Primjena pametnih kartica u financijama i bankarstvu

Pihler, Viktorija

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO-MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Viktorija Pihler

PRIMJENA PAMETNIH KARTICA U FINANCIJAMA I BANKARSTVU

Diplomski rad

Zagreb, rujan 2019.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Viktorija Pihler

PRIMJENA PAMETNIH KARTICA U FINANCIJAMA I BANKARSTVU

Diplomski rad

Voditelj rada:
prof. dr. sc. Andrej Dujella

Zagreb, rujan 2019.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____ , predsjednik
2. _____ , član
3. _____ , član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

Prema Eurosmartu tržište pametnih kartica u 2019. godini premašit će 10 milijardi jedinica. A sve je počelo izdavanjem plastičnih kartica prije sedamdesetak godina koje su bile namijenjene isključivo bogatim pojedincima dopuštajući im da plaćaju svojim imenom. Danas su kartice sveprisutne u našim životima. Koristimo ih za identifikaciju, putovanja, pristup zgradama, dobivanje gotovine, plaćanje robe i usluga itd. Zbog toga nam je vrlo važno da su svi podaci na kartici sigurno pohranjeni i zaštićeni od manipulacije.

U prvom poglavlju ćemo objasniti kako je tekao razvoj pametnih kartica te razliku među određenim vrstama pametnih kartica. U tom dijelu korisitit ćemo tri knjige koje govore o pametnim karticama i njihovim primjenama: M. Hendry, *Smart Card Security and Applications* [3], K. E. Mayes, K. Markantonakis, *Smart Cards, Tokens, Security and Applications* [4] i W. Rankl, W. Effing, *Smart Card Handbook* [5]. Za objašnjavanje matematičke pozadine, koja nam je potrebna u nastavku rada, koristit ćemo literaturu iz teorije brojeva, A. Dujella, *Uvod u teoriju brojeva* [1]. Kako bismo opisali kriptografske algoritme, poslužit ćemo se literaturom iz kriptografije, A. Dujella, M. Maretić, *Kriptografija* [2]. Kada ćemo govoriti o ostalim informatičkim temeljima pametnih kartica (hash funkcijama, autentifikaciji, digitalnim potpisima i certifikatima), pretežno ćemo se oslanjati na knjigu W. Rankla i W. Effinga [5]. U trećem poglavlju, za opisivanje primjene pametnih kartica u financijama i bankarstvu, ponovno ćemo se služiti knjigama [3], [4] i [5].

Diplomski rad napravljen je u sklopu aktivnosti Projekta KK.01.1.1.01.0004 - Znanstveni centar izvrsnosti za kvantne i kompleksne sustave te reprezentacije Liejevih algebri.

Poglavlje 1

Pametne kartice

1.1 Razvoj pametnih kartica

Širenje plastičnih kartica počelo je u SAD-u početkom pedesetih godina prošlog stoljeća. Prvu potpuno plastičnu karticu za opću upotrebu izdao je Diners Club 1950. godine. Takvo plaćanje bilo je namijenjeno visokoj klasi pojedinaca, pa je služilo i kao statusni simbol, dopuštajući vlasniku da plaća svojim imenom umjesto gotovinom.

Ulazak glavnih platnih sustava, Vise i MasterCarda, u to područje doveo je do vrlo brzog širenja „plastičnog novca" u obliku kreditnih kartica. To se prvo dogodilo u SAD-u, a potom i u Europi te ostalim zemljama svijeta nekoliko godina kasnije.

U početku su funkcije kartica bile vrlo jednostavne. One su služile kao medij za pohranu podataka koji su na taj način bili sigurni od krivotvorenja. Na površini su bile ispisane opće informacije poput imena izdavača kartice, dok su elementi osobnih podataka, kao što su ime vlasnika kartice i broj kartice, bili utisnuti. Mnoge su kartice imale i područje za potpis gdje se nositelj kartice mogao potpisati za referencu. Upravo su taj potpis te zaštitni znakovi na kartici bile vizualne značajke koje su služile tim karticama prve generacije kao zaštita od krivotvorenja.

Prvo poboljšanje sastojalo se od magnetske trake na poleđini kartice koja je omogućavala pohranu digitalnih podataka u strojno čitljivom obliku. Nova metoda za identifikaciju korisnika koja je došla u opću upotrebu uključivala je tajni osobni identifikacijski broj - PIN (koji se uspoređivao s referentnim brojem). Međutim, tehnologija magnetske trake imala je ključnu slabost, a to je da su podaci pohranjeni na traci bili izloženi svakome tko je imao pristup potrebnoj opremi. Stoga takve kartice nisu bile prikladne za pohranu povjerljivih podataka. Ideja kojom bi se mogao riješiti taj problem sa sigurnosti je da se referentna vrijednost za PIN može pohraniti u terminal ili sustav domaćina u sigurnom okruženju umjesto na magnetskoj traci. Većina sustava koji bi upotrebljavali kartice s magnetskom trakom koristili bi internetske veze s računalom domaćina sustava radi sigurnosti, ali to bi stvaralo značajne troškove za potreban prijenos podataka. Kako bi se smanjili troškovi, bilo je potrebno pronaći rješenja koja omogućuju da se kartične transakcije izvršavaju izvan mreže bez ugrožavanja sigurnosti sustava. Razvoj pametne kartice, u kombinaciji sa širenjem sustava elektroničke obrade podataka, stvorio je potpuno nove mogućnosti za osmišljavanje takvih rješenja.

Ogroman napredak u mikroelektronici 1970-ih omogućio je integriraciju pohrane podataka i logike obrade podataka na jednom silikonskom čipu dimenzija svega nekoliko kvadratnih milimetara. Ideja o uvrštavanju takvog integriranog sklopa u identifikacijsku karticu bila je sadržana u prijavi patenta koju su podnijeli njemački izumitelji Jurgen Dethloff i Helmut Grotrupp već ranije 1968. Nakon toga je 1970. uslijedila slična prijava patenta Kunitake Arimure u Japanu. Međutim, prvi pravi napredak u razvoju pametnih kartica došao je kada je Roland Moreno registrirao patente svojih pametnih kartica u Francuskoj 1974. godine. Tek tada je industrija poluvodiča bila u mogućnosti ponuditi potrebne integrirane sklopove po prihvatljivim cijenama.

Veliki napredak postignut je 1984. godine kada je francuska agencija PTT (agencija poštanskih i telekomunikacijskih usluga) uspješno obavila terensko ispitivanje s telefonskim karticama. U tom ispitivanju pametne kartice su pokazale da ispunjavaju sva očekivanja, od visoke pouzdanosti do zaštite od manipulacije. Integrirani sklopovi koji su se koristili u telefonskim karticama bili su relativno mali, jednostavni i jeftini memorijski čipovi sa specifičnom sigurnosnom logikom koja je štitila od manipulacije. Mikroprocesorski čipovi, koji su bili znatno veći i složeniji, prvi su put korišteni u velikom broju u telekomunikacijama, posebice za mobilne telekomunikacije.

Bitno je za napomenuti da je razvoj moderne kriptografije bio jednako presudan za širenje bankovnih kartica kao i razvoj tehnologije poluvodiča. Pametna kartica pokazala se kao idealan medij za sigurnu pohranu tajnih ključeva i izvršavanje kriptografskih algoritama. Uz to, pametne kartice su bile toliko malene i lagane za korištenje u svakodnevnom životu. Stoga je prirodna ideja bila pokušati koristiti ove nove sigurnosne značajke za bankovne kartice kako bi se suočili sa sigurnosnim rizicima koji su se javljali kod kartica s magnetskom vrpcom.

Važna prekretnica za buduću upotrebu pametnih kartica u svijetu bilo je dovršavanje EMV specifikacije, koja je plod triju najvećih organizacija kreditnih kartica: Europaya, MasterCarda i Vise. Prva verzija ove specifikacije objavljena je 1994. godine. Sadržavala je detaljne opise kreditnih kartica s mikroprocesorskim čipovima i njome je bila zajamčena međusobna kompatibilnost budućih pametnih kartica.

Dakle, pametnu karticu možemo definirati preko sljedećih značajki:

- ima jedinstven identifikator
- može sudjelovati u automatiziranoj elektroničkoj transakciji
- koristi se prvenstveno za povećanje sigurnosti
- nije ju lako krivotvoriti ili kopirati
- može sigurno pohraniti podatke
- može pokrenuti niz sigurnosnih algoritama i funkcija

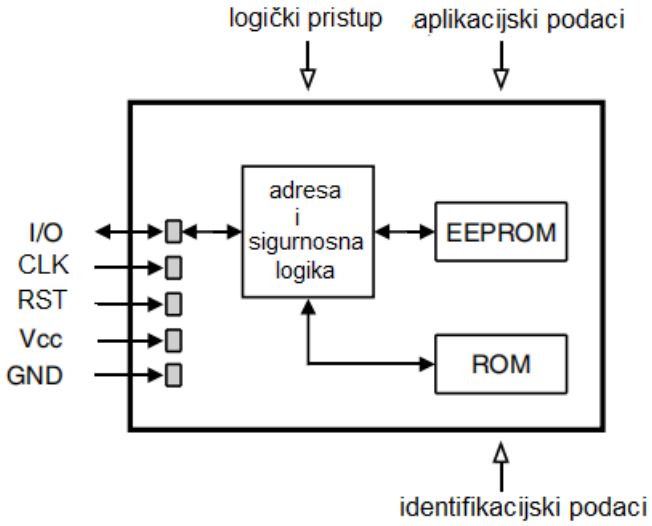
1.2 Podjela pametnih kartica

Pametne kartice dijelimo u dvije skupine (koje se razlikuju po funkcionalnosti i po cijeni):

1. memorijske kartice
2. mikroprocesorske kartice

Također ćemo spomenuti još jednu posebnu skupinu pametnih kartica, a to su beskontaktne kartice.

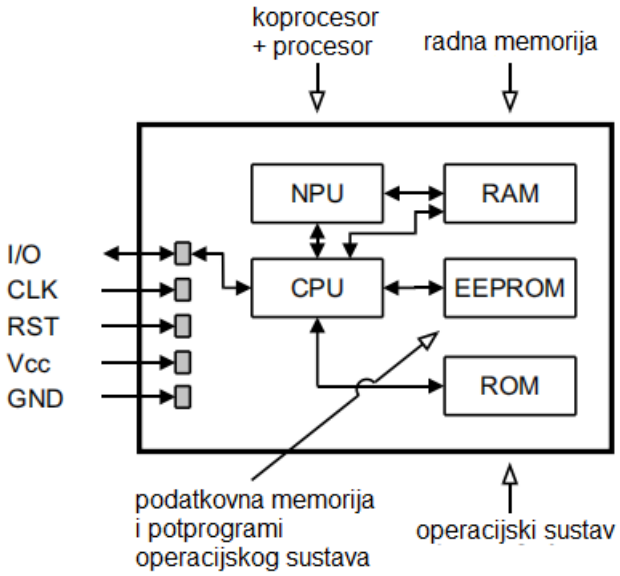
Memorijske kartice pohranjuju podatke (potrebne za određenu primjenu) u memoriji koja je obično EEPROM (*eng. Electrically Erasable Programmable Read-Only Memory*). Pristup memoriji kontrolira se sigurnosnom logikom koja se u najjednostavnijem slučaju sastoji od zaštite od pisanja ili zaštite od brisanja memorije. Međutim, postoje i memorijski čipovi složenije sigurnosne logike koji također mogu izvoditi jednostavnu enkripciju, tj. šifriranje. Preko I/O ulaza podaci se prenose na i sa kartice. Funkcionalnost memorijskih kartica obično je optimizirana za određenu primjenu. Time je ograničena fleksibilnost tih kartica, ali je zato njihova cijena jako povoljna. Memorijske kartice obično se koriste za pretplatne telefonske kartice i kartice zdravstvenog osiguranja.



Slika 1.1 Tipična arhitektura memorijske kartice kontaktnog tipa sa sigurnosnom logikom

Mikroprocesorske kartice sadrže čip čiji je glavni dio procesor. On je obično okružen s četiri dodatna funkcionalna bloka: mask ROM-om, EEPROM-om, RAM-om i I/O ulazom. Mask ROM sadrži operacijski sustav čipa te je njegov sadržaj identičan za sve čipove u proizvodnoj seriji i on se ne može mijenjati tijekom života čipa. EEPROM je nepromjenjiva memorija čipa čiji se podaci i programski kod mogu upisati i pročitati uz kontrolu

operacijskog sustava. RAM je radna memorija procesora. Ona je promjenjiva pa se svi podaci pohranjeni u njoj gube kad se čip isključi. Sučelje I/O ulaza obično se sastoji od jednog registra preko kojeg se podaci prenose bit po bit. Mikroprocesorske kartice vrlo su fleksibilne za upotrebu. U najjednostavnijem slučaju, one sadrže program optimiziran za jednu primjenu, ali moderni operativni sustavi pametnih kartica omogućuju nekoliko različitih primjena integriranih u jednu karticu.



Slika 1.2 Tipična arhitektura kontaktne mikroprocesorske kartice s koprocesorom

Beskontaktne kartice javile su se kao potreba zbog čestih kvarova u elektromehaničkim sustavima čiji su izvori nerijetko bili kontakti. Kod njih je mikrosklop potpuno zapečaćen unutar kartice i komunicira s vanjskim svijetom kroz antenu. Napajanje se može vršiti pomoću te iste antene ili preko baterije. Velika prednost beskontaktnih kartica je u tome što ne moraju nužno biti umetnute u čitač kartica jer postoje dostupni sustavi koji rade u rasponu do jednog metra. Također, beskontaktna tehnologija je povoljna u sustavima koji zahtijevaju svjesno umetanje kartice u čitač jer nije važan položaj kartice pri umetanju. Jedna od glavnih primjena ove tehnologije je javni prijevoz gdje je potrebno identificirati veliki broj ljudi u najkraćem mogućem vremenu. Proizvodna tehnologija za masovnu proizvodnju beskontaktnih kartica sazrela je do granice da su visokokvalitetni proizvodi dostupni po cijenama koje se ne razlikuju značajno od kontaktnih kartica.

Poglavlje 2

Informatički temelji pametnih kartica

2.1 Matematička podloga

Definicija 2.1.1. *Cijeli broj b djeljiv je cijelim brojem a* $(a \neq 0)$ *(odnosno a dijeli b), ako postoji cijeli broj x takav da je* $b = a \cdot x$*, te to zapisujemo kao* $a \mid b$*. U slučaju da b nije djeljiv s a, to zapisujemo* $a \nmid b$*.*

Definicija 2.1.2. *Prirodan broj* $n > 1$ *zove se prost broj (ili prim broj) ukoliko je djeljiv samo s brojem 1 i samim sobom. Broj koji nije prost naziva se složen broj.*

Definicija 2.1.3. *Neka su a i b cijeli brojevi razlličiti od nule. Neka je c cijeli broj. Broj c zovemo zajednički djelitelj brojeva a i b ako vrijedi* $c \mid a$ *i* $c \mid b$*. Nadalje c zovemo najveći zajednički djelitelj od a i b ako je c najveći cijeli broj koji ih dijeli (u oznaci* $c = (a,b)$ *ili* $c = nzd(a,b)$*)*

Definicija 2.1.4. *Dva prirodna broja n i m zovemo relativno prostim brojevim ukoliko im je najveći zajednički djelitelj broj 1.*

Definicija 2.1.5. *Neka je funkcija* $\varphi : \mathbb{N} \to \mathbb{N}$ *dana sljedećim pravilom:*

$$\varphi(n) = \mid \{a \in \{1,2,...,n\} : nzd(a,n) = 1\} \mid$$

Funkcija φ *zove se Eulerova funkcija.*

Propozicija 2.1.6. *Eulerova funkcija zadovoljava sljedeća svojstva:*
(a) Neka je p prost broj. Tada vrijedi:

$$\varphi(p) = p - 1$$

(b) Neka je $n = p \cdot q$*, gdje su p i q prosti brojevi. Tada vrijedi:*

$$\varphi(n) = \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1)$$

Dokaz. (a) slijedi direktno iz definicije prostog broja.
Kako bi dokazali *(b)* uočimo da zato što su p i q prosti brojevi, jedini brojevi koji nisu relativno prosti s brojem n bit će višekratnici brojeva p i q manji od n, a to su redom:

$$0, 1 \cdot p, 2 \cdot p, ..., (q-1) \cdot p, 1 \cdot q, 2 \cdot q, ..., (p-1) \cdot q$$

te ih je točno $1 + (q-1) + (p-1) = p + q - 1$. Tu se vidi da relativno prostih brojeva s n, a koji su manji od n ima točno $n - (p+q-1) = p \cdot q - p - q + 1 = (p-1) \cdot (q-1)$, pa je

$$\varphi(n) = (p-1) \cdot (q-1) = \varphi(p) \cdot \varphi(q)$$

□

Definicija 2.1.7. *Ako cijeli broj m, $(m \neq 0)$ dijeli razliku $a-b$, kažemo da je a kongruentan b modulo m i zapisujemo to $a \equiv b \pmod{m}$. Ako $a-b$ nije djeljivo sa m, tada kažemo da a nije kongruentan b modulo m i zapisujemo to $a \not\equiv b \pmod{m}$.*

Teorem 2.1.8 (Eulerov teorem)**.** *Ako su $a \in \mathbb{N}$ i $n \in \mathbb{Z}$ relativno prosti, tada je*

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

Teorem 2.1.9 (Mali Fermatov teorem)**.** *Neka je p prost broj. Ako $p \nmid a$, onda je $a^{p-1} \equiv 1 \pmod{p}$. Za svaki cijeli broj a vrijedi $a^p \equiv a \pmod{p}$.*

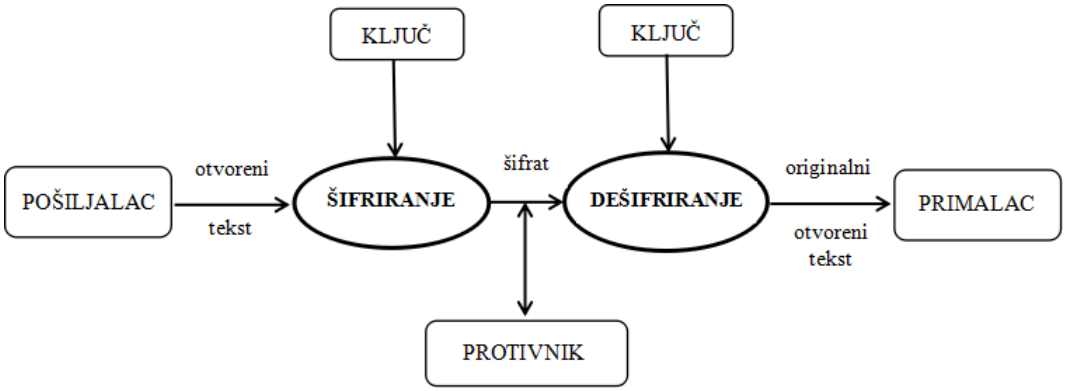
2.2 Kriptografija

Kriptografija je znanstvena disciplina koja se bavi proučavanjem metoda za slanje poruka u takvom obliku da ih samo onaj kome su namijenjene može pročitati. Sama riječ kriptografija je grčkog podrijetla i mogla bi se doslovno prevesti kao tajnopis.

Osnovni zadatak kriptografije je omogućiti dvjema osobama (zvane pošiljalac i primalac) komuniciranje preko nesigurnog komunikacijskog kanala na način da treća osoba (njihov protivnik), koja može nadzirati komunikacijski kanal, ne može razumjeti njihove poruke. Poruku koju pošiljalac želi poslati primaocu zvat ćemo otvoreni tekst. Pošiljalac transformira otvoreni tekst koristeći unaprijed dogovoreni ključ. Taj postupak se naziva šifriranje, a dobiveni rezultat šifrat ili kriptogram. Nakon toga pošiljalac pošalje šifrat preko nekog komunikacijskog kanala. Protivnik prisluškujući može doznati sadržaj šifrata, ali ne može odrediti otvoreni tekst. Za razliku od njega, primalac koji zna ključ kojim je šifrirana poruka može dešifrirati šifrat i odrediti otvoreni tekst.

Definicija 2.2.1. *Kriptosustav je uređena petorka (P, C, K, E, D) za koju vrijedi:*

1. *P je konačan skup svih mogućih osnovnih elemenata otvorenog teksta;*
2. *C je konačan skup svih mogućih osnovnih elemenata šifrata;*
3. *K je prostor ključeva, tj. konačan skup svih mogućih ključeva;*
4. *Za svaki $K \in K$ postoji funkcija šifriranja $e_K \in E$ i odgovarajuća funkcija dešifriranja $d_K \in D$. Pritom su $e_K : P \to C$ i $d_K : C \to P$ funkcije sa svojstvom da je $d_K(e_K(x)) = x$ za svaki otvoreni tekst $x \in P$.*



Slika 2.1 Model kriptosustava

Ovisno o načinu korištenja ključa postoje simetrični algoritmi kriptiranja i asimetrični algoritmi kriptiranja. Osnovna razlika je u tome da simetrični algoritmi koriste isti ključ za šifriranje i dešifriranje neke poruke, dok asimetrični algoritmi koriste različite ključeve za šifriranje i dešifriranje iste. Naime, asimetrični algoritmi sadrže javni ključ (dostupan svima i svi mogu pomoću njega šifrirati poruke) i privatni ključ (dostupan samo osobi koja ga posjeduje i pomoću njega dešifrira poruke).

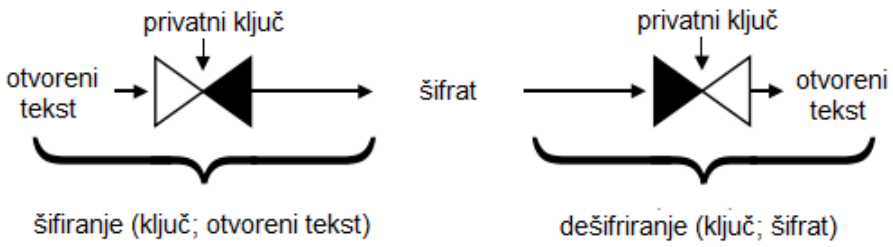
Uglavnom su simetrični algoritmi po svojoj prirodi brži, tj. implementacija na računalu se brže odvija od implementacije asimetričnih algoritama. Međutim, kod asimetričnih algoritama nema potrebe za sigurnim komunikacijskim kanalom za razmjenu ključeva (kao što je to slučaj kod simetričnih). U praksi se obje vrste algoritama isprepleću u cilju bolje zaštite poruka. Obično se asimetrični algoritmi koriste za šifriranje slučajno generiranog broja koji služi kao ključ za šifriranje originalne poruke metodama simetričnih algoritama. Ovo se naziva hibridna enkripcija.

U nastavku ćemo pobliže objasniti kako funkcioniraju simetrični algoritmi DES (*eng. Data Encryption Standard*) i Trostruki DES, te asimetrični algoritmi RSA (nazvan prema svojim tvorcima Ronaldu Rivestu, Adi Shamiru i Leonardu Adlemanu) i DSA (*eng. Digital Signature Algorithm*).

DES

Kao što smo već rekli, simetrični kriptografski algoritmi temelje se na načelu izvođenja šifriranja i dešifriranja istim tajnim ključem - otuda je i naziv „simetrični”.

Najpoznatija i najčešće korištena vrsta simetričnog kriptografskog algoritma je DES. Razvio ga je IBM u suradnji s NBS-om (*eng. US National Bureau of Standards*) i 1977. godine objavljen je kao američki standard FIPS 46 . DES algoritam je bio najkorišteniji simetrični kriptosustav do kraja 20. stoljeća, a nakon toga ga zamjenjuju Trostruki DES i AES (*eng. Advanced Encryption Standard*). Algoritam je zasnovan na tzv. Feistelovoj šifri. Gotovo svi simetrični blokovni algoritmi koji su danas u uporabi koriste ideju koju je



Slika 2.2 Šifriranje i dešifriranje koristeći simetrični algoritam

uveo Horst Feistel 1973. godine. Jedna od glavnih ideja je alternirana uporaba supstitucija i transpozicija kroz više iteracija (tzv. rundi).

Ključevi za DES algoritam mogu se generirati korištenjem generatora slučajnih brojeva koji proizvodi 8-bajtni slučajni broj koji se zatim uspoređuje s četiri slaba i dvanaest poluslabih ključeva. Ako se izračunata vrijednost ne podudara s nekim od ovih lako razbijenih ključeva, izračunaju se paritetni bitovi i rezultat je DES ključ. Prostor ključeva je veličine 2^{56} što znači da postoji otprilike $7.2 \cdot 10^{16}$ mogućih ključeva. Na prvi se pogled to možda čini kao veliki broj, međutim upravo je veličina prostora ključeva glavna slabost DES algoritma.

Feistelova šifra

Feistelova šifra iterativna je šifra koja preslikava blok otvorenog teksta duljine $2t$ bita u šifrat. Otvoreni tekst podijeljen je u dvije grupe (tj. bloka) (L_0, R_0) gdje svaki od blokova ima po t bita. Dva bloka otvorenog teksta (L_0, R_0) šifriraju se u šifrat (R_r, L_r) kroz iterativni postupak od r runda, gdje je $r \geq 1$. Za svaki $1 \leq i \leq r$, tijekom i-te runde preslikava se $(L_{i-1}, R_{i-1}) \mapsto (L_i, R_i)$ na sljedeći način:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \bigoplus f(R_{i-1}, K_{i-1})$$

gdje je svaki potključ K_i dobiven iz ključa K te gdje $\bigoplus$ označava operaciju „ekskluzivno ili" (`xor`). Operacija $\bigoplus$ zapravo predstavlja zbrajanje u Z_2. Nakon što se završi izvršavanje svih r rundi Feistelove šifre dobiva se blok (L_r, R_r), no zbog lakšeg dešifriranja uzima se blok (R_r, L_r) kao konačni rezultat Feistelove šifre.

Šifriranje

DES je Feistelova šifra koja kao ulaz prima otvoreni tekst (točnije jedan blok otvorenog teksta) i ključ. Duljina bloka otvorenog teksta i ključa je $n = 64$, a kao izlaz algoritam daje šifrat također duljine 64 bita. Ključ K je dugačak 64 bita, no njegov dio koji se koristi

u algoritmu je duljine 56 bita. Preostalih 8 bitova $(8,16,...,56,64)$ mogu se koristiti kao kontrolni bitovi.
Označimo s $O = (o_1 o_2 ... o_{64})$ blok otvorenog teksta koji treba šifrirati.
Algoritam je sljedeći:

- **Početna permutacija:**
  Bitovi otvorenog teksta ispermutiraju se početnom permutacijom **IP**.

$$\mathbf{IP}(O) = (o_{\mathbf{IP}(1)} ... o_{\mathbf{IP}(64)})$$

- **16 runda Feistelove šifre:**
  Nakon permutiranja bloka, blok se dijeli na dva početna bloka (L_0, R_0), gdje je $L_0 = (o_{\mathbf{IP}(1)} ... o_{\mathbf{IP}(32)})$, a $R_0 = (o_{\mathbf{IP}(33)} ... o_{\mathbf{IP}(64)})$ koji služe kao ulaz u Feistelovu šifru. Zatim se u svakoj rundi ponavlja sljedeći postupak (u oznakama za i-tu rundu):

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \bigoplus f(R_{i-1}, K_{i-1})$$

  Funkcija f ima dva ulaza, prvi je desni od dobivenih blokova iz prošle runde (duljine 32 bita), dok je drugi 48-bitni potključ K_i generiran za i-tu rundu. Funkcija f kao izlaz daje niz od 32 bita. Nakon 16 rundi Feistelove šifre i mijenjanja poretka blokova dobivamo rezultat (R_{16}, L_{16}).

- **Inverz početne permutacije:**
  Blok (R_{16}, L_{16}) ispermutira se inverzom početne permutacije $\mathbf{IP}^{-1}(\mathbf{IP}^{-1}(R_{16}, L_{16}))$ te se dobiva šifrat bloka otvorenog teksta.

IP							
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

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

Slika 2.3 Tablice permutacije

Generiranje potključeva

Neka je s $K = (k_1 k_2 ... k_{64})$ označen početni ključ te neka su s $K_i = (k_{i,1} k_{i,2} ... k_{i,48})$ označeni potključevi koji se generiraju za svaku rundu iz početnog ključa K, $i \in \{1, ..., 16\}$ s obzirom da algoritam ima 16 rundi. Postupak generiranja je gotovo isti za svaku rundu, a samo generiranje ovisi o dvije permutacije: **PC1** (*eng. Permuted Choice 1*) i **PC2** (*eng. Permuted Choice 2*).

PC1						
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

C

D

PC2					
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

Slika 2.4 Tablice permutacije izbora

Permutacija **PC1** podijeljena je na dva dijela (**C** i **D**) te ima 56 elementa. Kao što je već spomenuto efektivni dio ključa nisu 64 bita, već njih 56 pa se pomoću **PC1** eleminiraju točno ti suvišni, kontrolni bitovi (vidimo da $\mathbf{PC1}(y) \notin \{8, 16, 24, 32, 40, 48, 56, 64\}$). U svakoj rundi generiraju se nova dva bloka (C_i, D_i), gdje je svaki od njih duljine 28 bita. Označimo s $C_i = (c_{i,1} c_{i,2} ... c_{i,27} c_{i,28})$, gdje je $c_{i,j}$ j-ti bit u i-toj rundi bloka. Analogno definiramo D_i. Kao početno stanje definiramo $C_0 = (k_{\mathbf{PC1}(1)} ... k_{\mathbf{PC1}(28)})$ i $D_0 = (k_{\mathbf{PC1}(29)} ... k_{\mathbf{PC1}(56)})$. Blokove C_1 i D_1 dobivamo s cikličkim pomakom bitova za jedno mjesto ulijevo blokova C_0 i D_0 respektivno, tj.

$$C_1 = (c_{1,1} c_{1,2} ... c_{1,27} c_{1,28}) = (c_{0,2} c_{0,3} ... c_{0,28} c_{0,1})$$
$$D_1 = (d_{1,1} d_{1,2} ... d_{1,27} d_{1,28}) = (d_{0,2} d_{0,3} ... d_{0,28} d_{0,1})$$

Analogno vrijedi za svaku rundu, (C_i, D_i) dobiva se iz (C_{i-1}, D_{i-1}) cikličkim pomakom bitova ulijevo. U svakoj rundi rade se dva pomaka ulijevo, osim u prvoj, drugoj, devetoj i šesnaestoj rundi kada se radi jedan pomak ulijevo. Kada izgeneriramo blokove C_i i D_i za i-tu rundu, gledamo na njih kao na cjelinu, tj. na blok $H = h_1 h_2 ... h_{55} h_{56}$. Potključ K_i dobivamo iz bloka H i permutacije izbora 2

$$K_i = \mathbf{PC2}(H) = (h_{\mathbf{PC2}(1)}, ..., h_{\mathbf{PC2}(56)}) = (k_{i,1}, k_{i,2}, ..., k_{i,48})$$

Funkcija f

Funkcija f u i-toj rundi kao parametre prima R_i i K_i, gdje je R_i blok od 32 bita, a K_i blok od 48 bita. Ulaskom u funkciju prvo se blok R_i proširi na 48 bita pomoću preslikavanja **E** (funkcija odabira, $\mathbf{E}(R_i) = r_{i,1}...r_{i,48}$).

E					
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

Slika 2.5 Tablice odabira bitova **E**

Nakon proširivanja bloka R_i, prošireni blok zbrajamo s potključem generiranim za i-tu rundu i dobivamo novi blok B od 48 bita. Taj blok dijeli se na 8 blokova od po 6 bita koje označavamo s B_j:

$$B = B_1B_2...B_8 = \mathbf{E}(R_i) \bigoplus K_i$$

Svaki od blokova B_j proslijeđujemo $\mathbf{S}_j$-kutiji. Svaka **S**-kutija može se smatrati funkcijom koja prima 6 bita i vraća 4 bita. Također je svaka **S**-kutija definirana matricom dimenzija 4×16 koja sadrži elemente iz skupa $\{0, 1, ..., 15\}$.

Neka je $B = (b_1, b_2, b_3, b_4, b_5, b_6)$ proizvoljan blok od 6 bita koji ulazi u neku **S**-kutiju. Bitove iz B grupiramo u dvije grupe, tj. zapišemo kao uređen par $(b_1b_6, b_2b_3b_4b_5)$. Uvijek grupiramo prvi i zadnji bit te srednja četiri. b_1b_6 u bazi 2 čine broj između 0 i 3, dok $b_2b_3b_4b_5$ u bazi 2 čine broj između 0 i 15. Ako redove u promatranoj **S**-kutiji numeriramo redom s 0,1,2,3, a stupce radom s 0,1,...,15, tada bi uređen par $(b_1b_6, b_2b_3b_4b_5)$ označavao točno jedan redak i točno jedan stupac, tj. jedan element promatrane **S**-kutije. Kako je već navedeno u **S**-kutijama nalaze se elementi iz skupa $\{0, 1, ..., 15\}$ koji se svi mogu zapisati u 4 bita. Taj element (odn. tih 4 bita) je izlaz iz **S**-kutije za ulaznih 6 bita. Svaka od 8 **S**-kutija vraća 4 bita što je ukupno 32 bita. Označimo s B'_j 4-bitni blok koji vraća $\mathbf{S}_j$-kutija. Nakon **S**-kutija dobiva se blok $B' = B'_1B'_2...B'_8$. Dobiveni B' ispermutiramo prema tablici **P** te se taj blok $\mathbf{P}(B')$ vraća kao izlaz funkcije f.

Gornjim postupkom opisana je jedna runda Feistelove šifre unutar DES-a. Nakon 16 runda DES završava te se za ulazni blok otvorenog teksta dobiva blok šifrata.

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	12	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
S_2	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
S_3	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
S_4	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
S_5	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
S_6	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	5	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
S_7	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
S_8	0	13	2	8	4	6	15	11	1	10	9	3	14	15	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

Slika 2.6 S-kutije

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

Slika 2.7 Tablica permutacije **P**

Dešifriranje

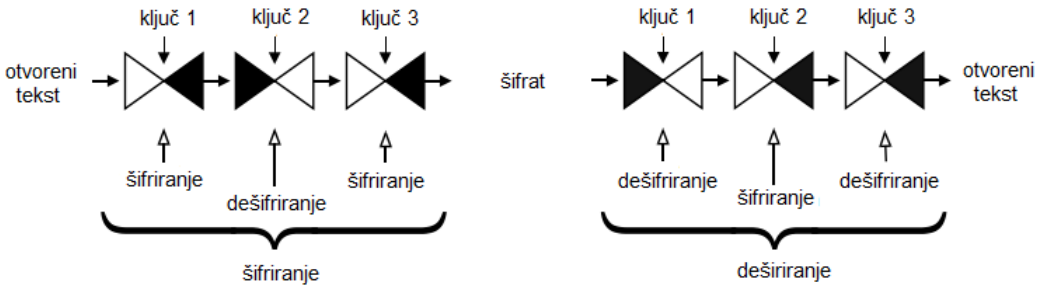
Dešifriranje DES-a potpuno je analogno šifriranju s jedinom razlikom u obrnutom raspo-

redu potključeva. Šifrirani blok stavimo kao ulaz u algoritam, te kao ključ u prvoj rundi stavlja se K_{16}, u drugoj rundi kao ključ stavlja se K_{15} itd. dok se u zadnjoj rundi kao ključ stavlja K_1. Kao izlaz dobiva se blok otvorenog teksta od kojeg je dobiven šifrirani blok.

Iako je sam opis DES-a dosta dug, on se može vrlo efikasno implementirati, i hardverski i softverski. Jedna važna primjena DES-a je u bankarskim transakcijama. Tako se, između ostalog, DES koristio za šifriranje PIN-ova , te transakcija preko bankomata. DES je također bio u širokoj uporabi u civilnim satelitskim komunikacijama.

Trostruki DES

Trostruka DES enkripcija je postupak šifriranja izvornih podataka koristeći DES algoritam tri puta. Potrebna su tri 56-bitna ključa umjesto samo jednog, s tim da su prvi i treći ključ obično isti. Tako se dobiva ključ duljine 2×56 bita. To znači da su u ovom postupku podaci kompatibilni s uobičajenim DES algoritmom i da ne nameću nikakve dodatne troškove osim dvostruke veličine ključa. To je jedan od glavnih razloga široke uporabe trostrukog DES-a u pametnim karticama. Primarno se koristi za generiranje ključeva i zaštitu vrlo osjetljivih podataka (primjerice kod prijenosa ključeva zbog poboljšane razine sigurnosti u odnosu na običan DES).

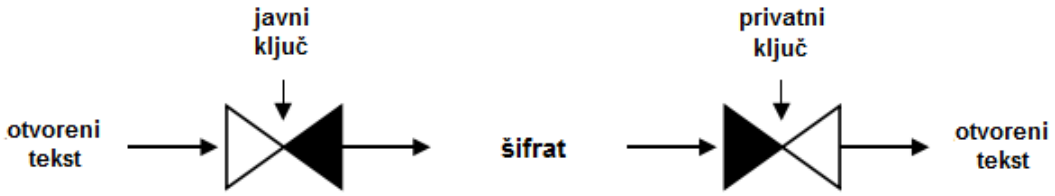


Slika 2.8 Shema trostrukog DES-a

Razlog zašto dvostruki DES nije u upotrebi je to što se pokazalo da dvije instance DES-a povećavaju sigurnost algoritma za samo jedan bit. Algoritam trostrukog DES-a značajno je sigurniji od dvostrukog DES-a jer napad "susret u sredini" nije učinkovit protiv ove metode.

RSA

Whitfield Diffie i Martin E. Hellman su 1976. opisali ideju razvoja algoritma šifriranja baziranu na dva različita ključa. Na taj način se željelo eliminirati problem povezan s razmjenom i distribucijom tajnih simetričnih ključeva. Također bi po prvi puta bili mogući procesi poput generiranja digitalnih potpisa (o kojima će više riječi biti u dijelu 2.5).



Slika 2.9 Šifriranje i dešifriranje koristeći asimetrični algoritam

Prvi, a ujedno i najpopularniji i najširenji kriptosustav s javnim ključem je RSA kriptosustav koji su izumili Ron Rivest, Adi Shamir i Len Adleman 1977. godine. Njegova sigurnost je zasnovana na težini faktorizacije velikih prirodnih brojeva.

Javni i privatni ključevi su obično različite duljine. Privatni ključ bi trebao biti što duži jer se time sprečavaju pokušaji probijanja koda, za razliku od javnog ključa za koji se preferira da je kraći jer se tako može značajno smanjiti vrijeme za potvrdu digitalnog potpisa. Trenutno su u uporabi ključevi duljine 1024 ili 2048 bitova. Količina računalnog napora potrebnog za šifriranje i dešifriranje raste približno eksponencijalno s dužinom ključa. Uz brojeve 7 i 17, četvrti Fermatov broj ($2^{16}+1 = 65537$) se često koristi kao javni ključ.

Generiranje ključeva:

1. Nasumično odabrati dva velika prosta broja p i q.

2. Izračunati vrijednost $n = p \cdot q$ i $\varphi(n)$, gdje je φ Eulerova funkcija (odnosno broj brojeva u nizu $1,2,...,n$ koji su relativno prosti s n).
   U ovom slučaju vrijedi:

$$\varphi(n) = \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1) = n - p - q + 1$$

3. Odabrati broj $e < n$ relativno prost s brojem $(p-1) \cdot (q-1)$ te izračunati njegov multiplikativni inverz modulo $(p-1) \cdot (q-1)$, tj. modulo $\varphi(n)$.
   Ako označimo multiplikativni inverz s d, tada vrijedi:

$$d \cdot e \equiv 1 \pmod{\varphi(n)}$$

Brojevi $[e,n]$ formiraju javni ključ koji bi trebao biti javno dostupan, dok brojevi $[d,p,q]$ formiraju privatni ključ koji bi trebao biti dostupan samo osobi koja je vlasnik odgovarajućeg javnog ključa.

Šifriranje i dešifriranje:

Pretpostavimo da pošiljatelj šalje poruku primatelju s javnim ključem $[e,n]$. RSA algoritam ne radi s bitovima i bajtovima, već s prirodnim brojevima modulo n pa tako otvoreni tekst X prikažemo kao niz brojeva modulo n $(x_1,x_2,...,x_m)$. Šifriranje otvorenog teksta svodi se na potenciranje brojeva otvorenog teksta s vrijednošću e modulo n. Neka je x jedan od brojeva otvorenog teksta X, tada je šifrat y jednak:

$$y = x^e \bmod n$$

Šifrat y je broj manji od n te se on šalje primatelju. Primatelj posjeduje privatni ključ d te se postupak dešifriranja svodi na potenciranje šifrata s privatnim ključem, tj.

$$x = y^d \bmod n = (x^e)^d \bmod n = x^{e\cdot d} \bmod n$$

Sada ćemo objasniti zašto vrijedi jednakost $x = x^{e\cdot d} \bmod n$. Prema Eulerovom teoremu znamo da vrijedi $x^{\varphi(n)} = 1 \bmod n$. Također znamo da vrijedi $e \cdot d = 1 \bmod \varphi(n)$. Iz toga slijedi da je $e \cdot d - 1 = t \cdot \varphi(n)$, za neki cijeli broj t. Sada imamo

$$x^{e\cdot d} = x^{(e\cdot d-1)+1} = x \cdot x^{e\cdot d-1} = x \cdot x^{t\cdot\varphi(n)}$$

No kako je $x^{\varphi(n)} = 1 \bmod n$, tako je i $x^{k\cdot\varphi(n)} = (x^{\varphi(n)})^k = 1^k \bmod n$ pa vrijedi

$$x \cdot x^{k\cdot\varphi(n)} = x \cdot 1^k = x \bmod n$$

odnosno vrijedi tvrdnja:

$$x = x^{e\cdot d} \bmod n.$$

Digitalni potpis i autentifikacija:

Kao što smo ranije spomenuli, RSA algoritam koristi se i prilikom generiranja digitalnog potpisa. To je sigurnosni mehanizam kojim možemo potvrditi autentičnost pošiljatelja. Pretpostavimo da pošiljatelj ima svoj javni ključ $[e_1,n_1]$ i privatni ključ $[d_1,n_1]$, te da primatelj ima javni ključ $[e_2,n_2]$ i privatni ključ $[d_2,n_2]$. Neka je m otvoreni tekst koji pošiljatelj želi poslati primatelju. Prema RSA algoritmu pošiljatelj šifrira tekst javnim ključem primatelja, no ukoliko pošiljatelj želi ostaviti svoj digitalni potpis postupak je sljedeći:

- Pošiljatelj komprimira otvoreni tekst x koristeći hash funkciju H (više o njoj u dijelu 2.3) :

$$h = H(x)$$

- Pošiljatelj potpisuje otvoreni tekst svojim privatnim ključem (postupak analogan šifriranju kod RSA) :

$$s = h^{d_1} \bmod n_1$$

- Pošiljatelj šalje primatelju šifrirani otvoreni tekst y i potpisani otvoreni tekst s.

Kako bi se primatelj uvjerio u autentičnost pošiljatelja, poduzima sljedeće korake:

- Primatelj dešifrira otvoreni tekst svojim privatnim ključem te računa hash vrijednost dobivene poruke (gdje je važno da hash funkcija bude identična onoj koja je korištena za generiranje potpisa) :

$$h = H(x)$$

- Primatelj dešifira potpisani otvoreni tekst javnim ključem pošiljatelja :

$$h_1 = s^{e_1 \bmod n_1}$$

- Potpis će biti valjan ukoliko vrijedi :

$$h_1 = h$$

U praksi je generiranje ključeva napornije jer je za velike brojeve vrlo teško odrediti jesu li prosti. Zbog toga se koriste određeni testovi kako bi se utvrdila vjerojatnost da je odabrani broj prost. Miller-Rabin test i Solovay-Strassen test tipični su primjeri takvih testova.

Iako je algoritam RSA vrlo siguran, zbog dugog vremena računanja rijetko se koristi za šifriranje podataka. Prvenstveno se koristi u području digitalnih potpisa gdje se prednosti asimetričnog postupka mogu u potpunosti realizirati. Velika prednost RSA algoritma je to što nije ograničen duljinom ključa (kao npr. algoritam DES), ali je zato količina memorijskog prostora potrebna za pohranu ključa u pametnim karticama dosta velika.

DSA

DSA je algoritam za digitalno potpisivanje odobren od strane DSS (*eng. Digital Signature Standard*) standarda savezne vlade Sjedinjenih Američkih Država. Koriste ga sve vladine organizacije te sve nevladine tvrtke i organizacije koje surađuju s vladom. Algoritam je

1991. godine razvio David Kravitz, bivši agent u NSA – u (*eng. National Security Agency*) i može se koristiti besplatno.

Uz DSA i RSA algoritme, koji su dvije najčešće korištene procedure za generiranje digitalnih potpisa, postoji i ECDSA algoritam (*Elliptic Curve Digital Signature Algorithm*) koji je sličan kao DSA, ali koristi eliptičke krivulje. Za razliku od RSA algoritma, sigurnost DSA algoritma ne ovisi o problemu faktorizacije velikih brojeva, već o problemu diskretnog logaritma. Izraz $y = a^x \bmod p$ može se izračunati brzo, čak i s velikim brojevima. Međutim obrnuti proces, koji izračunava vrijednost x za dane vrijednosti y, a i p, zahtijeva vrlo veliku količinu računarskog napora.

Kod svih algoritama za digitalno potpisivanje prvo se poruka, koja se treba potpisati, mora svesti na unaprijed definiranu duljinu pomoću hash funkcije. Stoga je objavljena odgovarajuća funkcija za upotrebu s DSA algoritmom. Ta funkcija naziva se SHA-1 (*eng. Secure Hash Algorithm*) te generira 160-bitnu hash vrijednost za poruku proizvoljne duljine.

Generiranje ključeva:

1) Generiranje parametara

- Odabrati odobrenu kriptografsku hash funkciju H (u originalnom DSA-u preporuka je bila koristiti SHA-1, ali u trenutnom DSS-u odobrena je i upotreba SHA-2)
- Odabrati duljine ključeva L i N, gdje se oznake odnose na duljine ključeva u bitovima, a ne na nazive samih ključeva. Prvi DSS je uvjetovao da L bude broj između 512 i 1024 (uključujući i granične vrijednosti) te da bude djeljiv sa 64. Duljina N mora biti manja ili jednaka duljini izlaza iz funkcije H.
- Odabrati prosti broj p duljine L.
- Odabrati 160-bitni prosti faktor od $(p-1)$.
- Odabrati proizvoljan broj h takav da vrijedi $1 < h < p-1$.
- Izračunati g takav da vrijedi $g = h^{(p-1)/q}$, gdje je $g > 1$ (ako se dobije $g = 1$, treba odabrati drugačiju vrijednost h).

2) Izračun privatnog i javnog ključa

- Odabrati privatni ključ x jednom od nasumičnih metoda tako da vrijedi $0 < x < q$.
- Izračunati javni ključ $y = g^x \bmod p$

Potpisivanje poruke:

Poruka m se potpisuje uz pomoć hash funkcije na sljedeći način:

- Odabrati nasumični broj k takav da vrijedi $1 < k < q$.

- Izračunati $r = (g^k \bmod p) \bmod q$.
- Izračunati vrijednost *hash* funkcije *H(m)*.
- Izračunati $s = k^{-1}(H(m) + x \cdot r) \bmod q$.
- Ponoviti postupak s drugačijim k ako su $s = 0$ ili $r = 0$.

Dobivene vrijednosti r i s su digitalni potpis poruke. Dio algoritma koji zahtijeva najviše resursa jest modularno eksponenciranje (računanje javnog ključa). Međutim ta vrijednost se može izračunati i prije nego poruka prođe kroz funkciju H. Modularni inverz koji se javlja pri računu vrijednosti s je drugi najzahtjevniji korak po pitanju korištenja resursa, a računa se pomoću proširenog Euklidovog algoritma ili Malog Fermatovog teorema kao $k^{q-2} \bmod q$.

Provjera autentičnosti poruke:

- Provjeriti vrijede li izrazi $0 < r < q$ i $0 < s < q$. Ukoliko samo jedan od njih nije zadovoljen, potpis se smatra nevažećim, tj. neispravnim.
- Izračunati $w = s^{-1} \bmod q$.
- Izračunati $u_1 = H(m) \cdot w \bmod q$.
- Izračunati $u_2 = (r \cdot w) \bmod p$.
- Izračunati $v = (g^{u_1} \cdot y^{u_2} \bmod p) \bmod q$.
- Potpis je valjan ako vrijedi $v = s$, u protivnom je nevaljan.

U praksi je RSA algoritam postigao širu upotrebu od DSA algoritma, koji je do sada imao vrlo ograničenu upotrebu. Razlog tome leži u složenosti DSA algoritma te u činjenici da se ne može koristiti za šifriranje.

2.3 Hash funkcije

Hash funkcije za ulaz uzimaju poruku koja je uglavnom fiksne duljine i za izlaz daju šifriranu poruku poznatiju kao hash vrijednost ili jednostavno hash. Preciznije rečeno, hash funkcija *h* pridružuje nizovima znakova proizvoljne konačne duljine nizove znakova fiksne duljine, od npr. *n* bita. Hash funkcije služe kako bi se što više podigla razina sigurnosti podataka.

Definicija 2.3.1. *Hash funkcija (u najširem smislu) je funkcija h koja zadovoljava iduća dva svojstva:*

1. kompresija - h ulazu x proizvoljne konačne duljine pridružuje izlaz h(x) fiksne duljine n

2. jednostavnost izračuna - zadani su h i ulaz x, h(x) je lako izračunati

Osnovno svojstvo svih efikasnih hash funkcija je zahtjev da za bilo koja dva izračunata različita hasha i ulazi iz kojih su oni izračunati moraju biti različiti. Drugo svojstvo efikasnih hash funkcija bi bilo to da za dva izračunata ista hasha ulazi iz kojih su izračunati ne moraju biti isti. Ako izračunamo hash vrijednost za jedan ulaz, a nakon toga ulazu promijenimo samo jedan bit, tada bi novi izračunati hash trebao biti potpuno različit od prethodnog.

Navest ćemo još neka dodatna svojstva za hash funkcije s ulazima *x, x'* te izlazima *y,y'*:

- jednosmjernost - za unaprijed određene izlaze računski je neizvedivo pronaći neki ulaz koji je hashiran u taj izlaz, tj. pronaći original *x'* takav da je $h(x') = y$ za neki y za koji odgovarajući ulaz nije poznat.
- jednoznačnost ili slaba otpornost na koliziju - računski je neizvedivo pronaći bilo koji drugi ulaz koji ima isti izlaz kao određeni ulaz, odnosno, za dani x pronaći drugi original $x' \neq x$ takav da je $h(x) = h(x')$.
- općenita jednoznačnost ili jaka otpornost na koliziju - računski je neizvedivo pronaći bilo koja dva različita ulaza x i *x'* koji hashirani daju isti izlaz, tj. takvi da je $h(x) = h(x')$.

Pronalaženje dva dokumenta s istom hash vrijednošću nije tako teško kao što se možda čini. Tu možemo iskoristiti poznati problem u teoriji vjerojatnosti zvan paradoks rođendana. Ovaj paradoks uključuje dva pitanja:

1. Koliko ljudi mora biti u sobi da vjerojatnost bude veća od 50%, a da pritom jedna od tih osoba ima isti rođendan kao i osoba postavljajući pitanje?
2. Koliko ljudi mora biti u sobi da vjerojatnost bude veća od 50%, a da pritom dvije osobe imaju isti rođendan?

Odgovor na prvo pitanje se lako može pronaći jer je potrebno samo usporediti rođendan ispitivača s rođendanom svih ostalih u sobi. Tako se dobije da najmanje 183 (365 : 2) osobe moraju biti u sobi.

Drugo pitanje otkriva paradoks ili, još bolje, iznenađujući rezultat ove usporedbe. Odgovor je samo 23 osobe. Razlog tome je da, iako su prisutne samo 23 osobe, to čini ukupno 253 para za usporedbu rođendana. Vjerojatnost da dvoje ljudi imaju isti rođendan temelji se upravo na tim parovima.

Upravo se ovaj paradoks koristi u napadu na hash funkciju. Mnogo je lakše stvoriti dva dokumenta koja imaju istu hash vrijednost nego izmijeniti dokument dok on ne poprimi određenu hash vrijednost. Posljedica toga je da rezultati hash funkcija moraju biti dovoljno veliki kako bi uspješno spriječili napade. Stoga većina hash funkcija ima vrijednosti s duljinom od najmanje 128 bita.

Standard ISO/IEC 10118-2 određuje hash funkciju koja se temelji na n-bitnom blok šifriranju (npr. DES). Uz ovaj algoritam duljina hash vrijednosti može biti n ili $2n$ bita. MD4

(*eng. Message Digest 4*) hash funkciju i njezinog nasljednika MD5 objavio je Ronald L. Rivest između 1990. i 1991. Obje funkcije temelje se na samostalnom algoritmu i generiraju 128-bitnu hash vrijednost. Godine 1992. NIST (*eng. National Institute of Standards and Technology*) je objavio hash funkciju za DSA algoritam poznatu pod nazivom SHA (*eng. Secure Hash Algorithm*). Nakon otkrića određenih slabosti funkcija je modificirana, a rezultat je funkcija poznata od sredine 1995. kao SHA-1. Godine 2002. američki stručnjak Ronald Rivest dizajnirao je poboljšanje SHA-1 pod nazivom SHA-2, a 2015. je NIST objavio najnovije poboljšanje SHA-3.

U području pametnih kartica ove se funkcije koriste isključivo za izračunavanje ulaznih vrijednosti za digitalne potpise. Budući da je prijenos podataka na pametne kartice općenito spor, hash funkcija je pohranjena u terminalu ili u računalu spojenom na terminal. Osim toga, u većini slučajeva, ograničenja memorije sprječavaju pohranjivanje hash funkcija na kartice.

2.4 Autentifikacija

Svrha autentifikacije je provjera identiteta i autentičnosti komunikacijskog partnera. Prevedeno u svijet pametnih kartica, to znači da kartica ili terminal određuje je li njezin komunikacijski partner autentičan terminal, odnosno autentična pametna kartica.

Autentifikacija zahtijeva da strane koje komuniciraju dijele zajedničku tajnu koja se može potvrditi postupkom provjere autentičnosti. Takav je postupak značajno sigurniji nego čisti postupak identifikacije, kao što je PIN test.

Razlikujemo statičku i dinamičku autentifikaciju. U statičkoj se za provjeru autentičnosti uvijek koriste isti (statički) podaci. Nasuprot tome konstruirana je dinamička autentifikacija kako bi se zaštitilo od napada unosom podataka snimljenih tijekom prethodne sesije.

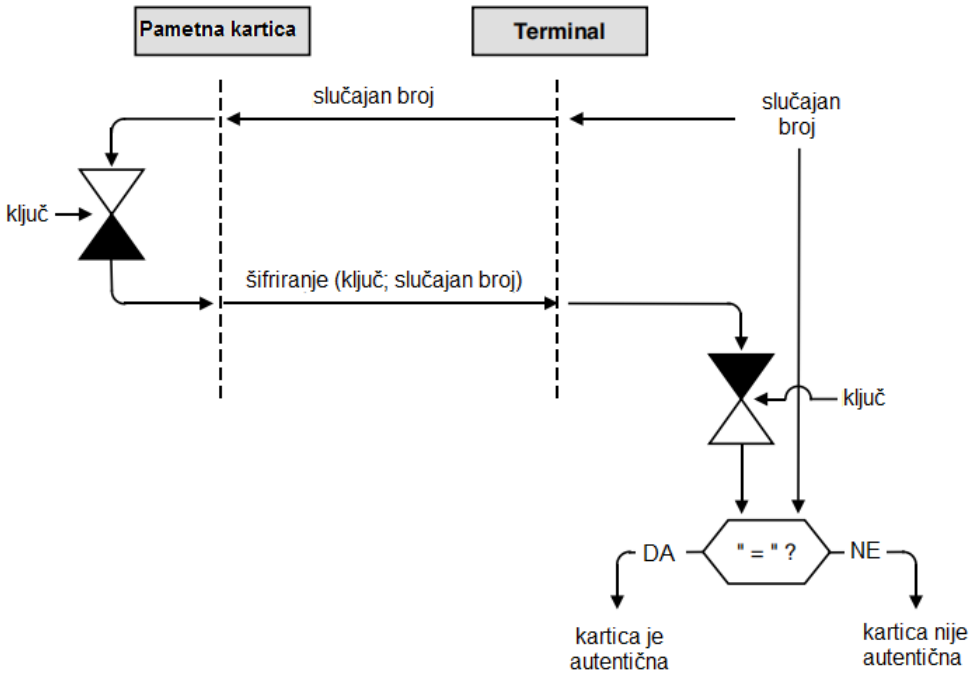
Autentifikacija se također dijeli i na jednostranu i uzajamnu. Jednostrana provjera autentičnosti, ako je uspješna, utvrđuje autentičnost jednog od dva komunikacijska partnera, a uzajamna utvrđuje autentičnost oba komunikacijska partnera.

Postupci autentifikacije temelje se na kriptografskim algoritmima, stoga se mogu podijeliti na one sa simetričnim i one s asimetričnim algoritmima. Zbog brzine izvršavanja pametne kartice uglavnom koriste simetrične algoritme.

Jednostrana autentifikacija sa simetričnim algoritmom

Jednostrana autentifikacija u komunikacji služi kako bi jedna strana potvrdila autentičnost druge strane. Da bi to bilo moguće, obje strane moraju imati tajni ključ za algoritam šifriranja. Cjelokupna sigurnost postupka autentifikacije ovisi o tom ključu. Kada bi ključ postao poznat, napadač bi se mogao autentificirati jednako lako kao originalan komunikacijski partner.

Terminal generira slučajni broj i šalje ga pametnoj kartici. Pametna kartica šifrira taj slučajni broj pomoću ključa poznatog njoj i terminalu. Sigurnost postupka ovisi o tom ključu budući da samo onaj koji posjeduje ključ može stvoriti ispravan odgovor koji će biti



Slika 2.10 Proces jednostrane autentifikacije sa simetričnim algoritmom

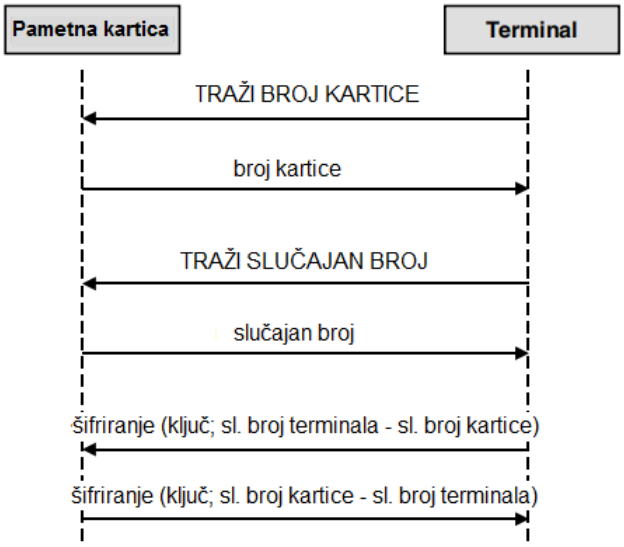
poslan terminalu. Kartica tada vraća rezultat šifriranja terminalu. Terminal koristi tajni ključ za dešifriranje kriptiranog slučajnog broja koji je primio, a zatim uspoređuje rezultat sa slučajnim brojem koji je izvorno poslan. Ako se dva broja podudaraju, tada terminal zna da je pametna kartica autentična.

Uzajamna autentifikacija sa simetričnim algoritmom

Princip uzajamne autentifikacije temelji se na dvostrukoj jednostranoj provjeri autentičnosti.

Prije nego što terminal može izračunati ključ za autentifikaciju (koji je specifičan za svaku karticu), prvo treba broj kartice. Nakon što terminal primi taj broj, on izračunava taj specifičan ključ. Potom traži od kartice da mu pošalje slučajni broj i istodobno stvara i sam slučajni broj. Terminal zatim zamjenjuje dva slučajna broja i spaja ih nakon čega šifrira dobiveni broj koristeći ključ za autentifikaciju. Napokon, dobiveni šifrat šalje kartici. Razlog preokretanja nasumičnih brojeva omogućuje razlikovanje izazova (ono što terminal šalje) i odgovora (ono što terminal prima).

Kartica može dešifrirati primljeni blok i provjeriti odgovara li slučajni broj, koji je prethodno poslala terminalu, broju koji je zauzvrat dobila. Ako je to slučaj, pametna kartica zna da terminal posjeduje tajni ključ. Time se potvrđuje autentičnost terminala. Zatim pametna kartica zamijeni dva slučajna broja, kriptira rezultirajući broj pomoću tajnog ključa i dobiveni šifrat šalje natrag terminalu.



Slika 2.11 Proces uzajamne autentifikacije sa simetričnim algoritmom

Terminal dešifrira primljeni blok i uspoređuje slučajni broj koji je prethodno poslao kartici s onim koji je dobio zauzvrat. Ako se podudaraju, pametna kartica je autentična. Time se završava postupak uzajamne autentifikacije.

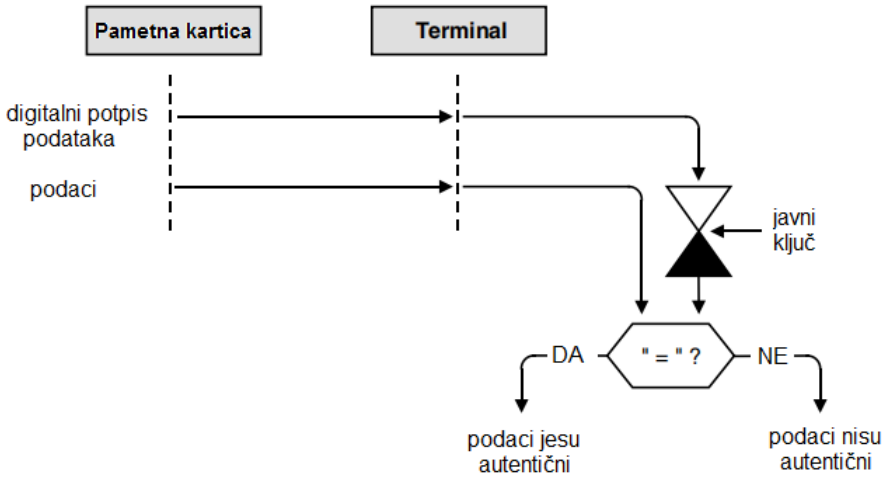
Statička autentifikacija s asimetričnim algoritmom

Uz statički postupak ne postoji zaštita od ponavljanja prethodnih podataka. To je razlog zašto se koristi samo kao dopunska provjera autentičnosti kartice koja je provjerena dinamičkim postupkom sa simetričnim algoritmom.

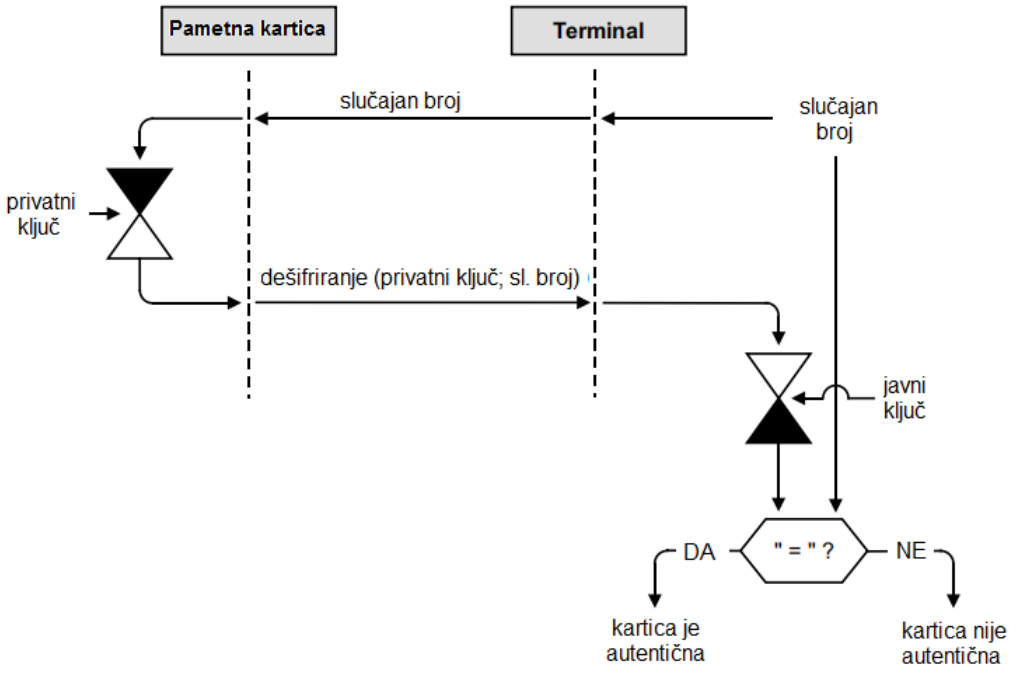
Postupak u osnovi funkcionira na sljedeći način. Tijekom personalizacije pametne kartice, na nju se unose podaci specifični za tu karticu. Na primjer to može biti broj kartice kao i ime i adresa vlasnika kartice. Te se informacije ne mijenjaju tijekom „života" kartice. Kao dio personalizacije kartice izračunava se digitalni potpis ovih podataka pomoću tajnog ključa. Ovaj ključ se koristi globalno u sustavu. Kada se kartica koristi na terminalu, terminal čita digitalni potpis i potpisane podatke iz datoteke na kartici. Terminal ima javni ključ koji vrijedi za sve kartice u sustavu i može ga koristiti za šifriranje potpisa koji je pročitao te usporediti rezultat s podacima koje je pročitao s kartice. Ako se ove dvije vrijednosti podudaraju, terminal je provjerio autentičnost kartice.

Dinamička autentifikacija s aimetričnim algoritmom

Prethodno opisan statički postupak ima određene nedostatke. To se može eliminirati tako da autentifikacija postane dinamička što pruža zaštitu od ponovnog unosa podataka iz ranijih sesija. Uobičajena praksa je korištenje slučajnog broja kao ulazne vrijednosti za asimetrični algoritam.



Slika 2.12 Proces jednostrane statičke autentifikacije s asimetričnim algoritmom



Slika 2.13 Proces jednostrane dinamičke autentifikacije s asimetričnim algoritmom

Kao i kod autentifikacije sa simetričnim algoritmom terminal generira slučajni broj i šalje ga pametnoj kartici. Kartica dešifrira slučajni broj pomoću privatnog ključa, a zatim

šalje rezultat natrag u terminal. Terminal drži globalni javni ključ i tim ključem šifrira primljeni slučajni broj. Ako je rezultat ovog izračuna isti kao slučajni broj koji je prethodno poslan na karticu, karticu je autentificirana.

2.5 Digitalni potpisi

Za uspostavljanje autentičnosti elektronički prenesenih poruka ili elektroničkih dokumenata koriste se digitalni potpisi. Provjera potpisa može se koristiti za utvrđivanje je li poruka ili dokument izmijenjen. Bitna karakteristika digitalnog potpisa jest da samo jedna osoba ili jedna pametna kartica može „potpisati" dokument, ali svatko može provjeriti je li potpis autentičan. Time se prirodno nameće primjena asimetričnih kriptografskih algoritama.

Poruka ili dokument koji treba potpisati obično je dugačak najmanje nekoliko tisuća bajtova. Kako bi se smanjilo vrijeme izračuna šifrata koristi se ranije opisana hash funkcija. Tom jednosmjernom funkcijom vrši se kompresija podataka. Ova kompresija nije reverzibilna što znači da se originalni podaci ne mogu rekonstruirati iz komprimiranih podataka. Budući da je izračunavanje hash vrijednosti vrlo brzo, hash funkcije idealna su pomoć pri računanju digitalnog potpisa.

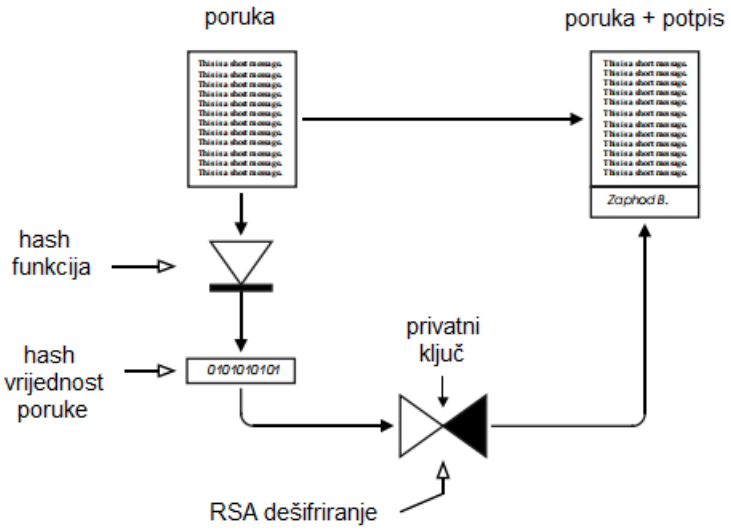
Postoje dvije vrste digitalnog potpisa s obzirom kako se dodaje poruci. Prvi je digitalni potpis s dodatkom. Njegova je prednost to što se poruka može u potpunosti pročitati bez potrebe za prethodnom potvrdom potpisa. Međutim, nedostatak je to što se veličina poruke povećava za duljinu potpisa. Taj se nedostatak može izbjeći korištenjem druge vrste digitalnog potpisa, a to je digitalni potpis s obnovom poruke. U ovoj metodi prvo se hash vrijednost stvarne poruke dodaje poruci, a potom se na kraju rezultirajućeg niza podataka formira ulazni blok za algoritam digitalnog potpisa. To znači da se veličina digitalno potpisane poruke povećava samo za duljinu hash vrijednosti, ali se ne može u potpunosti pročitati sve dok digitalni potpis nije provjeren.

Postupak generiranja digitalnog potpisa s dodatkom može se vrlo lako prikazati. Prvo se hash funkcijom dobije hash vrijednost iz sadržaja poruke. Ta vrijednost se dešifrira pomoću asimetričnog kriptografskog algoritma (npr. RSA). Rezultat je stvarni potpis koji se dodaje poruci.

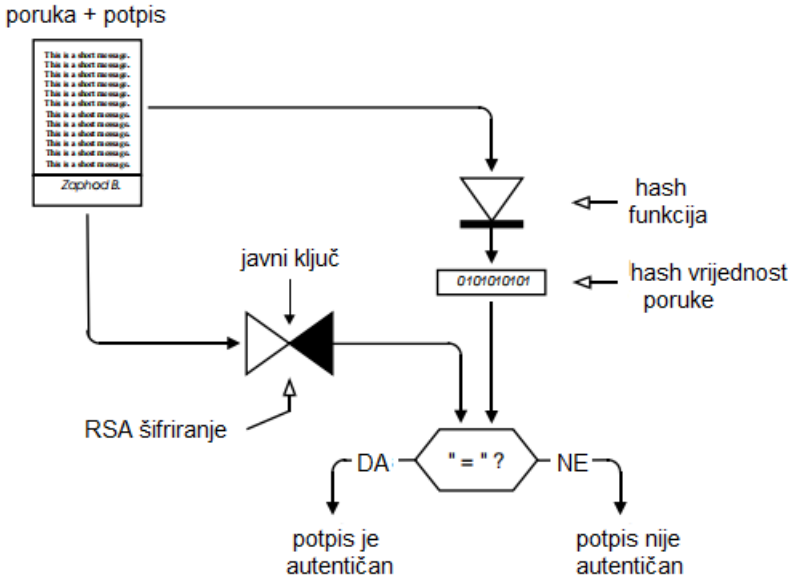
Potpisana poruka sada se šalje primatelju. Primatelj odvaja potpis od poruke i zatim komprimira poruku koristeći istu hash funkciju. Digitalni potpis se šifrira pomoću javnog ključa RSA algoritmom i taj rezultat se uspoređuje s hash vrijednosti. Ako su obje vrijednosti iste, to znači da poruka nije prepravljena tijekom prijenosa.

Zadatak pametne kartice u ovom primjeru je vrlo jednostavan. Ona pohranjuje privatni RSA ključ i dešifrira hash vrijednost koja se dobije komprimiranjem poruke, tj. to znači da kartica generira potpis. Sve ostalo, poput stvaranja hash vrijednosti ili naknadne provjere potpisa, u načelu se može podjednako dobro obavljati putem računala.

Ipak, idealna bi situacija bila da pametna kartica primi poruku preko svog sučelja, izračuna hash vrijednost i zatim potpisanu poruku vrati natrag u terminal. Provjera potpisa mogla bi se također obaviti pomoću pametne kartice. Ovaj postupak nije ništa sigurniji od stvaranja potpisa, ali je zato značajno jednostavniji za primjenu. To je zato što se hash



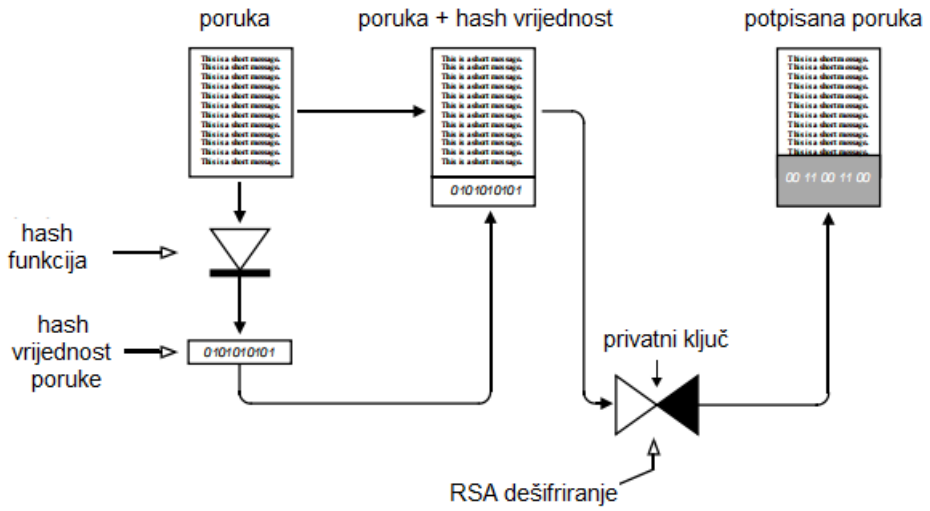
Slika 2.14 Potpisivanje poruke digitalnim potpisom s dodatkom



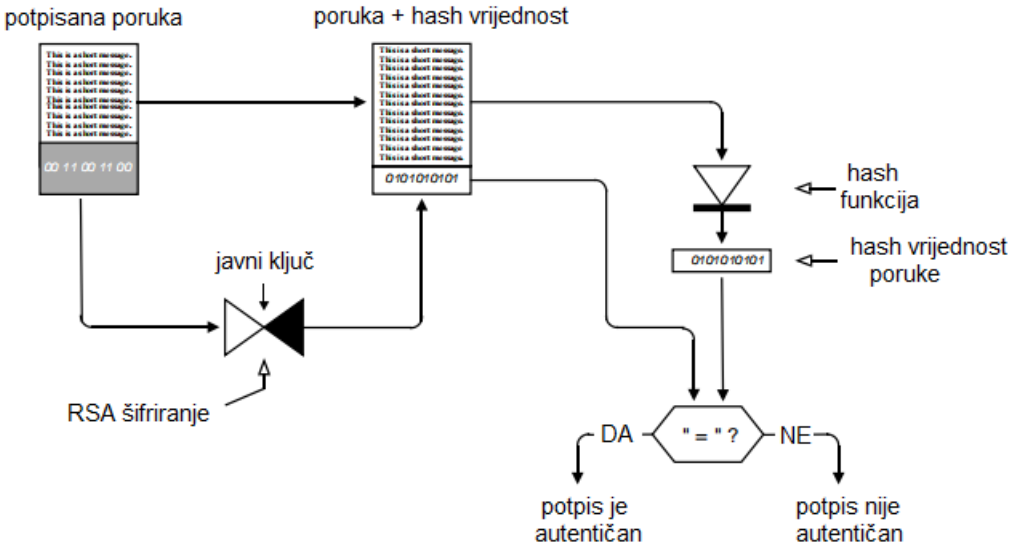
Slika 2.15 Provjera poruke koja je potpisana digitalnim potpisom s dodatkom

algoritmi i RSA ključevi mogu promijeniti zamjenom pametne kartice bez potrebe za izmjenom programa ili podataka na računalu.

Osim RSA algoritma, za dobivanje digitalnog potpisa koristi se i ranije opisani DSA algoritam koji je specifično razvijen za tu primjenu.



Slika 2.16 Potpisivanje poruke digitalnim potpisom s obnovom poruke



Slika 2.17 Provjera poruke koja je potpisana digitalnim potpisom s obnovom poruke

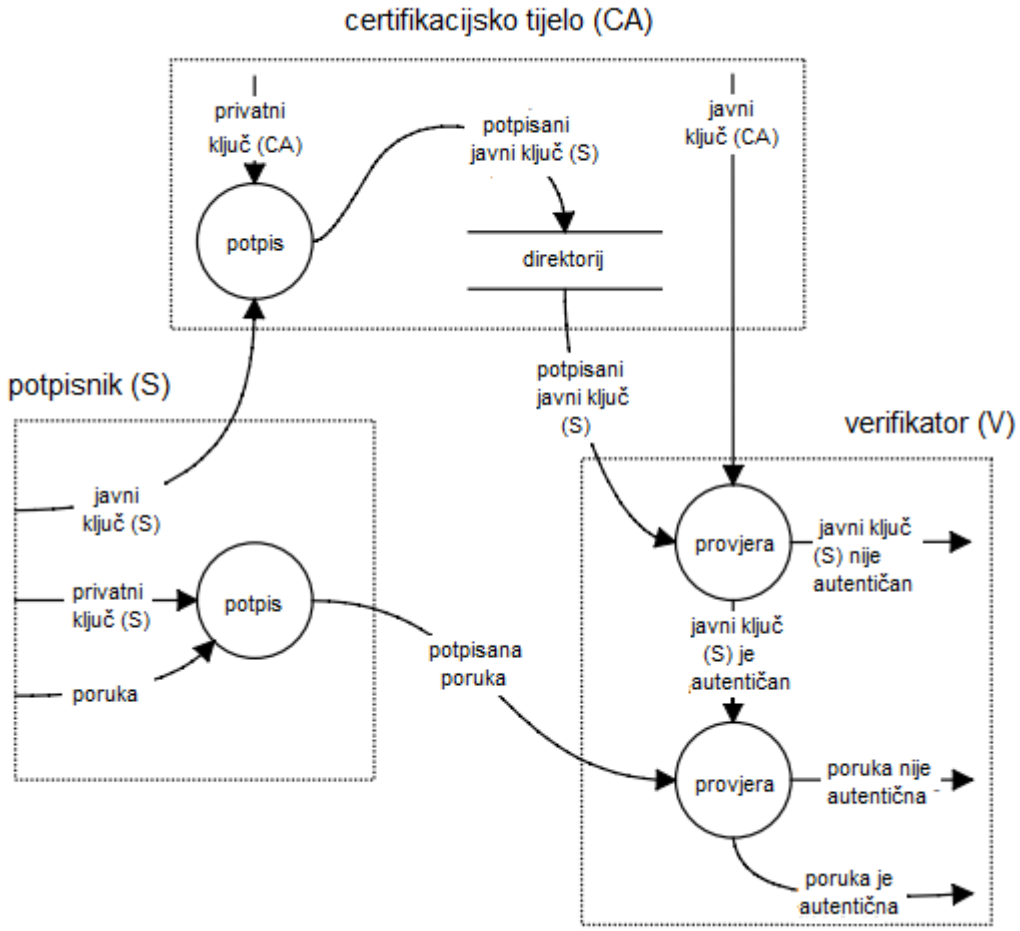
2.6 Certifikati

Prilikom upotrebe digitalnog potpisa javlja se jedan problem. Naime, svi koji žele provjeriti digitalni potpis poruke trebaju imati odgovarajući javni ključ. No javni ključ se ne može poslati bez zaštite jer primatelj inače ne može provjeriti autentičnost ključa. Kako bi se provjerila autentičnost javnog ključa, pouzdano tijelo mora potpisati taj ključ. To tijelo se zove CA (*eng. Certification Authority*). Kombinacija javnog ključa (koji je potpi-

san od strane CA), digitalnog potpisa i nekih dodatnih parametara (npr. serijski broj, ime izdavatelja) zove se certifikat.

U ovaj je proces uključeno i drugo tijelo, a to je TC (*eng. Trust Center*). TC generira i upravlja certifikatima te ima mogućnost generiranja ključeva za kartice s digitalnim potpisom. U pravilu TC i održava javni direktorij certifikata, tako da svi koji žele provjeriti potpisanu poruku mogu zatražiti povezani potpisani javni ključ iz centra (npr. putem interneta).

Certifikat ne sadrži samo potpisani javni ključ već i veliki broj dodatnih parametara i opcija. Iz ovoga proizlazi da algoritmi koji generiraju hash vrijednosti i digitalne potpise moraju biti jasno definirani. Zbog toga postoje standardi koji određuju strukturu certifikata, a jedan od napoznatijih je X.509. Tipičan X.509 certifikat u pametnim karticama obično je veličine od otprilike 1 kB.



Slika 2.18 Dijagram toka podataka osnovnih procesa za generiranje i provjeru poslane poruke pomoću certifikata

Poglavlje 3

Primjena pametnih kartica u bankarstvu i financijama

3.1 Kreditne i debitne kartice

Postoje tri osnovna modela za elektroničko plaćanje pametnim karticama:

(a) kreditne kartice kod kojih je plaćanje izvršeno nakon pružanja usluge („platite kasnije")

(b) debitne kartice kod kojih se plaćanje obavlja prilikom pružanja usluge („platite sada")

(c) elektronički novčanici u kojima se plaćanje obavlja prije pružanja usluge („platite prije").

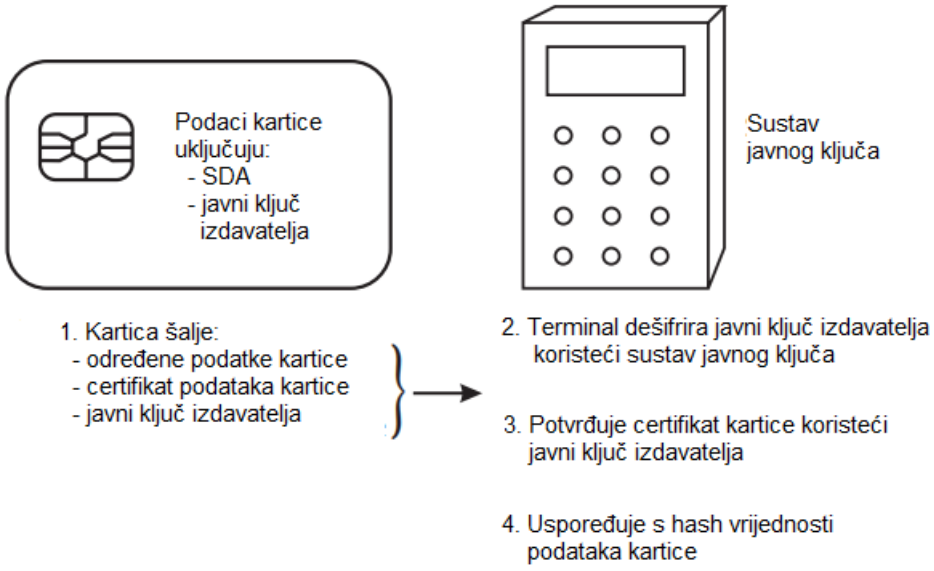
U ovom dijelu ćemo malo pobliže objasniti kreditne i debitne kartice te njihove funkcije.

Debitne kartice su sada najčešće korištene kartice u Europi. One se mogu koristiti u bankomatima i u trgovinama, a uvijek su povezane s tekućim računom. Transakcije se knjiže na račun čim su primljene; u slučaju trgovine na malo, to može biti odmah, sljedeći dan ili nakon obrade papirnatog bona. Većina debitnih kartičnih sustava inzistira na elektroničkoj obradi, ali transakciju često autorizira druga banka ili kartični sustav (Visa, Europay ili MasterCard), a ne izdavatelj kartice; to se naziva *stand-in* obrada.

Kreditne kartice su najstariji oblik bankovnih platnih kartica. One omogućuju kupcu plaćanje putem kredita do unaprijed postavljenog limita. Princip korištenja je jednostavan: plaćate karticom, a odgovarajući iznos kasnije se tereti s vašeg računa. Troškove ovog postupka snosi trgovac, koji obično plaća naknadu koja ovisi o iznosu transakcije. Ta naknada obično iznosi oko 2 do 5 % kupoprodajne cijene. Kreditne kartice se mogu koristiti online ili offline, a razina autentifikacije izdavatelja je niža nego za debitne kartice. Stoga je rizik kreditnih gubitaka i prijevara veći pa su i troškovi trgovaca i vlasnika kartica viši nego kod debitnih kartica.

Zahtjevi kreditnih/debitnih kartica

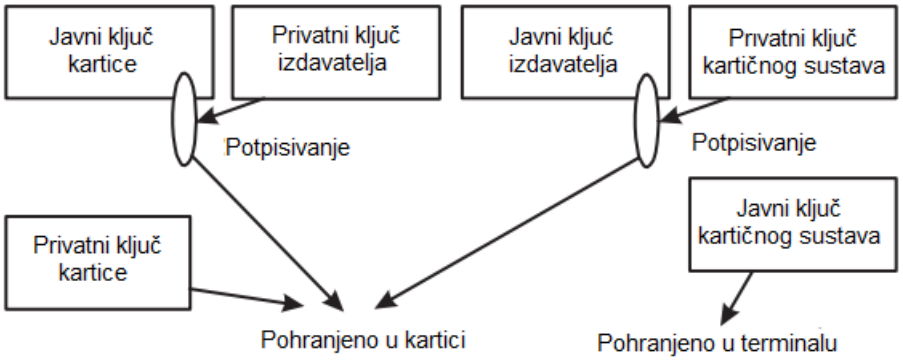
Glavni uvjet za pametnu kreditnu ili debitnu karticu je smanjenje mogućnosti za prijevaru bez povećavanja potrebe za online autentifikacijom. Zbog toga je prvi zahtjev mogućnost provjere autentičnosti kartice izvan mreže (offline). Prisjetimo se stoga statičke i dinamičke autentifikacije. Sve kartice izdaje izdavatelj koji pripada sustavu. Svi terminali povezani s tim sustavom moraju učitati javni ključ sustava. To omogućava provjeru autentičnosti kartice pomoću hijerarhije: sama kartica sadrži certifikat izdan pod tajnim ključem izdavatelja kartice čiji se javni ključ (potpisan u sustavu) također nalazi na kartici. Ovaj oblik statičke autentifikacije može provesti bilo koji terminal s javnim ključem sustava.



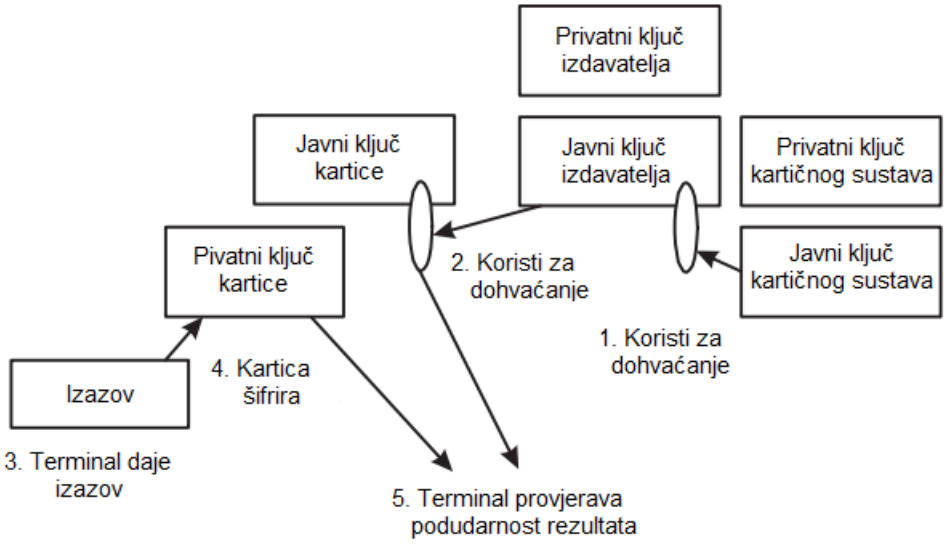
Slika 3.1 Statička autentifikacija

Međutim, neće otkriti karticu koja je napravljena koristeći točnu kopiju izvornih podataka kartice (uključujući certifikat). Ovo zahtijeva dinamičku autentifikaciju. Kad se koristi dinamička autentifikacija, sama kartica ima jedinstvene javne i privatne ključeve. Javni ključ kartice potpisuje izdavač kartice i taj se certifikat pohranjuje na karticu. Terminal započinje proces na isti način preuzimanjem javnog ključa izdavatelja, ali sad koristi javni ključ izdavatelja za preuzimanje javnog ključa kartice. Zatim kartici šalje poruku, koja se zove „izazov". Ta poruka ne sadrži samo neke podatke s kartice, već i slučajni broj. Kartica koristi svoj privatni ključ za šifriranje izazova, a terminal ga dešifrira pomoću javnog ključa kartice. Ako se taj rezultat i dani izazov podudaraju, onda možemo biti sigurni da je kartica zaista originalna kartica (budući da nikoje dvije kartice nemaju iste ključeve).

Daljnja provjera autentičnosti kartice može se provesti ako se transakcija događa online. U ovom slučaju se može koristiti simetrični ključ jer se ključevi ne moraju distribuirati



Slika 3.2 Dinamička autentifikacija - izdavanje



Slika 3.3 Dinamička autentifikacija - upotreba

preko terminalne mreže. Većina bankovnih kartica može izvršiti DES šifriranje unutar kartica.

Sljedeći zahtjev kartice je provjera identiteta osobe koja predočava karticu. Unatoč biometrijskim provjerama dostupnim na pametnim karticama, PIN je i dalje najčešći mehanizam za provjeru u bankarskim aplikacijama. Tamo gdje se PIN koristi za provjeru kartice, on se može provjeriti i offline. I na kraju, izdavatelj kartice nastoji osigurati da kupci ne prelaze kreditne limite.

Kartica također nudi mogućnost kontrole vlasniku kartice. Neki kupci žele biti u mo-

gućnosti kontrolirati potrošnju vrlo strogo i spremni su se suočiti s određenim neprilikama ili kašnjenjem kako bi to postigli. Za ostale su praktičnost i brzina važniji od detaljnih zapisa svih transakcija. .

EMV

Banke širom svijeta prihatile su ranije spomenute Europay–MasterCard–Visa (EMV) standarde kao osnovu za pametne kreditne i debitne kartice. Sami EMV standardi odnose se na interoperabilnost: nadovezuju se na ISO 7816 standarde za električna sučelja i protokole, ali dodavaju neke detalje u područjima koja su otvorena unutar ISO 7816. EMV također dodaje posebne funkcije potrebne za bankovne kartice (statička provjera podataka, PIN provjera) i definira elemente podataka (broj računa, brojač transakcija itd.) koje aplikacije mogu koristiti.

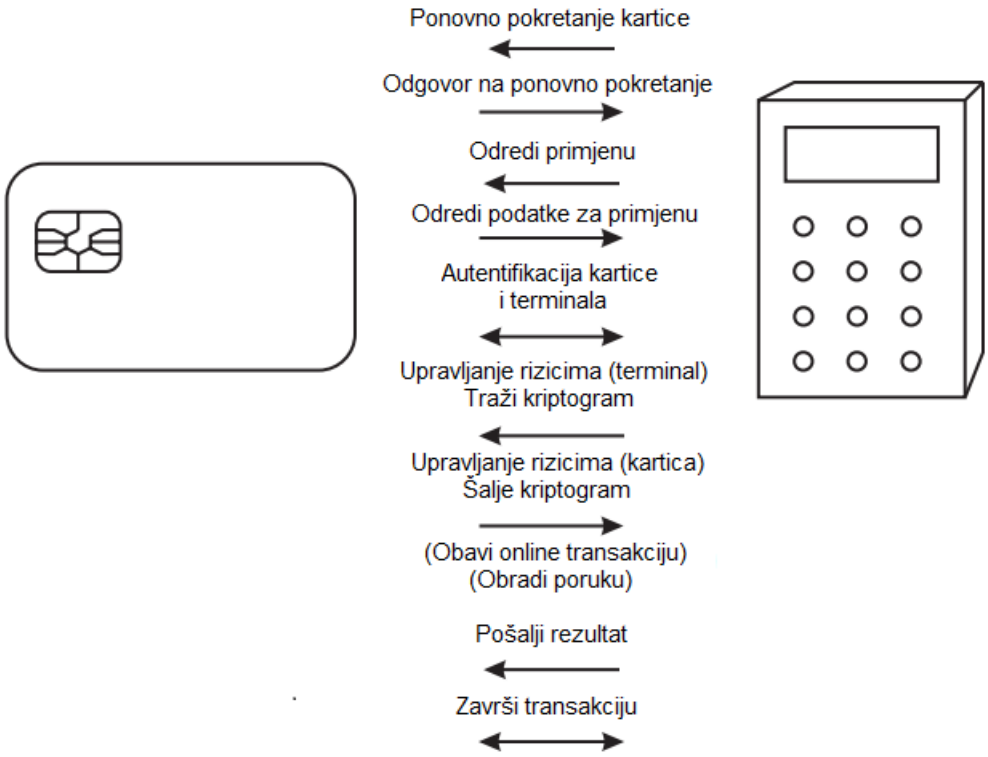
EMV je definiran s kreditnim i debitnim karticama na umu, ali EMV kompatibilna kartica može sadržavati i druge primjene, poput elektroničkog novčanika. EMV mehanizmi za provjeru autentičnosti kartice uključuju statičku provjeru autentičnosti podataka, dinamičku provjeru autentičnosti podataka (oba koriste šifiranje javnim ključem kao što je ranije opisano) i online autentifikaciju (pomoću Trostrukog DES-a s izvedenim ključem). Autentifikacija poruke se također bazira na DES-u te postoji odredba za PIN-ove, biometriju na kartici ili provjeru potpisa.

3.2 Elektronički novčanici

Pojava pametnih kartica omogućila je bankama da uvedu jednu novinu pod nazivom elektronički novčanici. Dok kreditne kartice nude „platite kasnije", a debitne kartice „platite sada", elektronički novčanici omogućuju plaćanje unaprijed. Elektroničke novčanike obično vežemo uz manje kupnje gdje nema velike potrebe za pojedinačnim kontrolama svake transakcije. Proces korištenja elektroničkog novčanika je obično vrlo jednostavnan, ali se upravo zbog te jednostavnosti javlja problem sigurnosti.

Većina elektroničkih novčanika prve generacije koristila je simetrično šifriranje za autentifikaciju. To je nudilo relativno jednostavnu i jeftinu implementaciju sa standardnim 8-bitnim karticama. Koristeći tu tehnologiju, vrijeme transakcija iznosilo je oko 2 sekunde. S druge strane, sigurnosni moduli SAM (*eng. Secure Access Modules*) bili su potrebni na svakom terminalu za pohranjivanje tajnih ključeva sustava. Time je SAM potencijalno bio slaba točka sustava i njegova sigurnost je bila ključna. Velikim sustavima, koji se baziraju na tajnim ključevima, teško je upravljati pa zbog toga moderniji sustavi koriste autentifikaciju s javnim ključem. Ovdje se u terminal treba pohraniti samo javni ključ sustava što je mnogo sigurnije i omogućuje lakše upravljanje. Nedostatak je to što kartice moraju imati više memorije, a i transakcije su sporije.

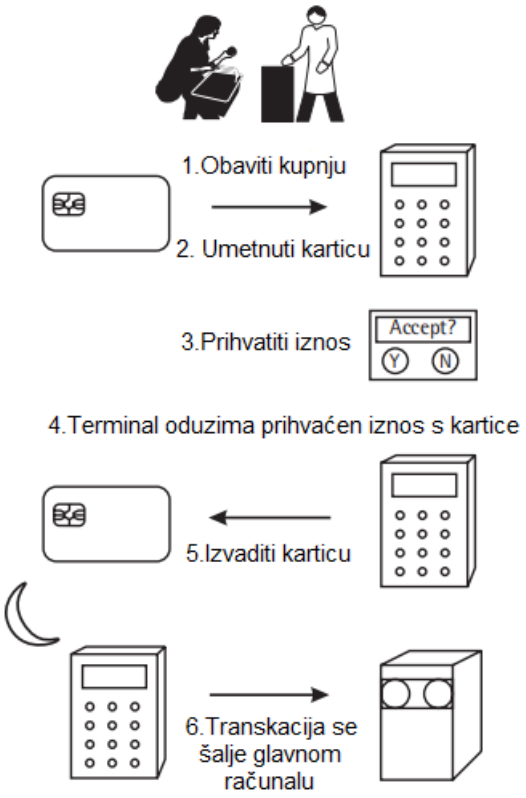
Čitava sigurnost sustava temelji se na kriptografskom algoritmu. Sve poruke koje se razmjenjuju sadrže priložene digitalne potpise kako bi se omogućilo otkrivanje manipulacija. Ovo je jedina zaštita za poruke koje se uvijek razmjenjuju u otvorenom tekstu. Razmjena poruka strukturirana je tako da se za generiranje potpisa može koristiti bilo koja



Slika 3.4 Model EMV transkacije

vrsta kriptografskog algoritma. Najčešće se koristi simetrični algoritam DES, ali standard omogućuje i asimetrične algoritme poput RSA ili DSA. Ova neovisnost algoritma je velika prednost jer znatno produžava vijek trajanja i fleksibilnost standarda. Da bi se elektronički novčanici smatrali sigurnima, moraju koristiti vrlo dugačke ključeve (128-bitne za Trostruki DES i 2048-bitne za RSA).

Procesi elektroničkih novčanika su: učitavanje, plaćanje, otkazivanje plaćanja, ispravljanje pogreške, pretvaranje valuta, promjena parametara novčanika. Svaki je postupak u osnovi podijeljen u tri faze. Potpuna inicijalizacija komponenti koje sudjeluju događa se u prvoj fazi. Stvarno izvršenje transakcije odvija se u drugoj fazi. Treća faza, koja nije obavezna, koristi se za potvrdu prethodnih radnji. Uspješno okončanje prve dvije faze predstavlja jednostranu (ili po želji uzajamnu) autentifikaciju dviju komponenti. U svim je postupcima jednostrana ili uzajamna autentifikacija prepletena sa stvarnim funkcijama novčanika (plaćanje, učitavanje itd.). Na ovaj način se smanjuje vrijeme potrebno za transakcije i povećava sigurnost (jer se značajno smanjuje broj naredbi potrebnih za obavljanje funkcija).



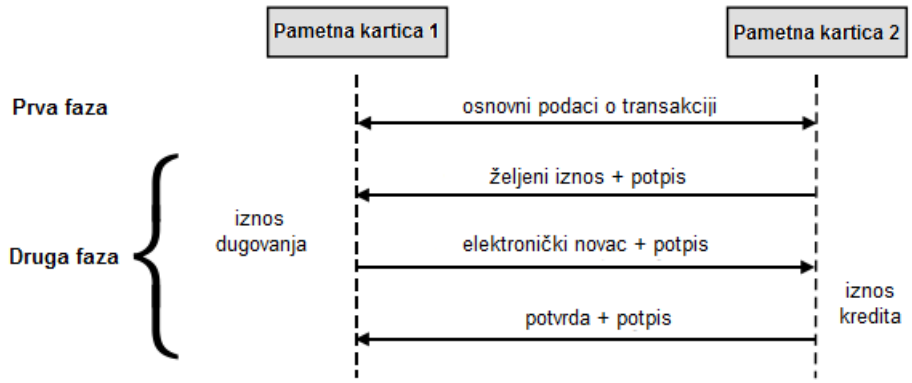
Slika 3.5 Korištenje elektroničkog novčanika

MONDEX

Jedan od prvih sustava za elektorničke novčanike baziran na javnom ključu bio je Mondex. Mondex sustav je jedinstven po tome što omogućuje transakcije među karticama koje ne prolaze kroz terminal trgovca. To omogućuje vlasnicima kartica da vrše isplate članovima vlastite obitelji, prijateljima, ili povremenim davateljima usluga koji nisu dio kartičnog platnog sustava. MONDEX podržava sve vrste transakcija koje su moguće s normalnim novcem. Pored toga, omogućuje plaćanje putem različitih telekomunikacijskih medija, poput telefonskog sustava. Ako se kartica koja sadržava novčanik izgubi, novac koji je u njemu također se prirodno gubi, jednako kao i kod pravog novčanika koji sadrži novac. Do određene mjere, Mondex sustav je simulacija stvarnog novčanog kruga. Budući da su mnoge središnje banke i državna tijela dosta rezervirana u vezi izravnih novčanih transkacija s kartice na karticu, razvijena je i inačica Mondexa u kojoj su transakcije među elektroničkim novčanicima blokirane. Tako se postiže novčani krug za sustav elektroničkih novčanika koji je sličan standardu EN 1546 (europski standard koji sadrži potpuni opis sustava elektroničkih novčanika, uključujući pametne kartice, terminale s njihovim sigurnosnim modulima te pozadinske i klirinške sustave).

Tipična transakcija plaćanja između dvije pametne kartice u Mondex sustavu podije-

ljena je u dvije faze. U prvoj fazi registrira se trenutna transakcija koja uključuje razmjenu svih podataka potrebnih za naknadni prijenos novca. Nakon toga slijedi druga faza u kojoj druga pametna kartica šalje željeni iznos prvoj pametnoj kartici. Kompletan skup podataka digitalno je potpisan tako da se njime ne može manipulirati tijekom prijenosa. Nakon primitka podataka, pametna kartica 1 provjerava potpis kako bi se potvrdila autentičnost pametne kartice 2 i autentičnost prenesenih podataka. Ako su sve ove provjere uspješne, željeni iznos se oduzima s pametne kartice 1 i šalje na pametnu karticu 2, zajedno s digitalnim potpisom. Pametna kartica 2 provjerava ovaj potpis kako bi se uklonila mogućnost da su podaci manipulirani, što omogućuje i autentifikaciju pametne kartice 1. Ako su i sve ove provjere uspješne, iznos se prenosi u novčanik. Nakon toga, pametna kartica 2 generira potvrdu da je iznos pravilno pripisan, dodaje digitalni potpis i te podatke šalje pametnoj kartici 1. Transakcija je zaključena kad je potvrda o plaćanju primljena i uspješno provjerena.



Slika 3.6 Transkacija između dvije pametne kartice u Mondex sustavu

CEPS

Temeljni preduvjet za postizanje međusobne kompatibilnosti između nekoliko sustava elektroničkih novčanika je dokument koji određuje značajke koje sustavi moraju imati za kompatibilnost. Ovaj dokument nosi naziv CEPS (*eng. Common Electronic Purse Specifications*), a prvu verziju je 1999. godine objavio CEPSCO. U ranijoj fazi specifikacije CEPS je bio fokusiran na međunarodno interoperabilni sustav elektroničkih novčanika, a ne samo na jedan sustav ograničen na europske interese. CEPS uključuje standardne funkcije za moderne sustave elektroničkih novčanika, kao što su offline plaćanje, online učitavanje i online konverzija valuta. Temelji se na europskom standardu za elektroničke novčanike (EN 1546), ali sadrži određena proširenja i modifikacije u odnosu na ovaj standard. Na primjer, za razliku od EN 1546, za provjeru autentičnosti terminala i pametnih kartica koriste se RSQ-certifikati. Kao kriptografski algoritam preporučuje se Trostruki DES. CEPS je, kao i mnogi sustavi elektroničkih novčanika, optimiziran za jednostavne mikrokontrolere

pametnih kartica. Tipična implementacija CEPS-a zahtijeva 8 kB ROM-a, 4 kB EEPROM-a, 1 kB RAM-a i numerički koprocesor za asimetrični kriptografski algoritam.

Proces plaćanja s elektroničkim novčanikom

Sljedeći primjer prikazuje postupak plaćanja koristeći komponente potrebne za ovu funkciju: elektronički novčanik na pametnoj kartici (IEP, *eng. Inter-sector Electronic Purse*), terminal (PDA, *eng. Personal Digital Assistant*) i sigurnosni modul na terminalu (PSAM, *eng. Purchase Secure Application Module*).

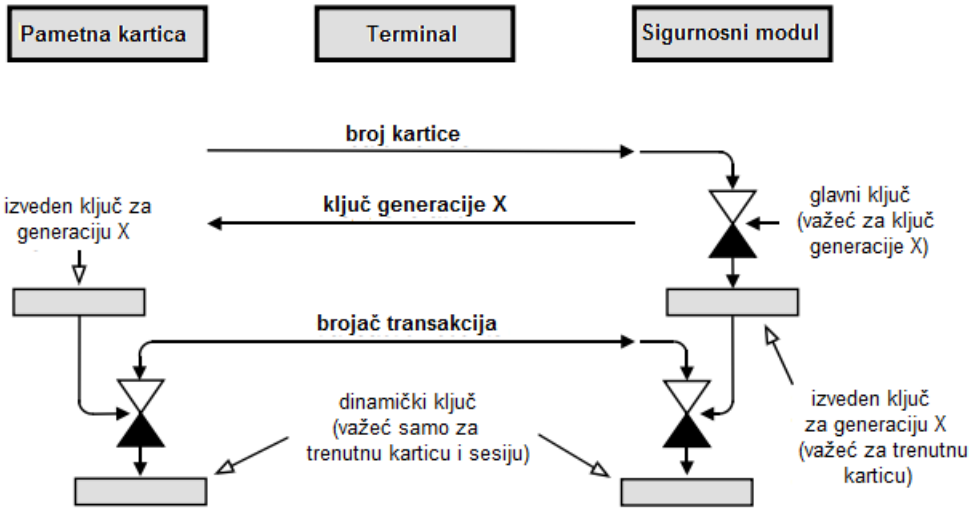
Nakon što je kartica novčanika umetnuta u terminal, terminal izvršava ponovno pokretanje kako bi zatražio odgovore od PSAM-a i IEP-a. Ako se bilo koji od ovih odgovora ne podudara s očekivanom vrijednosti, terminal prekida postupak plaćanja. Ako se odgovori podudaraju s njihovim očekivanim vrijednostima, terminal odabire DF (*eng. Dedicated File*, datoteka koja sadrži kompletnu prijavu za elektronički novčanik, a nalazi se u pametnoj kartici) u IEP-u. Ako se ova datoteka ne može odabrati, postupak se također prekida.

Nakon odabira DF-a, terminal šalje naredbu za inicijalizaciju. IEP prima ovu naredbu, povećava brojač transakcija, izračunava ključ i generira potpis (S1) za različite elemente podataka. Tada te elemente podataka i potpis šalje terminalu. Potom terminal šalje naredbu za inicijalizaciju na PSAM. Ova naredba jednostavno prebacuje podatke koji su primljeni s kartice na PSAM. PSAM potvrđuje ove elemente podataka, što znači da se datum isteka, valuta, korišteni kriptografski algoritam i ostali primljeni podaci uspoređuju s vrijednostima pohranjenim u PSAM-u. Ako su sve usporedbe uspješne, brojač transakcija se povećava. Ako bilo koja usporedba ne uspije (npr. ako je prošao datum isteka IEP-a), obrada naredbi odmah se prekida i na terminal se šalje odgovarajući povratni kod. Nakon toga PSAM generira izvedeni ključ koristeći podatke koje pruža IEP i generira ključ sesije, a zatim provjerava potpis S1. Ako je potpis točan, slijedi da su svi preneseni podaci autentični i time PSAM potvrđuje autentičnost IEP-a. Drugim riječima, PSAM zna da je kartica koja sadrži elektronički novčanik originalna. Zatim PSAM generira potpis (S2) koji se šalje terminalu zajedno s drugim elementima podataka.

Korisnik sada na terminal unosi iznos koji treba platiti i pridruženu valutu. Terminal zatim šalje uneseni iznos i elemente podataka prethodno primljene od PSAM-a na karticu. IEP sada provjerava ima li novca u novčaniku za plaćanje. Ako ima, provjerava potpis S2. Ako je potpis točan, podaci nisu promijenjeni za vrijeme prijenosa te je potvrđena autentičnost PSAM-a (jer samo pravi PSAM može posjedovati tajni ključ potreban za generiranje potpisa S2). Od salda novčanika oduzima se odgovarajući iznos, generira se treći potpis (S3) kako bi se potvrdila upravo izvršena debitna transakcija i ažurira se datoteka dnevnika. Potpis S3 i iznos dugovanja šalju se preko terminala PSAM-u koji provjerava S3. Ako je ovaj potpis točan, iznos dugovanja u IEP-u dodaje se internom elementu podataka. Potom se ažurira stanje PSAM-a dodavanjem tog internog elementa podataka u saldo novčanika. Napokon, PSAM dobiva potpis (S4) kojim se potvrđuje da je transakcija uspješno završena.

Na sljedećoj slici prikazan je postupak izvoda ključa za sustav elektroničkog novčanika EN 1546. Ključ ovisi o kombinaciji ključa specifičnog za karticu određene generacije i o brojaču transakcija specifičnog za određenu sesiju. Tako stvoren ključ može se koristiti za

plaćanje ili zaduživanje.



Slika 3.7 Proces izvoda ključa za sustav elektroničkog novčanika EN 1546

3.3 Online transakcije

Jedna od vrsta online financijskih transakcija je online bankarstvo. Funkcije koje se često koriste u online bankarstvu su pregledavanje računa, postavljanje ili pokretanje prijenosa između računa te u nekim slučajevima slanje uputa za plaćanje. Osim toga postoji druga vrsta online transkacija koja obuhvaća e-trgovinu. Jasno je da je ovdje autentifikacija kupca presudna.

Autentifikacija transkcija

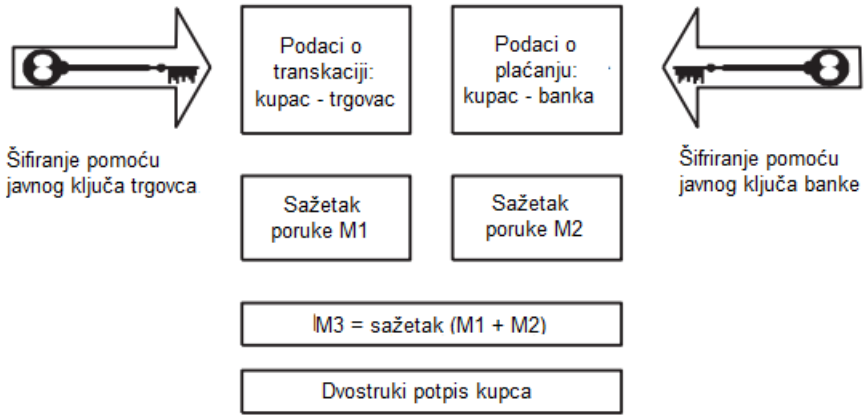
Bilo koja vrsta transakcije, poput slanja narudžbe, stavljanja oglasa ili plaćanja, može se autentificirati pomoću pametne kartice. Koristi se sustav javnih ključeva, s tajnim ključem koji se nalazi na kartici, a javni ključ je dostupan drugoj strani, tj. primatelju. Sustav stvara sažetak poruke, to potpisuje kartica tajnim ključem, a primatelj provjerava javnim ključem. Ako se postupak ponovi u obrnutom smjeru (kad se transakcija potvrdi), pošiljatelj će moći autentificirati primatelja. Ovaj mehanizam omogućuje potpunu provjeru autentičnosti poruke i niti jedna strana ne može nakon toga odbiti transakciju.

U SAD-u *American Express* je prednjačio u ponudi kartica s ovim oblikom autentifikacije za upotrebu u online transakcijama. Takozvana „Plava kartica" bila je prva financijska pametna kartica za masovno tržište u Sjedinjenim Državama koja je kombinirala funkcije kreditne kartice, digitalnog potpisa X.509 i elektroničkog novčanika za online plaćanja.

Sigurnost transakcija

Transakcije e-trgovine predstavljaju problem bankama zbog sigurnosti, tj. nesigurnosti. Stopa spornih transakcija najmanje je 10 puta veća nego za transakcije licem u lice. Osim provjere autentičnosti kupca (osigurati da ne koristi tuđu karticu), potrebno je i pohranjivanje cjelovitog zapisa transakcije (koja je roba naručena, trošak isporuke, kako je kupac izrazio svoj pristanak), kojem se može pristupiti u slučaju spora.

Da bi se riješio taj problem, glavni kartični sustavi razvili su protokol za sigurne elektroničke transakcije (SET). Vlasnici kartica moraju na svoje računalo učitati SET novčanik i certifikat koji su dobili od izdavača kartice. Trgovci imaju poseban SET sustav i certifikat koji je izdala banka prihvatiteljica. Pojedinosti transakcije se šifriraju i potpisuju ih obje strane - trgovac nikad ne vidi broj kartice, a banka nikada ne vidi podatke o transakciji. No cjelokupni potpis na poruci omogućuje svim stranama da provjere da li se transakcija dogodila, da li ju je odobrio vlasnik kartice i da li su pojedinosti bile kako tvrdi trgovac.



Slika 3.8 SET poruka

SET je vrlo sigurno rješenje, ali je naišao je na niz problema:

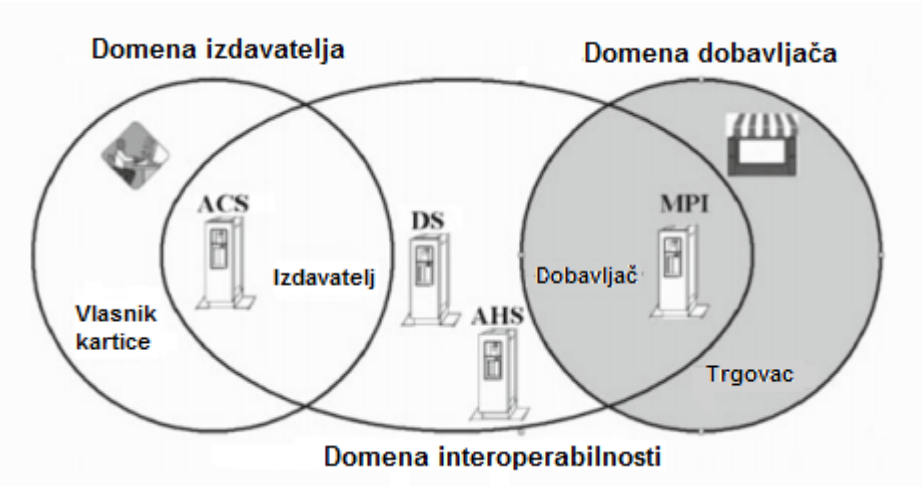
- novčanik vlasnika kartice bio je velik i spor za preuzimanje
- kupci su imali malo poticaja za upotrebu SET-a jer ga je malo trgovačkih stranica prihvatilo, a trgovci nisu stjecali dobitak od prodaje jer je malo vlasnika kartica imalo certifikate
- transakcije su bile vrlo spore (norma je bila 40–45 sekundi)

SET se poboljšava upotrebom elektroničkih novčanika i certifikata, kao što je prikazano na slici 3.8. Time se prevladavaju problemi distribucije i brzine.

Osim SET-a, postoji još jedan protokol pod nazivom SSL (*eng. Secure Sockets Layer*). SSL je standardna sigurnosna tehnologija za uspostavljanje šifrirane veze između poslužitelja i klijenta. SSL je obično web poslužitelj koji pruža razinu provjere autentičnosti: klijent i poslužitelj se potvrđuju međusobno, s time da klijent radi samo s certifikatom poslužitelja koji je izdan od strane certifikacijskog tijela koje prepoznaje.

3D - *Secure*

3D - *Secure* je globalni standard za provjeru autentičnosti kupaca u sustavu sigurnih internetskih transakcija. Globalni platni sustavi Mastercard i Visa razvili su vlastite programe sigurnosne kupovine na internetu bazirane na 3D-*Secure* modelu, *„Mastercard Secure Code"* i *„Verified by Visa"*. Ovaj standard bazira se na tri domene : domena izdavatelja, domena interoperabilnosti i domena dobavljača.



Slika 3.9 Model tri domena

Unutar domene izdavatelja vidljivo je da izdavatelj ima odnos s vlasnikom kartice. To znači da je postupak registracije vlasnika kartice dovršen pa izdavatelj može autentificirati vlasnika kartice pomoću poslužitelja poznatog kao ACS (*eng. Access Control Server*). Domena dobavljača sadrži trgovca, banku prihvatiteljicu i pridružene mrežne sustave. Budući da većina sustava poslužitelja za e-trgovinu nema ugrađen 3D-S protokol, potreban je dodatak MPI (*eng. Merchant Server Plug-in*) koji služi za pokretanje provjere autentičnosti u domeni izdavatelja. Još preostaje domena interoperabilnosti koja je u osnovi *vodovod* koji se koristi za povezivanje domene izdavatelja i domene dobavljača u svrhu osiguranja transakcija e-trgovine. Glavna komponenta unutar domene interoperabilnosti je poslužitelj zvan DS (*Directory Server*) koji se nalazi u platnom sustavu. Njime se utvrđuje koji ACS je potreban da bi se određena kartica uključila u transakciju. Drugim riječima, DS je učinkovito pridruživanje sudjelujućih/registriranih kartica odgovarajućoj adresi poslužitelja ACS. Također unutar ove domene postoji poslužitelj AHS (*eng. Authentication History Server*)

koji služi kao dnevnik za bilježenje svih provjera autentičnosti za potrebe rješavanja sporova. Nakon što se provjeri autentičnost, transakcija će biti autorizirana i obračunata putem uobičajenih mreža platnih sustava koji povezuju dobavljače s izdavateljima.

Bibliografija

[1] A. Dujella, *Uvod u teoriju brojeva*, PMF - Matematički odjel, Sveučilište u Zagrebu (skripta), 2007.

[2] A. Dujella, M. Maretić, *Kriptografija*, Element, 2007.

[3] M. Hendry. *Smart Card Security and Applications*, Artech House, 2001.

[4] K. E. Mayes, K. Markantonakis (Eds.), *Smart Cards, Tokens, Security and Applications*, Springer, 2017.

[5] W. Rankl, W. Effing, *Smart Card Handbook*, Wiley, 2003.

Sažetak

Ovaj rad proučava primjene pametnih kartica u bankarstvu i financijama te kriptografske algoritme koji se koriste u tim primjenama. Rad se može promatrati kroz tri osnovne cjeline. U prvom dijelu rada objašnjava se kako je tekao razvoj pametnih kartica, koje su njihove karakteristike te koje su razlike među određenim vrstama pametnih kartica.

Drugi dio rada opisuje sve informatičke elemente potrebne za funkcioniranje pametnih kartica. Vrlo važan dio čini kriptografija. Za početak se prolazi matematika koja je u pozadini kriptografskih algoritama. Od simetričnih kriptografskih algoritama obrađeni su DES i Trostruki DES, a od asimetričnih RSA i DSA. Potom su definirane hash funkcije i način na koji su one ukomponirane u rad pametnih kartica. Sljedeće što se obrađuje je proces autentifikacije. Promatraju se razlike između statičke i dinamičke, jednostrane i uzajamne autentifikacije kao i razlike među onima koji se temelje na simetričnim i onima koji se temelje na asimteričnim kriptografskim algoritmima. Za kraj se opisuju digitalni potpisi i certifikati koji pružaju dodatnu sigurnost u transakcijama koje se obavljaju preko pametnih kartica.

Treći dio rada odnosi se na konkretne primjene pametnih kartica u financijama i bankarstvu. U tom dijelu opisuju se debitne i kreditne kartice, elektronički novčanici te online transakcije i objašnjava njihov način funkcioniranja.

Summary

This thesis studies the applications of smart cards in banking and finance and the cryptographic algorithms used in those applications. It may be divided into three fundamental parts. The first part explains how smart cards are developed, what are their characteristics and what are the differences between different types of smart cards.

The second part describes all the IT elements needed to operate smart cards. A very important part is cryptography. For starters, the math underlying cryptographic algorithms is brought to light. DES and Triple DES are symmetric cryptographic algorithms which are explained also with asymmetric RSA and DSA. After that hash functions are defined and the way how they are integrated into the operation of smart cards. The next thing that is explained is the authentication process. There are differences between static and dynamic authentication, unilateral and mutual authentication, as well as differences between authentication based on symmetric and asymmetric cryptographic algorithms. Finally, digital signatures and certificates, which provide added security in smart card transactions, are described.

The third part of the paper deals with the specific applications of smart cards in finance and banking. This chapter describes debit and credit cards, electronic wallets, and online transactions and explains their way of functioning.

Životopis

Rođena sam 6. kolovoza 1995. u Zagrebu. Cijeli život provela sam u Zagrebu gdje sam završila Osnovnu školu Augusta Šenoe na Trešnjevci te opću II. gimnaziju. Oduvijek sam težila dvjema prirodnim znanostima, biologiji i matematici, te sam u konačnici odlučila nastaviti obrazovanje u smjeru matematike. Godine 2014. upisujem preddiplomski studij Matematike na Matematičkom odsjeku PMF-a. Titulu sveučilišne prvostupnice stekla sam 2017. kada sam i upisala diplomski studij Financijske i poslovne matematike na istom fakultetu.