

Sigurnost e-poslovanja

Digitalni potpis

Potpisi

- ✓ Pisani potpis
 - ✓ Ne može se krivotvoriti?!?
 - ✓ Autentičan je (karakterizira jednu osobu)?!?
 - ✓ Nije ga moguće ponovno koristiti?!?
 - ✓ Potpisani dokument je nepromjenjiv?!?
 - ✓ Ne može se poricati?!?

Potpisi

- ✓ Digitalni potpis
 - ✓ Ne može se krivotvoriti (samo pošiljatelj zna svoj privatni ključ)
 - ✓ Autentičan je (privatni ključ je svojstven jednoj osobi)
 - ✓ Nije ga moguće ponovno koristiti (potpis s jednog dokumenta vrijedi samo za taj dok.)
 - ✓ Potpisani dokument je nepromjenjiv
 - ✓ Ne može se poricati

Definicija

- ✓ Digitalni potpis - digitalni kod koji služi za zaštitu poruka koje se elektronički prenose putem javne mreže.
- ✓ Svrha digitalnog potpisa:
 - ✓ omogućiti identifikaciju pošiljaoca
 - ✓ osigurati autentičnost sadržaja poruke.

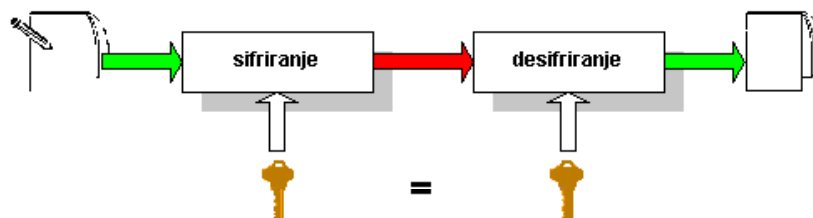
Načini realizacije digitalnog potpisa

- ✓ kriptografski sustav s tajnim ključem
- ✓ kriptografski sustav s javnim ključem
 - ✓ korištenje RSA algoritma
- ✓ algoritam digitalnog potpisa
(*Digital Signature Algorithm, DSA*)

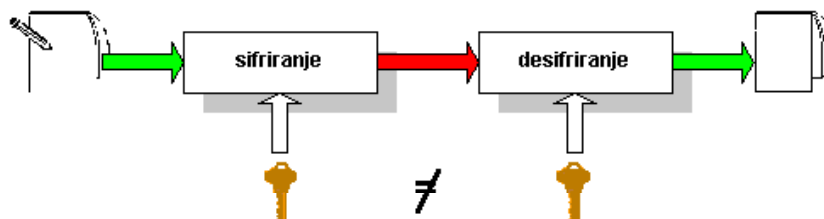
Pojmovi

- ✓ Ključ je niz alfanumeričkih znakova koji koristi kriptografski algoritam, a služi za određivanje izlaza iz funkcija kriptiranja i dekriptiranja.
- ✓ Služi za:
 - ✓ Kriptiranje (šifriranje) i dekriptiranje (dešifriranje)
 - ✓ Detekciju neovlaštenog pristupa
 - ✓ Provjeru vjerodostojnosti.
- ✓ Šifriranje je proces transformacije podataka u oblik nerazumljiv svima osim osobama koje međusobno komuniciraju.
- ✓ Dešifriranje je obratan proces, proces transformacije šifriranog teksta u korisniku prepoznatljiv oblik.

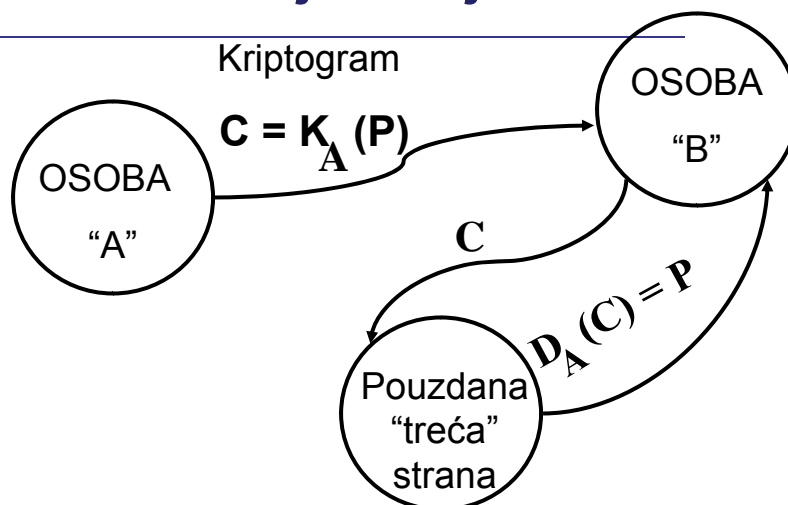
Sustav s javnim ključem - simetrični sustav -



Sustav s javnim ključem - asimetrični sustav -

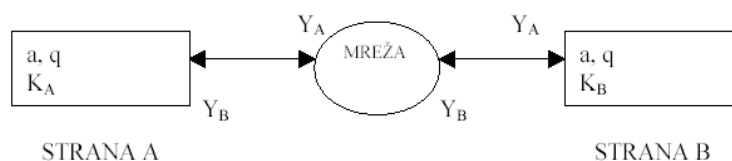


Sustav s tajnim ključem



*posjeduje transformacije šifriranja i dešifriranja osobe "A" i osobe "B"

Sustav s javnim ključem



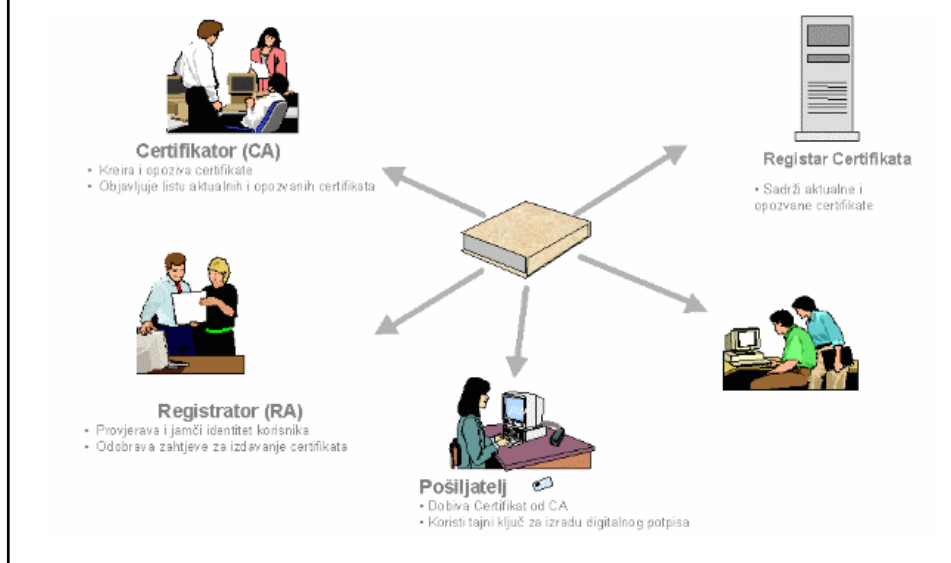
Sustav s javnim ključem - certifikati -

- ✓ Problem distribucije javnog ključa – kako povezati javni ključ sa vlasnikom?
- ✓ Rješenje - izdavanje certifikata.
- ✓ Certifikat - elektronički dokument koji identificira pojedinca, računalo ili neki drugi entitet koji posjeduje javni ključ.
- ✓ Treba povezati par ključeva s njihovim vlasnikom
- ✓ Izadavač certifikata ili Certificate Authority (CA)
- ✓ Publiciraju se u repozitoriju – bazi podataka u kojoj se nalaze certifikati i pripadajuće informacije

Sustav s javnim ključem - certifikati -

- ✓ Elementi certifikata:
 - ✓ Verzija
 - ✓ Serijski broj
 - ✓ Identifikacijska oznaka algoritma digitalnog potpisa
 - ✓ Ime ovlaštenog certifikatora (CA)
 - ✓ Vrijeme trajanja certifikata
 - ✓ Vlasnik javnog ključa

Sustav s javnim ključem - certifikati -



Sustav s javnim ključem – RSA algoritam (Rivest, Shamir, Adleman)

- ✓ RSA sustav se temelji na teoriji brojeva, a princip njegova djelovanja je slijedeći:
 - ✓ odabiru se dva velika prosta (prim) broja P i Q ($P > 10100$ i $Q > 10100$);
 - ✓ odredi se umnožak $N = P * Q$, te vrijednost $(P-1) * (Q-1)$ koju označimo sa $L(N)$;
 - ✓ odabire se broj d , tako da bude $\max(P, Q) < d < L(N)$;
 - ✓ izračunava se broj e tako da bude $0 < e < L(N)$ i da je $(e * d) \bmod L(N) = 1$, što je isto kao da odredimo najmanji k za koji vrijedi $e * d = k * L(N) + 1$;
 - ✓ par (e, N) se proglašava javnim ključem.
 - ✓ Iz teorije brojeva je poznato da uz $0 \leq M < N$, za tako izračunate e i d vrijedni da je:
($M \bmod N$) $d \bmod N = M \bmod N = M k L(N) + 1 \bmod N = M$

Sustav s javnim ključem – RSA algoritam

- ✓ Kriptosustav s javnim ključem djeluje na slijedeći način:
 - ✓ kriptiranje se vrši javnim ključem (e, N) tako da se:
 - ✓ razgovjetni tekst kodira u niz cijelih brojeva M_i koji smiju poprimiti vrijednosti iz intervala od 0 do $N-1$,
 - ✓ svaki od tih cijelih brojeva se kriptira u $C_i = (M_i) e \bmod N$.
 - ✓ dekriptiranje se obavlja na slijedeći način:
 - ✓ C_i se dekriptira u razgovjetni oblik $M_i = (C_i) d \bmod N$;
 - ✓ niz brojeva M_i se dekodira u izvorni tekst.

Sustav s javnim ključem – RSA algoritam (Rivest, Shamir, Adleman)

Na strani pošiljatelja:

P - dokument

$$DP = P^d \pmod{n}$$

DP - digitalni potpis

pomoću privatnog ključa (d, n)

Na strani primatelja:

$$P = DP^e \pmod{n}$$

pomoću javnog ključa (e, n)

izračunava P

provjerava $P = P$

Sustav s javnim ključem – RSA algoritam



Sustav s javnim ključem – RSA algoritam

1. Hash funkcijom Bob računa sažetak poruke koju šalje Alici,
2. Bob kriptira svojim tajnim ključem sažetak poruke i na taj način kreira digitalni potpis,
3. Zajedno s originalnim dokumentom, Bob šalje i digitalni potpis,
4. Alice dobiva Bobovu potpisanu poruku, a iz originalne poruke izračuna sažetak,
5. Alice dekriptira digitalni potpis Bobovim javnim ključem, te uspoređuje dekriptirani sažetak s onim koji je sama izračunala. Ako su jednaki, Alice je sigurna da je Bob poslao poruku i da se poruka nije mijenjala tokom slanja (integritet poruke). Bob ne može poreći da je on poslao poruku, jer se digitalni potpis može dekriptirati samo njegovim javim ključem, a kriptirati njegovim tajnim ključem.

DSA algoritam

- ✓ **Digital Signature Algorithm**
- ✓ Definira proces kreiranja (generiranja) i provjere (verifikacije) digitalnog potpisa
- ✓ Razvijen od strane *National Security Agency – NSA*, a *National Institute of Standards and Technology – NIST* ga je standardizirao unutar posebnog standarda za digitalni potpis (*Digital Signature Standard – DSS*)

DSA algoritam

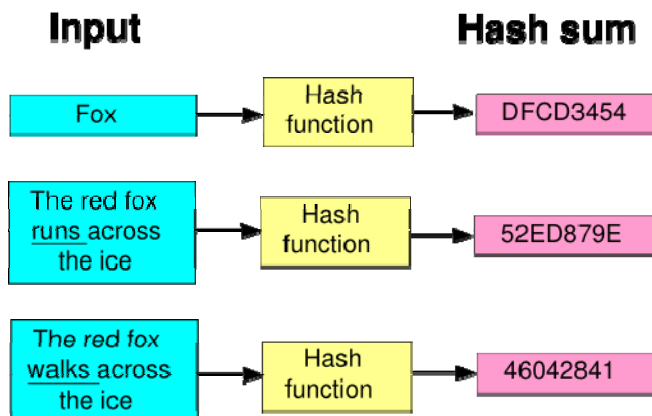
- ✓ Sigurnost DSA temelji se na problemu izračunavanja diskretnog algoritma
- ✓ Koristi se ključ veličine 1024 bita
- ✓ Primjenjuje se na *hash* vrijednost, a ne na cijeli dokument

Hash funkcija

- ✓ Matematičko gledište - funkcija koja transformira proizvoljan broj elemenata ulazne domene u jedan element kodomene
- ✓ ICT gledište - algoritam kojim se varijabilni ulaz proizvoljne duljine transformira u niz znakova fiksno određene duljine
- ✓ Karakteristike:
 - ✓ niz ulaznih podataka je proizvoljne veličine
 - ✓ izlazni podatak je stalne veličine
 - ✓ nemoguće je izvesti inverznu funkciju
 - ✓ ne daje dva ista izlaza za dva različita ulaza

Hash funkcija

- ✓ od ulaza varijabilne veličine vraća znakovni niz fiksne dužine



Hash funkcija

- ✓ Kriptografske hash funkcije su hash funkcije s dodatnim sigurnosnim svojstvima kako bi ih se moglo koristiti za autentifikaciju i očuvanje integriteta podataka
- ✓ Poruka se dijeli na blokove veličine 512 bita
- ✓ MD5 algoritam radi na 128-bitnom izrazu koji se dijeli na 4 32-bitne riječi
- ✓ Zatim se procesiraju redom svi 512-bitni blokovi kojima se mijenja 128-bitni izraz

Hash funkcija

- ✓ Procesiranje poruke sastoji se od 4 slične faze koje se nazivaju "rounds". Svaka faza sastoji se od 16 sličnih operacija baziranih na nelinearnoj funkciji F, modularnom zbrajanju i rotaciji bitova ulijevo.

$$F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$

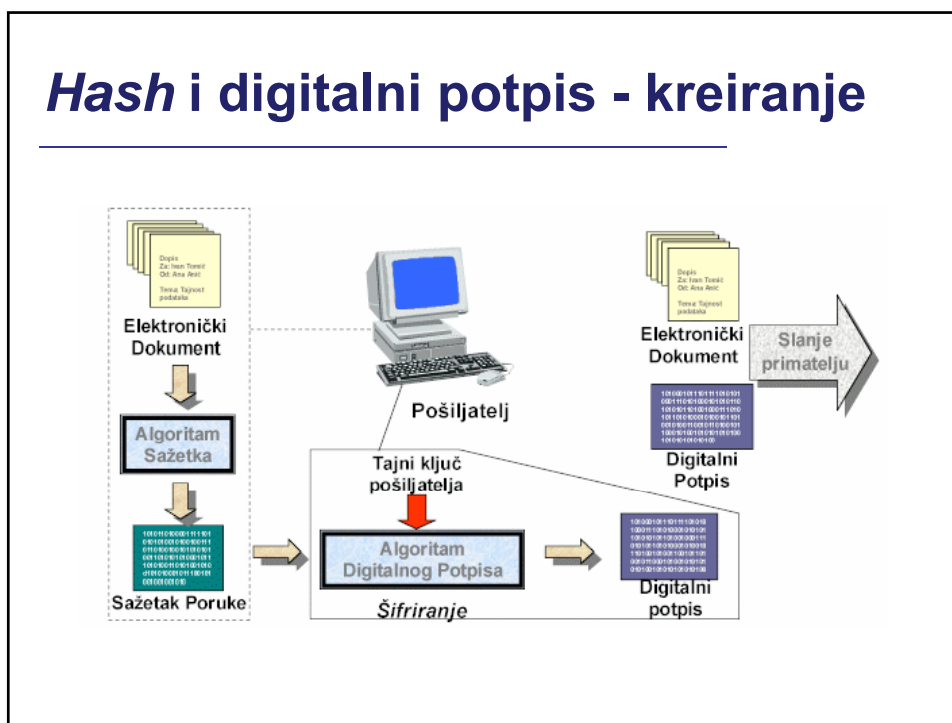
$$H(X, Y, Z) = X \oplus Y \oplus Z$$

$$I(X, Y, Z) = Y \oplus (X \vee \neg Z)$$

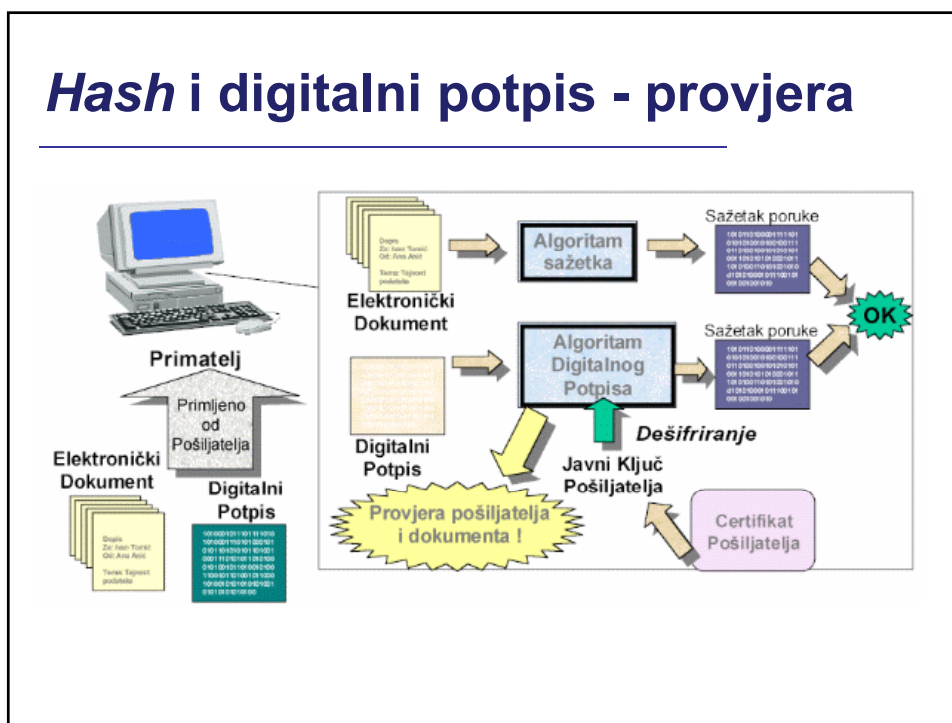
Izvod hash vrijednosti

- ✓ MD2, MD4, MD5 algoritmi
 - ✓ Rivest
 - ✓ 128-bitni izlaz
- ✓ SHA algoritam (*Secure Hash Algorithm*)
 - ✓ Dio SHS (*Secure Hash Standarda*) standarda
 - ✓ Razvile su ga i publicirale NIST i NSA 1994. godine
 - ✓ Algoritam na osnovu maksimalne duljine poruke od 2^{64} bita izračunava broj 164-bitna duljine

Hash i digitalni potpis - kreiranje



Hash i digitalni potpis - provjera



Štićeni prijenos podataka i digitalni potpis

- ✓ S/HTTP
- ✓ S/MIME
- ✓ SSL
- ✓ PGP

Štićeni prijenos podataka - S/HTTP -

- ✓ *Secure Hypertext Transfer Protocol*
- ✓ Nadopuna standardnom *HTTP*-u (*Hypertext Transfer Protokolu*) s ciljem dodavanja sigurnih servisa putem kriptografije.
- ✓ Temelji se na početnom pregovaranju između korisnika i poslužioca oko vrste kriptografije kojom će komunicirati.

Štićeni prijenos podataka - S/HTTP -

- ✓ četiri načina razmjene ključeva:
 - ✓ RSA - razmjenjuju se standardni javni *RSA* ključevi
 - ✓ *in-band* - transport ključeva putem *S/HTTP* zaštićene poruke
 - ✓ *out-band* - vanjski dogovor razmjene ključeva
 - ✓ *Kerberos* - ključ se dobavlja sa *Kerberos* poslužioca

Štićeni prijenos podataka - S/MIME -

- ✓ *Secure Multi-purpose Internet Mail Extensions*
- ✓ Internet standard za kodiranje i potpis poruka za siguran prijenos preko Interneta.
- ✓ S/MIME poruke mogu biti sastavljene od više dijelova s kombinacijama teksta, glasa, i grafike.

Štićeni prijenos podataka

- S/MIME -

- ✓ nadogradnja na postojeći Internet *MIME* protokol s mogućnošću digitalnog potpisivanja i kriptiranja elektronske pošte
- ✓ *S/MIME* standard je prihvatio velik broj proizvođača softvera (*ConnectSoft, Frontier, FTP Software, Microsoft, Lotus, Banyan, NCD, SecureWare, VeriSign, Netscape* i *Novell*)

Štićeni prijenos podataka

- SSL -

- ✓ *Secure Socket Layer*
- ✓ *protokol rukovanja (Handshake Protocol)*
- ✓ podržava identificiranje te osiguravanje autentičnosti i poslužioaca i korisnika
- ✓ transparentan protokol s mogućnost da se drugi protokoli (*HTTP, FTP, Telnet* i *Rsh*) na jednostavan način nadgrade kao komunikacijski sloj
- ✓ Sam protokol rukovanja se sastoji od dvije faze:
 - ✓ identifikacije poslužioaca i
 - ✓ identifikacije korisnika.

Štićeni prijenos podataka - PGP -

- ✓ *Pretty Good Privacy*
- ✓ De facto standard za šifriranje poruka elektronske pošte.
- ✓ Ostvarenje tajnosti, autentičnosti i integriteta poruke elektroničke pošte.
- ✓ Princip rada:
 - ✓ Generira se javni i privatni ključ,
 - ✓ Korisnik unosi tajni izraza (*pass phrase*),
 - ✓ Generirani ključevi se pohranjuju u certifikatima ključa,
 - ✓ Objavljuje se javni ključ.

Prednosti i nedostaci digitalnog potpisa



- ✓ nemogućnost prevare
- ✓ integritet poruka
- ✓ pravni zahtjevi
- ✓ otvoreni sustavi



- ✓ troškovi

Slijepi potpis

- ✓ Oblik digitalnog potpisa.
- ✓ Razvijen je zbog potrebe da se osigura autentičnost podataka, uz istovremeno osiguranje anonimnosti osobe koja je potpisala takav podatak.
- ✓ Važna primjena u sferi e-plaćanja.
- ✓ Matematička podloga – jednosmjerne funkcije (matematičke funkcije čije vrijednosti je moguće jednostavno odrediti, a vrlo je teško izračunati njihove inverzne vrijednosti).

Slijepi potpis

- ✓ Više modela:
 - ✓ *Random oracle model*
 - ✓ *Complexity-based proofs*
- ✓ Problem koji se javlja kod slijepog potpisa je neostavljanje tragova – slijepi potpis predstavlja priliku za “savršen” zločin.
- ✓ “Pravedni” slijepi potpis.

Slijepi potpis

- ✓ **“Savršen zločin”**
- ✓ otmičar zatraži plaćanje otkupnine u e-kovanicama
- ✓ otmičar objavi u novinama skup markiranih stringova (skrivenih svojim potpisom) - npr. 20 je skriveno množenjem s 5
- ✓ banka potpiše string 100 (npr. množenjem s 10) i objavi potpisane stringove u novinama
- ✓ otmičar kupi novine i skine maskirajući faktor (podijeli s 5) i ima potpisane e-kovanice



Slijepi potpis

- ✓ **“Pravedni” slijepi potpis:**
 - ✓ Signing protocol
 - ✓ Link recovery protocol

Slijepi potpis

- ✓ Princip rada:
 - ✓ Osoba A želi da osoba B potpiše poruku M
 - ✓ A množi M s proizvoljnim brojem ili *maskirajućim faktorom (MF)*
 - ✓ Maskiranu poruku ($M*MF$) osoba A šalje osobi B
 - ✓ Osoba B ne može pročitati poruku M (jer je maskirana), pa B ne zna sadržaj poruke M
 - ✓ Osoba B potpisuje maskiranu poruku $M*MF$ sa svojim privatnim ključem, i vraća takvu poruku k osobi A
 - ✓ Osoba A dijeli primljenu poruku s maskirajućim faktorom (MF) što rezultira s originalnom porukom koja je potpisana od strane B (npr. banke)

Slijepi potpis

- ✓ Problem – kako potpisati dokument (npr. apoen e-novca) koji ne možete pročitati?
 - ✓ Rješenje: više potpisa za različite apoene.
- ✓ Kovanje e-kovanica:
 - ✓ A šalje nepotpisane, maskirane e-kovanice u banku, uz oznaku ser. broja kovanica i apoena
 - ✓ Banka ih potpisuje slijepim potpisom i s računa osobe A skida novčani iznos
 - ✓ Banka zna vlasnika e-kovanica, ali ne zna njihove serijske brojeve – ANONIMNOST
 - ✓ Banka šalje potpisane e-kovanice osobi A koja ih demaskira.

Slijepi potpis

- ✓ Trošenje e-kovanica:
 - ✓ Osoba A plaća osobi C
 - ✓ C provjerava da li kovanice već nisu potrošene i polaže kovanice banku izdavača
 - ✓ Banka plaća prodavaču (C) novcem.
- ✓ Problem *replay-a* – kako se osigurati u slučaju višestrukog trošenja istih e-kovanica (problem kod off-line rada).
 - ✓ Kupac je anonimn – tko je odgovoran ako banka dobije već potrošene novčanice?

Slijepi potpis

- ✓ Modifikacija:
 - ✓ *Chaum double spending protocol*
 - ✓ A želi 100 e-kovanica
 - ✓ A šalje 200 e-kovanica banci na potpis
 - ✓ Za svaku e-kovanicu A kombinira “b” različitih slučajnih brojeva s brojem vlastitog računa i serijskim brojem kovanice (ex ILLI funkcija), i maskira e-kovanice
 - ✓ Banka odabire 100 od 200 e-kovanica, potpisuje ih i šalje k A, te od A traži slučajne brojeve za preostalih 100 e-kovanica (da bi pročitala broj računa)
 - ✓ Da li je A dao ispravan broj računa?
 - ✓ Da, jer je banka odabirala e-kovanice za potpis.

Slijepi potpis

- ✓ Modifikacija:
 - ✓ A plaća prodavaču C s e-kovanicama
 - ✓ Zajedno s kovanicama C zaprima i broj računa od A (u XOR obliku sa slučajnim brojem)
 - ✓ Ako je došlo do *replaya* onda je banka dobila za isti serijski broj kovanice dva različita broja koja su zaprimili prodavači C i C'
 - ✓ Njihovom i kombinacijom slučajnog broja "b" te primjenom XOR funkcije banka može doći do originalnog broja računa osobe A i identificirati počinitelja replaya.

Slijepi potpis

- ✓ Dobiven je dobar sustav:
 - ✓ Ako nije došlo do *replaya* osigurana je anonimnost kupca.
 - ✓ Ako je došlo do *replaya* moguće je identificirati počinitelja.

Slijepi potpis

- Ukoliko je broj računa 12, što je hex 0C= 00001100
- A odabire serijski broj 100 i maskirajući broj 5
- Zhtijeva od banke kovanicu sa serijskim brojem 100 x 5 = 500
- A odabire slučajan broj b i kreira b slučajnih brojeva za tu kovanicu. Uzmimo b = 6
- A radi XOR svakog slučajnog broja sa vlastitim brojem računa

i	rač.	sluč.	rač. XOR sluč.
0	0C	1B	17
1	0C	13	1F
2	0C	09	05
3	0C	05	09
4	0C	2B	27
5	0C	11	1D

Slijepi potpis

- B zaprima kovanice od A. Pronalazi b i odabire slučajan broj sa b bitova, npr. 111010
- Za svaku poziciju bita u kojoj B-ov broj ima 1, on zaprima slučajan broj od A za tu poziciju
- Za svaku poziciju sa 0, B zaprima broj računa od A XOR sa slučajnim brojem A za tu poziciju

i	rač.	sluč.	rač XOR sluč	B bit	B zaprima
0	0D	1B	17	0	17
1	0D	13	1F	1	13
2	0D	09	05	0	05
3	0D	05	09	1	05
4	0D	2B	27	1	2B
5	0D	11	1D	1	11

- B šalje zadnji stupac u banku kada polaže kovanicu

Slijepi potpis

- Sada A pokušava potrošiti kovanicu ponovno kod C. On pronalazi $b=6$ i odabire slučajni broj 010000.
- C prolazi istu proceduru kao B i šalje brojeve koje je primio u banku kamo polaže kovanicu

i	rač.	sluč.	rač. XOR sluč.	C-ov bit	C zaprima
0	0D	1B	17	0	17
1	0D	13	1F	0	13 1F
2	0D	09	05	0	05
3	0D	05	09	0	09
4	0D	2B	27	1	2B
5	0D	11	1D	0	1D

Slijepi potpis

- Banka odbija platiti C-u, jer je kovanicu položio B.
- Banka kombinira podatke od B i C korištenjem XOR gdje je pronašla podatke iz dva izvora

i	rač.	sluč.	rač. XOR sluč.	sluč. XOR rač. XOR sluč.	Vlasnik rač.
0	0C		17		
1	0C	13	1F	0C	A
2	0C		05		
3	0C	05	09	0C	A
4	0C	2B			
5	0C	11	1D	0C	A

- Ovo identificira A kao varalicu. Niti B niti A niti banka nisu to mogli napraviti sami.

Najčešći sigurnosni napadi u e-poslovanju

Prisluškivanje žičnih i bežičnih komunikacija

- iskorištava se nedovoljna (ili često puta i nepostojanje) zaštita žičnih vodova koje informacijski i telekomunikacijski sustav koristi za komunikaciju između svojih dijelova, a s ciljem prisluškivanja linija kojima putuju podaci, te nelegalnog pristupa tim istim podacima

Uskraćivanje ili degradacija usluge *Distributed Denial of Service (DDoS)*

- Prijetnja koja djeluje na smanjivanje dostupnosti informacijskog resursa.
- Dostupnost, u kontekstu sigurnosti IS-a, podrazumijeva "da sve unaprijed definirane računalne mogućnosti, sama računalna infrastruktura, programska podrška i mogućnosti upotrebe moraju biti na raspolaganju legalnom korisniku".
- Do uskraćivanja usluga može doći na različite načine, npr. fizičkim napadom (palež, eksplozije, fizičko uništenje opreme), isključivanjem pomoćnih uređaja (el. energija, generati, klimatizacijski uređaji), prirodnim katastrofama i slično, a svi oni rezultiraju fizičkim oštećenjem opreme ili zatrpavanje i preopterećenjem računalnog sustava (brisanje podataka, pretrpavanje prostora na disku, preopterećivanje procesora računala, zagušivanje mrežnog prometa)

Stražnja vrata

- stražnja vrata – *back door*
- programsko rješenje koje se koristi kod razvoja softvera kako bi se olakšao pristup programera pojedinim dijelovima aplikacije može biti zloupotrebjeno za privilegiran pristup resursima računala bez ikakvih sigurnosnih provjera i kontrola

Otmice sjednica

- otmice sjednica – *hijacking*
- korištenje tuđeg računala za izvršavanje nelegalnih aktivnosti u vremenu kad ga korisnik računala ne koristi, a nije se odjavio sa sustava (npr. prilikom odlaska na kratku pauzu ostavlja se nezaštićeno računalo);

Vremenski napadi

- vremenski napadi – sofisticiran oblik sigurnosnih prijetnji koji iskorištava princip rada računala i redoslijed izvršavanja naredbi i procesa.
- Cilj je promjenom prioriteta, odnosno redoslijeda izvršavanja procesa na računalu zaobići sigurnosne mjere i iskoristiti računalne resurse za neovlaštene aktivnosti.

Malware programi

- maliciozni računalni programi – programi poput crva, logičkih bombi, trojanskih konjeva, zamki, virusa, *hoaxeva*
- zlonamjerni te da se bez privole korisnika instaliraju na njegovo računalo (*malware* programi)
- mogućnost brzog, neprimjetnog i učinkovitog širenja, da zaobilazi obrambene mehanizme računala, da može preživjeti u zaraženom računalu bez da bude otkriven i uništen, te da na različite načine iskorištava zaraženo računalo i njegove resurse

Spoofing

- *spoofing* – "je napad u kojemu počinitelji dolaze do željenih podataka koristeći se ponajprije slabostima Interneta i nedovoljnom pažnjom korisnika".
- Riječ je o napadu u kojem počinitelji od korisnika pomoću prijevara izvlače povjerljive podatke (poput brojeva kreditnih kartica, IP adresa, zaporki, e-mail adresa) te ih iskorištavaju za kasnije aktivnosti.

Sniffing

- njuškanje za zaporkama
- programi koji prate mrežni promet s ciljem "hvatanja" dijelova u kojima korisnik unosi svoje korisničko ime i zaporku

SQL Injection

- Umetanje SQL-meta znakova u korisnički upit s ciljem izvršavanja upita u *back-end* sustavu baze
- Realizira se kroz slanje upita s jednostrukim navodnikom (') – rezultat je poruka s detaljnim informacijama o *back-end* sustavu baze ili čak i omogućen pristup *back-end* dijelovima baze (uvijek ispunjiv Boolean upit)

SQL Injection – primjer napada



SQL Injection – primjer napada

Quick Shopping Cart® Feedback: Home
Sign Out
Help
Wizard Builder
Preview Store

Froogle Data Feed Manager
View and Add Items to the Data Feed

Search: [60](#) [Catalog](#) | [Site Promotion](#) | [Orders](#) | [Members](#) | [Storefront](#) | [Processing](#) | [Reports](#) | [Publishing](#)

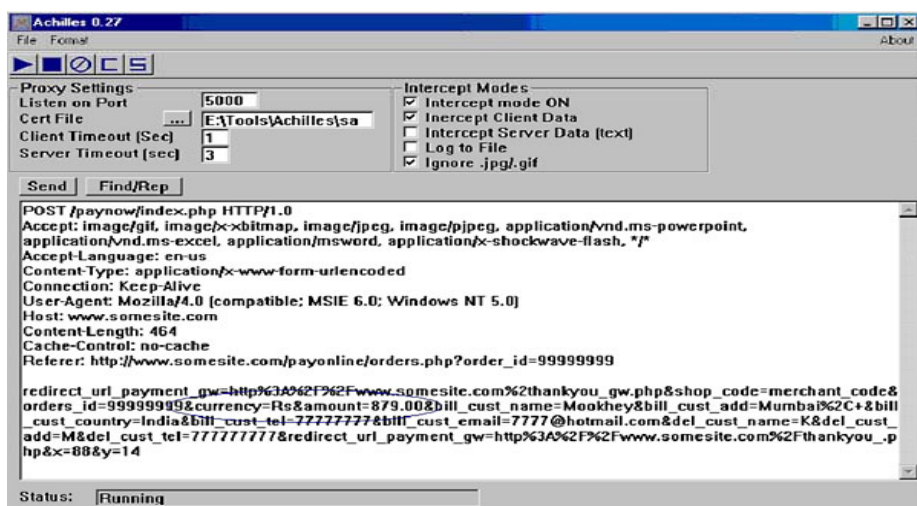
Tasks	Product Name	SKU/Part Number	Inventory
<input type="checkbox"/> Set Up Data Feed	<input type="checkbox"/> 1" Sprocket	123123	93
<input type="checkbox"/> Turn Off Data Feed	<input type="checkbox"/> 10.00 product	1000	N/A
About Froogle Data Feed Setup This page displays your entire product catalog and lets you select specific products (or all of them) to include in the Froogle data feed. Click Data Feed Set Up in the Tasks menu to modify the set up instructions for the data feed. Click Turn On/Off Data Feed to stop and start the transmission of your product information through the data feed. Froogle Help 	<input type="checkbox"/> 150.00 product	15000	N/A
	<input type="checkbox"/> 19.99 product	1999	N/A
	<input checked="" type="checkbox"/> 2" Sprocket	456456	25
	<input type="checkbox"/> 20.00 product	2000	N/A
	<input type="checkbox"/> 25.00 product	2500	N/A
	<input type="checkbox"/> 45.00 product	4500	N/A
	<input type="checkbox"/> 75.00 product	7500	N/A
	<input type="checkbox"/> 9.99 product	999	N/A
	<input checked="" type="checkbox"/> Daisies	456xxx	50
	<input type="checkbox"/> Dime	10	N/A
	<input checked="" type="checkbox"/> Download limited to one day	DL-1-day	N/A
	<input checked="" type="checkbox"/> Download limited to one day 2	DL-one-day	N/A
	<input checked="" type="checkbox"/> Download limited to one time	DL-1-time	N/A

1 2 next

Manipulacija cijenama

- Napad karakterističan samo za e-poslovanje
- Ukupan iznos za plaćanje se sprema u skriveno polje dinamički kreirane web stranice
- Napadač koristi proxy poslužitelj web aplikacije kako bi izmijenio iznos u trenutku njegove razmjene između web poslužitelja i korisničkog web preglednika

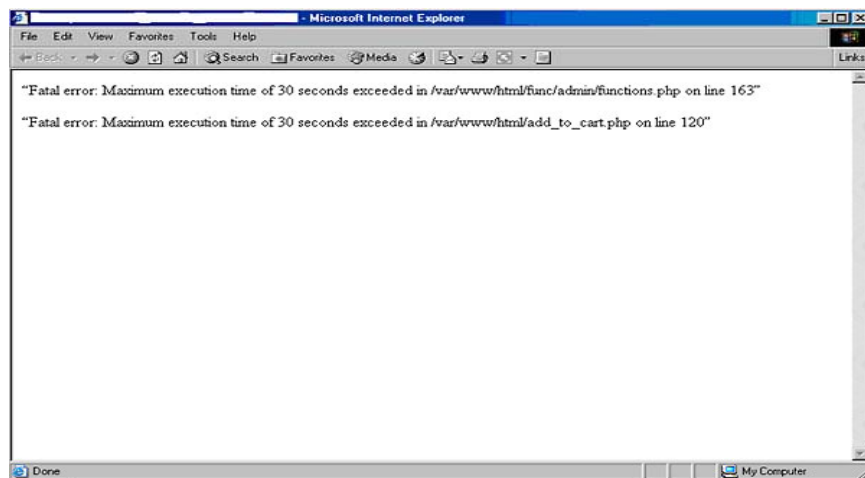
Manipulacija cijenama - primjer



Buffer overflow

- Slanje velike količine podataka (bitova) web aplikacijama koje nisu razvijene s ciljem obrade velikih količina podataka rezultira neželjenim posljedicama.
- Cilj napada je dobiti informaciju o pogrešci koja najčešće sadrži detaljnu putanju *admin* dijela poslužitelja.

Buffer overflow – PHP time out



Buffer overflow – iskorištavanje poruke o pogrešci za daljnje napade

```

java.lang.NumberFormatException: AA
    at java.lang.Integer.parseInt(Integer.java:409)
    at java.lang.Integer.parseInt(Integer.java:458)
    at _scripts._menu._jspService(_menu.java:57)
    [SRC:/scripts/menu.jsp:8]
    at com.orionserver[Oracle9iAS (9.0.3.0.0)] Containers for J2EE].http.OrionHttpJspPage.service(OrionHttp
    at oracle.jsp.runtimev2.JspPageTable.service(JspPageTable.java:317)
    at oracle.jsp.runtimev2.JspServlet.internalService(JspServlet.java:465)
    at oracle.jsp.runtimev2.JspServlet.service(JspServlet.java:379)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
    at com.evermind[Oracle9iAS (9.0.3.0.0)] Containers for J2EE].server.http.ResourceFilterChain.doFilter(
    at oracle.security.jazn.oc4j.JAZNFilter.doFilter(JAZNFilter.java:283)
    at com.evermind[Oracle9iAS (9.0.3.0.0)] Containers for J2EE].server.http.ServletRequestDispatcher.invo
    at com.evermind[Oracle9iAS (9.0.3.0.0)] Containers for J2EE].server.http.ServletRequestDispatcher.incl
    at com.evermind[Oracle9iAS (9.0.3.0.0)] Containers for J2EE].server.http.EvermindPageContext.include(E
    at _scripts._mtemplate._jspService(_mtemplate.java:138)
    [SRC:/scripts/mtemplate.jsp:143]
    at com.orionserver[Oracle9iAS (9.0.3.0.0)] Containers for J2EE].http.OrionHttpJspPage.service(OrionHttp
    at oracle.jsp.runtimev2.JspPageTable.service(JspPageTable.java:317)
  
```

XSS

- *Cross-site scripting*
- Napad usmjeren na krajnjeg korisnika
- Temelji se na:
 - Nedostatnoj verifikaciji od strane web aplikacije
 - Povjerenju krajnjeg korisnika u URL web mjesta koje je napadnuto

XSS

- Korisnik unosi podatke u web formu, koja ih obrađuje i ispisuje ih na web stranici zajedno sa originalnim korisničkim unosom.
- Ako nad podacima nije izvršen *parsing* napadač može umetnuti dio JavaScript koda kao korisnički unos.
- Takvim kodom moguće je modificirati URL, a korisnik ne sluteći prijevaru jednim klikom na “provjereni URL” pokreće izvođenje skripte na svom računalu.

XSS

- Najčešći motiv XSS napada je:
 - Ukrasti *cookie* (zbog informacija o sesiji koje on sadrži)
 - Preusmjeriti korisnika na napadačevo web mjesto s kojeg će se izvršiti maliciozni kod pomoću ActiveX kontrola
 - Realizirati *phishing* prijevaru – preusmjeriti korisnika na web stranicu koja izgleda kao originalna, ali to nije.