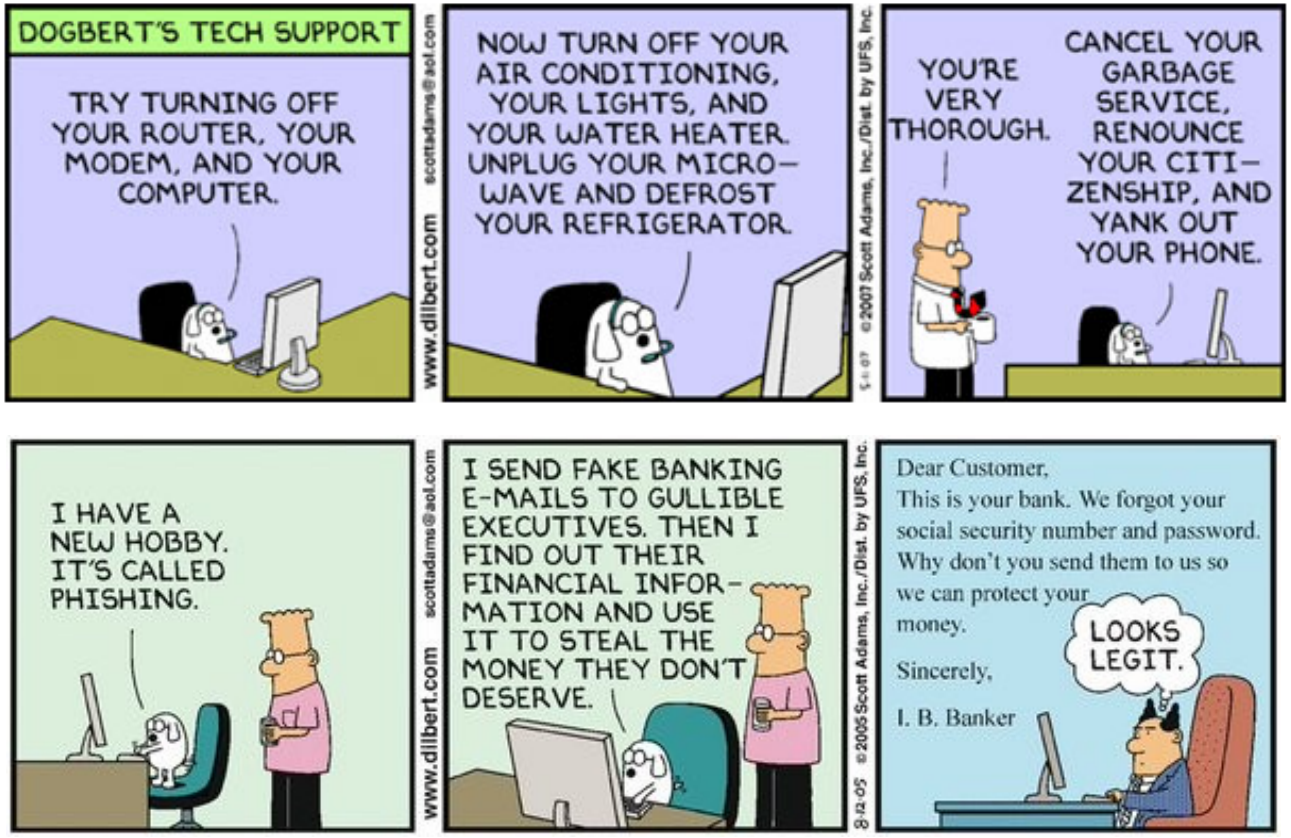
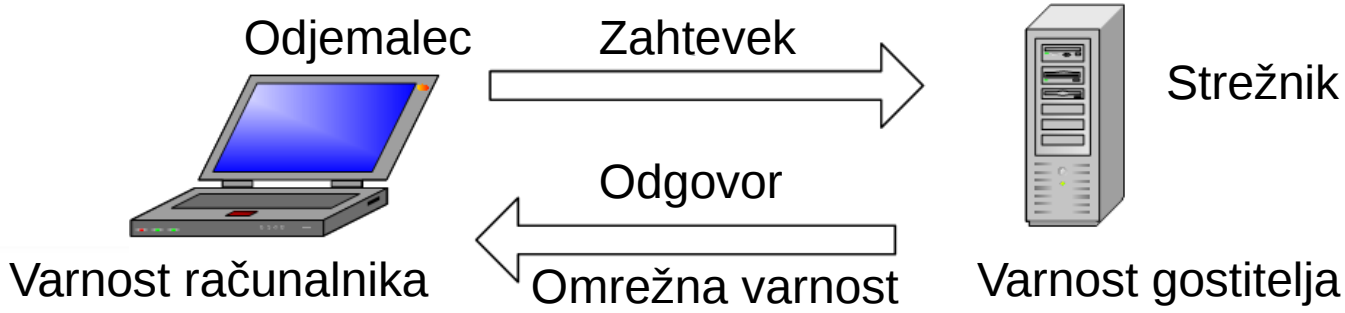
Varnost

Varnost je zelooooo pomembna



Zakaj varno?

- spletne aplikacije → povezava s svetom



- več potencialnih kritičnih točki:
  - zavarovati odjemalca in podatke uporabnika na njem
  - zavarovati podatke ob prenosu
  - varovanje strežnika in podatkov na njem

Vidiki varnosti

- za varnost je potrebno poskrbeti na strani odjemalca, torej da so podatki tam varni pred naključnimi dostopi → aktivna in poučena vloga uporabnika
- izjemno pomemben vidik je varnost podatkov na strani strežnika
  - varnost na strežniku se tiče vseh uporabnikov, ki so kdajkoli poslovali s to aplikacijo/storitvijo
  - vdor na odjemalca potencialno prizadene do nekaj uporabnikov
  - vdor na strežnik lahko potencialno ogrozi na sto tisoče uporabnikov

Vidiki varnosti

- izjemno pomembna je tudi varnost komunikacije
- pri komunikacijah pa imamo opravka z več vidiki:
  - zasebnost/zaupnost (Privacy/Confidentiality) → podatki, ki si jih dve strani izmenjujeta, ne smejo biti »ukradeni« med izmenjevanjem
  - integriteta (Integrity) → podatkov, ki se izmenjujejo, nima nihče možnost spreminjati
  - avtentikacija (Authentication) → mogoče mora biti tako za odjemalca kot tudi za strežnik, da se prepriča o identiteti nasprotne strani
  - neovrgljivost (Nonrepudiation) → možno mora biti legalno potrditi da je bilo sporočilo dejansko poslano in tudi sprejeto

Vidiki varnosti

- pri zagotavljanju varnosti pa moramo upoštevati/zagotavljati tudi:
  - avtorizacijo (Authorisation) → omogoča izvajanje pravic uporabnikom
    - lista pravic za uporabnika
    - kontrola dostopa za vloge
  - razpoložljivost (Availability) → zagotavljanje dostopnosti spletnih aplikacij
    - pomembno z ekonomskega vidika za ponudnika
    - enostavno/priučeno delo za uporabnika

Šifriranje

- šifriranje je že zelo stara tehnika zagotavljanja varnosti komunikacije
  - šifriranja so uporabljali že v antiki
  - je osnovna tehnologija varne komunikacije
  - različni algoritmi omogočajo pretvorbo sporočila v obliko, ki je nerazumljiva za naključnega opazovalca
- šifrirano sporočilo se pretvori v razumljivo obliko z obratnim postopkom, ki se mu reče dešifriranje
- šifriranje je eden temeljev varne komunikacije

Kriptografski algoritmi

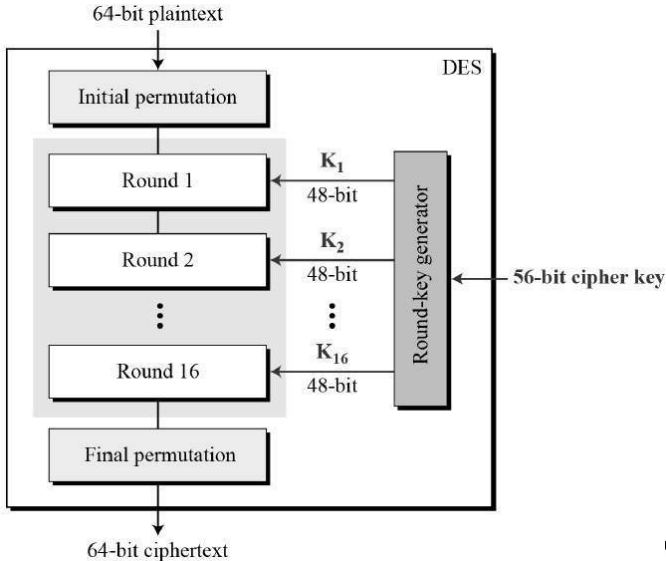
- pri tem gre za postopek kodiranja sporočila z uporabo ključev
  - simetrični algoritmi (tudi simetrični ključ) uporabljajo isti ključ za šifriranje in dešifriranje
  - asimetrični algoritmi (tudi javni ključ) uporabljajo različne ključe za šifriranje in dešifriranje
- pri vsakem pristopu kodiranja obstajata dva izziva:
  - distribucija ključev → kako dostaviti ključ za vzpostavitev varne komunikacije
  - upravljanje ključev → kako pri velikem številu ključev ohraniti varnost in dostopnost, ko jih rabimo

Simetrični algoritmi

- uporablja se ključ za šifriranje
  - ko se izvede akcija (npr. naročilo), se sporočilo s šifrirnim ključem šifrira
  - šifrirano sporočilo se prenese (lahko se mu tudi prisluškuje)
  - na strani prejemnika se sporočilo z uporabo dešifrirnega ključa dešifrira, dešifrirni ključ je enak šifrirnemu
- potrebna je varna izmenjava ključa pred izvedbo prenosa
- tipični predstavniki: DES, AES

DES in AES

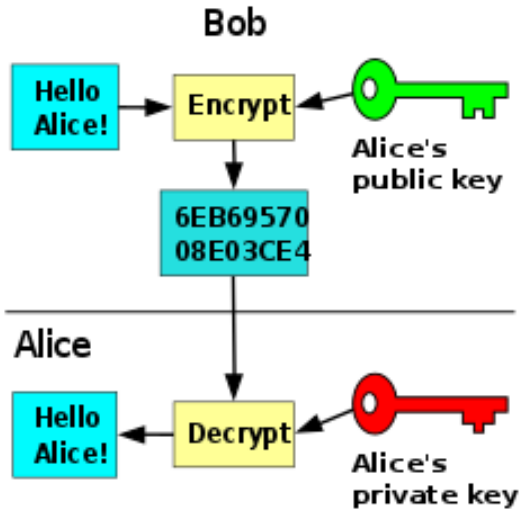
- Data Encryption Standard (DES) uporablja 64 bitni ključ za šifriranje
- 56 bitov je uporabljenih za ključ, medtem ko je 8 bitov (za vsakimi sedmimi biti je en paritetni bit) namenjenih za preverjanje paritete
- algoritem DES izvaja šifriranje 64 bitnega bloka:
  - izvede se začetna permutacija 64 bitov
  - permutacija služi kot vhod za računanje, ki ga določa ključ
  - ponovno se izvede permutacija, ki pa je obratna od začetne permutacije



DES in AES

- za dešifriranje se uporabi obraten postopek
- pri 64 bitnem ključu je 56 bitov namenjenih šifriranju, vseh ključev je 2^56
- že leta 1999 so uspeli razbiti ključ v 22 urah in 15 minutah
- 3DES (uradno Triple Data Encryption Algorithm, TDEA) uporablja tri ključe, skupaj torej 168 bitov → 1. ključ za šifriranje besedila; 2. ključ za dešifriranje šifriranega besedila; in 3. ključ za šifriranje dešifriranega besedila
  - šifrirano besedilo = Dk3(Ek2(Dk1(besedilo)))
  - 2TDEA: prvi in tretji ključ enaka → k1 = k3
  - 3TDEA: vsi ključi so neodvisni
  - vsi ključi so enaki: nazaj kompatibilno z DES, ISO ne dovoljuje te možnosti
  - besedilo = Ek1(Dk2(Ek3(šifrirano besedilo)))
- Advanced Encryption Standard (AES): zamenjava za DES → 128,192, 256 bitni ključi

Asimetrični algoritmi



- asimetrična kriptografija je bolj varna kot simetrična
  - obstaja javni ključ, ki se uporabi za šifriranje sporočila
  - za dešifriranje sporočila se uporabi privatni ključ
  - odpravlja problem distribucije ključev
- različni asimetrični algoritmi so na voljo za različne namene:
  - Digital Signature Standard
  - ElGamal
  - RSA
  - ...

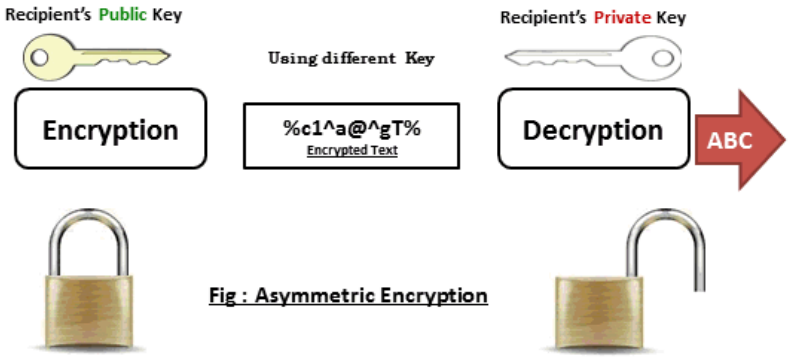


Fig : Asymmetric Encryption

RSA algoritem

- Rivest-Shamir-Adleman (RSA) algoritem je eden izmed prvih asimetričnih sistemov z javnim ključem
- asimetrija bazira na praktični težavnosti faktorizacije zmnožka dveh velikih števil
- prvič je bil objavljen že leta 1977
- uporablja se za varno prenašanje podatkov
- ekvivalenten sistem je bil razvit že leta 1973 za potrebe Britanske obveščevalne agencije a je bil predstavljen javnosti šele leta 1997
- RSA-768, ki vsebuje 232 decimalnih števil (768 bitov) je bil razbit leta 2009 v obdobju dveh let
- ostali ključi/algoritmi še niso bili faktorizirani/razbiti, navkljub nagradi

RSA algoritem

- RSA algoritem vključuje naslednje korake
  - generiranje ključev
  - distribucija ključev
  - šifriranje
  - dešifriranje
- osnovni princip je ta, da je mogoče najti tri velika števila *e*, *d* in *n* tako, da velja $(m^d)^e \equiv m \quad (mod\, n)$, pri tem pa je kljub poznavanju *e* in *n* ali celo *m* zelo težko najti *d*

Generiranje ključev

- Ključ za RSA se zgeneriran na sledeč način:
  - izberi dve različni praštevili *p* in *q*
  - izračunaj *n=pq* → *n* je modulo za javni in privatni ključ (njegova dolžina v bitih je dolžina ključa)
  - izračunaj $\lambda(n)=lcm(p-1,q-1)$, Carmichel-ovo totientno funkcijo
  - izberi celo število *e*, med 1 in $\lambda(n)$ tako, da je $gcd(\lambda(n),e)=1$
  - določi *d* kot $d\equiv e^{-1}(mod\,\lambda(n))$

Distribucija, šifriranje, dešifriranje

- oseba, ki želi prejeti sporočilo, mora poslati svoj javni ključ (*n*, *e*) pošiljatelju preko zanesljive povezave, za katero ni nujno, da je varna
- privatni ključ se ne distribuira!!!
- za kriptiranje sporočila mora pošiljatelj pretvoriti M (sporočilo) v celo število *m* tako, da je $0 \leqslant m < n$ z uporabo protokola imenovanega »padding scheme«
- nato izračuna šifrirano besedilo *c* z uporabo javnega ključa, ki ga nato pošlje prejemniku: $c \equiv (m^e) \pmod n$
- prejemnik lahko dešifrira prejeto sporočilo z uporabo privatnega ključa, (*n*, *d*): $c^d \equiv (m^e)^d \equiv m \pmod n$
- sporočilo M pa dobi z reverzom »padding scheme« nad *m*

Primer

- *p=47, q=59*
- *n=p*q=47*59=2773*
- izračunamo Carmichaelovo totientno funkcijo: *lcm (p-1,q-1) = 1334*
- izberemo število med 1 in 1334, ki je koprašteviloλ(n), na primer: *e=17*
- izračunamo *d* kot modulo multiplikativni inverz: *d=157*
- javni ključ → (*n=2773*, *e=17*)
- privatni ključ → (*n=2773 d=157*)

Primer

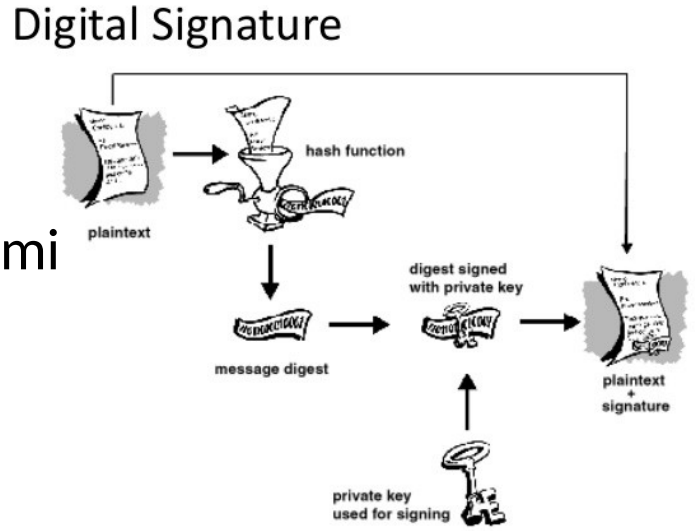
- *p=47*, *q=59*, javni (*n=2773*, *e=17*), privatni (*n=2773 d=157*)
- Besedilo → spletne tehnologije
- Kodiranje: pretvorimo v zaporedje številk, izberemo dogovorjeno kodiranje, npr. ' '=0, 'a'=1, 'b'=2.... →

  zakodirano→ 1917 1306 2115 0600 2106 0915 1613 1608 1011 0600

  s p l e t n e '' t e h n o l o g i j e ''
- šifriranje:
  šifrirano → 1264 0183 1645 1424 0654 0624 0264 1274 2602 1424
- Dešifriranje:
  dešifrirano: 1917 1306 2115 0600 2106 0915 1613 1608 1011 0600
  besedilo: s p l e t n e '' t e h n o l o g i j e ''

Digitalni podpis

- matematična shema za dokazovanje avtentičnosti digitalnega sporočila ali dokumenta, ki omogoča
  - preprečevanje spremembe sporočila pri prenosu
  - omogoča neovrgljivost
  - omogoča dokazovanje istovetnosti
- uporablja se lahko
  - razpršitvene algoritme – računanje z majhnimi količinami podatkov
    - md5 algoritem
    - sha1 algoritem
  - asimetrično šifriranje (RSA)



Certifikati in javni ključi

- certifikati omogočajo identifikacijo osebe
- certifikat vsebuje
  - javni ključ
  - informacijo o certifikatu → lastnik, veljavnost
  - digitalni podpis
- identifikacijo zagotavlja overitelj (CA-Certification Authority, SIGEN-CA, POŠTArCA, ...)
- identifikacijo preverja agencija za registracijo (RA-Registration Authority)
- certifikati imajo svojo veljavnost, po izteku veljavnosti so neuporabni

Certifikati in javni ključi

- X.509 je standard, ki definira format certifikatov z javnimi ključi
- certifikati X.509 se uporabljajo v številnih internetnih protokolih, tudi TLS/SSL, ki je temelj za HTTPS
- X.509 certifikati vsebuje poleg ostalih podatkov tudi javni ključ in je podpisan, bodisi s strani overitelja ali pa je samo-podpisan
- če je certifikat podpisan s strani overitelja, se lahko javni ključ uporabi
  - za vzpostavitev varne povezave
  - za povezovanje javnega ključa z identiteto lastnika privatnega ključa
  - za validacijo dokumentov, ki so bili digitalno podpisani z ustreznim privatnim ključem

Komunikacija odjemalec-strežnik

- komunikacija dveh točk (point-to-point) - P2PE
  - uporablja pri plačevanju
  - pri tem se zaupni podatki s kartice šifrirajo v trenutku, ko je kartica prebrana za to, da se prepreči prevara/hekiranje
  - celotna rešitev, vključno s strojno in programsko opremo
- komunikacija dveh končnih točk (end-to-end) - E2Ee
  - podobno kot P2Pe vendar niso izpolnjeni vse zahteve P2Pe
  - ponavadi je povezava med točkama varna, niso pa izpolnjeni pogoji glede šifriranja v trenutku, ko se podatki s kartice preberejo

Varnost: vidik odjemalca

- izmenjava osebnih podatkov med odjemalcem in strežnikom
- potrebno je vzpostaviti zaupanje v ponudnika storitve in rokovanje s prenesenimi podatki
  - ali se podatki shranjujejo in če se, kako varno so shranjeni
  - za katere namene bodo podatki uporabljeni
  - ali bo ponudnik posredoval podatke naprej
- pomembno je ohraniti zasebnost
  - »Platform for Privacy Preferences« → W3C standard za varovanje podatkov v XML,
    - kateri podatki se zbirajo
    - zakaj se podatki zbirajo
  - P3P agenti za preverjanje spletne strani za potencialne konflikte v zasebnosti

Varnost: vidik odjemalca

- varnost kode, ki se izvaja v brskalniku: JavaScript, apleti, razni obrazci
- lažno predstavljanje (Spoofing) in ribarjenje
  - 23. 1 2015 SI-CERT prejme prijave o sporočilih, ki uporabnike slovenskih bank prepričujejo, naj na lažne naslove vpišejo svoje podatke; podoben dogodek februarja 2017
  - 24. 6. 2008 se je (ponovno) pojavila ponarejene spletna stran nlb, ki je bila originalni zelo podobna, namenjena pa je bila kraji uporabnikovih podatkov (»DNS intrusion«)
- varnost računalnika/telefona/….
  - adware, spyware
  - virusi, črvi
  - trojanci

Varnost: vidik ponudnika

- napad na strežnik oziroma spletno aplikacijo
  - potrebno implementirati varno storitev
  - upoštevati in odstraniti tipične varnostne pomanjkljivosti
- prekdomenski napadi (cross-site scripting)
- injekcije SQL
- napad na dostopnost
  - DOS in DDOS napadi, preobremenitev gostitelja
  - sesutje spletne aplikacije zaradi sesutja aplikacije (buffer overrun)
- pomembnost varovanja gostitelja
  - posodabljanje
  - konstantno pregledovanje za varnostne pomanjkljivosti

Varnost: vidik ponudnika

- navodila kako zavarovati spletne aplikacije
  - https://msdn.microsoft.com/en-us/library/ff648636.aspx
  - http://www.it.northwestern.edu/policies/webapps.html
  - OWASP Guide Project: https://www.owasp.org/index.php/OWASP_Guide_Project
- vidiki varnosti
  - sprejem parametrov in dinamično generiranje strani
  - preveč zahtevkov ali »napačni« zahtevki
  - novo odkrite varnostne luknje v sistemu ali knjižnicah
  - varnostne luknje v sami aplikaciji