zbog toga svaki **DF** osim identifikatora ima još i '***DF ime***'. Po ISO/IEC 7816-4 standardu ono može imati dužinu između 1 i 16 bajtova. Ta imena se mogu dodeljivati slobodno, ali obično se koriste u

sprezi sa identifikatorima aplikacija (**AID**) (ISO/IEC 7816-5). Identifikator aplikacije ima dužinu od 5 do 16 bajta i sastoji se od dva elementa.



Slika 3.26. DF ime u sprezi sa AID-om

3.1.12.2. Identifikator aplikacije (AID)

Sastoji se od dva elementa podatka. Prvi element podatka je registrovani identifikator koji ima dužinu od 5 bajta. Dodeljen je od strane nacionalnog ili internacionalnog registracionog centra (**RA**) i uključuje kôd države, kategoriju aplikacije i broj koji se odnosi na proizvođača aplikacije. To znači da se **RID** dodeljuje samo jednom pa se identifikacija aplikacije može izvesti bilo gde na svetu. Nažalost liste dodeljenih **RID**-ova su tajne. Ukoliko je potrebno proizvođač aplikacije može u njen identifikator dodati i vlasničko produženje identifikatora (**PIX**). To produženje može biti dugačko dodatnih 11 bajtova i može služiti imenovanju aplikacije.

RID			Značenje	
D1	D2-D4	D5-D10		
X	-	-	Kategorija registracije:	'A' – internacionalna registracija 'D' – nacionalna registracija
-	X	-	Kod države	
-	-	X	Broj proizvođača aplikacije	

Tabela 3.25. Kodiranje RID-a

3.1.12.3 Strukuture elementarnih datoteka

Za razliku od datoteka na konvencionalnim računarima elementarne datoteke u pametnim karticama imaju internu strukturu. Ona može biti individualno odabrana za svaku datoteku s obzirom na svrhu u koju će se koristiti. To je velika prednost za korisnike jer strukture omogućuju vrlo brz i efektivan pristup elementima podatka. Baratanje strukturama zahteva značajne količine programskog kôda na samoj kartici.

Transparentna struktura

Naziva se još binarnom ili amorfnom strukturom. Drugim rečima transparentna struktura zapravo nije struktura u pravom smislu reči. Podacima u datoteci se pristupa u bajtovima ili blokovima koristeći vrednost pomaka. Naredbe za pristup su READ BINARY, WRITE BINARY i UPDATE BINARY.

Linearna fiksna struktura

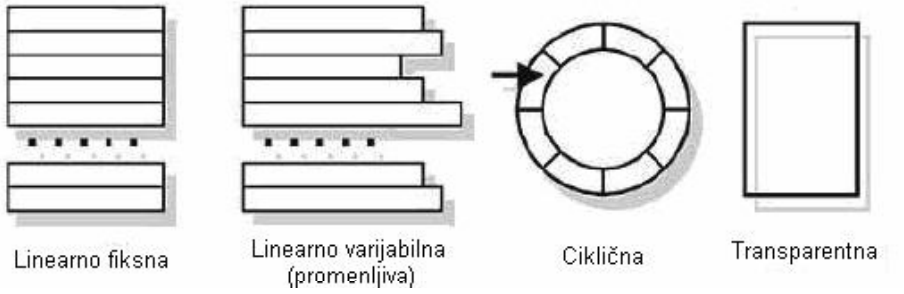
Bazira se na ulančavanju zapisa fiksne dužine. Zapis se sastoji od niza individualnih bajtova. Zapisi su najmanja jedinica kojoj se može pristupati, i može im se pristupati slobodno. U tu svrhu koriste se naredbe: READ RECORD, WRITE RECORD I UPDATE RECORD.

Linearna varijabilna (promenljiva) struktura

S obzirom da se upotrebom fiksne strukture zapisa rasipa memorijski prostor ako postoje zapisi različitih dužina uvodi se struktura sa varijabilnom strukturom zapisa. Ova struktura je slična prethodnoj s tim što svaki zapis mora sardržati i informaciju o svojoj dužini. Naredbe za pristup su iste kao i kod prethodne strukture.

Ciklična struktura zapisa

Zasniva se na lineranoj fiksnoj strukturi sa tom razlikom da sadrži pokazivač na poslednje zapisani zapis. Najnoviji zapis je uvek označen rednim brojem 1. Kad se prostor predviđen za zapise popuni pokazivač se premešta na najstariji zapis. Tako se postiže da zapisivanje novog zapisa uvek briše najstariji. Struktura se tipično koristi za log datoteke unutar kartice.



Slika 3.27. Strukture elementarnih datoteka

U kasnijim standardima definisane su još i:

Struktura izvršne datoteke

Ovo zapravo i nije zasebna struktura, bazira se na transparentnoj strukturi i namenjena je čuvanju sigurnosnog kôda. Predstavlja mogući sigurnosni problem jer svako sa pravima pisanja u takvu datoteku može svoj program učitati na karticu. Određena je u standardu *EN 7826-3*.

Struktura datoteke baze podataka

Standard ISO/IEC 7816-7 definiše verziju **SQL**-a pod nazivom **SCQL** (engl.***S**mart **C**ard **Q**uery **L**anguage*). Kako bi se sačuvanim podatcima moglo pristupiti korišćenjem *SCQL* naredbi potrebno je imati odgovarajuću strukturu zapisa. Ona međutim nije standardizovana već je njen dizajn ostavljen 'na dušu' proizvođačima sistema kartica.

3.1.12.4. Struktura poruka u komunikaciji kartice i čitača (APDU)

Ovaj deo standarda se nastavlja na definiciju protokola T=1 iz standarda ISO 7816-3 i definiše strukturu **APDU** (engl.***A**plication **P**rotocol **D**ata **U**nit*). APDU se koristi za razmenu svih podataka između kartice i čitača i on je standardizovan za aplikativni sloj OSI modela. Njegov sadržaj i struktura moraju biti nezavisni od komunikacionog protokola. Postoje dva tipa APDU-a, APDU naredbe i APDU odgovora.

3.1.12.5. APDU naredbe

Sastoji se od zaglavlja i tela. Veličina tela može varirati, a ukoliko je polje podatka prazno ne mora ga ni biti.



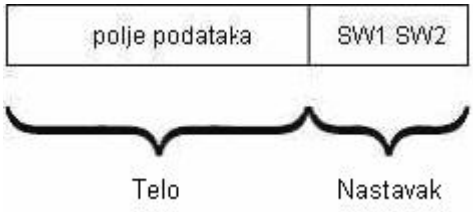
Slika 3.28. Struktura APDU naredbe

Skraćenica	Naziv	Dužina	Opis
CLA **INS** **P1** **P2**	Klasa Naredba Parametar 1 Parametar 2	1 1 1 1	Klasa naredbe Kod naredbe Naredbeni parametri
Lc field	Dužina podataka naredbe	varijabilna ≤ 3	Broj bajtova u delu sa podacima
Data field	Podaci	varijabilna =Lc	Niz bajtova podataka poslatih u naredbi
Le field	Dužina podataka odgovora	varijabilna ≤ 3	Maksimalan broj bajtova podataka očekivanih u odgovoru

Tabela 3.26. Polja u APDU naredbe

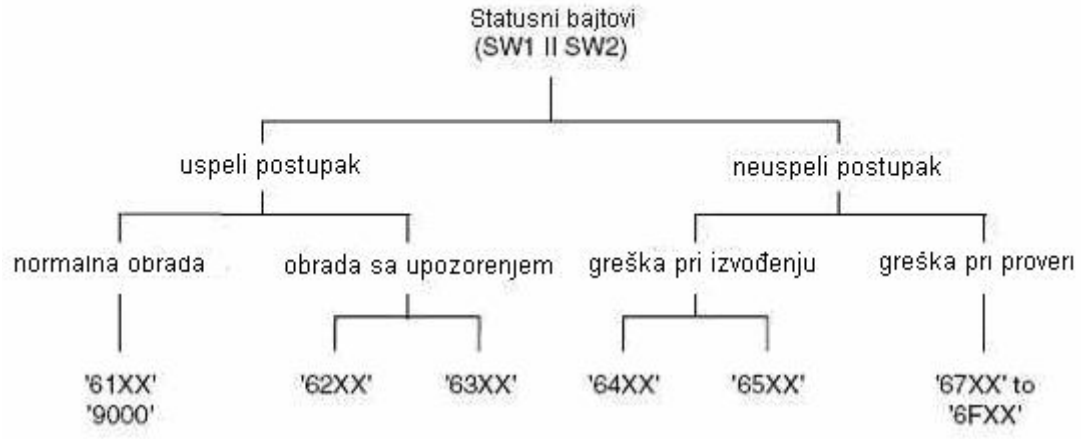
3.1.12.6. APDU odgovora

Sastoji se od opcionalnog tela i obaveznog nastavka. Telo se sastoji od dela sa podacima, čija je dužina određena **Le** poljem iz **APDU**-a naredbe. Nastavak sadrži dva statusna bajta **SW1** i **SW2**.



Slika 3.29. Struktura APDU-a odgovora

Ukoliko se dogodi greška prilikom izvođenja naredbe **APDU** ne mora sadržati deo sa podacima (bez obzira na **Le**). U tom slučaju se informacija o greški vidi iz statusnih znaka.



Slika 3.30. Informacija o greški iz statusnih znaka

3.1.12.7. Osnovne naredbe

ERASE BINARY

Postavlja deo ili celu elementarnu datoteku u logički obrisano stanje. Parametri daju pomak do prvog podatka koji treba obrisati. Naredba pretpostavlja da je datoteka prethodno selektovana.

VERIFY

Inicira upoređivanje podatka poslatog kartici sa podatkom sačuvanim na kartici.

EXTERNAL AUTENTIFICATION

Namenjena je autentifikaciji čitača odnosno osobe koja pristupa kartici. Kartica šalje slučajni broj čitaču koji ga šifruje svojim tajnim ključem. Dobijena šifra se vraća kartici (koristeći ovu naredbu) koja može šifrovanjem tog istog broja vlastitim ključem proveriti da li su ključevi isti. Drugi pristup je korišćenjem sistema javnog ključa kojim se može postići isti rezultat bez potrebe deljenja tajnog ključa. Parametri su izbor algoritma i tajnog podatka u kartici.

INTERNAL AUTENTIFICATION

Ova naredba upotpunjuje bilateralnu autentifikaciju tako da čitač proverava autentičnost kartice. Postupak je analogan onome kod pređašnje naredbe.

SELECT FILE

Svrha ove naredbe je da odabere datoteku nad kojom će se vršiti neka od drugih naredbi. Parametri nose informaciju o načinu adresiranja, a telo naredbe nosi identifikator datoteke.

READ BINARY

Služi za čitanje podataka direktno iz označene elementarne datoteke. Parametri služe da bi se odredio pomak od početka datoteke. Le polje definiše broj bajtova koje treba pročitati.

READ RECORD

Čita jedan ili više zapisa iz označene elementarne datoteke. Parametri uspostavljaju protokol po kome se pristupa zapisu. Le polje naredbe polju sa podacima sadrži broj bajtova koje treba pročitati.

GET RESPONSE

Dozvoljava dohvatanje podataka nastalih kao rezultat prethodno izvršene naredbe. Koristi se u T=0 protokolu koji ne dozvoljava slanje podataka u oba smera unutar jedne naredbe. Naredbu inicira čitač.

WRITE BINARY

Komplementarna naredbi READ BINARY. Piše podatke u elementarnu datoteku na nestrukturirani način.

WRITE RECORD

Komplementarna naredbi READ RECORD. Parametri označavaju željeni zapis, a polje Le broj bajtova bloka podataka. Naredba takođe dozvoljava tzv. append i update način zapisivanja. Append način dodaje zapise na kraj datoteke (ukoliko se radi o cikličnoj strukturi zapisuje zapis br 1). Update način menja zapis određen parametrima naredbe.

3.1.13. Drugi komunikacioni protokoli

3.1.13.1. Protokol 'T=14'

ISO/IEC 7816-3 standard u odgovoru na *reset* predviđa oznaku za nacionalne komunikacione protokole. Takav protokol je T=14. Stvoren je u 1987 Nemačkoj kad se ukazala potreba za protokolom koji bi služio komunikaciji između mobilnih telefona i pametnih kartica u njima. T=0 nije zadovoljavao, a u to vreme još nije bilo adekvatnog komunikacionog protokola orijentisanog na prenošenje blokova podataka. Nije imao značaja izvan Nemačke, ali je ostavio dubok uticaj na razvoj svog naslednika protokola 'T=1'. Upravo ga je on i zamenio pa se T=14 više ne koristi.

3.1.13.2. USB protokol prenosa

Budući dodatak standardu ISO/IEC 7816-3 sadržavaće specifikaciju novog prenosnog protokola kartice. To je ***U**niversal **S**erial **B**us* (**USB**) protokol, koji je

prevagnuo nad sličnima (*firewire*) kad je reč o primeni na pametnim karticama. On zahteva posebnu hardversku komponentu na procesoru kartice, ali ona je već dostupna u mnogim karticama makar kao opcija. Njegova prednost je ta, što je već uspostavljen industrijski standard u svetu personalnih računara. On takođe donosi veće brzine prenosa od T=0 i T=1 protokola. Čini se da će za komunikaciju sa karticama biti korišćena verzija 1.1, sa mogućnošću sporog (1.5 Mbit/s) i brzog (12 Mbit/s) prenosa. Navedene brzine se smanjuju kad se uzmu u obzir podaci protokola koji se takođe prenose.

3.2. EMV Standard

Zbog naglog porasta kupovine preko Interneta i plaćanjem kreditnim karticama (koje u svakom slučaju čine 99% svih plaćanja u svetu), finansijske institucije i njihove banke članice odlučile su napraviti pametnu kreditnu karticu kako bi postigli veću sigurnost kod on-line plaćanja. Naime, gubici nastali zbog pronevere kartica i drugih faktora prilikom kupovine preko Interneta ogromni su. Visa je, naprimer, dosad gubila oko 250 miliona dolara godišnje na razne zloupotrebe nastale kupovinom preko Interneta. Zbog sigurnosne infrastrukture implementirane na smart kartici, navedeni iznos smanjuje se do 40%, a uopšte pronevere u bankarskoj industriji (ne samo na Internetu) smanjuju se i do 70%. U pitanju su veliki vrednosni iznosi pa smart kartica daje veliki doprinos sigurnosti i veliki je pomak na tom području.

Da bi se omogućila multiaplikacijska kompatibilnost, potrebno je doneti standard. EMV standard to omogućuje s tim da je funkcija plaćanja postala središnja aplikacija. Standard EMV je nastao 1993. godine kao rezultat zajedničkog rada vodećih svetskih platnih institucija: ***E****uropay*-a, ***M****astercard*-a i ***V****isa*-e, po čemu je i standard dobio ime (početna slova udruženih institucija). Glavni zadatak im je bio definisanje skupa standarda za platne aplikacije temeljene na smart karticama. Ti standardi omogućuju siguran i nesmetan rad pametne kartice sa uređajima za čitanje, tzv. CAD (engl.***C****ard* ***A****cceptance* ***D****evice*), tj. daju skup pravila o njihovoj međusobnoj komunikaciji. EMV specifikacije napisane su sa sledećim ciljevima:

- Kartica i CAD uređaj (npr. bankomat, platni EFTPOS terminal, PC čitač), moraju da zajedničkom komunikacijom odrede koje su im zajedničke aplikacije i funkcije,
- CAD uređaj može izvoditi uobičajne aplikacije i osigurati minimalne sigurnosne standarde za kreditne aplikacije.
- Mikroprocesorske platne kartice moraju biti interoperabilne i prihvaćene u celom svetu.

EMV specifikacije temeljene su na ISO (engl.***I**nternational **O**rganization of **S**tandardisation*) setu standarda za čip-kartice i uređaje za njihovo isčitavanje. EMV sprecifikacije su ustvari implementacijski orijentisani podskup ISO 7813-3 standarda.

EMV se sastoji od četiri dela, tj. knjige:

1. *knjiga (*engl. ***ICC to Terminal Interface**)*
   - specifikacije zajedničkog interfejsa CAD uređaja i kartice koja ne zavisi od aplikacije.
   - opis minimalne funkcionalnosti potrebne za ispravan rad i interoperabilnost kartica i terminala nezavisno od aplikacije koja se izvršava.
2. *knjiga (*engl. ***Security and Key Management**)*
   - specifikacije sigurnosti i upravljanja šifarskim ključevima
   - definicije neophodnih sigurnosnih minimuma koje kartice i uređaji za čitanje moraju zadovoljiti s obzirom na on-line komunikaciju između kartice i izdavača kao i upravljanje kriptografskim ključevima u celom sistemu plaćanja.
3. *knjiga (*engl. ***Application Specification**)*
   - specifikacije aplikacija,
   - definicije procedura neophodnih da bi terminali i pametne kartice vršili međunarodne platne transakcije.
4. *knjiga (*engl. ***Other Interfaces**)*
   - specifikacija interface-a za platne sisteme,
   - davanje obveznih, preporučljivih i opcionalnih zahteva za terminale s obzirom na vlasnika kartice, trgovca i izdavača kartice radi prihvatanja pametnih kartica usklađenih sa prve tri knjige.

3.2.1. Prednosti EMV standarda

Postoje tri primarne prednosti koje se dobijaju upotrebom EMV platnih terminala i čitača pametnih kartica:

3.2.1.1. Povećana sigurnost i smanjena zloupotreba

Smanjena zloupotreba najviše se očekuje u on-line trgovini, gde je ona i najveća. Strah zbog zloupotrebe jedan je od glavnih razloga sporijeg razvoja trgovanja putem javne mreže. Poznato je da se oko dve trećine potencijalnih transakcija preko interneta prekidaju u trenutku kada se od korisnika zatraže njegovi finansijski podaci. Ogroman je problem i velik broj kasnije poreknutih transakcija.

Povećanjem sigurnosti korisnika platnih kartica na Internetu uveliko se doprinosi povećanju broja korisnika on-line trgovine. Pritom se smanjuju troškovi izdavača kartica.

Povećanjem prometa Internet-trgovci bi osigurali povoljnije popuste i nabavne cene, što bi doprinelo i povoljnijoj kupovini za krajnjeg korisnika. Moguće je i smanjenje

procenta provizije izdavača kartice zbog povećanog prometa i smanjenih troškova zloupotrebe, što se opet može pozitivno odraziti na trgovca i kupca.

EMV definiše korišćenje pametnih kartica u e-trgovini koristeći SET protokol (engl.***S**ecure **E**lectronic **T**ransaction*).

Postoje dve glavne vrste pronevera kartica: tzv. ***skimming*** (u slobodnom prevodu, "krađa zapisa") i tzv. ***counterfeiting*** ("krivotvorenje"). Skimming je slučaj kad se podaci sa zapisa "skinu" i kasnije koriste u nelegalne svrhe kada kartica nije više prisutna, na primer, na Internetu. EMV rešava taj problem šifarskom zaštitom broja kartice i autentifikacijom.

Kod krivotvorenja, kartice su na prvi pogled jednake legitimnim (npr. u slučaju magnetnim karticama), s tim što, najčešće, podaci otisnuti na kartici, ispupčeni na telu kartice ne odgovaraju zapisanim podacima. Zapisani su podaci ili ukradeni skimmingom ili uopšte ne postoje pa se kartica čini oštećenom. U tom slučaju se trgovac koristi ručnom naplatom. EMV infrastruktura to sprečava autentifikacijom čipa na terminalu kao i opcionalno on-line autentifikacijom na serveru banke.

3.2.1.2. Učinak procesiranja stalno rastućeg broja transakcija

Broj kreditnih transakcija rapidno se povećavao proteklih godina, a tendencija se nastavlja i u budućnosti. POS terminali koji prihvataju magnetne kartice traže on-line vezu sa bankom radi autorizacije kartica. EMV smart kartice povećavaju učinak takvih transakcija, a većina ih se može izvoditi u off-line modu, štedeći pri tome vreme odvijanja transakcije i novac, bez stalno dostupnih telefonskih i računarskih mreža. To sve zavisi od tzv. *risk management* politike banke izdavača kartice. EMV platna aplikacija može se dobro konfigurisati za off-line način rada. Za kontrolu rizika i pojednostavljenje procesiranja potrebno je definisati tzv. *Lower Consecutive Offline Limit* (**LCOL**), tj. niži uzastopni off-line limit, i *Upper Consecutive Offline Limit* (**UCOL**), tj. viši uzastopni off-line limit. Ukoliko je prilikom transakcije dosegnut jedan od ovih parametara, iduća se transakcija odvija on-line. Tada banka pokreće skripte za prikupljanje transakcijskih podataka i opcionalno, za update EMV parametara.

3.2.1.3. Interoperabilnost različitih platnih aplikacija i dodatnih usluga

Preduslov je za nastajanje globalne platne šeme kartica da sve transakcije budu procesirane bilo gde u svetu, nezavisno od granica. Taj se preduslov generalno naziva interoperabilnost i ima dva važna aspekta. Sa perspektive finansijske platne industrije znači da uređaji mogu procesirati platne kartice različitih platnih šema. Sa gledišta izdavača kartice znači mogućnost korišćenja kreditne kartice na svakoj lokaciji na kojoj je prikazan odgovarajući logotip, nezavisno od tehnologije uređaja za čitanje. To uključuje međudržavne transakcije i kupovine prilikom kojih se ne pokazuje kartica, kao što je kupovina preko Interneta. Gde god je prikazan odgovarajući logotip, plaćanje karticom mora funkcionisati. Korisnici su u mogućnosti koristiti svoju kreditnu karticu u bilo kojoj državi sa istom lakoćom i na isti način kao i u svojoj zemlji.

3.2.2. Hronologija EMV

1993: Formiranje EMV radne grupe
1994: Nastanak prvih specifikacija
1996: Izdaju se osnovne EMV specifikacije
1996: UKIS 3.0 (VISA) specifikacije
1996: Objavljivanje VIS 1.2 specifikacije
1997: Prvi EMV pilot (UKIS) u Engleskoj
1998: VIS 1.3.1 specifikacije
1998: Objava EMV '96 3.1.1 – prve verzije EMV standarda
1998: "EPI minimum requirements" specifikacije
1998: "EPI Card Technical Specs." V. 3.0.2
1998: Prvi EPI pilot (Off the Shelf) u Istočnoj Evropi
1999: Nacionalni rollout UKIS u Engleskoj
1999: "EPI Off the Shelf 3.0.3" specifikacije
1999: Nacionalni rollout UKIS u Slovačkoj
2000: VISA EMV projekt (VSDC) u SAD-u
2000: Izdate EMV 2000 4.0 specifikacije
2004: Revizija EMV 2000 4.0 specifikacije i dopuna u u verziji EMV v.4.1
(Napomena: Podaci sa sajta www.emvco.com [datum pristupanja 08.jan.2007]).

3.2.3. Sigurnost i rukovanje ključevima kod pametnih kartica

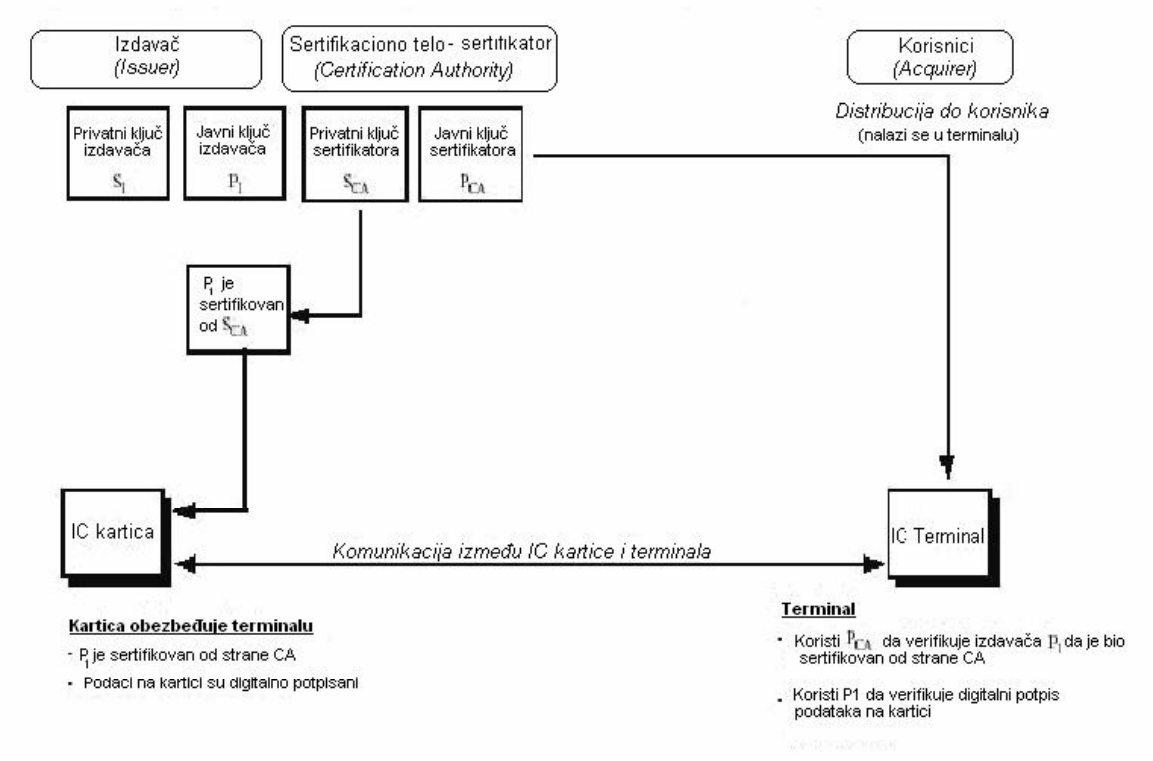
U ovome delu standarda opisuju se minimalni funkcionalni zahtevi za sigurnost komunikacije između kartice i terminala koji osiguravaju ispravnu operaciju i interoperatibilnost. U to spada offline statička i dinamička autentifikacija podataka, offline PIN enkripcija, aplikacija koja generiše kriptogram i autentifikuje izdavača, sigurnosno slanje poruka, principi rukovanja javnim ključem i sigurnost terminala.

3.2.3.1. Statička autentifikacija podataka

Statičku autentifikaciju podataka izvršava terminal upotrebljavajući digitalni potpis. Šema digitalnog potpisa je bazirana na tehnici javnog ključa pomoću koje se potvrđuje legitimnost kritičnih podataka na kartici. Kritični statični podaci se identifikuju pomoću **AFL**-a (engl.***A**pplication **F**ile **L**ocator*). Pomoću toga se detektuju neautorizovane izmene nad podacima nakon identifikacije izdavača.

Statička autentifikacija podataka zahteva postojanje sertifikatora (engl. ***C**ertification **A**uthority*, tj. treća strana od poverenja koja dokazuje vezu između javnog ključa (engl.***P**ublic **K**ey*) i izdavača (engl.***I**ssuer*)), kao sredstva visokog nivoa sigurnosti za označavanje javnog ključa izdavača. Svaki terminal koji odgovara toj specifikaciji mora sadržati prikladan sertifikator javnih ključeva koji prepoznaje svaka aplikacija u

terminalu. Ta specifikacija dopušta višestruke AID-ove (engl.***A**pplications **ID**entifier*) koji deli isti sertifikator javnog ključa. Poveznost između podataka i šifarskih ključeva je prikazana na slici 3.31:



Slika 3.31. Dijagram statičke autentifikacije podataka

Kartice koje podržavaju statičku autentifikaciju morale bi sadržati sledeće elemente podataka:

- *Indeks sertifikatora javnog ključa*: jednobajtni element podataka koji sadrži binarni broj koji određuje koji će aplikacioni javni ključ sa svojim asocijativnim algoritmom biti upotrebljen u operaciji između kartice i terminala,
- *Sertifikat javnog ključa izdavača*: element podataka promenjive veličine koji osigurava odgovarajući sertifikator izdavača kartice. Nakon što terminal ispita taj element podataka, autentifikuje javni ključ izdavača sa dodatnim podacima,
- *Označeni statički aplikacioni podaci*: element podataka promenjive veličine generisan od izdavača upotrebom privatnog ključa koji korespondira sa javnim ključem autentifikovanim sa sertifikatom javnog ključa izdavača,
- *Podsetnik javnog ključa proizvođača,*
- *Predstavnik javnog ključa proizvođača.*

Za podržavanje statičke autentifikacije, svaki terminal mora imati mogućnost spremanja šest sertifikatora javnih ključeva pomoću RID-a (engl. ***R**egistered **A**pplication **P**rovider **I**dentifier*). Isto tako mora znati povezivati svaki ključ sa određenom

informacijom koja onda može biti upotrebljena zajedno sa ključem. Terminal mora još biti u mogućnosti locirati ključ i relevantnu informaciju datu RID-u osiguranu od kartice.

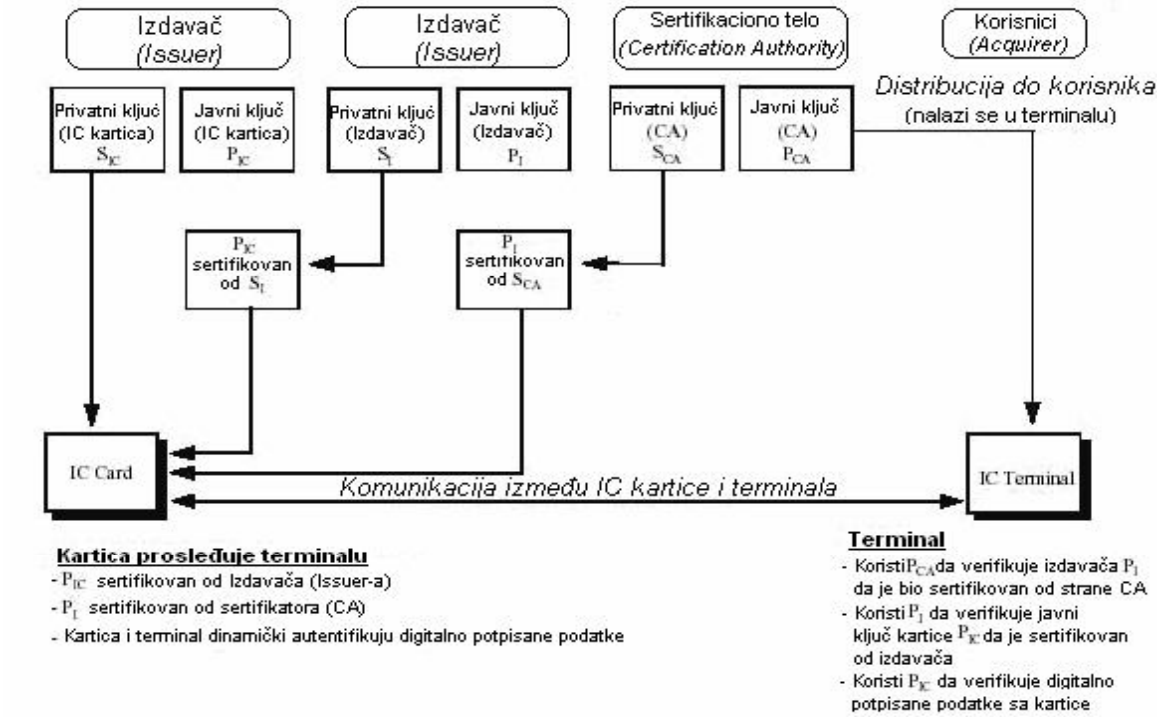
3.2.3.2. Dinamička autentifikacija podataka

Dinamičku autentifikaciju podataka sprovodi terminal koji upotrebljava digitalni potpis. Šema digitalnog potpisa se bazira na tehnikama javnog ključa za autentifikaciju kartice i potvrdu legitimnosti kritičnih podataka na kartici (rezidentnih i generisanih od kartice) i podataka primljenih od terminala.

Postoje dve opcije:

- Standardna dinamička autentifikacija podataka izvršava se pre analize akcije sa karticom, gde kartica generiše digitalni potpis nad rezidentnim/generisanim podacima na kartici,
- Kombinacija dinamičke autentifikacije podataka i aplikacije za generisanje kriptograma (rezultata šifarske operacije) se izvršava kod izdavanja prve GENERATE AC naredbe.

Dinamička autorizacija podataka zahteva postojanje sertifikatora kao sredstva visokog nivoa sigurnosti za označavanje javnog ključa izdavača. Svaki terminal koji odgovara toj specifikaciji mora sadržati prikladnu autorizaciju sertifikata javnog ključa koji prepoznaje svaka aplikacija u terminalu. Ta specifikacija dopušta višestruke **AID**-ove (engl. ***A**pplications **I**dentifier*) koji dele istu autorizaciju sertifikata javnog ključa. Poveznost između podataka i šifarskih ključeva je prikazana na slici 3.32:



Slika 3.32. Dijagram dinamičke autentifikacije podataka

Kartica koja podržava dinamičku autentifikaciju podataka mora sadržati sledeće elemente podataka:

- *Indeks sertifikatora javnog ključa*: jednobajtni element podataka koji sadrži binarni broj koji određuje koji će aplikacioni javni ključ sa svojim asocijativnim algoritmom biti upotrebljen u operaciji između kartice i terminala,
- *Sertifikat javnog ključa izdavača*: element podataka promenjive veličine koji osigurava odgovarajući sertifikat autorizacije izdavača kartice. Nakon što terminal ispita taj element podataka autentifikuje javni ključ izdavača sa dodatnim podacima,
- *Sertifikat javnog ključa kartice*: element podataka promenjive veličine koji osigurava proizvođač kartice. Nakon što što terminal ispita taj element podataka autentifikuje javni ključ kartice sa dodatnim podacima,
- *Podsetnik javnog ključa proizvođača,*
- *Predstavnik javnog ključa proizvođača,*
- *Podsetnik javnog ključa kartice,*
- *Predstavnik javnog ključa kartice,*
- *Privatni ključ kartice.*

Kartica koja podržava dinamičku autentifikaciju podataka mora generisati sledeći element podataka:

- Označena dinamička aplikacija podataka: element podataka promenjive veličine generisan od kartice koristeći privatni ključ koji korespondira sa javnim ključem autentifikovanim u sertifikatu javnog ključa kartice.

Za podržavanje dinamičke autentifikacije, svaki terminal mora imati mogućnost baferovanja šest sertifikatora javnih ključeva pomoću **RID**-a (engl.***R**egistered **A**pplication **P**rovider **I**dentifier*). Isto tako mora znati povezivati svaki ključ sa određenom informacijom koja onda može biti upotrebljena zajedno sa ključem. Terminal mora još biti u mogućnosti da locira ključ i relevantnu informaciju datu RID-u osiguranu od kartice.

3.2.3.3. Enkripcija ličnog identifikacionog broja (PIN)

Ako je podržana, enkripcija ličnog identifikacionog broja (**PIN**) za off-line proveru se izvodi od strane terminala koji upotrebljava asimetrični mehanizam enkripcije za sigurno prenošenje identifikacionog broja sa tastature do kartice. Preciznije, kartica mora posedovati par javnih ključeva povezanih sa šifrovanim identifikacionim brojem. Javni ključ upotrebljava tastatura za unošenje identifikacionog broja ili sigurna komponenta terminala (koja nije tastatura) za enkripciju identifikacionog broja, a privatni ključ upotrebljava kartica za proveru šifrovanog identifikacionog broja.

Sigurna komponenta terminala se upotrebljava za enkripciju identifikacionog broja, jer transfer identifikacionkog broja sa tastature do sigurne komponente mora biti na visokom nivou sigurnosti.

3.2.3.4. Aplikativni kriptogram i autentifikacija izdavača

Pod aplikativnim kriptogramom se podrazumeva protokol komunikacije između kartice i terminala.

3.2.3.4.1. Generisanje aplikativnog kriptograma

Aplikativni kriptogram se sastoji od poruke autentifikacionog koda:
- poslatog od strane terminala prema kartici (izvršavanjem GENERATE AC ili neke druge naredbe),
- preuzetog interno sa kartice.

Preporučeni minimalni set podataka od kojih bi se moralo sastojati generisanje kriptograma se nalaze u sledećoj tabeli 3.27:

Vrednost	Izvor
Vrednost Autorizacije (tip Numeric)	Terminal
Ostale Vrednosti (tip Numeric)	Terminal
Terminal Kôd Države	Terminal
Terminal Vrednosti Verifikacije	Terminal
Transakcioni kôd prometa	Terminal
Datum Transakcije	Terminal
Tip Transakcije	Terminal
Nekarakterističan Broj	Terminal
Aplikacija Razmene Profila	Kartica
Brojač Transakcija Aplikacije	Kartica

Tabela 3.27. Preporučeni minimalni set podataka za generisanje kriptograma

Algoritam za generisanje kriptograma uzima kao ulaz 16-bajtni glavni ključ (engl.*Master Key*) i ostale potrebne podatke pa iz njih izračunava 8-bajtni kriptogram u sledeća dva koraka:

1. Prvi korak se sastoji u derivaciji funkcije sesijskog ključa koji se sastoji od 16-bajtnog sesijskog ključa kriptograma i 2-bajtnog **ATC**-a (engl.***A**pplication **T**ransaction **C**ounter*),

2. Drugi korak se sastoji u generisanju 8-bajtnog kriptograma uz pomoć **MAC** algoritma (engl.***M**essage **A**uthentication **C**ode*) i sesijskog ključa kriptograma iz prethodnog koraka.

3.2.3.4.2. Autentifikacija izdavača

Postupak generisanja 8-bajtnog **ARPC**-a (engl.***A**uthorisation **R**esponse **C**ryptogram*) se dobija upotrebom *Triple-DES* algoritma. Kao ulazni podaci za algoritam su 8-bajtni ARQC generisan od strane kartice i 2-bajtni ARC. Upotrebljava se još i sesijski ključ kriptograma *SKac* na sledeći način:

1. ARC se proširuje sa 6 bajta da bi postao 8-bajtni broj:
   X := (ARC || '00'|| '00' || '00' || '00' || '00' || '00').
3. Izračunava se Y: = ARQC ⊕ X.
4. 8-bajtni ARPC se na kraju dobija kao ARPC := DES3(SKac)[Y].

3.2.3.5. Sigurno slanje poruka

Ciljevi sigurnog slanja poruka su osiguravanje poverljivosti, intergritet podataka i autentifikacija pošiljaoca. Integritet podataka i autentifikacija izdavača su postignute pomoću **MAC**-a (engl. ***M**essage **A**uthentication **C**ode*). Poverljivost podataka se ostvaruje upotrebom šifrovanja polja podataka.

3.2.3.5.1. Formati poruka

Postoje dva formata poruka:

- ***Format 1*:** Format je nastao prema standardu ISO/IEC 7816-4, poglavlje 5.6. Nad poljem podataka je sprovedeno BER-TLV kodiranje po pravilima ASN.1/ISO 8825 standardu. To je eksplicitno određeno u bitovima najmanje težine bajta naredbe. To ističe da je zaglavlje naredbe uvek integrisano u izračunavanje **MAC**-a.

- ***Format 2*:** Format kod kog se ne upotrebljava BER-TLV kodiranje. U ovom slučaju pošiljalac naredbe mora znati koji podaci se šalju i koje su veličine. Format je eksplicitno definisan u najnižim bitovima naredbe.

3.2.3.5.2. Sigurno slanje poruka u svrhu zaštite integriteta i autentifikacije

Polje podataka naredbe

FORMAT 1:

Polje podataka naredbe je sastavljeno od sledećih TLV objekata podataka kako je to prikazano na na slici 3.33:

Tag 1	Dužina 1	Vrednost 1	Tag 2	Dužina 2	Vrednost 2
T	L	Vrednost (L bajt)	' 8E '	' 04 ' - ' 08 '	MAC (4-8 bajta)

Slika 3.33. Format 1 polja podataka naredbe u svrhu zaštite inegriteta i autentifikacije

Ako postoje, podaci o naredbi moraju biti posebno eksplicitno naznačeni.

Ako je polje podataka BER-TLV kodirano, ne bi smelo pripadati specifičnom kontekstu (obeležje ne sme biti u rasponu od "*80*" do "*BF*") ili mora imati oznaku sa neparnom vrednošću (treba voditi računa da to može biti i složeni objekt podataka).

Ako polje podataka nije BER-TLV kodirano, oznaka ima vrednost "81".

Drugi objekt podataka je MAC. Zastavica (engl.*Flag*) je "8E", a dužina mora biti u rasponu od 4 do 8 bajta.

FORMAT 2:

Elementi podataka (uključujući i MAC) su sadržani u polju podataka i njihovu dužinu bi morao znati pošiljalac naredbe. MAC nije BER-TLV kodiran i mora biti uvek na poslednjem mestu polja podataka. Njegova dužina mora biti od 4 do 8 bajta (slika 33).

Vrednost 1	Vrednost 2
Komandni podaci (ako postoje)	MAC (4-8 bajta)

Slika 3.34. Format 2 polja podataka naredbe u svrhu zaštite inegriteta i autentifikacije

Izračunavanje MAC-a

Izračunavanje MAC-a S nad porukom MSG pomoću sesijskog ključa Ks se izvodi u sledećim koracima:

1. Dodavanje oznake "80" sa desne strane poruke i tada dodavanje još najmanji broj "00" sa desne tako da je dužina rezultujuće poruke umnožak broja 8 (bajta):
   MSG := (MSG || '80' || '00' || '00' || . . . || '00').
   MSG se tada još deli u 8-bajtne blokove X1, X2, …, Xk.

2. MAC sesijski ključ Ks se sastoji samo od krajnje levog bloka ključa Ks = Ksl ili konkatenacije (*sastavljanja*) krajnje levog i krajnje desnog bloka ključa
   Ks = (Ksl | | Ksr).

3. Izračunavanje kriptograma

   Upotrebljavanjem 8-bajtnih blokova X1, X2, …, Xk i krajnje levog MAC sesijskog ključa Ksl izračunava se:

$$Hi := ALG(Ksl)[Xi \oplus Hi\text{-}1], \text{ za } i = 1, 2, \ldots, k.$$

Inicijalna vrednost H0 := ('00' || '00' || '00' || '00' || '00' || '00' || '00' || '00').
"ALG" je funkcija šifrovanja.

Izračunavanje 8-bajtnih blokova Hk+1 se obavlja na jednan od sledećih načina:

- prema standardu ISO/IEC 9797-1 algoritam 1:

$$Hk+1 := Hk.$$

- prema standardu ISO/IEC 9797-1 algoritam 3:

$$Hk+1 := ALG(Ksl)[ALG\text{-}1(Ksr)[Hk]].$$

MAC S onda je jednak s (s je prema potrebi od 1 do 8) bajtova najveće težine od Hk+1.

3.2.3.5.3. Sigurno slanje poruka za osiguranje poverljivosti

Polje podataka naredbe

FORMAT 1:

Format šifrovanih podataka u polju podataka naredbe je prikazano na slici 3.35:

Tag	Dužina	Vrednost
T	L	Kriptogram (šifrovani podaci)

Slika 3.35. Format 1 šifrovanih objekata podataka u polju podataka naredbe

Zavisno u tekstu koji se šifruje, ISO/IEC 7816-4 određuje obeležje; oznaku (engl. *Tag*) koja će biti dodata rezultujućem šifratu: neparna oznaka se upotrebljava kada se objekt koristi u izračunavanju MAC-a, a inače se upotrebljava parna oznaka.

FORMAT 2:

Šifrovani podaci se odnose na celi tekst bez MAC-a:

Vrednost 1	Vrednost 2
Kriptogram (šifrovani podaci)	MAC (ako postoji)

Slika 3.36. Format 2 polja podataka naredbe za sigurno slanje poruka

Kriptogram i MAC se dobijaju na sličan način kako je to navedeno u prethodnom poglavlju.

3.2.3.6. Javni ključ sertifikatora

U ovom poglavlju definiše se pravac principa i načina rukovanja sa platnim sistemom koji koristi statičku i dinamičku autentifikaciju podataka.

Principi su koncepti kao osnovica za implementaciju rukovanja javnim ključem sertifikatora. Ti principi definišu postupke koji se koriste u svim platnim sistemima, ili su adaptirani u individualnim platnim sistemima. Svaki platni sistem će razviti svoj set procedura za implementaciju.

3.2.3.6.1. Životni ciklus javnog ključa sertifikatora

Normalni životni ciklus javnog ključa sertifikatora

Životni ciklus javnog ključa sertifikatora se u normalnim okolnostima može podeliti na sledeće faze:

- planiranje,
- generisanje,
- distribucija,
- upotreba,
- poništavanje.

Planiranje

Tokom faze planiranja, platni sistem istražuje zahteve za uvođenje novog para ključeva sertifikatora u bližoj budućnosti. Ti zahtevi su vezani uz broj potrebnih ključeva i parametara tih ključeva.

Važan deo faze planiranja je pregled sigurnosti RSA za odlučivanje, o dužini života novih i potencijanih ključeva. Primarna funkcija pregleda je u tome da donese odluke o postavkama dužina trajanja validnosti novih ključeva i datuma isteka validnosti novih ključeva kao i potencijalne modifikacije u odnosu na postojeće ključeve i zamena ključeva.

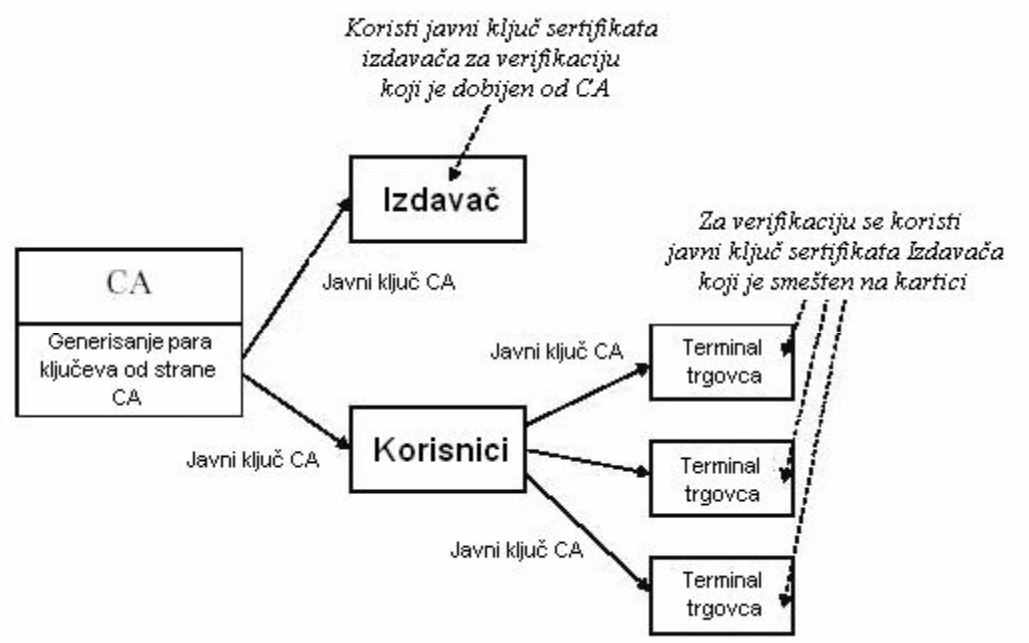
Generisanje

Ako rezultat faze planiranja zahteva uvođenje novog para ključeva sertifikatora, oni moraju biti generisani od platnog sistema. Preciznije, deo platnog sistema za autorizaciju sertifikata (fizička i logička infrastruktura sa visokim nivoom sigurnosti upravljanja platnim sistemom) će generisati na siguran način potreban par RSA ključeva za autorizaciju sertifikata (ključeva sertifikatora) za buduću upotrebu.

Zato se posebno mora izgraditi podsekvenca koja se brine za tajnost privatnog ključa za autorizaciju sertifikata. Takođe se mora omogućiti integritet privatnih i javnih ključeva.

Distribucija

U fazi distribucije ključa, deo platnog sistema za autorizaciju sertifikata će izvršiti distribuciju novo-generisanih javnih ključeva za autorizaciju sertifikata traženim izdavačima (engl.*Issuer*) i korisnicima (engl.*Acquirer*) za sledeće svrhe (slika 3.36):



Slika 3.37. Distribucija javnog ključa za autorizaciju sertifikata

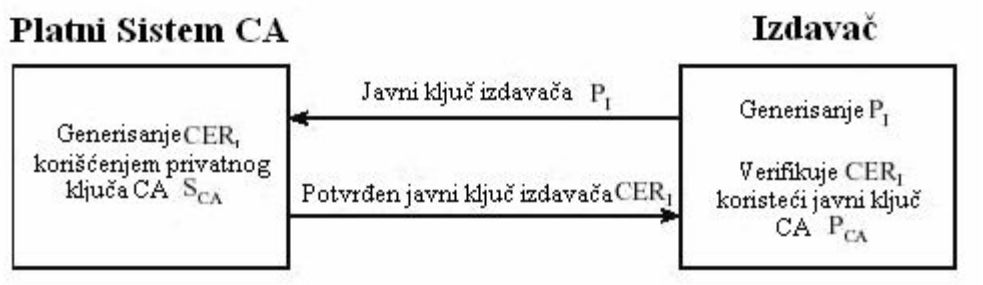
- Izdavač verifikuje javni ključ sertifikatora izdatog od platnog sistema tokom faze korišćenja ključa
- Korisnici ga koriste za siguran prenos javnih ključeva za autorizaciju sertifikata do terminala trgovca

Da bi se sprečilo uvođenje lažnih javnih ključeva za autorizaciju sertifikata, interfejsi između platnog sistema, izdavača i korisnika moraju osigurati integritet distribucije javnih ključeva.

Upotreba

Javni ključ sertifikatora se upotrebljava u terminalima trgovaca da bi se mogla sprovoditi statička ili dinamička autorizacija podataka.

Privatni ključ sertifikatora je upotrebljen u delu platnog sistema za autorizaciju sertifikata za generisanje sertifikata javnog ključa izdavača. Preciznije, događaju se sledeće interakcije (slika 3.37):



Slika 3.38. Distribucija javnog ključa izdavača

- izdavač generiše svoj javni ključ izdavača i šalje ga platnom sistemu,
- platni sistem označi javni ključ izdavača sa privatnim ključem sertifikatora da bi dobio sertifikat javnog ključa izdavača i vraća ga izdavaču,
- sa javnim ključem sertifikatora, izdavač proverava ispravnost primljenog sertifikata javnog ključa izdavača. Ako je u redu, izdavač može tada uključiti svoj deo podataka za otkrivanje identiteta sa podacima kartice.

Da bi se sprečilo uvođenje lažnih javnih ključeva izdavača, interfejs između izdavača i platnog sistema mora osigurati integritet javnog ključa izdavača koji je izdat za sertifikaciju.

Poništavanje

Kad par ključeva ispuni svoj radni vek tokom faze planiranja, moraju se odstraniti iz usluge. U praksi, to znači sledeće:

- kad im istekne radni vek, sertifikat javnog ključa izdavača zajedno sa privatnim ključem sertifikatora prestaju biti važeći. Izdavač zato mora osigurati da kartice personalizovane sa tim sertifikatom javnog ključa izdavača prestaju važiti onda kada i zadati par ključeva,
- određeno vreme pre isteka validnosti, platni sistem će prestati označavati javne ključeve izdavača sa korespondiranim privatnim ključem sertifikatora,
- Kad isteknu, korisnici moraju imati mogućnost da sa lakoćom odstrane javne ključeve za autorizaciju sertifikata iz usluge u terminalu.

Kompromitovanje para javnih ključeva za autorizaciju sertifikata

U slučaju kompromitovanja (probijanje tajnosti ili sigurnosti) para javnih ključeva sertifikatora, mora se pokrenuti hitni postupak koji može na kraju rezultovati ubrzanim poništavanjem para javnih ključeva sertifikatora pre njihovog isteka radnog veka. U ovom slučaju postoje sledeće faze životnog ciklusa ključa:

- detektovanje,
- procena,
- odluka,
- poništavanje (ubrzano).

Detektovanje

Kompromitovanje para javnih ključeva sertifikatora može biti aktuelno kompromitovanje, na primer potvrđeni sigurnosni prekid u platnom sistemu, ili potvrđeno probijanje ključa šifarskom analizom. Kompromitovanje može biti:

- ***Očekivano:*** praćenje sistema ili član transakcije i vlasnik kartice zapaža sprovođenje lažnih transakcija koje mogu biti zbog kompromitovanja ključa, ali nije potvrđeno, ili
- ***Potencijalno:*** tehnikama šifarskih analiza, na primer faktorizacijom, se razvila dužina ključa koji može biti kompromitovana, ali nema dokaza da je to istina.

Detektovanje kompromitovanog ključa može nastati zbog aktuelnog fizičkog provaljivanja u platni sistem. To se može zapaziti preko izveštaja lažnih off-line transakcija između platnog sistema i članova transakcija. Probijanje se izvodi inteligentnim unapređenim faktorizovanjem otkrivene od strane kriptografske zajednice.

Procena

Procena potencijalnog kompromitovanja para ključeva će uključiti tehničke, rizične, lažne, i, najvažnije, udare na poslovnu sigurnost između platnog sistema i članova transakcija. Rezultat procene uključuje potvrđivanje kompromitovanja, određivanje mogućih smerova akcija protiv troškova i rizika kompromitovanja, i prezentovanje rezultata procene za donošenje odluke.

Odluka

Na osnovi rezultata faze procene, platni sistem odlučuje o akciji koja će biti preduzeta u vezi kompromitovanja ključa. U najgorem slučaju, ova odluka povlači dotični ključ iz upotrebe pre njegovog isteka radnog veka.

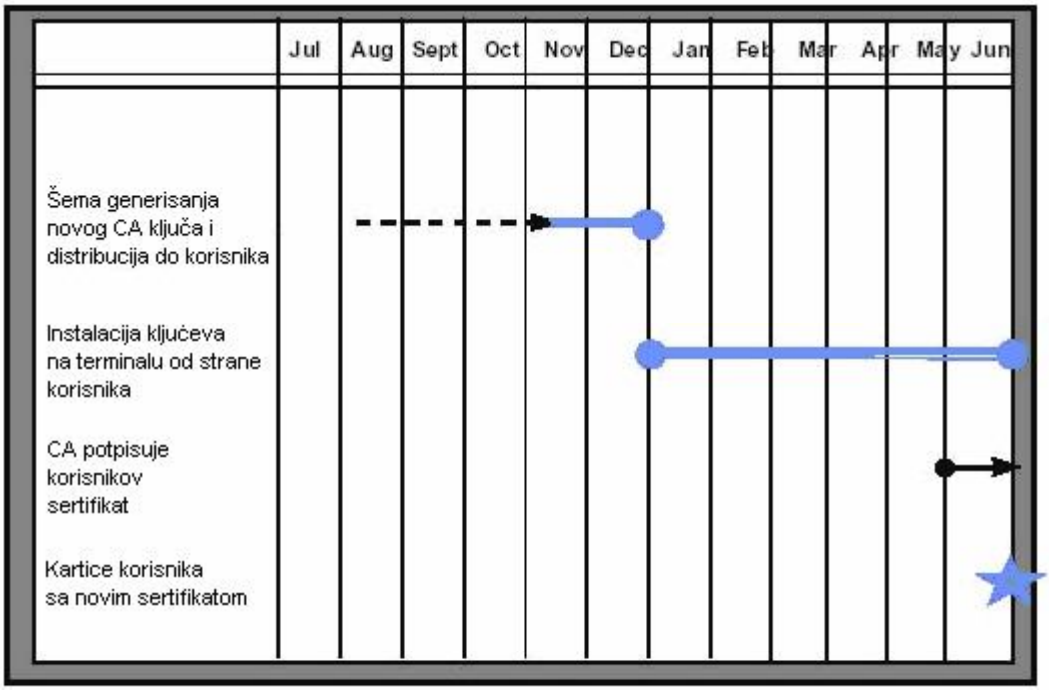
Poništavanje (ubrzano)

Poništavanje dotičnog ključa vodi do određivanja trajanja novog radnog veka tog ključa. Postupak je isti kao što je opisan u prethodnom odeljku.

3.2.3.6.2. Vremenski odnosi

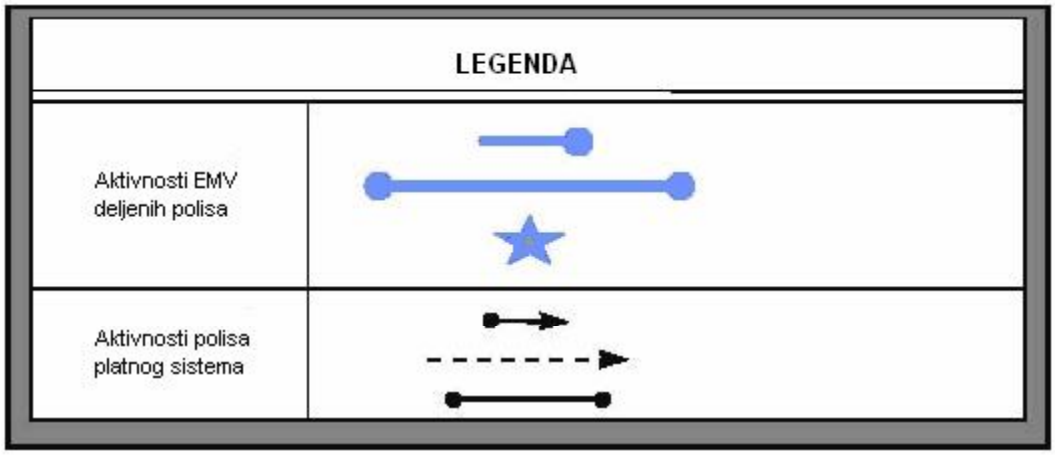
Sledeći dijagrami prikazuju vremenske odnose za poništavanje i uvođenje javnih ključeva sertifikatora. Svaka vremenska linija predstavlja raspored uvođenja ili ukidanja ključa. U slučaju ubrzanog uvođenja ili ukidanja ključa, vremena ostaju ista, ali za mesece uvođenja i ukidanja ključa brine platni sistem.

Uvodjenje ključa



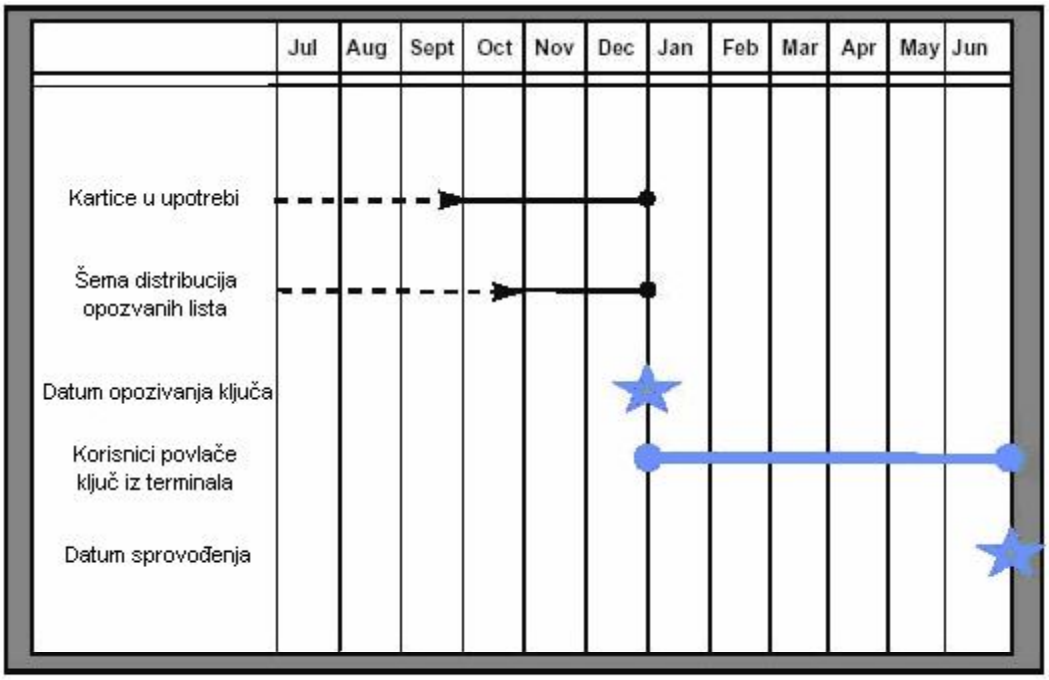
Slika 3.39. Dijagram uvođenja ključa

Legenda značenja pojedinih oznaka se nalazi na sledećoj slici:



Slika 3.40. Legenda oznaka dijagrama

Ukidanje ključa



Slika 3.41. Dijagram ukidanja ključa

3.2.3.7. Sigurnost terminala i zahtevi za rukovanje ključevima

Ovo poglavlje opisuje generalne zahteve terminala za rukovanje osetljivim podacima, kao što su tekst identifikacionog broja (**PIN**) ili kriptografski ključevi. Preciznije, imenuje zahteve za sigurnost tastature za unošenje identifikacionog broja i zahteve za rukovanje javnim ključevima za autorizaciju sertifikata.

3.2.3.7.1. Sigurnosni zahtevi

Sigurni uređaji (engl.Tamper-Evident Devices)

Sigurni uređaji moraju u normalnom radnom okruženju osigurati da uređaj ili njegov interfejs ne otkriva ili menja osetljive podatke koji ulaze ili izlaze iz uređaja, ili koji su spremljeni ili se obrađuju u uređaju.

Kada siguran uređaj radi u sigurnosno-kontrolisanom okruženju, zahtevi za karakteristike uređaja su redukovani jer se zaštita sprovodi sigurnosno-kontrolisanim okruženjem i rukovanjem uređaja.

Fizička sigurnost

Siguran uređaj mora biti dizajniran tako da onemogućuje fizički pristup interno spremljenim osetljivim podacima i da odvraća krađu, neautorizovanu upotrebu ili neautorizovanu modifikaciju opreme. Navedeni ciljevi zahtevaju otpornost, detekciju ili mehanizme zaštite kao što su vizuelni ili zvučni alarmi.

Siguran uređaj, kada ne obavlja nikakve operacije, ne bi smeo sadržati nikakve šifarske ključeve ili druge osetljive podatke (na primer PIN-ove) koji su korišćeni u nekoj od prethodnih transakcija. Probijanje u sistem može biti bez gubljenja sigurnosti tj. ne detektuje ih uređaj, zato se takva probijanja moraju detektovati pre uređaja i pre nego što se šifarski ključevi i osetljivi podaci opet stave u operativni sistem. Ako je uređaj dizajniran tako da je omogućen interni pristup, moraju se odmah osetljivi podaci obrisati, čim dođe do probijanja u sistem. Sigurni uređaj je zavistan od fizičke sigurnosti što se tiče detekcije napada. Zato mora biti dizajniran tako da ima svojstva koja odmah vlasnika kartice ili trgovca upućuju na zločin.

Uređaj mora biti dizajniran i sastavljen tako da se:

- Ne može izvesti probijanje u uređaj u smislu izvođenja nekih promena, dodataka ili zamena hardverske i programske opreme uređaja; ili promene osetljivih podataka i naknadno reinstaliranje uređaja bez upotrebe posebnih veština i uređaja koji nisu generalno dostupni, i bez oštećenja uređaja tako da se ne bi moglo detektovati oštećenje,
- Svaki neautorizovani pristup ili izmena osetljivih podataka, koji se unose, spremaju ili obrađuju, se pripisuje samo aktuelnom probijanju uređaja,
- Nije omogućeno na bilo kakav način krivotvorenje i izrađivanje sličnih komponenti od kojih je uređaj sastavljen,
- Kvar na bilo kom delu uređaja ne sme prouzrokovati oštećenje tajnih i osetljivih podataka,
- Ako je uređaj dizajniran tako da delovi mogu biti fizički odvojeni od uređaja, svako obrađivanje podataka između vlasnika kartice i tih odvojenih delova mora imati isti nivo sigurnosti kao što je ima i celi, nerastavljeni uređaj,
- Integrisanje drugačijih delova uređaja u kućište sigurnog uređaja je nužan uslov za izmenu osetljivih podataka kao što je PIN.

Logička sigurnost

Siguran uređaj mora biti dizajniran tako da nijedna funkcija ili kombinacija funkcija ne prouzrokuje otkrivanje osetljivih podataka, osim ako je to eksplicitno dozvoljeno od strane sigurnosti implementirane u terminalu. Logička zaštita mora uvek štititi osetljive podatke, pa i ako su samo upotrebljene legitimne funkcije. To svojstvo se

može postići internim praćenjem statistike ili merenjem minimalnog vremena između poziva osetljivih funkcija.

Ako terminal može doći u osetljivo stanje (stanje u kome se mogu pozivati funkcije koje nisu omogućene u normalnom načinu rada, na primer, korisničko rukovanje šifarskim ključevima), takav prenos mora biti dodatno nadgledan od dve ili više poverljivih strana. Ako su lozinka ili drugi tekstualni podaci upotrebljeni kod kontrole prenosa u osetljivom stanju, unos te lozinke ili tekstualnih podataka mora biti zaštićen na način kao i ostali osetljivi podaci.

Za minimizovanje rizika koji rezultuje neautorizovanom upotrebom osetljivih funkcija, osetljivo stanje mora biti postignuto sa ograničenim brojem poziva funkcija (gde je to prikladno) i vremenskim ograničenjem. Ako je neko od tih ograničenja postignuto, uređaj se mora vratiti u normalno stanje.

Siguran uređaj mora automatski isprazniti svoje unutrašnje bafere (engl.*buffers*) na kraju svake transakcije ili u time-out situaciji.

Tastatura za unos identifikacionog broja (engl. PIN Pad)

Tastatura za unos identifikacionog broja mora biti siguran uređaj. Mora podržavati unos 4-12 cifrenog identifikacionog broja. Ako je priložen i ispis uz tastaturu (engl.*display*), indikacija unošenja pojedinog broja mora biti ispisana. Ali, stvarne vrednosti brojeva se ne smeju videti, već se smeju videti samo oznake (vizuelne ili zvučne) da su brojevi unešeni.

Kad terminal podržava off-line verifikaciju PIN-a, **IFD** (engl.***Inter****F****ace*** ***D****evice*) i tastatura za unos PIN-a moraju biti intergrisane u jedan uređaj ili IFD i tastatura moraju biti dva posebno odvojena uređaja.

Ako su IFD i tastatura integrisani i offline PIN se šalje kartici u obliku teksta, tastatura ne šifruje offline PIN kada je tekstualni PIN poslat od tastatura do IFD-a.

Ako su IFD i tastatura integrisani i offline PIN se šalje kartici u obliku teksta, ali offline PIN tekst nije poslat direktno od tastature do IFD-a, tada tastatura mora šifrovati offline PIN za prenos do IFD-a. U tom slučaju IFD dešifruje offline PIN za prenos u tekstu do kartice.

Ako IFD i tastatura nisu integrisani zajedno, a offline PIN se mora poslati kartici u tekstualnom obliku, tastatura mora šifrovati offline PIN za prenos do IFD-a. U tom slučaju IFD dešifruje offline PIN za prenos u tekstu do kartice.

Ako se offline PIN šalje kartici u šifrovanom obliku, šifrovanje se sprovodi na:

- tastaturi samog uređaja,
- sigurnoj komponenti terminala.

Ako terminal podržava on-line verifikaciju PIN-a, kada je PIN unešen mora biti zaštićen šifrovanjem unutar terminala i terminal mora slati šifrovani PIN u skladu sa pravilima sistema plaćanja.

Ispis prompt-a za unos PIN-a mora biti generisan od same tastature. To znači da se mogu ispisivati sve vrste poruka koje su autorizovane od strane tastature, a ne samo koje su u vezi s PIN-om. Tastatura mora odbiti ispis bilo kakve neautorizovane poruke.

Za praćene terminale, unos iznosa mora biti separiran od unosa PIN-a da bi se izbeglo ispisivanje PIN-a na terminalu. Ako se PIN i iznos unose sa iste tastature, iznos mora biti proveren od vlasnika kartice pre nego što se unosi PIN.

Tastatura mora biti dizajnirana tako da štiti privatnost i tajnost, da u normalnoj upotrebi samo vlasnik može videti informacije ispisane na zaslonu terminala. Tastatura mora biti tako ugrađena da vlasnik kartice može uneti PIN sa minimalnim rizikom da su taj unos videli drugi.

Tastatura mora isprazniti svoje unutrašnje bafere u sledećim uslovima:

- pri završetku transakcije,
- u time-out situaciji, uključujući neodređeni period vremena otkako je unesen PIN.

3.2.3.7.2. Zahtevi za rukovanje ključem

Ovo poglavlje specifikuje zahteve za korisničko rukovanje javnim ključevima sertifikatora u terminalu. Zahtevi pokrivaju sledeće faze:

- uvođenje javnih ključeva sertifikatora u terminal,
- spremanje javnih ključeva sertifikatora u terminal,
- upotreba javnih ključeva sertifikatora u terminalu,
- ukidanje javnih ključeva sertifikatora iz terminala.

Uvođenje javnih ključeva sertifikatora u terminal

Kada platni sistem donese odluku o uvođenju javnih ključeva sertifikatora u terminal, proces je izvršen tako da osigurava distribuciju novog ključa od platnog sistema do korisnika. Od tog trenutka je odgovornost na korisniku da se brine o prenosu novog javnog ključa sertifikatora i podataka vezanih za njega do terminala.

Uvaženi su sledeći principi kod uvođenja javnog ključa sertifikatora od korisnika do terminala:

- Terminal mora imati mogućnost verifikacije da je javni ključ sertifikatora primljen bez greške od korisnika,
- Terminal mora imati mogućnost verifikacije da je primljen javni ključ sertifikatora originalan od legitimnog korisnika,
- Korisnik mora imati mogućnost potvrde da je javni ključ sertifikatora u potpunosti ispravno uveden u terminal.

Spremanje javnih ključeva sertifikatora u terminal

Terminal koji podržava statičku i/ili dinamičku autentifikaciju podataka mora podržavati šest javnih ključeva sertifikatora preko RID-a za kreditne kartice Mastercarda, Vise i Europay-a bazirane na EMV standardu.

Svaki javni ključ sertifikatora je jednoznačno određen 5-bajtnim RID-om koji identifikuje platni sistem i 1-bajtni indeks javnog ključa sertifikatora, jednoznačnog za RID dodeljen od platnog sistema da bude delimični javni ključ sertifikatora.

Minimalni set elemenata podataka vezanih za javni ključ sertifikatora koji se spremaju u terminal je dat u sledećoj tabeli 28:

Ime	Dužina	Opis	Format
Pokazivač na registrovanu aplikaciju provajdera	5	Određuje platni sistem sa kojim radi javni ključ CA	b
Indeks javnog ključa CA	1	Određuje konjukciju javnog ključa CA sa RID-om	b
Indikator Hash (potpisnog) algoritma CA	1	Određuje potpisni algoritam na osnuvu rezultata Hash algoritma za digitalni potpis	b
Indikator algoritma javnog ključa CA	1	Određuje digitalni potpisni algoritam koristeći javni ključ CA	b
Moduo javnog ključa CA	Promenljiva (maks.248)	Vrednost modua dela javnog ključa CA	b
Eksponent javnog ključa CA	1 ili 3	Vrednost eksponenta dela javnog ključa CA	b
Provera sume javnog ključa CA	20	Provera vrednosti izračunate konkatenacijom (sastavljenjem) svih elemenata sertifikata javnog ključa CA (RID, Indeks javnog ključa, Moduo javnog ključa CA, Eksponent javnog ključa CA) koristeći algoritam SHA-1	b

Tabela 3.28. Minimalni set elemenata podataka vezanih za javni ključ za autorizaciju sertifikata koji se spremaju u terminal

RID i indeks javnog ključa sertifikatora su jednoznačni za svaki javni ključ sertifikatora i jednoznačno povezani sa platnim sistemom.

Indikator algoritma za javni ključ sertifikatora identifikuje algoritam digitalnog potpisa koji se upotrebljava sa korenspondirajućim javnim ključem sertifikatora. Jedino prihvatljiva vrednost u tom slučaju je heksadecimalno "01", koja označava upotrebu *RSA* algoritma kod digitalnog potpisa. Indikator *hash* algoritma određuje hash algoritam koji se koristi kod digitalnog potpisa. Jedino prihvatljiva vrednost je u tom slučaju heksadecimalno "01", koja označava da se radi o *SHA-1* hash algoritmu.

Suma provere javnog ključa sertifikatora služi za ispitivanje ispravnosti transfera javnog ključa sertifikatora na odredište. Terminal može koristiti taj podatak za naknadno re-verifikovanje javnog ključa za autorizaciju sertifikata i podatke vezane uz njega. Osim tog, terminal može koristiti i druga sredstva za proveravanje integriteta tih podataka.

Integritet spremljenih podataka bi trebao biti svakog određenog vremenskog intervala.

Ukidanje javnog ključa sertifikatora

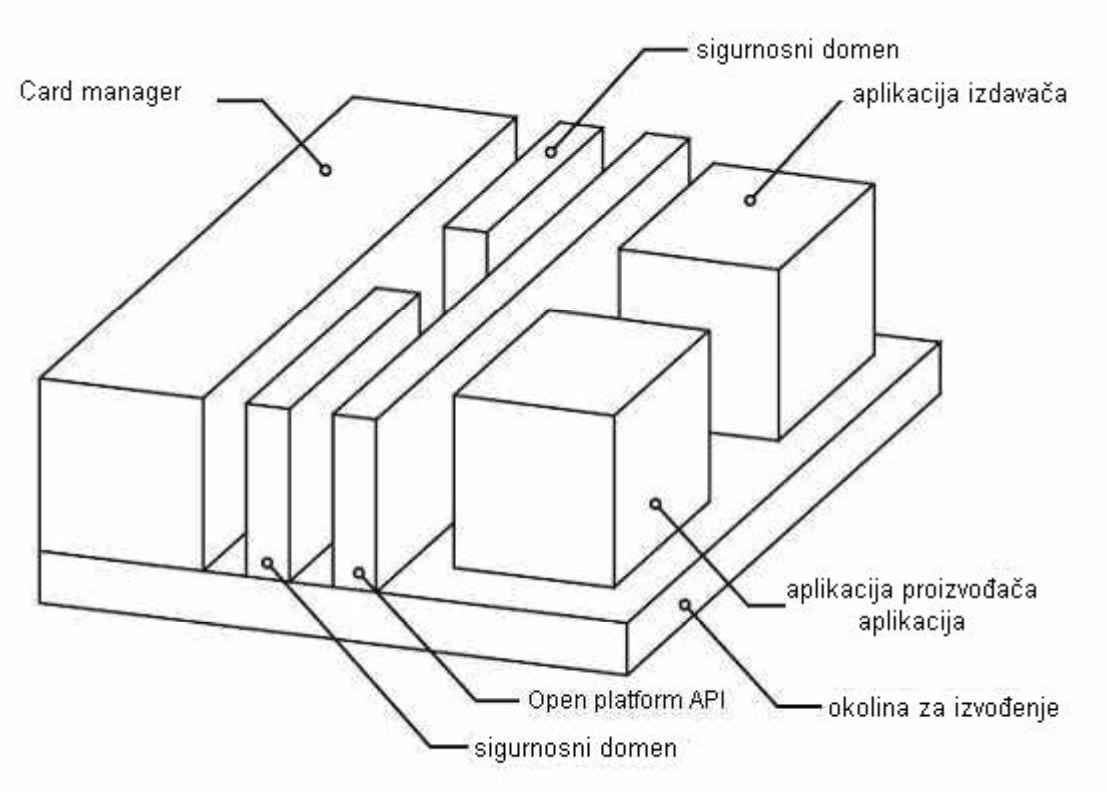
Kad platni sistem odluči povući neki javni ključ sertifikatora iz upotrebe, korisnik mora osigurati da se taj ključ više neće koristiti u terminalima za statičku ili dinamičku autentifikaciju podataka nakon trenutka ukidanja.

Sledeći principi su uvaženi kod korisničkog ukidanja validnosti javnog ključa sertifikatora u terminalu:

- Terminal mora imati mogućnost utvrđivanja ispravnosti informacije o ukidanju ključa dobijenog od korisnika,
- Terminal mora imati mogućnost utvrđivanja originalnosti informacije o ukidanju ključa od legitimnog korisnika,
- Korisnik mora imati mogućnost potrvde da je zaista zatražio ukidanje specifičnog javnog ključa sertifikatora u terminalu.

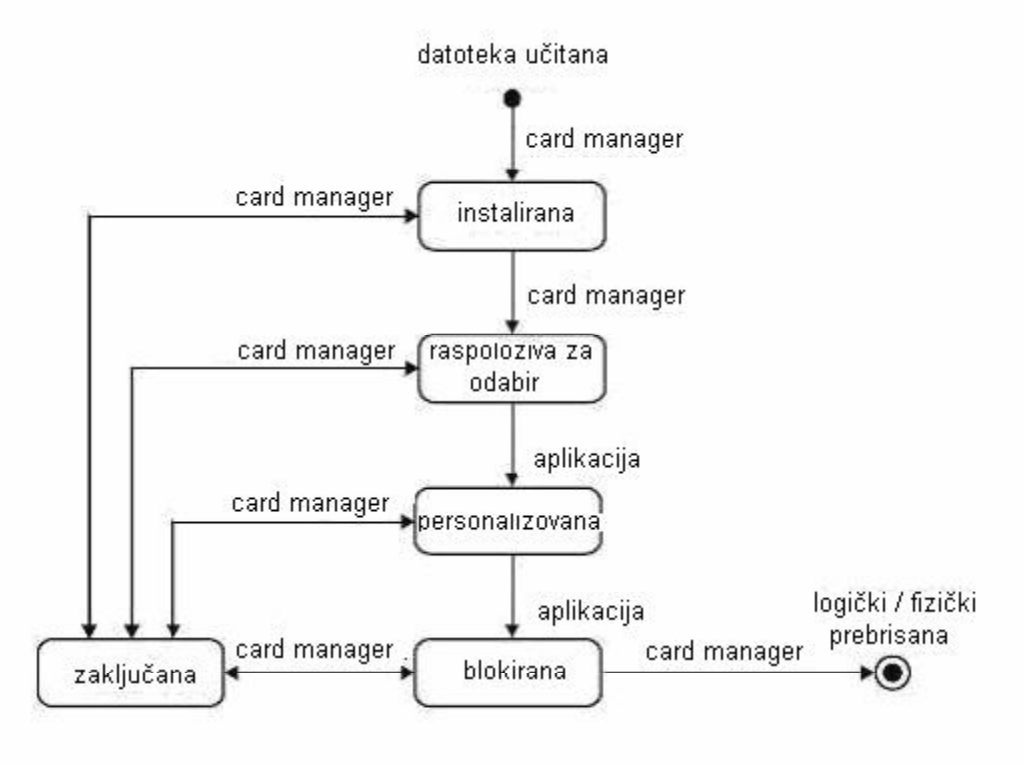
3.3. Global platform

Kao jedan od najvećih izdavača kartica *Visa International* rano se susrela sa problemom baratanja većim brojem aplikacija različitih proizvođača na višenamenskim karticama. To je dovelo do stvaranja *Visa Open Platform* specifikacije. Kasnije tu specifikaciju izdaje *Global Platform Committee* i ime specifikacije se menja u *Open Platform*. Specifikacija je napravljena za svrhu pružanja standarda kartice koji će biti nezavisan od proizvođača, tipa hardvera i biti podržan od različitih operativnih sistema kartica. Kako otvorenih tako vlasničkih. U praksi ova specifikacija je postala *de facto* standard za smeštaj i baratanje aplikacijama zasnovanim na *Javi* u *Java Card* operativnom sistemu. *Open Platform* specifikacija definiše osnovnu arhitekturu višenamenske pametne kartice. Okolina za izvođenje (engl.*runtime environment*) je temelj svih aplikacija. Ona pruža interfejs nezavisno od hardvera na kartici i prostor za čuvanje svih podataka. *Card manager* je na njoj izgrađen. To je najbitnija komponenta ove specifikacije. Može se smatrati predstavnikom izdavača kartice. On upravlja okolinom za izvođenje u odnosu na aplikacije, daje im interfejs za servise ugrađene na karticu kao i interfejse za komunikaciju sa spoljnim svetom. To npr. uključuje raspoređivanje dolazećih APDU-ova odgovarajućim aplikacijama. Registar kartice (engl.*card registry*) služi za upravljanje sigurnosnim domenima, stanjima životnog ciklusa itd. Sigurnosni domen (engl.*security domain*) predstavlja proizvođača aplikacije u pitanjima sigurnosti. On daje ključeve i šifarske servise aplikacijama koje su nezavisne od proizvođača kartice. *Open platform API* pruža aplikacijama interfejs prema manager-u kartice.



Slika 3.42. Global platform specifikacija

Postoje dve vrste aplikacija prema Open Platform specifikaciji. Nepromenljive i promenljive. Ove prve se smeštaju u memoriju kartice tokom njene proizvodnje i ostaju u fiksnoj formi. Ove druge se mogu dodati, instalirati i odstraniti nakon što je kartica proizvedena ili nakon što je ona izdata. Životni ciklus aplikacije sastoji se od više koraka. Prvi korak je instalacija aplikacije tj. njen smeštaj tako da ona može biti pokrenuta. U tom koraku ona još ne može biti odabrana od spolja. To je moguće u sledećem koraku '*raspoloživa za odabir'*. Sledeće stanje '*personalizovana'* se postiže kada se aplikaciji daju individualni podaci zavisno od korisnika kartice. Nakon toga sledi period korišćenja. Sada je aplikaciju po potrebi moguće i blokirati. To stanje je reverzibilno za razliku od zaključanog stanja koje je takođe moguće. '*Logički/fizički prebrisano'* stanje označava da je aplikacija uklonjena sa kartice. Specifikacija naredbi implementiranih ovim standardom snažno se oslanja na ISO7816-4.



Slika 3.43. Stanja aplikacije prema *Open Platform* arhitekturi

3.4. Standardizacija pametnih kartica u šifarskim sistemima

3.4.1. Mogućnosti pametnih kartica u šifarskim sistemima

Pametne kartice su sigurne računarske platforme idealne za ostvarenje sigurnosnih funkcionalnosti u raznim primenama. Mogu baratati sertifikatima šifarskim ključevima ili uopšte autentifikacijskim podacima. Takođe mogu na siguran način čuvati osetljive podatke poput:

- privatnih ključeva,
- brojeva računa i salda,
- lozinki,
- autorizacionih dozvola,

U isto vreme one predstavljaju izolovano okruženje sposobno da poverljive podatke koristi bez izlaganja spoljnom svetu štiteći ih od potencijalnih napada. To je posebno važno pri operacijama poput kreiranja digitalnih potpisa, korišćenja privatnih

ključeva i identifikacije. Nažalost upotreba kartica u ove svrhe često je ograničena. Budući da šifarski sistemi različitih pružalaca usluga mogu koristiti različite standarde zapisa digitalnih potvrda upotreba jedne kartice može biti ograničena na samo jedan sistem. To predstavlja naročit problem za krajnjeg korisnika. Cilj je omogućavanje interoperabilnosti između raznih šifarskih sistema.

3.4.2. Digitalni sertifikat

Ukoliko se želi postići funkcionalnost digitalnog potpisa susreće se sa važnim problemom. Ukoliko primalac želi proveriti digitalni potpis treba mu javni ključ pošiljaoca. Kako bi primalac bio siguran u verodostojnost tog ključa on ga dobija potpisanog od strane sertifikacionog centra. Tako potpisani javni ključ u kombinaciji sa još nekim parametrima naziva se sertifikatom. Kako bi se sertifikati mogli izmenjivati treba odrediti njihovu strukturu. Najrašireniji je opisana u standardu X.509.

3.4.2.1. Standard X.509

Ovaj opširni standard je čini osnovu mnogih primena digitalnog potpisa. Npr. **SSL** (engl.***S**ecure **S**ocket **L**ayer*).

Element	Objašnjenje
Verzija	Označava verziju X.509 standarda koja definiše elemente sertifikata. Obično je to verzija 3.
Serijski broj	Serijski broj sertifikata. Određuje ga izdavač sertifikata i mora biti jedinstven.
Identifikator algoritma potpisivanja	Označava šifarski algoritam korišćen za digitalni potpis.
Ime izdavača	Ime izdavača je takođe jedinstveno
Razdoblje validnosti	Period u kojem je sertifikat važeći
Ime učesnika	Identifikator vlasnika ključa ovog sertifikata
Javni ključ	Autentični javni ključ učesnika
Potpis	Potpis sertifikacionog centra

Tabela 3.29. Sadržaj sertifikata X.509

U ovom standardu su definisani i razni drugi opcionalni elementi sertifikata. Na primer moguće je u jednom sertifikatu imati više različitih javnih ključeva potpisanih od strane različitih sertifikacionih centara. Na taj način sertifikati mogu narasti do veličina nepogodnih za smeštaj na kartice. Tipična veličina X.509 sertifikata na pametnoj kartici je 1kB.

Deo IV

4. JAVA Card tehnologija

Usled zahteva na sve široj primeni pametnih kartica javila se potreba za bržim razvojem aplikacija kartica. Brži razvoj aplikacija kartica ne sme ugroziti temeljni razlog korišćenja pametnih kartica, sigurno procesiranje. To je za razvojne programere pametnih kartica najvažnije pitanje, pitanje sigurnosti.

Java Card tehnologija omogućuje da se programi pisani u Java programskom jeziku izvode na pametnim karticama i drugim uređajima ograničenih sredstava. Nasleđujući sigurnosne osobine Java jezika, istovremeno pruža još jedno svojstvo, nezavisnost aplikacije kartice od operativnog sistema pametne kartice. Dodatno svojstvo je i mogućnost učitavanja aplikacija na kartici u korisničkoj fazi životnog ciklusa pametne kartice. Time pametna kartica postaje aktivni deo mrežne arhitekture.

Sigurnost se mora uzeti u obzir sa tačke gledišta ukupnog, opšteg sistema. Java Card platforma je izgrađena na vrhu kartične platforme, koju sačinjava kartični hardver, izvorni operativni sistem, terminal s kojim kartica komunicira i sistem u pozadini. Ovo poglavlje ističe deo sigurnosti kartične platforme kroz sigurnost Java Card platforme.

4.1. Pregled arhitekture

Pametne kartice danas predstavljaju jednu od najmanjih računarskih platformi. Njihova konfiguracija memorija prati otprilike ove brojke: 1K RAM-a, 16K EEPROM-a, 24K ROM-a. Najveći izazov oblikovanja Java Card tehnologije bio je kako smestiti Java programski sistem na pametnu karticu, a istovremeno očuvati dovoljno prostora za aplikacije. Rešenje se pronašlo na načinu podržavanja samo podskupa osobina jezika Jave i primene podeljenog modela implementacije Java virtuelne mašine (engl. *Java Virtual Machine, **JVM***)

Java Card virtuelna mašina (engl. *Java Card Virtual Machine, **JCVM***) podeljena je na dva dela: prvi koji se izvodi izvan kartice i drugi koji se izvodi na kartici. Zadaci koji nisu ograničeni vremenom izvođenja (engl. *runtime*), kao što je učitavanje klasa, provera bajtkoda (engl. *bytecode*), rastavljanje i povezivanje, optimizacija, obrađuju se u delu virtuelne mašine koji se nalazi izvan kartice, gde resursi obično ne predstavljaju problem.

Da bi omogućila podršku za Java programski jezik, Java Card tehnologija definisala je Java Card izvršnu okolinu (engl. *Java Card Runtime Environment, **JCRE***). Ona podržava memoriju kartice, komunikacioni model, sigurnost i aplikativni izvršni model. Java Card izvršna okolina pridržava se međunarodnog standarda ISO 7816.

Jedno od najznačajnijih obeležja Java Card izvršne okoline je da omogućuje nezavisnost aplikacije na kartici od operativnog sistema pametne kartice. Izvršna okolina enkapsulira složenost i detalje operativnog sistema kartice. Aplikacije traže servise i resurse operativnog sistema preko dobro definisanog programskog interfejsa visokog nivoa.

Java Card tehnologija definiše platformu za izvršavanje aplikacija pisanih u Java programskom jeziku (aplikacije pisane za Java Card platformu nazivaju se ***appleti***). Zbog podeljene arhitekture virtuelne mašine, platforma je raspodeljena između pametne kartice i ostatka kartičnog sistema (npr. personalni računar PC) u vremenu i prostoru. Sastoji se od tri dela, svaki definisan u svojoj specifikaciji:

- Specifikacija Java Card 2.1 Virtual Machine – JCVM definiše podskup programskog jezika Jave i definiciju virtuelne mašine prikladnu za aplikacije na kartici,
- Specifikacija Java Card 2.1 Runtime Environment – JCRE precizno opisuje Java Card izvršno ponašanje, uključujući upravljanje memorijom, upravljanje appletom i druge izvršne osobine.
- Specifikacija Java Card 2.1 Java Card 2.1 Application Programming Interface – API opisuje jezgro i proširenja Java paketa i klasa potrebnih za programiranje aplikacija na kartici.

4.2. Java Card jezički podskup

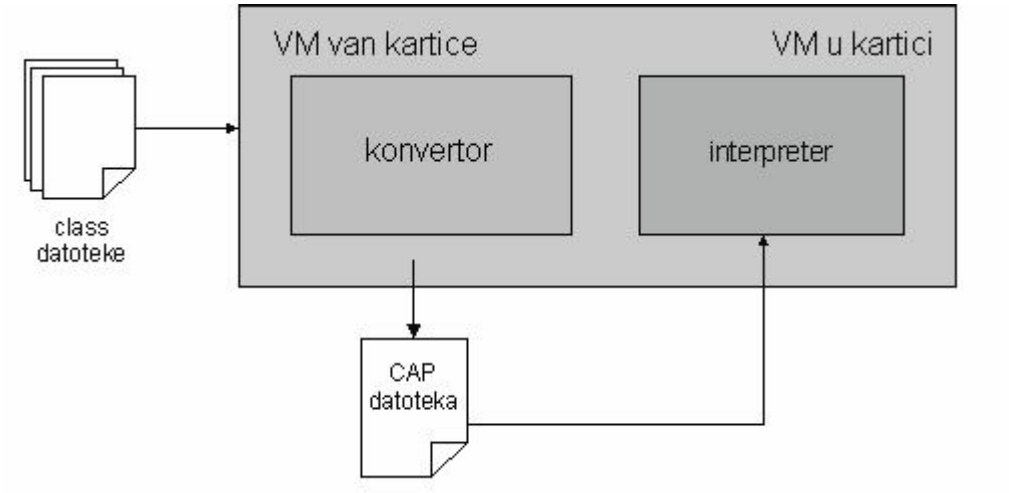
Zbog svog malog memorijskog kapaciteta, Java Card platforma podržava samo pažljivo odabrani, prilagođeni podskup osobina Java jezika. Taj podskup uključuje osobine koje su prilagođene pisanju programa za pametne kartice i druge male uređaje, istovremeno čuvajući objektno-orijentisane osobine Java programskog jezika. Tabela 4.1 ističe neka bitna podržane i nepodržane osobine Java jezika.
Ključne reči nepodržanih svojstava su takođe ispuštene iz jezika. Neke od naprednih Java kartica implementiraju mehanizam sakupljanja smeća, odnosno mehanizam brisanja objekata nasleđen iz Java platforme.

Podržane osobine Jave	Nepodržane osobine Jave
· Mali primitivni tipovi podataka: *boolean*, *byte*, *short* · Java paketi, klase, interfejsi, funkcije · Objektno-orijentisane osobine: nasleđivanje, virtuelne metode, nadjačavanje, dinamičko kreiranje objekta, opseg pristupa, pravila povezivanja · Opcionalni su ključna reč *int* i 32-bitni tip podatka	· Veliki primitivni tipovi podataka: *long, double, float* · Znakovi i stringovi · Višedimenzionalna polja · Dinamičko učitavanje klasa · Sigurnosni manager · Sakupljanje smeća i finalizacija · Serijalizacija objekata · Kloniranje objekata

Tabela 4.1. Podržane i nepodržane osobine Jave

4.3. Java Card virtuelna mašina

Osnovna razlika između Java Card virtuelne mašine (JCVM) i Java virtuelne mašine (JVM) je u tome što je JCVM implementiran u dva odvojena dela (slika 4.1). Deo kartice Java Card virtuelne mašine uključuje Java Card bajtkod interpreter. Java Card konvertor se izvršava na PC-u. Konvertor je van-kartični deo virtuelne mašine. Gledajući ih zajedno, oni implementiraju sve funkcije virtuelne mašine. Konvertor učitava i procesira *class* datoteke koje čine Java paket i proizvodi CAP (engl. *Converted APplet*, ***CAP***) datoteke. CAP datoteka se onda učitava u Java pametnu karticu i izvodi od strane interpretera. Kao dodatak prilikom procesa kreiranja CAP datoteke, konvertor generiše izvedenu datoteku (engl. *export file*). Izvedena datoteka predstavlja javne API-je konvertovanog paketa.



Slika 4.1. Java Card virtuelna mašina

Java Card tehnologija podržava samo podskup osobina jezika Jave. Sve nepodržane jezičke oblike korišćene u appletu otkriće konvertor.

4.3.1. CAP datoteke i izvedene datoteke

Java Card tehnologija uvodi dva nova formata binarne datoteke koji omogućuju platformsku nezavisnost razvoja, distribucije i izvršavanja Java Card programa. CAP datoteka sadrži izvršnu binarnu prezentaciju klasa Java paketa. Format JAR (engl. *Java Archive,* ***JAR***) datoteke se koristi kao "*kontejner ili skladište*" CAP datoteka. Format CAP datoteke optimiziran je za male memorije korišćenjem kompaktnih struktura podataka. Definiše skup bajtkod instrukcija, temeljen i optimiziran nad skupom instrukcija Java bajtkoda.

Svojstvo Java programa "*napiši jednom, izvodi svuda*" je možda i najvažnija osobina Java platforme. U Java tehnologiji centralni deo Java arhitekture je class datoteka. Ona definiše standard binarne kompatibilnosti Java platforme. Zbog raspodeljenih karakteristika Java Card sistemske arhitekture, standard binarne kompatibilnosti Java Card platforme definiše CAP datoteka. Prema tome znači, da je format CAP datoteke oblik u kome se programi učitavaju na Java pametnu karticu. Otuda i naziv CAP datoteke - *converted applet file*.

Izvedene datoteke se ne učitavaju na pametnu karticu, prema tome njih direktno i ne koristi interpreter. Njih proizvodi, ali i koristi kompejler prilikom procesa provere i povezivanja, a mogu se predočiti kao zaglavlja datoteka u C jeziku. Jedna izvedena datoteka sadrži javne API informacije za jedan paket klasa. Ona definiše ime klase i njen opseg pristupa (engl. *access scope*), definiše metode i polja klase i njihov opseg pristupa. Izvedena datoteka takođe sadrži informacije o povezivanju koje se koriste na kartici prilikom određivanja međupaketskih referenci.

Izvedena datoteka ne sadrži nikakvu implementaciju, odnosno ne sadrži bajtkod. Zbog toga izvedenu datoteku razvojni programer appleta slobodno distribuira do potencijalnih korisnika appleta bez otkrivanja detalja unutrašnjeg izvođenja.

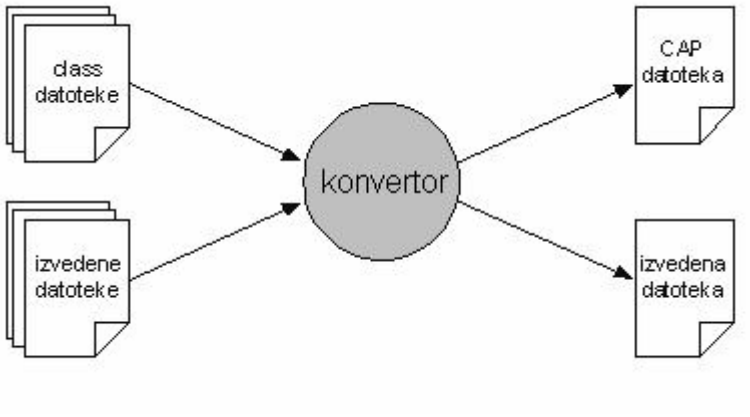
4.3.2. Java Card konvertor

Za razliku od Java virtuelne mašine koja procesira jednu klasu, pretvaračka jedinica kompajlera je paket. Prevodilac (engl. *compiler*) iz izvornog koda produkuje class datoteke. Zatim konvertor pretprocesira sve class datoteke koje čine Java paket i konvertuje ih u CAP datoteku.

Za vreme te konverzije, konvertor izvodi zadatke koje Java virtuelna mašina izvodi za vreme učitavanja klasa (engl. *class-loading time*):

- Potvrđuje da su učitane Java klase ispravno formirane,
- Proverava odstupanja od Java Card jezika,
- Izvodi inicijalizaciju statičkih varijabli (promenljivih),
- Razrešava simboličke reference na klase, metode i polja u jedan zbijeniji oblik kojim se može efikasnije rukovati na kartici,
- Optimizuje bajtkod uzimajući kao prednost informaciju dobijenu za vreme učitavanja klasa i procesa povezivanja,
- Alocira (razmešta) memoriju i kreira strukture podataka virtuelne mašine koje predstavljaju klase.

Konvertor uzima kao ulaz ne samo class datoteke koje će se konvertovati već takođe jednu ili više izvedenih datoteka. Osim produkcije CAP datoteka, konvertor generiše i izvedenu datoteku za dotični konvertovani paket. Slika 4.2 pokazuje kako se konvertuje paket. Konvertor učitava sve klase u Java paket. Ako paket preuzima (engl. *import*) klase iz drugih paketa, konvertor takođe učitava izvedene datoteke dotičnih paketa. Izlaz iz konvertora je CAP datoteka i izvedena datoteka konvertovanog paketa.



Slika 4.2. Konverzija paketa

4.3.3. Java Card interpreter

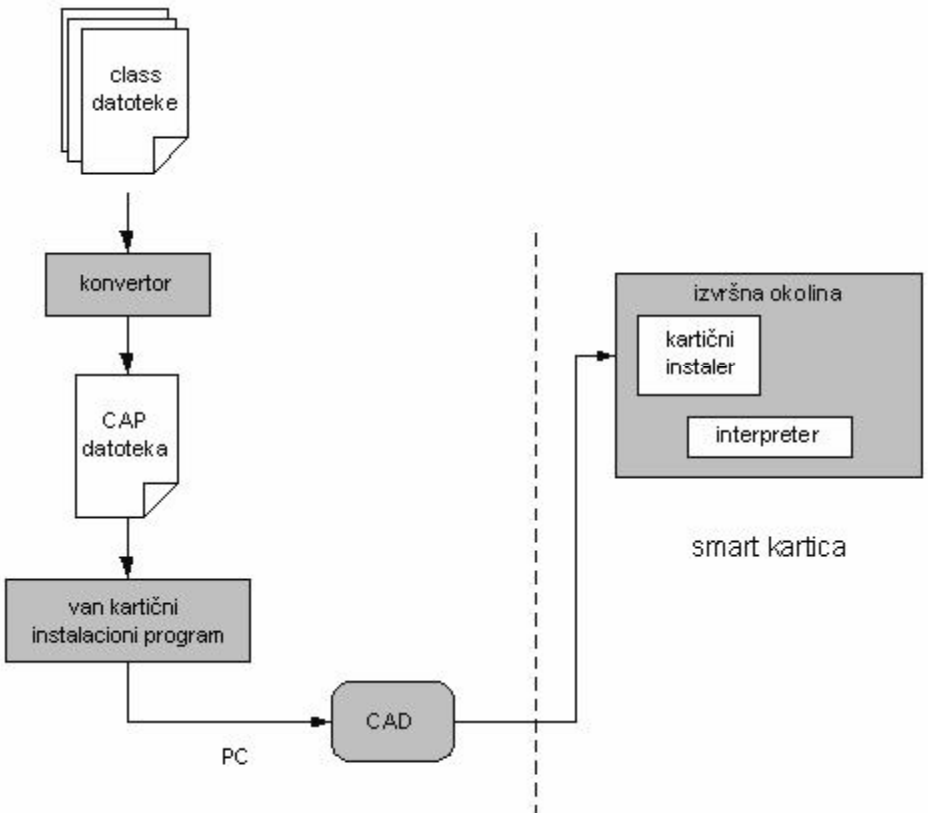
Java Card interpreter predstavlja izvršnu podršku Java jezičkog modela i time omogućava nezavisnost appleta od platforme. Interpreter izvodi sledeće zadatke:

- izvršava bajtkod instrukcije, time i applete,
- kontroliše alokaciju memorije i kreiranje objekata,
- igra ključnu ulogu u sigurnosti prilikom izvršavanja aplikacije na kartici.

Do sada je pod pojmom Java Card virtuelne mašine uključivan interpreter i konvertor. Međutim Java Card virtuelna mašina se definiše kao deo kartice virtuelne mašine - interpreter. Termini Java Card interpreter i Java Card virtuelna mašina se često koriste kao sinonimi, ali kada se upoređuju Java Card platforma sa Java platformom treba biti svestan da se funkcije izvršavanja Java class datoteka postižu zajednički i kroz konvertor i kroz interpreter.

4.4. Java Card instaler i instalacija programa izvan kartice

Java card interpreter ne učitava CAP datoteke. On samo izvršava kod pronađen u CAP datoteci. U Java Card tehnologiji su mehanizmi preuzimanja i instaliranja datoteka ustanovljeni u jedinici koja se zove instaler (engl. *installer*).



Slika 4.3. Java Card instaler i van kartični instalacioni program

Java Card instaler se nalazi unutar kartice. On sarađuje sa van kartičnim instalacionim programom. Van kartični instalacioni program šalje kartici CAP datoteku

preko uređaja za prihvatanje kartice, CAD uređaja. Unutar CAP datoteke nalazi se binarni kod koji instaler upisuje u memoriju pametne kartice, povezuje ga sa ostalim klasama koje se već nalaze na kartici, kreira i inicijalizuje sve strukture podataka koje interno koristi Java Card izvršna okolina. Na slici 4.3 prikazani su instaler, instalacioni program i njihov odnos sa ostatkom Java Card platforme.

Podela funkcionalnosti između interpretera i instalera CAP datoteka čini interpreter malim i omogućava fleksibilnost implementacije instalera.

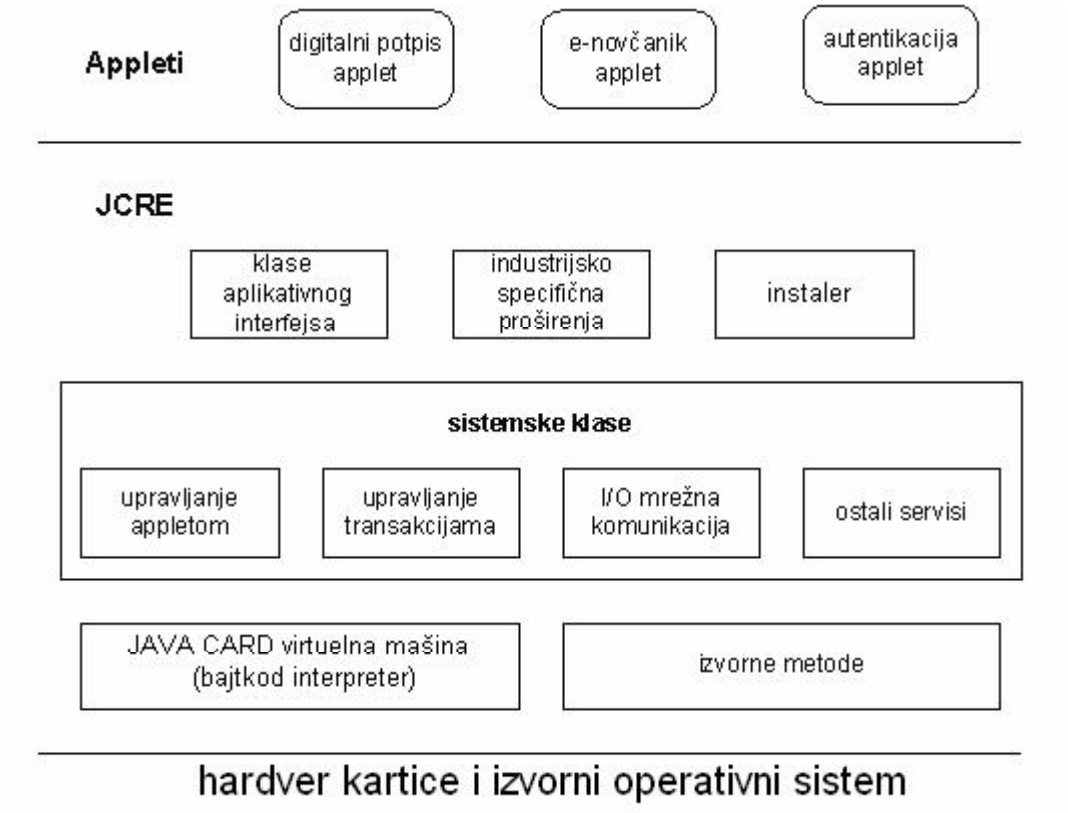
4.5. Java Card izvršna okolina

Java Card izvršnu okolinu sačinjavaju Java Card sistemske komponente unutar pametne kartice. JCRE (engl. *Java Card Runtime Environment,* ***JCRE***) je odgovoran za upravljanje resursima kartice, mrežnom komunikacijom, izvršavanje appleta, i sigurnost programa na kartici. Ali, njena osnovna uloga je uloga operativnog sistema pametne kartice.

Kao što je prikazano na slici 4.4, JCRE se nalazi iznad hardvera pametne kartice i izvornog operativnog sistema. JCRE se sastoji od Java Card virtuelne mašine (bajtkod interpreter), Java Card osnovnih aplikativnih klasa (engl. *Java Card application framework classes*), industrijsko-specifičnih proširenja i JCRE sistemskih klasa. JCRE elegantno odvaja applete od vlasničkih tehnologija izdavača pametnih kartica i osigurava standardizovani sistem i API interfejs za applete. Kao rezultat, applete je jednostavnije i lakše pisati, i prenosivi su na različite arhitekture kartica.

Donji sloj JCRE čine Java Card virtuelna mašina (JCVM) i izvorne metode. JCVM izvršava bajtkod, kontroliše alokaciju memorije, upravlja objektima i nameće sigurnosne mehanizme prilikom vremena izvođenja. Izvorne metode predstavljaju podršku JCVM-u i sledećem sloju sistemskih klasa. One su odgovorne za rukovanje nisko-nivovskim komunikacionim protokolima, upravljaju memorijom, osiguravaju kriptografsku podršku, itd.

Sistemske klase su izvršni alat Java Card izvršne okoline. One su analogija jezgru operativnog sistema. Sistemskim klasama upravlja se transakcijama, one su u službi upravljanja komunikacijom između aplikacija domaćina (*host application, host applications are the applications running at the terminal side with which applets communicate*) i Java Card appleta, kontrole kreiranja appleta, selekcije i deselekcije appleta. Da izvrše svoje zadatke, sistemske klase po pravilu pozivaju izvorne metode.



Slika 4.4. JCRE arhitektura

Java Card aplikativni interfejs (engl. *Java Card Application Framework*) definiše interfejs za programiranje aplikacija. Interfejs se sastoji od osnovnih i dodatnih API paketa. Klase API paketa su prilagođene razvoju appleta na kartici. Glavna prednost ovog interfejsa je da čini relativno jednostavnim proces kreiranja appleta. Razvojni programeri mogu usredsrediti većinu svojih napora na detalje vezane za applet, nego na detalje vezane za infrakstuturu operativnog sistema kartice. Kroz API klase appleti pristupaju JCRE servisima.

Specifične industrije pretiču se dodatnim bibliotekama što im omogućava nove servise i poboljšava sigurnosni i sistemski model. Kao primer može se navesti biblioteka specifikacije *Open Platform* koja proširuje JCRE servise da zadovolje specifične sigurnosne potrebe finansijske industrije. Između ostalog nameće kontrolu kartica od strane izdavača i određuje standardni skup naredbi za personalizaciju kartica.

Instaler omogućuje sigurno preuzimanje sofvera i appleta na karticu nakon što je kartica proizvedena i izdata vlasniku (nosiocu) kartice. Instaler sarađuje sa van kartičnim instalacionim programom. Zajedno, ispunjavaju zadatak učitavanja binarnog sadržaja CAP datoteka. Sam instaler je opcijska JCRE komponenta. Bez njega, ceo kartični softver, uključujući applete, mora biti upisan u kartičnu memoriju za vreme procesa proizvodnje kartice.

Java Card appleti su korisničke aplikacije na Java Card platformi. Oni su naravno pisani u podskupu Java programskog jezika i kontrolisani i upravljani od strane JCRE-a. Appleti mogu učitati Java pametnu karticu i u korisničkoj fazi životnog ciklusa kartice.

4.5.1. Vreme života Java Card izvršne okoline

Java virtuelna mašina izvršava se na PC-u ili radnoj stanici kao proces operativnog sistema. Podaci i objekti se kreiraju u RAM-u. Kada se ugasi OS proces, Java aplikacije i njihovi objekti su automatski uništeni.

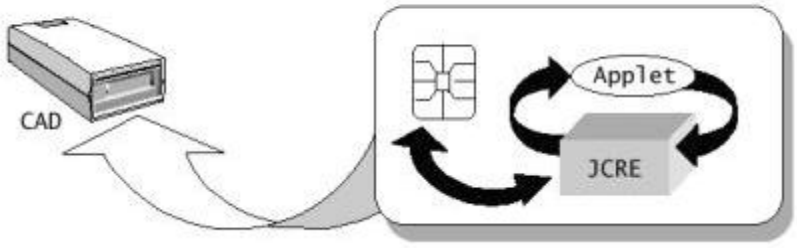
Na Java pametnoj kartici, Java virtuelna mašina se izvršava unutar Java Card izvršne okoline. JCRE se inicijalizuje za vreme inicijalizacije kartice. Inicijalizacija JCRE se izvodi samo jednom u životnom ciklusu kartice. Za vreme tog procesa JCRE inicijalizuje virtuelnu mašinu i kreira objekte koji omogućuju JCRE servise i upravljanje appletima. Nakon učitavanja appleta, JCRE kreira njihove nadležnosti, a appleti kreiraju svoje objekte za čuvanje podataka.

Većina informacija na kartici mora biti očuvana čak i kad se kartica ukloni iz napajanja. U tu svrhu se koristi ispisna (perzistentna) memorijska tehnologija (kao što je EEPROM). U perzistentnoj memoriji se kreiraju podaci i objekti. Vreme života JCRE je ekvivalentno kompletnom vremenu života kartice. Kad se ukloni napajanje zaustavlja se virtuelna mašina, ali stanje JCRE-a i objekti kreirani na kartici ostaju sačuvani.

Sledeći put kad se na karticu dovede napajanje, JCRE ponovno pokreće izvršavanje virtuelne mašine učitavanjem podatka iz perzistentne memorije. Treba napomenuti da JCRE ne nastavlja operacije virtuelne mašine na mestu na kojem je virtuelna mašina stala kad je kartica ostala bez napajanja. Virtuelna mašina se ponovno pokreće i izvršava od početka glavne petlje. Prekid sesije JCRE-a se razlikuje se od JCRE inicijalizacije u tome što appleti i objekti kreirani na kartici ostaju sačuvani. Za vreme nove sesije, ako transakcija prethodno nije završila, JCRE izvodi potrebno čišćenje da dovede JCRE u konzistentno stanje.

4.5.2. Funkcionisanje JCRE za vreme CAD sesije

Razdoblje između vremena kad je kartica stavljena u CAD uređaj i priključena na napajanje, i vremena kad se kartica ukloni iz CAD-a, naziva se CAD sesija. Za vreme CAD sesije JCRE radi kao obična pametna kartica – podržava APDU I/O komunikaciju sa aplikacijom terminala (slika 4.5). APDU–ovi su paketi podataka koje izmenjuju appleti sa aplikacijom terminala (engl. *host application*). Svaki APDU sadrži naredbu terminala ili odgovor appleta.



Slika 4.5. APDU I/O komunikacija

Ponovnim uspostavljanjem sesije (*JCRE reset*) JCRE ulazi u petlju, i čeka APDU naredbu od terminala. Terminal šalje APDU naredbe Java Card platformi koristeći interfejs serijske komunikacije ulaznog/izlaznog kontakta na kartici.

Kad stigne naredba, JCRE ili odabire applet za izvođenje, na što ju upućuje pristigla naredba, ili prosleđuje naredbu trenutno aktivnom, selektovanom appletu. Selektovani applet tada preuzima kontrolu i procesira APDU naredbu. Kad završi šalje odgovor terminalu i predaje kontrolu JCRE-u. Taj proces se ponavlja prilikom pristizanja sledeće naredbe.

5.5.3. Svojstva Java Card izvršne okoline

Osim što podržava izvršni model Java jezika, JCRE podržava tri dodatne osobine vremena izvođenja:

- *Perzistentni i prelazni objekti* – Standardno, Java Card objekti su perzistentni i kreirani su u perzistentnoj memoriji. Prostor i podaci takvih objekata traju, odnosno premošćuju više CAD sesija. Zbog sigurnosnih i razloga vezanih za učinak, appleti mogu kreirati objekte i u RAM memoriji. Takvi objekti se nazivaju transparentni objekti. Transparentni objekti sadrže privremene podatke koji ne perzistiraju kroz više CAD sesija,
- *Atomske operacije i transakcije* – Java Card virtuelna mašina osigurava da je svaka operacija pisanja u pojedinačno polje objekta ili klase atomska. Ažurirano polje dobija novu vrednost ili mu se vraća prethodna vrednost. Dodatno, JCRE pruža transakcijske API-je. Applet može uključiti nekoliko operacija pisanja u

transakciju. Tada su sva ažuriranja u transakciji završena ili (ako se dogodi greška usred transakcije) ništa ne sprovodi,

- *Firewall appleta i mehanizmi deljenja* – Firewall appleta izoluje applete. Svaki applet se izvodi unutar označenog prostora. Postojanje i operacije jednog appleta nemaju uticaja na druge applete na kartici. Firewall appleta nameće Java Card virtuelna mašina dok izvršava bajtkod. U situacijama kad appleti žele deliti podatke ili pristupati JCRE servisima, virtuelna mašina dopušta takve funkcije kroz sigurne mehanizme deljenja.

4.6. Java Card API

Java Card API se sastoji se od skupa prilagođenih klasa za programiranje pametnih aplikacija na kartici prema ISO 7816 modelu. API-ji sadrže tri osnovna paketa i jedan dodatni. Tri osnovna paketa su

1. *java.lang*,
2. *javacard.framework,*
3. *javacard.security,.*

(4.) *java.cardx.crypto (*dodatni paket*).*

Mnoštvo Java platformskih klasa nije podržano u Java card API-jima. Na primer nisu podržane Java platformske klase za grafičko korisnički interfejs (engl. *Graphical User Interface, **GUI***), klase za mrežni I/O, klase za I/O nad datotekama. Razlog je u tome što pametne kartice nemaju podlogu, koriste drugačiji mrežni protokol i drugačiju strukturu file sistema. Takođe, da se udovolji strogim memorijskim zahtevima, nije podržano mnoštvo Javinih uslužnih klasa.

Klase u Java Card API-jima su celovite i zbijene. Osim što sadrže klase preuzete od Java platforme, takođe sadrže i klase specijalno stvorene za pametne kartice kompatibilne sa standardom ISO 7816.

4.6.1. java.lang paket

Java Card *java.lang* paket je podskup odgovarajućeg *java.lang* paketa Java platforme. Podržane klase su *Object*, *Throwable* i neke klase vezane uz funkcije mašine stanja. Za podržane klase mnoštvo Java metoda nije dostupno, npr. klasa Java Card *Object* definiše samo standardni konstruktor i *equals()* metodu.

Paket *java.lang* pruža osnovnu podršku Java jeziku. Klasa *Object* predstavlja koren u hijerarhiji Java Card klasa, a klasa *Throwable* zajedničkog pretka svih kartičnih funkcija. Podržane klase funkcija osiguravaju nepromenjenu semantiku prilikom pojave greške. Kao primer može poslužiti funkcija *NullPointerException*, javlja se zbog prihvatanja *null* reference, koju izbacuje i Java mašina stanja i Java Card mašina stanja kad prihvate *null* referencu.

4.6.2. javacard.framework paket

Paket *javacard.framework* je središnji paket koji sa svojim klasama i interfejsima omogućuje osnovnu funkcionalnost Java Card appleta. Što je još važnije, on definiše bazičnu klasu *Applet*, koja za vreme života appleta predstavlja interfejs za izvođenje appleta i interfejs za interakciju appleta sa JCRE-om. Odnos te klase i JCRE-a je sličan odnosu Java Applet klase i preglednika (engl. *Viewer*) na računaru. Korisnička applet klasa mora naslediti bazičnu *Applet* klasu, a da implementira funkcionalnost appleta mora nadjačati (engl. *override*) metode u *Applet* klasi.

Druga važna klasa u *javacard.framework* paketu je *APDU* klasa. APDU paketi se prenose transportnim protokolom. Postoje dva standardizovana transportna protokola, protokol T=0 i protokol T=1. *APDU* klasa je tako osmišljena da bude nezavisna od korišćenog transportnog protokola. Drugim rečima ona je pažljivo dizajnirana tako da sakrije zamršenosti i razlike T=0 i T=1 protokola. Programeri mogu dosta lakše rukovati APDU naredbama koristeći metode koje pruža *APDU* klasa. Appleti rade ispravno nezavisno od transportnog protokola kojeg podržava platforma.

Nije podržana klasa Java platforme *java.lang.System*. Umesto nje, kao interfejs sistema, Java Card platforma sadrži klasu *javacard.framework.JCSystem*. Ona uključuje skup metoda za kontrolu izvođenja appleta, za upravljanje resursima, za upravljanje transakcijama i metode za deljenje objekata između appleta na Java Card platformi.

Druge klase podržane u *javacard.framework* paketu su PIN, funkcije i uslužne klase. PIN se često koristi kod pametnih kartica kao lozinka za autentifikaciju vlasnika kartice.

4.6.3. javacard.security paket

Paket *javacard.security* predstavlja interfejs za šifarske funkcije podržane na Java Card platformi. On je baziran na paketu *java.security*.

Paket *javacard.security* definiše klasu za proizvodnju ključeva *keyBuilder* i različite interfejse koji predstavljaju šifarske ključeve korišćene u simetričnim (DES) i asimetričnim (DSA i RSA) algoritmima. Kao dodatak podržava i bazične apstraktne klase *RandomdData*, *Signature i MessageDigest*, koje se koriste prilikom generisanja slučajnih brojeva, izračunavanja sažetka poruke (engl. *message digest*) i digitalnog potpisivanja.

Algoritam	Uloga
DES, DSA, RSA	implementacija različitih šifarskih ključeva
DES/64, TripleDES/128/192, DSA/512/768/1024, RSA/512/768/1024/2048	mogućnost stvaranja (polje `KeyBuilder`) ključeva
MD5, RIPEMD-160, SHA	izračunavanje šifarskog sažetka poruke (`MessageDigest`) pomoću nekoliko algoritama
DES/MAC4, DES/MAC8, DSA/SHA, RSA/MD5, RSA/RIPEMD160, RSA/SHA	digitalno potpisivanje (`Signature`) pomoću celog niza kombinacija algoritama

Tabela 4.2. Prikaz algoritama i uloga koju izvršavaju

Osnovnu sigurnost čine sledeći paketi:

- `java.security`,
- `java.security.cert`,
- `java.security.interfaces`,
- `java.security.spec`.

4.6.3.1. Paket `java.security`

Paket `java.security` pruža osnovni skup klasa i interfejs potreban za implementaciju sigurnosti.

Uključuje:

- klase koji implementiraju lako ostvarivu, sitno zrnastu arhitekturu kontrole pristupa,
- podršku za generisanje i spremanje parova asimetričnih šifarskih ključeva kao i niz šifarskih operacija koji nemaju ograničenja izlaza kao što su generisanje šifarskih sažetaka i digitalnih potpisa,
- klase koje podržavaju potpisane/čuvane objekte kao i generisanje šifarski sigurnih pseudo slučajnih brojeva.

Veliki broj klasa (pogotovo onih šifarskih i za generisanje pseudo slučajnih brojeva) predstavljaju samo programski interfejs bez stvarne implementacije pojedinih algoritama. Implementacija se ostavlja nekoj trećoj strani. Zbog toga se klasama koje koriste određene algoritme predaju imena tih algoritama kao znakovne konstante. Java 2 SDK Security API zahteva i koristi skup standardnih imena za algoritme, digitalne sertifikate i

prezentacije ključeva i digitalnih sertifikata, ali to ne sprečava neku treću stranu da implementira neki novi algoritam i definiše vlastito ime za njega. Na taj način omogućena je jednostavna nadogradnja sigurnosnih mehanizama Java platforme.

Operacija	Algoritmi
Generisanje šifarskih sažetaka	**MD2, MD5, SHA-1, SHA-256, SHA-384, SHA-512**
Generisanje para asimetričnih ključeva	**RSA**, **DSA**,**Diffie-Hellman**
Generisanje digitalnih potpisa	**MD2withRSA**,**MD5withRSA**, **SHA1withDSA**, **SHA1withRSA**
Generisanje šifarskih pseudo slučajnih brojeva	**SHA1PRNG**
Digitalni sertifikati	**X.509**
Prezentacija ključeva	**JKS**, **PKCS12**

Tabela 4.3. Skup operacija koji definiše Java 2 SDK Security API.

Sledeće tabele prikazuju neke značajne interfejse, klase i funkcije paketa `java.security`.

Interfejs	
Ime	**Opis**
Key	Osnovni interfejs koji predstavlja ključ i definiše zajedničku funkcionalnost svih vrsta ključeva.
PrivateKey	Osnovni interfejs koji predstavlja privatni ključ korišćen u asimetričnoj kriptografiji.
PublicKey	Osnovni interfejs koji predstavlja javni ključ korišćen u asimetričnoj kriptografiji.

Tabela 4.4. Značajni interfejsi `java.security` paketa

Funkcije	
Ime	**Opis**
`DigestException`	nastaje u slučaju greške prilikom generisanja šifarskih sažetaka.
`GeneralSecurityException`	za sve sigurnosne probleme.
`InvalidAlgorithmParameterException`	ukazuje na neispravne ili neadekvatne parametre algoritama.
`InvalidKeyException`	ukazuje na neispravni ključ (neispravan zapisni format, dužina, neinicijalizacija,...)
`KeyException`	osnovna operacija vezana uz ključeve.
`KeyStoreException`	osnovna operacija vezana uz prezentaciju ključeva.
`NoSuchAlgorithmException`	ukazuje na to da dotični algoritam nije dostupan.
`NoSuchProviderException`	ukazuje da dati provajder Java šifarskih proširenja nije dostupan.
`SignatureException`	osnovna operacija vezana za digitalno potpisivanje.

Tabela 4.5. Značajne funkcije `java.security` paketa

4.6.3.2. Paket `java.security.cert`

Pruža klase i interfejse potrebne za korišćenje digitalnih sertifikata (engl.*digital certificate*), lista opozvanih sertifikata (engl. *certificate revocation lists – CRL*), sertifikacionih puteva (engl. *certificate path, certificate chain*) kao i odlaganje digitalnih sertifikata i CRL-ova. Sadrži podršku za treću tačku X.509 digitalnih sertifikata i drugu tačku X.509 CRL-a. Java 2 SDK Security API ne podržava generisanje digitalnih sertifikata već samo čitanje i spremanje već generisanih. Uz standardnu instalaciju Java 2 (RE, SDK, JDK,...) dobije se i konzolni program `keytool` koji omogućuje stvaranje samo-potpisanih digitalnih sertifikata (digitalni sertifikat koji sadrži korisnikov javni ključ potpisan sa korisnikovim privatnim ključem), njihovo spremanje u datoteku koja predstavlja spremište digitalnih sertifikata (`KeyStore`), generisanje zahteva za potpisivanje digitalnog sertifikata (engl. *Certificate Signing Request – CSR*), kao i import digitalnog sertifikata od nekog sertifikacijskog centra (engl. *Certificate Authority - CA*). Java 2 SDK Security API ima ugrađenu podršku za dve vrste spremišta digitalnih sertifikata i CRL-ova: "LDAP" i "Collection".

Sledeće tabele prikazuju značajne interfejse, klase i funkcije paketa java.security.cert.

Interfejs	
Ime	**Opis**
CertSelector	Predstavlja skup zahteva prema kojima se odabiraju digitalni sertifikati. Klase koje implementiraju ovaj interfejs često se koriste prilikom dohvatanja digitalnih sertifikata iz spremišta digitalnih sertifikata (*CertStore*).
CertStoreParameters	Predstavlja osnovni interfejs za definisanje parametara pomoću kojih se stvaraju spremišta digitalnih sertifikata.
CRLSelector	Predstavlja skup zahteva prema kojima se odabiraju CRL-ovi. Klase koje implementiraju ovaj interfejs često se koriste prilikom dohvatanja CRL-ova iz spremišta digitalnih sertifikata (*CertStore*).
X509Extension	Predstavlja definiciju X.509 dodatka za treću tačku X.509 digitalnog sertifikata i drugu tačku CRL-a.

Tabela 4.6. Interfejs `java.security.cert` paketa

Klase	
Ime	**Opis**
`Certificate`	Apstraktna klasa koja predstavlja digitalni sertifikat.
`Certificate.CertificateRep`	Alternativna klasa za opis digitalnog sertifikata pogodna za serijalizaciju.
`CertificateFactory`	Klasa koja omogućava stvaranje digitalnog sertifikata na temelju nekog zapisnog formata, sertifikacionog puta i CRL-a.
`CertStore`	Spremište digitalnih sertifikata i CRL-ova.
`CollectionCertStoreParameters`	Predstavlja parametre za Collection tip spremišta digitalnih sertifikata i CRL-ova.
`CRL`	Apstraktna klasa koja predstavlja listu opozvanih digitalnih sertifikata.
`LDAPCertStoreParameters`	Predstavlja parametre za LDAP tip spremišta digitalnih sertifikata i CRL-ova.
`X509Certificate`	Apstraktna klasa koja predstavlja X.509 digitalni sertifikat.
`X509CertSelector`	Predstavlja skup zahteva prema kojima se vrši odabir X.509 digitalnih sertifikata.
`X509CRL`	Apstraktna klasa koji predstavlja X.509 CRL.
`X509CRLEntry`	Apstraktna klasa koja predstavlja opozvane X.509 digitalne sertifikate.
`X509CRLSelector`	Predstavlja skup zahteva prema kojima se vrši odabir X.509 CRL-ova.

Tabela 4.7. Klase `java.security.cert` paketa

Funkcije	
Ime	**Opis**
`CertificateEncodingException`	Funkcija koja ukazuje da digitalni sertifikat nije u ispravnom zapisnom formatu.
`CertificateException`	Opšta funkcija vezana za digitalne sertifikate.
`CertificateExpiredException`	Funkcija koja ukazuje da je digitalnom sertifikatu istekao period validnosti.
`CertificateNotYetValidException`	Funkcija koja ukazuje da digitalni sertifikat još nije postao validan.
`CertificateParsingException`	Funkcija koja ukazuje da digitalni sertifikat nije u validnom DER zapisnom formatu ili sadrži ne podržane DER atribute.
`CertStoreException`	Opšta funkcija vezana za spremište digitalnih sertifikata.
`CRLException`	Opšta funkcija vezana uz CRL.

Tabela 4.8. Funkcije `java.security.cert` paketa

4.6.3.3. Paket `java.security.interfaces`

Pruža interfejs za generisanje RSA (engl. *Rivest, Shamir i Adleman*) ključeva definisanih prema PKCS#1 i DSA (engl. *Digital Signature Algorithm*) ključeva prema FIPS-186 (engl. *Federal Information Processing Standards*).

4.6.3.4. Paket `java.security.spec`

Pruža klase i interfejse za specifikaciju ključeva i parametara različitih algoritama. Specifikacija ključeva je transparentna reprezentacija materijala koji predstavlja ključ. Ključ može biti definisan zavisno od algoritma ili nezavisno od algoritma u nekom zapisnom formatu npr. ASN.1 (engl. *Abstract Syntax Notation One*). Ovaj paket sadrži specifikacije DSA i RSA javnih i privatnih ključeva, PKCS#8 privatnih ključeva u DER (engl. *Distinguished Encoding Rules*) zapisnom formatu i X.509 javne i privatne ključeve u DER zapisnom formatu. Specifikacija parametara algoritama je transparentna reprezentacija skupa parametara korišćenih u nekom algoritmu. Ovaj paket sadrži specifikaciju parametara DSA algoritama.

4.6.4. javacardx.crypto paket

Paket *javacardx.crypto* je dodatni paket. Sadrži šifarske klase i interfejse koje podležu izvoznim zahtevima SAD-a. Definiše bazičnu apstraktnu klasu *Chipher* koja je podrška funkcijama enkripcije i dekripcije.

Paketi *javacard.security* i *javacardx.crypto* definišu API interfejse koje pozivaju appleti kad potražuju šifarske servise. Međutim, ti paketi ne predstavljaju implementaciju. Izdavač kartica, odnosno proizvođač kartica koji vrši implementaciju JCRE-a dužan je dopuniti skup klasa koje implementiraju interfejse za ključeve, a nasleđuju apstraktne klase *RandomData*, *MessageDigest*, *Signature* i *Chiper*. Obično na pametnoj kartici postoji i odvojeni koprocesor koji izvodi zahtevnije šifarske funkcije.

`Cipher` pruža metode za šifrovanje i dešifrovanje poruka pomoću niza algoritama u različitim režimima rada sa više načina dopune (tabela 4.9).

Algoritam	Režimi rada algoritama
DES/Triple DES	u **CBC** (engl. *Cipher Block Chaining*) načinu rada sa dopunom prema prvoj metodi definisanoj sa ISO 9797
DES/Triple DES	u **CBC** načinu rada sa dopunom prema drugoj metodi definisanoj sa ISO 9797 (ISO 7816-4, EMV'96)
DES/Triple DES	u **CBC** načinu rada bez implicitne dopune
DES/Triple DES	u **CBC** načinu rada sa nadopunom prema PKCS#5 (engl. *Password-Based Cryptography Standard*)
DES/Triple DES	u **ECB** (engl. *Electronic Codebook*) načinu rada sa nadopunom prema prvoj metodi definisanoj sa ISO 9797
DES/Triple DES	u **ECB** načinu rada sa dopunom prema drugoj metodi definisanoj sa ISO 9797 (ISO 7816-4, EMV'96)
DES/Triple DES	u **ECB** načinu rada bez implicitne dopune
DES/Triple DES	u **ECB** načinu rada sa dopunom prema PKCS#5
RSA	sa dopunom prema ISO 14888
RSA	sa dopunom prema ISO 9796 (EMV'96)
RSA	bez implicitne dopune
RSA	sa dopunom prema PKCS#1 (v1.5) (engl. *RSA Cryptography Standard*)

Tabela 4.9. Niz algoritama u različitim režimima rada sa više načina dopune

4.7. Sertifikacioni putevi

Interfejs, klase i funkcije za rad sa sertifikacionim putevima su prikazani u sledećim tabelama.

Interfejs	
Ime	**Opis**
CertPathBuilderResult	Predstavlja specifikaciju rezultata stvaranja sertifikacionog puta.
CertPathParameters	Predstavlja specifikaciju parametara algoritmu za stvaranje sertifikacionog puta.
CertPathValidatorResult	Predstavlja specifikaciju rezultata provere sertifikacionog puta.
PolicyNode	Predstavlja nepromenjiv (engl. *immutable*) validni čvor stabla u politici PKIX algoritma za proveru sertifikacionih puteva.

Tabela 4.10. Interfejs za rad sa sertifikacionim putevima

Klase	
Ime	**Opis**
`CertPath`	Nepromenjiv niz digitalnih sertifikata (sertifikacioni put).
`CertPath.CertPathRep`	Alternativna klasa za opis sertifikacionog puta pogodnog za serijalizaciju.
`CertPathBuilder`	Klasa za izgradnju sertifikacionih puteva.
`CertPathBuilderSpi`	SPI (engl. *Service Provider Interface*) za klasu `CertPathBuilder`.
`CertPathValidator`	Klasa za proveru sertifikacionog puta.
`CertPathValidatorSpi`	SPI za klasu `CertPathValidator`.
`PKIXBuilderParameters`	Parametri korišćeni u PKIX algoritmu izgradnje sertifikacionog puta.
`PKIXCertPathBuilderResult`	Ova klasa predstavlja uspešan rezultat PKIX algoritma izgradnje sertifikacionog puta.
`PKIXCertPathChecker`	Apstraktna klasa koja sprovodi jednu ili više provera X509 digitalnog sertifikata.
`PKIXCertPathValidatorResult`	Ova klasa predstavlja uspešan rezultat PKIX algoritma provere sertifikacionog puta.
`PKIXParameters`	Parametri korišćeni u PKIX algoritmu za proveru sertifikacionog puta.

Tabela 4.11. Klase za rad sa sertifikacionim putevima

Funkcije	
Ime	**Opis**
`CertPathBuilderException`	Funkcija koja ukazuje na jedan od mogućih problema nastalih prilikom izgradnje sertifikacionog puta.
`CertPathValidatorException`	Funkcija koja ukazuje na jedan od mogućih problema nastalih prilikom provere sertifikacionog puta.

Tabela 4.12. Funkcije za rad sa sertifikacionim putevima

4.8. Java™ Secure Socket Extension

Java™ Secure Socket Extension predstavlja implementaciju **SSL** (engl. *Secure Sockets Layer*) i **TLS** (engl.. *Transport Layer Security*) protokola čime omogućuje sigurnu Internet komunikaciju. Omogućuje šifrovanje podataka, autentifikaciju servera, čuvanje i proveru integriteta poruka i neobveznu autentifikaciju klijenta. Koristeći **JSSE** programer može pružiti sigurnu komunikaciju, preko **TCP** protokola, između klijenta i servera nezavisno od aplikativnog protokola (**HTTP, Telnet, FTP**,...) koji se koristi. **JSSE** je postao sastavni deo Java™ 2 SDK Standard Edition platforme od verzije 1.4.

JSSE implementacija koja dolazi sa Java™ 2 SDK Standard Edition 1.4 platformom pruža "jaku" kriptografiju, ali zbog izvoznih pravila SAD-a (Sjedinjene Američke Države) ne dopušta zamenu sa alternativnim SSL/TLS implementacijama.

4.9. Java™ Authentication and Authorization Service

JAAS se može koristiti za potrebe autentifikacije korisnika tj. pouzdano i sigurno određivanje ko trenutno izvodi Java kôd i za autorizaciju korisnika tj. proveravanje da li korisnik ima potrebne dozvole za obavljanje određene operacije.

4.10. Java™ Generic Security Service

Java GSS-API se koristi za sigurnu razmenu poruka između aplikacija. GSS-API omogućuje aplikativnom programeru jedinstven pristup sigurnosnim uslugama preko čitavog niza sigurnosnih mehanizama uključujući i Kerberos.

4.11. Java™ Cryptography Extension

JCE pruža mogućnosti šifrovanja podataka, generisanje simetričnih šifarskih ključeva, sesijske razmene ključeva kao i generisanje MAC-ova (engl. *Message Authentication Code*). Podrška za šifrovanje podataka uključuje simetričnu i asimetričnu kriptografiju, blokovsko orijentisano šifrovanje i šifrovanje toka podataka.

JCE se sastoji od tri paketa:

- `javax.crypto`,
- `javax.crypto.interfaces`,
- `javax,crypto.spec`.

4.11.1. Paket `javax.crypto`

Pruža klase i interfejs potreban za šifarske operacije kao što su šifrovanje, generisanje simetričnih ključeva, sesijske razmene ključeva, kao i generisanje MAC-ova. Podrška za šifrovanje podataka uključuje simetričnu i asimetričnu kriptografiju, blokovsko orijentisano šifrovanje i šifrovanje toka podataka ("*oktetno orijentisano*").

Ovaj paket omogućuje korišćenje sigurnih tokova podataka i zapečaćenih objekata. Značajni interfejsi, klase i funkcije prikazani su u sledećim tabelama.

Interfejs	
Ime	**Opis**
SecretKey	Osnovni interfejs za simetrične šifarske ključeve.

Tabela 4.13. Interfejs paketa `javax.crypto`

Klase	
Ime	**Opis**
`Cipher`	Klasa koja pruža mogućnosti šifrovanja i dešifrovanja podataka i predstavlja jezgro JCE okvira (engl. *framework*).
`CipherInputStream`	Ova klasa koristi primerak (engl. *instance*) klase `InputStream` i `Cipher` tako da njegova metoda `read()`, pre nego što vrati pročitane podatke sa ulaznog toka, predprocesira podatke pomoću `Cipher` objekta.
`CipherOutputStream`	Ova klasa koristi primerak klase `OutputStream` i `Cipher` tako da njegova metoda `write()`, pre nego što zapiše primljene podatke na izlazni tok, predprocesira podatke pomoću `Cipher` objekta.
`KeyAgreement`	Klasa koja pruža mogućnost razmene (dogovora) ključeva. Ključevi koji se koriste pri uspostavljanju deljene tajne (engl. *shared secret*) koje se stvaraju pomoću klasa `java.security.KeyPairGenerator` ili `javax.crypto.KeyGenerator`.
`KeyGenerator`	Klasa koja pruža mogućnost generisanja simetričnih šifarskih ključeva.
`Mac`	Klasa koja pruža mogućnost generisanja MAC-ova.
`SealedObject`	Klasa koja pruža mogućnost šifarske zaštite objekata.
`SecretKeyFactory`	Klasa koja omogućuje stvaranje simetričnih ključeva iz datog spremišta ključa ili stvaranje određene reprezentacije ključa iz stvarnog objekta.

Tabela 4.14. Značajne klase paketa `javax.crypto`

Funkcije	
Ime	**Opis**
`BadPaddingException`	Funkcija koja ukazuje da dati ulazni podaci nisu dopunjeni na očekivani način.
`IllegalBlockSizeException`	Funkcija koja ukazuje da dati ulazni podaci uz blokovsko orijentisani šifarski algoritam nisu veličine koja je umnožak veličine osnovnog bloka.
`NoSuchPaddingException`	Funkcija koja ukazuje da zahtevani mehanizam dopune podataka nije raspoloživ.

Tabela 4.15. Značajne funkcije paketa `javax.crypto`

Klase predstavljaju samo programski interfejs bez stvarne implementacije pojedinih algoritama. Implementacija se ostavlja nekoj trećoj strani. Zbog toga se klasama koje koriste određene algoritme predaju imena tih algoritama kao znakovne konstante. Java 2 SDK Security API zahteva i koristi skup standardnih imena za algoritme, načine šifrovanja, mehanizme dopune, protokole razmene ključeva, generisanje ključeva i MAC-ova, ali to ne sprečava neku treću stranu da implementira neki novi algoritam i definiše vlastito ime za njega. Na taj način omogućena je jednostavna nadogradnja sigurnosnih mehanizama Java platforme. Java 2 SDK Security API definiše sledeći skup imena algoritama:

- za generisanje šifarskih algoritama: **AES**,**Blowfish**,**DES**, **DESede**, **PBE With<***algoritamSažimanja***>And<***šifarskiAlgoritam***>** ili **PBE With<***generatorPseudoSlučajnihBrojeva***>And<***šifarskiAlgoritam***>** (npr. **PBE With MD5 And DES**, **PBE With Hmac SHA1 And DESede**) **RC2**, **RC4**, **RC5**, **RSA**,
- za način šifrovanja: **NONE**, **CBC**,**CFB**, **ECB**, **OFB**, **PCBC**
- za mehanizam dopune: **NoPadding**, **OAEPWith<***algoritamSažimanja***>And<***funkcijaGenerisanjaMaske***>Padding**,**PKCS5 Padding**, **SSL3 Padding,**
- za protokole razmene ključeva: **Diffie Hellman**,
- za generisanje simetričnih ključeva: **AES**,**Blowfish**,**DES**, **DESede**, **Hmac MD5**, **Hmac SHA1**,
- za stvaranje ključeva sa SecretKeyFactory: **AES**,**DES**, **DESede**, **PBE With<***algoritamSažimanja***>And<***šifarskiAlgoritam***>** ili **PBE With<***funkcijaGenerisanjaMaske***>And<***šifarskiAlgoritam***>,**
- za MAC: **Hmac MD5**, **Hmac SHA1**,**PBE With<***mac***>.**

4.11.2. Paket `javax.crypto.interfaces`

Pruža interfejs za Diffie-Hellman ključeve definisane sa standardom PKCS#3.

4.11.3. Paket `javax.crypto.spec`

Pruža klase i interfejs za specifikaciju ključeva i parametara različitih algoritama. Specifikacija ključeva je transparentna reprezentacija materijala koju predstavlja ključ. Ključ može biti određen zavisno od algortima ili nezavisno od algoritma u nekom zapisnom formatu npr. ASN.1. Ovaj paket sadrži specifikacije Diffie-Hellman javnih i privatnih ključeva, DES, Triple DES i PBE tajnih ključeva (šifrovanje zasnovano na šifri, *engl. Password-Based Encryption*). Specifikacija parametara algoritama je transparentna reprezentacija skupa parametara korišćenih u nekom algoritmu. Ovaj paket sadrži specifikaciju parametara Diffie-Hellman, DES, Triple DES, PBE, RC2 i RC5 algoritama.

4.12. Java Card Appleti

Java Card appleti ne smeju se zameniti sa Java appletima samo zato što se i jedni i drugi nazivaju appletima. Java Card applet je Java program koji zadovoljava skup konvencija koje mu omogućuju izvođenje u Java Card izvršnoj okolini. On nije namenjen izvođenju unutar preglednika. Razlog zašto je izabran naziv applet za Java Card aplikacije dolazi od toga što se Java Card appleti mogu učitati u Java Card izvršnu okolinu nakon što je kartica proizvedena. Odnosno, za razliku od kartičnih aplikacija u mnoštvu postojećih sistema, appleti se ne moraju upisati u ROM prilikom proizvodnje, jer se mogu nakon što je kartica proizvedena, dinamički učitati na karticu.

Appletova klasa mora naslediti klasu *javacard.framework.Applet*, tako da je klasa *Applet* nadklasa svih appleta koji se nalaze na Java kartici. Ona je okvir koji definiše applet-ove metode i varijable (promenljive). Applet koji se izvodi na kartici je nadležnost klase *Applet*, objekt klase applet. Kao svaki perzistentni objekt, jednom kreiran, applet zauvek postoji na kartici.

Java Card izvršna okolina podržava postojanje više appleta na istoj kartici, tzv. multi-aplikativno okruženje. Osim toga jedan applet može imati više nadležnosti. Na primer, može se kreirati nadležnost appleta koji implementira novčanik sa valutom američkog dolara i druga nadležnost sa valutom srpskog dinara.

4.13. Dogovor vezan za pridruživanje imena paketu i appletu

Paketi i programi se unutar Java platforme jedinstveno indentifikuju koristeći Unicode stringove i nacrt imena baziran na imenima internet domena. Kod Java Card platforme svaka se nadležnost appleta jedinstveno identifikuje i selektuje korišćenjem aplikativnog identifikatora – AID (engl. *Application Identifier,* ***AID***). Svakom se Java paketu takođe dodeljuje AID. Kada se učita na karticu, paket se povezuje sa drugim paketima na kartici pomoću njihovih AID-ova.

ISO 7816 opisuje izgled AID-a i korisiti ga za jedinstvenu identifikaciju kartičnih aplikacija i nekih vrsta datoteka u kartičnim file sistemima. Sam AID je polje bajtova koje se može prikazati kroz dva odvojena dela (slika 4.6). Prvi deo je 5-bajtna vrednost koja se zove identifikator izvora – RID (engl. *Resource Identifier,* ***RID***). Drugi deo je promenljive dužine, i zove se identifikator vlasničkog proširenja – PIX (engl. *Proprietary Identifier Extension,* ***PIX***). PIX može biti dužine od 2 do 12 bajtova. Zbog toga AID može u ukupnoj dužini varirati od 5 do 16 bajta.

RID (5 bajtova)	PIX (0 - 11 bajtova)

Slika 4.6. Aplikativni identifikator

ISO kontroliše dodelu RID-ova kompanijama; svaka kompanija mora imati jedinstveni RID. Dodela PIX-ova ostavljena je na upravljanje kompanijama. Detalji vezani uz AID mogu se naći u ISO 7816-5, *AID Registration Category D Format*.

Kod Java Card platforme AID paketa se konstruiše ulančavanjem RID-a kompanije i PIX-a za dotični paket. Slično se konstruiše i AID appleta. Ulančava se RID izdavača appleta i PIX dotičnog appleta. Appletov AID ne sme imati vrednost jednaku vrednosti nekog AID paketa ili vrednost jednaku vrednosti AID-a nekog drugog appleta. Naravno, kako RID u AID-u određuje izdavača appleta, AID paketa i AID-ovi appleta definisanih u paketu moraju deliti isti RID.

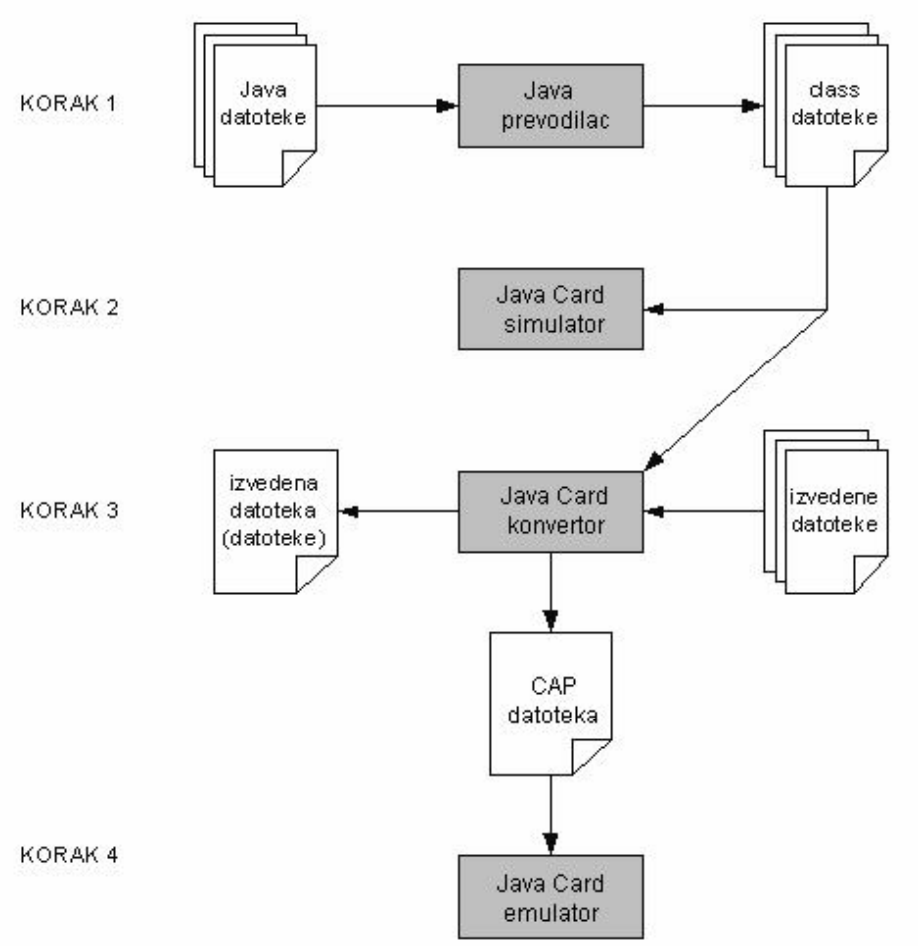
AID paket i standardni appletov AID, za svaki applet definisan u dotičnom paketu, definisani su u CAP datoteci. Oni se isporučuju konvertoru kad generiše CAP datoteku.

4.14. Razvojni proces appleta

Programiranje Java Card appleta počinje kao sa svakim drugim Java programom. Razvojni programer piše jednu od Java klasa i prevodi izvorni tekst sa Java prevodiocem (engl. *java compiler,* ***javac***), stvarajući tako jednu ili više class datoteka. Slika 4.7 prikazuje applet-ov razvojni proces.

Sledeći korak je izvođenje appleta u simulacijskom okruženju. Tu se ispituje ponašanje appleta u radu, pronalaze i ispravljaju greške. Simulator na PC-u oponaša Java Card izvršnu okolinu. U simulacijskom okruženju applet se izvodi na Java mašini stanja odnosno njegove class datoteke izvršava Java mašina stanja.

Na taj način simulator može iskoristiti mnoštvo Java razvojnih alata i omogućiti razvojnom programeru ispitivanje ponašanja appleta i brzo sagledavanje rezultata njegovog izvođenja, bez prolaska kroz proces pretvaranja. Svakako, neka svojstva vremena izvođenja Java Card mašine stanja, kao što je firewall appleta i ponašanje prelaznih i perzistentnih objekata, ne mogu se proučiti.



Slika 4.7. Razvojni proces appleta

Treći korak je da se class datoteke appleta koje čine Java paket pretvore u CAP datoteku korišćenjem Java Card konvertora. On uzima kao ulaz ne samo class datoteke predodređene za pretvaranje, već i jednu ili više izvedenih datoteka. Kad je paket appleta pretvoren, konvertor kao izlaz može dati i izvedenu datoteku za dati paket. CAP datoteka i izvedena datoteka predstavljaju jedan Java paket. Ako applet obuhvata nekoliko paketa, za svaki paket biće kreirana izvedena datoteka i CAP datoteka.

U četvrtom koraku se CAP datoteka/e koje predstavljaju applet učitavaju i testiraju u emulacionom okruženju. Emulator takođe simulira Java Card izvršnu okolinu na PC-u ili radnoj stanici. Međutim, emulator je sofisticiraniji alat za testiranje jer sadrži implementaciju Java Card mašine stanja. Ponašanje appleta prilikom izvršavanja u emulatoru trebalo bi biti jednako njegovom ponašanju na kartici. U ovoj fazi razvoja mere se i testiraju ponašanje appleta prilikom izvođenja.

Većina Java Card simulatora i emulatora dolazi sa ugrađenim programom za ispitivanje grešaka (engl. *debugger*). On omogućuje programeru postavljenje ispitnih tački u programski tok i izvršavanje programa korak po korak, gde tada programer, u simuliranoj ili emuliranoj Java Card izvršnoj okolini, posmatra promene stanja appleta u izvođenju.

Konačno, poslednji korak u razvoju appleta, nakon što je applet testiran i doveden u oblik da je predstavljen sa jednom ili više CAP datoteka, je učitavanje i instalacija appleta na karticu.

4.15. Instalacija appleta

Operativni sistem na kartici, koji je obično specifičan za svakog proizvođača, i Java Card izvršna okolina – uključujući i izvorne metode (engl. *native methods*), Java Card mašina stanja, API klase i biblioteke – upisuju se u ROM prilikom proizvodnje kartice. Taj proces pisanja stalnih komponenti u ispisnu neprogramabilnu memoriju čipa naziva se maskiranje i varira od proizvođača do proizvođača.

4.15.1. ROM appleti

Klase Java Card appleta mogu se za vreme procesa proizvodnje kartice maskirati u ROM zajedno sa JCRE-om i drugim sistemskim komponentama. Nadležnosti appleta se stvaraju u EEPROM-u od strane JCRE-a za vreme JCRE inicijalizacije ili u kasnijim fazama proizvodnje. Takvi appleti se nazivaju ROM appleti.

ROM appleti su predefinisani appleti koji dolaze sa karticom, a izdaju ih izdavači kartica. Zbog toga što sadržaj appleta kontroliše izdavač, Java Card tehnologija dopušta ROM appletima da deklarišu izvorne metode čija implementacija može biti u drugim programskim jezicima, kao što su C ili mašinski jezici. Izvorne metode ne podležu sigurnosnim mehanizmima koje nameće Java Card mašina stanja.

4.15.2. Predizdati i postizdati appleti

Alternativno, nakon što je kartica proizvedena, mogu se klase Java Card appleta i pripadne klase biblioteka učitati, odnosno upisati u ispisnu programabilnu memoriju (kao što je EEPROM) Java pametne kartice. Takvi appleti se mogu dalje deliti na predizdate i postizdate aplete. Nazivi predizdati i postizdati dolaze iz činjenice da se appleti mogu učitati pre ili nakon izdavanja kartice. Predizdati appleti tretiraju se na isti način kao i ROM appleti; i jedni i drugi su kontolisani od izdavača.

Za razliku od ROM ili predizdatih appleta, postizdatim appletima nije dozvoljena deklaracija izvornih metoda. Razlog tome je što JCRE ne može kontrolisati sadržaj appleta i dozvoljavajući učitanim appletima da sadrže izvorni kod postojala bi mogućnost narušavanja Java Card sigurnosti (engl. *Java Card Security*).

Podpoglavlja koja slede koncentrisaće se na instalaciju postizdatih appleta. Obično se predizdati appleti učitavaju korišćenjem istih mehanizama za učitavanje postizdatih appleta, ali Java Card tehnologija ostavlja tu odluku izdavačima kartica.

4.15.3. Instalacija postizdatih appleta

Instalacija appleta se odnosi na proces učitavanja applet-ovih klasa u CAP datoteku, njihovog spajanja sa stanjem izvođenja Java Card izvršne okoline i na proces

kreiranja nadležnosti appleta, da se omogući dovođenje appleta u stanje selekcije, a zatim i u stanje izvođenja.

Kod Java Card platforme osnovna jedinica koja se učitava i instalira je CAP datoteka. CAP datoteka se sastoji od klasa koje čine Java paket. Najmanji applet predstavljen je sa Java paketom koji sadrži jednu klasu, koja nasleđuje klasu *javacard.framework.Applet.* Složeniji appleti sa većim brojem klasa mogu biti organizovani u jedan Java paket ili skup Java paketa.

Da bi učitao applet, van kartični instalacioni program uzima CAP datoteku i transformiše je u skup APDU naredbi koje sadrže CAP datoteku. Kartični instaler, izmenjujući APDU naredbe sa van kartičnim instalacionim programom, upisuje sadržaj CAP datoteke u perzistentnu kartičnu memoriju i povezuje klase iz CAP datoteke sa ostalim klasama koje se nalaze na kartici. Instaler takođe kreira i inicijalizuje sve podatke koje interno koristi JCRE da podrži applet. Ako applet zahteva za svoje izvođenje nekoliko paketa svaka CAP datoteka se učita na karticu.

Kao poslednji korak u instalaciji appleta instaler stvara nadležnost appleta i registruje je kod JCRE-e. Da bi to učinio poziva *install* metodu:

public static void install (byte[] bArray, short offset, byte length)

Ta metoda predstavlja početnu metodu za applet, ulaznu tačku, slično *main* metodi u Java aplikaciji. Applet mora implementirati *install* metodu. U njoj poziva appletov konstruktor da kreira i inicijalizuje nadležnost appleta. Preko parametra *bArray* prenose se u *install* metodu instalacioni parametri. Oni se mogu koristiti prilikom inicijalizacije appleta. Instalacioni parametri šalju se kartici zajedno sa CAP datotekom. Razvojni programer appleta definiše format i sadržaj instalacionih parametara.

Nakon inicijalizacije appleta i njegove registracije kod JCRE-a, on se može selektovati i izvršiti. JCRE identifikuje trenutno izvršavani applet, odnosno njegovu nadležnost, koristeći njegov AID. Applet se može registrovati kod JCRE koristeći svoj predodređeni AID pronađen u CAP datoteci, ili može izabrati drugi, alternativni AID. U tu svrhu se mogu koristiti, da pruže izbor alternativnog AID-a, instalacioni parametri.

U kreiranju višestrukih nadležnosti appleta potrebno je za svaku nadležnost pozvati *install* metodu. Svaka nadležnost appleta se identifikuje jedinstveni AID-om.

Java Card okolina omogućava da applet bude napisan i izvršen, pritom neznajući da su njegove klase učitane. Jedina appletova obaveza u procesu instalacije je implementacija *install* metode.

4.15.4. Oporavak od greške za vreme instalacije appleta

Proces instalacije ima svojstvo transakcije. U slučaju greške, kao što je nedostatak memorije, fizičko oštećenje kartice i niz drugih grešaka, instaler odbacuje CAP datoteku i sve applete stvorene za vreme instalacije i obnavlja prostor i pređašnje stanje JCRE-a.

4.15.5. Instalaciona ograničenja

Treba napomenuti da se instalacija appleta razlikuje od dinamičkog učitavanja klasa za vreme izvođenja, koje podržava Java mašina stanja na PC-u. Instalacija Java

Card appleta jasno označava preuzimanje i učitavanje klasa kroz instalacioni proces nakon što je kartica proizvedena.

Zbog toga u instalaciji Java Card appleta treba istaći dve činjenice. Prvo, izvršavanje appleta na kartici odnosi se samo na klase koje već postoje na kartici jer ne postoji način učitavanja klase za vreme normalnog izvršavanja appletovog koda (izvršnog teksta).

Drugo, redosled učitavanja mora garantovati da svaki novo učitani paket referencira samo one pakete koji se nalaze na kartici. Kao primer uzmma se instalacija appleta. Da bi instalacija bila moguća, na kartici se mora nalaziti paket *javacard.framework*, jer sve klase appleta nasleđuju klasu *javacard.framework.Applet.* U slučaju kružnih, unakrsnih referenci, npr. paket A i paket B referenciraju jedan drugog, instalacija rezultira greškom.

4.16. Sigurnosne osobine Java Card platforme

Sigurnosne osobine Java Card platforme kombinacija su osnovnih sigurnosnih osobina jezika Jave i dodatne sigurnosne zaštite definisane Java Card platformom.

4.16.1. Sigurnosne osobine Java jezika

Java Card platforma podržava podskup programskog jezika Jave i specifikaciju mašine stanja prilagođenu aplikacijama pametnih kartica. Zbog toga Java Card platforma nasleđuje sigurnosne osobine ugrađene u podržani podskup jezika Jave. Sigurnost jezika Jave stvara temelje sigurnosti Java Card platforme:

- Java jezik je strogo definisan. Ne može se učiniti ni jedna ilegalna konverzija podataka, kao što je pretvaranje celobrojnog tipa podatka u pokazivač (engl. *pointer*),
- Java jezik nameće proveru granice, dužine polja, prilikom pristupanja polju podataka,
- Java jezik ne poznaje aritmetiku pokazivača. Zbog toga ne postoji mogućnost krivotvorenja pokazivača sa ciljem da se dozvoli zlonamernim programima "njuškanje" po sadržaju memorije,
- Incijalizacija promenljivih mora biti sprovedena pre korišćenja istih,
- Nivo pristupa svim klasama, metodama i poljima je strogo kontrolisan. Na primer, privatna metoda ne može biti pozvana izvan njene definisane klase.

4.16.2. Dodatne sigurnosne osobine Java Card platforme

U izrazito sigurnosno-prosvećenom svetu pametnih kartica, želja izdavača je da kartica ima sigurnu računarsku platformu koja će zadovoljiti specijalne zahteve kartičnih sistema. Dodatne sigurnosne osobine definisane u Java Card platform su:

- *Prelazni i perzistentni objektni modeli* – Na Java Card platformi objekti se standardno čuvaju u perzistentnoj (stalnoj) memoriji. Iz sigurnosnih razloga i razloga vezanih uz performanse, Java Card platforma dopušta da se privremeni podaci, kao što su ključevi sesija, čuvaju kao prelazni objekti u RAM memoriji.

Životno vreme takvih objekata može biti deklarisano kao CLEAR_ON_RESET ili CLEAR_ON_DESELECT. To znači da se sadržaj prelaznog objekta postavlja na predefinisanu vrednost (zero, false, null) svaki put kad kartica uspostavi novu sesiju (reset) ili kad se deselektuje trenutno označeni, selektovani applet.

- *Atomarnost i transakcije* – Na Java Card platformi podaci se čuvaju kao objekti u perzistentnoj memoriji. Nad njima se vrše operacije pisanja i čitanja. Za vreme operacije pisanja može doći do hardverske greške ili do gubitka napajanja. Da u takvim slučajevima osigura integritet podataka Java Card tehnologija definiše tri sigurnosna svojstva. Kao prvo, Java Card platforma garantuje atomarnost u pojedinačnom ažuriranju elementa perzistentnog objekta ili klase. Što znači da, kad dođe do greške za vreme ažuriranja elementa, platforma osigurava da se sadržaj elementa obnovi na pređašnju vrednost. Drugo, metoda *arrayCopy* u klasi *javacard.framework.Util* garantuje atomarnost blokovskog ažuriranja višestrukih elemenata podataka u polju. Ovde atomarnost znači da će svi bajtovi biti ili ispravno kopirani ili će kompletno odredišno polje biti obnovljeno na pređašnje vrednosti. Treće, Java Card platforma podržava transakcioni model u kome applet atomarno ažurira nekoliko različitih elemenata unutar različitih perzistentnih objekata. Sva ažuriranja unutar transakcije moraju izvesti ispravno i konzistentno ili se svi perzistentni elementi vraćaju u pređašnja stanja.
- *Firewall appleta* – On čuva kako sigurnost i integritet sistema (JCRE) tako i sigurnost i integritet svakog appleta koji se nalazi na Java pametnoj kartici. Firewall appleta sprovodi izolaciju appleta i odvaja sistemski prostor od appletovog prostora. U nacrtu firewall-a svaki se applet izvodi unutar konteksta (engl. *context*) (prostor objekta), a sam firewall predstavlja granicu među njima. Appleti ne mogu međusobno pristupati svakom objektu osim ako su definisani unutar istog paketa (time dele isti kontekst) ili ako za pristup objektu koriste dobro definisane, sigurnosne mehanizme deljenja objekata podržane platformom.
- *Deljenje objekata* – Deljenje objekata je u Java Card sistemu postignuto na sledeće načine. Prvo, JCRE je ovlašćen korisnik sa mogućnošću pristupa svim appletima i svim objektima kreiranim od strane appleta. Drugo, applet potražuje pristup JCRE servisima i resursima kroz tzv. ulazne tačke, ulazne objekte JCRE-a. Treće, appleti koji pripadaju različitim kontekstima mogu deliti objekte ako su oni u nadležnosti klasa koje implementiraju interfejs za deljenje objekata. Takvi objekti nazivaju se deljeni objekti (engl. *shareable interface objects*). I konačno, appleti i JCRE mogu deliti podatke preko globalno definisanih polja.
- *Izvorne metode u appletima* – Izvorne metode ne izvršava Java Card mašina stanja i kao takve nisu predmet sigurnosne zaštite Java Card platforme. Zbog toga je postizdatim appletima (appleti koji su učitani na karticu nakon što je kartica izdata) zabranjeno sadržati izvorne metode. ROM appleti i predizdati appleti su kontrolisani od strane izdavača pa se njima dopušta korišćenje izvornih metoda. U takvim appletima je interfejs izvornog koda predstavljen vlasničkom tehnologijom izdavača kartica.

4.16.3. Sigurnost appleta

Uopšte, sigurnost aplikacije određena je sigurnošću platforme na kojoj se aplikacija izvodi, kao i sigurnosnim osobinama implementiranim u sam dizajn aplikacije. Java Card platforma dizajnirana je tako da pruži izgradnju, instalaciju i izvođenje appleta u visoko zaštićenom okruženju.

Određeni nivo sigurnosti potrebno je ugraditi u dizajn appleta. Zbog prisutnosti sigurnosnih osobina u samoj platformi, razvojni programeri appleta mogu usredsrediti svoje napore na definisanje strategije zaštite appleta, pre nego na programiranje appleta na način da appleti kompenzuju nesigurnost platforme.

Dobro definisana strategija zaštite mora zadovoljiti sledeća svojstva:

- *Autentifikacija* – pristup funkcijama i podacima appleta mora biti strogo kontrolisan. Neophodno je sprovesti proceduru autentifikacije u cilju provere identiteta aplikacije terminala. Takođe prilikom deljenja objekata applet server mora proveriti identitet appleta klijenta pre nego što mu dozvoli korišćenje svojih deljenih objekata.
- *Tajnost* – neophodna je zaštita privatnosti appletovih podataka; onemogućavanje pristupa zaštićenim podacima, kao što su broj računa i iznos kredita, bez prethodno izvršene autenitifikacije. Tajni podaci, PIN i tajni šifarski ključevi, ne smeju nikad napustiti karticu.
- *Integritet* – potrebno je osigurati ispravnost appletovih podataka. Podaci appleta ne smeju biti menjani bez prethodno izvršene procedure autentifikacije. Promena podataka štiti se i proverom potencijalnih greški, (npr. iznos e-novčanika ne sme preći maksimalni iznos, ali ni postati negativan).

Različiti appleti zahtevaju različite nivoe sigurnosti. Implementirani sigurnosni mehanizmi često nameću zahteve na računarske resurse, smanjujući time perfomanse izvođenja appleta. Zbog toga je potrebno za svaki applet odrediti njegove sigurnosne zahteve i prema njima implementirati odgovarajuće sigurnosne mehanizme da se zadovolji željeni nivo sigurnosti.

Deo V

5. Sigurna autentifikacija

Komunikacija, pristup informacijama i obavljanje različitih vrsta transakcija u mrežnom okruženju postala je svakodnevica. Podaci koji se razmenjuju mogu lako biti vidljivi svima prisutnim u istom mrežnom okruženju. Samim time nameće se više pitanja vezanih za računarsku sigurnost. Najvažnija pitanja su sigurna autentifikacija i tajnost informacija koje se razmenjuju. Najčešći oblik autentifikacije je autentifikacija gde se korisnik predstavlja korisničkim imenom i lozinkom. Takva autentifikacija međutim ne garantuje veliku sigurnost.

5.1. Osnove računarske sigurnosti

Elektronske informacije mogu se jednostavno umnožiti i izmeniti, a umnožene informacije nemoguće je razlikovati od originala. U mrežnoj okolini informacije je moguće prisluškivati, presresti i izmeniti. Upravo navedene osobine elektronskih podataka i osobine mrežnog okruženja predstavljaju sigurnosni problem. Računarska sigurnost ugrožena je bilo kojim oblikom pretnje. Pretnja se definiše kao stanje ili događaj koji može naškoditi osoblju, mreži i računarskim resursima u obliku uništavanja, razotkrivanja ili modifikacije podataka, uskraćivanja usluge, prevare i zloupotrebe.

5.1.1. Sigurnosni incidenti

U mrežnom okruženju definisani su sigurnosni incidenti:

- ***Prisluškivanje*** - U mrežnom okruženju lako se mogu prisluškivati, odnosno videti svi paketi na mreži pa se na taj način može doći do informacija koje su namenjene nekom drugom,
- ***Lažno predstavljanje*** - Lažno se može predstaviti osoba ili računar,
- ***Prekidanje.***
- ***Modifikacija paketa.*** Paketi namenjeni nekom drugom mogu se presresti i modifikovati,
- ***Ponovno slanje starih snimljenih paketa.*** Komunikacija se prisluškuje i paketi koji učestvuju u komunikaciji se snime. Nakon završene komunikacije ti isti paketi se ponovo šalju.

5.1.2. Sigurnosni zahtevi

U cilju postizanja računarske sigurnosti definisani su sigurnosni zahtevi:

- ***Autentičnost -*** Potvrda identiteta korisnika ili računara koja sprečava lažno predstavljanje,
- ***Integritet -*** Garancija da su informacije poslate, primljene i ako treba sačuvane u izvornom i nepromenjenom obliku. Integritet sprečava sigurnosni incident odnosno modifikaciju paketa,
- ***Poverljivost ili tajnost -*** Sprečava prisluškivanje ili presretanje,
- ***Neporecivost*** – Učesnici transakcije ne mogu odbiti ili poreći akciju u kojoj su učestvovali čime se sprečava incident ponovnog slanja snimljenih starih paketa,
- ***Kontrola pristupa,***
- ***Raspoloživost.***

5.1.3. Kriptografija

Kriptografija (grč. *kryptós* = skriven, grč. *gráphein* = pisati) je matematička i računarska oblast koja se bavi sigurnošću informacija. Šifrovanje je postupak pretvaranja podataka u drugi oblik koji se ne može razumeti bez dešifrovanja. Šifrovanje se formalno može opisati izrazom:

$$c=E(m,e)$$

gde su:

m – jasni podaci,
c – šifrovani podaci (šifrat),
E – postupak šifrovanja,
e – ključ šifrovanja.

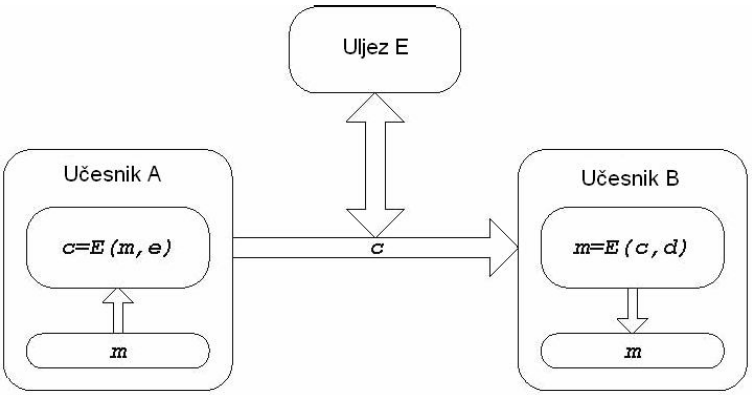
Šifrovane informacije mogu se razumeti samo nakon dešifrovanja. Dešifrovanje je postupak kojim se šifrovani podaci vraćaju u izvorni oblik. Dešifrovanje se formalno može opisati izrazom:

$$m=D(c,d)$$

gde su:

m – jasni podaci,
c – šifrovani podaci,
D – postupak dešifrovanja,
d – ključ dešifrovanja.

Šifrovanje i dešifrovanje omogućuju poverljivost, odnosno tajnost informacija u smislu da informaciju mogu saznati samo oni kojima je namenjena. Slika 5.1. prikazuje postupak razmene tajne informacije preko nesigurnog kanala koju smeju znati samo učesnici komunikacije, odnosno učesnik *A* i učesnik *B*. Komunikacija se odvija u otvorenom okruženju gde se svi podaci mogu prisluškivati. Učesnik *A* šifruje informaciju i šalje šifrovanu informaciju učesniku *B*. Iako uljez *E* može prisluškivati komunikaciju ne može razumeti podatke. Da bi se podaci mogli razumeti potrebno ih je dešifrovati, što mogu samo oni koji znaju ključ, u ovom slučaju učesnik *B*.



Slika 5.1. Komunikacija preko nesigurnog kanala

Za šifrovanje koriste se simetrični i asimetrični algoritmi kao dva osnovna tipa šifarskih algoritama. Simetrični algoritmi koriste isti ključ za šifrovanje i dešifrovanje, dok asimetrični algoritmi koriste različite ključeve za šifrovanje i dešifrovanje.

5.1.3.1. Simetrična kriptografija

Kod simetrične kriptografije koristi se isti ključ za šifrovanje i dešifrovanje i taj ključ se naziva simetrični, tajni ili sesijski ključ. Postupak dešifrovanja može i ne mora biti jednak postupku šifrovanja. Neki od najpoznatijih simetričnih algoritama su *DES*, *Triple DES* i *AES*. *DES* (engl. *Data Encryption Standard* -DES) je simetrični šifarski algoritam razvijen sredinom 70-tih u okviru kompanije *IBM*. 1981. godine. *DES* je potvrđen kao *ANSI* norma (*ANSI X3.92 Data Encryption Standard*). *DES* koristi ključ dužine 64 bita. Od svakog okteta ključa jedan bit služi za proveru pariteta tako da osam bita ključa otpada na proveru pariteta. Prema tome efektivna dužina ključa je 56 bita. Šifrovanje se izvodi po blokovima dužine 64 bita. S obzirom na dužinu ključa, *DES* ključ može se otkriti grubim pretraživanjem prostora u relativno kratkom vremenu. Iz tog razloga nastao je *Triple DES* poznat pod nazivima *TDES* i kao norma *TDEA* (engl. *Triple Data Encryption Algorithm* -TDEA).

TDEA je simetrični šifarski algoritam nastao na temelju *DES* algoritma. *TDEA* koristi ključ dužine 192 bita od čega 24 bita otpada na zaštitu pariteta tako da je efektivna dužina 168 bita. Ključ *TDEA* algoritma sastoji se od tri ključa *DES* algoritma, a postupak šifrovanja i dešifrovanja se izvodi pozivanjem *DES* algoritma tri puta. Dakle šifrovanje se takođe izvodi po blokovima dužine 64 bita. *TDEA* algoritam, odnosno povećanje dužine

ključa bio je najbrži i najjednostavniji način sprečavanja napada na *DES* grubim pretraživanjem prostora.

1997. godine. *NIST* (engl. *National Institute of Standards and Technology*) izdaje poziv na takmičenje za *AES* (engl. *Advanced Encryption Standard*) koji će zameniti *DES* algoritam. U oktobru 2000. godine na takmičenju je pobedio *Rijndael* algoritam. *Rijndael* algoritam izradili su Joan Daemen i Vincent Rijmen. *Rijndael* je iterativni algoritam koji radi sa varijabilnim (promenljivim) dužinama ključa i bloka podataka.

Ključ *DES* algoritama čija je dužina 56 bita, kao i ključevi male dužine ostalih simetričnih algoritama obično se mogu otkriti grubim pretraživanjem prostora u relativno kratkom vremenu. Prema tome se simetrični ključ mora često menjati i obično se koristi za jednu sesiju, a u većini slučajeva se generiše nasumično (engl. *random*). Ukoliko se koriste algoritmi sa većom dužinom ključa ne dolazi do takvih problema. Međutim, ipak se postavlja pitanje sigurne razmene simetričnog ključa nesigurnim kanalom između dve strane koje žele vršiti komunikaciju. Odgovor na to pitanje daje asimetrična kriptografija koja je nastala upravo zbog problema razmene simetričnog ključa.

5.1.3.2. Asimetrična kriptografija

Utemeljivači simetrične kriptografije su W. Diffie i E. Hellman koji su 1976. godine opisali ideju kriptografije koja se temelji na dva ključa, privatnom i javnom ključu. Informacije šifrovane javnim ključem mogu se dešifrovati samo privatnim ključem ali i obratno. Asimetrična kriptografija nastala je zbog potrebe sigurne razmene simetričnog ključa što je moguće šifrovanjem simetričnog ključa asimetričnim javnim ključem. Šifrovani sadržaj može se dešifrovati samo privatnim ključem, odnosno to može samo osoba koja je vlasnik javnog asimetričnog ključa. Osim toga šifrovanje privatnim, a dešifrovanje javnim ključem pokazalo se takođe kao odlično svojstvo jer omogućava ostvarenje elektronskog potpisa, dakle samo vlasnik privatnog ključa može šifrovati podatke, a svi ih mogu proveriti.

Ključevi trebaju biti povezani jednosmernom funkcijom. Odnosno ne sme se ostvariti mogućnost da se može izračunati privatni ključ iz javnog ključa, ili se barem ne sme omogućiti izračunavanje u razumnom vremenu. Asimetrična kriptografija pruža daleko veću sigurnost, ali potrebno je i puno više vremena i računskih resursa za šifrovanje i dešifrovanje informacija. Kako je brzina u većini slučajeva bitan faktor, a računarski resursi su takođe ograničeni, asimetrična kriptografija koristi se samo za šifrovanje malih količina podataka. Najpoznatiji i najkorišćeniji asimetrični algoritam je *RSA* algoritam. *RSA* algoritam su osmislili *Ron **R**ivest*, *Adi **S**hamir* i *Leonard **A**dleman* 1978. godine, po čijim je inicijalima dobio naziv *RSA*. Jednosmerna funkcija za generisanje para ključeva u *RSA* algoritmu je množenje dva velika prosta broja, tako da se sigurnost algoritma zasniva se na težini faktorizacije velikih prostih brojeva. Taj problem je jedan od najstarijih u teoriji brojeva, rešenje je jednostavno, ali zahteva mnogo vremena.

5.1.4. Sažetak poruke

Sažetak podataka dobija se funkcijama sažimanja koje iz varijabilne dužine podataka daju sažetak fiksne dužine. Zavisno od korišćene funkcije sažimanja, razlikuje se maksimalna dužina podataka koji se mogu sažimati kao i fiksna dužina sažetka. Maksimalna dužina podataka koji se mogu sažimati je toliko velik broj da zapravo i ne predstavlja ograničenje, a sažetak je par desetina okteta. Za *SHA-1* algoritam maksimalna dužina podataka koji se mogu sažimati je 2^{64} bita, a dužina sažetka je 160 bita.

Funkcije sažimanja imaju sledeće osobine:

1 Jednosmerne su, tako da je iz sažetka nemoguće doći do originalne informacije ili je vrlo teško,
2 Različite informacije preslikavaju se u različite sažetke, čak i ako se informacije razlikuju za samo jedan bit. Postoji verovatnoća da različite informacije rezultiraju istim sažetkom ali ta verovatnoća je jako mala,
3 Ista informacija uvek se preslikava u jedan sažetak,
4 Dužina sažetka je konstantna bez obzira na dužinu informacije. Ipak, dužina informacije ima gornju granicu zavisno od vrste algoritma.

Najpoznatije funkcije sažimanja su: *MD4*(*RFC 1320*), *MD5* (*RFC 1321*), *SHA*(*FIPS PUB 180-2*).

5.1.5. Elektronski potpis

5.1.5.1. Svojeručni potpis

Svojeručni potpis je jedinstven i nezavistan od dokumenta koji se potpisuje. Potpis može proveriti bilo koja osoba koja primi potpisani dokument, a koja je već videla potpis ili ima kopiju za upoređivanje.

5.1.5.2. Potpis elektronskih informacija

Potpis elektronske informacije nije i ne može biti digitalizovana slika svojeručnog potpisa. Kod elektronske informacije koncept potpisa se menja. Potpis više ne sme biti samo jedinstven osobi koja potpisuje informacije nego i nezavistan od informacije. U tom slučaju potpis bi mogao bilo ko kopirati, umnožiti i koristiti za potpisivanje drugih informacija. Potpis elektronskih informacija mora biti jedinstven, odnosno samo jedna osoba ili jedan uređaj može potpisati elektronsku informaciju i potpis mora biti zavistan od informacije koja se potpisuje. Ubuduće, svako ko ima potpisanu informaciju mora biti u mogućnosti proveriti ispravnost potpisa ili barem odvojiti potpis od informacije u slučaju da ne želi proveravati integritet informacije. Proverom potpisa vrši se autentifikacija potpisnika i proverava se integritet informacija.

5.1.5.3. Razlike između elektronskog i digitalnog potpisa

Često dolazi do zabune i koristi se jedan od izraza bilo elektronski ili digitalni potpis a da se u stvari radi o onom drugom.

Elektronski potpis je opšti izraz neutralan na tehnologiju i način izvođenja koji obuhvata sve metode kojima se može potpisati elektronska informacija. Za elektronski potpis definišu se zahtevi kao što su očuvanje integriteta potpisanih informacija, neporecivost potpisivanja i identifikaciju potpisnika.

Digitalni potpis je izveden iz elektronskog potpisa, temeljen na asimetričnoj kriptografiji i funkcijama sažimanja.

5.1.6. Digitalni potpis

Jedna od najpoznatijih ostvarenja elektronskog potpisa je digitalni potpis koji je zasnovan na asimetričnoj kriptografiji i algoritmima sažimanja. S obzirom da je asimetrična kriptografija zahtevna za računarski moćne resurse, u zavisnosti od računarskoj moći i dužini informacije koja se želi šifrovati traje postupak šifrovanja. Zbog toga se koriste funkcije sažimanja koje su izuzetno brze jednosmerne funkcije (*surjekcije*) koje daju sažetak podataka male konstantne dužine od par desetine okteta. Ta operacija je ireverzibilna, odnosno iz sažetka se ne mogu dobiti originalni podaci. Pomoću funkcija sažimanja i asimetrične kriptografije definisan je koncept potpisivanja elektronskih informacija. Postupak je uveliko ubrzan funkcijama sažimanja zbog njihove brzine u odnosu na asimetrično šifrovanje.

Digitalni potpis definisan je kao asimetrično šifrovani sažetak informacije privatnim ključem.

$$Digitalni\ potpis = E(Q(m), S_a))$$

gde su:

- m - informacija,
- E - asimetrična funkcija šifrovanja,
- Q - funkcija sažimanja,
- S_a - privatni ključ.

Digitalni potpis osigurava integritet elektronske informacije, neporecivost i identifikaciju potpisnika što se dokazuje proverom potpisa. Ako je informacija izmenjena nakon što je bila potpisana to će se detektovati proverom potpisa.

5.1.6.1. Provera digitalnog potpisa

Provera potpisa podrazumeva dešifrovanje sažetka, računanje sažetka primljenih informacija i upoređivanjem sažetka primljenih informacija sa sažetkom koji je primljen šifrovan.

Postupak provere digitalnog potpisa je sledeći:

1. Dešifrovanje sažetka javnim ključem:

$$Q(m)=D(E(Q(m),Sa),Pa)$$

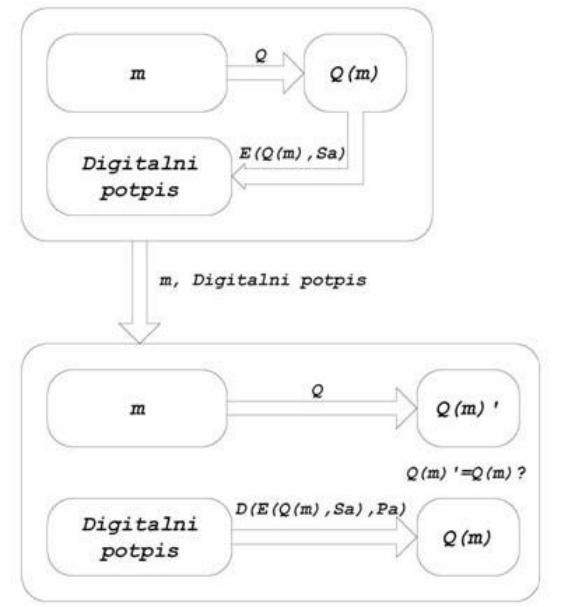
gde su:

D – asimetrična funkcija dešifrovanja,

P_a – javni ključ,

$Q(m)$ – sažetak originalne informacije.

2. Računanje sažetka primljene informacije $Q(m)'$,
3. Upoređivanje sažetka primljene informacije sa sažetkom koji je primljen šifrovan. Upoređivanjem $Q(m)'$ i $Q(m)$ proverava se integritet informacije i identifikacija potpisnika. Prema tome $Q(m)'$ i $Q(m)$ moraju biti identične, u protivnom ili je informacija naknadno izmenjena ili je autor nije potpisao ili oboje.



Slika 5.2. Digitalni potpis i provera potpisa

Najpoznatiji i najčešće korišćeni algoritmi za digitalni potpis su *RSA* i *DSA* (engl. *Digital Signature Algorithm* -DSA).

RSA algoritam digitalnog potpisa koristi *RSA* šifarski algoritam i *MD2*, *MD5* ili *SHA1* funkciju sažimanja.

NIST institut je 1991. godine objavio *DSA* algoritam kao normu za digitalni potpis (engl. *Digital Signature Standard* -DSS), za stvaranje i proveru digitalnog potpisa. *DSS* je odabran u saradnji sa *NSA* (engl. *National Security Agency*) kako bi se koristio kao norma

za autentifikaciju Vlade SAD-a. *DSA* se temelji na diskretnom logaritamskom problemu (engl. *discrete logarithm problem*). Diskretni logaritamski problem predstavlja jednosmernu funkciju kao i problem faktorizacije velikih prostih brojeva koji se koristi kod *RSA* algoritma generisanja ključeva. Pretpostavka je da je diskretan logaritamski problem teško rešiv i da se ne može izračunati u realnom vremenu. *RSA* se može koristiti i za šifrovanje i za digitalni potpis dok se *DSA* može koristiti samo za digitalni potpis ali ne i za šifrovanje što je bila i ideja norme za digitalni potpis. Zbog toga je *DSA* komplikovaniji od *RSA* algoritma. Ipak, pokazano je da se *DSA* algoritmom mogu šifrovati podaci. *DSA* koristi *SHA-1* funkciju sažimanja i osnovna namena mu je stvaranje i provera digitalnog potpisa.
Tabela 5.1. daje pregled trajanja generisanih ključeva kao i stvaranja i provere potpisa za *RSA* i *DSA* algoritme na računaru.

Algoritam	Dužina ključa (bit)	Generisanje ključeva (ms)	Stvaranje potpisa (ms)	Provera potpisa (ms)
RSA	512	544,610	0,915	0,160
RSA	1024	1120,460	4,188	0,263
DSA	512	6,620	0,634	0,988
DSA	1024	17,870	1,775	3,397

Tabela 5.1. Trajanje generisanja ključeva i potpisa kao i provere potpisa za *RSA* i *DSA* algoritme

Izmerena vremena pokazuju da je generisanje ključeva znatno brže kod *DSA* algoritma. Stvaranje potpisa *DSA* algoritmom takođe je brže, ali provera potpisa *RSA* algoritmom je znatno brža.

S obzirom da se potpis stvara samo jednom, a proverava se više puta, brzina provere potpisa je važnije svojstvo. Ipak u nekim primenama gde je bitna brzina stvaranja potpisa *DSA* je prihvatljivije rešenje.

5.1.7. Digitalni sertifikat

Digitalni sertifikat je svedočenje koje potvrđuje da je vlasnik sertifikata u trenutku izdavanja sertifikata posedovao privatni ključ koji odgovara javnom ključu sadržanom u sertifikatu. Digitalni sertifikat je baziran na asimetričnoj kriptografiji odnosno kriptografiji javnog i tajnog ključa, kao i digitalnom potpisu koji dokazuje integritet samog sertifikata. Postoji nekoliko normi sertifikata kao što *X.509*, *PGP*, *SPKI/SDSI,* a najpoznatija je *X.509* norma pa se većinom za digitalni sertifikat podrazumeva *X.509* sertifikat.

5.1.7.1. X.509 sertifikat

Najpoznatija i najraširenija norma za definisanje digitalnog sertifikata je X.509 norma. X.509 je ITU-T norma za PKI sisteme. Izgrađen je na temeljima X.500 norme o servisima nad datotekama (engl. *directory devices*). X.509 norma propisuje koje informacije i na koji način se informacije zapisuju u sertifikat.

X.509 sertifikat je javni dokument koji sadrži skup bitnih informacija koje identifikuju osobu, organizaciju, računarski program, računar ili uopšte neki entitet digitalno potpisan od treće strane od poverenja, odnosno od sertifikacionog centra. Poverenje prema sertifikacionom centru i provera potpisa predstavlja autentifikaciju vlasnika sertifikata.

X.509 sertifikat služi u dve svrhe:

1 Javni ključ dostupan je svima i može se koristiti za šifrovanje podataka koje samo vlasnik sertifikata može dešifrovati privatnim ključem koji odgovara javnom ključu iz sertifikata. Isto tako, javni ključ koristi se za proveru digitalnog potpisa informacija koje je vlasnik sertifikata potpisao privatnim ključem,
2 Potvrđuje identitet vlasnika sertifikata u koji mogu imati poverenja svi koji veruju izdavaču sertifikata.

Tabela 5.2. daje popis informacija sadržanih u *X.509* sertifikatu.

Polje	Opis
Verzija	Verzija sertifikata (v1, v2, v3,..)
Serijski broj	Serijski broj sertifikata
Algoritam potpisa	Identifikator algoritma digitalnog potpisa
Izdavač	*X.500* prepoznatljivo ime izdavača
Period validnosti	Datum početka i datum isteka validnosti
Subjekt	*X.500* prepoznatljivo ime subjekta
Javni ključ	Identifikator algoritma i javni ključ subjekta
Jedinstveni identifikator izdavača	Uveden u verziji 2 zbog mogućnosti postojanja istog *X.500* imena izdavača
Jedinstveni identifikator subjekta	Uveden u verziji 2 zbog mogućnosti postojanja istog *X.500* imena subjekta
Proširenja	U verziji 3 uvedena su proširenja kao mogućnost uključivanja dodatnih atributa
Digitalni potpis	Vrednost potpisa

Tabela 5.2. Sadržaj *X.509* sertifikata

X.500 prepoznatljivo ime (engl. *distinguished name*), prema *X.500* normi, koristi se zbog toga što jedan entitet može imati više sertifikata. Osoba može imati više sertifikata koje koristi u različite svrhe pa se ne može samo po imenu razlikovati o kom sertifikatu se radi.

X.500 prepoznatljivo ime je po originalnoj definiciji globalno, ipak u praksi prepoznatljivo ime je lokalno i verovatno jedinstveno za sertifikacioni centar koji je potpisao sertifikat. U tabeli 5.3. su prikazana polja *X.500* prepoznatljivog imena.

Polje	Eng. naziv	Skraćenica	Opis
Ime	*Common Name*	*CN*	Ime koje se certificira
Organizacija ili tvrtka	*Organization or Company*	*O*	Ime je povezano s ovom organizacijom
Odjel	*Organization Unit*	*OU*	Ime je povezano s ovim odjelom
Grad	*City/Location*	*L*	Ime je locirano u ovom gradu
Država	*State/Province*	*ST*	Ime je locirano u ovoj državi
Kratica države	*Country*	*C*	Kratica države (*ISO* kod)

Tabela 5.3. Polja *X.500* prepoznatljivog imena

U verziji 2 *X.509* sertifikata uveden je jedinstveni identifikator izdavača i subjekta kako bi se moglo rukovati sa sertifikatima koji imaju identično *X.500* prepoznatljivo ime. Međutim, u većini slučajeva se preporučuje korišćenje jedinstvenih *X.500* imena i izbegavanje jedinstvenih identifikatora pa se verzija 2 *X.509* sertifikata retko koristi.

Verzija 3, uvedena 1996. godine najčešće se koristi, a omogućuje proširenja koja se mogu uključiti u sertifikat. Proširenje, između ostalog, može sadržati:

1. *Identifikator ključa potpisnika sertifikata* - omogućava identifikovanje javnog ključa koji odgovara privatnom ključu korišćenom za potpisivanje sertifikata,
2. *Identifikator ključa subjekta* -omogućava identifikacije sertifikata koji sadrže

određeni javni ključ,
3. *Upotreba ključa* -definiše namenu javnog ključa sadržanog u sertifikatu. Tako se može odrediti namena javnog ključa, što može biti npr. samo za autentifikaciju,
4. *Politika sertifikata*,
5. *Alternativno ime subjekta,*
6. *Alternativno ime izdavača sertifikata*.

5.1.7.2. Formati zapisa X.509 sertifikata

Norme za čuvanje *X.509* struktura u datoteke donosi udruženje za razvoj i određivanje Internet normi (engl. *Internet Engineering Task Force* -IETF). Za svaku *X.509* strukturu propisana je *ASN.1* sintaksa koja se zatim kodira prema *DER* (engl. *ASN.1 Distinguished Encoding Rules*) pravilima. *DER* kodirani zapis sertifikata je binarni zapis, a imena datoteka sa takvom vrstom zapisa obično se završavaju sa *.der*, *.crt* ili *.cer* nastavkom.

PEM je *Base64* kodiran *X.509* format certifikata. Imena datoteka koje sadržavaju *PEM* format zapisa certifikata obično završavaju s *.pem*, *.crt* ili *.cer* nastavkom. Datoteke čije ime se završava sa *.pem* nastavkom obično sadrže tekst, sertifikat, privatni ključ, a mogu sadržati i druge podatke. Slika 5.3. prikazuje sadržaj datoteke čije se ime završava sa *.pem* nastavkom. Datoteka sadrži sertifikat u *PEM* formatu sa dodatnim tekstom.

```
Certificate:

    Data:Version: 3 (0x2)Serial Number: 1 (0x1)Signature Algorithm:
        md5WithRSAEncryptionIssuer: O=SIGA, L=Zagreb, ST=Croatia, C=HR, CN=SIGA
        Certification AuthorityValidity
            Not Before: Jun 1 13:14:46 2006 GMT

            Not After : May 27 13:14:46 2007 GMTSubject: C=HR, ST=Croatia, O=SIGA,
        CN=siga.hrSubject Public Key Info:
            Public Key Algorithm: rsaEncryptionRSA
            Public Key: (1024 bit)
                Modulus (1024
                    bit):00:e2:23:fb:8b:4d:4e:b5:5f:5b:70:80:6c:7a:e8
                    :9e:cd:7a:dd:36:76:b3:40:4f:5c:16:a2:1f:5e:98:ac:
                    33:64:fb:62:e7:f7:4f:60:8b:70:0f:31:3a:b3:45:95:a
                    1:36:19:82:fb:db:7d:d7:a0:96:c7:2f:d6:ec:72:b3:83
                    :c9:0e:36:42:dc:e9:99:12:f4:9b:8a:08:dd:68:23:40:
                    f4:68:6f:39:57:ff:36:54:a9:db:93:71:9e:87:02:88:3
                    5:3d:37:e5:d8:55:55:c0:61:5c:e2:6e:27:d5:83:02:05
                    :2e:05:5b:0a:f3:c5:d9:82:77:05:d8:46:fa:70:79:cd
                Exponent: 65537 (0x10001)

        X509v3 extensions:X509v3 Basic Constraints:CA:FALSEX509v3 Subject Key
            Identifier:
           FE:4A:02:73:3A:AC:63:FA:73:3C:35:C4:70:36:8E:E5:CF:3C:51:96

    Signature Algorithm:
        md5WithRSAEncryption4a:fe:13:41:a6:7e:b0:8f:9d:7e:6a:3c:fe
        :78:95:60:91:af:31:8e:16:ac:7e:5b:0b:de:90:38:d9:24:5b:4f:
        d2:7d:ee:cb:1d:f0:f4:ac:73:28:e3:ab:0b:ad:68:c4:04:86:f1:f
        0:27:fd:b8:fc:a4:48:ad:db:04:3e:6d:29:03:39:d7:83:ff:48:4f
        :71:ce:15:19:16:46:bb:e7:59:98:b0:25:c2:a1:c2:94:6b:05:9e:
        e3:58:7e:9d:15:06:51:fa:09:27:dd:78:cd:42:c7:19:ca:c5:61:3
        3:26:f5:ed:96:6b:8d:4c:ed:05:aa:da:c2:be:32:54:d3:e3:14
-----BEGIN CERTIFICATE-----
MIICSDCCAbGgAwIBAgIBATANBgkqhkiG9w0BAQQFADBmMQ0wCwYDVQQKEwRTSUdBMQ8wDQYDVQQHEwZaYWdyZWIx
EDAOBgNVBAgTB0Nyb2F0aWExCzAJBgNVBAYTAkhSMSUwIwYDVQQDExxTSUdBIENlcnRpZmljYXRpb24gQXV0aG9y
aXR5MB4XDTA2MDYwMTEzMTQ0NloXDTA3MDUyNzEzMTQ0NlowQDELMAkGA1UEBhMCSFIxEDAOBgNVBAgTB0Nyb2F0
aWExDTALBgNVBAoTBFNJR0ExEDAOBgNVBAMTB3NpZ2EuaHIwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAOIj
+4tNTrVfW3CAbHrons163TZ2s0BPXBaiH16YrDNk+2Ln909gi3APMTqzRZWhNhmC+9t916CWxy/W7HKzg8kONkLc
6ZkS9JuKCN1oI0D0aG85V/82VKnbk3GehwKINT035dhVVcBhXOJuJ9WDAgUuBVsK88XZgncF2Eb6cHnNAgMBAAGj
LDAqMAkGA1UdEwQCMAAwHQYDVR0OBBYEFP5KAnM6rGP6czw1xHA2juXPPFGWMA0GCSqGSIb3DQEBBAUAA4GBAEr+
E0GmfrCPnX5qPP54lWCRrzGOFqx+WwvekDjZJFtP0n3uyx3w9KxzKOOrC61oxASG8fAn/bj8pEit2wQ+bSkDOdeD
/0hPcc4VGRZGu+dZmLAlwqHClGsFnuNYfp0VBlH6CSfdeM1CxxnKxWEzJvXtlmuNTO0FqtrCvjJU0+MU-----END
CERTIFICATE-----
```

Slika 5.3. *PEM* format sertifikata sa dodatnim tekstom

Datoteka koja predstavlja isti sertifikat može sadržati i samo *Base64* kodiran *X.509* format sertifikata odnosno podatke između *-----BEGIN CERTIFICATE-----* i *-----END CERTIFICATE-----* oznaka za početak i kraj sertifikata uključujući i te oznake. Ime takve datoteke obično se završava sa *.cer* ili *.crt* nastavkom.

5.1.8. Protokoli za sigurnu razmenu podataka

Najčešće korišćeni protokoli za sigurnu razmenu podataka su *SSL* (engl. *Secure Socket Layer*) i *TLS* (engl. *Transport Layer Security*). *SSL* i *TLS* protokoli osiguravaju tajnost, verodostojnost, integritet, neporecivost podataka i opcionalno autentifikaciju.

TLS protokol (engl. *Transport Layer Security*) razvijen je u okviru *IETF* (engl. *Internet Engineering Task Force*) 1999. godine. *TLS* protokol verzije 1 je nastao na temelju *SSL* protokola verzije 3 kojoj je vrlo sličan. Većina aplikacija podržava i *TLS* i *SSL* protokol.

SSL protokol detaljnije je opisan u nastavku.

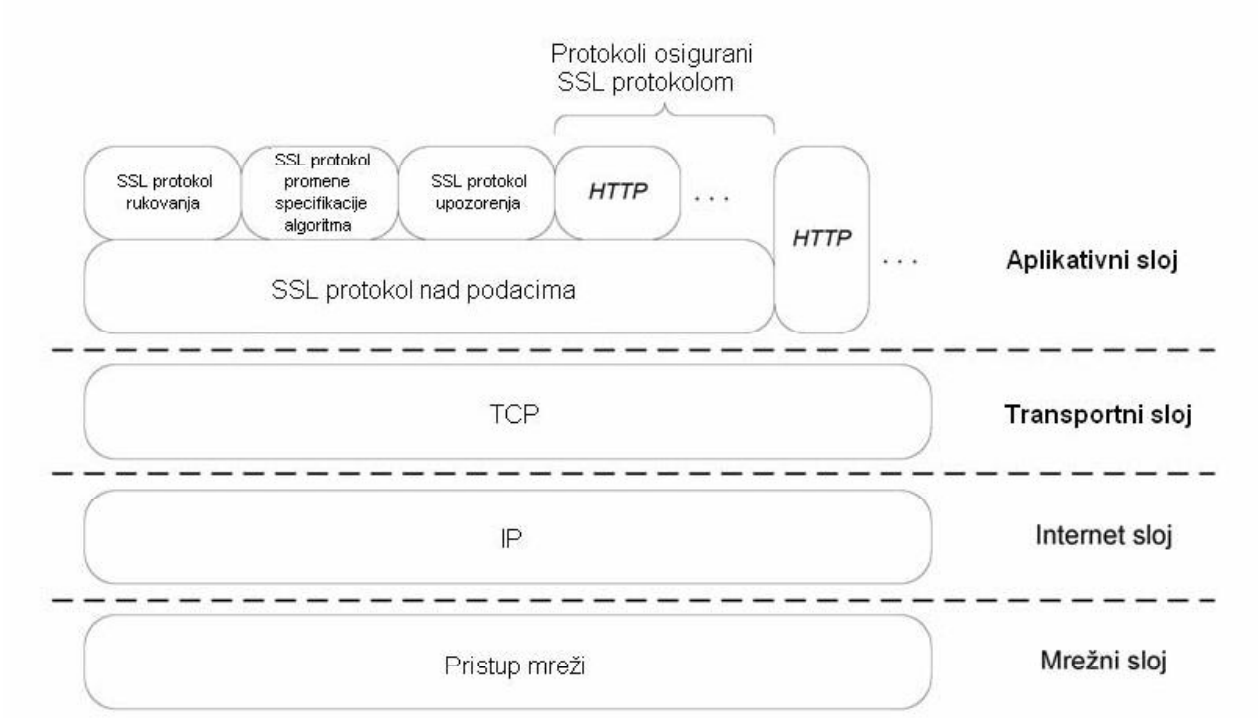
5.1.8.1. SSL protokol

SSL protokol je razvijen 1994. godine u okviru kompanije *Netscape Communications*. Primarni cilj *SSL* protokola je osigurati tajnu i pouzdanu komunikaciju dveju aplikacija u mrežnom okruženju. U *SSL* protokolu dogovaraju se šifarski algoritmi i simetrični ključ između dva učesnika u komunikaciji tako da osigurava tajni kanal koji može služiti za prenos ostalim protokolima višeg nivoa. Opcionalno *SSL* protokolom može se autentifikovati i prvi i drugi učesnik u komunikaciji. *SSL* protokol je trenutno u verziji 3. Verzija 1 *SSL* protokola nije bila izdata. Verzija 2 *SSL* protokola je izdata 1994. godine. Verzija 2 je imala nekoliko sigurnosnih slabosti tako da nije bila sigurna za komercijalnu upotrebu. U verziji 3 *SSL* protokola ispravljene su sigurnosne slabosti iz verzije 2. Osim toga u verziji 3 dodata je podrška za lanac sertifikata pa osim podrške za *RSA* šifarski algoritam dodata je podrška za još nekolicinu šifarskih algoritama. *SSL* u verziji 3 je vrlo brzo postao najpopularniji protokol za sigurnu razmenu podataka.

SSL protokol sastoji se od sledećih protokola:

1 *SSL* protokol nad podacima (engl. *SSL Record Layer*)
2 *SSL* protokol promene specifikacije algoritma (engl. SSL *Change Cipher Spec Protocol*)
3 *SSL* protokol upozorenja (engl. *SSL Alert Protocol*)
4 *SSL* protokol rukovanja (engl. *SSL Handshake Protocol*)

Protokol je podeljen u dva nivoa. Donji nivo *SSL* protokola, koja leži na prenosnom sloju, je *SSL* protokol nad podacima. Gornji nivo čine ostali *SSL* protokoli koji se enkapsuliraju u *SSL* protokol nad podacima. Slika 5.4. prikazuje položaj i slojeve *SSL* protokola u *TCP/IP* mrežnom okruženju. Na slici je prikazan i *HTTP* (engl. *Hyper Text Transfer Protocol*) protokol koji može biti osiguran *SSL* protokolom.



Slika 5.4. SSL sloj u TCP/IP mrežnom okruženju

SSL protokol nad podacima

SSL protokol nad podacima (engl. *SSL Handshake Protocol*) koristi se za enkapsulaciju ostalih protokola na višem nivou. Ovaj protokol prihvata podatke od aplikacija na višem nivou, fragmentira podatke koje treba poslati, enkapsulira podatke sa odgovarajućim zaglavljima i kreira objekt koji je šifrovan i može se proslediti prenosnom nivou.

SSL protokol promene specifikacije algoritma

SSL protokol promene specifikacije algoritma (engl. *SSL Change Cipher Spec Protocol*) služi za signalizaciju prelaza između promene šifarske strategije. Protokol se sastoji od jedne poruke koja sadrži vrednost 1. Poruku promene specifikacije algoritma šalje i klijentska i serverska strana čime se označava korišćenje upravo dogovorenog šifarskog algoritma i ključa.

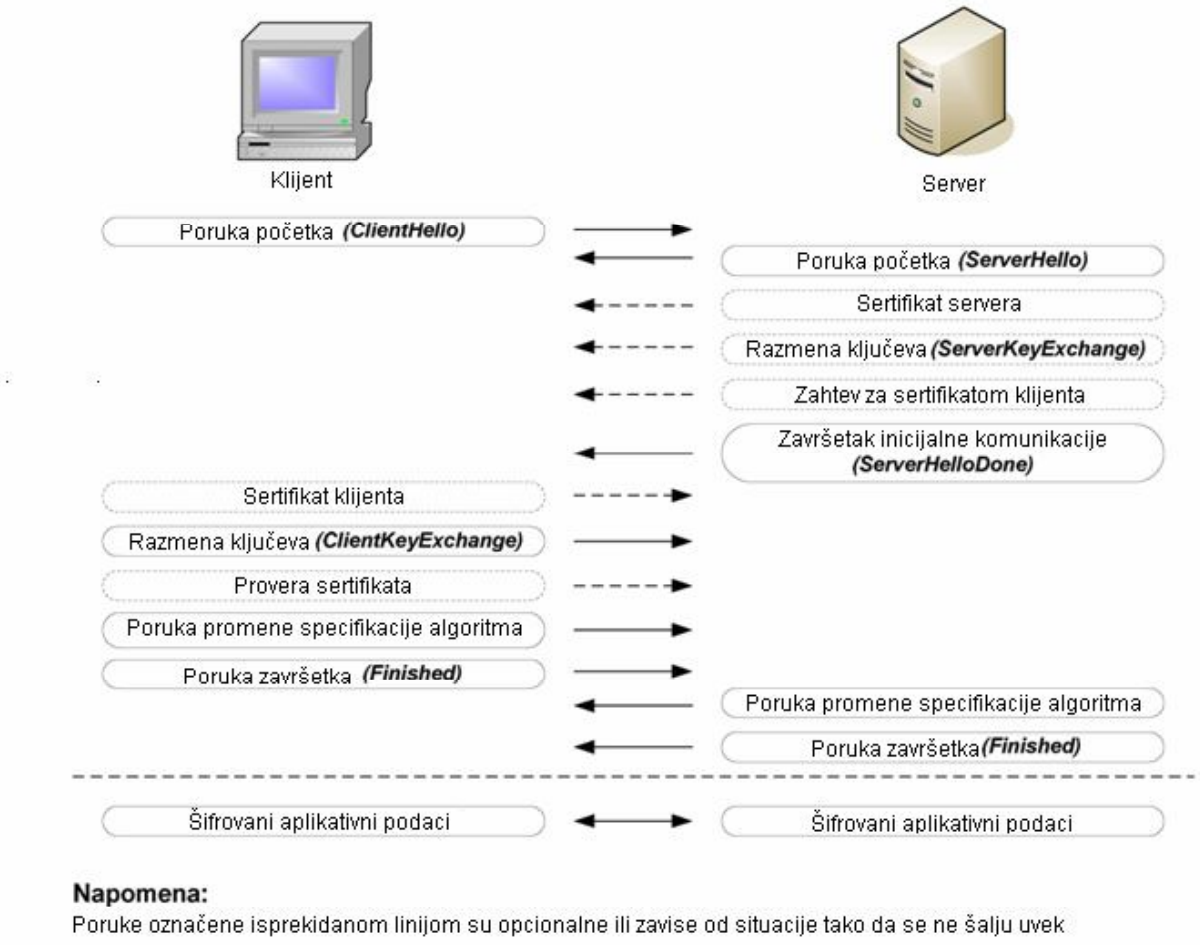
SSL protokol upozorenja

SSL protokol upozorenja (engl. *SSL Alert Protocol*) služi za obaveštavanje druge strane o ispravnosti prenosa podataka i funkcionisanju protokola. Poruka upozorenja sastoji se od nivoa upozorenja i opisa upozorenja. Poruku čine dva okteta. Prvi oktet

sadrži vrednost 1 u slučaju upozorenja ili vrednost 2 u slučaju greške.

SSL protokol rukovanja

SSL protokol rukovanja (eng. *SSL Handshake Protocol*) služi za iniciranje sesije između klijenta i servera kao i dogovara komunikacije koja uključuje dogovor šifarskih algoritama i ključeva. Unutar protokola rukovanja opcionalno se može izvršiti međusobna autentifikacija. Slika 5.5. prikazuje protokol rukovanja.



Slika 5.5. SSL protokol rukovanja

Poruka početka klijenta (engl. ClientHello)

Poruka sadrži verziju SSL protokola koju podržava, nasumični broj, identifikator sesije, popis podržanih šifarskih algoritama u poretku koji preferira klijent i popis podržanih algoritama za komprimovanje (engl. *zipped*).

Poruka početka servera (engl. ServerHello)

Server odgovara klijentu sa *handshake_failure* upozorenjem ili porukom početka. U poruci početka šalje jednu verziju protokola niže od one koju je tražio klijent i najvišu koju server podržava, nasumični broj, identifikator sesije, odabrani šifarski algoritam i odabrani algoritam za komprimovanje iz listi algoritama koje podržava klijent.

Sertifikat servera

Sertifikat servera je opcionalna poruka, a šalje se ako će server biti autentifikovan što je uobičajno. Vrsta sertifikata mora biti prikladna odabranom šifarskom algoritmu za razmenu ključeva i to je obično *X.509* sertifikat verzije 3.

Poruka razmene ključeva servera (engl. ServerKeyExchange)

Server šalje poruku razmene ključeva ukoliko nema prikladan sertifikat.

Poruka zahteva server za sertifikatom klijenta

Server koji se predstavio klijentu može porukom zahtevati od klijenta da pošalje svoj sertifikat.

Poruka servera o završetku inicijalne komunikacije (engl. ServerHelloDone)

Porukom o završetku inicijalne komunikacije server označava završetak poruke početka (engl. *ServerHello*). Nakon te poruke server očekuje odgovor klijenta.

Sertifikat klijenta

Poruka se šalje samo ukoliko je server tražio sertifikat klijenta. Ukoliko klijent nema sertifikat može obavestiti server o tome porukom upozorenja (engl. *no_certificate*).

Razmena ključeva klijenta

Poruka zavisi od odabranog postupka razmene ključeva. U slučaju *RSA* algoritma klijent generiše nasumičnu vrednost dužine 48 okteta i šifruje je javnim ključem servera. Generisana nasumična vrednost koristi se za generisanje glavnog ključa po određenom postupku.

Provera sertifikata

Poruka se koristi kako bi se eksplicitno zatražila provera sertifikata klijenta. Poruka se šalje samo u slučaju da je prethodno poslat sertifikat.

Poruka promene specifikacije algoritma

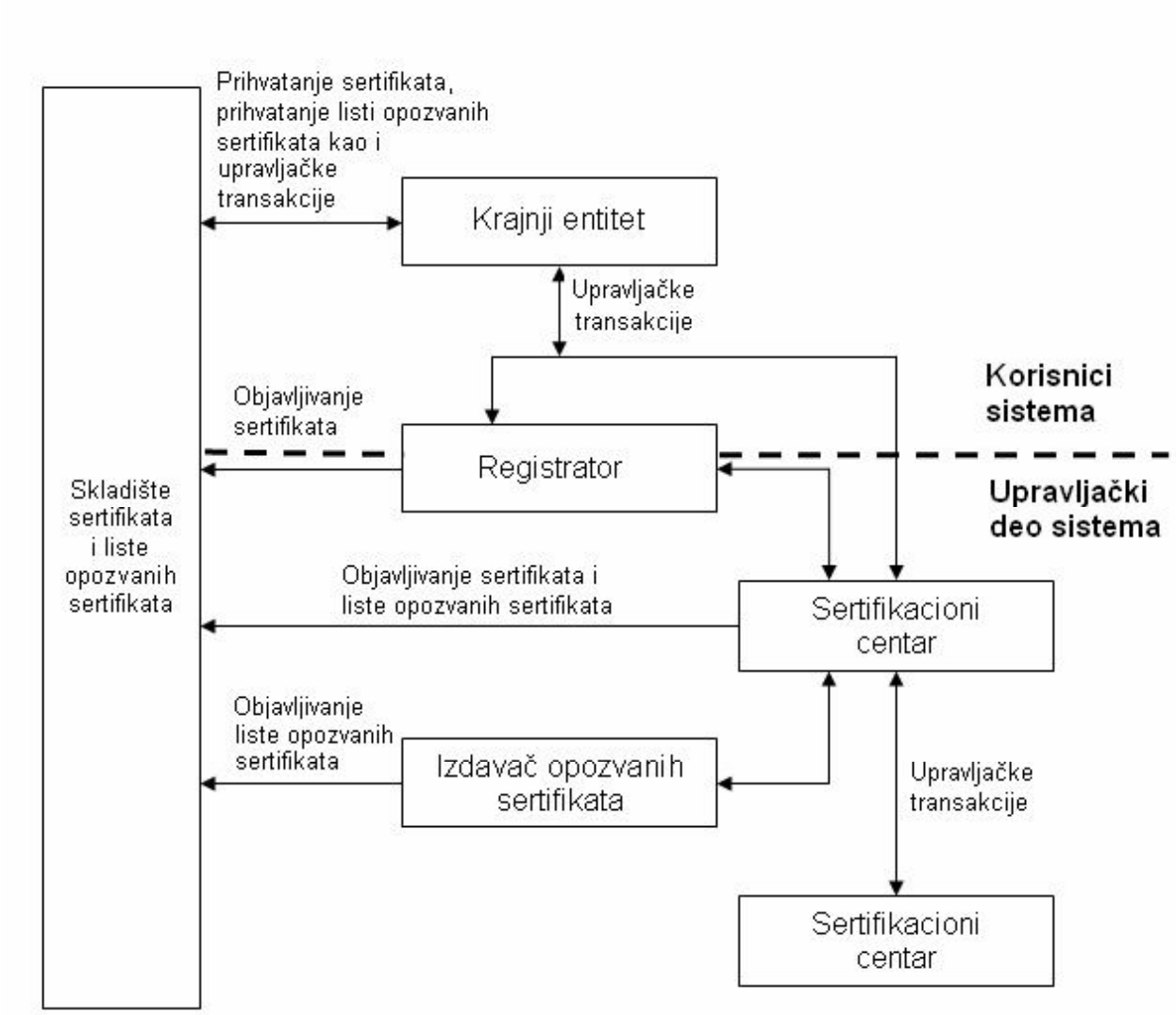
Poruka je nezavistan tip poruke *SSL* protokola i ustvari nije deo protokola rukovanja.

Poruka završetka (engl. Finished)

Poruka završetka predstavlja završetak protokola rukovanja. Poruka završetka šalje se uvek nakon poruke o promeni specifikacije algoritma kako bi se proverio uspeh procesa autentifikacije i razmene ključeva. Poruka završetka je prva poruka zaštićena sa upravo dogovorenim algoritmima i ključevima. Nakon poruke završetka obe strane mogu slati šifrovane podatke.

5.2. Infrastruktura javnog ključa

Infrastruktura javnog ključa (engl. *Public Key Infrastructure* -PKI) je skup tehnologija, protokola, normi i usluga koji zajedno omogućavaju sigurnu komunikaciju temeljenu na sistemu javnih ključeva preko nesigurnih mreža. Osnovni zadatak infrastrukture javnog ključa je neraskidivo povezivanje javnih ključeva sa korisnicima i provera identiteta. Infrastruktura javnog ključa treba pružiti tajnost i integritet informacija, sigurnu autentifikaciju učesnika komunikacije, pouzdanost vremena i datuma kao i formalno pravnu validnost. Slika 5.6. prikazuje osnovnu arhitekturu infrastrukture javnog ključa.



Slika 5.6. Osnovna arhitektura infrastrukture javnog ključa

Komponente infrastrukture javnog ključa su:

1 Krajnji entitet je korisnik *PKI* sertifikata koji ne mora nužno biti fizička ili pravna osoba već može biti i uređaj ili bilo koji entitet koji treba posedovati sertifikat,
2 Sertifikacioni centar je najvažnija komponenta infrastrukture javnog ključa,
3 Registracioni centar je opcionalna komponenta *PKI* sistema upravljana od sertifikacionog centra, a služi za izvršavanje određenih upravljačkih transakcija. Osnovna namena registracionog centra, ako postoji, je registracija krajnjih korisnika. Registracioni centar je i sam korisnik *PKI* sistema tako da ima svoj privatni i javni ključ,
4 Izdavač opozvanih sertifikata je opcionalna komponenta *PKI* sistema upravljana od sertifikacionog centra, a služi za objavljivanje liste opozvanih sertifikata (engl. *Certificate Revocation List* -CRL),
5 Skladište sertifikata i liste opozvanih sertifikata je sistem koji može biti distribuiran, a služi za čuvanje sertifikata i liste opozvanih sertifikata. Sistem osigurava dostupnost, odnosno uvid krajnjim korisnicima u sertifikate i listu opozvanih sertifikata.

Krajnji korisnici koji žele postati vlasnici sertifikata odnosno žele postati deo *PKI* sistema moraju imati sertifikat potpisan od sertifikacionog centra. Sigurnosna politika o izdavanju sertifikata (engl. *Certificate Policy* – CP) kao i pravila o izdavanju sertifikata (engl. *Certificate Practice Statement* – CPS) zavise od ustanova koje izdaju sertifikat. Jedan od načina izdavanja sertifikata može biti sledeći:

Korisnik generiše zahtev za potpis sertifikata (engl. *Certificate Signing Request* - CSR). Kod stvaranja zahteva za potpis sertifikata generiše se par javnog i privatnog ključa, a sam zahtev za potpis sertifikata sadrži informacije o identitetu korisnika i javni ključ korisnika. Korisnik je sam odgovoran za tajnost privatnog ključa. Nakon toga korisnik upućuje zahtev za potpis sertifikata ustanovi nadležnoj za registraciju sertifikata. Ustanova nadležna za registraciju sertifikata proverava identitet korisnika. Identitet korisnika, takođe zavisno od sigurnosne politike, može se proveriti na više načina na temelju čega sledi i klasa sertifikata odnosno poverenje u identitet.

Klase sertifikata zavisno od načina provere identiteta su sledeće:

1 Vlasnik sertifikata je identifikovan adresom elektronske pošte,
2 Vlasnik sertifikata je identifikovan podacima o identitetu koje je sam podneo,
3 Vlasnik sertifikata je identifikovan podacima o identitetu koji su provereni službenom ispravom,
4 Vlasnik sertifikata je identifikovan podacima o identitetu koji su provereni službenom ispravom i dodatno je napravljena fizička provera identiteta što može biti od provere slike do metoda biometrije poput otiska prsta.

Nakon provere identiteta zahtev za potpis sertifikata se potpisuje od strane sertifikacionog centra čime korisnik postaje deo *PKI* sistema i dobija sertifikat.

Neki sistemi ponekad pamte i privatni ključ čime se narušava privatnost, ali se dodatno sprečava pojava kriminala. Korisnik ima pravo pokrenuti pronalaženje javnog ključa, a čak i privatnog ukoliko sistem pamti i privatne ključeve korisnika. Korisnik

takođe može tražiti obnavljanje kao i opoziv sertifikata.

5.2.1. Sertifikacioni centar

Sertifikacioni centar (engl. *Certificate Authority* -CA) je polazna tačka poverenja. Uloga sertifikacionog centra jeste da bude treća strana od poverenja kojoj veruju učesnici u komunikaciji. Sertifikacioni centar izdaje sertifikate, garantuje njihovu verodostojnost, omogućava povlačenje sertifikata i uvid u nevažeće sertifikate i omogućava autentifikaciju vlasnika sertifikata.

Sertifikacioni centar poseduje svoj par ključeva pomoću kojih digitalno potpisuje podatke o identitetu korisnika zajedno sa javnim ključem korisnika što mu je primarna uloga.

Zavisno od sigurnosne politike, sertifikacioni centar prilikom izdavanja sertifikata može proveravati i posedovanje privatnog ključa koji odgovara javnom ključu korisnika koji će biti sadržan u sertifikatu ili čak može sam za korisnika generisati par privatnog i javnog ključa. Ukoliko u *PKI* sistemu postoji registracioni centar prepuštaju mu se zadaci vezani za registraciju korisnika.

Sertifikat iz više razloga može biti opozvan, bilo da je kompromitovan privatni ključ, bilo da informacije sadržane u sertifikatu više ne vrede ili nešto treće. U tom slučaju sertifikat se proglašava nevažećim i objavljuje se u listi opozvanih sertifikata koja je javno dostupna. Ukoliko u *PKI* sistemu postoji izdavač opozvanih sertifikata prepuštaju mu se zadaci vezani za opoziv sertifikata.

Pomoću javnog ključa sertifikacionog centra, kojem se veruje, može se proveriti integritet podataka navedenih u sertifikatu i samim time autentifikovati vlasnika sertifikata. Osim toga važno je proveriti validnost sertifikata što znači proveriti da li sertifikat vredi u datom periodu i proveriti da li je sertifikat opozvan.

Postoje dve vrste sertifikacionih centara, a to su vrhovni (engl. *root*) i ulančani sertifikacioni centar.

5.2.1.1. Vrhovni sertifikacioni centar

Vrhovni sertifikacioni centar nema izdavača sertifikata već on sam sebi izdaje sertifikat i potpisuje ga. Takav sertifikat je samo potpisani sertifikat (engl. *Self-Signed Certificate*). Prema tome taj sertifikat se nema gde proveriti i vrhovni sertifikat je polazna tačka poverenja odnosno njemu se veruje ili ne. Neki od poznatijih javnih sertifikacionih centara, koji su obično uključeni kao poverljivi na web-u , su *VeriSign*, *GlobalSign*, *Thawte*, *Entrust* i drugi.

5.2.1.2. Ulančani sertifikacioni centar

Sertifikacioni centar može izdati sertifikat drugom sertifikacionom centru koji postaje ulančani sertifikacioni centar. Kod provere sertifikata traži se prvi sertifikacioni centar kojem se veruje. Međutim, obično se ograniči poverenje u sertifikat na određenu dubinu lanca izdavača sertifikata kako bi se izbegao rizik od nepoverljivih sertifikacionih centara.

5.3. Autentifikacija

Autentifikacija (grč. *authentes* = autor) osobe ili uopšte entiteta predstavlja potvrdu identifikacije odnosno dokaz da je entitet stvarno taj za koga se predstavlja. Autentifikacija nije sama sebi svrha već se koristi za potrebe privatnosti, autorizacije kao osnova za neporecivost za napravljene akcije.

Svi principi autentifikacije temelje se na posedovanju neke tajne informacije poznate samo korisniku i eventualno sistemu autentifikacije. U interakciji sa učesnicima komunikacije koristi se tajna informacija na način koji dozvoljava drugoj strani da proveri posedovanje tajne informacije. Ukoliko druga strana mora znati tajnu prve strane ne sme je odati drugima. Sistemi za autentifikaciju najviše se razlikuju po načinu dokazivanja poznavanja tajne bez odavanja.

Najpoznatiji sistemi za autentifikaciju su *Kerberos* sistem koji koristi *Needham-Schroeder* protokol kao *PKI* sistem koji koristi asimetričnu kriptografiju.

Nepoverljivi sistemi autentifikacije su sistemi temeljeni na lozinkama. Lozinka je jednostavna tajna koju korisnik šalje primaocu. Lozinke su obično sačuvane na serveru u šifrovanom obliku ili u obliku sažetka. Osnovni problem sa sistemima baziranim na lozinkama je što lozinke vrlo lako postanu poznate trećim osobama koje se tada mogu lažno predstavljati tuđom lozinkom. Lozinke mogu postati poznate trećim osobama na više načina. Sam korisnik može podeliti lozinku sa trećim osobama. U mrežnom okruženju lozinka se može videti prilikom slanja mrežom. Isto tako na najrazličitije načine napadači mogu saznati lozinku. Lozinka je neprihvatljiv način autentifikacije i smatra se slabom autentifikacijom.

Jakom autentifikacijom nazivaju se sistemi koji koriste više od jednog faktora autentifikacije dok se sistemi koji koriste samo jedan faktor autentifikacije nazivaju slabom autentifikacijom.

Faktori autentifikacije mogu biti:

- nešto što osoba ili računar zna (*PIN* ili lozinka),
- nešto što osoba ili računar poseduje (pametna kartica ili šifarski token) ili
- nešto što predstavlja osobu (otisak prsta ili slične metode biometrije).

Dva faktora autentifikacije (engl. *Two-Factor Authentication*) podrazumevaju dva odvojena načina za dokazivanje identiteta. Kod većine ostvarenja jedan od faktora je

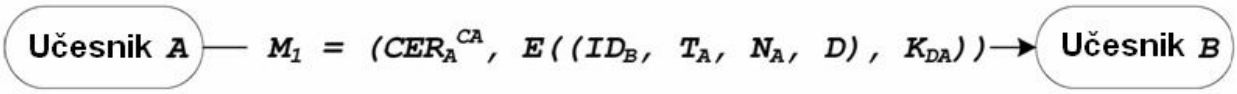
znati nešto, a drugi od faktora je posedovati nešto ili imati neko biometrijsko obeležje.

5.3.1. X.509 autentifikacioni protokoli

X.509 temelji se na *X.509* sertifikatima i asimetričnoj kriptografiji. *X.509* preporučuje tri protokola za autentifikaciju: protokol sa jednom porukom (engl. *One-Way Protocol*), protokol sa dve poruke (engl. *Two-Way Protocol*) i protokol sa tri poruke (engl. *Three-Way Protocol*). *X.509* autentifikacioni protokoli omogućavaju sigurnu autentifikaciju između dva učesnika komunikacije. U protokolima se podrazumeva da učesnici komunikacije mogu na siguran način doći do sertifikata i proveriti validnost sertifikata kao i da imaju poverenje u sertifikacione centre koji su potpisali sertifikate.

5.3.1.1. X.509 protokol sa jednom porukom

Protokol sa jednom porukom osigurava integritet sadržaja koji se šalje učesniku B. Upotrebom vremenske oznake sprečava se napad ponavljanjem poruke.



Slika 5.7. X.509 protokol s jednom porukom

U protokolu sa jednom porukom obavljaju se sljedeći koraci:

1. Kada učesnik A želi uspostaviti komunikaciju sa učesnikom B on čini sledeće:
   - Generiše nasumični broj N_A,
   - Oblikuje vremensku oznaku T_A koja se sastoji od dve komponente: početnog vremena važenja poruke i trajanja validnosti oznake,
   - Iz sertifikata učesnika B (CER_B^{CB}) saznaje javni ključ učesnika B koji je potpisan od sertifikacionog centra C_B,
   - Oblikuje četvorku (ID_B, T_A, N_A, D), gde je D komponenta sa podacima koja može biti šifrovana ključem K_{EB}.
   - Šifruje oblikovanu četvorku svojim privatnim ključem K_{DA} i šalje poruku CA:

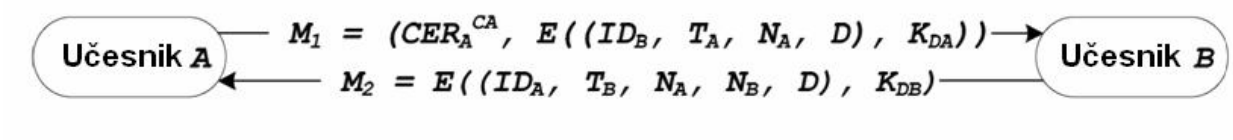
$$M_1 = (CER_A^{CA}, E((ID_B, T_A, N_A, D), K_{DA}))$$

   gde je CER_A^{CA} sertifikat učesnika A koji je potpisao sertifikacioni centar CA.

2. Kada učesnik B primi poruku M_1 on čini sledeće:
   - Iz poruke izdvaja sertifikat učesnika A, proverava ga, i iz njega saznaje javni ključ učesnika A (K_{EA}),
   - Dešifrovanjem pomoću javnog ključa KE_A dobija četvorku: $(ID_B, T_A, N_A, D) = D(E((ID_B, T_A, N_A, D), K_{DA}), K_{EA})$,
   - Na temelju ID_B utvrđuje da je poruka stvarno upućena njemu. Na temelju vremenske oznake T_A utvrđuje da je poruka "sveža",
   - Dešifruje svojim privatnim ključem komponentu sa podacima D ako je bila šifrovana.
   - Dodatno može uporediti dobijeni N_A sa sačuvanim nasumičnim brojevima iz prethodnih poruka kako bi ustanovio da poruka nije ponovljena.

5.3.1.2. X.509 protokol sa dve poruke

Protokol sa dve poruke se nadovezuje na protokol sa jednom porukom, a dodaje se odgovor učesnika B. Na temelju odgovora učesnika B utvrđuje se da je upravo učesnik B, a ne neki napadač, odgovorio na prvu poruku. Upoređivanjem vremenske oznake u drugoj poruci sprečava se napad ponavljanjem druge poruke.



Slika 5.8. X.509 protokol sa dve poruke

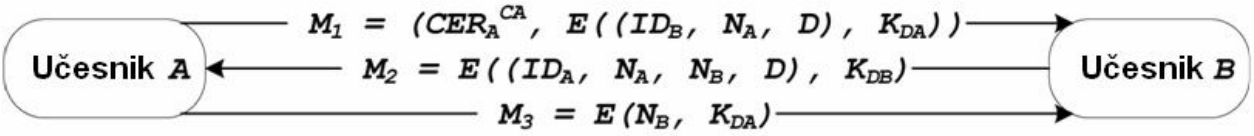
Prema tome protokol sa dvije poruke se nastavlja na protokol sa jednom porukom tako da se obavljaju sledeća dva koraka:

3. Učesnik B čini sledeće:
   a. Generiše svoj nasumični broj NB,
   b. Generiše vremensku oznaku TB,
   c. Oblikuje petorku (IDA, TB, NA, NB, D), gde je D komponenta sa podacima koja može biti šifrovana ključem KEA,
   d. Šifruje oblikovanu petorku upoređivanjem svog privatnog ključa KDB i šalje učesniku A poruku:
   $$M_2 = E((ID_A, T_B, N_A, N_B, D), K_{DB}).$$
4. Kada učesnik A primi poruku $M2$ on čini sledeće:
   a. Pomoću javnog ključa KEB dešifruje poruku $M2$ i dobija:
   $$(ID_A, T_B, N_A, N_B, D) = D(E((ID_A, T_B, N_A, N_B, D), K_{DB}), K_{EB})$$
   b. Na temelju IDA utvrđuje da je poruka stvarno upućena njemu. Na temelju vremenske oznake TB utvrđuje da je poruka "sveža",
   c. Dešifruje svojim privatnim ključem KDA komponentu sa podacima D ako je bila šifrovana,
   d. Dodatno može uporediti dobijeni NB sa sačuvanim nasumičnim brojevima iz prethodnih poruka kako bi ustanovio da poruka nije

ponovljena.

5.3.1.3. X.509 protokol sa tri poruke

Protokol sa tri poruke se nadovezuje na protokol sa dve poruke, sa razlikom što se ne koriste vremenske oznake u svim porukama, a učesnik *A* vraća treću poruku učesniku *B*.



Slika 5.9. X.509 protokol sa tri poruke

Protokol sa tri poruke obavlja se na isti način kao i protokol sa dve poruke s tim da se pri odvijanju prethodnih četiri tačke ignorišu vremenske oznake. Vrednosti vremenskih oznaka postavljaju se na nulu (*TSA= TSB = 0*). Protokol se nastavlja tako da se obavljaju sledeća dva koraka:

5. Učesnik *A* čini sledeće:
   - Upoređuje dobijeni *NA* iz poruke *M2* sa izvornom vrednošću i utvrđuje da je poruka *M2* odgovor na *M1*,
   - pomoću ključa *KDA* šifruje dobijeni *NB* i šalje poruku:

$$M_3 = E(N_B, K_{DA})$$

6. Po prijemu poruke *M3* učesnik *B* čini sledeće:
   - Pomoću ključa *KEA* dešifruje poruku *M3* i dobija:

$$N_B = D(E(N_B, K_{DA}), K_{EA})$$

   - Upoređuje dobijeni *NB* iz poruke *M3* sa izvornom vrednošću i utvrđuje da je *M3* odgovor na *M2*.

Zaključak

U ovom magistarskom radu detaljno je obrađena tema pametnih kartica kako kroz osnovna znanja tako i kroz sigurnost koja predstavlja najvažnije svojstvo. Današnje društvo evoluira u informaciono društvo. Tehnologija postaje alat u službi informacije, a informacija znanje, moć i novac. Sigurno čuvanje informacija i zaštita protoka informacija postaju nužnost i neophodnost. Kao idealno rešenje javlja se pametna kartica. Njen visok nivo sigurnosti čini informacije dostupne svakom, na siguran način čuva informacije i omogućuje da se procesiranje podataka izvodi u sigurnom okruženju.

Kao i na drugim tehničkim područijima i kod pametnih kartica postoje standardi koji opisuju njihove osobenosti. Nedostatak standarda prvih godina razvoja ogledala se u nekompatibilnosti kartica pojedinih proizvođača, iako se situacija poslednjih godina poboljšala. Opisani su najprihvaćeniji segmenti standarda ISO 7816 i EMV specifikacije. EMV standard temeljen je na ISO setu standarda za čip-kartice i uređaje za njihovo isčitavanje. EMV sprecifikacije su ustvari implementacioni orijentisani podskup ISO 7813-3 standarda.

Sigurnost se mora uzeti u obzir sa tačke gledišta ukupnog, opšteg sistema. Java Card platforma je izgrađena na vrhu platforme, koju sačinjava hardver kartice, izvorni operativni sistem, terminal sa kojim kartica komunicira i sistem u pozadini. Opisano poglavlje ističe deo sigurnosti kartične platforme kroz sigurnost Java Card platforme.

Sigurna autentifikacija i tajnost informacija su najvažnija pitanja koje se razmenjuju. Najčešći oblik autentifikacije je autentifikacija gde se korisnik predstavlja korisničkim imenom i lozinkom. Takva autentifikacija međutim ne garantuje veliku sigurnost. Pretnju kao stanje ili događaj koji može naškoditi osoblju, mreži i računarskim resursima u obliku uništavanja, razotkrivanja ili modifikacije podataka, uskraćivanja usluge, prevare i zloupotrebe moguće je kombinovanjem različitih navedenih mehanizama svesti na minimum.

Pravci razvoja

Kao što su lični računari postali deo svakidašnjice, pametne kartice su postale nešto što se samo po sebi podrazumeva i omogućuje aktivno prisustvo u digitalnom svetu. Neke prognoze za budućnost postaju sve više pregledne. Šifarske operacije će se izgubiti u infrastrukturi, kompleksnost kriptografskih upravljanja ključevima će biti skriveni od korisnika. Novi protokoli će postati praktični, biće omogućeni novi načini poslovanja. Sa primernim planiranjem može se većina računanja i vođenja dnevnika preneti sa pametne kartice na lični računar koji ima veće računske i memorijske sposobnosti. Manje je očita budućnost do nekih drugih vidika kao što su npr. društveni ekonomski i politički. Može se zamisliti samo da se uskoro neće moći više sesti u auto i voziti, a da nas pri tom sve vreme ne nadzire pametna kartice (radarske kontrole uopšte više neće biti potrebne).

Zbog novih napada će izdavači biti prisiljeni svakih nekoliko godina izraditi nove verzije pametnih kartica. Kod toga će naručioci sistema pametnih kartica morati imati aktivniju ulogu nego ikada do sada jer samo obrazovan naručilac će znati šta može zahtevati i šta može za svoj novac dobiti.

Literatura

1. www.smartcard.co.uk/tutorials/sct-itsc.pdf, *Tutorial - Introduction to Smart Cards,*
2. ccrma.stanford.edu/~jhw/bioauth/andre/SmartCardTutorialNov01.pdf
3. www.cardwerk.com/smartcards/smartcard_overview.aspx
4. www.javaworld.com/javaworld/jw-07-1999/jw-07-javacard.html
5. www.iec.org/online/tutorials/smartcard/
6. www.gnupg.org/(en)/howtos/card-howto/en/smartcard-howto.html
7. www.citi.umich.edu/projects/smartcard/smartcard_seminar/ , *Smartcard Programming Seminar. (Or how to hack M-Card).*
8. www.citi.umich.edu/projects/smartcard/smartcard_seminar/6.html , *What is in smartcard? - Smartcard Directory Structure.*
9. people.cs.uchicago.edu/~dinoj/smartcard/security.html, *An Overview of Smart Card Security*
10. www.clipclip.org/clips/detail/19239/java-card-tutorial , *Java Card Tutorial*
11. www.linux-tutorial.info/HOWTO/Smart-Card-HOWTO , *Linux Knowledge Base and Tutorial,*
12. www.smartcardalliance.org/articles/2007/01/18/smart-card-alliance-brings-advances-in-chip-technology , *Smart Card Alliance Brings Advances in Chip Technology to RSA*
13. www.iec.org/online/tutorials/acrobat/smartcard.pdf , *Smart Cards in Wireless Services. Definition.*
14. ersads99.di.fc.ul.pt/tutorial.html ,
15. www.codeproject.com/netcf/pocketscardfmwk.asp , *Smartcard Framework for .NET Compact Framework - The Code Project ,*
16. www.devarticles.com/c/a/Cplusplus/Writing-a-Smart-Card-Library/ , *Writing a Smart Card Library,*
17. www.linux-tutorial.info/modules.php?name=Howto&pagename=Smart-Card-HOWTO/classification.html , *Linux Knowledge Base and Tutorial*
18. ec2007.feri.uni-mb.si/tutorials.htm, *IWSSIP 2007 Conference and EC-SIPMCS 2007 Conference,*
19. www.infopeople.org/resources/security/users/smart_cards.html , *Library Computer and Network Security: User Security - Smart Cards*
20. www.mycplus.com/cplus.asp?CID=78 , *Using MFC & WinScard API to write Smard Card Applications- MYCPLUS,*
21. www.mycplus.com/tutorial.asp?TID=303 , *Implementation of Smart Card Library :: Using MFC & WinScard API ,*
22. www.microexpert.com/?page=training, *Smart Card Tutorial,*
23. www.iaik.tugraz.at/aboutus/people/poschkc/other%20stuff/smartcard.php, *Iinks to smartcard related information on the web,*
24. www.cardwerk.com/smartcards/smartcard_applications.aspx, *smart card solutions for banking, logical access control,*
25. www.extremetech.com/category2/0,1695,16359,00.asp, *To learn about smart cards the easy way,*

26. otavo.com/quests/java-card-tutorial , *Java Card Tutorial | Otavo - The Intention Engine*,
27. www.cs.auckland.ac.nz/~pgut001/tutorial/ , *Encryption and Security Tutorial*,
28. www.2000trainers.com/windows-2000/configure-hardware-devices/, *Configuring Hardware Devices · Tutorial · 2000Trainers.com*,
29. www.usasmartcard.com/ , *Smart Card - Home*
30. www.aidia-i.ba/activcard/smartcard/,
31. os2.zemris.fer.hr/stari_web/kartice/index.html
32. www.bug.hr/izdanja/tekst.asp?id=71958
33. www.span.hr/proizvodi_i_usluge/ActivCard.htm
34. www.case.hr/smartcard2005/anketa.htm
35. www.trend.hr/clanak.aspx?BrojID=29&KatID=5&ClanakID=353 , *:: INFOTREND::*
36. si.exportpages.com/auswahlc/66566/Pametne%20kartice/1/0/data.html, *Pametne kartice - Exporter, Manufacturer & Suppliers*,
37. hr.zone-h.org/index.php?option=com_content&task=view&id=139&Itemid=9
38. www.spica.hr/verticals/kontrolapristupaSeminarBiometrijaIPametneKartice.html
39. www.jantar.si/products/smartcards/hr.php
40. https://online.bancaintesabeograd.com/help
41. www.tutorial-reports.com/wireless/rfid/news.php?, *RFID | Tutorial-Reports.Com*
42. portal.acm.org/citation.cfm?id=518247&dl=ACM&coll=portal , *Java Card Technology for Smart Cards*
43. www.codeproject.com/smartclient/ , *The Code Project - Free Source Code and Tutorials*
44. findarticles.com/p/articles/mi_zdext/is_200202/ai_ziff22209 , *Smart Card Technology: A Tutorial ExtremeTech - Find Articles*
45. www.cs.rice.edu/~dwallach/courses/comp527_f2002/SmartCard.html, *Rajnish's Smart Card Tutorial*
46. medi-smart.com/tutorials.htm, *Medical Patient Simulators & Tutorials*
47. www.visualbuilder.com/viewnews.php?id=54134&news_id=686, *New Dekart Logon for Citrix ICA Client to Ease Administrators*
48. www.wrankl.de/ , *Homepage of Wolfgang Rankl*
49. power.elecdesign.com/tutorial/index.cfm?action=Tutorial&tutorialID=26, *Tutorial - PC Card Power Management ICs*