Univerza v Ljubljani
Fakulteta za elektrotehniko

TEORIJA KRIPTOLOGIJE

Seminar pri predmetu Porazdeljeni informacijski sistemi in celovitost podatkov

Borut Pogačnik

Kranj 2006

Kazalo

1 Pregled kriptografije

1.1 Uvod

Kriptologija je veda o tajnosti, šifriranju, zakrivanju sporočil (kriptografija) in razkrivanju šifriranih podatkov (kriptoanaliza). Kriptologija je študij kriptografije in kriptoanalize. Beseda kriptologija izzvira iz grščine: kryptos logos, kar pomeni skrita beseda.
Kriptologijo je potrebno obravnavati širše, in sicer kot vedo o varnih sporočilih in informacijah. Tajnost je namreč le del celovitosti informacij in besedilo je le del možnih sporočil (poleg slik, zvoka, animacije…).
Kriptografija je veda o matematičnih tehnikah za dosego informacijske varnosti, kot je zaupnost, celovitost podatkov, overjanje identitete in podatkov.
Kriptografija se ukvarja s študijem in razvojem metod in metodologij za šifriranje, kjer se običajno uporabljajo skrivni ključi, s katerimi je mogoče dešifrirati šifrirano sporočilo ali informacijo.
Kriptografija ne pomeni samo zagotavljanje informacijske varnosti, ampak tudi določene matematične tehnike za doseganje le-te. Kriptografija je nastala kot posledica komunikacije v prisotnosti nasprotnikov oz. tekmecev. Uporablja se za preprečevanje in odkrivanje zlorab in ostalih zlonamernih dejanj.
Kriptoanaliza je študij o matematičnih tehnikah za premagovanje kriptografskih sredstev in informacijsko varnostnih storitev.
Kriptoanaliza išče oziroma razvija tehnike za dešifriranje šifriranih sporočil, brez apriornega poznavanja ključa.

Celovitost informacije je lastnost, ki zagotavlja, da je informacija resnična, pravočasna, nespremenjena in ima tudi vse tiste lastnosti pri hranjenju in prenašanju, ki ohranjajo informacijsko vrednost, na primer tajnost, verodostojnost, istovetnost avtorja, preverjenost, podpisanost, časovno opredeljenost, imetje certifikata in podobno. Celovitost informacije pomeni, da je bila vsakršna napaka ali goljufanje preprečeno ali vsaj odkrito. Celovitost informacije je sestavljena kot mozaik iz veliko delov, za zagotavljanje številnih od njih pa obstaja kriptografska rešitev in se zato z njimi ukvarja sodobna kriptologija.

Zgodovina kriptografije
Začetki kriptologije segajo že v rimsko dobo, saj je znano, da je že Julij Cezar uporabljal šifrirana sporočila (Cezarjeva šifra). Julij Cezar je svojim vojskovodjem pošiljal sporočila tako, da je vsako črko zamenjal s črko, ki je bila v abecedi tri mesta za njo.
Do 20. stol. se je kriptologija razvijala bolj kot umetnost kot znanost in kriptoanalitiki so v glavnem imeli veliko uspeha.
Pomemben kriptografski dogodek je bil leta 1926, ko je ameriški inženir Vernam objavil edinstveno šifro, pri katerem je bila poglavitna nova zamisel, da se naključni ključ uporabi za šifriranje samo enkrat, zato se šifra imenuje enkratna prevleka.
Leta 1949 je Shannon objavil delo Communication Theory of Secrecy Systems, s katerim je postavil temelje kriptološke znanosti. Medtem ko se je Shannovo delo ukvarja s simetrično kriptografijo in zagotavljanjem tajnosti, sta Diffie in Hellman utemeljila asimetrično kriptografijo s številnimi novimi cilji v članku New directions in Cryptography. Soavtor te nove vrste kriptografije je tudi Merkle, vendar je bil njegov prispevek Secure communication over insecure cannels objavljen šele leto pozneje, zato mu pogosto po krivici niso priznavali njegovih zaslug. Diffie in Hellman sta tako prva pokazala, da je mogoč praktično varen sistem tajnega komuniciranja brez varnega prenosa tajnega ključa med pošiljateljem in prejemnikom. S tem sta pretresla kriptološki svet in izzvala pravo raziskovalno eksplozijo. Bistvo njunega

članka sta dve nenavadno zviti definiciji enosmerne funkcije in enosmerne funkcije s stranskimi vratci, ki nista matematično rigorozni.

1.2 Varnost informacij in kriptografija

Elementi varnosti informacij so:

- zasebnost (privacy),
- zaupnost (confidentiality),
- celovitost podatkov (data integrity),
- overjanje osebe (rač. terminala, kreditne kartice) (entity authentication, identification),
- overitev sporočila (message authentication),
- podpis (signature),
- avtorizacija (authorization),
- potrjevanje veljavnosti (validation),
- kontrola dostopa (access control),
- certificiranje (certification),
- prejem (receipt),
- potrditev (confirmation),
- lastništvo (ownership),
- anonimnost (anonymity)
- preprečevanje tajenja (non-repudiation),
- preklic (revocation).

Kriptografija nudi metode, ki omogočajo naslednje cilje:

- *Zasebnost.* Nasprotnik ne izve nič koristnega iz poslanega sporočila.
- *Zaupnost.* Vsebina podatkov je dostopna le tistim, ki so za dostop pooblaščeni.
- *Pristnost*. Sprejemnik sporočila se lahko sam prepriča oziroma preveri, ali je sporočilo res poslal naveden pošiljatelj.
- *Celovitost*. Sprejemnik ima zagotovilo, da prispelo sporočilo nasprotnik ni spremenil.
- *Podpis*. Sprejemnik sporočila lahko prepriča tretjo osebo, da je sprejeto sporočilo celovito in izvira od navedenega pošiljatelja.
- *Preprečevanje nepriznavanja*. To pomeni preprečiti osebi, da bi zanikala predhodno obveznost ali dejanje.
- *Istočasna menjava*. Vredno sporočilo, kot je podpis ali pogodba, ni poslano, dokler ni sprejeto neko drugo vredno sporočilo, npr. podpis druge osebe.
- *Uskladitev*. Pri pogovoru več oseb, so le-te zmožne uskladiti svoje aktivnosti proti skupnemu cilju, četudi v prisotnosti nasprotnika.

Osnovni cilj kriptografije je zadostna podpora tem področjem v teoriji in praksi. V zadnjih dvajsetih letih se je kriptografija razvila v eksaktno vedo. Trenutno obstaja več mednarodnih znanstvenih konferenc (CRYPTO, ASIA-CRYPT, EUROCRYPT) in znanstvena organizacija International for Cryptologic Research (IACR), ki se ukvarjajo izključno z raziskovanjem na področju kriptografije.

1.3 Osnovno znanje o funkcijah

1.3.1 Funkcije (1 – 1, enosmerna funkcija, enosmerna funkcija s stranskimi vrati)

Definicija
Če imamo dve množici X in Y ter predpis f , ki vsakemu elementu množice X priredi natanko en element množice Y. X, Y, f določajo funkcijo z definicijskim območjem X, zalogo vrednosti v Y in s predpisom f.

1 – 1 funkcija (preslikava) je, če vsak element iz zaloge vrednosti Y preslikamo v največ en element v definicijskem območju X.
Preslikava množice X v množico Y je predpis f : x → y, ki vsakemu elementu množice X priredi ustrezni element množice Y.
Preslikane elemente imenujemo praslike (originale), iz njih sestavljeno množico pa originalno množico ali definicijsko območje preslikave (domeno). Elemente, ki so praslikam prirejeni, imenujemo slike, iz njih sestavljeno množico pa zaklad (zalogo) vrednosti preslikave (kodomeno). Vsaka preslikava je tudi funkcija, pri kateri sta domena in kodomena v bistvu količini, ki ju lahko predstavimo z množico števil, predvsem realnih števil.
Vsaka preslikava je enolična. Če pa je preslikava mnogolična, se imenuje relacija.

Preslikava je:
- Surjektivna, če ustreza vsakemu elementu iz Y vsaj en element iz X (za preslikavo f, ki preslika množico X v množico Y pravimo, da je surjektivna, če je vsak element iz množice Y slika vsaj enega elementa iz množice X, pravimo tudi, da f preslika množico A na množico B).
- Injektivna, če ustreza vsakemu elementu iz Y največ en element iz X (za preslikavo f , ki preslika množico X v množico Y, pravimo, da je injektivna, če imata dva različna elementa iz množice X vedno različni sliki v množici Y).
- Bijektivna, če ustreza vsakemu elementu iz Y natančno en element iz X (za preslikavo f, ki preslika množico X v množico Y pravimo, da je bijektivna, če je injektivna, surjektivna in povsod definirana).

Definicija
Če je f bijektivna preslikava iz X v Y, potem enostavno definiramo bijektivno preslikavo g iz Y v X kot : za vsak $y \in Y$ definiramo $g(y) = x$ kjer $x \in X$ in $f(x) = y$. Potem funkcijo g, ki jo dobimo iz f , imenujemo inverzna funkcija f in zapišemo $g = f^{-1}$.

Enosmerne funkcije (one-way function)
Definicija
Funkcijo f , ki množico X preslika v množico Y imenujemo enosmerna funkcija, če je f(x) lahko izračunljiva za vse $x \in X$, toda skoraj nemogoče je izračunati x, ki nam da vrednost $f(x) = y$.

Enosmerna funkcija s stranskimi vratci (trapdoor one-way function)
je družina obrnljivih funkcij, za katere je vedno enostavno najti algoritma za računanje funkcije in njene inverzne funkcije. Toda skoraj vedno je nemogoče izračunati vrednost nasprotne funkcije, če se pozna algoritem za računanje funkcije. To ni natančna matematična definicija, a je dovolj razumljiva za uporabo v asimetrični kriptografiji.

1.3.2 Permutacije

Definicija
Vzemimo množico S, ki ima končno število elementov. Permutacija p na množici S je bijektivna iz S v S p: S → S.

Permutacije ali razvrščanje elementov dane končne množice so pravzaprav bijektivne preslikave te množice same nase; po razporejanju se namreč v nizu pojavi vsak element natanko enkrat, v splošnem na nekem drugem mestu, kot je bil najprej. Zato lahko imenujemo vsako bijektivno preslikavo množice nase permutacija.

Primer
Imamo množico S = { 1, 2, 3, 4, 5} in permutacijo p: S → S, ki je definirana kot
p(1) = 3, p(2) = 5, p(3) = 4, p(4) = 2, p(5) = 1.
Permutacijo lahko predstavimo na različne načine, npr. kot polje:

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix} \qquad p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 1 & 3 & 2 \end{pmatrix}$$

Ker so permutacije dane končne množice $\{a_1, a_2, \ldots a_n\}$ preslikava te množice same nase, jih lahko poljubno komponiramo. Če recimo prva permutacija p_1 preslika element a_i v element $b_i = p_1(a_i)$ in druga permutacija p_2 preslika element b_i v element $c_i = p_2(b_i)$, je kompozitum permutacij tista preslikava, ki prenese a_i neposredno v c_i.

Primer
Za permutaciji

$$p_1 = \begin{pmatrix} a & b & c & d & e \\ d & c & a & e & b \end{pmatrix} \quad \text{in} \quad p_2 = \begin{pmatrix} a & b & c & d & e \\ c & a & d & b & e \end{pmatrix} \quad \text{je njun kompozitum enak}$$

$$p_1 \circ p_2 = \begin{pmatrix} a & b & c & d & e \\ d & c & a & e & b \end{pmatrix} \circ \begin{pmatrix} a & b & c & d & e \\ c & a & d & b & e \end{pmatrix} = \begin{pmatrix} a & b & c & d & e \\ b & d & c & e & a \end{pmatrix}.$$

Ker je vsaka permutacija p bijektivna preslikava dane množice nase, obstaja tudi inverzna preslikava oz. inverzna permutacija p^{-1}.Po definiciji inverzne preslikave dobimo s kompozitumom permutacije p in k njej inverzne permutacije p^{-1} tako permutacijo p_E, ki ohranja naravni vrstni red.

Primer

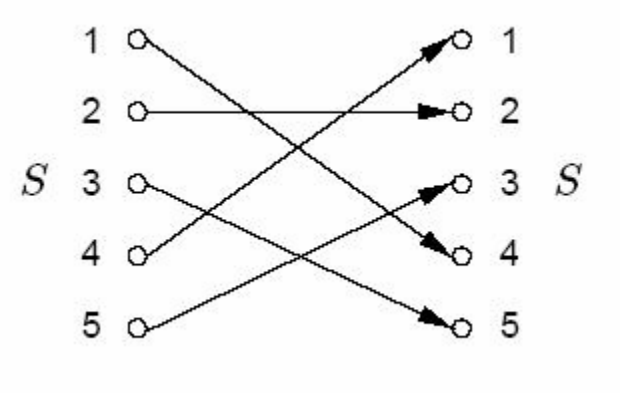
$$p = \begin{pmatrix} a & b & c & d & e \\ d & c & a & e & b \end{pmatrix} \Rightarrow \quad p^{-1} = \begin{pmatrix} a & b & c & d & e \\ c & e & b & a & d \end{pmatrix}$$

$$p \circ p^{-1} = \begin{pmatrix} a & b & c & d & e \\ d & c & a & e & b \end{pmatrix} \circ \begin{pmatrix} a & b & c & d & e \\ c & e & b & a & d \end{pmatrix} = \begin{pmatrix} a & b & c & d & e \\ a & b & c & d & e \end{pmatrix} = p_E .$$

1.3.3 Involucija

Definicija
Involucija je preslikava, ki je sama sebi inverz. Natančneje f, S→ S je involucija, če za vsak $x \in S$ velja f(f(x)) = x.

Primer
Slika 1 prikazuje primer involucije.



Slika 1: Involucija na množici S

Primer
Konjugiranje kompleksnih števil je involucija.
Transponiranje matrik je involucija.
Zrcaljenje preko točke, premice, ali ravnine je involucija.

1.4 Osnovna terminologija in koncepti

A označuje končno množico imenovano osnovna abeceda. Primer: binarna abeceda A= {0,1}, ki je pogosto uporabljena kot osnovna abeceda. Vsaka druga abeceda je lahko opisana z binarno abecedo.

M označuje prostor sporočil. M vsebuje nize, sestavljene iz znakov osnovne abecede. Elemente M imenujemo sporočilo ali čistopis. Primer: M lahko vsebuje binarne nize, navadne besede, računalniško kodo, itd.

C označuje prostor šifriranih sporočil . C vsebuje nize sestavljene iz znakov osnovne abecede, ki se razlikujejo od abecede, definirane za M. Elemente množice C imenujemo šifropis.

K označuje prostor ključev. Element množice K imenujemo ključ.

Vsakemu ključu $e \in K$ priredimo bijekcijo E_e iz prostora sporočil M v prostor šifriranih sporočil C. Funkcijo E_e imenujemo šifrirna funkcija (ali transformacija). Proces uporabe šifrirne funkcije E_e na sporočilu $m \in M$ imenujemo šifriranje ali enkripcija.

Vsakemu ključu $d \in K$ priredimo bijekcijo D_d iz prostora šifriranih sporočil C v prostor sporočil M. D_d imenujemo dešifrirana funkcija (ali transformacija).Proces uporabe dešifrirne funkcije D_d na sporočilu $c \in C$ imenujemo dešifriranje ali dekripcija.

Enkripcijska shema vsebuje množico $\{E_e: e \in K\}$kot enkripcijsko transformacijo in ustrezno množico $\{D_d: d \in K\}$kot dekripcijsko transformacijo, z lastnostjo da za vsak $e \in K$ obstaja unikaten ključ $d \in K$, tako da velja $D_d = E_e^{-1}$ oz. $D_d(E_e(m)) = m$ za vse $m \in M$.

Ključa e in d, definirana zgoraj, imenujemo par, to označujemo (e,d).

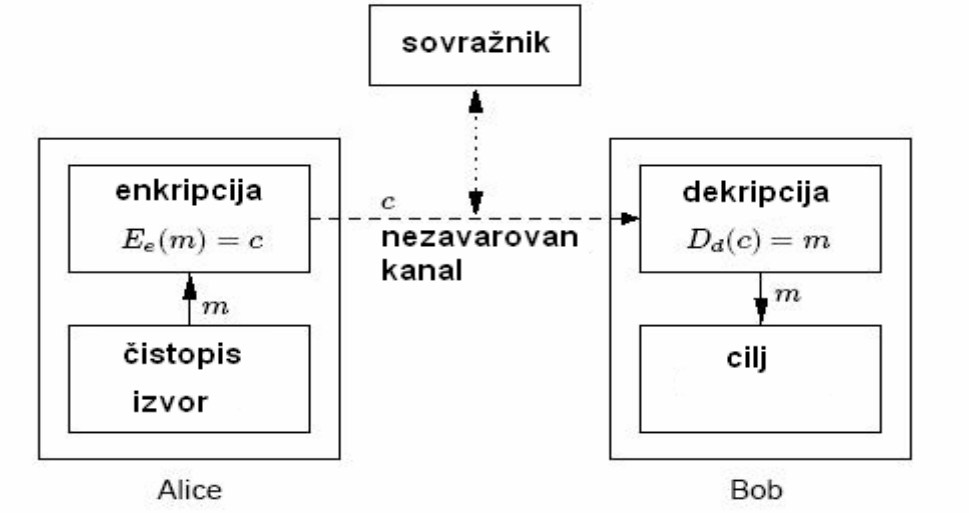
Sestavljanje enkripcijske sheme zahteva izbran prostor sporočil M, prostor šifropisov C, prostor ključev, množico enkripcijske transformacije $\{E_e: e \in K\}$ in ustrezno množico dekripcijske transformacije $\{D_d: d \in :K\}$.

Kriptosistem
Kriptosistem je v splošnem pojem, ki se nanaša na niz kriptofgrafskih temeljev za doseganje kriptografskih ciljev v informacijsko-varnostnih storitvah. Največkrat je pojem vezan na temelje v povezavi z zasebnostjo, v splošnem, na šifriranje. Konkretno implementacijo nekega kriptosistema za šifriranje imenujemo šifrirna shema. Oglejmo si definicijo kriptosistema.

Kriptosistem je peterica (M, C, K, E, D), za katero velja:

1. M je končna množica možnih sporočil,
2. C je končna množica možnih šifriranih sporočil,
3. K je končna množica ključev,
4. za vsak ključ $k \in K$ imamo šifrirno transformacijo $E_k \in E$ in ustrezno dešifrirno transformacijo $D_k \in D$. $E_k : M \rightarrow C$ in $D_k: C \rightarrow M$ sta taki funkciji, da je $D_k(E_k(x))=x$ za vsak $x \in M$.

Najpomembnejša je lastnost 4, ki nam pove, da če čistopis m zašifriramo s pomočjo E_k in tako dobljeni šifropis dešifriramo z D_k , dobimo nazaj originalni čistopis m.



m – čistopis, c – šifropis
Slika 2: Shema komuniciranja z uporabo enkripcije

Komunikacijski udeleženci:

- osebek (nekdo ali nekaj, ki pošilja, prejema ali manipulira z informacijo- osebek je lahko oseba, računalniški terminal…)
- pošiljatelj je osebek, ki legitimno pošilja informacijo (Alice)
- prejemnik je osebek, ki namerava sprejeti informacijo (Bob)
- sovražnik je osebek, ki z uporabo nedovoljenih metod skuša priti do varovanih informacij.

Vrste kanalov:

- fizično varni kanal je tisti, ki ni fizično dostopen sovražniku,
- nezavarovan kanal osebek(sovražnik) lahko informacijo na nezavarovanem kanalu prebere, zbriše, posname …,
- varni kanal zagotavljamo s kriptografskimi tehnikami.

Alice in Bob bosta uporabljala naslednji protokol za uporabo specifičnega kriptosistema. Najprej si izbereta naključni ključ $k \in K$. To naredita takrat, ko sta skupaj in ju sovražnik ne opazuje, ali ko imata dostop do zaščitenega kanala (v tem primeru ni nujno da sta skupaj). Recimo, da želi Alice kasneje Bobu poslati sporočilo preko nezaščitenega kanala. Predpostavimo, da je to sporočilo niz $m = m_1m_2 \ldots m_3$ za neko naravno število $n \geq 1$ in za $m_i \in M$, $1 \leq i \leq n$. Vsak m_i je zašifriran s pomočjo šifrirnega postopka E_k, ki ga določa vnaprej izbran ključ k. Torej Alice izračuna $c_i = E_k(m_i)$, $1 \leq i \leq n$ in dobljeni šifropis $c = c_1c_2 \ldots c_n$ pošlje preko kanala. Ko Bob sprejme $c_1c_2 \ldots c_n$, ga dešifrira s pomočjo dešifrirnega postopka D_k in na ta način dobi čistopis $m_1m_2 \ldots m_n$. Vsaka šifrirna funkcija E_k mora biti injektivna, sicer tajnopisa ne bi mogli enolično dešifrirati. Na primer, če $c = E_k(m_1) = E_k(m_2)$ kjer je $m_1 \neq m_2$ potem Bob ne more vedeti, ali naj c dešifrira kot m_1 ali kot m_2. Opazi, da je vsaka šifrirna funkcija permutacija če M = C. Torej, če sta množici čistopisov in tajnopisov enaki, potem vsaka šifrirna funkcija zgolj premeša (permutira) elemente te množice.

Osnovna lastnost v kriptografiji je, da so množice M, C, K, $\{E_e: e \in K\}$ in $\{D_d: d \in K\}$ javno znane. Varno komuniciranje dveh oseb z izbranim kriptosistemom predpostavlja varovanje uporabljenega ključa, ki ga morata ti dve osebi tudi izbrati. Dodatno varnost lahko pridobimo z varovanjem nekega razreda šifrirnih/dešifrirnih transformacij, vendar graditi varnost celotnega sistema na tej osnovi ni priporočljiva. Zgodovina je že pokazala, da je vzdrževanje varnosti šifrirnih/dešifrirnih transformacij zelo težka naloga.
Pravimo, da je kriptosistem zlomljiv, če lahko tretja oseba, brez prejšnjega poznavanja ključa, sistematično pridobi sporočilo iz šifriranega sporočila v nekem primerno določenem časovnem okviru.
Primerno določen časovni okvir je funkcija uporabnega časa, v katerem morajo biti podatki zavarovani. Npr. naročilo za prodajo neke delnice je lahko varovana skrivnost le nekaj minut, državna skrivnost pa se lahko varuje tudi več desetletij.

Kriptosistemi se delijo na dva osnovna tipa:

Simetrični kriptosistem. Standardna kriptografska rešitev za problem zasebnosti je kriptositem s simetričnim ključem. Ključ je en sam in skrbno varovana skrivnost. Govorimo o tajnem (ali skrivnem) ključu. Šifrirna in dešifrirna transformacija E_k in D_k pri danem $k \in K$ sta bodisi enaki, bodisi da iz ene transformacije na enostaven način dobimo drugo. Primer simetričnega kriptosistema je DES (Data Encryption Standard).

Asimetrični kriptosistem je kriptosistem, kjer je del ključa javno znan, del pa je zaseben. Ideja je ta, da je v praksi računsko neizvedljivo poiskati ustrezno dešifrirno transformacijo D_k iz dane šifrirne transformacije E_k. Če je to tako, potem lahko objavimo šifrirno transformacijo E_e, kjer je e javni del ključa k (ali javni ključ). Prednost javnega kriptosistema pred simetričnim je v tem, da lahko pošiljamo šifrirana sporočila, ne da bi se prej zmenili za skrivni ključ. Oseba, kateri pošljemo sporočilo, šifrirano z njenim javnim ključem e, je po zgornjem edina, ki lahko to sporočilo dešifrira z zasebnim (ali privatnim) ključem d. Primer javnega kriptosistema je RSA kriptosistem.

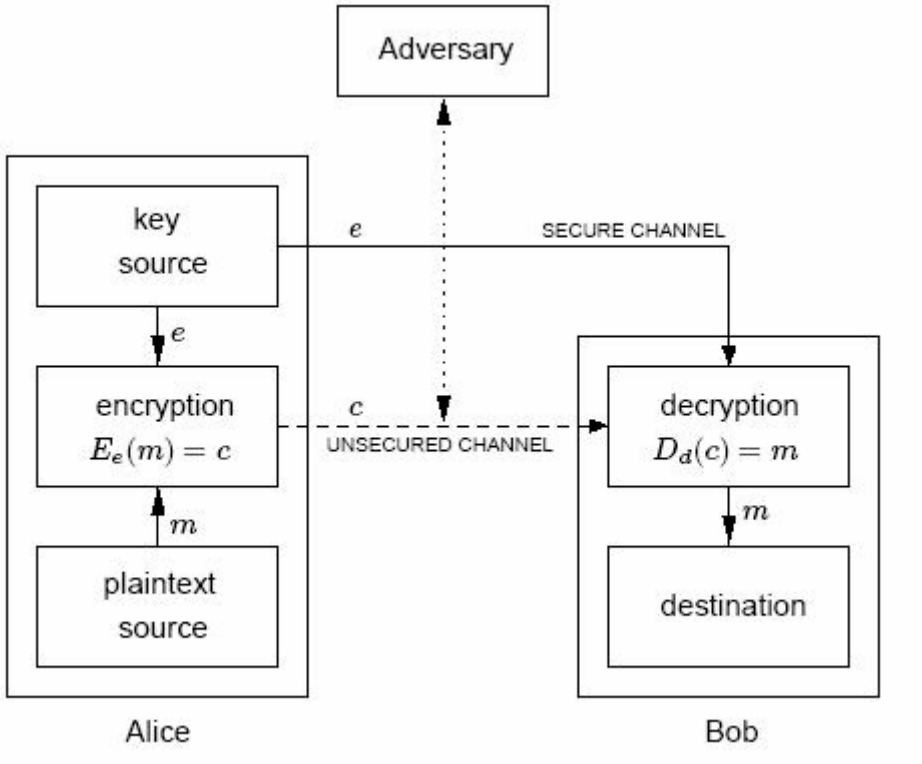
Zakaj so ključi sploh potrebni ? Če imamo transformacijo, parametrizirano s ključi, potem nam v primeru, da je določena šifrirna/dešifrirna transformacija odkrita, ne bo potrebno zgraditi nove šifrirne sheme, pač pa bomo zamenjali le ključ. V praksi je dobro pogosto menjavati ključe (saj se s tem spreminjajo tudi šifrirne/dešifrirne transformacije).

1.5 Simetrična kriptografija

Lastnosti dobrega šifriranja so:

- zasebnost ne sloni na tajnosti šifrirnega postopka, ampak na tajnosti ključa za dešifriranje,
- postopek šifriranja mora biti izvedljiv na računalniku v realnem času,
- postopek dešifriranja mora biti izvedljiv na računalniku v realnem času za tistega, ki pozna dešifrirni ključ,
- postopek dešifriranja ne sme biti izvedljiv v realnem času za napadalca, ki ne pozna ključa, čeprav razpolaga z zelo zmogljivim računalnikom.

Simetrični kriptosistemi pripadajo področju simetrične kriptografije ali kriptografije s tajnim ključem (secret key cryptography). Simetrična kriptografija se pogosto imenuje še klasična kriptografija ali pa kriptografija z enim ključem.
Za enkripcijo in dekripcijo se pri simetričnih kriptosistemih uporabljajo isti ključi in isti algoritmi. Razlika med enkripcijo in dekripcijo je samo v tem da se za dekriptiranje podatkov postopek kriptiranja izvaja v obratnem vrstnem redu. Slika 2 prikazuje uporabo simetrične kriptografije.



Slika 3: Simetrična kriptografija

1.5.1 Bločno in pretočno šifriranje

Bločno šifriranje je enkripcijska shema, kjer sporočilo razdelimo na krajše bloke konstantne dolžine, ki jih nato drugega za drugim šifriramo.
Blokovno šifriranje se naprej deli na *substitucijsko* in *transpozicijsko* šifriranje.
Pri transpozicijskem šifriranju so simboli ohranjeni, njihova sekvenca pa se spremeni (določa jo ključ).
Pri substitucijskem šifriranju pa se sekvenca ne spremeni , spremenijo pa se simboli.

Transpozicijske šifre – primer
V šifriranem tekstu je vrstni red simbolov v bloku tak, kot ga določa ključ, v tem primeru (3 2 5 1 4)

Čistopis:	Včeraj so ujeli medveda.
Delitev v bloke:	včera jsouj elime dveda
Šifropis:	ečavr osjju ileem evadd

Tako šifriranje si lahko predstavljamo kot transpozicija v okviru posameznih blokov oz. transpozicija stolpcev, če si bloke zapišemo enega pod drugim:

Zapis blokov v stolpce:	Uporaba ključa (3 2 5 1 4):
včera	ečavr
jsou	osjju
elime	ileem
dveda	evadd

Sedaj je mogoče opisati transpozicijsko šifriranje bolj splošno. Pri dolžini bloka t je celotno število ključev enako t! oziroma t!-1, ker en ključ (1 2 3 4 5) ponovi čistopis. V praksi je problem hitro rešen, če poznamo dolžino bloka, še posebej z uporabo računalnika. Težava nastopi pri obsežnih sporočilih in ekstremno dolgih blokih.
Problemi s transpozicijskimi šiframi so dvojne narave: prva je, kako ugotoviti dolžino bloka. Tukaj se uporablja izraz: L = n T, kjer je L dolžina sporočila, vendar je treba paziti še na eventuelne prazne simbole. Drug problem pa je nato poiskati ključ, ne da bi preizkusili vseh kombinacij. Če je tekst v naravnem jeziku, potem je mogoče uporabiti znanje o jeziku, npr. frekvenco posameznih črk ali parov črk. Z izenačenjem simbolov določene frekvence iz šifriranega teksta s črkami abecede je mogoče poiskati povezavo med črkami osnovnega in šifriranega teksta.

Substitucijske šifre -primeri
Tukaj so simboli v čistem tekstu zamenjani z drugimi simboli. V splošnem opisujemo simbole čistega teksta z abecedo A = (a_1, … a_{25}) in simbole šifriranega teksta z abecedo B = (b_1, ..b_{25}). Šifriranje sedaj omogoča povezava med simboli obeh abeced, npr. :

Čistopis:	a_3, a_{23}, a_9, a_{17}, a_4
Šifropis:	b_3, b_{23}, b_9, b_{17}, b_4

Primer 1
Ena najbolj znanih in enostavnih substitucij je Cezarjeva substitucija (po Juliju Cezarju), ki ima za abecedo B zamaknjeno abecedo A. npr. pri zamiku 3 dobimo:

abeceda A	a b c č d e f
abeceda B	č d e f g h i

Če zgornjo substitucijo uporabimo na prejšnjem zgledu, dobimo naslednji šifrirani tekst
Čistopis: Včeraj so ujeli medveda.
Šifropis: afhtčm ur žmhol phgahgč
Pri tem je bila za abecedo A uporabljena slovenska abeceda s 25 znaki. Za Cezarjevo substitucijo je karakteristično, da ostane abeceda ista. Število ključev je pri tem omejeno s številom črk, zato je dešifriranje enostavno. Dovolj je, da poznamo substitucijo za eno črko in že poznamo ključ. Takšno črko je mogoče hitro najti, če je le šifropis dovolj dolg. Potrebno je le poiskati črko z največjo frekvenco in jo izenačiti z najpogostejšo črko v čistopisu.
Če so črke v abecedi B v poljubnem redu, potem postane število ključev precej večje, namreč 25!.

Primer 2

abeceda A	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž
abeceda B	e	s	t	v	f	u	z	o	g	d	r	b	h	k	c	i	a	j	l	p	m	n	č	ž	š

Čistopis: Včeraj so ujeli medveda.
Šifropis: čvujer li nruhd kufčufe

Kjub 25! možnih ključev je mogoče relativno enostavno rešiti takšen problem. Razlog je v znanih frekvencah črk naravnega jezika in v redundancah jezika. Zato se pri šifriranju uporablja več kot ena substitucija. Tedaj govorimo o polialfabetni substituciji. Primer takšne substitucije je Vigener-jev sistem, kje je uporabljenih več Cezarjevih substitucij. Prva črka v čistopisu je zamaknjena npr. za 20 mest, druga za 17, itd. Pri takšnem šifriranju je ugodno uporabljati t.i. Vigener-jevo tabelo, ki podaja osnovno abecedo A kot prvo vrstico v tabeli, sledijo pa ji vrstice z zamiki 1, 2, 3, … . Število vrstic v tabeli je tako enako 25. Šifriranje se izvaja s pomočjo t.i. ključnega teksta, ki pove, kako izvesti substitucijo. Ključni tekst je enake dolžine kot čistopis. Zapišemo ga pod čistopis. Nato po vrsti preko celega teksta vzamemo črki iz čistopisa in ključnega teksta, ter v Vigener-jevi tabeli (tabela 1) poiščemo presek stolpca (določa ga črka iz čistopisa) in vrstice (določa jo črka ključnega teksta), ki določa črko v šifropisu.

Primer 3
Ključ: radior ad iorad ioradio
Čistopis: včeraj so ujeli medveda
Šifropis: nčiboc aš eavam vtuaimo

V tem primeru so karakteristike jezika veliko bolj skrite kot prej. V zgornjem primeru je ključni tekst sestavljen iz ponavljajoče ključne besede dolžine 5 (radio), kar pomeni, da smo iz Vigenerj-eve tabele uporabili 5 substitucij. Zato poznavanje dolžine ključne besede zelo pomaga pri dešifriranju, oziroma pri dobrem šifriranju je zaželena uporaba čim daljše ključne besede oz. teksta.
Pomembno pri šifriranju je prikriti karakteristike čistega teksta oz. jezika, v katerem je napisan, kar lahko dosežemo z izenačitvijo pogostosti znakov v šifrirnem tekstu. Tedaj moramo najprej šifrirati črke čistopisa npr. z Huffmanovo metodo, zatem pa uporabiti transpozicijsko ali substitucijsko metodo. Tedaj bodo imeli vsi kodni simboli enako verjetnost pojavljanja v šifropisu.
Poznamo več različnih postopkov substitucijskega šifriranja, in sicer:

- enostavno šifriranje,
- homofonično šifriranje in
- polialfabetično šifriranje.

a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž
b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a
c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b
č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c
d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č
e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d
f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e
g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f
h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g
i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h
j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i
k	l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j
l	m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k
m	n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l
n	o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m
o	p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n
p	r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o
r	s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p
s	š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r
š	t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s
t	u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š
u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t
v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u
z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v
ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z

Tabela 1: Viegenerjeva tabela za slovensko abecedo

Produktno šifriranje
je sestavljeno iz več enostavnejših šifriranj , ki so ponavadi zamenjave (enostavna zamenjava, homofonično šifriranje, polialfabetično šifriranje) in prerazporeditve (transpozicijsko šifriranje). Ključ take šifre je ponavadi neki naključni niz, ki vpliva na ključe posamičnih enostavnih šifer. Cilj takih šifer je čim bolj premešati čistopis in čim bolj razpršiti ključ ter vpliv posameznih delov čistopisa po šifropisu.

Pretočno šifriranje
Pri pretočnem šifriranju se predpostavlja, da sporočilo nastaja z zaporednimi sekvencami prostih elementov, ki so ali črke ali binarni znaki. Tekst se tukaj šifrira element za elementom. Pri pretočnih postopkih šifriramo sporočilo sproti, tako da izvajamo XOR operacijo med posameznimi biti sporočila in biti ključa. Dolžina ključa mora biti enaka dolžini sporočila ali pa moramo uporabiti nek postopek, ki krajši ključ podaljša na dolžino sporočila.

Enkratna prevleka (one – time pad, Vernam cipher)
je šifra, pri kateri se čistopisu prišteva ključ po modulu števila znakov v abecedi čistopisa. Ključ mora biti enako dolg, kot čistopis, biti mora naključen, z enakomerno porazdelitvijo znakov in sme biti uporabljen samo enkrat. Ključ tako pomeni niz naključnih znakov. Taka šifra teoretično zagotavlja popolno varnost, vendar ima druge slabosti, kot so dolžina ključa, hranjenje in razdeljevanje takih ključev po varnem kanalu ter primernost šifre samo za zagotavljanje tajnosti (za zagotavljanje verodostojnosti na primer ni primerna). Posledica slabosti je visoka cena pri uporabi te šifre, zato se uporablja le v izjemnih varnostnih razmerah.

1.6 Digitalni podpis

Digitalni podpis je kriptografski temelj za doseganje pristnosti, dodeljevanje pooblastil in za preprečevanje nepriznavanja storjenih dejanj in obveznosti. Namen elektronskega podpisa je priskrbeti način povezovanja identitete neke osebe z informacijo. Proces podpisovanja iz sporočila in zasebne informacije osebe naredi podpis.

Definicija
Digitalni podpis je podatkovni niz v digitalni obliki, namenjen zagotavljanju pristnosti in /ali porekla podatkov.

- M množica sporočil, ki jih želimo podpisati.
- S označuje prostor podpisov.
- S_A je funkcija iz prostora sporočil M v prostor podpisov S in jo imenujemo podpisna funkcija (ali transformacija) osebe A. transformacijo S_A varuje oseba A in jo uporablja za podpisovanje sporočil iz M.
- V_A je funkcija iz prostora $M \times S$ v prostor {da,ne}. V_A imenujemo funkcija preverjanja podpisov osebe A, je javno znana in jo uporabljajo druge osebe za preverjanje podpisov, ki jih je naredila oseba A.

Proces podpisovanja
Oseba A (podpisnik) naredi podpis za sporočilo $m \in M$ takole:
1. Izračuna $s = S_A(m)$, kjer je S_A podpisna funkcija.
2. Pošlje par (m,s). s imenujem podpis za sporočilo m.

Proces preverjanja
Preverjanje podpisa osebe A opravi oseba B takole:
1. Pridobi funkcijo preverjanja V_A od osebe A.
2. Izračuna $u = V_A(m, s)$.
3. Sprejme podpis osebe A, če je u = da, zavrne podpis, če je u = ne.

Transformaciji S_A in V_A sta ponavadi določeni s ključem, tj. razred transformacij podpisovanja in preverjanja je javno znan, vsaka transformacija pa je določena s ključem. Transformacija podpisovanja S_A osebe A je določena s ključem k_A, ki jo mora le-ta varovati. Podobno je transformacija preverjanja V_A osebe A določena s ključem l_A, ki pa je javno znan.

1.7 Avtentikacija

Avtentikacija je postopek, s katerim se ugotavlja identiteta udeleženca v komunikaciji oz. se preverja, če je udeleženec, s katerim se komunicira, resnično tisti, za katerga se predstavlja. Poznamo tri osnovne skupine avtentikacijskih postopkov, in sicer:

- avtentikacija, zasnovana na uporabniškem imenu in geslu,
- avtentikacija, ki temelji na simetrični kriptografiji in
- avtentikacija, ki sloni na asimetrični kriptologiji.

Avtentikacija zasnovana na uporabniškem imenu in geslu
Da bi se uspešno opravil postopek avtentikacije, je potrebno pravilno vpisati kombinacijo uporabniškega imena in gesla. Uporabniško ime je javni podatek, medtem ko je tajnost gesla zelo pomembna. Kdor pozna uporabniško ime in geslo, lahko prevzame identiteto uporabnika.

Avtentikacija, ki temelji na simetrični kriptografiji
Sistem in uporabnik si delita skupno skrivnost (tajno geslo). Pomembno je, da se to geslo nikoli ne prenaša preko telekomunikacijskega sistema, temveč sistem izzove uporabnika tako, da mu pošlje neko naključno sporočilo. Uporabnik odgovori tako, da šifrira to sporočilo s tajnim geslom, in ga pošlje nazaj. Sistem, ki pozna tajni ključ, šifrira svoje sporočilo in ga primerja s sporočilom, ki ga je prejel od uporabnika. Če sta enaka, je identiteta uporabnika potrjena.

Avtentikacija, ki sloni na asimetrični kriptografiji
V tem primeru je avtentikacija možna s pomočjo digitalnega podpisa, ki je predstavljen v prejšnjem poglavju.
″V določenih primerih ne zadošča, da sistem zgolj preveri identiteto uporabnika, ampak mora biti poskrbljeno tudi za to, da kasneje uporabnik ne more zanikati sporočil, ki jih je poslal, ker za ta sporočila tudi odgovarja. Tak primer je elektronsko bančništvo. To, da uporabnik z digitalnim podpisom podpisanih sporočil ne more zanikati, je mogoče zagotoviti samo , če uporabnik prej podpiše dogovor, v katerem potrjuje veljavnost svojega digitalnega podpisa (overi svoj javni ključ) in prevzame odgovornost za uporabo svojega digitalnega podpisa, to je uporabo tajnega ključa, ki je par overjenemu javnemu ključu.″ (Sašo Tomažič Varnost v telekomunikacijah in kako jo zagotoviti)

1.8 Kriptografija z javnim ključem

Asimetrična kriptografija ali kriptografija javnih ključev rešuje problem zamenjave tajnih ključev pri simetrični kriptografiji. Koncept asimetrične kriptografije sta postavila Whitfield Diffie in Martin Hellman leta 1976.

Asimetrična kriptografija za razliko od simetrične kriptografije, kjer se za enkripcijo in dekripcijo uporablja isti ključ, uporablja dva kluča. Ključi se vedno nahajajo v paru in se imenujejo javni ključ (public key) in tajni ključ (private key). Enkripcija podatkov se opravlja z javnim ključem, medtem ko pripadajoči tajni ključ služi za dekripcijo. Čeprav je to splošno načelo, je možen tudi obraten postopek, ko so podatki kriptirani s tajnim ključem, dekriptirani pa z javnim ključem. Lastnost asimetričnih kriptosistemov, da se lahko javni in tajni ključi uporabljajo v različnih vlogah, nam daje dobre avtentikacijske možnosti takih sistemov.
Pri uporabi asimetričnih kriptosistemov je potrebno javni ključ objaviti, da je javno dostopen in poznan, medtem pa se mora tajni ključ držati v tajnosti in ga ne sme poznati nihče drug kot lastnik. Vsak javni ključ ima samo en pripadajoči tajni ključ in obratno, vsak tajni ključ ima samo en pripadajoči javni ključ. Teoretično lahko z matematičnimi metodami dobimo pripadajoči tajni ključ, če poznamo javni ključ in obratno. Moč in odpornost asimetričnih kriptosistemov temelji ravno na matematični povezavi med ključi. Matematična povezava med javnim in tajnim ključem je zasnovana na matematično težko izračunljivih problemih za, katere ne obstajajo oz. niso poznani učinkoviti algoritmi za reševanje. Pogosto uporabljen matematični problem je faktorizacija produkta dveh velikih praštevil. V praktični uporabi to pomeni, če poznamo javni ključ in uporabimo veliko računalniško moč, podobno kot pri preverjanju vseh možnih ključev pri simetričnem kriptosistemu, ni možno v realnem času izračunati pripadajoči tajni ključ. Dolžine ključev pri asimetričnih kriptosistemih so mnogo večje kot pri simetričnih. Za varne ključe štejejo ključi dolžine 1024 bitov, a uporabljajo se tudi daljši.
Tabela 1 prikazuje primerjavo dolžine ključev pri simetričnih in asimetričnih kriptosistemih, ki nudijo enako stopnjo varnosti.

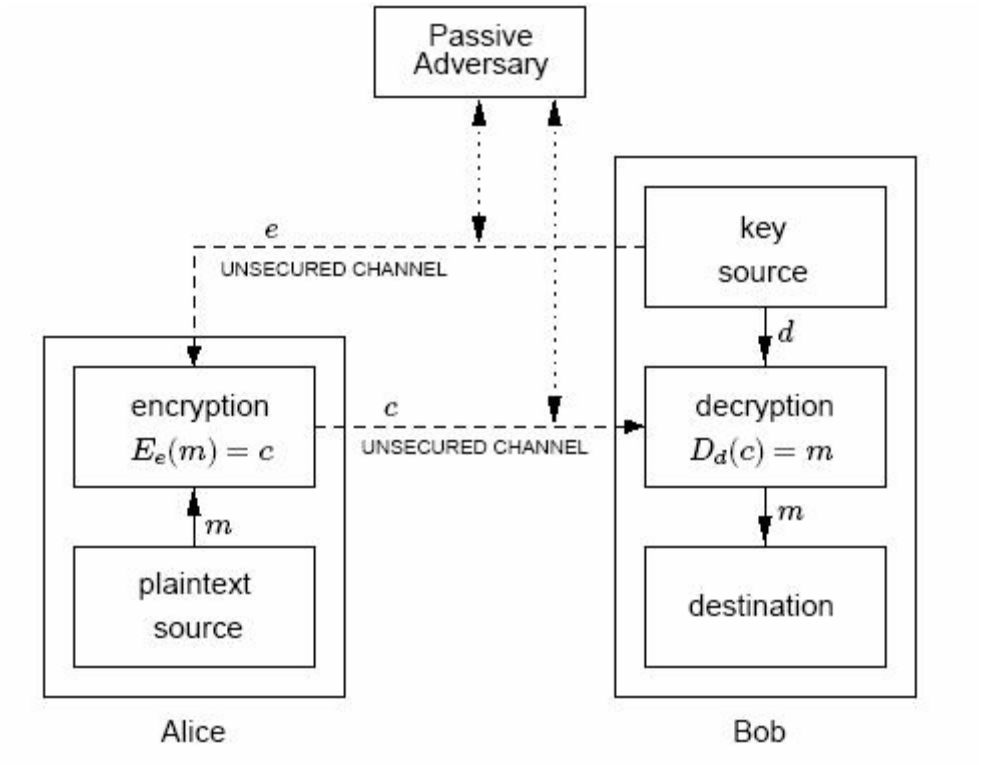
Simetrični kriptosistem	Asimetrični kriptosistem
80	1024
112	2048
128	3072
192	7680
256	15360

Tabela 1

Pomembna prednost asimetrične kriptografije je, da ne potrebuje vzpostavljanje varnega komunikacijskega kanala za zamenjavo tajnega ključa pred vzpostavljanjem varne komunikacije, za katero simetrična kriptografija ni našla odgovora. Druga pomembna prednost asimetrične kriptografije so zelo dobre avtentikacijske možnosti. Slabost asimetrične kriptografije je počasnost v primerjavi s simetrično. Asimetrični kriptografski algoritmi so matematično mnogo zahtevnejši kot simetrični, saj so postopki kriptiranja in dekriptiranja mnogo počasnejši, tudi do 1000 krat. Asimetrični kriptografski algoritmi zaradi tega niso primerni za kriptiranje velikih količin podatkov. Kot rešitev problema pomankljivosti asimetrične in simetrične kriptografije so nastali t.i. hibridni kriptosistemi.

Le-ti uporabljajo asimetrično kriptografijo za zamenjavo tajnega ključa. Nadaljnja komunikacija pa se odvija z uporabo simetrične kriptografije in predhodno izmenjanim tajnim ključem.
Poznamo naslednje asimetrične kriptosisteme:

- Elgemal - imenovan po iznajditelju Taheru Elgamalu,
- RSA - imenovan po iznajditeljih Ronu Rivestu, Adiu Shamiru in Leonardu Adlemanu,
- Diffie-Helman tudi imenovan po iznajditeljih Whitfieldu Diffieu in Martinu Hellmanu,
- DSA Digital Signature Algorithm.



Slika 3: Asimetrična kriptografija

1.8.1 Digitalni podpis iz obratne javne šifrirne sheme

Naj bo E_e šifrirna transformacija na prostoru sporočil M v šifrirni prostor C. Predpostavimo, da je M = C. Naj bo D_d ustrezna dešifrirna transformacija. Potem velja

$$D_d(E_e(m)) = E_e(D_d(m)) = m \text{ za vse } m \in M.$$

Predpostavka, da je M = C je potrebna, da velja enakost za vse $m \in M$, sicer $D_d(m)$ ni definiran za $m \notin C$.

Konstrukcija sheme digitalnega podpisa iz obrnljive javne šifrirne sheme poteka takole:

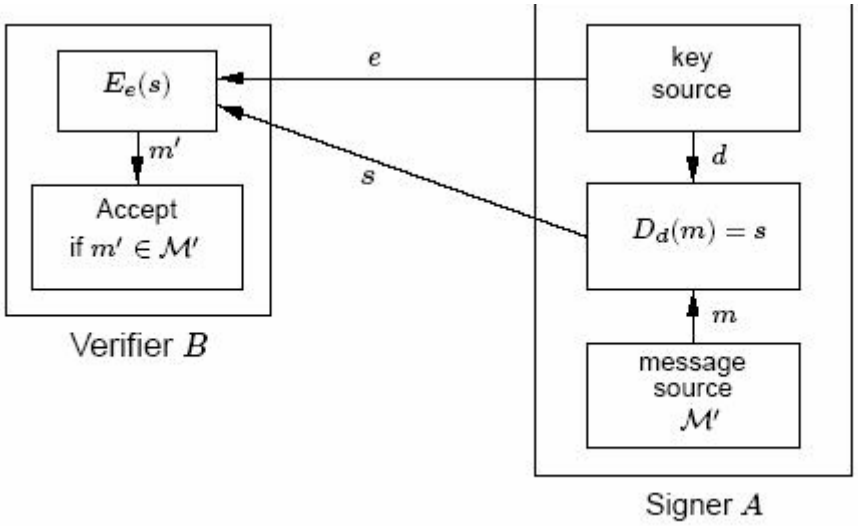
1. Naj bo M prostor sporočil za šifrirno shemo.
2. C = M je prostor podpisov S.
3. Naj bosta (e,d) par ključev za javno šifrirno shemo.
4. Podpisno funkcijo S_A definiramo kot D_d, tj. podpis za sporočilo $m \in M$ je enako $s = D_d(m)$.
5. Funkcijo preverjanja V_A definiramo takole:

$$V_A(m,s) = \begin{cases} da \_\_ če \_ E_e(s) = m \\ ne \_\_ sicer. \end{cases}$$

V tem primeru govorimo seveda o shemi digitalnega podpisa z dodatkom.
Podobno lahko naredimo v primeru sheme digitalnega podpisa s sporočilom. Ker sporočilo dobimo iz podpisa ($m = E_e(s)$), vzamemo za funkcijo preverjanja

$$V_A(s) = \begin{cases} da \_\_ če \_ E_e(s) \in M' \\ ne \_\_ sicer. \end{cases}$$

Kjer je $M' \subseteq M$ prostor sporočil, katerega elementi so izbrane oblike. Npr. sporočilo se začne z določenim znakom ali številom.



Slika 4: Digitalni podpis s sporočilom

Digitalni podpis je v praksi uporaben, če ga lahko podpisnik enostavno izračuna, prejemnik enostavno preveri in je varen pred ponaredbo v času njegove veljavnosti. Podpis $s \in S$ za sporočilo $m \in M$ je veljaven natanko takrat, ko $V_A(m,S)$ = da. Varnost pred ponaredbo pa pomeni, da je za vse, z izjemo podpisnika, računsko neizvedljivo poiskati tak $m \in M$ za $s \in S$, da je $V_A(m,s)$ = da (vsaj v času veljavnosti podpisa).

1.9 Zgostitvene (zgoščevalne) funkcije

Eden od pomembnih temeljev moderne kriptografije predstavljajo zgoščevalne funkcije (angl. hash function), imenovane tudi enosmerne zgoščevalne funkcije. Te se uporabljajo za preverjanje celovitosti podatkov v povezavi z digitalnim podpisom, kjer na sporočilu najprej uporabimo zgoščevalno funkcijo, potem pa zgoščeno vrednost (ang. hash-value), kot predstavnika tega sporočila, podpišemo.
Formalna definicija zgoščevalna funkcije je naslednja.
Zgoščevalna funkcija je v splošnem funkcija h, ki preslika binarni vnosni niz x poljubne dolžine v binarni niz h(x) fiksne dolžine. Vrednost h(x) imenujemo zgoščena vrednost. Zgoščevalna funkcija mora imeti vsaj dve lastnosti: stiskanje(veliko definicijsko območje slika v majhno zalogo vrednosti) in enostavno računanje. Osnovna ideja zgoščevalnih funkcij je ta, da zgoščena vrednost predstavlja kompaktno reprezentativno sliko vnosnega niza, imenovano tudi digitalni odtis, ki jo lahko uporabimo tako, kot da bi bila enolično določena z vnosnim nizom.
Zgoščevalna funkcija iz sporočila izdela zgoščeno vrednost. Bolj natančno, zgoščevalna funkcija h preslika niz bitov poljubne dolžine v niz določene dolžine, npr. n bitov.Funkcija h z definicijskim območjem D in zalogo vrednosti R, kjer $|D| > |R|$, slika več elementov v eno sliko , zato prihaja do t.i. trkov. Če funkcijo h omejimo na definicijsko območje z elementi dolžine t, $t > n$, in je h 'naključna' v smislu, da je vsaka slika enako verjetna, potem ima približno 2^{t-n} elementov isto sliko, dva naključno izbrana elementa pa z verjetnostjo 2^{-n} isto sliko (neodvisno od t).
Tipičen primer uporabe zgoščevalnih funkcij je naslednji. Ob trenutku T_1 izračunamo zgoščeno vrednost sporočila x. Prisotnost te vrednosti (ne pa tudi sporočila) na nek način ohranimo. Čez čas, v trenutku T_2, opravimo naslednji test, da ugotovimo, ali je bilo sporočilo spremenjeno. Izračunamo zgoščeno vrednost sporočila x' in jo primerjamo z varovano zgoščeno vrednostjo x. Če sta vrednosti enaki, potem verjamemo, da sta tudi sporočili enaki oziroma, da sporočilo ni bilo spremenjeno. Problem ohranjanja velikega sporočila se zmanjša na problem ohranjanja zgoščene vrednosti. Ker za zgoščevalne funkcije obstaja verjetnost trkov, mora biti zgoščena vrednost enolično določena s sporočilom vsaj v praksi. Iskanje trkov mora biti računsko neizvedljivo (v praksi se trki tako rekoč ne smejo pojavljati).
Primeri zgoščevalnih funkcij so MD4, RIPMED-160 in SHA-1 (Secure Hash Algorithm).

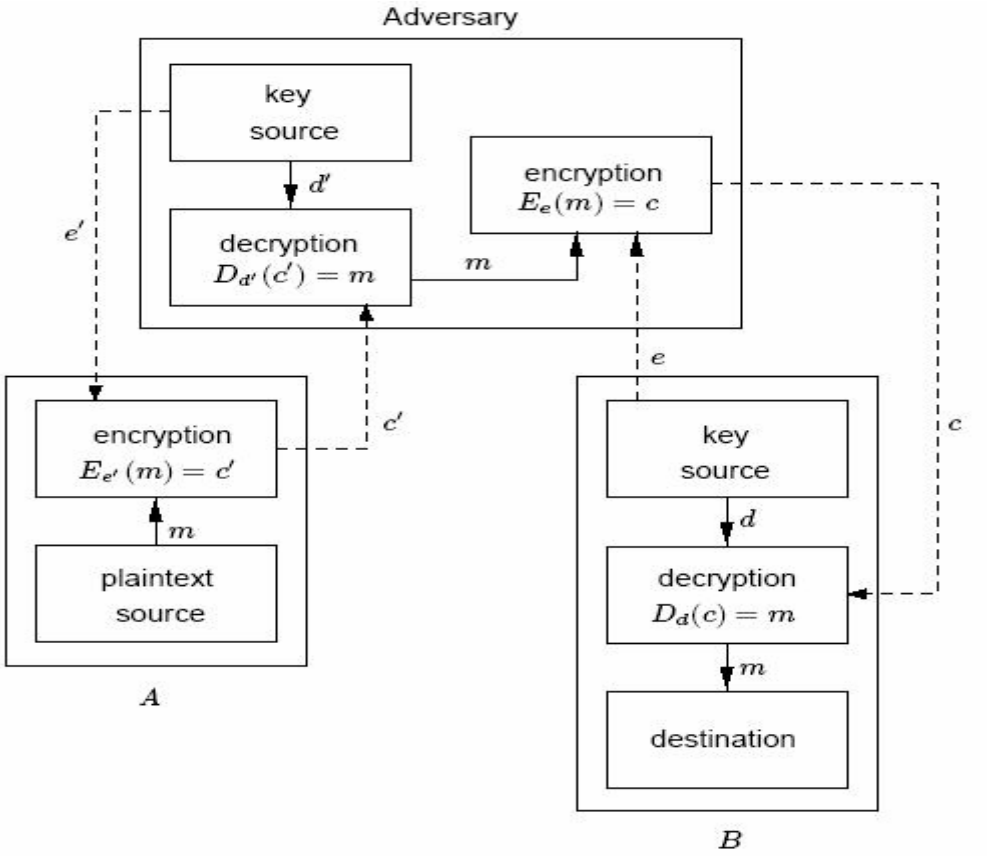
1.10 Tvorba , upravljanje in certificiranje ključa

1.10.1 Pomen ključev

Zasebnost pri asimetričnem šifriranju in verodostojnost digitalno podpisanega dokumenta temelji na tajnosti skritega ključa. V primeru, da nekdo tajni ključ odkrije, z njim lahko dešifrira vse zaupne dokumente, ki so šifrirani s pripadajočim javnim ključem, ter lahko digitalno podpisuje dokumente, katerih mi ne bi podpisali. Za tajnost skritega ključa moramo sami prevzeti odgovornost in ga ustrezno varovati.

Javni ključ pa mora biti čim bolj javen. V primeru, da želimo prejemniku B, poslati zaupne podatke, jih šifriramo z njegovim javnim ključem in smo prepričani, da teh podatkov ne bo mogel prebrati nihče drug kot prejemnik B, ki pozna svoj tajni ključ. Trditev drži v primeru, da je bil javni ključ res ključ prejemnika B.

Problem lažne identitete
Lahko pa se zgodi, da sovražniku uspe podtakniti lažen javni ključ in s tem lahko v nadaljevanju dešifrira zaupne podatke. Zaupnih podatkov, šifriranih z podtaknjenim javnim ključem, prejemnik B ne more dešifrirati in na ta način se ugotovi, da je nekaj narobe, vendar je takrat lahko že prepozno.
Sovražnik, ki je podtaknil javni ključ, lahko šifrirane zaupne podatke tudi v celoti prestreže, jih dešifrira, prebere in ponovno šifrira s pravim javnim ključem in jih pošlje prejemniku B. Prejemnik sploh ne opazi, da so bili podatki prestreženi, in na videz je vse v najlepšem redu.



Slika 5: Problem lažne identitete

Problem zanikanja identitete
Pri digitalnem podpisu služi javni ključ za preverjanje podpisa na ustreznem dokumentu. Pri določenih dokumentih (pogodbah) s podpisom ne potrjujemo samo vsebine, ampak sprejemamo tudi določene obveznosti. Podpis je tisti, ki nas obvezuje, da obveznosti tudi opravimo. V primeru, da ne opravimo določenih obveznosti se lahko pojavi problem zanikanja javnega ključa in s tem tudi svojega podpisa. Navedeni problem se imenuje problem zanikanja identitete.

1.10.2 Digitalna potrdila javnih ključev in overitelji

Digitalno potrdilo javnega ključa (public key certificate) je digitalni dokument, ki potrjuje povezavo med javnim ključem in osebo ali institucijo ali strežnikom. Z njim lahko preverimo, komu pripada javni ključ. Potrdilo vsebuje javni ključ in informacijo o njegovem imetniku, ki ju podpiše oseba ali institucija, ki ji zaupamo. Potrdila so objavljena v splošno dostopnih imenikih ali na spletnih straneh. Uporabljamo jih za identifikacijo v elektronskem poslovanju, prav tako kot osebno izkaznico v vsakdanjem življenju.
Podobno, kot je nastajal sistem elektronskih naslovov in imen računalnikov, so začele nastajati posamezne infrastrukture javnih ključev (udomačena je kratica *PKI za public key infrastructure*), ki jih vpeljujejo vlade (Kanada, ZDA, Singapur, nekatere evropske države) ali posebne organizacije (Verisign, Thawte, EuroTrust, ...). Če imata dva overitelja sorodno politiko preverjanja, kompatibilno opremo in si zaupata, lahko skleneta dogovor, da se medsebojno priznavata. Tako se njuni infrastrukturi javnih ključev povežeta in uporabniki obeh lahko varno izmenjujejo podatke. Prvotna ideja je bila, da bi se tako postopoma gradila svetovna PKI obenem z imenikom po standardu X.500. Zdaj takih pričakovanj ni več - PKI bodo ostale omejene na posamezna območja ali aplikacije, znotraj katerih je možno natančno določiti imetnika digitalnega potrdila in namen uporabe potrdila. Potrdilo na ime "Janez Novak" ne pove dovolj, če je oseb s tem imenom več, ključa, kot je EMŠO, za povezavo z različnimi bazami podatkov pa zaradi zakona o varstvu osebnih podatkov ne sme vsebovati. Potrdilo za Janeza Novaka z neko negovorečo enolično identifikacijsko številko mora biti na nek način povezano z zalednimi aplikacijami, katerim je namenjeno - zagotovljena mora biti povezava na običajne identifikatorje (EMŠO, davčna številka, številka bančnega računa itd.). Tudi ugotavljanje veljavnosti digitalnega potrdila je v omejenih področjih lažje rešljivo.
Infrastrukturo javnih ključev določajo postopki in oprema za:

- generiranje in hranjenje ključev,
- overjanje imetnikov ključev in izdajanje digitalnih potrdil javnih ključev,
- objavljanje digitalnih potrdil (imeniki),
- preklicevanje digitalnih potrdil,
- časovno označitev postopkov.

Središčni del predstavlja *overitelj javnih ključev* (Certification Authority -CA). Vsak overitelj objavi svoj javni ključ in dokument (Certification Policy), ki opisuje postopek, kako in komu podeljuje potrdila ter na kakšen način varuje svoj zasebni ključ.
Glede na zahtevnost postopka preverjanja identitete tistega, ki mu bo izdal digitalno potrdilo, overitelj lahko izdaja digitalna potrdila na različnih nivojih zaupanja. Lahko npr. določi, da se mora posameznik osebno zglasiti in predložiti osebni dokument, lahko pa podeli digitalno potrdilo na osnovi zahtevka, poslanega po elektronski pošti. Jasno je, da je mogoče bolj zaupati digitalnemu potrdilu, podeljenemu po prvem postopku kot po drugem. Poskrbeti mora, da so imetniki digitalnih potrdil enolično določeni (posameznik ima lahko več javnih ključev in torej tudi digitalnih potrdil) in za poseben seznam preklicanih digitalnih potrdil (torej tistih digitalnih potrdil, ki so iz različnih vzrokov neveljavni).

Pomembno je tudi, da overitelj poskrbi za varnost svojega zasebnega ključa, saj bi bila sicer potrdila, ki jih je izdal, brez pomena lahko bi prišlo celo do poneverb, ki bi jih prepozno opazili. Hraniti ga morajo na dobro zaščitenem računalniku.
Čeprav je ideja PKI stara že več kot dvajset let, proces izčiščevanja standardov na tem področju ni končan in smemo zaradi tega uporabniki pričakovati le omejeno povezljivost med različnimi produkti. Izhodišče je standard za digitalno potrdilo X.509V3, ki je splošno sprejet (če izvzamemo PGP). V okviru IETF deluje posebna delovna skupina PKIX, ki pripravlja standarde za PKI. Eden od evropskih projektov je bil PKI challenge in v njegovem oviru je bilo pripravljeno priporočilo za uporabo PKI.
Oblika digitalnega potrdila po standardu ISO/IEC X.509V3:

- verzija (zdaj do verzije 3),
- serijska številka (enolična za potrdila posameznega overitelja),
- algoritmi in parametri (npr. SHA1 in RSA),
- izdajatelj (overitelj javnih ključev),
- datuma veljavnosti od -do ,
- prejemnik digitalnega potrdila (njegovo ime, drugi podatki o njem),
- podatki o njegovem javnem ključu:
    - algoritem,
    - parametri,
    - javni ključ,
- enolična oznaka uporabnika (samo v verzijah 2 in 3),
- razširitve (verzija 3),
- digitalen podpis teh podatkov, ki je narejen z zasebnim ključem CA.

Če pride do zlorabe, če pozabimo geslo za uporabo svojega zasebnega ključa ali pa se je pokvarila naprava, kjer smo ključ hranili, je treba tvoriti nov par ključev in dobiti novo digitalno potrdilo, staro pa preklicati. Vsa digitalna potrdila, ki so iz različnih razlogov neveljavna, objavljajo overitelji na posebnih seznamih, za katere se je uveljavila kratica *CRL (Certificate Revocation List)*. Ti seznami se objavljajo na spletnih strežnikih overiteljev ali pa v imenikih po standardu X.500, kjer so dostopni prek protokola LDAPv3.

1.11 Vrste napadov

Cilj napadov na šifrirne sheme je pridobivanje sporočil iz šifriranih sporočil ali celo, pridobitev zasebnega ključa. Večino napadov na šifrirne sheme lahko uporabimo tudi za sheme digitalnih podpisov. V tem primeru je cilj napadov najti zasebni ključ ali pa vsaj ponaredba digitalnega podpisa, tj. izdelati podpis, ki bo sprejet kot podpis neke druge osebe.

Napade nasprotnikov razdelimo v dva razreda:

- Pasivni napadi. V tem primeru nasprotnik spremlja le komunikacijske kanale. Pasivni napadalec lahko ogrozi le zaupnost podatkov.
- Aktivni napadi. V tem primeru nasprotnik želi zbrisati, dodati ali kako drugače spremeniti pretok podatkov po informacijskih kanalih. Aktivni napadalec ogroža poleg zaupnosti še pristnost in celovitost podatkov.

Kriptosistemi z javnim ključem niso nikoli brezpogojno varni. Nasprotnik lahko ob poznavanju šifriranega sporočila c uporabi javno šifrirno funkcijo E_k za vsa možna sporočila, dokler ne velja $c = E_k(m)$. Sporočilo m je dešifrirano sporočilo za c. Zato, ko govorimo o varnosti, ponavadi mislimo o izračunljivi varnosti določenega kriptosistema. Pri izračunljivi varnosti merimo količino potrebnega računskega truda za razbitje sistema po trenutno najboljših metodah. Varnost temelji na argumentih, da uspešen napad potrebuje neko stopnjo sredstev (npr. časa, prostora in denarja), ki je večja od razpoložljivih. Javni kriptosistemi navadno temeljijo na težko rešljivih računskih problemih.

1.11.1 Napadi na šifre

Napad na šifro je poskus kriptoanalize šifre. Pri napadu se predpostavlja, da kriptoanalitik pozna vse podrobnosti šifre (Kerckhoffosova predpostavka) in ima na voljo nekaj šifropisa. Poleg tega lahko kriptoanalitik pozna kakšen par čistopisa in šifropisa, lahko celo izbira čistopis, ki ga hoče šifrirati, in podobno. Glede na možnosti, ki jih ima kriptoanalitik, ločimo več vrst napadov na šifro. Najpomembnejše vrste so:

- napad samo s šifropisom,
- napad z grobo silo (izčrpno iskanje),
- napad z izbranim čistopisom,
- napad z izbranim šifropisom in
- napad z znanim čistopisom.

Napad samo s šifropisom (ciphertext-only attack)
je vrsta napada na šifro, pri katerem ima kriptoanalitik na voljo šifropis več sporočil (iste šifre) in skuša odkriti čim več ustreznega čistopisa ali celo ključ ali ključe, ki so bili uporabljeni pri šifriranju.

Napad z grobo silo (brute-force attack)
je vrsta napada z znanim čistopisom, pri čemer kriptoanalitik pri šifriranju znanega čistopisa preizkuša vse ključe po vrsti in rezultat primerja z danim šifropisom. Če je šifra tako dobra, da je najboljši napad nanjo izčrpno iskanje, mora le imeti dovolj dolg ključ, da je ni mogoče kriptoanalizirati.

Napad z izbranim čistopisom (chosen-plaintext attack)
je vrsta napada na šifro, pri katerem ima kriptoanalitik možnost izbirati čistopis, ki se šifrira , in dostop do izhoda šifre. Ker lahko izbere kakršenkoli čistopis, ima tako še boljše možnosti

za uspeh kot pri napadu z znanim čistopisom. Naloga kriptoanalitika je odkriti ključ ali ključe, ki so uporabljeni pri šifriranju, ali poiskati algoritem za (nelegitimno) dešifriranje bodočih šifropisov, pri katerem so bili uporabljeni isti ključi. Proti tej vrsti napda morajo biti odporne šifre z javnimi ključi, ker ima kriptoanalitik vedno možnost takega napada, saj pozna javni ključ.

Napad z izbranim šifropisom (chosen-ciphertext attack)
je vrsta napada na šifro, pri katerem ima kriptoanalitik možnost, da izbira domnevne šifropise, ki se dešifrirajo, in dostop do rezultata dešifriranja. Naloga kriptoanalitika je odkriti ključ. Tak napad je mogoč na primer s krajo dešifrirne naprave.

Napad z znanim čistopisom (known-plaintext attack)
je vrsta napada na šifro, pri kateri ima kriptoanalitik poleg šifropisa več sporočil na voljo tudi nekaj parov čistopisa in ustreznega šifropisa. Kriptoanalitik si prizadeva odkriti ključe, ki so bili uporabljeni za šifriranje, ali algoritem za (nelegitimno) dešifriranje bodočih šifropisov, pri katerih so bili uporabljeni isti ključi. Ta vrst napada je možna, če je mogoče priti do čistopisa z drugimi sredstvi (sporočilo je dobljeno nezakonito ali je preprosto objavljeno v časopisu)

2 Matematična orodja

V tem poglavju so predstavljena osnovna matematična poglavja, kot so teorija verjetnosti, informacijska teorija, teorija zahtevnosti, teorija števil in abstraktna algebra.

2.1 Teorija verjetnosti

Teorija verjetnosti se ukvarja s pojmi, kot so poskus, dogodek, naključnost, verjetnost itd. Poskus je realizacija natanko določenih pogojev, pri katerih opazujemo enega ali več pojavov. Če pogoje spremenimo, s tem spremenimo celoten poskus. Pojav, ki ga opazujemo pri poskusu imenujemo dogodek.
Poznamo tri vrste dogodkov:

- gotov dogodek (G) (dogodek se zgodi vedno),
- nemogoč dogodek (N) (dogodek se ne zgodi nikoli),
- slučajni dogodek (dogodek se včasih zgodi, včasih pa ne).

2.1.1 Osnovne definicije

Aksiomatična definicija verjetnosti:

1. Način dogodkov: $E_1 \subseteq E_2$. Vedno, ko se zgodi E_1, se zgodi tudi E_2.
2. Enakost dogodkov: $E_1 = E_2 \Leftrightarrow E_1 \subseteq E_2$ in $E_2 \subseteq E_1$.
3. Vsota dogodkov : $E_1 + E_2$. Zgodi se vsaj eden od dogodkov E_1, E_2.
4. Produkt dogodkov: E_1E_2. Zgodita se oba dogodka E_1 in E_2 hkrati.
5. Nezdružljiva dogodka: $E_1E_2 = N$.
6. Nasprotna dogodka (komplementarna dogodka) $E_1E_2 = N$ in $E_1 + E_2 = G$. Tedaj pišemo $E_2 = \overline{E}_1$ in imenujemo $\overline{E}_1$ nasprotni (komplementarni) dogodek k dogodku E_1.
7. Pri večkratni ponovitvi poskusa, se lahko zgodijo dogodki $E_1, E_2, \ldots, E_n$. množica dogodkov $S = \{E_1, E_2, \ldots, E_n\}$ je poln sistem, če se pri vsaki ponovitvi poskusa zgodi natanko eden izmed njih. Iz tega sledi, da so dogodki paroma nezdružljivi $E_i E_j = N$, $(i \neq j)$ in vsota dogodkov iz S je gotov dogodek G $E_1 + E_2 + \ldots + E_n = G$.

Definicija
Verjetnostna porazdelitev P na S je niz števil $p_1, p_2, \ldots, p_n$, ki so nenegativna in njihova vsota je 1. Število p_i interpretiramo kot verjetnost, da se je s_i zgodil.
(S – vzorčni prostor, s_i – elementarni dogodek)

Definicija
Dogodek E je podmnožica vzorčnega prostora S. Verjetnost, da se je dogodek E zgodil, označimo z P(E), ki je enaka vsoti verjetnosti p_i vseh elementarnih dogodkov s_i, ki pripadajo E. Če $s_i \in S$, $P(\{s_i\})$ to lahko enostavno zapišemo z $P(s_i)$.

Definicija
Komplementarni dogodek $\overline{E}$ je množica elementarnih dogodkov, ki ne pripadajo E.

Vzemimo $E \subseteq S$, potem velja:

- $0 \leq P(E) \leq 1$ (P(S) = 1, in P(O) = 0 O-prazna množica),
- $P(\overline{E}) = 1 - P(E)$,

- če so izidi v S enako verjetni, potem velja P(E) = $\frac{|E|}{|S|}$.

2.1.2 Pogojna verjetnost

Definicija
Dana sta dva dogodka E_1 in E_2 in njuni verjetnosti $P(E_1)$ in $P(E_2)$.Vzemimo da se je E_2 zgodil in ima nato E_1 neko verjetnost, ki je različna od $P(E_1)$. Tej verjetnosti pravimo pogojna verjetnost dogodka E_1 pri pogoju E_2 in velja zveza:

$$P(E_1|E_2) = \frac{P(E_1E_2)}{P(E_2)}.$$

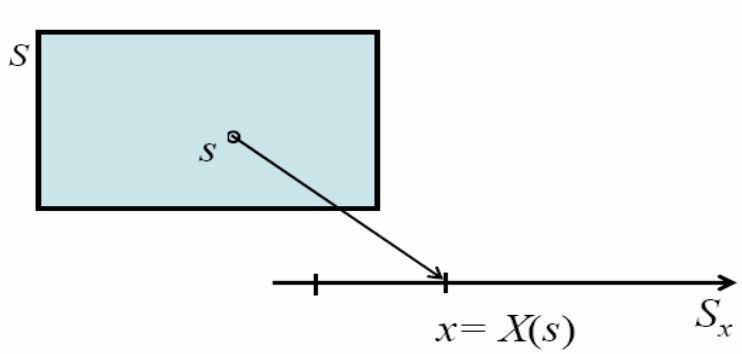
Definicija
Dogodka E_1 in E_2 sta medsebojno neodvisna, kadar velja $P(E_1|E_2) = P(E_1)$ in $P(E_2|E_1) = P(E_2)$ in za neodvisna dogodka velja $P(E_1E_2) = P(E_1)\,P(E_2)$.

Bayesova formula

Če sta E_1 in E_2 dogodka z $P(E_2) > 0$, potem velja $P(E_1|E_2) = \frac{P(E_1)P(E_2|E_1)}{P(E_2)}$.

2.1.3 Naključne spremenljivke

Definicija
V splošnem izidi naključnega poskusa predstavljajo elementarne dogodke s vzorčnega prostora S, $s \in S$. Elementarnim dogodkom s prostora S priredimo neko realno spremenljivko X. Pri tem posamezne vrednosti spremenljivke X označimo z x, $x \in X$. Vsakemu elementarnemu dogodku $s \in S$ priredimo pripadajočo realizacijo x spremenljivke $X(s) = x \in X$.



Ker so izidi poskusa različni in naključni, zavzame tudi spremenljivka X različne in naključne vrednosti $\Rightarrow$ X je naključna spremenljivka. Naključna spremenljivka X(s) je predpis ali funkcija, ki vsaki vzorčni točki s prostora S priredi vrednost x. Predpis X(s) je smiseln, če vsakemu elementarnemu dogodku $s \in S$ pripada ena sama vrednost $x \in X$.

Definicija
Naj bo X naključna spremenljivka v prostoru S. Pričakovana vrednost ali povprečje X je enako $E(X) = \sum_{s_i \in S} X(s_i)P(s_i)$.

Naj bo X naključna spremenljivka v prostoru S, potem velja $E(X) = \sum_{x \in R} x \cdot P(X = x)$.

Če so $X_1, X_2, \ldots, X_m$ naključne spremenljivke v S in $a_1, a_2, \ldots, a_m$ realna števila potem velja $E(\sum_{i=1}^{m} a_i X_i) = \sum_{i=1}^{m} a_i E(X_i)$.

Definicija
Varianca ali disperzija naključne spremenljivke X je definirana z naslednjim izrazom $\mathrm{Var}(X) = E((X - \mu)^2)$.
Standardna deviacija ali standardni odklon X je kvadratni koren iz Var(X).

2.1.4 Binomska porazdelitev

Definicija

Vzemimo, da sta n in k nenegativni celi števili. $\binom{n}{k}$ je binomski koeficient in pomeni število podmnožic z k elementi iz množice z n elementi.

Lastnosti binomskega koeficienta so:

- $\binom{n}{k} = \frac{n!}{k!(n-k)!}$,
- $\binom{n}{k} = \binom{n}{n-k}$,
- $\binom{n+1}{k+1} = \binom{n}{k} + \binom{n}{k+1}$.

Binomski izrek
Za vsaki realni števili a, b in nenegativno celo število n velja naslednji izraz

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} \cdot a^k \cdot b^{n-k} .$$

Bernoullijev poskus
Imamo končno ponavljanje istega poskusa. Pri vsakem od n poskusov se zgodi dogodek A s stalno verjetnostjo p ali komplementaren dogodek A' z verjetnostjo 1-p. Verjetnost, da se v seriji n poskusov zgodi dogodek A natanko k-krat izračunamo po naslednji enačbi:

$$\binom{n}{k} \cdot p^k (1-p)^{n-k} \quad \text{za vsak} \quad 0 \le k \le n .$$

2.1.5 Rojstnodnevni problem

Definicija
Za pozitivni celi števili m, n s tem, da velja de je m≥ n, je število $m^{(n)}$ definirano z naslednjim izrazom: $m^{(n)} = m(m-1)(m-2)...(m-n+1)$.
Vzemimo m, n nenegativni celi števili m≥ n. Stirlingovo število drugega reda je
$\left\{ {m \atop n} \right\} = \frac{1}{n!}\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}\cdot k^m \qquad \left\{ {0 \atop 0} \right\} = 1$.

Problem
V posodi imamo m žogic, ki so oštevilčene od 1 do m. Iz posode potegnemo žogico, zapišemo številko in jo vrnemo v posodo, to naredimo n-krat. Verjetnost, da potegnemo natančno t različnih žogic iz posode je $P_1(m,n,t) = \left\{ {n \atop t} \right\} \frac{m^{(t)}}{m^n} \quad 1 \le t \le n$.

Rojstnodnevni paradokos
V posodi imamo m žogic, ki so oštevilčene od 1 do m. Iz posode potegnemo žogico, zapišemo številko in jo vrnemo v posodo to naredimo n-krat. Verjetnost da sta vsaj dve žogici enako oštevilčeni je $P_2(m,n) = 1 - P_1(m,n,n) = 1 - \frac{m^{(n)}}{m^n} \quad 1 \le n \le m$.
Verjetnost, da imata dve osebi v sobi s 23 osebami isti dan rojstni dan, je $P_2(365,23) \cong 0{,}507$.
Verjetnost $P_2(365,n)$ zelo narašča, če narašča n (primer $P_2(365, 30) \cong 0{,}706$).

2.2 Informacijska teorija

2.2.1 Entropija

Entropija je veličina, ki določa (podaja) mero nedoločenosti.

Naj bo X naključna spremenljivka, katera zavzame vrednosti $x_1, x_2, \ldots, x_n$, z verjetnostjo $P(X = x_i) = p_i$, kjer je $0 \le p_i \le 1$ za vsak i $1 \le i \le n$ in kjer je $\sum_{i=1}^{n} p_i = 1$.

Definicija
Entropija oziroma mera nedoločenosti je definirana z naslednjo enačbo
$H(X) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i = \sum_{i=1}^{n} p_i \cdot \log_2\left(\frac{1}{p_i}\right)$. (Če je $p_i = 0$ potem ni nedoločenosti.)

Lastnosti entropije
X naj bo naključna spremenljivka, ki lahko zavzame n vrednosti.

- $0 \le H(X) \le \log_2 n$ (zgornjo mejo entropije določa logaritem števila stanj X),
- H(X) = 0, če je $p_i = 1$ za nek i in $p_j = 0$ za vse $i \neq j$,
- $H(X) = \log_2 n$ nedoločenost je največja, če so vsa stanja enako verjetna $p_i = 1/n$.

Definicija
Sestavljena entropija X in Y je definirana z
$H(X,Y) = -\sum_{x,y} P(X = x, Y = y) \log_2 (P(X = x, Y = y))$.
Če sta X in Y naključni spremenljivki, potem velja $H(X,Y) \le H(X) + H(Y)$ pri pogoju, da sta X in Y neodvisna.

Definicija
Če sta X in Y naključni spremenljivki, potem je entropija spremenljivke X, ko je spremenljivka Y enaka y podana z enačbo
$H(X|Y = y) = -\sum_{x} P(X = x|Y = y) \cdot \log(P(X = x|Y = y))$.
Pogojna entropija spremenljivke X glede na spremenljivko Y je enaka
$H(X|Y) = \sum_{y} P(Y = y) \cdot H(X|Y = y)$.
Lastnosti pogojne entropije
X in Y naj bosta naključni spremenljivki.

- $H(X|Y) \ge 0$ in $H(X|X) \ge 0$,
- $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$,
- $H(X|Y) \le H(X)$.

2.2.2 Medsebojna informacija

Definicija
Medsebojna informacija naključnih spremenljivk X in Y je enaka $I(X;Y) = H(X) - H(X|Y)$.
Podobno je medsebojna informacija spremenljivke X in para Y, Z definirana z
$I(X;Y,Z) = H(X) - H(X|Y,Z)$.

Lastnosti medsebojne informacije

- Če sta X in Y neodvisna, potem je $H(X|Y) = H(X)$ in $I(X,Y) = 0$.
- Velja simetrija $I(X,Y) = I(Y,X)$.
- $I(X;Y) \geq 0$.

Definicija
Pogojna medsebojna informacija je dana z $I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$.

2.3 Teorija zahtevnosti

2.3.1 Osnovne definicije

Glavna naloga teorije zahtevnosti je klasifikacija računskih problemov glede na vložena sredstva za dosego rešitev. Sredstva so lahko čas, prostor, število procesorjev, denar, itd. Probleme najpogosteje klasificiramo glede na vloženi čas, tj. časovno zahtevnost.

Definicija
Algoritem je računski postopek, ki nam pri danih podatkih v končnem času reši problem in vrne rezultat.

Definicija
Delovni čas algoritma pri določenih podatkih je enak številu izvedenih enostavnih operacij. Največkrat enostavna operacija pomeni osnovno računsko operacijo procesorja v računalniku, npr. seštevanje, množenje, SHL (pomik v levo) ali XOR (ekskluzivni ALI).

Velikost podatkov se ponavadi meri v številu bitov, ki jih zaseda. Npr. velikost pozitivnega števila n je lg n = $1+\lfloor \log_2 n \rfloor$. V tem primeru je algoritem linearen, kvadratičen ali polinomski v lg n, če je njegov delovni čas zaporedoma O(lg n), O((lg n)^2) ali O(P(lg n)), kjer je P polinom.

2.3.2 Asimptotična notacija

Večkrat je težko določiti natančen delovni čas algoritma, zato si pomagamo z asimptotično oceno, kjer gledamo, kako se delovni čas algoritma povečuje s povečevanjem velikosti podatkov.

Definicija (vrste notacij):

- asimptotična zgornja meja
  f(n) = O(g(n)), če obstaja pozitivna konstanta c in tako pozitivno število n_0, da velja $0 \le f(n) \le c \cdot g(n)$ za vse $n \ge n_0$.Z drugimi besedami, f(n) ne raste asimptotično hitreje kot g(n) do multiplikativne konstante natančno.
- asimptotična spodnja meja
  f(n) = Ω(g(n)), če obstaja pozitivna konstanta c in tako pozitivno število n_0, da velja $0 \le c \cdot g(n) \le f(n)$ za vse $n \ge n_0$.
- Asimptotična srednja(?) meja
  f(n) = Θ(g(n)) če obstajata taki pozitivni konstanti c_1 in c_2 in tako pozitivno število n_0, da velja $c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n)$ za vse $n \ge n_0$.
- o-notacija f(n) = o(g(n)), če za vsako pozitivno konstanto c, obstaja tako število $n_0 > 0$, da velja $0 \le f(n) \le c \cdot g(n)$ za vse $n \ge n_0$.

2.3.3 Razredi zahtevnosti

Definicija
Polinomski časovni algoritem je algoritem, katerega najslabši delovni čas je funkcija $O(n^k)$, kjer je n velikost podatka in k konstanta. Algoritem, katerega delovni čas ne moremo tako omejiti, imenujemo eksponentni algoritem.

Polinomskim algoritmom pravimo tudi učinkoviti, medtem ko so eksponentni neučinkoviti algoritmi. Vendar to v praksi ne drži vedno. Včasih se zgodi, da je eksponentni algoritem pri določenih podatkih učinkovitejši od polinomskega. Nasploh je v kriptografiji povprečni delovni čas algoritma pomembnejši od najslabšega. Npr. kriptosistem smatramo za varnega, če je ustrezen kriptoanaliticen problem v povprečju (ali še bolje, vedno) težak problem.

Teorija zahtevnosti se ukvarja predvsem z odločitvenimi problemi. To so problemi, za katere je rešitev problema odgovor DA ali NE. V praksi to ne predstavlja kakšne posebne omejitve. Računski problem lahko prevedemo v primeren odločitveni problem tako, da nam učinkovite rešitve odločitvenih dajo učinkovite rešitve računskih problemov in obratno.

Definirajmo naslednje razrede odločitvenih problemov

Definicija
Razred P je množica odločitvenih problemov, ki so rešljivi v polinomskem času.

Definicija
Razred NP je množica odločitvenih problemov, za katere lahko odgovor DA preverimo v polinomskem času s pomočjo neke dodatne informacije, ki pa jo je ponavadi težko pridobiti, če jo imamo, potem lahko učinkovito preverimo odgovor. V razredu NP govorimo o nedeterministično polinomskih odločitevnih problemih.

Definicija
Razred co-NP je množica odločitvenih problemov, za katere lahko odgovor NE preverimo v polinomskem času z ustrezno dodatno informacijo (podobno kot pri razredu NP).

Primer

Trditev
Velja $P \subseteq NP$ in $P \subseteq$ co-NP, še vedno pa so odprta naslednja vprašanja teorije računske zahtevnosti:
ali je P = PN ?
ali je NP = co-NP ?
ali je $P = NP \cap$ co-NP ?
Verjamemo, da je odgovor na vsa tri vprašanja ne, čeprav tega še nikomur ni uspelo dokazati.

Definicija
Naj bosta L_1 in L_2 dva odločitvena problema. Pravimo, da se da problem L_1 polinomsko prevesti na problem L_2 (pišemo $L_1 \leq_P L_2$), če obstaja algoritem, ki reši problem L_1, za katerega velja:
kot podrutino vsebuje hipotetični algoritem, ki reši problem L_2 in izvede se v polinomskem času, če se polinomsko izvede tudi algoritem za L_2.

Z drugimi besedami, če $L_1 \leq_P L_2$, potem je problem L_2 vsaj tako težak kot problem L_1, ali ekvivalentno, problem L_1 ni nič težji od L_2.

Definicija
Naj bosta L_1 in L_2 dva odločitvena problema. Če velja $L_1 \leq_P L_2$ in $L_2 \leq_P L_1$, potem pravimo, da sta problema računsko ekvivalentna.

Trditev
Za poljubne odločitvene probleme L_1, L_2, in L_3 velja naslednje:
Če velja $L_1 \leq_P L_2$ in $L_2 \leq_P L_3$, potem velja $L_1 \leq_P L_3$ (tranzitivnost).
Če velja $L_1 \leq_P L_2$ in $L_2 \in P$, potem velja $L_1 \in P$.

Definicija
Odločitveni problem L je NP-popoln, če $L_1 \leq_P L_2$ in $L_1 \leq_P L$ za vsak $L_1 \in NP$. Razred vseh NP-popolnih problemov označujemo z NPC.
Lahko rečemo, da so NP-popolni problemi najtežji problemi v razredu NP.

2.4 Teorija števil

2.4.1 Cela števila

Definicija
Naj bosta a in b celi števili. Pravimo, da število a deli število b (ekvivalentno, a je delitelj ali faktor za b), če obstaja tako število c, da je b = ac. Če a deli b, to lahko zapišemo a | b.

Trditev (lastnosti deljivosti)
Za vse a, b, c $\in$ Z velja sledeče:

- a | a,
- če a | b in b | c, potem a | c,
- če a | b in a | c, potem a | (bx+cy) za vse x,y $\in$ Z,
- če a | b in b | a, potem a = $\pm$ b.

Definicija
Deljenje celih števil a in b, kjer b $\geq$ 1, nam natanko določi števili q in r, tako da je a = qb + r, kjer 0 $\leq$ r < b. Število q imenujemo kvocient in ga označimo z a div b. Število r imenujemo ostanek, pišemo ga a mod b.

Definicija
Celo število c je skupni delitelj a in b, če velja c | a in c | b.

Definicija
Število d je največji skupni delitelj števil a in b , pišemo d = gcd(a,b), če d | a in d | b in za vsako število c, ki deli a in b, velja c | d. Definiramo gcd(a,0) = a.

Definicija
Število d je najmanjši skupni večkratnik števil a in b, pišemo d = lcm(a,b), če a | d in b | d in za vsako število c, ki ga delita a in b, velja d | c.

Definicija
Pravimo, da sta si števili a in b tuji, če je njun največji skupni delitelj enak 1.

Definicija
Število p $\geq$ 2 imenujemo praštevilo, če sta 1 in p edina pozitivna delitelja za p. Vsa druga števila imenujemo sestavljena števila.

Trditev
Če je p praštevilo in p | ab, potem velja p | a ali p | b ali oba.

Trditev
Praštevil je neskončno mnogo.

Izrek o gostoti praštevil
Naj bo $\pi(x)$ število praštevil $\leq$ x.
Izrek o gostoti praštevil pravi, da lahko za veliko število x aproksimiramo $\pi(x)$ z x/ln x.

Trditev (osnovni teorem aritmetike)
Vsako število n ≥ 2 ima faktorizacijo, ki je produkt potenc praštevil: $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$, kjer so p_i različna praštevila in $e_i \geq 0$. Faktorizacija je enolično določena do vrstnega reda potenc natančno.

Trditev
Največji skupni delitelj števil $a = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$ in $b = p_1^{f_1} p_2^{f_2} \dots p_k^{f_k}$, kjer je $e_i \geq 0$ in $f_i \geq 0$, dobimo tako, da zmnožimo potence skupnih praštevil p_i z manjšim od eksponentov e_i in f_i. Torej

$$\gcd(a,b) = p_1^{\min\{e_1,f_1\}} p_2^{\min\{e_2,f_2\}} \dots p_k^{\min\{e_k,f_k\}}.$$

Podobno izračunamo najmanjši skupni večkratnik števil a in b, takole

$$\operatorname{lcm}(a,b) = p_1^{\max\{e_1,f_1\}} p_2^{\max\{e_2,f_2\}} \dots p_k^{\max\{e_k,f_k\}}$$

Definicija
Za celo število n ≥ 1 naj ϕ(n) označuje število celih števil na intervalu $[1,n]$, ki so tuja številu n. Funkcijo ϕ imenujemo Eulerjeva phi funkcija.

Trditev (lastnosti Eulerjeve phi funkcije):

- če je p praštevilo, potem je ϕ(p) = p-1,
- Eulerjeva funkcija je multiplikativna , tj. če velja gcd(m,n) = 1, potem je ϕ(mn) = ϕ(m) ϕ(n),
- če je $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$ faktorizacija na praštevila, potem je

$$\Phi = \prod_{i=1}^{k} (p_i - 1) \cdot p_i^{e_i - 1} = n \cdot \prod_{i=1}^{k} (1 - \frac{1}{p_i})$$

2.4.2 Algoritmi v Z

Naj bosta a in b nenegativni celi števili, obe manjši ali enaki številu n. Število bitov binarne reprezentacije števila n je $\lfloor \log_2 n \rfloor + 1$, ki ga lahko aproksimiramo z $\log_2 n$. V tabeli je predstavljena časovna zahtevnost štirih osnovnih operacij seštevanja, odštevanja, množenja in deljenja z uporabo klasičnih algoritmov. Algoritmi, ki opišejo te operacije, predpostavljajo uporabo predznačene reprezentacije števil, kjer je predznak impliciten. Časovna zahtevnost operacije je enaka oceni za število enostavnih operacij, ki jih pri danem algoritmu izvede procesor.

OPERACIJA	ZAHTEVNOST
Seštevanje $a + b$	$O(\log_2 a + \log_2 b) = O(\log_2 n)$
Odštevanje $a - b$	$O(\log_2 a + \log_2 b) = O(\log_2 n)$
Množenje $a \cdot b$	$O((\log_2 a)(\log_2 b)) = O((\log_2 n)^2)$
Deljenje $a = q \cdot b + r$	$O((\log_2 q)(\log_2 b)) = O((\log_2 n)^2)$

Največji skupni delitelj števil a in b lahko izračunamo tako, da poiščemo razcep števil a in b na potence praštevil. Vendar ta postopek ni učinkovit, saj je problem faktorizacije na

praštevila težak problem. Evklidov algoritem pa je učinkovit algoritem za izračun največjega skupnega delitelja dveh števil, ki ne potrebuje faktorizacije števil.

Evklidov algoritem za izračun največjega skupnega delitelja
Podatki: dve nenegativni celi števili a in b, a≥ b.
Rezultat : največji skupni delitelj a in b.

1. while $(b \neq 0)$ do
   r := a mod b, a := b, b := r .
2. return (a).

Evklidov algoritem ima pri podatkih a in b časovno zahtevnost :

$$O((\log_2 a)(\log_2 b)) = O((\log_2 n)^2).$$

Evklidov algoritem je moč razširiti tako, da ne vrne le največji skupni delitelj d števil a in b, ampak tudi števil x in y, ki zadoščata Diofantski enačbi ax + by = d (ali kongruenčni enačbi $ax \equiv d \pmod b$. Imenujemo ga kar razširjeni Evklidov algoritem.

Razširjen Evklidov algoritem
Podatki: dve nenegativni celi števili a in b, a≥ b.
Rezultati: d = gcd(a,b) in števili x in y, ki zadoščata enačbi ax + by = d.

1. if (b = 0) then d := a, x := 1, y := 0 ;
2. $x_0 := 1$, $y_0 := 0$, $x_1 := 0$, $y_1 := 1$;
3. while ($(b \neq 0)$ do
   3.1. q := $\lfloor a/b \rfloor$, r := a –qb, x := $x_0 – qx_1$, y := $y_0 – qy_1$;
   3.2. a := b, b := r, $x_0 := x_1$, $y_0 := y_1$, x_1 := x, y_1 := y ;
4. d := a, x := x_0, y :=y_0
5. return (d,x,y).

Razširjen Evklidov algoritem pri podatkih a in b, izračuna d = gcd(a,b) in x,y ∈ Z, rešitvi enačbe ax + by = d, v času $O((\log_2 a)(\log_2 b)) = O((\log_2 n)^2)$.

2.4.3 Števila ostankov po modulu n

Definicija
Naj bo n pozitivno celo število in a in b celi števili. Pravimo, da je a kongruenten b po modulu n in pišemo $a \equiv b \pmod n$, če n | (a –b). Število n imenujemo modul kongruence. Relacijo ≡ imenujemo kongruenca.

Trditev (lastnosti kongruence): za vse a, a_1, b, b_1, c ∈ Z velja naslednje:

- $a \equiv b \pmod n$ natanko takrat, ko imata a in b enak ostanek pri deljenju s številom n
- $a \equiv a \pmod n$ (refleksivnost)
- če $a \equiv b \pmod n$ potem je $b \equiv a \pmod n$ (simetričnost)
- če $a \equiv b \pmod n$ in $b \equiv c \pmod n$ potem je $a \equiv c \pmod n$ (tranzitivnost)
- če $a \equiv a_1 \pmod n$ in $b \equiv b_1 \pmod n$ potem velja $a + b = a_1 + b_1 \pmod n$ in $a \cdot b \equiv a_1 \cdot b_1 \pmod n$

Ker je za fiksen n relacija ≡ ekvivalenčna, razdeli cela števila Z na ekvivalenčne razrede. Ekvivalenčni razred števila a je množica vseh števil kongruentnih a po modulu n. Torej, če je

$a = q \cdot n + r$, kjer $0 \le r \le n$, potem je $a \equiv r$ (mod n). Vsako število a je kongruentno nekemu številu med 0 in n-1, ki ga imenujemo najmanjši ostanek a po modulu n. Tako sta a in r v istem ekvivalenčnem razredu, ki ga predstavlja r.

Definicija
Označimo števila ostankov po modulu n $Z_n = \{0, 1, \ldots, n-1\}$.

Definicija
Naj bo $a \in Z_n$. Multiplikativni inverz števila a po modulu n je tako število $x \in Z_n$, da velja $ax \equiv 1$ (mod n). Če obstaja tak x, je en sam in pravimo, da je a obrnljiv. Pišemo a^{-1}.

Definicija
Naj bosta $a,b \in Z_n$. Deljenje a z b po modulu n je produkt števil a in b^{-1} po modulu n in je definiran samo, če je b obrnljiv po modulu n.

Trditev
Število $a \in Z_n$ je obrnljivo natanko takrat, ko je gcd(a,n) = 1.

Trditev
Naj bo d = gcd(a,n). Kongruenčna enačba $ax \equiv b$ (mod n) ima rešitev $x \in Z_n$ natanko tedaj, ko d | b. Rešitev je enolična po modulu n/d. Število rešitev med = in n-1 je enako d.

Trditev (Kitajski izrek o ostankih (CRT))
Če so števila $n_1, n_2, \ldots, n_k$ paroma tuja, potem ima sistem kongruenčnih enačb $x \equiv a_1$ (mod n_1), …, $x \equiv ak$ (mod n_k), enolično rešitev po modulu $n = n_1\, n_2 \ldots n_k$.

Rešitev je dana z algoritmom

$$x = \sum_{i=1}^{k} a_i \cdot N_i \cdot M_I \cdot \mathrm{mod}\, n$$

kjer je$N_i = n/n_i$ in $M_i = N_i^{-1}$ mod n_i za $1 \le i \le k$. $O((\log_2 n)^2)$

Trditev
Naj bo gcd(n_1 , n_2) = 1. Potem imata kongruenci $x \equiv a$ (mod n_1) $x \equiv a$ (mod n_2) enolično rešitev $x \equiv a$ (mod n_1 n_2).

Definicija
Multiplikativna grupa množice Z_n je $Z_n^* = \{a \in Z_n | \gcd(a,n) = 1\}$. V posebnem, če je n praštevilo, potem je $Z_n^* = \{a \in Z_n | 1 \le a \le n-1\}$.

Definicija
Red grupe Z_n^* je definiran kot število elementov v Z_n^*, torej $|Z_n^*|$. Iz definicije Eulerjeve funkcije sledi, da je $|Z_n^*| = \Phi(n)$.

Trditev (Eulerjev izrek):
Naj bo celo število $n \geq 2$. Če je $a \in Z_n^*$, potem je $a^{\Phi(n)} \equiv 1 \pmod{n}$.
Če je $n \geq 2$ produkt različnih praštevil in $r \equiv s \pmod{\phi(n)}$, je $a^r \equiv a^s \pmod{n}$ za vsa števila $a \in Z_n$. Posebej, če je p praštevilo in $r \equiv s \pmod{p-1}$, potem je $a^r \equiv a^s \pmod{p}$ za vsa števila $a \in Z_n$.

Trditev (Fermatov izrek):
Naj bo p praštevilo. Če je gcd(a,p) = 1, potem je $a^{p-1} \equiv 1 \pmod{n}$.
Še več, velja $a^p \equiv a \pmod{n}$ za vsa števila $a \in Z_p$.

Definicija
Red števila $a \in Z_n^*$ je najmanjše pozitivno število t, za katerega je $a^t \equiv 1 \pmod{n}$.

Definicija
Naj ima $a \in Z_n^*$ red t. Če za število s velja $a^s \equiv 1 \pmod{n}$, potem t | s. Red števila a deli red multiplikativne grupe Z_n^*, torej t | ϕ(n).

Definicija
Če ima število $\alpha \in Z_n^*$ red ϕ(n), potem pravimo, da je α generator multiplikativne grupe Z_n^*.
Če ima Z_n^* generator, potem pravimo, da je multiplikativna grupa Z_n^* ciklična.

Trditev (lastnosti generatorjev multiplikativne grupe Z_n^*):

- Multiplikativna grupa Z_n^* ima generator natanko tedaj, ko je število n enako 2, 4, p^k ali $2p^k$, kjer je p liho praštevilo in $k \geq 1$.
- Če je α generator Z_n^*, potem je $Z_n^* = \{ \alpha^i \bmod n \mid 0 \leq i \leq \Phi(n) - 1 \}$.
- Naj bo α generator Z_n^*. Potem je red števila $\beta = \alpha^i \bmod n$ enak $\phi(n)/gdc(\phi(n),i)$
  Če je Z_n^* ciklična, potem je število generatorjev enako $\phi(\phi(n))$.
- α je generator Z_n^* natanko tedaj, ko je $\alpha^{\phi(n)/p} \equiv 1 \pmod{n}$ za vsako praštevilo p, ki deli ϕ(n).

2.4.4 Algoritmi v Z_n

Naj bo n pozitivno celo število . Kot prej, elemente Z_n predstavljajo števila {1, 2, …, n-1 }.
Operaciji, ki izračuna ostanek po modulu n, pravimo modularna redukcija po modulu n.
klasična metoda za modularno redukcijo je izračun ostanka po modulu n s celoštevilčnim deljenjem, vendar obstajajo še druge , npr. Montgomeryjeva redukcija.
Seštevanje, odštevanje in množenje v Z_n izvajamo po modulu n. Imenujemo jih modularno seštevanje, modularno odštevanje in modularno množenje. Vidimo, če sta a,b$\in$ Z_n, potem je

$$(a+b) \bmod n = \begin{cases} a+b, a+b \leq n \\ a+b-n, a+b \geq n \end{cases}$$

Zato lahko modularno seštevanje (in odštevanje) izvedemo brez zahtevnega deljenja.
Modularno množenje a in b izvedemo tako, da enostavno zmnožimo a in b kot cela števila,

potem pa rezultat reduciramo po modulu n. Inverze v Z_n lahko izračunamo z razširjenim Evklidovim algoritmom. V nadaljevanju bomo poleg naštetih operacij opisali še modularno eksponiranje. V tabeli je predstavljena računska zahtevnost osnovnih operacij v Z_n.

OPERACIJA	ZAHTEVNOST
Modularno seštevanje (a+b) mod n	$O(\log_2 n)$
Modularno odštevanje (a-b) mod n	$O(\log_2 n)$
Modularno množenje (a b) mod n	$O((\log_2 n)^2)$
Modularno invertiranje a^{-1} mod n	$O((\log_2 n)^2)$
Modularno eksponiranje a^k mod n, k<n	$O((\log_2 n)^3)$

Izračun multiplikativnih inverzov v Z_n
Podatki: $a \in Z_n$.
Rezultat: a^{-1} mod n, če le-ta obstaja.
1.uporabi razširjen Evklidov algoritem in poišči števili x in y, rešitvi enačbe ax+ny =d, kjer je d = gcd(a,n).
2.if (d >1) then ″a^{-1} mod n ne obstaja″ else return(x).

Časovna zahtevnost algoritma je $O((\log_2 n)^2)$ saj uporabimo le razširjeni Evklidov algoritem.

Modularno eksponiranje
Modularno eksponiranje lahko učinkovito izvedemo z uporabo metode kvadriraj-in-množi, ki je pomembna za več kriptografskih sistemov. Ena verzija te metode je algoritem 16. Temelji na naslednji enakosti. Naj bo binarna reprezentacija za k enaka $\sum_{i=0}^{t-1} k_i \cdot 2^i$, kjer $k_i \in \{0,1\}$.
Potem je

$$a^k = \prod_{i=0}^{t} a^{k_i 2^i} = (a^{2^0})^{k_0} (a^{2^1})^{k_1} \ldots (a^{2^t})^{k_t}$$

Algoritem kvadriraj-in-množi za eksponiranje v Z_n.
Podatki: $a \in Z_n$ in število k, ki ima binarno reprezentacijo $\sum_{i=0}^{t-1} k_i \cdot 2^i$.
Rezultat: b := a^k mod n.

1. b := 1, if (k = 0) then return (b);
2. A := a ;
3. if (k_0 = 1) then b := a ;
4. for i := 1 to (t-1) do
   4.1. A := A^2 mod n ;
   4.2. if (k_i = 1) then b := A b mod n ;
5. return (b),

2.5 Abstraktna algebra

Definicija
Binarna operacija (dvočlena operacija) na množici S je funkcija oblike f: S x S→S.
Binarne operacije ponavadi zapišemo z vsajenim zapisom kot je $a + b$, $a \cdot b$, $a * b$ ali $a \times b$ in ne s funkcijskim zapisom oblike $f(a, b)$. Včasih jih zapišemo tudi poleg $a\ b$.

2.5.1 Grupa

Definicija
Grupa G = {a, b, ..} je par (G,*), kjer je G neprazna množica in * asociativna binarna operacija na G: G x G →G, ki jo imenujemo operacija grupe in ki vsakemu urejenemu paru (a, b) ∈G priredi natanko en element a * b ∈ G. Operacija * mora zadoščati pogojem – aksiomom grupe.

- Za vsak a,b ∈ G, velja a * b ∈ G. Zakon o zaprtosti.
- Za vsak a, b, c ∈ G, velja (a * b) * c = a * (b * c). Zakon o asociativnosti. (zakon o združevanju faktorjev)
- V G obstaja takšen element e, za katerega za vsak a ∈ G velja e * a = a * e = a. Zakon o identiteti.
- Za vsak a ∈ G obstaja takšen element b ∈ G, za katerega velja a * b = b * a = e. Zakon o inverzu.

V abstraktni algebri je Abelova grupa takšna grupa (G, *) , ki je tudi komutativna, se pravi, v kateri enakost a * b = b * a velja za poljuben element a in b iz G. Če je grupa Abelova, operacijo navadno pišemo kot + namesto *, nevtralni element kot 0 in inverz elementa a kot – a.

Definicija
Grupa G je končna, če ima končno število elementov. V tem primeru je moč grupe označena z $|G|$ in je enaka številu elementov grupe.

Primera
Naj je Z množica celih števil {…, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, ….} in naj znak + kaže na operacijo seštevanja. Potem je par (Z, +) grupa zapisana aditivno.
Dokaz:

1. Če sta a in b celi števili, potem je a + b celo število.
2. Če so a,b, c cela števila, potem velja (a + b) + c = a + (b + c) .
3. Število 0 je celo število. Za vsako celo število a velja 0 + a = a + 0 = a.
4. Če je a celo število, potem obstaja celo število b = -a, za katerega velja a + b = b + a = 0

Imamo množico celih Z in operacijo množenja, označeno z . Ali je par (Z,*) grupa?

1. Če sta a in b celi števili, potem je a * b celo število.
2. Če so a,b, c cela števila, potem velja (a * b) * c = a * (b * c) .
3. Število 1 je celo število. Za vsako celo število a velja 1 * a = a * 1 = a.
4. **Toda** če je a celo število, potem ni nujno, da obstaja celo število b, za katerega velja a * b = b * a = 1.

Vidimo, da nima vsak element iz para (Z, *) obratnega elementa in zaradi tega (Z, *) ni grupa.

2.5.2 Kolobarji

Definicija
Kolobar (R, +, *) vsebuje množico R z dvema binarnima operacijama seštevanje in množenje na R, ki mora zadoščati naslednjim aksiomom:

- (R, +) je abelova grupa z identitetnim elementom 0
- Operacija * je asociativna (a * (b * c) = (a * b) * c za vse a, b, c $\in$ R)
- Množilna identiteta je 1 (1 * a = a * 1 = a za vse a $\in$ R.
- Za operaciji + in * velja zakon distributivnosti a*(b+c) = (a*b) + (a*c).

Kolobar je komutativen, če velja a * b = b * a za vse a, b $\in$ R.

Primera
Množica celih števil z operacijama seštevanja in množenja (Z, +, *) je komutativni kolobar z enoto, ni pa obseg, saj v splošnem nimamo inverza za množenje.

Definicija
Element a kolobarja R se imenuje enota ali obrnljiv element, če ima element b $\in$ R, tako da velja a * b =1.

2.5.3 Obsegi

Definicija
Obseg je komutativni kolobar, v katerem imajo vsi neničelni elementi multiplikativni inverz.

3 Zaključek

V seminarski nalogi sem spoznal osnove pojme kriptologije in osnovna matematična orodja, ki se uporabljajo v kriptologiji.
V poglavju o matematičnih orodjih sem predstavil le osnovne pojme, ki bi koristili pri nadaljnem študiju kriptologije
Pri pripravi seminarske naloge sem spoznal, da je področje kriptologije kompleksno, zahtevno in zelo obsežno.

4 Seznam uporabljenih virov

A. Menezes, P.van Oorschot and S. Vanstone: Handbook of Applied Cryptography,
Sašo Tomažič: Varne komunikacije preko interneta,
Sašo Tomažič: Varnost v telekomunikacijah in kako jo zagotoviti,
Andrej Dobnikar: Osnove teorije informacije (knjiga v elektronski obliki 9.2.2006),
Milan Hladnik: Verjetnost in statistika, FERI 2002,
Internetna stran http://www.lkn.fe.uni-lj.si/vaje/KK/,
Internetna stran http://www.ca.gov.si/kripto/index.htm,
Tone Vidmar, Informacijsko komunikacijski sistem, Pasadena 2002