Informaciona bezbednost

Uvod u kriptografiju

dr Milan Stojkov

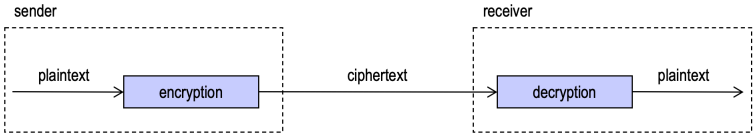
Katedra za informatiku

Fakultet tehničkih nauka
Univerzitet u Novom Sadu

2022.

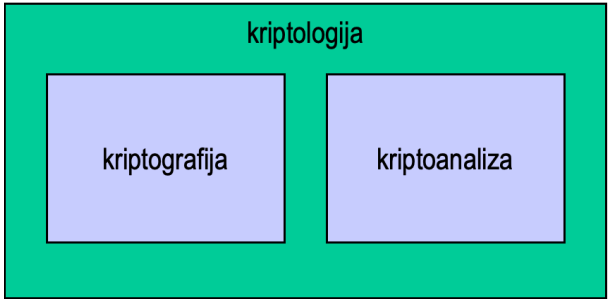
Terminologija

- pošiljalac / sender
- primalac / receiver
- poruka / message
  - otvoreni tekst / plaintext
  - šifrovani tekst / ciphertext
- šifrovanje / encryption
- dešifrovanje / decryption



Terminologija

- kriptografija / cryptography
  - obezbeđivanje tajnosti poruka
- kriptoanaliza / cryptanalysis
  - čitanje tajnih poruka
- kriptologija / cryptology
  - kriptografija + kriptoanaliza



Osnovni zadaci kriptografije

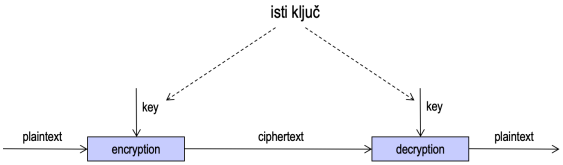
- poverljivost poruka
  - očuvanje tajnosti poruka u komunikaciji između pošiljaoca i primaoca
- autentifikacija
  - potvrđivanje porekla poruke
- integritet
  - čuvanje sadržaja poruke od (zlonamernih/slučajnih) izmena
- neporecivost
  - nemogućnost pošiljaoca da negira slanje svoje poruke

Šifre i ključevi

- kriptografski algoritam (šifra) je skup dve matematičke funkcije
  - za šifrovanje
  - za dešifrovanje
- algoritam može biti
  - tajni
    - nema kontrole kvaliteta → slaba sigurnost
  - javni
- tajnost se postiže ključevima
- sigurnost komunikacije leži u tajnosti ključeva

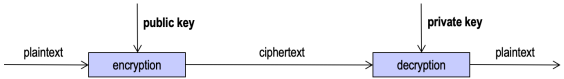
Simetrični algoritmi

- ključ za dešifrovanje se može izračunati na osnovu ključa za šifrovanje i obrnuto
- najčešće su ova dva ključa jednaka
- pošiljalac i primalac se moraju dogovoriti o korišćenom ključu pre šifrovane komunikacije



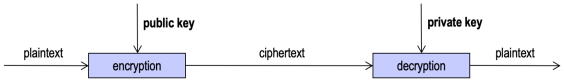
Asimetrični algoritmi

- algoritmi sa javnim ključem (public-key algorithms)
- ključ za šifrovanje se razlikuje od ključa za dešifrovanje
- ključ za dešifrovanje se ne može (u razumnom vremenu!) izračunati na osnovu ključa za šifrovanje
- ključ za šifrovanje može biti javni
  - bilo ko može uputiti šifrovanu poruku primaocu
  - samo je primalac može dešifrovati
- ključ za šifrovanje = “javni ključ”
- ključ za dešifrovanje = “tajni ključ”



Kriptoanaliza

- svrha kriptografije
  - čuvanje tajnosti otvorenog teksta i ključeva
- svrha kriptoanalize
  - pristup otvorenom tekstu bez prethodnog poznavanja ključa
- kriptoanaliza podrazumeva da napadač poznaje
  - korišćeni algoritam i
  - detalje implementacije
- oslanjanje na tajnost algoritma nije osnova za sigurnost komunikacije



Tipovi napada

1. cyphertext-only (known-ciphertext)
   - napadač poseduje sadržaj šifriranih poruka
   - pokušava da dođe do otvorenog sadržaja poruka ili da izračuna ključ za dešifrovanje
2. known-plaintext
   - napadač poseduje sadržaj otvorenih poruka i odgovarajućih šifriranih poruka
   - pokušava da dođe do ključa ili da razvije algoritam koji bi mu omogućio da dešifruje sve dalje poruke
3. chosen-plaintext
   - slično prethodnom, ali napadač može i da bira koji tekst će biti šifrovan
4. adaptive-chosen-plaintext
   - slično prethodnom, ali napadač može i da bira tekst za šifrovanje na osnovu rezultata svojih prethodnih pokušaja

Tipovi napada

5. chosen-ciphertext
   - napadač može da analizira bilo koju odabranu šifriranu poruku zajedno sa njenim odgovarajućim otvorenim porukama
   - pokušava da dođe do ključa ili da dobije što više informacija o sistemu koji napada
6. chosen-key
   - napadač poseduje informacije o odnosima između različitih ključeva
   - napadač ima za cilj da sruši sistem koji se oslanja na te ključeve
7. rubber-hose
   - napadač pokušava da dođe do ključa pretnjama, ucenom, podmićivanjem, mučenjem

Kriptografija pre i posle pojave računara

- pre pojave računara
  - algoritmi zasnovani na karakterima (slovima)
  - šifre zamene: zamena karaktera drugim
  - šifre premeštanja: premeštanje karaktera u tekstu
- posle pojave računara
  - algoritmi rade nad nizovima bitova
  - alfabet nema 26 nego 2 znaka
  - mnogi današnji algoritmi kombinuju zamenu i premeštanje

Šifre zamene

- svaki karakter otvorenog teksta se zamenjuje nekim drugim znakom u šifrovanom tekstu
- monoalfabetske šifre
  - svaki karakter otvorenog teksta se zamenjuje jednim znakom u šifrovanom tekstu
- homofonske šifre
  - jednom karakteru otvorenog teksta odgovara više karaktera u šifrovanom tekstu
- poligramske šifre
  - zamena se vrši nad grupama karaktera
- polialfabetske šifre
  - koristi se više monoalfabetskih šifara koje se smenjuju sa svakim šifriranim znakom

Šifre zamene

- Cezarova šifra: svaki znak se zamenjuje znakom udaljenim za 3 u desno u alfabetu A $\rightarrow$ D, B $\rightarrow$ E, ...
- ROT13: svaki znak se rotira za 13 mesta - text = ROT13(ROT13(text))
- napadi na jednostavne šifre zamene zasnivaju se na statističkim karakteristikama jezika

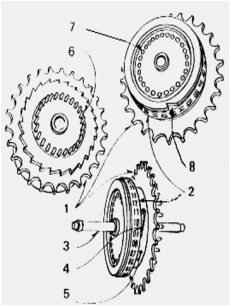
Šifre premeštanja

- znakovi u otvorenom tekstu se premeštaju
- primer: kolonska zamena

```
COMPUTER GRAPHICS MAY BE SLOW BUT AT LEAST IT'S EXPENSIVE
                          ⇩
                  COMPUTERGR
                  APHICSMAYB
                  ESLOWBUTAT
                  LEASTITSEX
                  PENSIVE
                          ⇩
CAELPOPSEEMHLANPIOSSUCWTITSBIVEMUTERATSGYAERBTX
```

Rotorske mašine

- rotorska mašina ima tastaturu i niz rotora
  - rotor predstavlja permutaciju alfabeta
  - rotori su međusobno povezani
  - rotori se okreću u različitim koracima
  - period ponavljanja za n-rotorsku mašinu je 26^n
- Enigma
  - radi sa tri rotora iz skupa od pet
  - svaki rotor se primeni dva puta za jedan znak
  - permutacija alfabeta pre rotora



Jednokratna sveska - savršena šifra

- one-time pad: veliki neponavljajući niz slučajnih slova
  - beskonačna traka za teleprintere
- šifrovanje znaka: sabiranje znaka iz otvorenog teksta sa znakom iz niza po modulu 26
- kada se znak iz niza upotrebi, ne može se više koristiti
- pošiljalac i primalac moraju posedovati istu beskonačnu traku
- sigurnost ove šifre zavisi od beskonačne trake
  - ona mora da sadrži zaista slučajan niz, a ne rezultat rada generatora pseudo-slučajnih brojeva
- svaki niz znakova je podjednako verovatan kandidat za ključ
  - svaki dešifrovani tekst je podjednako verovatan
- problemi:
  - generisanje zaista slučajnog niza
  - distribucija niza
  - sinhronizacija učesnika u komunikaciji
  - konačna veličina trake u praksi

Protokoli

- serija postupaka, sa najmanje dva učesnika, namenjena obavljanju nekog zadatka
- svi učesnici moraju poznavati protokol i znati potrebne korake unapred
- svi učesnici u protokolu se moraju dogovoriti da ga koriste
- protokol mora biti nedvosmislen
  - koraci moraju biti dobro definisani
  - ne sme biti prilike za nesporazum
- protokol mora biti kompletan
  - mora imati definisanu akciju za svaku moguću situaciju
- ne bi trebalo da je moguće učiniti više ili saznati više nego što je predviđeno protokolom

Protokoli

- primer: kupoprodaja automobila
- uloge
  - Alice: prodavac
  - Bob: kupac
  - Trent: advokat
- scenario
  - Alice daje potpisan ugovor Trentu
  - Bob daje ček Alice
  - Alice podiže novac
  - Trent čeka određeni period vremena da Alice javi da li je ček prošao
- ako jeste, daje ugovor Bobu
- ako nije, vraća ugovor Alice
- ako se Alice ne javi na vreme, daje ugovor Bobu

Protokoli

- standardne uloge u literaturi

Alice	First participant in all the protocols
Bob	Second participant in all the protocols
Carol	Participant in the three- and four-party protocols
Dave	Participant in the four-party protocols
Eve	Eavesdropper
Mallory	Malicious active attacker
Trent	Trusted arbitrator
Walter	Warden; guards Alice and Bob in some protocols
Victor	Verifier

Tipovi protokola

- arbitrated
- adjudicated
- self-enforcing

Tipovi protokola - Arbitrated

- arbitrator je treća osoba kojoj svi veruju i koja nije zainteresovana za ishod protokola
- u realnom svetu, advokati, banke i notari se koriste kao arbitratori
- Trent će igrati ulogu arbitratora

Tipovi protokola - Arbitrated problemi

- lakše je naći treću osobu u realnom svetu nego na mreži
- uvek postoji dodatno kašnjenje
- arbitrator mora da učestvuje u svakoj transakciji i tako postaje usko grlo u komunikaciji
- pošto svi veruju arbitratoru, on postaje ranjiva tačka

Tipovi protokola - Adjudicated

- može se podeliti na dva potprotokola nižeg nivoa
  - jedan nearbitrirani protokol će se pokrenuti svaku put kada učesnici žele da kompletiraju protokol
  - drugi je arbitrirani protokol koji se pokreće samo kada postoji spor između učesnika
- adjucitator takođe nije zainteresovana strana za ishod protokola
  - poziva se samo je potrebno utvrditi da li je protokol izvršen pravično
- sudije su profesionalni adjudicatori

Tipovi protokola - Self-enforcing

- sam protokol garantuje pravičnost
  - nije potreban arbitrator da se kompletira protokol
  - nije potreban adjudicator da se reši spor
- protokol je dizajniran tako da ne može biti sporova
- varanje se može otkriti a protokol prekinuti
- protokoli koji se samosprovode ne postoje za svaku situaciju

Napadi na protokole

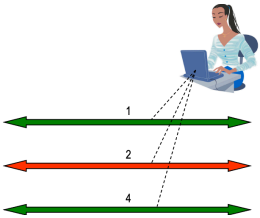
- pasivni napadi
  - neko ko nije uključen u protokol pokušava da prisluškuje neki deo ili ceo protokol
  - teško je otkriti, pa pokušavamo da sprečimo prisluškivanje
- aktivni napadi
  - napadač može da pokuša da izmeni protokol u svoju korist
  - napadač se pretvara da je neko drugi
  - napadač menja poruke, uvodi nove poruke, briše postojeće poruke, reprodukuje stare poruke
- varalice
  - učesnici koji su uključeni u protokol
  - pasivne varalice prate protokol ali pokušavaju da dobiju više informacija o njemu
  - aktivne varalice ometaju protokol koji je u toku da bi varali

Komunikacija pomoću simetričnih algoritama

1. Alice i Bob dogovore algoritam
2. Alice i Bob dogovore ključ
3. Alice svoju poruku šifruje dogovorenim algoritmom i ključem
4. Alice šalje šifrirani tekst Bobu
5. Bob dešifruje poruku istim algoritmom i ključem

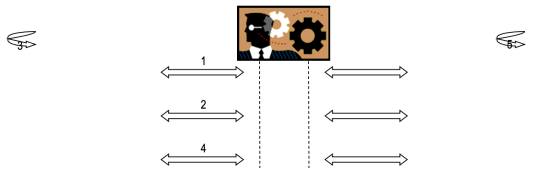
Komunikacija pomoću simetričnih algoritama

- mogućnosti napada – Eve
  - prisluškuje komunikaciju u koraku 4
    - known-ciphertext napad
  - prisluškuje komunikaciju u koraku 1
    - dopustivo, postoje javni simetrični algoritmi koji su dovoljno dobri
  - prisluškuje komunikaciju u koraku 2
    - nije dopustivo! Alice i Bob moraju dogovoriti ključ u tajnosti
    - ključ mora ostati tajan sve dok poruke koje su njime šifrovane moraju ostati tajne



Komunikacija pomoću simetričnih algoritama

- mogućnosti napada – Mallory
  - pokušava da presretne komunikaciju između Alice i Boba
  - ako je presretne u koraku 2, može da šalje lažne poruke



Komunikacija pomoću simetričnih algoritama

- ključevi se moraju distribuirati sigurnim komunikacionim kanalom
- ako je ključ kompromitovan
  - sve poruke šifrovane njime su kompromitovane
  - napadač može da se lažno predstavlja kao učesnik u komunikaciji
- ako se koristi poseban ključ za komunikaciju svakog para učesnika u mreži
  - za mrežu od n učesnika potrebno je n(n-1)/2 ključeva
  - 10 korisnika $\rightarrow$ 45 ključeva
  - 100 korisnika $\rightarrow$ 4950 ključeva

Jednosmerne funkcije

- funkcije čiji rezultat je lako izračunati, ali rezultat inverzne funkcije nije, odnosno
  - za dato x lako je izračunati f(x),
  - za dato f(x) nije lako izračunati x
- lako/teško: računska složenost algoritma
- ne postoji matematički dokaz da jednosmerne funkcije postoje, ali za neke funkcije možemo reći da su jednosmerne jer ne znamo lak način da izračunamo inverznu funkciju
- primer: x^2 u konačnom polju je lako izračunati, ali $x^{1/2}$ nije
- trapdoor one-way functions
  - vrednost funkcije može se lako izračunati
  - inverzna vrednost može se lako izračunati ako se zna neka tajna

Jednosmerne hash funkcije

- jednosmerne funkcije koje imaju
  - ulaz promenljive dužine
  - izlaz fiksne dužine
- primer: java.lang.String.hashCode()
  $h = s[0] * 31^{n-1} + s[1] * 31^{n-2} + ... + s[n-1]$

Jednosmerne hash funkcije

- collision-free funkcije
  - teško je generisati dva ulaza koji daju isti izlaz
- funkcije su javne
- tajnost je sadržana u jednosmernosti
  - promena jednog bita u ulazu menja u proseku polovinu bitova na izlazu
- message authentication codes (MAC)
  - jednosmerna hash funkcija + ključ za šifrovanje
  - hash vrednost može da proveri samo onaj ko ima i ključ

Komunikacija pomoću asimetričnih algoritama

1. Alice i Bob dogovore algoritam
2. Bob šalje Alice svoj javni ključ
3. Alice šifruje svoju poruku Bobovim javnim ključem
4. Alice šalje šifrovanu poruku Bobu
5. Bob dešifruje poruku svojim tajnim ključem

Komunikacija pomoću asimetričnih algoritama

- nema problema sa razmenom tajnih ključeva – tajni ključevi se ne razmenjuju
- grupa učesnika u komunikaciji može da
  - usvoji jedinstveni asimetrični algoritam i
  - formira bazu podataka sa ključevima

1. Alice uzima Bobov javni ključ iz baze podataka
2. Alice šifruje svoju poruku Bobovim javnim ključem
3. Alice šalje šifrovanu poruku Bobu
4. Bob dešifruje poruku svojim tajnim ključem

Komunikacija pomoću asimetričnih algoritama

- osnova asimetričnih algoritama: trapdoor one-way funkcije
  - enkripcija: “lak” smer
  - dekripcija (bez tajnog ključa): “težak” smer

Komunikacija pomoću asimetričnih algoritama

- asimetrični algoritmi ne predstavljaju univerzalnu zamenu za simetrične
  - spori su
- najmanje 1000 puta sporiji od simetričnih
  - ranjivi su na chosen-plaintext napade
- ako je tajna poruka jedna iz konačnog skupa od n poruka, potrebno je šifrovati (javnim ključem!) svih n poruka i rezultat uporediti sa tajnom porukom

Hibridni kriptosistemi

- asimetrični algoritmi se koriste za razmenu ključeva za simetrične algoritme
- ključ za simetričan algoritam se koristi samo u jednoj sesiji (session key)
- potencijalni (ali mnogo manji) problem – kompromitovanje tajnog ključa

1. Bob šalje Alice svoj javni ključ
2. Alice generiše slučajni ključ za sesiju, šifruje ga Bobovim javnim ključem i šalje Bobu
3. Bob dešifruje poruku i dobija ključ za sesiju
4. Alice i Bob nastavljaju komunikaciju koristeći ključ za sesiju

Digitalni potpisi

- osobine klasičnih potpisa
  - autentičnost - potvrđuje da je baš potpisnik potpisao dokument
  - nije ponovo iskoristiv - potpis se ne može preneti na drugi dokument
  - potpisani dokument - je nepromenljiv nakon što se dokument potpiše, ne može se više menjati
  - neporeciv potpisnik - ne može kasnije poricati da je potpisao dokument

Digitalni potpisi

- pomoću simetričnog algoritma i arbitratora
  - treći učesnik je arbitrator kome obe strane veruju
  - ključevi za simetrični algoritam su definisani između svih učesnika i arbitratora
  - postupak:

1. Alice šifruje svoju poruku za Boba ključem K_A i šalje je Trentu
2. Trent dešifruje poruku pomoću K_A
3. Trent šifruje poruku i izjavu da ju je primio od Alice ključem K_B
4. Trent šalje šifrovanu poruku Bobu
5. Bob dešifruje poruku ključem K_B i može da pristupi i originalnoj poruci i Trentovoj potvrdi da je poruka stigla od Alice

Digitalni potpisi

- pomoću simetričnog algoritma i arbitratora
  - autentičnost
  - nije ponovo iskoristiv
  - nepromenljivost potpisanog dokumenta
  - neporecivost
- obavezan uslov
  - svi veruju Trentu
  - Trent ne pravi greške
  - ALI Trent može biti preopterećen u uslovima velikog broja učesnika i/ili intenzivne komunikacije

Digitalni potpisi

- pomoću asimetričnog algoritma
  1. Alice šifruje poruku svojim tajnim ključem – time je i potpisuje
  2. Alice šalje poruku Bobu
  3. Bob dešifruje poruku Alicinim javnim ključem – time potvrđuje i potpis

Digitalni potpisi

- pomoću asimetričnog algoritma
  - Bob može da pokuša da više puta iskoristi isti Alicin dokument (npr. ček)
- u potpisani dokument se ugrađuje i timestamp
- praktična ograničenja
  - asimetrični algoritmi su previše spori za potpisivanje velikih dokumenata

Digitalni potpisi

- pomoću jednosmernih hash funkcija
  - umesto da se potpisuje ceo dokument, potpisuje se njegov hash
  - postupak

1. Alice izračunava hash svog dokumenta
2. Alice šifrira hash svog dokumenta pomoću svog tajnog ključa – time potpisuje hash
3. Alice šalje dokument i potpisani hash Bobu
4. Bob izračunava hash primljenog dokumenta; dešifruje primljeni hash; ako su dve hash vrednosti jednake, potpis je ispravan

Digitalni potpisi

- pomoću jednosmernih hash funkcija
  - višestruki potpisi istog dokumenta
  - postupak

1. Alice potpisuje hash dokumenta
2. Bob potpisuje hash dokumenta
3. Bob šalje svoj potpis Alice
4. Alice šalje dokument i oba potpisa Carol
5. Carol proverava i Alicin i Bobov potpis

Digitalni potpisi

- digitalni potpisi sa šifrovanjem dokumenata
  1. Alice potpisuje poruku svojim privatnim ključem
  2. Alice šifruje poruku Bobovim javnim ključem
  3. Alice šalje poruku Bobu
  4. Bob dešifruje poruku svojim tajnim ključem
  5. Bob proverava potpis Alicinim javnim ključem i dobija otvorenu poruku

Sertifikati

- šta je sertifikat

We certify that the domain **www.somefirm.com** is the owner of the following public key:

This key is valid from: **01/01/2004**
This key expires on: **01/01/2006**

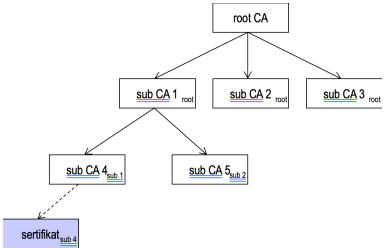
Signed,
VeriSign, Inc.
Certificate Authority

Sertifikati

- Certificate Authority (CA)
  - (pravno) lice od poverenja
  - njegov javni ključ je poznat
  - potpisuje sertifikate

Sertifikati

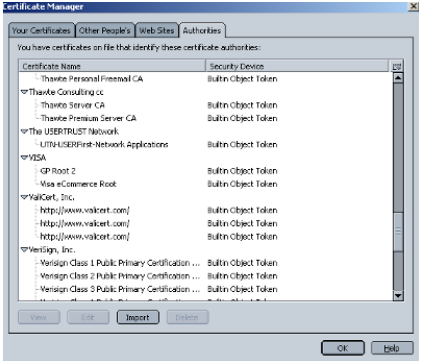
- ulančavanje sertifikata (certificate chaining)
  - CA može da izda sertifikat sa naznakom da je primalac ovlašćen da izdaje dalje sertifikate
  - CA hijerarhija



- lanac sertifikata – putanja od neposrednog CA do korenskog CA

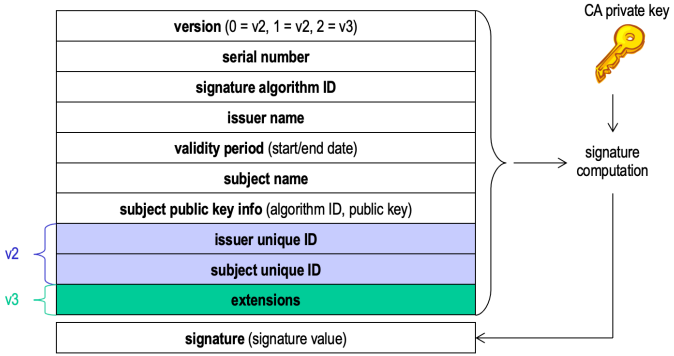
Sertifikati

- “root CA” je self-signed
- browseri sadrže “root CA” sertifikate
- svaki sertifikat sadrži “CA flag”
  - da li vlasnik ima pravo da izdaje nove sertifikate, tj. da li je vlasnik takođe CA



Sertifikati

- X.509 standard



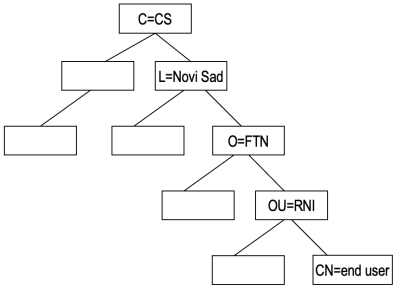
Sertifikati

- X.509 standard
  - primer sertifikata

Version: 3 (0x2)
Serial Number: 1 (0x1)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=CS, L=Novi Sad, O=FTN, OU=Odeljenje za sertifikate, CN=FTN CA, Email=ca@uns.ac.rs
Validity:
Not Before: Jun 8 10:00:00 2004 GMT
Not After: Jun 7 10:00:00 2005 GMT
Subject: C=CS, L=Novi Sad, O=FTN, OU=Katedra za informatiku, CN=Milan Stojkov, Email=stojkovm@uns.ac.rs
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit): 00:b3:4e:75:76:fc:4c:c3:bd:61:6c:14:41:8f:47:...
Exponent: 65537 (0x10001)
X.509v3 Extensions:
X.509v3 Basic Constraints
CA: false
Netscape Comment:
OpenSSL Generated Certificate
X.509v3 Subject Key Identifier:
a6:db:b8:78:19:7a:c4:67:23:de:03:a3:ee:d4:26:5e:78:14:71:61
Signature:
9f:15:a8:cb:6c:a9:0d:d4:61:24:b9:7a:bc:29:e4:29:8b:4c:...

Sertifikati

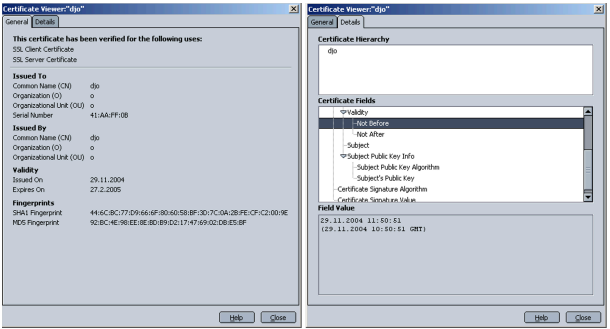
- X.509 standard
  - hijerarhijska organizacija imena (C=CS, L=Novi Sad, O=FTN, ...) potiče od X.500 standarda, sveobuhvatnog direktorijumskog servisa



- svaki čvor stabla ima svoj CA

Sertifikati

- X.509 standard
  - problem distribucije ključeva pretvoren je u problem distribucije imena
- ljudi sa istim imenom i prezimenom u istoj organizaciji
- kreiranje jedinstvenih naziva – pretraživanje po imenu više nema smisla
  - John Smith 1 vs John Smith 2 vs John Smith 3



Razmena ključeva

- šifrovanje svake pojedine konverzacije posebnim ključem $\sim$ session key
- distribucija ključeva za sesije je poseban problem

Razmena ključeva

- pomoću simetričnog algoritma
  1. Alice traži od Trenta novi session key
  2. Trent generiše session key i šifruje ga dva puta: pomoću Alicinog i Bobovog ključa i obe poruke šalje Alice
  3. Alice dešifruje svoju session key kopiju
  4. Alice šalje Bobu njegovu session key kopiju
  5. Bob dešifruje svoju kopiju
  6. Alice i Bob koriste session key za dalju komunikaciju
- Trent mora biti apsolutno siguran
- Trent je potencijalno usko grlo

Razmena ključeva

- pomoću asimetričnog algoritma
  1. javna baza podataka sa svim potpisanim javnim ključevima = KDC (Key Distribution Center)
  2. Alice uzima Bobov javni ključ iz KDC
  3. Alice generiše slučajni session key, šifruje ga Bobovim javnim ključem i šalje ga Bobu
  4. Bob dešifruje Alicinu poruku svojim tajnim ključem
  5. dalja komunikacija koristi session key

Razmena ključeva

- šta može Eve?
  - ciphertext-only napad

Razmena ključeva

- šta može Mallory?
  - man-in-the-middle napad
    1. Alice šalje Bobu svoj javni ključ; Mallory presreće ovaj ključ i šalje Bobu svoj javni ključ
    2. Bob šalje Alice svoj javni ključ; Mallory presreće i ovaj ključ i šalje Alice svoj javni ključ
    3. kada Bob šalje Alice poruku, Mallory je presreće i otvara (svojim ključem), šifruje ponovo Alicinim i šalje njoj
    4. isto i kada Alice šalje poruku Bobu
  - Alice i Bob nemaju način da provere da li zaista komuniciraju

Razmena ključeva

- interlock protokol
  1. Alice šalje Bobu svoj javni ključ
  2. Bob šalje Alice svoj javni ključ
  3. Alice šifruje poruku Bobovim javnim ključem, i šalje polovinu poruke Bobu
  4. Bob šifruje svoju poruku Alicinim javnim ključem, i šalje polovinu poruke Alice
  5. Alice šalje drugu polovinu poruke Bobu
  6. Bob rekonstruiše celu poruku iz dve njene polovine, potom šalje svoju drugu polovinu
  7. Alice rekonstruiše celu poruku iz dve polovine
- “polovina poruke”
  - svaki drugi bit
  - prva polovina – hash, druga polovina – sama poruka

Razmena ključeva

- šta može Mallory?
  - može da presretne ključeve iz koraka 1. i 2.
  - presretanjem polovine poruke iz koraka 3, ne može je dešifrovati svojim tajnim ključem pa šifrovati Bobovim javnim ključem – to više neće biti ta polovina
  - isto i u drugom smeru

Razmena ključeva

- razmena ključeva sa digitalnim potpisima
  - potreban je KDC (Trent) koji potpisuje sve javne ključeve
  - kada Alice i Bob prime javne ključeve, proveravaju ih pomoću Trentovog potpisa
  - Mallory ne može da zameni javne ključeve prilikom presretanja komunikacije, jer je njegov javni ključ potpisan od strane Trenta da je njegov
- kompromitacija KDC
  - ako Mallory upadne u KDC, dobija Trentov privatni ključ
  - ovaj ključ mu omogućava da podmeće lažne javne ključeve, ali
  - da bi mogao da dešifruje session ključeve, mora biti u stanju da presreće i menja poruke između Alice i Boba

Razmena ključeva

- komunikacija bez prethodne razmene ključeva
  1. Alice generiše session ključ, i njime šifruje poruku
  2. Alice uzima Bobov javni ključ iz KDC
  3. Alice šifruje session ključ Bobovim javnim ključem
  4. Alice šalje šifrovanu poruku i šifrovani ključ Bobu (ceo paket se može potpisati)
  5. Bob dešifruje session ključ svojim tajnim ključem
  6. Bob dešifruje poruku session ključem

Ostali protokoli

- autentifikacija
  - jednosmerne hash funkcije
  - međusobna autentifikacija pomoću interlock protokola
  - Kerberos
- podela tajne
- timestamping
- zero-knowledge proofs
- blind signatures
- elektronske glasačke mašine
- digitalni keš

Dužina ključeva

- sigurnost kriptosistema zavisi od
  - sigurnosti algoritma
  - dužine ključa
- siguran algoritam:
  - nema boljeg načina za razbijanje od brute-force napada
  - teško dostižno u praksi
- brute-force napad je known-plaintext napad
  - realna mogućnost ovakvog napada

Dužina ključeva

- simetrični algoritmi
  - ključ dužine n bita $\rightarrow$ postoji 2^n mogućih ključeva / pokušaja
  - verovatnoća je 50% da će traženi ključ biti u prvoj polovini pokušaja
  - problem idealan za paralelno procesiranje
- mogućnost pronalaženja ključa: procena troškova i vremena
  - Murov zakon: procesna moć se duplira svakih 18 meseci
  - troškovi se smanjuju 10 puta svakih 5 godina
  - oprema koja je 2012. koštala 1.000.000 € u 2020. košta 100.000 €
- brzina otkrivanja ključa $\sim$ količina novca
- procena "vrednosti ključa", tj. vrednosti informacija koje ključ čuva
- vrednost informacija može da opada vremenom
- ključ vredan 100 € nema smisla razbijati opremom od 10.000.000 €
- dužina ključa se projektuje prema potrebnoj dužini trajanja tajnosti ključa

Dužina ključeva

- simetrični algoritmi
  - hardverski i softverski sistemi za razbijanje
- softver je oko 1000 puta sporiji od hardvera
- veliki broj besposlenih servera na Internetu koji se može angažovati (sa ili bez volje administratora)
- dužina ključa klasičnih algoritama: 56-256 bita
- preporuka da od 2016. ključevi budu veći od 112 bita

Procena potrebnog vremena/novca za hardverski brute-force napad

Cena	56 bita	64 bita	80 bita	112 bita	128 bita
$100K	35 sati	1 godina	70.000 god	10^{14} godina	10^{19} godina
$1M	3.5 sata	37 dana	7000 godina	10^{13} godina	10^{18} godina
$10M	21 minut	4 dana	700 godina	10^{12} godina	10^{17} godina
$100M	2 minuta	9 sati	70 godina	10^{11} godina	10^{16} godina
$1G	13 sekundi	1 sat	7 godina	10^{10} godina	10^{15} godina
$10G	1 sekunda	5.4 minuta	245 dana	10^{9} godina	10^{14} godina
$100G	0.1 sekunda	32 sekunde	24 dana	10^{8} godina	10^{13} godina
$1T	0.01 sekunda	3 sekunde	2.4 dana	10^{7} godina	10^{12} godina

Dužina ključeva

- simetrični algoritmi
  - granice dužine
- drugi zakon termodinamike
  - čuvanje jednog bita promenom stanja sistema troši energije minimalno kt
  - računar koji radi na temperaturi pozadinskog zračenja (3,2°K) bi potrošio energiju koju Sunce emituje tokom 32 godine da napaja 192-bitni brojač koji će da obrne ceo krug
  - napajanje pomoću supernove $\rightarrow$ 219-bitni brojač
  - 256-bitni ključevi će trajati koliko i Vasiona

Dužina ključeva

- asimetrični algoritmi
  - brute-force napad ne predstavlja testiranje svih mogućih ključeva, nego izračunavanje tajnog ključa na osnovu javnog
  - tajni $\rightarrow$ javni $\sim$ množenje dva velika prosta broja - računski jednostavno
  - javni $\rightarrow$ tajni $\sim$ faktorisanje količnika - (za sada) računski složeno
  - period tajnosti ključeva može se odabrati proizvoljno dugo (sa današnjim znanjem matematike)

Dužina ključeva

- NIST preporučena dužina ključeva

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm		Elliptic Curve	Hash (A)	Hash (B)
				Key	Group			
(Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1**	
2016 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2016 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1
2016 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224
2016 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512

Dužina ključeva

- NIST preporučeno trajanje ključeva

Key Type *Move the cursor over a type for description*	Cryptoperiod Originator Usage Period (OUP)	Recipient Usage Period
Private Signature Key	1-3 years	-
Public Signature Key	Several years (depends on key size)	
Symmetric Authentication Key	<= 2 years	<= OUP + 3 years
Private Authentication Key	1-2 years	
Public Authentication Key	1-2 years	
Symmetric Data Encryption Key	<= 2 years	<= OUP + 3 years
Symmetric Key Wrapping Key	<= 2 years	<= OUP + 3 years
Symmetric RBG keys	Determined by design	-
Symmetric Master Key	About 1 year	-
Private Key Transport Key	<= 2 years (1)	
Public Key Transport Key	1-2 years	
Symmetric Key Agreement Key	1-2 years (2)	
Private Static Key Agreement Key	1-2 years (3)	
Public Static Key Agreement Key	1-2 years	
Private Ephemeral Key Agreement Key	One key agreement transaction	
Public Ephemeral Key Agreement Key	One key agreement transaction	
Symmetric Authorization Key	<= 2 years	
Private Authorization Key	<= 2 years	
Public Authorization Key	<= 2 years	

Upravljanje ključevima

- u praksi najranjiviji deo sistema `login:root password:root`
- $100M za brute-force napad ili $100K za podmićivanje?
- loša implementacija
  - DiskLock for Mac - DES algoritam za šifrovanje fajlova, ali se ključ smešta u fajl

Upravljanje ključevima

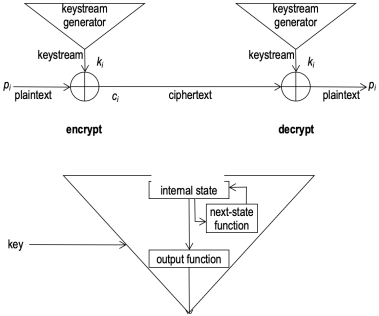
- generisanje ključeva
  - redukovan prostor ključeva
  - upotreba samo ASCII karaktera za ključeve
- loš izbor ključeva
  - ime drage osobe, kućnog ljubimca, broj lične karte
  - dictionary attack
- bolji izbor ključeva
  - duže fraze kao lozinke
  - izračunavanje hash vrednosti za duži string koji sadrži neku lako pamtivu frazu
  - izbegavati suviše poznate fraze

Upravljanje ključevima - o čemu još treba razmišljati?

- distribucija ključeva
- skladištenje ključeva
- backup
- “rok trajanja” ključa

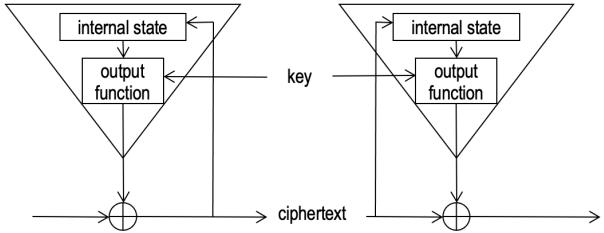
Tipovi algoritama

- block cipher
  - operišu nad blokovima otvorenog i šifriranog teksta
  - blok je tipično 64 bita
- stream cipher
  - operišu nad tokom otvorenog/šifriranog teksta po 1 bit/bajt/reč istovremeno
  - keystream generator



Tipovi algoritama

- stream cipher
  - self-synchronizing stream cipher
- next-state function = ciphertext

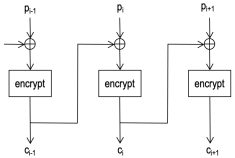


Tipovi algoritama

- block vs stream
  - block algoritmi se mogu implementirati kao stream i obrnuto
  - “block šifre operišu nad podacima sa fiksnom transformacijom nad blokovima otvorenog teksta; stream šifre operišu sa transformacijama nad pojedinim ciframa otvorenog teksta koje se menjaju tokom vremena”

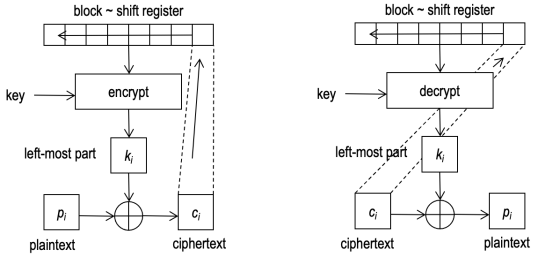
Režimi rada

- electronic codebook (ECB)
  - jedan blok otvorenog teksta se uvek šifruje nezavisno istim ključem
  - moguće je napraviti rečnik (codebook) sa svim kombinacijama otvorenih i šifriranih blokova
  - podložno aktivnim napadima (Mallory)
- cipher block chaining (CBC)
  - otvoreni tekst narednog bloka se XORuje sa šifriranim tekstom prethodnog bloka
  - prvi blok se XORuje sa slučajnim inicijalizacionim blokom (“vektorom”)
  - dešifrovanje analogno šifrovanju – nakon dešifrovanja bloka on se XORuje sa prethodno dešifrovanim blokom



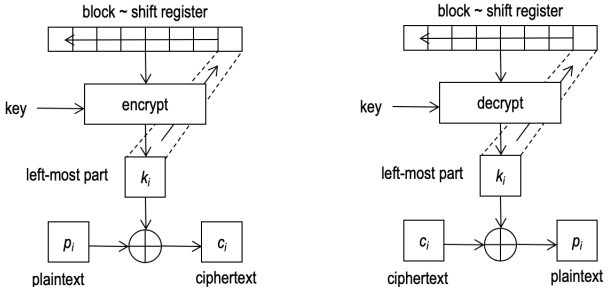
Režimi rada

- cipher feedback (CFB)
  - blok šifra implementirana kao samosinhronizujuća stream šifra
  - blok se tretira kao red (shift registar), inicijalno popunjen slučajnim sadržajem
  - blok se šifrira, i početak reda (n bita) se XORuje sa ulaznim podatkom iz otvorenog teksta (n bita) – tako se dobija izlazni podatak, koji se upisuje na kraj blok registra



Režimi rada

- output feedback (OFB)
  - slično kao CFB, s tim što se u blok registar dodaje element iz rezultata šifrovanja (pre XORovanja)



Režimi rada

- counter (CTR)
  - svaki blok otvorenog teksta se XORuje sa brojačem koji mora biti različit za svaki blok koji se šifrira
  - brojač se inkrementira za svaki naredni blok
  - vrednost brojača se šifrira i tako XORuje

Režimi rada

- rezultat šifrovanja različitim režimom rada