SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1445

SIGURNOSNI MEHANIZMI ZA RAD S PAMETNIM KARTICAMA

Aleksander Radovan

Zagreb, rujan 2004.

Sadržaj

1. Uvod

Pametna kartica (*engl. smart card*) je plastična kartica koja u sebi ima ugrađen integrirani sklop (*engl. integrated circuit chip*), a dimenzije su joj iste kao i kod kreditnih kartica. Glavne funkcije kartice su autentifikacija, pohranjivanje podataka, vrednovanje (*engl. validation*), te mehanizam samozaključavanja. Otporna je na vanjske napade i ne ovisi o potencijalno ranjivim vanjskim resursima. Baš zbog tih svojstava se pametne kartice često koriste u različitim aplikacijama koje zahtijevaju visoku razinu sigurnosne zaštite i autentifikacije. Na primjer, pametne kartice mogu služiti za identifikaciju vlasnika kartice, kao zdravstvena kartica, ili kao kreditna/debitna bankovna kartica koja omogućava financijske transakcije. Sve te aplikacije koriste osjetljive podatke spremljene na kartici kao što su biometrijski podaci o vlasniku, medicinski podaci, kriptografski ključevi za autentifikaciju itd. Uskoro će uobičajene kartice s magnetskom trakom biti potpuno zamijenjene i integrirane unutar jedne multiaplikacijske pametne kartice koja se u industriji pametnih kartica naziva još i "elektronički novčanik". Neprestanim unaprijeđivanjem sigurnosnih svojstava pametna kartica postaje vrlo pouzdan medij za pohranjivanje najosjetljivijih podataka čime se smanjuje broj argumenata za sumnju u njezinu isplativnost.

Sigurnost pametnih kartica se razmatra iz tri različita aspekta. U prvom se opisuje fizička struktura kartice i način na koji je provedena zaštita tijekom životnog ciklusa pametne kartice. Drugi opisuje način na koji je provedena zaštita podataka logičkim kontrolama nad datotekama koje se nalaze na kartici. Treći aspekt opisuje kako kartica pruža sigurnosno i autentifikacijsko okruženje za aplikacije pomoću proceduralnih operacija i mehanizama. Nakon toga se u radu opisuju neke tehnike narušavanja sigurnosti i napada na pametne kartice.

Postojanje niza načina ugrožavanja sigurnosti pametnih kartica ne znači da pametna kartica nije sigurna. Napadi da sigurnosne sustave bilo koje vrste nisu novost. Nigdje ne postoji 100%-tna sigurnost, pa tako ni kod pametnih kartica. Glavna odluka o tome da li je sustav siguran ili ne, ovisi o ispunjavanju sigurnosnih zahtjeva sustava. Nadalje, većina napada na pametne kartice se mogu klasificirati u tri klase, što znači da je cijena probijanja sustava puno veća nego njegova vrijednost, ili je potreban vrlo dugi vremenski period da bi se probili kriptografski algoritmi. No, kako tehnologija vrlo brzo napreduje, proizvođači su primorani stalno ažurirati sigurnosna svojstva svojih produkata.

Postoji nekoliko vrsta pametnih kartica, od običnih memorijskih kartica koje nemaju svojstvo procesiranja podataka, do kartica s procesorom i kriptografskim procesorom koje proširuju područje upotrebe kartice, ali i razinu sigurnosti. Najnovija vrsta kartica su beskontaktne kartice koje, osim izbjegavanja problema s kontaktima na prihvatnim uređajima, donose i niz prednosti kojima postižu veću popularnost kod krajnjih korisnika.

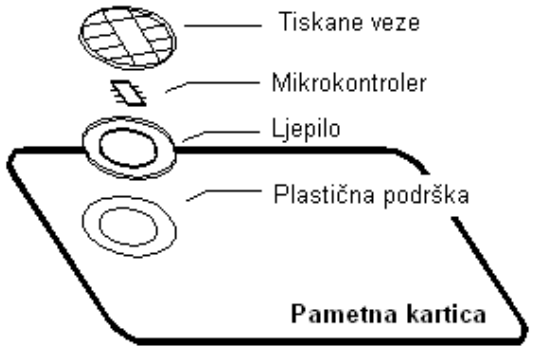
Sigurnosni mehanizmi su osnovna sredstva kojima se postiže vrlo visoka razina sigurnosti kod pametnih kartica. Oni se temelje na vrlo kompleksnim matematičkim proračunima koje vrlo teško mogu izračunavati obična računala bez specijaliziranog sklopovlja. U mehanizme se ubrajaju algoritmi kriptiranja, funkcije sažimanja, te algoritmi za ramjenu ključeva.

Vrlo visoka razina sigurnosti kod pametnih kartica je postignuta algoritmima koji podržavaju duljine ključeva od nekoliko tisuća bitova (na primjer, 2048 ili 4096). No, ubrzani razvoj računalne snage primorava autore kriptografskih algoritama da konstantno povećavaju razinu sigurnosti algoritama kako bi izbjegli ugrožavanje sigurnosti tehnologije koja ih koristi. Zbog sve češćih napada na prihvatne uređaje za pametne kartice (čitače na kojima su neovlašteno ugrađeno skopovlje koje omogućava krivotvorenje kartice i krađu PIN-a) nastoje se proizvesti kartice koje minimalno ovise o vanjskim resursima. Prvi korak planiranja u tom smjeru je tzv. super pametna kartica (*engl. Super Smart Card*) koja osim vlastitog napajanja sadržava ekran, tipkovnicu itd., što je veliki korak naprijed na području sigurnosti, ali i veliki udarac potencijalnim kriminalcima koji nikad ne spavaju i neprestano ugrožavaju postojeću sigurnost pametnih kartica i sličnim tehnologija.

2. Fizička struktura i životni ciklus kartice

2.1. Fizička struktura pametne kartice

Fizička struktura pametne kartice je specificirana standardom ISO (*engl. International Standards Organization*) 7810, 7816/1 i 7816/2. Struktura se sastoji od tri osnovna dijela: plastične kartice (dimenzija 85,80 mm x 53,98 mm x 0.80 mm), sklopa s integriranim vezama zajedno s tiskanim vezama (*engl. printed circuits*) koji su ugrađeni na kartici. Slika 2.1. opisuje fizičku strukturu kartice.



Slika 2.1: Fizička struktura pametne kartice

Tiskane veze su u skladu s ISO 7816/3 standardom i imaju pet komunikacijskih točaka za napajanje i podatke. Veze su hermetički fiksirane u udubini kartice i povezane s integriranim sklopom, popunjene vodljivim materijalom, te zatvorene uz izbočene kontakte. Tiskane veze štite sklop s integriranim vezama od mehaničkih naprezanja i statičkog elektriciteta. Komunikacija sa sklopom se ostvaruje preko kontakata koji prekrivaju tiskane veze.

Mogućnosti pametne kartice su određene integriranim sklopom. Obično se vezni sklop sastoji od mikroprocesora, memorije za čitanje (*engl. Read Only Memory - ROM*), nestatičke memorije sa slučajnim pristupom (*engl. Random Access Memory - RAM*) i električki izbrisive programirljive memorije za čitanje (*engl. Electrically Erasable Programmable ROM - EEPROM*) koja zadržava podatke i nakon prestanka napajanja. Integrirani sklopovi su izgrađeni od silikona koji nije mehanički fleksibilan zbog čega se lako lomi, pa se izrađuje u veličini svega nekoliko milimetara.

Fizičko sučelje koje omogućuje izmjenu podataka između integriranog sklopa i prihvatnog uređaja za karticu (*engl. Card Acceptance Device – CAD*) je ograničeno na brzinu komuniciranja od 9600 bita po sekundi. Komunikacijska linija je bidirekciona serijska linija koja je u skladu sa standardom ISO 7816/3. Sve podatke koji se razmjenjuju kontrolira centralna procesna jedinica koja se nalazi unutar integriranog sklopa. Naredbe i podaci se šalju prema sklopu, a on odgovara statusnim riječima (*engl. status words*) i izlaznim podacima koji se stvaraju na temelju tih

naredbi i ulaznih podataka. Informacije se šalju istovremeno samo u jednom smjeru (*engl. half duplex*). Taj protokol zajedno s restrikcijom brzine slanja podataka spriječava napadanje podataka koji se nalaze na kartici.

Veličina, debljina i otpornost na savijanje pametne kartice su dizajnirani tako da zaštite karticu od fizičkog uništavanja, ali ujedno i ograničavaju količinu memorijskih i procesorskih resursa koji se mogu ugraditi na kartici. Kao rezultat toga, pametne kartice rade uvijek s nekim vanjskim perifernim uređajima. Na primjer, potreban je uređaj koji prima korisničke podatke i obrađuje ih, posjeduje informacije o trenutnom vremenu i datumu, osigurava napajanje itd. Ova ograničenja mogu degradirati sigurnost pametnih kartica u nekim okolnostima, pogotovo ako su ti vanjski uređaju nepovjerljivi i nepouzdani.

2.2. Životni ciklus pametne kartice

U svakoj pametnoj kartici postoji operacijski sustav kojeg određuju podaci kao što su identifikacijski broj proizvođača (*engl. manufacturer identification number*), tip komponente, serijski broj, informacije o profilu itd. Najvažniji su sigurnosni ključevi koje sustavsko područje sadrži, kao što su ključ proizvođača (*engl. manufacturer/fabrication key*) i ključ personalizacije. Sve te informacije moraju biti sakrivene i nedostupne drugim subjektima. One se najčešće nalaze unutar posebnih datoteka kojima mogu pristupiti samo ovlaštene osobe.

Dakle, od proizvođača, do poslužitelja aplikacija, pa sve do vlasnika kartice, proizvodnja pametne kartice je podijeljena na nekoliko različitih faza. U svakoj od faza su ograničenja kod transfera i pristupa podacima različita, tj. povećavaju se u svrhu zaštite različitih područja na pametnoj kartici. Postoji pet različitih faza kod uobičajenog životnog ciklusa pametne kartice koje se objašnjavaju u nastavku rada.

2.2.1. Faza proizvodnje

Ova faza je određena proizvođačima sklopova. Integrirani sklop izgrađen od silikona se u ovoj fazi proizvodi i testira. Ključ proizvođača se dodaje da bi zaštitio sklop od neovlaštene modifikacije sve do ugradnje u plastičnu karticu. Ključ je jedinstven za svaki sklop, a dobiven je derivacijom glavnog ključa proizvođača. Ostali podaci proizvođača se zapisuju u sklop na kraju ove faze. Nakon toga je integrirani sklop spreman za dostavu proizvođačima zaštićen ključem proizvođača.

2.2.2. Pripremna faza personalizacije

Ova faza je određena proizvođačima kartica, a sastoji se od ugradnje integriranog sklopa u karticu. Izrađuju se veze između sklopa i tiskanih veza nakon čega je kartica spremna za testiranje. Dodatna sigurnost se postiže zamjenom ključa proizvođača s ključem personalizacije. Nakon toga se zapisuje personalizacijska brava V_{per} (*engl. personalization lock*) koja označava kraj zapisivanja podataka koji se odnose na

pripremnu fazu personalizacije, i koja štiti sklop od daljnje modifikacije personalizacijskog ključa. Personalizacijskom bravom se onemogućava izvođenje instrukcija koje pristupaju fizičkoj memoriji, pa je pristup kartici ograničen samo na korištenje logičkog memorijskog adresiranja. Brava je predstavljenja zapisom u datoteci na kartici, odnosno njenom zaglavlju. Logičko adresiranje čuva sustav i štiti rezervirana područja na sklopu od neovlaštene modifikacije.

2.2.3. Faza personalizacije

Ova faza je usko vezana s izdavačima kartica, a sastoji se od završavanja strukture logičkih podataka. Sadržaj podatkovnih datoteka i podataka vezanih uz aplikaciju se zapisuju na karticu. Osim tih podataka pohranjuju se još informacije o identitetu korisnika kartice, PIN i deblokacijski PIN. Na kraju se zapisuje i brava korištenja V_{util} (*engl. utilisation lock*) u datoteku kojom je završena faza zapisivanja podataka koji se odnose na fazu personalizacije. Zapisivanjem brave korištenja je kartica spremna za korištenje.

2.2.4. Faza korištenja

U ovoj fazi korisnik koristi karticu. Aktiviraju se aplikacijski sustav, kontrole za pristup logičkoj datoteci i ostale kontrole koje su potrebne. Pristup informacijama spremljenim na kartici je ograničem sigurnosnom politikom aplikacije koja koristi pametnu karticu.

2.2.5. Završna faza (faza poništavanja kartice)

Postoje dva načina kako kartica dolazi u ovo stanje. Prvi se inicira od strane aplikacije koja zapisuje bravu poništavanja (*engl. invalidation lock*) u individualnu datoteku ili glavnu datoteku. Sve operacije, uključujući pisanje i ažuriranje, se onemogućavaju od strane operacijskog sustava. Ostaju samo operacije čitanja koje su predviđene za svrhe analiziranja. Nepostojanje ostalih operacija nakon zapisivanje brave poništavanja kartica postaje velikim dijelom beskorisna jer više ne može spremati nikakve nove podatke na sebi (na primjer nove certifikate nakon isteka njihove valjanosti ili kriptografske ključeve nakon isteka perioda njihove validnosti), ali ni koristiti svoje kriptografske mogućnosti (generiranja para ključeva, vrednovanja digitalnog potpisa i ostalih kriptografskih operacija).

Drugi način dovođenja kartice u ovo stanje se postiže ireverzibilnim blokiranjem pristupa kartici od strane kontrolnog sustava. To nastaje zbog blokiranja PIN-a i blokiranja deblokacijskog PIN-a (obje vrste PIN-a su tri puta krivo unesene). U ovom slučaju je onemogućena i operacija čitanja podataka s kartice. Tablica 2.1. sažima uvjete i pristupanje memoriji pametne kartice tijekom različitih faza koje su prije navedene.

Tablica 2.1.: Faze i prava pristupa u tijekom životnog ciklusa pametne kartice

Područja/faze	**Proizvodnja**	**Pripremna faza Personalizacije**	**Personalizacija, korištenje i završna faza**
Način pristupa	Fizičko adresiranje		Logičko adresiranje
Sustav	Nije dostupno		
Ključevi proizvođača	Zapisivanje ključa proizvođača	Zapisivanje ključa personalizacije	Nije dostupno
Podaci proizvođača	Čitanje, pisanje, brisanje	Čitanje	Čitanje
Direktorij	Čitanje, pisanje, brisanje		U skladu s uvjetima logičkog pristupa datotekama
Podaci	Čitanje, pisanje, brisanje		U skladu s uvjetima logičkog pristupa datotekama
Opcionalni kôd	Čitanje, pisanje, brisanje		Nije dostupno

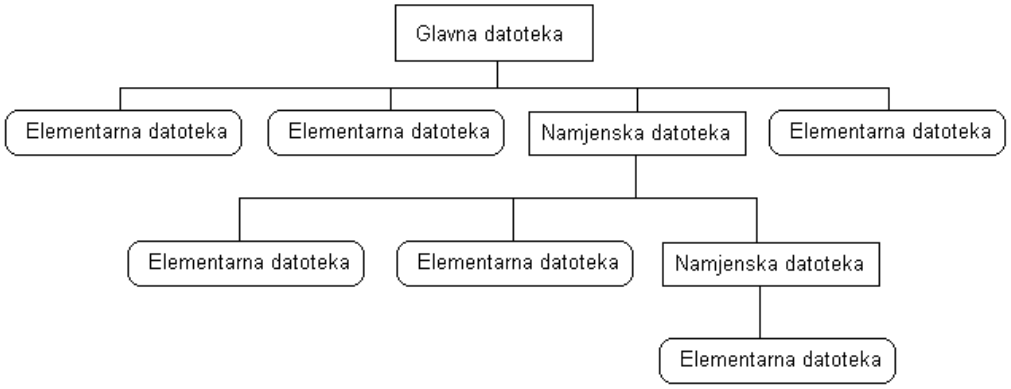
[2]

3. Logička struktura i kontrola pristupa

Nakon što je pametna kartica izdana od aplikacijskog poslužitelja korisniku, zaštita kartice se uglavnom kontrolira aplikacijskim operacijskim sustavom. Fizički način adresiranja podataka je onemogućen, nego se pristup podacima ostvaruje preko logičke strukture datoteka na kartici.

3.1. Logička struktura datoteka

U pogledu pohranjivanja podataka, na pametnu karticu se može gledati kao na disk gdje su podaci ogranizirani u hijerarhiji direktorija. Slično kao i kod MS-DOS sustava, postoji jedna glavna datoteka (*engl. master file - MF*) koja ima funkciju sličnu korijenskom direktoriju. Unutar glavne datoteke mogu postojati različite datoteke koje se nazivaju elementarnim datotekama (*engl. elementary files - EF*). Također mogu postojati i različiti poddirektoriji koji se nazivaju namjenske datoteke (*engl. dedicated files - DF*). Svaki poddirektorij opet može sadržavati nove elementarne datoteke i namjenske datoteke. Glavna razlika između datotečne strukture kod pametne kartice i strukture datoteka u MS-DOS sustavu je ta što namjenske datoteke mogu sadržavati i podatke. Slika 3.1. pokazuje logički pogled na datotečnu strukturu pametne kartice.



Slika 3.1. Logička struktura datoteka kod pametnih kartica

U terminologiji pametnih kartica korijen ili glavna datoteka, osim zaglavlja koje sadrži informacije o datoteci, sadrži i tijelo u kojem se nalaze sva zaglavlja namjenskih i elementarnih datoteka koje sadrži glavna datoteka u hijerarhiji.

Način rukovanja podacima unutar datoteke je ovisan o operacijskom sustavu. Neki operacijski sustavi rukuju podacima uz pomoću pomaka (*engl. offset*) i duljine podataka, dok drugi organiziraju podatke u fiksnim ili varijabilnim duljinama zapisa kao što GSM (*engl. Global System for Mobiles*) sustav. U svim slučajevima se datoteka mora prvo odabrati prije bilo kakve operacije nad njom, što odgovara otvaranju datoteke.

Zbog automatskog odabira glavne datoteke mehanizmi logičkog pristupa i odabira su aktivirani nakon priključivanja napajanja na karticu. Operacija odabira omogućava kretanje po hijerarhijskom stablu logičke strukture, a može biti padajuća odabirom elementarne ili namjenske datoteke, ili rastuća odabirom glavne ili namjenske datoteke.

Nakon uspješnog odabira, dohvaća se zaglavlje datoteke koje sadrži informacije o datoteci kao što su identifikacijski broj datoteke, opis, tip, veličina itd. Zaglavlje sadrži i informacije o uvjetima pristupa datoteci te trenutnom statusu. Pristup podacima u datoteci ovisi o ispunjenju uvjeta pristupa koji će biti opisani u nastavku rada.

Struktura datoteka kod operacijskih sustava pametnih kartica je slična ostalim operacijskim sustavima kao što su MS-DOS i Unix. Da bi se povećala sigurnosna kontrola, u zaglavlju svake datoteke su dodana informacije o uvjetima pristupa i trenutnom statusu datoteke. Pristutna je i brava datoteke (*engl. file lock*) koja štiti samu datoteku od neovlašenog korištenja. Ovi sigurnosni mehanizmi i algoritmi pružaju logičku zaštitu pametne kartice.

3.2. Kontrola pristupa

O kontroli pristupa datotekama kod pametnih kartica se brine sustav za kontrolu pristupa. Za svaku datoteku postoji zaglavlje koje sadrži informacije o uvjetima pristupa, zahtjevima i trenutnom statusu datoteka. Fundamentalni princip kontrole pristupa se temelji na ispravnom unosu PIN brojeva i njihovom rukovanju.

3.2.1 Razine uvjeta pristupa

Uvjeti pristupa datotekama se definiraju u slijedećih pet razina. Neki operacijski sustavi mogu imati i više od pet razina, što prvenstveno ovisi o aplikaciji koja se koristi. Razine su slijedeće:

- **Stalni pristup** (*engl. always - ALW*) – omogućen je neograničen pristup datoteci.
- **Verifikacija korisnika 1** (*engl. card holder verification – CHV1*) – pristup se omogućava samo uz ispravnu CHV1 vrijednost. CHV vrijednosti predstavljaju PIN-ove (za korištenje i deblokiravanje kartice), a spremljenje su unutar posebnih datoteka na kartici. Kod uspoređivanja CHV vrijednosti se provjerava zapis u glavnoj datoteci na kartici, a u slučaju da se vrijednosti ne poklapaju, onemogućen je pristup pametnoj kartici. CHV1 vrijednost predstavlja PIN kojim se korisnik prijavljuje u sustav.
- **Verifikacija korisnika 2** (*engl. card holder verification – CHV2*) – pristup se omogućava samo uz ispravnu CHV2 vrijednost. CHV2 vrijednost predstavlja deblokacijski PIN kojima sigurnosni autoriteti (*engl. Security officer*) deblokiravaju karticu nakon unošenja određenog broja neispravnih PIN-ova.
- **Administrativni pristup** (*engl. administrative – ADM*) – pristup koji imaju samo administrativni autoriteti

- **Potpuna zabrana pristupa** (*engl. never - NEV*) – pristup podacima je zabranjen

Navedene razine uvjeta pristupa nisu hijerarhijski raspoređene. Na primjer, ispravna CHV2 vrijednost ne znači da je pristup omogućen i onim datotekama koje zahtijevaju spravnu CHV1 vrijednost. Uvjeti pristupa moraju biti ispunjeni prije odabira datoteke tijekom operacije čitanja. Na primjer, kod odabira datoteka za čitanje mora prethodno biti unesena ispravna vrijednost PIN-a, odnosno mora se podudarati s zapisom CHV1 vrijedosti unutar posebne datoteke, ako je to uvjet pristupa datoteci.

3.3. Prezentacija PIN-a

PIN-ovi su najčešće spremljeni u dvije odvojene elementarne datoteke, na primjer u EF_{CHV1} i $EF_{CHV2.}$ Upotrebom uvjeta pristupa nad tim datotekama su PIN-ovi zaštićeni od neovlaštene izmjene. PIN može biti promijenen uz pomoć instrukcije za promjenu PIN-a zajedno s novim i starim PIN-om. Međutim, za većinu operacijskih sustava odgovarajući PIN će postati nevažeći ili blokiran u slučaju kad se unese fiksan broj neispravnih PIN-ova zaredom. Broj pokušaja unosa PIN-a ovisi o sustavu.

U slučaju da se kartica blokira zbog prekoračenog broja neispravnog unosa PINA-a, PIN se blokira i postaje nedostupan. Deblokiranje PIN-a se izvodi pomoću ispravnog PIN-a i određenog deblokacijskog PIN-a spremljenog u posebnoj datoteci na kartici. Ako se kod deblokacije nekoliko puta zaredom navede krivi deblokacijski PIN, vrši se blokiranje deblokacijskog PIN-a. Tad postaju nevažeća oba PIN-a što izaziva ireverzibilnu blokadu i kartica postaje neupotrebiva.

3.3.1. Rukovanje PIN-om

Za postizanje zaštite i blokiranja PIN-a, potrebno je implementirati dva brojača, za svaki od verifikacijskih brojeva vlasnika kartice (CHV). Brojači su zaštićeni od ostatka memorijskog prostora na kartici čime je onemogućen bilo kakav utjecaj promjene ili brisanja podataka na njih. Postoje tri stanja kod rukovanja PIN-om:

1. Unošenje PIN-a

Datoteke ili funkcije koje predstavljaju PIN dohvaćaju se tijekom procesa unošenja PIN-a. Svaki put kad je PIN unesen ispravno, brojač se resetira na maksimalnu vrijednost pokušaja unošenja, npr. tri.

2. PIN nije unesen ili je neispravan

Brojač se dekrementira nakon svakog unošenja neispravnog PIN-a. Nijedna operacija ili instrukcija, koja kao nužan uvjet za korištenje zahtijeva unošenje ispravnog PIN-a, nije dostupna. Ako brojač dostigne vrijednost nula, PIN se blokira.

3. PIN je blokiran

U ovom stanju se blokiraju sve operacije ili instrukcije koje nužno zahtijevaju unošenje PIN-a. Tada je potrebno izvesti instrukciju za deblokiranje. Ako je unesen ispravan PIN za deblokiranje, brojač unošenja PIN-a se resetira na maksimalnu vrijednost pokušaja i kartica se postavlja u početno stanje. Međutim, ako se unese neispravan PIN za deblokiranje, njegov brojač se također dekrementira sve dok ne dostigne vrijednost nula, a tada PIN više ne može biti deblokiran.

S obzirom na strukturu datoteka i kontrolu pristupa koju pametna kartica podržava, podaci na kartici se mogu individualno zaštiti postavljanjem uvjeta u zaglavlju svake datoteke, ili hijerarhijskim grupiranjem datoteka ispod jedne namjenske datoteke koja ima postavljene uvjete pristupa. Nadalje, ireverzibilnim blokiranjem se postiže maksimalna razina zaštite kartice pa velike štete nisu moguće. Prava pristupa su objašnjena u dijelu 3.2.1.

[2]

4. Proceduralna zaštita

Nakon pregleda fizičke i logičke zaštite na pametnoj kartici, u nastavku rada se opisuje način na koji se koristi pametna kartica kako bi zaštitila i povećala razinu sigurnosti korisničkih sustava.

Zbog ugrađenih (*engl. on-board*) računalnih sposobnosti na pametnoj kartici moguće je obavljati off-line transakcije i verifikaciju. Na primjer, pametna kartica i prihvatni uređaj (*engl. card acceptor device – CAD*) se mogu međusobno identificirati korištenjem aktivne autentifikacijske metode. K tome, podaci i kodovi na kartici su kriptirani čipom proizvođača upotrebom računske pseudoslučajne enkripcije (*engl. computational scrambling encryption*). Računska pseudoslučajna enkripcija se odnosi na bilo koju zamjenu različitih simbola, slova ili znamenki, ili pak sheme pseudoslučajnog kodiranja koja kriptira informacije tijekom procesa transmisije podataka da bi se postigla odgovarajuća razina sigurnosti i privatnosti podataka.

Pametne kartice se koriste u različitim područjima jer su kompatibilne s ostalim tehnologijama, kao što su asimetrični kriptografski algoritmi i biometrijska identifikacija. U ovom poglavlju se opisuju tri područja u kojima se demonstrira kako različiti sustavi koriste pametne kartice da bi poboljšali svoja sigurnosna svojstva.

4.1. Identifikacija dokumenata

Uobičajeni dokumenti za identifikaciju, kao što su osobna karta i putovnica, nisu uvijek najpouzdanije rješenje zbog toga jer se mogu falsificirati ili kopirati. Današnja tehnologija s vrlo kvalitetnim fotokopirnim uređajima, pisačima, skenerima je široko dostupna što omogućava olakšanu proizvodnju visokokvalitetnih lažnih dokumenata koje je vrlo teško otkriti.

Pametna kartica je vrlo dobro rješenje tog problema. Ispisane informacije i fotografije mogu biti digitalizirane i spremljene na kartici. Postavljanjem uvjeta pristupa i zaporke na datoteke samo autorizirane osobe ili autoriteti, kao što su vladine službe, mogu pristupiti tim informacijama. K tome, zajedno s biometrijskom tehnologijom, biometrijske informacije vlasnika kartice se mogu pohraniti na karticu pa ona može surađivati s biometrijskim skenerom da bi autentificirala vlasnikov identitet. To svojstvo znatno poboljšava pouzdanost dokumenata koji su pohranjeni na pametnoj kartici.

Operacijske procedure mogu biti slične tradicionalnim identifikacijskom sustavu baziranom na papiru. Međutim, umjesto verificiranja dokumenata pregledavanjem od strane inspekcijskog osoblja, koristi se uređaj za prihvat kartice. Uređaj koji sadrži autorizacijski kod i PIN može otključati datoteku i dohvatiti podatke o korisniku za verifikaciju. U slučaju kad se koristi biometrija, autentifikacija se obavlja tako da korisnik izloži dio svog tijela biometrijskom čitaču, a prikupljeni podaci se uspoređuju s onima koji su pohranjeni na kartici.

Ta tehnologija se koristi u nekoliko različitih područja. Na primjer, mnoge zračne luke upotrebljavaju pametne kartica u svrhu povećanja sigurnosti i unaprijeđenja sustava za rukovanje prtljagom. Pametna kartica pohranjuje detalje putnika kao što su ime, broj sjedala, broj leta, detalji prtljage itd. To pomaže kod verifikacije identiteta putnika i identificira vlasnika u slučaju gubitka prtljage. Što je još važnije, ovaj sustav može pomoći u otkrivanju kriminalaca i terorista.

Uporaba pametne kartice kao identifikacijskog dokumenta sve više zamjenjuje tradicionalne certifikate na papiru. Količina informacija o korisniku pohranjene na kartici će se sve više povećavati i biti sve osjetljivija. Zbog toga trenutni sustav za kontrolu pristupa baziran na unosu PIN-a neće biti dovoljno siguran. Zato postoje rješenja u kojima operacijski sustav kartice koristi autentifikacijske algoritme za zaštitu svih datoteka, ili čak cijelog sustava.

4.2. Autentifikacija Kerberos sustavom

U otvorenom distribuiranom računalnom okruženju (*engl. distributed computing environment – DCE*) vrlo je često radna stanica locirana daleko od centralnog poslužitelja i u nepovjerljivom okruženju. Korisnik može biti napadač na sustav ili se pretvarati da je netko drugi kako bi došao do povjerljivih informacija. Zbog toga je potrebno uvesti zaštitu sustava od napada udaljenim domaćina u mreži (*engl. remote network host*).

Kerberos je autentifikacijski protokol koji omogućava klijentima i poslužiteljima pouzdanu međusobnu verifikaciju identiteta prije nego se uopće uspostavi komunikacija. Osim toga osigurava integritet poruka i tajnost podataka. Kerberos mora uvijek proći kroz proces uspostavljanja sigurne autentificirane mrežne konekcije. Taj proces izvršavaju klijent i poslužitelj vrednovanjem međusobnih identiteta prije izvođenja bilo kakvog dijela aplikacije. Klijent i server moraju uspostaviti „povjerenje“ (*engl. trust*) prije uspostave mrežne konekcije. To znači da servis mora znati identificirati klijenta bez traženja ikakve informacije od njega, a klijent mora znati identificirati servis bez traženja ikakve informacije od servisa.

Kerberos obuhvaća tri protokola:

1. Razmjena autentifikacijskih servisa (*engl. Authentification Service Exchange – AS*): protokol u kojem centar za distribuciju ključeva (*engl. Key Distribution Center – KDC*) izdaje klijentu sjednički ključ za logiranje (*engl. logon session key*) i kartu za odobravanje karata (*engl. Ticket-Granting Ticket*).

2. Razmjena karata za odobravanje karata (*engl. Ticket-Granting Ticket Exchange – TGT*): protokol koji omogućava centru za distribuciju ključeva distribuiranje sjednički ključ servisa i ključ za taj servis.

3. Razmjena između klijenta i poslužitelja (*engl. Client/Server Exchange – CS*): u ovom protokolu klijent pokazuje kartu koja omogućava korištenje servisa.

Paketi koji putuju po mreži mogu biti neovlašteno pročitani, presretnuti ili čak promijenjeni. Zato metode zaštite moraju osim tajnosti informacije, moraju osigurati i zaštitu integriteta. Kerberos koristi lozinku i kriptiranje simetričnim ključem da bi autentificirao korisnike i na taj način omogućava komunikaciju između dva ili više računala koja pruža određen stupanj sigurnosti, jer je transparentna prema korisnicima. Protokol za enkripciju podržava simetrične algoritme DES i TripleDES koji su opisani u poglavlju 7.1.

Kerberos autentifikacijski protokol se bazira na interakciji triju strana. Dvije strane, obično u obliku klijenta i poslužitelja, žele otvoriti kanal za međusobno komunikaciju. Svaka od tih dviju strana posjeduje jedinstvenu dugotrajnu lozinku koja se koristi kod logiranja u sustav. Treća strana u autentifikacijskom procesu je centar za distribuciju ključeva. On sadrži tajnu koju klijent i poslužitelj dijele između sebe kod identifikacije, i također zna lozinke svakog klijenta i poslužitelja. Te informacije su pohranjene u bazi u kriptiranom obliku.

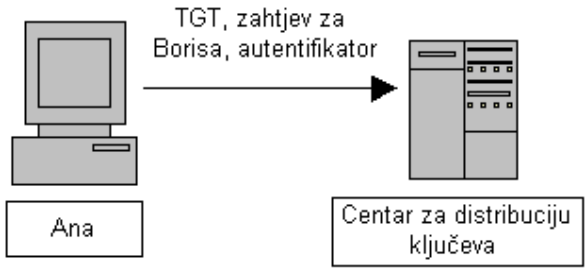
Kerberos karta obuhvaća podatke koji uključuju i vremensku oznaku (*engl. timestamp*) koja je određena trenutkom kreiranja same karte. Klijent i poslužitelj tu vremensku oznaku koriste kao informaciju koja se koristi u procesu autentifikacije pa je vrlo važno da su njihovi satovi precizno sinkronizirani. Vrijeme valjanosti karte je određuje vremenski period za koji je karta predviđena za korištenje. Ono je uvedeno da bi ograničilo mogućnost nastanka štete koju mogu prouzročiti napadači. Uobičajeno vrijeme valjanosti karte iznosi jedan radni dan, ali može varirati što ovisi o sigurnosnoj politici organizacije.

U Kerberos okruženju proces autentifikacije počinje kod logiranja u sustav. Prva osoba uključenja u komunikaciju (neka se radi lakšeg označavanja zove Ana) unosi svoje korisničko ime i lozinku u svoju klijentsku radnu stanicu. Ta informacija se prenosi do centra za distribuciju ključeva, koji sadrži glavnu bazu podataka s jedinstvenim i dugotrajnim ključevima svih subjekata koji su pod njegovim nadzorom. Centar pretražuje bazu podataka i prema Aninoj lozinki pronalazi njen glavni ključ (*engl. master key – K_A*). Nakon pronalaženja glavnog ključa, centar generira sessijski ključ (*engl. session key – S_A*) koji dijeli s Anom, i kartu za odobravanje karata (TGT) koja sadrži niz informacija (među kojima je druga kopija sjedničkog ključa S_A, Anino korisničko ime i datum ukidanja). Ta karta je kriptirana pomoću glavnog ključa centra za distribuciju ključeva koji posjeduje samo centar.

Sve te informacije nakon toga centar kriptira i zajedno s njenim glavnim ključem K_A prosljeđuje do Anine radne stanice. Kriptirana informacija se još sažima funkcijom sažimanja (*engl. hash function*), koja pretvara Aninu lozinku u njen glavni ključ K_A. Upotrebom tog ključa radna stanica dobiva sjednički ključ S_A, koji dijeli zajedno sa centrom za distribuciju ključeva, i kartu za odobravanje karata. S ta dva parametra radna stanica posjeduje sve što joj je potrebno za komunikaciju s centrom.

Ako Ana želi komunicirati s nekom drugom osobom (neka se zove Boris), mora centru za distribuciju poslati zahtjev za komunikaciju s njegovom radnom stanicom.

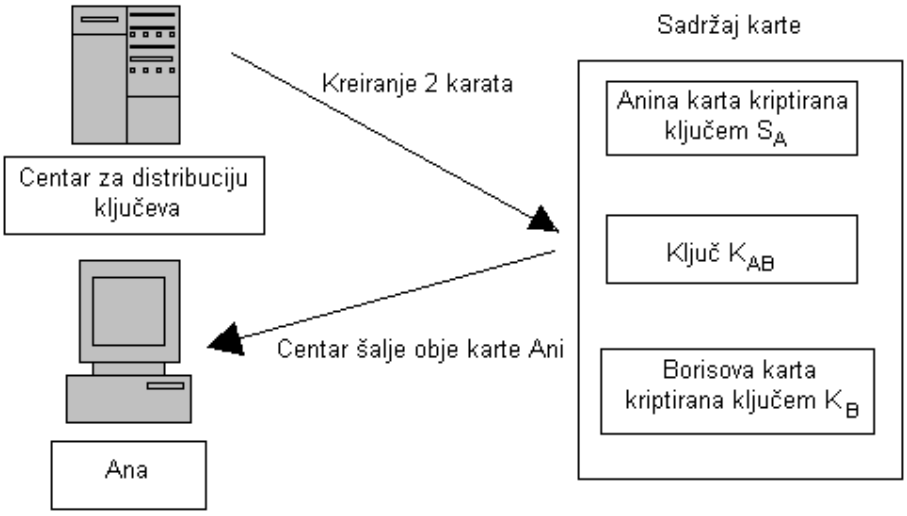
Zahtjev se sastoji od karte za odobravanje karata, konteksta zahtjeva i autentifikatora. Autentifikator je kritički dio informacije u obliku vremenske značke (*engl. timestamp*). Radna stanica mora biti vrlo dobro vremenski sinkronizirana sa centrom za distribuciju ključeva da bi Kerberos protokol ispravno funkcionirao. Autentifikator je kriptiran korištenjen Aninog sjedničkog ključa S_A koji ona dijeli sa centrom. Sljedeća slika prikazuje slanje opisanog zahtjeva od Ane.



Slika 4.1. Slanje zahtjeva za komunikaciju s Borisom

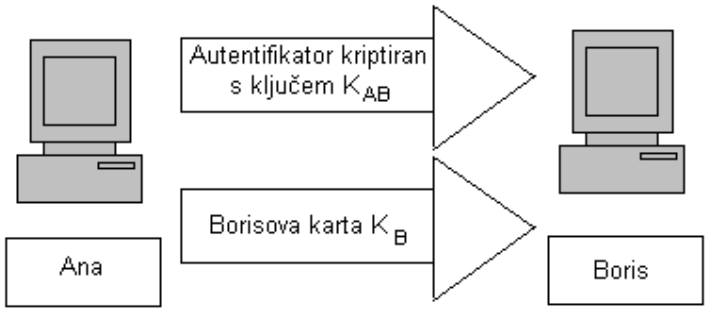
Nakon primanja zahtjeva, centar za distribuciju ključeva upotrebom svog glavnog ključa dekriptira kartu za odobravanje karata čime dobiva Anino korisničko ime i kopiju njenog sjedničkog ključa S_A. Korištenjem tog ključa dekriptira autentifikator i dobiva informaciju o vremenskoj znački koja definira trenutak kada je zahtjev poslan. Kako samo Ana posjeduje te informacije, centar vrlo lako provjerava identitet Ane kao pošiljatelja.

U slijedećem koraku centar kreira par karata, jednu za Anu, a drugu za Borisa. Podaci koje sadrže te karte su ključni za nastavak procesa uspostavljanja komunikacije. Ti podaci sadrže imena sudionika transakcije, vremensku značku kada je karta kreirana, i trenutak kada karta prestaje važiti. Karta sadrži i novi ključ K_{AB} koji će dijeliti Ana i Boris tijekom komunikacije. Borisova karta je kriptirana upotrebom Borisovog glavnog ključa K_B. Karta je ugnežđena unutar Anine karte koja tokeđer sadrži ključ K_{AB}. Svi važni podaci se kriptiraju sjedničkim ključem S_A. Nakon toga se karta šalje Ani. Slika 4.2. prikazuje taj korak procesa.



Slika 4.2. Kreiranje dviju karata i njihovo slanje Ani

Ana dekriptira dobivenu kartu sjedničkim ključem S_A, čime dobiva ključ K_{AB} i Borisovu kartu. Anina radna stanica i dalje ne može čitati Borisovu kartu. Autentifikator se kriptira korištenjem ključa K_{AB} i šalje s Borisovom kartom Borisovoj radnoj stanici. Kad Boris primi te podatke, on prvo dekriptira svoju kartu upotrebom svog glavnog ključa K_B. Nakon toga je može dohvatiti sjednički ključ K_{AB}, kojim tada može dekriptiraju autentifikator od Ane. Slika 4.3. opisuje Anino slanje poruke Borisu.



Slika 4.3. Ana šalje poruku Borisu

U ovom trenutku i Ana i Boris posjeduju nove ključeve. Boris je siguran da mu je baš Ana poslala poruku, jer je za kriptiranje autentifikatora koristila ključ K_{AB}. Kad će Boris komunicirati s Anom, on će isto koristiti novi ključ K_{AB}. Ana može biti sigurna da komunicira s Borisom, jer je samo on mogao svojim glavnim ključem dekriptirati novi ključ K_{AB}. Za svaki zahtjev za komunikacijom se mora provesti cijeli navedeni proces. Centar za distribuciju ključeva kreira jedinstvene sjedničke ključeve za svakog sudionika komunikacije.

Autentifikacija kod Kerberos sustava se od autentifikacije pametnom karticom razlikuje u nekoliko važnih karakteristika. Dok je kod Kerberosa nužno unositi

korisničko ime i lozinku koja se šalje centru za distribuciju ključeva, kod korištenja pametne kartice je korisnik jednoznačno određen (tj. korisničko ime je samim time određeno) jer posjeduje pametnu karticu, a verifikacija lozinke se obavlja lokalno na terminalu. Druga razlika se sastoji u tome što se u Kerberosu izdaju karte koje sadrže kriptografske ključeve, dok se kod pametnih kartica koriste digitalni certifikati koji jednoznačno određuju korisnika, a samim time i sadržavaju njegov javni ključ koji je potreban kod komunikacije.

4.3. Kontrola pristupa operacijskom sustavu

Kontrola pristupa je jedna od važnijih primjena pametnih kartica. U ovom poglavlju se opisuje način kontrole pristupa operacijskom sustavu na osobnom računalu upotrebom pametne kartice.

Obični korisnici osobnih računala često ne vode dovoljnu brigu o sigurnosnoj zaštiti sustava, pogotovo područja sustava kao što su sektor podizanja (*engl. boot sector*) ili ulazno/izlazna jedinica. Njih je moguće mijenjati jer ne posjeduju nikakvu zaštitu, a to otvara mogućnosti zaraze virusima. U današnjim mrežnim okruženjima to može biti opasnije nego što na prvi pogled izgleda.

Zbog toga je razvijen sustav s integritetnih tokenom za podizanje (*engl. boot integrity token system*) koji koristi pametnu karticu kako bi zaštitio operacijski sustav. Sastoji se od osnovne ideje da se računalo domaćin (*engl. host computer*) podiže uz pomoć pametne kartice ili kod podizanja zahtijeva kritične informacije koje se nalaze na pametnoj kartici. Zato i u slučaju kad napadač ima fizički pristup skopovskoj opremi (*engl. hardware*), ne može ugroziti integritet sustava.

Pametna kartica je konfigurirana tako da zahtijeva autentifikaciju korisnika prije bilo kakvog pristupa podacima na njoj. Tijekom pokretanja sustava se moraju provesti dvije autentifikacije prije završavanja sekvence podizanja (*engl. boot sequence*). Kod prve se korisnik autentificira uz pomoć lozinke, a kod druge računalo domaćin autentificira karticu čitanjem tajnih podataka s kartice. Nakon što se uspješno obave obje autentifikacije, računalo domaćin čita podatke potrebne za podizanje uz pomoć pametne kartice, te završava sekvencu podizanja. Poslije tih aktivnosti je osobno računalo raspoloživo za normalan rad.

Pametna kartica isto tako može sadržavati sumu provjere (*engl. checksum*) kritičnih podataka i programa koji se pokreću. Ona je vrlo korisna protiv virusa jer validira integritet datoteka, za razliku od uobičajenog skeniranja virusa. Uporabom pametnih kartica se u ovom slučaju povećava razina sigurnosti računala korištenjem svojstava sigurnog pohranjivanja i procesiranja podataka.

4.4. Napadi na pametne kartice

Kao što se može zaključiti iz navedenog, pametna kartica je dobar alat za unaprijeđivanje sigurnosti sustava. Jedno od glavnih sigurnosnih svojstava koje podržavaju operacijski sustavi pametnih kartica je kriptografija. Njome se provodi kriptiranje i dekriptiranje podataka pametne kartice, te korištenje kriptografskih ključeva.

Najčešći ciljevi napadača pametnih kartica se fokusiraju na pohranjene kriptografske ključeve na kartici i kontrolu pristupa. Napadi mogu biti logički, fizički ili se mogu koristiti različiti matematički algoritmi za probijanje kriptografskih ključeva.

4.4.1. Logički napadi

Svi ključni podaci na pametnoj kartici se nalaze u elektronički izbrisivoj programirljivoj memoriji samo za čitanje (*engl. Electricaly Erasable Programmable Read Only Memory – EEPROM*), a kako je njezina operacija pisanja osjetljiva na promjenjiv napon napajanja i temperaturu, informacije mogu biti pročitane dizanjem i spuštanjem napona napajanja mikrokontrolera.

Zbog toga se u sigurnosne procesore ugrađuju senzori koji aktiviraju alarme u slučaju kad detektiraju bilo kakve promjene u okolini. Međutim, oni nisu vrlo pouzdani pa se događaju alarmi kod male fluktuacije struje u fazi stabiliziranja.

4.4.2. Fizički napadi

Fizički napadi su češći od logičkih. Prije bilo kakvog napada se mora odstraniti integrirani sklop od plastične kartice, što može biti izvedeno najobičnijim naoštrenim nožem. Smola koja okružuje sklop se lako može otopiti kiselinom (HNO_3). Nakon ispiranja vodom je pristup silikonskom sklopu sasvim otvoren.

Osim toga, moguće je još brisanje sigurnosnog bita zaključavanja (*engl. security lock bit*) fokusiranjem ultraljubičaste zrake na EEPROM, praćenje operacija integriranih veza pomoću mikroiglice, ili pak korištenje mikroskopa za lasersko rezanje kako bi se istražio sam sklop, itd. Međutim, takva vrsta napada dolazi u obzir samo u slučaju postojanja vrlo opremljenih laboratorija što za sobom povlači visoku cijenu.

4.4.3. Probijanje kriptografskih algoritama

Pojam probijanja kriptografskih algoritama je vezan za pronalaženje kriptografskih ključeva kojima su kriptirane informacije na kartici. To se najčešće postiže specijaliziranim računalima koja posjeduju specijalizirano sklopovlje za tu svrhu. Što je računalo sofisticiranije, to mu treba manje vremena za otkrivanje kriptografskih ključeva. Zbog današnje vrlo velike brzine razvoja računala se algoritmi s kraćim ključevima (do 256 bitova) vrlo lako probijaju pa se zbog toga više ne koriste. No,

zbog postojanja konstantnih prijetnji se kriptografija razvija proporcionalno s razvojem računala, tako da se u redovitim periodima veličine ključeva udvostručuju što maksimalno otežava ugrožavanje njene sigurnosti.

[2], [11]

5. Arhitektura i vrste pametnih kartica

Kartice s integriranim vezama (*engl. integrated circuit cards*) su danas puno poznatije pod konvencijalnim imenom pametne kartice. To su najnovije i najpametnije inačice ID-1 obitelji kartica koje su izrađene u skladu s ISO 7816 standardom. Pametne kartice posjeduju memoriju i procesor, te se lako izrađuju u formatu kakav je propisan ISO standardom. Vrlo su slične običnim kreditnim karticama, ali s ugrađenim mikroračunalom posjeduju snagu kao neka starija osobna računala. Mikroračunalo može spremati i upravljati podacima, te rješavati matematičke probleme. Mirkočip ima 60 puta više memorije nego obična kartica s magnetskom trakom. Zbog tih svojstava se pametna kartica upotrebljava u mnogim aplikacijama u kojima je vrlo bitna sigurnost. Pametne kartice se trenutno upotrebljavaju za plaćanje telefonskih poziva, plaćanje parkirnih karata i cestarina, spremanje identifikacijskih i liječničkih zapisa, pristupanje satelitskoj televiziji i dr.

5.1. Arhitektura pametnih kartica

Sve pametne kartice imaju tri fundamentalna elemenata kao i ostala računala: snagu procesiranja, memoriju za podatke i ulazno/izlazno sučelje. Snagu procesiranja osigurava ugrađeni mikroprocesor (npr. Intel 8051 ili Motorola 6805), a memoriju za podatke čini memorijski čip (EEPROM, FLASH, ROM i RAM). U nekim karticama su ti elementi kombinirani unutar jednog čipa. Ulazno/izlazno sučelje preko kojeg se obavlja komunikacija s vanjskim uređajima se razlikuje od kartice do kartice. Da bi mogla funkcionirati, kartica mora imati izvor napajanja koji joj osigurava čitač (pasivne kartice) ili je izvor ugrađen na samoj kartici (napredne pametne kartice).

5.1.1 Mikroprocesor

Mikroprocesor je inteligentni element pametne kartice koji interpretira podatke i upravlja njima. Programska podrška koja služi za interpretiranje i upravljanje podacima je ugrađena u memoriju tijekom proizvodnje kartice ili dodana pod kontrolom mikroprocesora. Mikroprocesori u pametnim karticama mogu biti do 16 bitova, a brzine im mogu biti do 10MHz.

5.1.2 Memorija

Memorija pametnih kartica može biti neizbrisiva koja zadržava sadržaj i kad se isključi napon napajanja, te izbrisiva koja gubi sadržaj nakon odspajanja kartice s izvora napajanja što znači da mora imati bateriju. Memorija može imati mogućnost čitanja i pisanja podataka, te samo čitanja (*engl. read-only-memory*). U većini kartica su u neizbrisivoj memoriji upisani podaci kao što su identitet vlasnika ili aplikacijska programska oprema, memoriji s mogućnošću čitanja i pisanja se nalaze podaci koji se često ažuriraju, kao što su iznos na računu nakon obavljene transakcije.

Memorija pametnih kartica može biti kategorizirana u tri tipa: ROM, RAM i programirljivi ROM (PROM). ROM je neizbrisiva memorija čiji sadržaj se puni tijekom faze proizvodnje kartice, nakon čega više ne može biti promijenjena. Trenutno su dostupni čipovi kapaciteta do 32 kilobajta. RAM je izbrisiva memorija, a koristi se za privremeno pohranjivanje informacija. Podaci u njoj se mogu zapisivati, mijenjati, čitati i brisati. Trenutno se koriste čipovi kapaciteta i do 64 kilobajta. Postoje dvije vrste PROM memorije: električno-programirljivi ROM (EPROM) i električno-izbrisivi programirljivi ROM (EEPROM). EPROM memorija ne može biti reprogramirana, za razliku od EEPROM-a koji se može reprogramirati, ali je njegova struktura puno složenija otporna na oštećenja zbog čega joj je cijena puno veća. Trenutno se koriste EEPROM memorije kapaciteta do 8 kilobajta.

Memorija može biti strukturirana tako da omogućava kreiranje nekoliko različitih razina sigurnosnih zona. Otvorena zona se sastoji od nepovjerljivih podataka kao što je identitet vlasnika kartice, ali se njen sadržaj može mijenjati samo od strane autoriziranog osoblja. Radna zona (*engl. working zone*) se sastoji od povjerljivih podataka do kojih se može doći ovisno o ispunjenju određenih sigurnosnih uvjeta (kao što je unošenje PIN-a). Tajna zona sadržava vrlo povjerljive podatke, kao što je PIN. Podatke iz te zone dohvaća prilikom uspoređivanja unešenog PINa-a s onim koji je zapisan na kartici, čime je osigurano da podaci nikada ne napuštaju karticu.

5.1.3. Ulazno/izlazno sučelje

Postoji nekoliko načina na koje pametna kartica komunicira s vanjskim svijetom. Kontaktne kartice obično na svojoj površini imaju metalni kontakt koji se unutar čitača spaja na konektor za komuniciranje. Beskontaktne kartice koriste beskontaktnu metodu transmisije i primanja podataka, što zahtjeva samo približavanje kartice beskontaktnom čitaču. Napredne pametne kartice imaju integriranu tipkovnicu i ekran za prikaz pa ne trebaju vanjski uređaj kao što je čitač. Mogu imati ugrađene kontakte na površini kartice za slučaj kad je potrebno prenijeti podatke do nekog drugog elektroničkog uređaja.

5.1.4. Izvor napajanja

Postoje tri načina napajanja pametnih kartica:

1. Napajanje iz vanjskog izvora kroz metalne kontakte

Kartica električnu energiju dobiva preko dva kontakata od čitača u koji se stavlja. Prilikom toga se kartica resetira i izvršava program.

2. Napajanje slanjem električne energije

Vrsta beskontaktne operacije kao što je induktivno spajanje (*engl. inductive coupling*) šalje energiju za napajanje i podatke od čitača kroz zrak ili nemetalnu površinu do pametne kartice.

3. Napajanje baterijom ugrađenom u kartici

U ovom slučaju je baterija sastavni dio kartice. Ova metoda nije previše raširena zbog poteškoća usklađivanja s ISO standardom u pogledu dimenzija, povećanja cijene zbog ugrađivanja baterije u karticu i problema kod mehaničkog naprezanja kartice s ugrađenom baterijom.

5.2. Vrste pametnih kartica

Postoje četiri osnovne vrste pametnih kartica: memorijske, mikroprocesorske, kartice s kripografskim procesorom i beskontaktne pametne kartice. Osim njih postoje i hibridne kartice koje ujedinjuju oba tipa kartica (kontaktne i beskontaktne), te najnaprednije kartice koje ne trebaju ni vanjski izvor energije zbog čega ime je razina sigurnosti dodatno povećana.

5.2.1. Memorijske kartice

Memorijske kartice, koje se još nazivaju i sinkorne kartice, su puno jeftinije, ali podržavaju puno manje funkcionalnosti s obzirom na kartice s mikroprocesorom. Sadrže EEPROM i ROM memoriju, te adresnu sigurnosnu logiku, ali ne sadrže mikroprocesor ni operacijski sustav koji ih kontrolira. Najjednostavniji dizajn podržava logiku koja onemogućava pisanje i brisanje podataka. Složeniji dizajn nudi mogućnost ograničenog pristupa kod čitanja podataka s kartice. Memorija može sadržavati samo statičke podatke (kao što su razni identifikatori, imena i sl.) ili podatke za koje nije potrebna dinamička enkripcija. Tipične aplikacije s memorijskim karticama su telefonske kartice s "plaćanjem unaprijed" (*engl. pre-paid*) i kartice za zdravstveno osiguranje.

Bezprocesorski čipovi koji se nalaze unutar kartica posjeduju tvrdo-ožičenu logiku (*engl. hard wired logic*) kojom se upravlja sigurnost pristupa kartici. Upravljanje pristupom je izvedeno postavljanjem unutrašnjeg prekidača (*engl. on/off switch*) baziranog na uspoređivanju PIN-a i zapisa u zaštićenoj memoriji u unutrašnjosti kartice. Nakon što test PIN-a uspješno prođe, prekidač je postavljen i podaci su raspoloživi za korištenje.

U slučaju telefonskih kartica i sličnih aplikacija, tvrdo-ožičena logika koristi memoriju čipa kao brojač koji postavlja bitove "1" u "0", ali ne i obrnuto. Bitovi mogu biti tretirani kao zasebni ili grupirani u grupe tako da formiraju oktalni ili heksadecimalni zapis.

Bez obzira na to da li se koristi tvrdo-ožičena logika ili ne, operacija pristupanja podacima na čipu i izvođenje ulazno/izlaznih operacija su pod nadzorom domaćina (*engl. host*). Svaki proizvođač implementira svoje čipove na drugačiji način pa se nakon umetanja u čitač pametna kartica mora ispitati da li je ispravna.

Na primjer, programska podrška domaćina će preko čitača izvršiti nekoliko testova nad karticom. Prvo će poslati zahtjev kojim će saznati vrstu kartice. Ako je na kartici ugrađen procesor, ona će odgovoriti s ATR signalom (*engl. Answer To Reset*) i

ostatak komunikacije se znatno pojednostavljuje. Ako pak kartica ne odgovori za zahtjev, onda znači da je to obična memorijska kartica. Komunikacija je složenija s obzirom na kartica koja sadrži procesor, jer je prije dohvaćanja identifikacijskoj bajta potrebno obaviti i neke druge radnje. Prva od njih je detektiranje ispravnog mehanizma za komunikaciju s memorijskim čipom. Čitači tu detekciju provode tako da šalju različite zahtjeve kartici sve dok ne dobiju odgovor koji određuje mehanizam s kojim će se komunicirati s memorijskim čipom. Iz toga se može zaključiti da se memorijske kartice mogu koristiti samo u okruženjima koja su unaprijed poznata da podržavaju taj tip kartice, jer u suprotnom aplikacija ne reagira na umetanje kartice u čitač.

5.2.2. Mikroprocesorske kartice

Arhitektura ovakvih kartica (koje se još zovu i asinkrone kartice zbog protokola koji koriste za komunikaciju) uključuje komponente kao što su središnja procesna jedinica (*engl. Central Process Unit – CPU*) ili procesor, RAM, ROM i EEPROM. Procesor omogućava zaštitu informacija u memoriji, obavljanje instrukcija te pisanje i čitanje memorije kartice.

Nakon što je procesor priključen na napajanje umetanjem kartice u čitač, čip unutar kartice postaje malo računalo. Za upravljanje koristi operacijski sustav koji se također nalazi na kartici, poznatiji kao SCOS (*engl. Smart Card Operating System*) koji je jedinstven za svaki čip ili proizvođača kartice. Operacijski sustav se često naziva i ROS (*engl. Reader Operating System*). Operacijski sustav je obično smješten unutar ROM memorije, CPU koristi RAM za radnu memoriju, a većina podataka je spremljena u EEPROM memoriji. Empirijsko pravilo kod silikona za pametne kartice govori o tome da RAM zahtijeva četiri puta veći prostor od EEPROM, dok pak on zahtjeva četiri puta više prostora od ROM memorije. Tipična konvencionalna arhitektura pametnih kartica ima sljedeća svojstva:

Tablica 5.1: Svojstva komponenata današnjih pametnih kartica.

Komponenta	**Opis**
RAM	256 bajtova do 64 kilobajta
EEPROM	1 kilobajt do 64 kilobajta
Mikroprocesor	1 kilobajt do 64 kilobajta
Brzina sučelja	Minimalno 9600 bps, half duplex

Zbog postojanja procesora i operacijskog sustava, ulazno-izlazna linija koja spaja pametnu karticu i čitač se ponaša kao obična RS232 kominkacijska linija. Standard ISO 7816-3 koji se odnosi na pametne kartice i čitače, definira komunikacijski mehanizam sličan RS232 sustavu koji operira brzinom od 9600 bauda s parnim paritetom. ISO 7816-3 razmatra i prirodu komunikacijskog prometa i protokola po linku. ISO 7816-3 T=0 definira prirodu poruka i odgovora po komunikacijskom linku bazirane komnikaciji znak po znak.

Komunikacija znak po znak povlači za sobom niz problema u tome što, kad se dogodi greška u poruci, je vrlo teško otkriti koje podatke je potrebno ponovno poslati, a koji su podaci ispravni. Taj problem je bio riješen proširenjem standarda na ISO 7816-3 T=1 koji definira blokovni protokol koji kreira pakete ISO 7816-3 T=0 protokola. T=1 format je puno bolji od T=0 formata u pogledu transmisijskih pogrešaka, ali se format poruke odgovora ne mijenja zbog čega postoji problem koji je prije opisan. Mnoge kartice koje su trenutno na tržištu koji uvijek podržavaju protokol T=0 bez obzira na postojeći problem.

Serijsko ulazno/izlazno sučelje sačinjava jedan registar kroz koji se podaci prenose istovremeno samo u jednom smjeru (*engl. half duplex*), bit po bit. Mikroprocesor za svoj rad treba napajanje, masu i takt (*engl. clock*) što mu osigurava vanjski terminal. Brzim razvojem kartica se postižu sve veći kapaciteti RAM, EEPROM i ROM memorije koji premašuju navedene iznose u tablici 5.1.

Nakon što se procesor spoji na napajanje i resetira, odmah pošalje tok podataka čitaču kojim se određuje čip i protokol koji će se koristiti za komunikaciju. Taj tok podataka je poznatiji pod imenom ATR (*engl. Answer To Reset*).

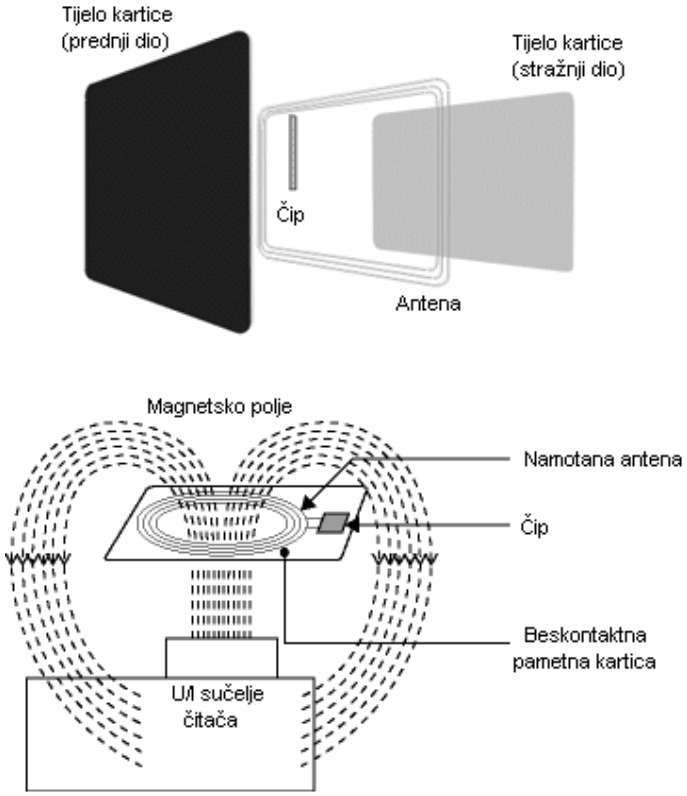
5.2.3. Kartice s kriptografskim koprocesorom

Iako su tehnički u istoj kategoriji kao i mikroprocesorske kartice, ove kartice su od nje odvojene zbog razlika u cijeni i funkcionalnosti. Zbog toga što asimetrični algoritmi kriptiranja (kao što je RSA) zahtijevaju vrlo velike cjelobrojne proračune, 8 bitni procesori s vrlo malo RAM memorije trebaju za izvršavanje operacije s privatnim ključem od 1024 bitova i do nekoliko minuta. Ako se arhitekturi doda kriptografski koprocesor, vrijeme potrebno za takve operacije smanjeno je na nekoliko stotina mikrosekundi. Koprocesori uključuju dodatnu aritmetičku jedinicu koja služi za računanje s velikim cijelim brojevima, kao i za brzo potenciranje. Loša strana toga je cijena. Dodavanja kriptografskog koprocesora povećava cijenu kartice za 50-100%, ali se ta cijena postepeno smanjuje povećanom proizvodnjom kartica. Unatoč većoj cijeni, koprocesor donosi velike prednosti računalnoj i mrežnoj sigurnosti, a omogućeno je to da privatni ključ nikad ne napušta karticu. Sigurnost predstavlja kritičan faktor u operacijama kao što je digitalni potpis, autentikacija i neporecivost. Povećanjem snage i funkcionalosti osnovnih procesora je također moguće da kriptografski koprocesori uskoro više neće biti nužno potrebni. Osim intenzivnih matematičkih algoritama, postoji i tehnologija s eliptičnim krivuljama (*engl. elliptic curve technology*) koja ne koristi velika matematička proračunavanja.

5.2.4. Beskontaktne pametne kartice

Beskontaktne pametne kartice umjesto kontakata na površini kartice upotrebljavaju neku vrstu električnog spoja (*engl. electrical coupling*). Kartica se mora približiti na određenu udaljenost od čitača (koja ovisi o vrsti tehnologije koja se koristi). Čitač zatim preko induktivnog (transformator) ili kapacitivnog spoja transferira električnu energiju i snagu do kartice. Unutrašnji sat kartice se aktivira i komunikacija se

provodi pomoću modulacije signala snage. Struktura beskontaktne kartice i komunikacija sa čitačem je prikazana na slici 5.1.



Slika 5.1. Struktura i komunikacija beskontaktne pametne kartice sa čitačem

5.2.4.1. Induktivni spoj

Induktivni spoj (*engl. inductive coupling*) uključuje dvije vrste namotaja žice: primarni i sekundarni. Izmjenična struja prolaskom kroz primarni namotaj stvara izmjenično magnetsko polje koje inducira struju struju u sekundarnom namotaju, koji se nalazi blizu primarnog namotaja. Modulacijom struje na dvije različite frekvencije je prolaskom kroz primarni namotaj omogućeno slanje podataka do sekundarnog namotaja. Kad kartica primi električnu energiju, demodulira signal, te istovremeno dohvaća podatke i aktivira unutrašnji strujni krug dobivenom energijom od čitača. Mogućnost istovremenog transferiranja informacija i energije za napajanje daje ovom procesu prednost pred kapacitivnim spojem.

5.2.4.2. Kapacitivni spoj

Kapacitivni spoj (*engl. capacitive coupling*) uključuje ugradnju para vodiča ispod površine pametne kartice. Kad se vodiči izlože naponskom signalu, nastaje razlika napona koja generira električno polje. Polje se proširuje iza površine kartice i inducira drugu razliku napona na drugom paru vodiča na ulazno/izlaznoj jedinici

čitača, koje šalje podatke između kartice i ulazno/izlazne jedinice. Prednost ove tehnike je u tome što digitalna informacija može biti izravno transferirana bez potrebe modulacije.

Neke vrste beskontaktnih pametnih kartica mogu raditi na većim udaljenostima od čitača pomoću energije radio valova. Međutim, za takvu komunikaciju je često potrebna vrlo velika snaga, pa većina takvih radio sustava koristi kartice s ugrađenim baterijama. Zbog toga takve kartice često ne mogu poštivati dimenzije koje su propisane standardom (ISO 7816-1) pa to više nisu ispravne pametne kartice.

Današnje su beskontaktne pametne kartice definirane standardom ISO 10536 koji je dosta problematičan, jer različiti proizvođači koriste različite spojeve (kapacitivni, induktivni i dr.) koji nisu međusobno kompatibilni. Kako tehnologija napreduje, postaju dostupni čipovi vrlo male snage kao i vrlo tanke baterije pa je moguće da takve kartice onda zadovoljavaju standarde fizičkih dimenzija.

Beskontaktne pametne kartice imaju nekoliko prednosti pred kontaktnim pametnim karticama:

1. **Pouzdanost** – kvarovi kod električnih uređaja se često događaju baš na kontaktima zbog prljavštine i istrošenosti, što je izbjegnuto beskontaktnim karticama zbog nepostojanja fizičkih kontakata

2. **Dulji životni vijek** – zbog nepostojanja fizičkih kontakata, a samim time i izbjegavanja održavanja istih

3. **Fleksibilnost** – beskontaktne pametne kartice mogu biti stavljene blizu čitača u bilo kojem smjeru i poziciji, za razliku od kontaktnih koje se u čitač moraju staviti u određenom smjeru

4. **Praktičnost uporabe –** čitač beskontaktnih pametnih kartice može biti ugrađen ispod ili unutar bilo kakve nemetalne površine

5. **Lagano održavanje** - čitači beskontaktnih pametnih kartica ne sadržavaju nikakve pomične dijelove čime je omogućeno lagano održavanje

6. **Robusnost** – čitači i beskontaktne pametne kartice mogu podnositi teške vremenske uvjete i problematična okruženja pa su prikladne za korištenje u različitim okruženjima gdje mogu doći u kontakt s prljavštinom i ostalim neželjim tvarima

Osim navedenih problema sa standardima, trenutna generacija pametnih kartica ima i neke druge nedostatke. Sporo se proizvode, proizvodni postupak je skup, a zbog komplicirane izvedbe s mnogo povezanih komponenata su još uvijek iza kartica koje sadržavaju samo jedan čip. Nadalje, postoje problemi prilikom ispupčavanja (*engl. embossing*) znakovlja na kartici, jer ono oštećuje komponente unutar kartice. Imaju probleme i sa sigurnošću zbog mogućnosti prisluškivanja komunikacije kojom čitač i kartice komuniciraju kroz zračni prostor.

Tijekom protekle i ove godine se u SAD-u i Južnoj Koreji provodilo nekoliko pilot projekata u kojima su se koristile beskontakne pametne kartice, prvenstveno u prometu (naplaćivanje putnih karata) i trgovinama (plaćanje računa). Naplaćivanje karata se obavljalo na način da se za svakog putnika detektira mjesto ulaska u prijevozno sredstvo (npr. vlak ili podzemna željeznica), te kod izlaska izračunavala cijena prijevozne usluge. Nakon toga je putnikov trenutni novčani iznos na kartici umanjen za troškove putovanja. Novčani iznos na kartici se mogao obnavljati na posebnim terminalima koji su bili na lako dostupnim mjestima. Sličan koncept se provodio i kod plaćanja računa u trgovinama, gdje je kupac nakon prolaska kroz blagajnu moga platiti kupljenu robu jednostavnim prolaskom kroz označeno područje ili senzor (čitač).

5.2.5. Hibridne pametne kartice

Hibridne pametne kartice (*engl. Combi Smart Card)* posjeduju mogućnost rada kao kontaktne i beskontaktne kartice, a uz to imaju na sebi i magnetsku traku, te podržavaju tehnologiju s jedno ili dvodimenzionalnim bar kodom. Ta svojstva omogućavaju kartici široko područje upotrebe multiaplikativno korištenje.

5.2.6. Napredne pametne kartice

Do sada opisane kartice su pasivne, jer za svoj rad zahtijevaju vanjski izvor napajanja i terminal (čitač). Ta ograničenja dosta utječu na njihovu prikladnost za neke tipove aplikacija. Na primjer, svaki terminal mora osiguravati dostupnost, ali i zadovoljavajuću razinu sigurnosti koja može biti narušena vanjskim napadačima na sustav. Ti nedostaci su doveli do razvijanja aktivnih pametnih kartica treće generacije, koje su poznatije pod imenom napredne pametne kartice (*engl. Super Smart Card*).

Napredne pametne kartice sadržavaju tipkovnicu i zaslon (*engl. display*) koji se nalaze na samoj površini kartice. Mogu funkcionirati kao potpuno zasebne jedinice (*engl. standalone unit*) ili se priključiti na računalo kontaktima na svojoj površini. Nedostatak naprednih pametnih kartica je visoka cijena u usporedbi s drugim vrstama pametnih kartica, poteškoće kod usklađivanja s ISO standardom i male dimenzija tipkovnice na kartici.

Glavna prednost naprednih pametnih kartica je off-line funkcionalnost samovrednovanja (*engl. self-validating*). Za razliku od pasivnih kartica koja trebaju izvor napona od terminala, napredne pametne kartice se mogu koristi uvijek i svugdje, a zajedno s ugrađenim programima za vrednovanje PIN-a i ostalih sigurnosnih svojstava, postižu vrlo visoku razinu zaštite postojećeg sustava pametnih kartica.

[1], [3], [4], [10], [14]

6. Sigurnosni mehanizmi

Sigurnosni mehanizmi predstavljaju temelj sigurnosti pametnih kartica. Sastoje se od algoritama kriptiranja, funkcija sažimanja, te algoritama za razmjenu ključeva. Opisani su mehanizmi koji se najčešće koriste kod pametnih kartica, a osim navedenih postoje još neki manje zastupljeni.

6.1. Algoritmi kriptiranja

Algoritmi kriptiranja služe za šifriranje podataka koji se prenose tijekom komunikacije da bi se zaštitila njegova tajnost i integritet. Postoje dvije vrste algoritama kriptiranja: simetrični i asimetrični. Kod simetričnih algoritama se za kriptiranje i dekriptiranje koristi jedan te isti ključ koji se naziva tajnim. Sigurnost komunikacije ovisi o tome koliko sigurno sudionici komunikacije čuvaju taj ključ. Najpoznatiji simetrični algoritmi su RC5, IDEA, DES i 3DES. Kod asimetričnih algoritama postoje dvije vrste ključeva koje mora posjedovati svaki sudionik komunikacije: javni i privatni. Javni ključ je dostupan svima koji žele komunicirati s tom osobom, a privatni ne smije nitko posjedovati osim te osobe. Najpoznatiji asimetrični algoritmi, koji se još nazivaju i algoritmi za razmjenu ključava, su RSA, ElGamal i Diffie-Hellman.

Primarna prednost asimetričnih algoritama je ta što se privatni ključeva ne moraju slati ni pokazivati bilo kome. Kod simetričnih algoritama se tajni ključ mora poslati drugom sudioniku komunikacije što za sobom povlači rizik otkrivanja podataka tijekom slanja, a samim time i narušavanje sigurnosti same komunikacije. Daljnja prednost asimetričnih algoritama je mogućnost kreiranja digitalnog potpisa. Kod simetričnih algoritama je za autentifikaciju potrebna razmjena neke predefinirane lozinke, a ponekad zahtijeva i vjerovanje nekoj trećoj strani. Kao rezultat, pošiljatelj može poreći svoju transakciju tvrdeći da je prethodno autenticirana poruka bila nekako kompromitirana, od neke od strana koje dijele istu lozinku. Kod asimetričnih algoritama je osigurana neporecivost, jer ako netko digitalno potpiše podatke privatnim ključem onda je time jednoznačno i određen – samo on posjeduje svoj privatni ključ.

Mana korištenja asimetričnih algoritama je u brzini, tj. simetrični algoritmi su puno brži. No, moguće je iskoristiti najbolje iz oba svijeta. Za enkripciju, na primjer, je najbolje rješenje kombiniranje javnog i tajnog ključa te time dobiti sigurnosne prednosti asimetričnih algoritama i brzinu simetričnih algoritama. Enkripcija asimetričnim algoritmima se može iskoristiti da bi se kriptirao tajni ključ (koji je puno kraći od same poruke) s kojim je kriptirana poruka simetričnim algoritmom.

Kriptiranje poruka se obavlja navedenim ključevima koji uz pomoć logičkih operacija nad binarnim zapisom podataka šifriraju podatke. Razina sigurnosti ovisi u duljini ključeva koji se koriste u kriptiranju i dekriptiranju podataka.

6.1.1. DES

DES (*engl. Data Encription Standard*) je jedan od najrasprostranjenijih algoritama kriptiranja podataka na svijetu. Temelji se na kriptiranju grupa od 64 bitova (ili 16 heksadecimalnih znamenaka). Za enkripciju se koriste ključevi koji su također dugi 64 bita. No, svaki osmi bit je u algoritmu ignoriran pa efektivna duljina ključa iznosi 56 bitova. Svaki blok podataka od 64 bita se dijeli na dva dijela, svaki po 32 bita. Lijevi blok se označava slovom **L**, a desni sa slovom **R**.

Primjer:

Neka je **M** tekstualna poruka koja se kriptira: M = 0123456789ABCDEF (heksadecimalni zapis). Pretvaranjem poruke u binarni zapis dobivamo: 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111. Dijeljenjem na dva bloka od po 32 bita dobivamo:

L = 0000 0001 0010 0011 0100 0101 0110 0111,
R = 1000 1001 1010 1011 1100 1101 1110 1111.

Neka ključ **K** heksadecimalni broj **K** = 133457799BBCDFF1, a u binarnom obliku je **K** = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001.
DES se sastoji od dva koraka:

1. KORAK: Stvaranje 16 podključeva, svaki je dug 48 bitova

64-bitni ključ se permutira prema tablici permutacije PC-1. Prva znamenka 57 u tablici označava da će 57. bit originalnog ključa **K** postati 1. bit permutiranog ključa **K+**. Tako će i 49. bit ključa K postati 2. bit ključa **K+**. Nadalje, četvrti bit ključa **K** će biti posljednji bit permutiranog ključa **K+**. Kako je prije spomenuto, samo 56 bitova se koristi kod originalnog ključa **K** kod kreiranja permutiranog ključa **K+**.

Tablica 6.1. Tablica permutacija PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Opisanim postupkom iz originalnog ključa **K** = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001, dobivamo 56-bitnu permutaciju **K+** = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111. Nadalje, potrebno je podijeliti ovaj dobiveni ključ u dvije polovice C_0 **i** D_0, svaka po 28 bitova:

C_0 = 1111000 0110011 0010101 0101111 i
D_0 = 0101010 1011001 1001111 0001111.

Sad kad su definirani parametri C_0 i D_0, mogu se kreirati 16 blokova C_n i D_n, gdje je $1 \leq n \leq 16$. Svaki par C_n i D_n je kreiran od prethodnog para C_{n-1}, D_{n-1}, za n = 1, 2, ..., 16, upotrebom sljedećeg rasporeda "pomaka ulijevo" prethodnog bloka. Kod pomicanja ulijevo pomiču se svi bitovi osim prvog, a on se dodaje na kraj tog bloka koji se pomiče.

Tablica 6.2. Tablica broja pomaka ulijevo za određeni broj iteracija

Redni broj iteracije	Broj pomaka ulijevo
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

To znači da su, na primjer, C_3 i D_3 dobiveni od C_2 i D_2 dvostrukim pomakom ulijevo, a C_{16} i D_{16} su dobiveni od C_{15} i D_{15} jednim pomakom ulijevo. Od početno para C_0 i D_0 dobivaju se ostali parovi (navedeno je samo nekoliko parova radi primjera, a svi ostali se mogu pronaći u Dodatku A na kraju rada):

C_0 = 1111000011001100101010101111 i D_0 = 0101010101100110011110001111
C_1 = 1110000110011001010101011111 i D_1 = 1010101011001100111100011110
...
C_{16} = 1111000011001100101010101111 i D_{16} = 0101010101100110011110001111.

Nakon toga je potrebno formirati ključeve K_n, gdje je $1 \leq n \leq 16$, primjenjujući permutacijsku tablicu PC-2 nad svakim od parova C_nD_n. Svaki par ima 56 bitova, ali se koriste samo 48.

Tablica 6.3. Permutacijska tablica PC-2 za formiranje ključeva K_n, gdje je $1 \leq n \leq 16$

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Prema tablici je prvi bit ključa K_n 14. bit para C_nD_n, drugi bit ključa K_n 17. bit para C_nD_n, i tako dalje sve do posljednjeg (48.) bita ključa K_n, koji je 32. bit para C_nD_n. Tako za prvi par C_1D_1 = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110 nakon permutiranja dobivamo K_1 i ostale ključeve K_n (ostali ključevi se nalaze u Dodatku A na kraju rada):

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010,
K_2 = 011110 011010 111011 011001 110110 111100 100111 100101,
...
K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101.

Time je završena faza dobivanja pomoćnih ključeva uz pomoć permutacije.

2. KORAK: KODIRANJE SVAKOG 64-BITNOG BLOKA PODATAKA

Postoji inicijalna tablica permutacije TIP kojom se permutiraju 64 bitova poruke **M**. Razmještavanje bitova se odvija kao i kod permuniranja bitova ključeva opisanog prije: 58. bit poruke **M** postaje prvi bit inicijalne permutacije itd.

6.4. Tablica inicijalne permutacije TIP bitova poruke

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Primjer:

Primjenjujući tablicu na 64-bitnu poruku **M**, dobivamo inicijalnu permutaciju **IP**:

M=0000000100100011010001010110011111000100110101011110011011111011111I

IP = 1100110000000000110011001111111111110000101010101111000010101010

Nakon toga je potrebno dobiveni blok **IP** podijeliti na sva dijela pa 32 bita. L_0 i R_0:

L_0 = 1100 1100 0000 0000 1100 1100 1111 1111,
R_0 = 1111 0000 1010 1010 1111 0000 1010 1010.

Upotrebom funkcije f se obavlja 16 iteracija ($1 \leq n \leq 16$) nad dva bloka, podatkovnim blokom od 32 bita i ključem K_n, da bi producirao blok od 32 bita.

$L_n = R_{n-1}$
$R_n = L_{n-1}$ ***XOR*** $f(R_{n-1}, K_n)$ (gdje ***XOR*** predstavlja operaciju "isključivo ili").

Na primjer, za n = 1 imamo:

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
$L_1 = R_0$ = 1111 0000 1010 1010 1111 0000 1010 1010
$R_1 = L_0$ ***XOR*** $f(R_0, K_1)$.

Kod izračunavanja funkcije f je najprije potrebno proširiti blok R_{n-1} od 32 na 48 bitova. To se postiže upotrebljavanjem tablice 6.5. koja ponavlja neke bitove od R_{n-1}.

Tablica 6.5. Tablica za proširivanje podatkovnih blokova s 32 na 48 bitova

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Primjer:

Neka je tablica zadana funkcijom $E(R_{n-1})$. Za R_0 dobivamo $E(R_0)$:

R_0 = 1111 0000 1010 1010 1111 0000 1010 1010,
$E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101.

Može se primjetiti da je svaki blok od 4 bitova proširen na 6 bitova, što na kraju daje konačno proširivanje od 32 na 48 bitova. Slijedeći korak kod izračunavanja funkcije ***f*** je operacija XOR između $E(R_{n-1})$ i ključa K_n.

Primjer:

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010,
$\mathbf{E}(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101,

K_1 **XOR** $\mathbf{E}(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111.

Nadalje, dobiveni blokovi od po 6 bitova se koriste kao adrese u tablicama zvanim S-kutije (*engl. S-boxes*). Na tim adresama se nalaze 4-bitni brojevi kojima se mijenjaju 6-bitni brojevi koji predstavljaju adresu. Ako operaciju K_1 ***XOR*** $E(R_0)$ označimo s $B_1B_2B_3B_4B_5B_6B_7B_8$, gdje je B_i grupa o od 6 bitova, onda u pomoću S kutija izračunavamo $S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$, gdje $S_i(B_i)$ predstavlja izlaz iz ***i***-te S kutije. Za zadani S-kutiju S_1 koja je prikazana tablicom 6.6., pretvorba se vrši na slijedeći način:

Tablica 6.6. $\mathbf{S_1}$ kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1.	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2.	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3.	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Prvi i posljednji bit bloka ***B*** predstavljaju binarni broj u decimalnom razmaku od 0 do 3 (ili binarno 00 i 11). Neka se taj broj označava s ***i***. Središnja četiri bita bloka B predstavljaju binarni broj u decimalnom razmaku od 0 do 15 (ili binarno 0000 i 1111). Neka se taj broj označava s ***j***. Traženi četverobitni broj koji se traži se nalazi u tablici u ***i***-tom retku i ***j***-tom stupcu. Na primjer, za ulazni blok B = 011011 prvi bit je 0, a posljednji 1 što daje binarni broj 01, odnosno decimalni 1 (1. redak). Srednja četiri bita su 1101 što je decimalno 13 (13. stupac). U 1. retku i 13. stupcu tablice se nalazi broj 5, tj. funkcija S_1(011011) = 0101.

Ostalih sedam S-kutija (S_2 do S_8)se nalaze u Dodatku A na kraju rada. Na primjer, za zadanu vrijednost K_1 ***XOR*** $\mathbf{E}(R_0)$ se nakon primjene **S**-kutija dobiva:

K_1 ***XOR*** $\mathbf{E}(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111
$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111.

Posljednji korak u izračunavanju funkcije ***f*** je provođenje permutacije **P** nad dobivenim izlazom: $f = \mathbf{P}(S_1(B_1)S_2(B_2)...S_8(B_8))$. Permutacija **P** za ulazna 32 bita daje također 32 izlazna bita, a njena tablica P je prikazana u nastavku.

Tablica 6.7. Tablica permutacije **P**

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Tako se za niz bitova koje dobivamo na izlazu iz S-kutija $S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111, dobiva izlaz f = 0010 0011 0100 1010 1010 1001 1011 1011. Izračunavanjem izraza $R_1 = L_0$ **XOR** $f(R_0, K_1)$ dobivamo

= 1100 1100 0000 0000 1100 1100 1111 1111
+ 0010 0011 0100 1010 1010 1001 1011 1011
= 1110 1111 0100 1010 0110 0101 0100 0100

U slijedećem koraku imamo $L_2 = R_1$, a nakon toga je potrebno izračunati $R_2 = L_1$ **XOR** $f(R_1, K_2)$ i tako sve do 16. koraka kada se izračunavaju veličine L_{16} i R_{16}. Nakon toga se slijedi spajanje dobivenih blokova u jedan 64-bitni blok $R_{16}L_{16}$ nad kojim se provodi posljednja permutacija $\mathbf{IP^{-1}}$:

Tablica 6.8. Permutacija $\mathbf{IP^{-1}}$

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Na kraju to sve zajedno izgleda ovako:

L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100
R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101

$R_{16}L_{16}$ = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100
IP^{-1} = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101,

što je u heksadecimalnom zapisu: 85E813540F0AB405.

Znači, iz početne poruke **M** = 0123456789ABCDEF pomoću DES algoritma smo dobili kriptiranu poruku **C** = 85E813540F0AB405.

6.1.2. Trostruki DES

Trostruki DES (*engl. triple DES*) algoritam se sastoji od korištenja običnog DES algoritma u tri koraka i korištenja dvaju (u nekim slučajevima se koriste i tri) 56-bitnih ključeva. Prvi ključ se koristi za DES enkripciju tekstovne poruke. Drugi ključ se koristi za DES dekripciju dobivene poruke iz prvog kriptiranja (ako drugi ključ nije pravi, dobije se dodatno miješanje podataka). Poruka nakon dekriptiranja se još jednom kriptira pomoću prvog ključa i dobiva se konačni šifrirani tekst. Ovaj algoritam od tri koraka se naziva trostruki DES.

6.2. Algoritmi sažimanja

Algoritmi sažimanja (*engl. Digest Algorithms*) se koriste kod izračuvanja sažetka neke poruke koja se prenosi tijekom komunikacije. Pošiljatelj prilikom slanja poruke šalje i njen sažetak, a primatelj prilikom primanja cijele poruke može izračunati sažetak dobivene poruke i usporediti ga s poslanim i time utvrditi integritet poruke. Algoritmi sažimanja se najčešće koriste kod digitalnog potpisivanja podataka o čemu će više riječi biti u nastavku rada. Jedan od najpoznatijih algoritama sažimanja je SHA-1.

6.2.1. SHA-1

SHA-1 (*engl. Secure Hash Algorithm*) algoritam služi za izračunavanja sažetka (*engl. message digest*) poruke ili podatkovne datoteke. Poruka smije biti duga najviše 2^{64} bitova, a SHA-1 izračunava sažetak veličine 160 bitova. Sažetak tada može biti ulazni parametar nekog od algoritama potpisivanja koji generira ili verificira potpis poruke. Potpisivanje sažetka umjesto cijele poruke je puno efikasnije, jer je najčešće sažetak puno manji od same poruke. Kod potpisivanja poruka se mora koristiti isti algoritam za izračunavanje sažetka od strane kreatora potpisa i od verifikatora potpisa. SHA-1 algoritam se naziva "sigurnim" jer je izuzetno teško pronaći dvije iste poruke koje bi dale isti sažetak. Svaka promjena poruke će s vrlo velikim postotkom vjerojatnosti rezultirati s promjenom samog sažetka što dovodi do pogreške kod verificiranja samog potpisa.

6.2.1.1. Izračunavanje sažetka poruke

Sažetak poruke se izračunava za poruku čija je duljina u bitovima višekratnih broja 512. U slučaju da duljina poruke ne zadovoljava taj kriterij, poruci se dodaje potrebni broj bitova koji joj ne mijenjaju značenje. To proširenje se provodi na slijedeći način (uz pretpostavku da je duljina poruke manja od 2^{64}):

1. Na kraj poruke se dodaje "1" (Primjer: ako je originalna poruka 01010000, nakon dodavanja "1" je 010100001).

2. Na kraj poruke se dodaju "0" čiji broj ovisi o duljini originalne poruke. Posljednja 64 bita poruke su rezervirana za duljinu originalne poruke **L**.

   Primjer: Ako je originalna poruka nakon 1. koraka ima oblik 01100001 01100010 01100011 01100100 01100101 1, njena duljina **L** iznosi 40 bitova (bez dodane "1") pa je potrebno dodati 407 "0" na kraj poruke (uzimajući u obzir da su posljednja 64 bita rezervirana za duljinu originalne poruke **L**: 512 – 64 = 448), jer je 41 + 407 = 448. Na kraju koraka dopunjena poruka izgleda ovako (heksadecimalni oblik):

   61626364 65800000 00000000 00000000
   00000000 00000000 00000000 00000000
   00000000 00000000 00000000 00000000
   00000000 00000000.

3. Dodavanje duljine poruke na kraj proširene poruke. Za **L** = 40 u heksadecimanom obliku na kraj poruke dodajemo 00000000 00000028. Na kraju proširena poruka izgleda ovako:

   61626364 65800000 00000000 00000000
   00000000 00000000 00000000 00000000
   00000000 00000000 00000000 00000000
   00000000 00000000 00000000 00000028.

Proširena poruka sadržava 16 * **n** riječi (svaka riječ ima 32 bita), gdje je **n** > 0. Ovisno o tom broju **n**, proširena poruka je podijeljena u niz od **n** blokova M_1, M_2, ..., M_n, gdje svaki M_i zadržava 16 riječi i M1 sadržava prvi znak (ili bitove) poruke.

Nakon proširenja poruke slijedi izračunavanje sažetka. Kod izračunavanja se koriste dva međuspremnika (*engl. buffer*), svaki od njih se sastoji od pet 32-bitnih riječi, i niza od osamdeset 32-bitnih riječi. Riječi prvog međuspremnika su označeni s A, B, C, D i E. Riječi drugog međuspremnika se označavaju s H_0, H_1, H_2, H_3 i H_4. Niz od 80 riječi je označen s W_0, W_1, ..., W_{79}. Tijekom računanja sažetka se također koristi i jedan pomoćni međuspremnik TEMP. Da bi se generirao sažetak, obrađuju se blokovi M_1, M_2, ..., M_n, a svaka se obrada odvija u 80 koraka. Prije izračunavanja sažetka se postavljaju inicijalne vrijednosti međuspremnika H_i:

H_0 = 67452301
H_1 = EFCDAB89
H_2 = 98BADCFE
H_3 = 10325476
H_4 = C3D2E1F0

Nakon toga se procesiraju blokovi M_1, M_2, ..., M_n. Nad svakim od blokova M_i se obavljaju slijedeće operacije:

1. Dijeljenje blokova M_i u 16 riječi W_0, W_1, ..., W_{15}, je W_0 krajnje lijeva riječ.
2. Za **t** = 16 do 79 izračunaj

$W_t = S^1 (W_{t-3}$ XOR W_{t-8} XOR W_{t-14} XOR $W_{t-16})$

Funkcija $S^n(X)$ predstavlja operaciju pomicanja bitova unutar riječi X na slijedeći način: $S^n(X) = (X << n)$ OR $(X >> 32-n)$.

3. Postavi $A = H_0$, $B = H_1$, $C = H_2$, $D = H_3$, $E = H_4$.
4. Za **t** = 0 do 79 izračunaj $TEMP = S^5(A) + f_t(B, C, D) + E + W_t + K_t$; E = D; D = C; $C = S^{30}(B)$; B = A; A = TEMP;

Funkcija f_t (B, C, D) se izračunava u ovisnosti o parametru t na slijedeći način:

$f_t(B,C,D) = (B$ AND $C)$ OR $((NOT\ B)$ AND $D)$ ($0 <= t <= 19$)
$f_t(B,C,D) = B$ XOR C XOR D $(20 <= t <= 39)$
$f_t(B,C,D) = (B$ AND $C)$ OR $(B$ AND $D)$ OR $(C$ AND $D)$ $(40 <= t <= 59)$
$f_t(B,C,D) = B$ XOR C XOR D $(60 <= t <= 79)$.

5. Izračunaj $H_0 = H_0 + A$, $H_1 = H_1 + B$, $H_2 + C$, $H_3 = H_3 + D$, $H_4 = H_4 + E$.

Nakon obrade i posljednjeg bloka M_n, sažetak poruke se dobiva spajanjem korištenih blokova H_0 H_1 H_2 H_3 H_4.

Primjer izračunavanja sažetka poruke:

Neka je zadana slijedeća poruka u binarnom obliku: 01100001 01100010 01100011. Može se primjetiti da je dužina poruke **L** = 24 bitova. Proširivanjem poruke se postiže dodavanjem "1", 423 puta "0", te oznaku duljine poruke koja u heksadecimalnom obliku "18". Konačna poruka ima duljinu od jednom bloka pa će parametar **n** poprimiti vrijednost 1. Inicijalne heksadecimalne vrijednosti parametara H_i izgledaju kako je prije opisano:

H_0 = 67452301,
H_1 = EFCDAB89,
H_2 = 98BADCFE,
H_3 = 10325476 i
H_4 = C3D2E1F0.

Podjelom bloka M_1 dobivamo slijedećih 16 riječi:

W[0] = 61626380, W[1] = 00000000, W[2] = 00000000, W[3] = 00000000,
W[4] = 00000000, W[5] = 00000000, W[6] = 00000000, W[7] = 00000000,
W[8] = 00000000, W[9] = 00000000, W[10] = 00000000,
W[11] = 00000000, W[12] = 00000000, W[13] = 00000000,
W[14] = 00000000 i W[15] = 00000018.

Nakon postavljanja parametara A, B, C, D i E, petlja **t** = 0 do 79 dalje slijedeće (navedeni su samo neki koraci, a ostali se nalaze u Dodatku A na kraju rada):

	A	B	C	D	E
t = 0:	0116FC33	67452301	7BF36AE2	98BADCFE	10325476
t = 1:	8990536D	0116FC33	59D148C0	7BF36AE2	98BADCFE
...					
t = 79:	42541B35	5738D5E1	21834873	681E6DF6	D8FDF6AD.

Na kraju se dobijaju konačni parametri H_i koji čine sažetak poruke:

H_0 = 67452301 + 42541B35 = A9993E36
H_1 = EFCDAB89 + 5738D5E1 = 4706816A
H_2 = 98BADCFE + 21834873 = BA3E2571
H_3 = 10325476 + 681E6DF6 = 7850C26C
H_4 = C3D2E1F0 + D8FDF6AD = 9CD0D89D.

Sažetak poruke = A9993E36 4706816A BA3E2571 7850C26C 9CD0D89D.

6.3. Digitalni potpis

Digitalni potpis je kombinacija sažetka poruke i asimetričnog algoritma, a pruža funkcionalnosti autentificiranja pošiljatelja, očuvanje integriteta poruke, te nemogućnost krivotvorenja i ponovnog korištenja.

Digitalni potpis se dobiva izračunavanjem sažetka poruke, a zatim kriptiranjem tog sažetka privatnim ključem pošiljatelja. Kriptirani sažetak može svatko dekriptirati dohvaćanjem pošiljateljevog javnog ključa, ali je samo on mogao stvoriti svoj digitalni potpis jer samo on posjeduje svoj privatni ključ.

Primatelj poruke može provjeriti digitalni potpis tako da javnim ključem pošiljatelja dekriptira sažetak poruke, a zatim izračuna svoj sažetak primljene poruke i usporedi ga dobivenim dekriptiranjem sažetka. Ako su sažeci identični, poruka je autentična jer je jedino pošiljatelj mogao kriptirati sažetak svojim privatnim ključem.

Digitalni potpis ne osigurava tajnost pa je poruku potrebno zaštititi kriptiranjem, npr. javnim ključem primatelja nakon potpisivanja.

6.4. Algoritmi za razmjenu ključeva

Algoritmi za razmjenu ključa služe za sigurnu i povjerljivu komunikaciju između dva ili više subjekta. Povjerljivost informacije se temelji na paru ključeva kojim se poruka kriptira, odnosno dekriptira. Svaki subjekt u komunikaciji posjeduje svoj privatni ključ koji nitko osim njega ne zna, a ostali subjekti imaju javni ključ subjekta s kojim komuniciraju. Tajnost poslane poruke se temelji na tome što nitko osim primatelja ne može pročitati poruku jer nema pravi ključ.

6.4.1. RSA

RSA algoritam je dobio ime po početnim slovima prezimena triju autora, Rona **R**ivesta, Adia **S**hamira i Lena **A**dlemana, a izumljen je 1977. godine. Algoritam se može koristiti za enkripciju javnim ključem, te kod digitalnog potpisivanja. Njegova sigurnost se temelji na težini faktoriziranja velikih cijelih brojeva. Algoritam se sastoji u slijedećim koracima:

1. Generiraj dva velika prosta broja, **p** i **q**, približno iste veličine i takve da njihov produkt daje broj određene duljine u bitovima (npr. 1024 ili 2048).
2. Izračunaj $\mathbf{n} = \mathbf{p} * \mathbf{q}$ i $\varphi = (\mathbf{p}-1)(\mathbf{q}-1)$.
3. Izaberi cjelobrojni broj **e**, $1 < \mathbf{e} < \varphi$, takav da najveći zajednički djelitelj od (**e**, φ) iznosi 1 (**e** i φ su relativno prosti brojevi).
4. Izračunaj tajni eksponent **d**, $1 < \mathbf{d} < \varphi$, takav da je $\mathbf{e}*\mathbf{d} \equiv 1 \pmod{\varphi}$.
5. Javni ključ je (**n**, **e**), a privatni ključ je (**n**, **d**). Vrijednosti **p**, **q**, φ bi također morale biti tajne.

6.4.1.1. Enkripcija

Pošiljatelj A kriptira poruku na slijedeći način:

1. Dohvati javni ključ primatelja B (n, e).
2. Izračunaj cijeli broj $m < n$ koji će predstavljati poruku.
3. Izračunaj kriptirani tekst $c = m^e \bmod n$.
4. Pošalji kriptirani tekst primatelju.

6.4.1.2. Dekripcija

Primatelj dekriptira poruku na slijedeći način:

1. Upotrijebi svoj privatni ključ (n, d) da bi mogao izračunati $m = c^d \bmod n$.
2. Izvrši pretvorbu cjelobrojne reprezentacije poruke m u njen tekstualni oblik.

6.4.1.3. Digitalno potpisivanje

Pošiljatelj digitalno potpisuje poruku na slijedeći način:

1. Kreira sažetak poruke koja se šalje (npr. SHA-1 algoritmom opisanim prije).
2. Izračunaj cijeli broj m, gdje je $0 < m < n - 1$, koji će reprezentirati sažetak.
3. Upotrijebi svoj privatni ključ (n, d) da bi izračunao digitalni potpis
   $\mathbf{s} = m^d \bmod n$.
4. Pošalji potpis **s** primatelju.

6.4.1.4. Verificikacija digitalnog potpisa

Primatelj verificira digitalni potpis na slijedeći način:

1. Uz pomoć javnog ključa pošiljatelja izračunaj cijeli broj $v = s^e \bmod n$.

2. Izračunaj sažetak poruke iz broja v.
3. Ponovno izračunaj sažetak primljene poruke.
4. Ako su oba sažetka identična, potpis je valjan.

6.4.1.5. Jednostavan primjer RSA enkripcije

1. Izabrana su dva prim broja, p = 11 i q = 3.
2. Izračunavanje parametara n = p*q = 3*11 = 33; φ= (p-1)(q-1) = 10*2 = 20.
3. Izabire se e = 3, jer je nzd(e, p-1) = nzd(3, 10) = 1, i nzd(e, q-1) = nzd(3, 2) = 1. Zbog toga slijedi da je i nzd(e, (p-1)(q-1)) = (3, 20) = 1. Funkcija nzd(X, Y) predstavlja operaciju određivanja najvećeg zajedničkog djelitelja između brojeva X i Y.
4. Izračunavanje parametra d za koji vrijedi e*d ≡ (mod φ) daje rezultat d = 7 (provjera 3*7 – 1 = 20 je djeljivo s φ koji iznosi 20).

Na primjer, ako želimo kriptirati poruku m = 7, onda dobivamo

$c = m^e \bmod n = 7^3 \bmod 33 = 343 \bmod 33 = 13$. Znači, kriptirani tekst iznosi c = 13. Na strani primatelja se kriptirani tekst dekriptira:

$m' = c^d \bmod n = 13^7 \bmod 33 = 7$.

Kod dekriptiranja nije potrebno izračunavati cijelu vrijednost 13^7, već je moguće iskoristiti pravilo da je

a = b*c mod n = (b mod n)*(c mod n),

čime se potencijalno veliki cjelobrojni brojevi razbijaju u brojeve s kojima je lakše računati. Na taj način se dobiva:

$m' = 13^7 \bmod 33 = 13^{(3+3+1)} \bmod 33 = 13^3*13^3*13 \bmod 33 = (13^3 \bmod 33)*(13^3 \bmod 33)*(13 \bmod 33) \bmod 33 = (2197 \bmod 33)*(2197 \bmod 33)*(13 \bmod 33) = 19*19*13 \bmod 33 = 4693 \bmod 33 = 7$.

6.4.1.6. Složeniji primjer RSA enkripcije

Na primjer, poruka koju želimo kriptirati je "ATTACKxATxSEVEN". Poruka se prvo podijeli u blokove od po tri znaka kojima se dodjeljuju cjeloborne vrijednosti koje ih reprezentiraju. Te vrijednosti se izračunavaju na sličan način kao što se decimalni brojevi predstavljaju sumom potencija broja 10 (npr. $135 = 1*10^2 + 3*10^1 + 5$). Kako u (engleskoj) abecedi ima 26 znakova, kod izračunavanja cjelobrojnih brojeva koji reprezentiraju blokove poruke se koristi baza 26 (A = 0, B = 1, C = 2, ..., Z = 25). Tako za svaki blok poruke imamo:

$ATT = 0 * 26^2 + 19 * 26^1 + 19 = 513$

$ACK = 0 * 26^2 + 2 * 26^1 + 10 = 62$

$XAT = 23 * 26^2 + 0 * 26^1 + 19 = 15567$

$XSE = 23 * 26^2 + 18 * 26^1 + 4 = 16020$

$VEN = 21 * 26^2 + 4 * 26^1 + 13 = 14313.$

Radi pojednostavljena se zamenaruju ostali znakovi (mala slova, prazna mjesta...) kod dodjeljivanja cjelobrojne vrijednosti bloku poruke. U ovom sustavu kodiranja je najveći cjelobrojni broj koji možemo dobiti od bloka poruke ZZZ, a on iznosi $26^3 - 1 = 17575$. Zbog toga nam treba modul n veći od toga. Izabiremo slijedeće parametre:

Generiramo prim brojeve p = 137 i q = 131 (brojevi su izabrani tako da se znatno ne razlikuju od broja $\sqrt{n}$.

$n = p * q = 137 * 131 = 17947.$
$\varphi = (p - 1) * (q - 1) = 136 * 130 = 17680.$

Neka je e = 3. Provjera:

$nzv(e, p - 1) = nzv(3, 136) = 1,$
$nzv(e, q - 1) = nzv(3, 130) = 1.$

Izračunava se $d = e^{-1} \bmod \varphi = 3^{-1} \bmod 17680 = 11787$.

Javni ključ je (n, e) = (17947, 3), a privatni (n, d) = (17947, 11787).

Kriptiranjem prvog cjelobrojnog broja koji reprezentira "ATT" dobivamo:

$c = m^e \bmod n = 513^3 \bmod 17947 = 8363.$

Dekriptiranjem dobivamo izvornu poruku:

$m' = c^d \bmod n = 8363^{11787} \bmod 17947 = 513.$

6.4.1.7. Stvarni primjer

U praksi se ne kriptira niz malih brojeva kao što je to prikazano u gornjim primjerima, nego postoji samo jedan veliki cijeli broj. Isto tako se za reprezentaciju tekstualnih blokova ne koristi izravno cijeli broj, nego se koristi slučajni sjednički ključ (*engl. session key*) kojim se kriptira tekst pomoću puno bržeg simetričnog algoritma kao što je Trostruki DES. Tada se može koristiti i puno sporiji algoritam za kriptiranje javnog ključa da bi se kriptirao samo sjednički ključ.

Pošiljatelj A šalje poruku primatelju B u formatu sličnu ovom:

Kriptirani sjednički ključ = xxxx

Tekst kriptiran sjedničkim ključem = xxxxxxxxxxxxxxxxx

Primatelj B kriptirani sjednički ključ dekriptira pomoću svog privatnog ključa (n, d). Nakon toga koristi dobiveni sjednički ključ zajedno sa simetričnim algoritmom za dekriptiranje stvarne tekstualne poruke koju mu je poslao Pošiteljatelj A. Obično su u poslanoj poruci uključeni i detaljne informacije o tekstu (npr. informacije o algoritmu kriptiranja, metodama proširivanja poruke, metodama kodiranja poruke, inicijalizacijski vektori i ostali detalji koji trebaju primatelju). Jedina tajna koja se mora čuvati su, naravno, ključevi. Ako kriminalac presretne slanje kriptirane poruke, može pokušati izravno probiti kriptiranu poruku ili probiti kriptirani sjednički ključ i njega upotrijebiti u probijanju poruke. Očito je da sigurnost sustava ovisi o njegovoj najslabijoj vezi.

6.4.1.8. Duljine ključeva i njihov životni ciklus

Duljine ključeva sigurne RSA transmisije iznose obično 1024 bitova, jer su ključevi od 512 bitova izašli iz upotrebe. Naravno, za veću razinu sigurnosti se koriste ključevi duljine od 2048 ili 4096 bitova. Korištenjem snage današnjih računala, kriptiranje i dekriptiranje 4096-bitni ključevima ne predstavlja nikakav problem. U praksi je danas nemoguće probiti sustave koji koriste duljine ključeva od 512 bitova, pa duljine veće od toga drastično podižu razinu sigurnosti sustava. Ako se tekst kriptira simetričnim algoritmom kao što je DES, sjednički ključ bi morao biti duljine 64 bitova. Trostruki DES bi tada imao ključ veličine 192 bita što također pruža vrlo visoku razinu sigurnosti.

Ključevi ima ograničeno vrijeme trajanja iz niza razloga. Najbitniji od njih je zaštita od kriptoanalitičara (napadača). Svaki put kada se ključ koristi, generira niz kriptiranih tekstova. Opetovanim korištenjem je omogućeno napadaču generiranje baze kriptiranih tekstova (uz moguće čitke tekstove iz kojih su nastali kriptirani) koji mogu biti dostatni uspješnom kriptoanalitičaru da pronađe vrijednost ključa. Ukoliko je to slučaj, ključ se smatra kompromitiranih i svako njegovo sljedeće korištenje će biti vrlo rizično.

Razvoj kriptoanalize može dovesti do realnih napada na ključ, odnosno algoritam. Na primjer, duljine RSA ključeva se uvećavaju svakih nekoliko godina da bi se osigurao od sve naprednijih algoritama faktorizacije.

Sljedeći razlog ograničavanja vremena trajanja ključa je minimizacija štete nastale kompromitiranih ključem. Ako napadač ostane pasivan, kompromitiranje se vrlo teško otkriva. Relativno često obnavljanje ključeva može limitirati potencijalnu štetu nastalu kompromitiranih ključem.

Životnotni ciklus se može definirati slijedećim točkama:

1. Generiranje ključa, te moguća registracija javnog ključa.
2. Distribucija ključa
3. Aktivacija, odnosno deaktivacija ključa
4. Zamjena, odnosno nadogradnja ključa
5. Povlačenje ključa
6. Terminacija ključa koja uključuje uništavanje ili arhiviranje ključa

6.5. Digitalni certifikati

Prethodno opisanim mehanizmima je moguće zaštititi tajnost poruke (algoritmima za kriptiranje) i dokazivanje identiteta pošiljatelja (digitalni potpis). No, za potvrđivanje pripadanja javnog ili privatnog ključa nekoj osobi služe certifikati. Svaki sudionik komunkacije mora imati certifikat koji dokazuje njegov identitet, potvrđuje njihov javni ključ koji je izdan digitalno potpisan od neke povjerljive agencije (treće strane kojoj svi bezuvjetno vjeruju). Povjerljiva agencija koja certifikatom jamči da javni ključ zaista pripada određenoj osobi, koja potpisuje certifikate svojim privatnim ključem i koja dopušta svima da provjere vjerodostojnost certifikata korištenjem pripadajućeg javnog ključa, zove se certifikator (*engl. Certificate Authority – CA*).

Prilikom kriptiranja poruke pošiljatelj koristi javni ključ certifikatora kojim kako bi provjerio vjerodostojnost primateljevog certifikata te kako bi se uvjerio da javni ključ stvarno pripada primatelju. Isto tako i primatelj koristi javni ključ certifikatora s kojim provjerava vjerodostojnost certifikata pošiljatelja i kako bi se uvjerio da javni ključ stvarno pripada pošiljatelju.Certifikator može biti bilo koja centralizirana administrativna organizacija kojoj se vjeruje, a koja je voljna garantirati (svojim postupcima i implementacijom sigurnosti) identitet onih kojima izdaje certifikate i asocijaciju za dani ključ. Na primjer, tvrtka može izdati certifikate za svoje uposlenike, fakultet za studente, itd. Da bi se izbjegli krivotvoreni certifikati, javni ključ certifikatora mora biti objavljen ili dostavljen na upit.

Lanac certifikata je skup logički povezanih certifikata digitalnim potpisom. Drugim riječima, to je skup svih certifikata koji su nužni da se sagradi niz certifikata od samo-potpisanog certifikata do nekog certifikata kojemu je neki certifikator izdao certifikat.

Certifikator izdaje certifikate sljedećom procedurom. Osoba koja želi dobiti svoj certifikat generira par ključeva (javni i privatni), te šalje javni ključ zajedno s nekim dokazom o identitetu certifikatoru. Certifikator provjerava identitet, te izvršava dodatne radnje kojima provjerava da li je zahtjev stvarno došao od osobe koje se identificirala da želi certifikat. Nakon toga generira certifikat s dobivenim javnim ključem koji dokazuje povezanost dotične osobe s javnim ključem, uz hijerarhiju certifikata (ukoliko postoji) koji su potrebni za provjeru certifikata certifikatora. Ako osoba želi demonstrirati legitimnost svog javnog ključa, mora prezentirati svoj lanac certifikata.

Budući da certifikator mora provjeriti identifikaciju subjekta kojemu se izdaje certifikat, organizacijama je zgodno zbog lakših provjera identiteta da je certifikator i dio tih organizacija.

Vrlo je bitno da se privatni ključevi servisa certifikatora sigurno pohranjeni, jer bi kompromitacija privatnih ključeva omogućila falsifikate koje nije moguće razlikovati od originala.

Certifikati su u skladu s X.509 formatom koji sadržava sljedeće informacije o certifikatima:

- Trenutnu verziju formata certifikata (najčešće je to verzija 3)
- Serijski broj certifikata kojeg generira izdavač (obično je to inkrementirajući unutrašnji brojač)
- Ime entiteta kojemu je certifikat izdan
- Ime entiteta koji je izdao certifikat
- Algoritam kojim je obavljeno potpisivanje certifikata
- Period valjanosti certifikata
- Javni ključ subjekta koji se certificira
- Ekstenzije certifikata

Osim podataka o identitetu vlasnika i javnom ključu, za svaki certifikat postoji i period valjanosti. To je period unutar kojeg certifikator jamči za ispravnost podataka. Izvan tog perioda certifikator više ne jamči za ispravnost podataka te se takav certifikat smatra nevažećim, a podaci koje sadrži nesigurnima. Certifikat certifikatora ima puno dulji period valjanosti od korisničkih certifikata.

Moguće je da vlasnik certifikata unutar perioda valjanosti promijeni svoj javni ključ ili neke druge osobne podatke naznačene na certifikatu. U tom mu slučaju certifikator mora izdati certifikat s novim podacima, a stari certifikat proglasi nevažećim. Kako se certifikati vrlo često razmjenjuju Internetom, certifikator mora na neki način naznačiti ostalim korisnicima da je stari certifikat postao nevažeći.

U tu svrhu certifikator izdaje listu poništenih certifikata (*engl. Certification Revocation List – CRL*). Na toj listi se nalaze certifikati koji su iz različitih razloga postali nevažeći, a svrha joj je da korisnike, prije korištenja nekog javnog ključa, obavijesti da certifikator više ne jamči za vjerodostojnost tog ključa. Nakon izdavanja novog certifikata, stari je odmah dodaje u CRL kako bi se korisnike obavijestilo o prestanku njegove valjanosti.

[5], [6], [7], [8]

7. Praktični rad – demonstracija sigurnosnih mehanizama koristeći pametne kartice

Kao pokazni primjer upotrebe prethodno navedenih sigurnosnih mehanizama se uzima pojednostavljena aplikacija elektroničkog plaćanja. Za razvijanje aplikacije je izabran programski jezik Java. Aplikacija obuhvaća osnovne operacije sa sigurnosnim mehanizmima koje posjeduje pametna kartica, a koji su opisani su prethodnom dijelu rada. Pametna kartica koja se koristi je Cryptoflex V2 s 8 kilobajta raspoložive memorije. Kartica je tipa mikroprocesorskih pametnih kartica koje imaju ugrađeni kriptografski koprocesor što joj omogućava obavljanje kriptografskih operacija na samoj kartici. Za čitač pametne kartice je izabran Towotoko Chipdrive čitač.

Za potrebe razvijanja aplikacije se koristi gotov programski omotač i poslužitelj usluga standarda PKCS #11 (*engl. Public Key Cryptography Standard #11 Wrapper and Provider*) proizvođača IAIK-a (*engl. Inistitute for Applied Information Processing and Communications*). PKCS #11 je standardno sučelje za kriptografske tokene koje definira programsko sučelje (*engl. Application Programmable Interface – API*) prema uređajima koji sadrže kriptografske podatke ili vrše kriptografske funkcije. Naziva je još i Cryptoki.

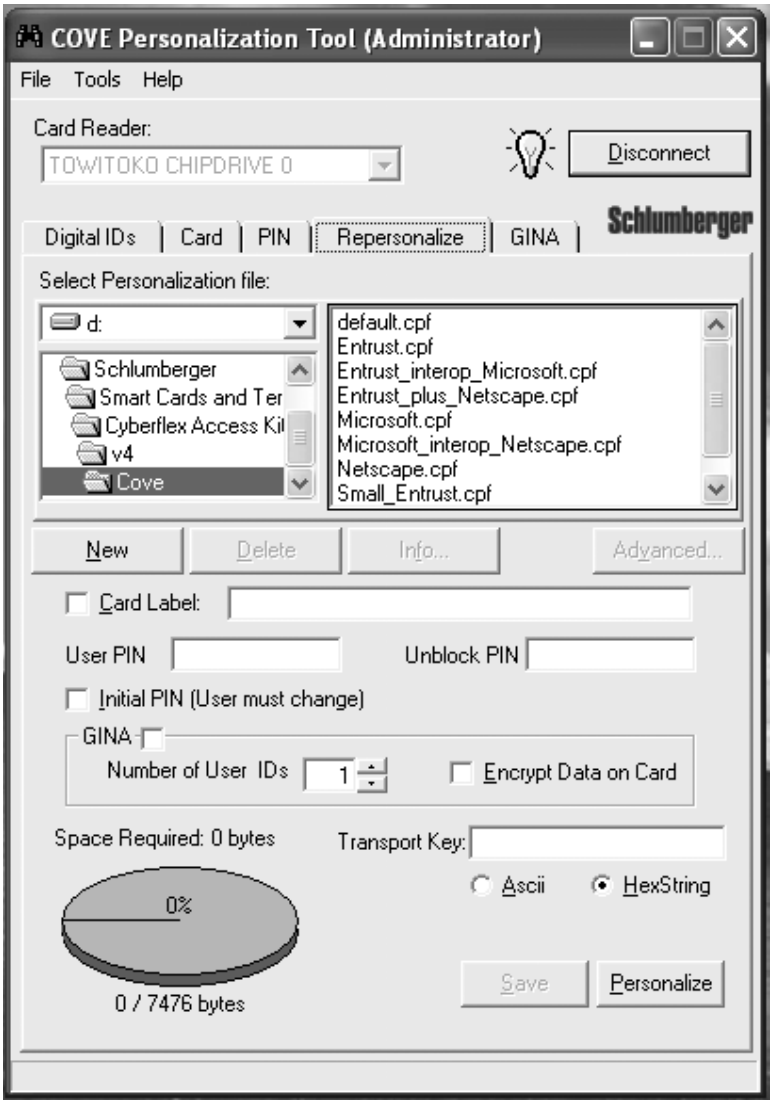
PKCS #11 programski omotač predstavlja knjižnicu (*engl. library*) preko koje je moguće pristupiti PKCS #11 modulima (karticama i čitačima koji su predstavljeni konfiguracijskim datotekama proizvođača) preko programskog jezika Java. Za pristup modulima upotrebljava multijezičnu knjižnicu (*engl. Java Native Library*) kojom je moguće koristiti omotač i u Javi, jer je on izvorno implementiran u programskom jeziku C++.

PKCS #11 poslužitelj usluga implementira kriptografsko proširenje u Javi (*engl. Java Cryptography Extension - JCE*). Bazira se na PKCS #11 programskom omotaču i poslužitelju usluga IAIK proizvođača. Poslužitelj osigurava kriptografske funkcionalnosti kao što su funkcije sažimanja, kodovi za autentificiranje poruke (*engl. message authentication code*), simetrično, asimetrično i tokovno (*engl. stream*) kriptiranje, blokovno kriptiranje, te rukovanje ključevima i certifikatima. Za obavljanje kriptografskih operacija je moguće koristiti algoritme RSA, MD2, MD5, DSA, SHA1, ECDSA, DH, KEA, RC2, RC4, RC5, DES, DES3, CAST, CAST3, CAST128, IDEA, CDMF, SKIPJACK, BATON, JUNIPER, FASTHASH i PBE. U aplikaciji se koriste algoritmi koji su detaljno opisani u radu, tj. RSA, SHA1 i DES.

U aplikaciji se također koristi baza podataka iz koje se vade podaci o vlasniku pametne kartice. Ti podaci se popunjavaju na uplatnicu kojoj se obavlja plaćanje, a iz baze podataka se dohvaćaju po serijskom broju certifikata kao ključu. Baza podataka je implementirana pomoću alata Hypersonic Sql koji se može besplatno skinuti sa njegove službene stranice čiji link je naveden u literaturom.

7.1. Personalizacija kartice

Programski paket koji predstavlja vezni modul (*engl. middleware*) između operacijskog sustava te same kartice i čitača pruža mogućnost personalizacije pametne kartice. Personalizacija čini jednu fazu pripreme kartice za njeno korištenje, a opisana je u dijelu 2.2.3. Za ovu aplikaciju je korišten vezni modul Cyberflex Access Software Development Kit release 4.4. proizvođača Schlumberger. Vezni modul sadrži i preglednik COVE (*engl. Cryprographic Object Viewer and Editor*) koji omogućava personalizaciju kartice. Pomoću njega je moguće pregledati sadržaj memorije kartice, promijeniti PIN i deblokacijski PIN, repersonalizirati karticu i dr.
Sljedeća slika prikazuje grafičko korisničko sučelje preko kojeg ovlaštena osoba može mijenjati podatke na kartici, tj. izvršiti presonalizaciju prazne kartice, ili repersonalizirati već otprije personaliziranu karticu.



Slika 7.1. Korisnično grafičko sučelje za personalizaciju pametne kartice

Na slici je moguće vidjeti da se nudi nekoliko datoteka kojima je moguće automatsko personaliziranje, moguće je unijeti oznaku (*engl. label*) kartice, PIN i deblokacijski PIN, broj korisnika kartice, način spremanja podataka na kartici (u kriptoranom obliku ili ne), te biranje transportnog ključa koji predstavlja autorizacijski ključ za aplikaciju. On se koristi kod novih kartica da bi se omogućio pristup kartici prije same personalizacije.

7.2. Rad aplikacije

Nakon što je obavljena personalizacija kartice, ona je spremna za korištenje u aplikaciji. Aplikaciju nije moguće koristi ako se ne unese ispravan PIN na početnom ekranu koji se pojavljuje nakon pokretanja same aplikacije.

7.2.1. Unošenje PIN-a

Nakon pokretanja aplikacije se pojavljuje login prozor pomoću kojeg se unosi PIN. PIN se sastoji od maksimalno osam znamenaka. Nakon unošenja PIN-a u odgovarajuće polje, pritisak na tipku "Login" se pokreće akcija provjere unesenog PIN-a, a tipkom "Odustani" se izlazi iz aplikacije. Sljedeća slika prikazuje login prozor za unos PIN-a.

Slika 7.2. Login prozor aplikacije

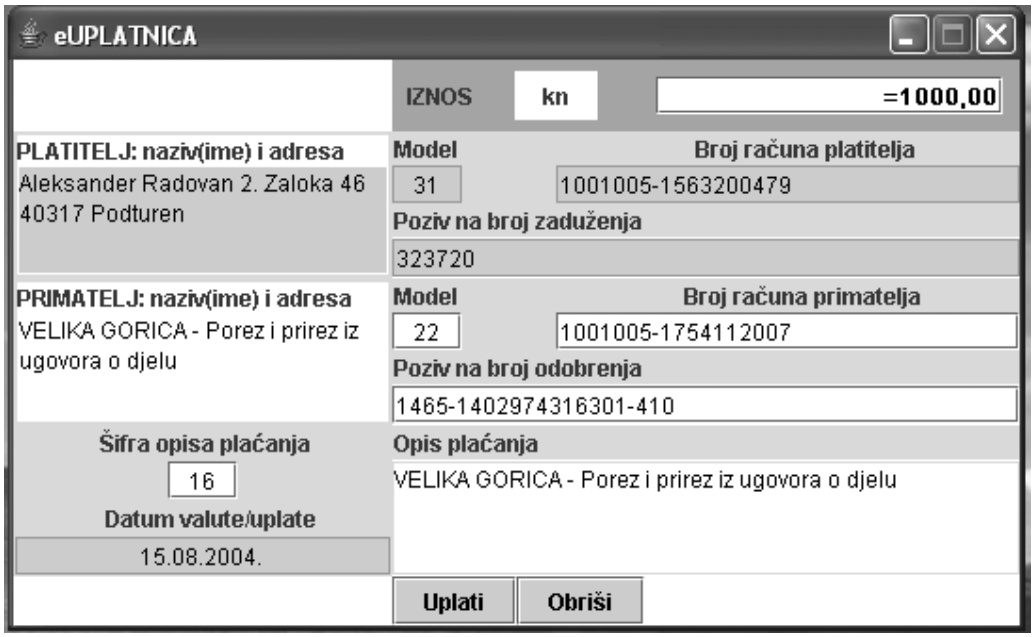
Proces provjere PIN-a se sastoji od inicijalizacije modula, dohvaćanje liste slotova koji su pristutni u operacijskom sustavu, odabira kartice u jednom od slotova (ako ih ima više od jednog), te otvaranje sjednice nad izabranom karticom. U slučaju unošenja krivog PIN-a je ulaz u aplikaciju onemogućen i pojavljuje se sljedeći prozor.

Slika 7.3: Poruka koju ispisuje aplikacija u slučaju unošenja krivog PIN-a.

U slučaju da se krivi PIN unese tri puta, kartica se blokira i više nije moguće koristiti aplikaciju sve dok se deblokira kartica pomoću deblokacijskog PIN-a.

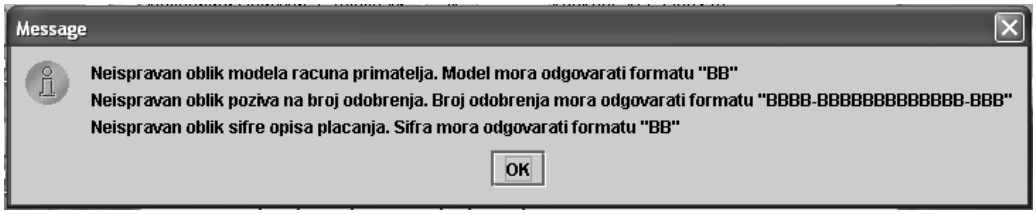
7.2.2. Popunjavanje elektroničke uplatnice

Ako je unesen ispravan PIN, počinju se čitati podaci s kartice (ključevi i certifikat) i prikazuje se prozor elektroničke uplatnice. Uplatnica je ispunjena inicijalnim podacima koji se dohvate iz baze podataka, a koji su vezani uz certifikat na kartici. Inicijalni podaci (polja u kojima se nalaze su sive boje) se ne mogu mijenjati, tj. jednoznačno su određeni vlasnikom kartice. Ostali podaci se upisuju u skladu s potrebama korisnika. Primjer ispunjene elektroničke uplatnice je prikazan na sljedećoj slici.



Slika 7.4. Primjer ispravo ispunjene elektroničke uplatnice.

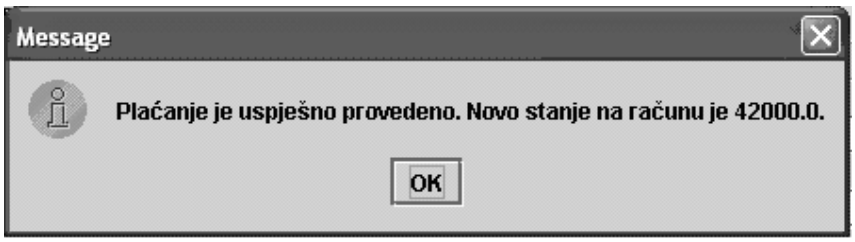
U slučaju da se neki podaci ne unesu u ispravnom obliku, ili se uopće ne unesu, nakon pritiska na tipku "Uplati" se pojavljuje određeno upozorenje koje opisuje razlog zašto uneseni podaci na uplatnici nisu ispravni. Svi podatak koji se nalazi na uplatnici nakon pritiska na tipku "Uplati" podliježe vrednovanju koje provjerava njegov sadržaj i javlja pogrešku u slučaju krivog unošenja. Na sljedećoj slici je prikazan primjer takvih poruka kojima se korisniku opisuje koji podaci nisu ispravni, te u kakvom formatu trebaju biti.



Slika 7.5. Prozor s opisom pogrešaka unešenih podataka na elektroničkoj uplatnici

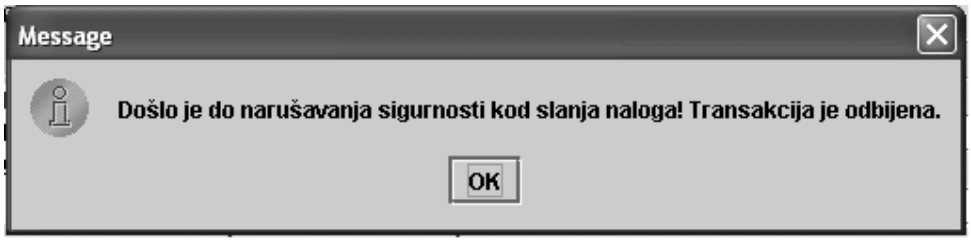
7.2.3. Provođenje transakcija

U slučaju da su svi podaci uneseni na uplatnici uspješno prošli vrednovanje, nalog koji sadržava unesene podatke se zajedno sa certifikatom šalje k serveru koji ga obrađuje. Zbog pojednostavljenja aplikacije je obrada podataka od server predstavljena jednom posebnom metodom koja prima digitalno potpisani nalog i certifikat, te verificira taj potpis i na osnovu rezultata verifikacije obavlja ili ne obavlja transakciju. U slučaju kad verifikacija prođe bez pogreške, na ekranu se pojavljuje prozor s porukom koja sadrži rezultat verifikacije i stanje računa platitelja nakon obavljanja novčane transakcije. Taj prozor je prikazan na sljedećoj slici.



Slika 7.6. Primjer prozora s porukom u slučaju uspješnog provođenja transakcije

U slučaju da verifikacija digitalnog potpisa ne prođe, tj. kad dođe do vanjskog napada na integritet naloga, pojavljuje se poruka koja poručuje korisniku da je transakcija odbijena i da plaćanje nije obavljeno. Sljedeća slika prikazuje taj slučaj.



Slika 7.7. Prozor s porukom u slučaju neuspješnog provođenja transakcije zbog narušavanja sigurnosti sustava

7.3. Izvorni tekstovi programa

U prilogu 1 se nalaze najznačajniji dijelovi tekstova programa. Oni opisuju na koji način je ostvareno opisano ponašanje aplikacije u trenucima kad se koriste sigurnosni mehanizmi.

7.4. Zaključak o praktičnom radu

U radu je detaljno opisana pametna kartica kao sigurni medij za pohranjivanje vrlo osjetljivih podataka, ali i kao alat za obavljanje kriptografskih operacija. Glavni dio pametne kartice je mikroprocesor koji uz pomoć ugrađenog operacijskog sustava, memorije, te ulazno/izlazne jedinice sadržava sve elemente koji sačinjavaju jedno računalo. Upravljanjem datotečnim sustavom je postignuto omogućavanje korištenja podataka na kartici samo od strane ovlaštenih osoba, tj. vlasnika kartice koji posjeduju tajni broj za pristupanje pametnoj kartici, tj. PIN.

PIN je vrlo bitan faktor sigurnosti pametnih kartica, bez njega nije moguće koristiti karticu, a u krajnjem slučaju ju je moguće nepovratno zaključati. Osim korisničkog PIN-a koji je dodijeljen vlasnicima kartice, postoji i deblokacijski PIN kojim se deblokira kartica u slučaju prekoračenja dozvoljenonog broja unošenja krivog PIN-a. Deblokacijski PIN najčešće imaju ustanove koje su izdale pametnu karticu, tako da samo one mogu otključati karticu.

Pametna kartica se može istovremeno koristiti za nekoliko različitih stvari, osim korištenja u standardnim aplikacijama kao što je obavljanje bankovnih transakcija, zbog svoje raznolikosti i velikog kapaciteta za pohranjivanje podataka može pohranjivati sve identifikacijske dokumente na jednom mjestu. Zbog toga se takva multiaplikacijska kartica naziva i "elektronički novčanik". Osim toga se mogu koristiti i za pohranjivanje vrlo osjetljivih podataka koji zahtijevaju vrlo visoku razinu sigurnosti, a to su sektor za podizanje (*engl. boot sector*) operacijskog sustava računala. Njenim korištenjem je omogućena zaštita od računalnih virusa koji vrlo često napadaju sektor za podizanje i time ugrožavaju sigurnost i stabilnost računala, a i tajnost podataka koji su pohranjeni na njemu.

Kako se pametne kartice koriste u aplikacijama koje izvode transakcije vrlo visoke novčane vrijednosti, vrlo brzo su postale meta kriminalaca koji na sve načine pokušavaju narušiti sigurnost tih sustava. Ako se kriminalci uspiju domoći same pametne kartice, raznim postupcima se mogu domoći vrlo osjetljivih podataka koji su spremljeni na samoj kartici, kao što je privatni ključ. Njime onda mogu obavljati transakcije u ime vlasnika ukradene pametne kartice, što se vrlo teško otkriva. U slučaju da kriminalci nemaju karticu, onda se njihov napad bazira na probijanju kripografskih algoritama kriptoanalizom (uz pomoć velike količine kriptiranih informacija pokušaju otkriti ključeve koji se koriste za kriptiranje) ili posebnim računalima koja su specijalizirana za takve svrhe.

Postoji nekoliko vrsta pametnih kartica, čije mogućnosti ovise o njihovoj cijeni. Memorijske kartice su najjeftinije, ali zato nemaju nikakvu moć procesiranja, već se isključivo koriste za pohranjivanjue podataka (kao što su npr. telefonske kartice). Kartice s ugrađenim mikroprocesorom su puno naprednije, jer sadrže i memoriju s

operacijskim sustavom. Njihova cijena je puno veća od memorijskih, ali je puno veća i razina sigurnosti. Kartice s ugrađenim kriptografskim koprocesorom imaju mogućnost obavljanja kriptografskih operacija na samoj kartici, tako da osjetljivi podaci, kao što je privatni ključ, nikad ne moraju napustiti pametnu karticu. Time je sigurnost sustava podignuta na još veću razinu sigurnosti. Beskontaktne pametne kartice donose veliku odobnost korištenja za krajnje korisnike, jer ih uopće nije potrebno stavljati u čitač. Korisnici ne moraju vaditi karticu iz novčanika ili torbice, već je dovoljno da se samo približe čitaču i komunikacija se automatski uspostavlja. Nakon velikog broja pilot projekata se širenje beskontaktnih kartica očekuje u skorije vrijeme, jer je u početku bilo puno sigurnosnih problema koji riješeni tijekom pilot projekata. Da bi spojili najbolje iz oba svijeta, prizvedena je hibridna pametna kartica koja ima svojstva kontaktne i beskontaktne pametne kartice čime je njena multifunkcionalnost i multiaplikativnost još više proširenja. Najnovija vrsta pametnih kartica su napredne pametne kartice (*engl. Super Smart Card*) koja je za razliku od uobičajenih pasivnih pametnih kartica aktivna, tj. posjeduje izvor napajanja, tipkovniku i ekran. Time je riješen problem ranjivosti prihvatnih uređaja koji često znaju biti meta napada kriminalaca čime se narušava sigurnost cijelog sustava s pametnim karticama. Najveća mana naprednih pametnih kartica je njihova cijena, ali i usklađivanje sa standardima, jer postoje mnogi problemi kod postavljanja svih perifernih uređaja na standardni oblik pametne kartice. Zbog toga je napredna pametna kartica još uvijek u fazi proizvodnje.

Kriptografski mehanizmi su temelj sigurnosti sustava koji koristi pametne kartice. Čine ih algoritmi kriptiranja (simetrični i asimetrični), funkcije sažimanja (*engl. Digest Functions*), digitalni potpis, algoritmi za razmjenu ključeva, te digitalni certifikati. Njima je osim tajnosti podataka, zaštićen integritet podataka i osigurana neporecivost provedenih transakcija. Najkritičnija informacija pohranjena na kartici je privatni ključ kod asimetričnih algoritama. Do njega je moguće doći samo uz ispravan PIN koji posjeduje samo vlasnik pametne kartice. Sigurnost kriptografskih algoritama, tj. otpornost na napade kriminalaca je obavezno mora konstantno ažurirati (što se postiže povećavanjem duljine ključeva), jer velikom brzinom raste i snaga računala, a samim time i opasnost od probijanja kriptografskih algoritama.

Za primjer korištenja pametne kartice je uzeta pojednostavljena aplikacija elektroničnog plaćanja. Aplikacijom je prikazano kako se obavlja autentifikacija korisnika unošenjem PIN-a, čitanje podataka s kartice, te potpisivanje i verificiranje digitalno potpisanog naloga za elektroničko plaćanje.

[9], [12]

8. Zaključak

Sigurnosni mehanizmi za rad s pametnim karticama su prikazani aplikacijom pojednostavljenog elektroničkog plaćanja koja je priložena uz rad. Aplikacija obuhvaća temeljne mehanizme kojima je osigurana visoka razina sigurnosti, u koje spadaju unošenje PIN-a (autentifikacija vlasnika kartice), te kriptografski algoritmi pomoću kojih su implementirani osnovni principi sigurnosnih mehanizama (kao što su integritet i neporecivost). Sigurnost pametne kartice je postignuta vrlo visokom razinom zaštite osjetljivih podataka pohranjenih na njoj, a procesiranje podataka i obavljanje složenih kriptografskih operacija je postignuto ugrađenim procesorom, te specijaliziranim sklopovljem. Svaka sofisticiranija pametna kartica osim slopovlja u svojoj memoriji sadržava i operacijski sustav što ih čini ekvivalentnima slabijim osobnim računalima. S obzirom na njihovu veličinu, mogućnosti te lakoću korištenja, pametne kartice su vrlo koristan alat pred kojim se nalazi vrlo svijetla budućnost.

_____

Vlastoručni potpis studenta

Prilog 1 – najvažniji dijelovi programskog koda

P.1.1. Provjera unesenog PIN-a

```
boolean uspjesnostLogina = false;

log.info("Instanciranje pkcs11 modula...");

//instanciranje pkcs11 modula kojim se obavljaju sve
//operacije nad citacem i karticom
Module pkcs11Module = null;
try {
      pkcs11Module = Module.getInstance("slbCk.dll");
} catch (IOException e) {
      throw new EplacanjeLoginException(
            "Doslo je do pogreske kod "
            + "instanciranja pkcs 11 modula.",
            e);
}

log.info("Inicijalizacija pkcs11 modula...");

//inicijaliziranje pkcs11 modula
try {
      pkcs11Module.initialize(null);
} catch (TokenException e1) {
      throw new EplacanjeLoginException(
            "Doslo je do pogreske kod inicijaliziranja "
            + "pkcs 11 modula.",
            e1);
}

log.info("Dohvacanje liste slotova...");

//dohvacanje liste slotova u kojima bi se
//mogla nalaziti pametna kartica
Slot[] slots = null;
try {
      slots = pkcs11Module.getSlotList(
                  Module.SlotRequirement.TOKEN_PRESENT);
} catch (TokenException e2) {
      throw new EplacanjeLoginException(
      "Doslo je do pogreske kod dohvacanja "
      + "liste slotova.",
      e2);
}

//ako je broj dohvacenih slotova jednak nuli,
//znaci da nema slotova s umetnutim karticama.
if (slots.length == 0) {
      JOptionPane.showMessageDialog(
            loginOkvir,
            "Molimo stavite karticu u čitač!");
}

log.info("Dohvacanje informacija o tokenu unutar
      slota...");
```

```
//ako je dohvacanje slotova s kartiom proslo bez
//pogreske, dohvaca se prvi slot jer se u njemu nalazi
//kartica koju trazimo
Slot slot = slots[0];
Token token = null;

//dohvacanje kartice (tokena) koji se nalazi unutar
//citaca
try {
      token = slot.getToken();
} catch (TokenException e3) {
      throw new EplacanjeLoginException(
            "Doslo je do pogreske kod dohvacanja
            + "informacija o tokenu koji je umetnut u
            + "citac.",
            e3);
}

log.info("Otvaranje sessiona nad tokenu...");

//otvaranje session-a nad tokenom
Session loginSession = null;
try {
      loginSession = token.openSession(true, true, null,
      null);
} catch (TokenException e4) {

throw new EplacanjeLoginException(
      "Doslo je do pogreske kod otvaranja "
      + "session-a nad tokenom.",
      e4);
}

log.info("Logiranje u sessionu...");

//logiranje u session koristenjem unesenog PIN-a
//ako dodje do pogreske, znati da uneseni PIN nije
//ispravan
try {
      loginSession.login(Session.UserType.USER, pin);
      uspjesnostLogina = true;
} catch (TokenException e5) {
      JOptionPane.showMessageDialog(
      loginOkvir,
      "Unijeli ste pogrešan PIN! "
      + "Molimo pokušajte ponovo.");

return uspjesnostLogina;
}

log.info("Finaliziranje sessiona...");
```

```
//zavrsavanje aktivnosti s pkcs11 modulom
try {
      pkcs11Module.finalize();
} catch (Throwable e6) {
      throw new EplacanjeLoginException(
            "Doslo je do pogreske kod "
            + "finaliziraja sessiona.",
            e6);
}

log.info("Gasenje login prozora...");

//gasenje prozora za unos PIN-a
loginOkvir.setVisible(false);
```

P.1.2. Dohvaćanje certifikata s kartice

```
try {
      log.info(
      "Instanciranje keystora u kojem ce se
      nalaziti certifikati i kljucevi s
      kartice...");

      KeyStore tokenKeyStore =
            KeyStore.getInstance("PKCS11KeyStore");

      if (tokenKeyStore == null) {
            log.error(
                  "Doslo je do pogreske kod kreiranja "
                        + "keystorea...");
      }

      log.info("Punjenje keystorea podacima s
      kartice...");

      tokenKeyStore.load(null, userPIN_);

      log.info(
            "Dohvacanje aliasa svih elemenana "
            + "na kartici...");

      Enumeration aliases = tokenKeyStore.aliases();

      //petlja koja prolazi svim aliasima dobivenim
      //dohvacanjem elemenata s kartice
      while (aliases.hasMoreElements()) {

            String keyAlias =
                  aliases.nextElement().toString();
            Key key = tokenKeyStore.getKey(keyAlias,
                                          null);

            log.info(
                  "Dohvacen je alias " + keyAlias +
                  "...");
```

```
            //ispitivanje da li je dohvaceni alias
            //privatni kljuc RSA algoritma koji se
            //koristi u digitalnom potpisivanju
            if (key instanceof RSAPrivateKey) {

            log.info(
                  keyAlias + " je RSA privatni
                  kljuc...");

                  //dohvacanje lanca certifikata koji su
                  //povezani RSA privatnih kljucem koji
                  //je dohvacen
                  Certificate[] certificateChain =
                  tokenKeyStore.getCertificateChain(keyAlias);

                  //uzimanje prvog certifikata iz tog
                  //lanca koji ce se uzeti kao primarni
                  //za tog korisnika
                  signerCertificate =
                        (X509Certificate)
                              certificateChain[0];

                  log.info(
                        "Dohvacen je certifikat za
                        potpisivanje sa serijskim brojem
                        "+
                        ignerCertificate.getSerialNumber()+
                        "...");

                  //dohvacanje informacija o tome da li
                  //se kljucevi mogu koristiti za
                  //digitalno potpisivanje i da
                  //li vrijedi pravilo neporicanja

                  boolean[] keyUsage =
                        signerCertificate.getKeyUsage();
                  if ((keyUsage == null)
                        || keyUsage[0]
                        || keyUsage[1]) {

                  log.info(
                        "Certifikat se moze
                        upotrebljavati za digitalno
                        potpisivanje podataka...");

                        signatureKey_ = (PrivateKey) key;
                        verificationKey_ =
                        signerCertificate.getPublicKey();
                        break;
                  }
            }
      }
} catch (KeyStoreException e) {
      log.error("Doslo je do pogreske kod " +
            "kreiranja keystore-a!", e);
} catch (CertificateException e) {
      log.error("Doslo je do pogreske kod dohvacanja " +
            "certifikata!", e);
```

```
} catch (NoSuchAlgorithmException e) {
      log.error("Doslo je do pogreske kod "
            + "koristenja navedenog " +
            "kriptografskog algoritma!", e);
} catch (IOException e) {
      log.error("Doslo je ulazno/izlazne pogreske " +
            "kod dohvacanja podataka!", e);
} catch (UnrecoverableKeyException e) {
      log.error("Doslo je do pogreske s kljucem koja " +
            "od koje se sustav ne moze oporaviti!", e);
}
      return signerCertificate;
}
```

P.1.3. Digitalno potpisivanje naloga za elektroničko plaćanje

```
log.info("Zapocinje potpisivanje naloga...");

Signature signatureEngine =
      rsaSigner.getSignatureEngine();
byte[] signedNalog = null;

try {
      signatureEngine.initSign(
      rsaSigner.getPrivateKey());
      signatureEngine.update(

      nalog.toString().getBytes());
      signedNalog = signatureEngine.sign();
} catch (InvalidKeyException e1) {
      log.error(
            "Koristi se neispravan kljuc za "
                  "potpisivanje naloga...",
            e1);
} catch (SignatureException e2) {
      log.error(
            "Doslo je do pogreske kod " +
            "potpisivanja naloga...");
}

log.info("Nalog se poslao na server i " +
            "ceka se odgovor...");

boolean rezultatTransakcije = sendSignedNalogToServer(
                              nalog,
                              signedNalog,
                              certificate);

if (rezultatTransakcije) {

      log.info("Provjera potpisivanja je prosla
                  u redu...");

      String transactionResult =
            getTransactionResult(nalog);
      JOptionPane.showMessageDialog(okvir,
                  transactionResult);
```

```
} else {
      log.info("Provjera potpisivanja nije
                        prosla u redu...");

      JOptionPane.showMessageDialog(okvir,
                        "Došlo je do narušavanja sigurnosti kod " +
                        "slanja naloga! Transakcija je
                        odbijena.");
                              }
      }
}
```

P.1.4. Provjera digitalnog potpisa

```
boolean rezultatProvjereDigPotpisa = true;

Signature signatureEngine = null;

log.info("Dohvacanje mehanizma za potpisivanje...");

try {
      signatureEngine =
            Signature.getInstance("SHA1withRSA", "IAIK");
} catch (NoSuchAlgorithmException e) {
      log.error("Doslo je do pogreske kod kreiranja" +
                  " mehanizma za potpisivanje!", e);
} catch (NoSuchProviderException e) {
      log.error("Doslo je do pogreske kod kreiranja" +
                  " mehanizma za potpisivanje!", e);
}

try {
      signatureEngine.initVerify(
                  certificate.getPublicKey());
} catch (InvalidKeyException e1) {
      log.error("Doslo je do pogreske kod iniciranja" +
                  " metode za verifikaciju potpisa!", e1);
}

try {

      signatureEngine.update(nalog.toString().getBytes());
            } catch (SignatureException e2) {
                  log.error("Doslo je do pogreske kod verifikaciju "
                  + "potpisa!", e2);
            }

try {
      rezultatProvjereDigPotpisa =
                        signatureEngine.verify(signedNalog);
} catch (SignatureException e3) {
                  log.error("Doslo je do pogreske kod verifikaciju "
                  + "potpisa!", e3);
      return false;
}

return rezultatProvjereDigPotpisa;
```

Dodatak A

A.1. Popis svih parova vrijednosti C_nD_n

Od početnog para

$\boldsymbol{C_0}$ = 11110000110011001010101011111 i $\boldsymbol{D_0}$ = 01010101011001100111100011111

pomicanjem ulijevo (u skladu s tablicom X.1.2.) dobivamo i ostale parove:

$\boldsymbol{C_1}$ = 11100001100110010101010111111 i $\boldsymbol{D_1}$ = 10101010110011001111000111110,

$\boldsymbol{C_2}$ = 11000011001100101010101111111 i $\boldsymbol{D_2}$ = 01010101100110011110001111101,

$\boldsymbol{C_3}$ = 00001100110010101010111111111 i $\boldsymbol{D_3}$ = 01010110011001111000111101101,

$\boldsymbol{C_4}$ = 00110011001010101011111111100 i $\boldsymbol{D_4}$ = 01011001100111100011110101101,

$\boldsymbol{C_5}$ = 11001100101010101111111110000 i $\boldsymbol{D_5}$ = 01100110011110001111010101101,

$\boldsymbol{C_6}$ = 00110010101010111111111000011 i $\boldsymbol{D_6}$ = 10011001111000111101010101101,

$\boldsymbol{C_7}$ = 11001010101011111111100001100 i $\boldsymbol{D_7}$ = 01100111100011110101010110110,

$\boldsymbol{C_8}$ = 00101010101111111110000110011 i $\boldsymbol{D_8}$ = 10011110001111010101010111001,

$\boldsymbol{C_9}$ = 01010101011111111100001100110 i $\boldsymbol{D_9}$ = 00111100011110101010110110011,

$\boldsymbol{C_{10}}$ = 01010101111111110000110011001 i $\boldsymbol{D_{10}}$ = 11110001111010101010111001100,

$\boldsymbol{C_{11}}$ = 01010111111110000110011001011 i $\boldsymbol{D_{11}}$ = 11000111101010101011001100111,

$\boldsymbol{C_{12}}$ = 01011111111000011001100101011 i $\boldsymbol{D_{12}}$ = 00011110101010101100110011111,

$\boldsymbol{C_{13}}$ = 01111111100001100110010101011 i $\boldsymbol{D_{13}}$ = 01111010101010110011001111100,

$\boldsymbol{C_{14}}$ = 11111110000110011001010101011 i $\boldsymbol{D_{14}}$ = 11101010101011001100111100011,

$\boldsymbol{C_{15}}$ = 11111000011001100101010101111 i $\boldsymbol{D_{15}}$ = 10101010101100110011111000111,

$\boldsymbol{C_{16}}$ = 11110000110011001010101011111 i $\boldsymbol{D_{16}}$ = 01010101011001100111100011111.

A.2. Popis svih ključeva dobivenih permutacijom parova C_nD_n

U nastavku je dan popis svih ključeva koji se dobivaju iz parova C_nD_n korištenjem tablice permutacija X.1.3:

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

K_2 = 011110 011010 111011 011001 110110 111100 100111 100101

K_3 = 010101 011111 110010 001010 010000 101100 111110 011001

K_4 = 011100 101010 110111 010110 110110 110011 010100 011101

K_5 = 011111 001110 110000 000111 111010 110101 001110 101000

K_6 = 011000 111010 010100 111110 010100 000111 101100 101111

K_7 = 111011 001000 010010 110111 111101 100001 100010 111100

K_8 = 111101 111000 101000 111010 110000 010011 101111 111011

K_9 = 111000 001101 101111 101011 111011 011110 011110 000001

K_{10} = 101100 011111 001101 000111 101110 100100 011001 001111

K_{11} = 001000 010101 111111 010011 110111 101101 001110 000110

K_{12} = 011101 010111 000111 110101 100101 000110 011111 101001

K_{13} = 100101 111100 010111 010001 111110 101011 101001 000001

K_{14} = 010111 110100 001110 110111 111100 101110 011100 111010

K_{15} = 101111 111001 000110 001101 001111 010011 111100 001010

K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101.

A.3. Popis svih S-kutija

A.3.1. S_1 kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1.	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2.	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3.	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

A.3.2. S_2 kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1.	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2.	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3.	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

A.3.3. S_3 kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1.	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2.	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3.	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

A.3.4. S_4 kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1.	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2.	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3.	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

A.3.5. S_5 kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1.	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2.	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3.	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

A.3.6. S_6 kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1.	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2.	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3.	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

A.3.7. S_7 kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1.	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2.	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3.	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

A.3.8. S_8 kutija

retci\stupci	0.	1.	2.	3	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
0.	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1.	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2.	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3.	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

A.4. Popis svih koraka u petlji (t = 0 do 79) u SHA-1 algoritmu

A B C D E

t = 0: 0116FC33 67452301 7BF36AE2 98BADCFE 10325476

t = 1: 8990536D 0116FC33 59D148C0 7BF36AE2 98BADCFE

t = 2: A1390F08 8990536D C045BF0C 59D148C0 7BF36AE2

t = 3: CDD8E11B A1390F08 626414DB C045BF0C 59D148C0

t = 4: CFD499DE CDD8E11B 284E43C2 626414DB C045BF0C

t = 5: 3FC7CA40 CFD499DE F3763846 284E43C2 626414DB

t = 6: 993E30C1 3FC7CA40 B3F52677 F3763846 284E43C2

t = 7: 9E8C07D4 993E30C1 0FF1F290 B3F52677 F3763846

t = 8: 4B6AE328 9E8C07D4 664F8C30 0FF1F290 B3F52677

t = 9: 8351F929 4B6AE328 27A301F5 664F8C30 0FF1F290

t = 10: FBDA9E89 8351F929 12DAB8CA 27A301F5 664F8C30

t = 11: 63188FE4 FBDA9E89 60D47E4A 12DAB8CA 27A301F5

t = 12: 4607B664 63188FE4 7EF6A7A2 60D47E4A 12DAB8CA

t = 13: 9128F695 4607B664 18C623F9 7EF6A7A2 60D47E4A

t = 14: 196BEE77 9128F695 1181ED99 18C623F9 7EF6A7A2

t = 15: 20BDD62F 196BEE77 644A3DA5 1181ED99 18C623F9

t = 16: 4E925823 20BDD62F C65AFB9D 644A3DA5 1181ED99

t = 17: 82AA6728 4E925823 C82F758B C65AFB9D 644A3DA5

t = 18: DC64901D 82AA6728 D3A49608 C82F758B C65AFB9D

t = 19: FD9E1D7D DC64901D 20AA99CA D3A49608 C82F758B

t = 20: 1A37B0CA FD9E1D7D 77192407 20AA99CA D3A49608

t = 21: 33A23BFC 1A37B0CA 7F67875F 77192407 20AA99CA

t = 22: 21283486 33A23BFC 868DEC32 7F67875F 77192407

t = 23: D541F12D 21283486 0CE88EFF 868DEC32 7F67875F

t = 24: C7567DC6 D541F12D 884A0D21 0CE88EFF 868DEC32

t = 25: 48413BA4 C7567DC6 75507C4B 884A0D21 0CE88EFF

t = 26: BE35FBD5 48413BA4 B1D59F71 75507C4B 884A0D21

t = 27: 4AA84D97 BE35FBD5 12104EE9 B1D59F71 75507C4B

t = 28: 8370B52E 4AA84D97 6F8D7EF5 12104EE9 B1D59F71

t = 29: C5FBAF5D 8370B52E D2AA1365 6F8D7EF5 12104EE9

t = 30: 1267B407 C5FBAF5D A0DC2D4B D2AA1365 6F8D7EF5

t = 31: 3B845D33 1267B407 717EEBD7 A0DC2D4B D2AA1365

t = 32: 046FAA0A 3B845D33 C499ED01 717EEBD7 A0DC2D4B

t = 33: 2C0EBC11 046FAA0A CEE1174C C499ED01 717EEBD7

t = 34: 21796AD4 2C0EBC11 811BEA82 CEE1174C C499ED01

t = 35: DCBBB0CB 21796AD4 4B03AF04 811BEA82 CEE1174C

t = 36: 0F511FD8 DCBBB0CB 085E5AB5 4B03AF04 811BEA82

t = 37: DC63973F 0F511FD8 F72EEC32 085E5AB5 4B03AF04

t = 38: 4C986405 DC63973F 03D447F6 F72EEC32 085E5AB5

t = 39: 32DE1CBA 4C986405 F718E5CF 03D447F6 F72EEC32

t = 40: FC87DEDF 32DE1CBA 53261901 F718E5CF 03D447F6

t = 41: 970A0D5C FC87DEDF 8CB7872E 53261901 F718E5CF

t = 42: 7F193DC5 970A0D5C FF21F7B7 8CB7872E 53261901

t = 43: EE1B1AAF 7F193DC5 25C28357 FF21F7B7 8CB7872E

t = 44: 40F28E09 EE1B1AAF 5FC64F71 25C28357 FF21F7B7

t = 45: 1C51E1F2 40F28E09 FB86C6AB 5FC64F71 25C28357

t = 46: A01B846C 1C51E1F2 503CA382 FB86C6AB 5FC64F71

t = 47: BEAD02CA A01B846C 8714787C 503CA382 FB86C6AB

t = 48: BAF39337 BEAD02CA 2806E11B 8714787C 503CA382

t = 49: 120731C5 BAF39337 AFAB40B2 2806E11B 8714787C

t = 50: 641DB2CE 120731C5 EEBCE4CD AFAB40B2 2806E11B

t = 51: 3847AD66 641DB2CE 4481CC71 EEBCE4CD AFAB40B2

t = 52: E490436D 3847AD66 99076CB3 4481CC71 EEBCE4CD

t = 53: 27E9F1D8 E490436D 8E11EB59 99076CB3 4481CC71

t = 54: 7B71F76D 27E9F1D8 792410DB 8E11EB59 99076CB3

t = 55: 5E6456AF 7B71F76D 09FA7C76 792410DB 8E11EB59

t = 56: C846093F 5E6456AF 5EDC7DDB 09FA7C76 792410DB

t = 57: D262FF50 C846093F D79915AB 5EDC7DDB 09FA7C76

t = 58: 09D785FD D262FF50 F211824F D79915AB 5EDC7DDB

t = 59: 3F52DE5A 09D785FD 3498BFD4 F211824F D79915AB

t = 60: D756C147 3F52DE5A 4275E17F 3498BFD4 F211824F

t = 61: 548C9CB2 D756C147 8FD4B796 4275E17F 3498BFD4

t = 62: B66C020B 548C9CB2 F5D5B051 8FD4B796 4275E17F

t = 63: 6B61C9E1 B66C020B 9523272C F5D5B051 8FD4B796

t = 64: 19DFA7AC 6B61C9E1 ED9B0082 9523272C F5D5B051

t = 65: 101655F9 19DFA7AC 5AD87278 ED9B0082 9523272C

t = 66: 0C3DF2B4 101655F9 0677E9EB 5AD87278 ED9B0082

t = 67: 78DD4D2B 0C3DF2B4 4405957E 0677E9EB 5AD87278

t = 68: 497093C0 78DD4D2B 030F7CAD 4405957E 0677E9EB

t = 69: 3F2588C2 497093C0 DE37534A 030F7CAD 4405957E

t = 70: C199F8C7 3F2588C2 125C24F0 DE37534A 030F7CAD

t = 71: 39859DE7 C199F8C7 8FC96230 125C24F0 DE37534A

t = 72: EDB42DE4 39859DE7 F0667E31 8FC96230 125C24F0

t = 73: 11793F6F EDB42DE4 CE616779 F0667E31 8FC96230

t = 74: 5EE76897 11793F6F 3B6D0B79 CE616779 F0667E31

t = 75: 63F7DAB7 5EE76897 C45E4FDB 3B6D0B79 CE616779

t = 76: A079B7D9 63F7DAB7 D7B9DA25 C45E4FDB 3B6D0B79

t = 77: 860D21CC A079B7D9 D8FDF6AD D7B9DA25 C45E4FDB

t = 78: 5738D5E1 860D21CC 681E6DF6 D8FDF6AD D7B9DA25

t = 79: 42541B35 5738D5E1 21834873 681E6DF6 D8FDF6AD.

Literatura

[1.] Tipovi pametnih kartica – dostupno na internet adresi http://www.getoranged.co.za/sc_cardtypes.php (18.4.2004.)

[2.] Uvod u sigurnost pametnih kartica – dostupno na internet adresi http://home.hkstar.com/~alanchan/papers/smartCardSecurity (15.5.2004.)

[3.] Mogućnosti tehnologije s pametnih karticama – dokument u pdf formatu dostupan na internet adresi http://csrc.nist.gov/publications/nistir/IR-7056/Capabilities/Jun-SmartCardTech.pdf (19.6.2004.)

[4.] Kartice s dva sučelja – dostupno na internet adresi http://www.itsa.org/itsnews.nsf/0/2f9234cc1fe1f89d852568d60057cee8?OpenDocument (19.6.2004.)

[5.] Trostruki DES algoritam – specifikacija dostupna na internet adresi http://www.aci.net/kalliste/des.htm (19.6.2004.)

[6.] SHA-1 algoritam – specifikacija dostupna na internet adresi http://www.itl.nist.gov/fipspubs/fip180-1.htm (22.6.2004.)

[7.] RSA algoritam – specifikacija dostupna na internet adresi http://www.di-mgt.com.au/rsa_alg.html (25.6.2004.)

[8.] Z. Regvart, I. Krnić (Nove tehnologije d.o.o) : Tečaj PKI i PKI programiranje

[9.] Sigurnosna politika Schlumberger Cryptoflex 8k pametne kartice – dokument u pdf formatu dostupan na adresi http://csrc.nist.gov/cryptval/140-1/140sp/140sp206.pdf (11.8.2004.)

[10.] Team 1 Smart card project – dostupno na internet adresi http://www.a-game.net/report.htm (11.8.2004.)

[11.] Kerberos – specifikacija u pdf formatu dostupna na internet adresi http://www.sans.org/rr/papers/66/206.pdf (11.8.2004.)

[12.] Izvor informacija za PKCS #11 Wrapper i Provider – dostupno na internet adresi http://jce.iaik.tugraz.at (15.8.2004.)

[13.] Vrste pametnih kartica – dokument u pdf obliku dostupan na internet adresi http://www.otiglobal.com/pdf/The%20Smart%20Card%20Evolution.pdf (16.8.2004.)

[14.] Vrste pametnih kartica – dostupno na internet adresi http://www.javacard.org/others/smart_card.htm (16.8.2004.)

[15.] Baza podataka koja se koristi u aplikaciji – dostupno na internet adresi http://hsql.sourceforge.net/ (19.8.2004.)

[16.] Generator izvršnih (exe) datoteka koji se koristi u aplikaciji – dostupno na internet adresi http://jsmooth.sourceforge.net (19.8.2004.)