Seminar podiplomskega študija pri predmetu

PORAZDELJENI INFORMACIJSKI SISTEMI IN CELOVITOST PODATKOV

KRIPTOANALIZA

Eva Stergaršek

maj 2006

Kazalo vsebine

Kazalo slik

Kazalo tabel

1 Uvod

Izvor besede kriptoanaliza gre verjetno iskati v grščini. Grška beseda *kryptós* pomeni skrito, beseda *analýein* pa sprostiti ali odvezati.

Kriptoanaliza je študija metod za pridobivanje pomena kodiranih informacij brez dostopa do skrivne informacije, ki bi jo v normalnem primeru potrebovali. To navadno pomeni iskanje skrivnega ključa.

Kriptoanaliza označuje tudi poskuse zaobiti varnosti drugih tipov šifrirnih algoritmov in protokolov na splošno. Kakor koli že, kriptoanaliza normalno ne vključuje ostalih napadov, ki ne izkoriščajo slabosti kriptografije. To so lahko podkupovanje, fizična prisila, vlom, napad s strani in podobno. Le-ti imajo pri računalniški varnosti pomembno vlogo, saj postajajo bolj učinkoviti kot tradicionalna kriptoanaliza.

Kljub temu, da cilji ostajajo isti, so se metode in tehnike v kriptoanalizi skozi zgodovino presunljivo spreminjale. Prilagajale so se vse večji kriptografski kompleksnosti: od metod preteklosti svinčnik-in-papir, preko naprav kot je bila na primer Enigma v 2. svetovni vojni, do računalniško osnovanih shem sedanjosti.

1.1 Šifrirni postopki

Pri šifrirnem postopku pride do preslikave razumljivega sporočila (čistopis) v nerazumljivo sporočilo (šifropis). Dešifriranje kot obratna preslikava je mogoča le, če poznamo ključ, torej neko dodatno skrivno informacijo. Cilj vsakega šifriranja je sistem, kjer za pridobivanje čistopisa iz šifropisa potrebujemo ključ. Vdiralec brez ključa naj ne bi mogel, čeprav ima popolno znanje o uporabljenih algoritmih in ostale informacije, priti do čistopisa.

Klasične šifrirni postopke po načinu zakrivanja ločimo na:

- substitucijske metode zakrivanja: vsak znak ali skupino znakov nadomestimo z drugimi. Primer: Cezarjeva šifra;
- transpozicijske metode zakrivanja: spreminjamo vrstni red znakov.

Glede na vrsto ključev ločimo dve vrsti šifrirnih postopkov:

- simetrične in
- asimetrične šifrirne postopke.

1.1.1 Simetrični šifrirni postopki

Pri simetričnih postopkih šifriramo in dešifriramo z istim ključem, zato morajo udeleženci pred začetkom komunikacije ta ključ na nek varen način izmenjati. Postopke podrobneje delimo na:

- pretočne šifrirne postopke (ang. stream cipher) in
- bločne šifrirne postopke (ang. block cipher).

Pretočni postopki delujejo na principu sprotnega šifriranja sporočila, zato operacije izvajamo nad posameznimi biti. Sprotno šifriranje, ki poteka nad posameznimi biti sporočila in ključ, zahteva enako dolžino ključa in sporočila.

Pri bločnih postopkih sporočilo razdelimo na krajše bloke, ki jih zaporedno šifriramo, tako da uporabimo zaporedje zamenjav (ang. substitution) in premikov (ang. transposition), ki jih določa ključ.

1.1.2 Asimetrični šifrirni postopki

Pri asimetričnih postopkih uporabljamo dva ključa. S šifrirnim ključem lahko sporočilo šifriramo, ne moremo ga pa dešifrirati. Za to potrebujemo njegov par – dešifrirni ključ. Šifrirni ključ sporočila ne more dešifrirati, zato ni potrebe, da bi bil tajen. Imenujemo ga javni ključ. Dešifrirni ključ, v nasportju s tem, mora ostati tajen. Postopek šifriranja izvedemo s šifrirnim - javnim ključem, zato ni potrebe po predhodni varni izmenjavi ključev.

Ideja asimetričnih postopkov je relativno preprosto izvedljivo šifriranje in - brez dešifrirnega ključa - praktično neizvedljivo (matematično zelo zahtevno) dešifriranje.

Razloček med javnimi in simetričnimi ključi je na prvi pogled malenkosten, izkaže pa se, da z javnim ključem dosežemo stvari, ki s simetričnim ključem niso mogoče. Vse kar je mogoče s simetričnim ključem je mogoče tudi z javnim. Razlog, zakaj so simetrični ključi še v uporabi, pa je hitrost.

Obstajajo tri kategorije šifrirnih postopkov: simetrični šifrirni postopki (ang. symmetric ciphers), šifriranje z javnim ključem in zgostitvene (ang. hash) funkcije.

1.1.3 Zgoščevalne funkcije

Kriptografska zgoščevalna funkcija je zgoščevalna funkcija (ang. Hash function) z dodanimi varnostnimi lastnostmi, zaradi česar je uporabna kot primitiv v različnih aplikacijah informacijske varnosti, na primer pri avtentikaciji in integriteti sporočila. Funkcija vhodni niz poljubne dolžine spremeni v izhodni niz določene stalne dolžine, ki ga imenujemo digitalni prstni odtis ali angleško message digest.

1.2 Varnost šifrirnih postopkov

Da je šifrirni postopek varen, mora zagotavljati odpornost proti različnim kriptografskim napadom. Napadalec lahko napada na različnih nivojih: nivo sporočila, ključa ali postopka.

Poskuša lahko dešifrirati določeno sporočilo in tako priti do določenih informacij. Če uspe odkriti tajni ključ, lahko dešifrira vsa sporočila, ki so bila šifrirana s tem ključem. Če odkrije šibko točko šifrirnega postopka, pa lahko dešifrira vsa sporočila, ki so bila šifrirana s tem postopkom.

Perfektna (brezpogojna) varnost

Napadalec naj bi imel neomejene izračunske vire (ang. Computational resources), vprašanje pa je zadostna količina dostopnih podatkov. Preučevanje šifropisov naj ne bi ponujalo informacij napadalcu.

Nujen pogoj za perfektno varnost pri simetrični shemi kodiranja je. Da je ključ vsaj tako dolg kot sporočilo. Primer brezpogojno varnega algoritma je postopek z enkratno uporabo ključa (ang. one time key pad).

Ključ je popolnoma naključen in enak dolžini ključa. Na osnovi znanega para čistopis - šifropis je pri tem postopku preprosto priti do čistopisa, kar pa napadalcu ne pomaga, saj se vsak ključ uporabi natančno enkrat. Postopek je žal nepraktičen zaradi treh zahtev: velika dolžina ključa, zahteve, da je popolnoma naključen in da se vedno uporabi nov ključ. Uporablja se le izjemoma za varovanje najstrožje zaupnih sporočil.

Dokazljiva varnost

Šifrirna metoda je dokazljivo varna, če jo je v osnovi tako težko razbiti kot dobro poznan in domnevno težak problem (razcep na prafaktorje ali izračun diskretnega logaritma).

Praktična varnost

Praktična varnost ocenjuje količino izračunske moči z najboljšimi trenutno poznanimi metodami. Tehnika naj bi bila praktično varna, če nivo potrebnih izračunov (z najboljšim poznanim napadom) presega izračunske vire hipotetičnega napadalca.

Priložnostna varnost

Priložnostno (Ad hoc) varni postopki, pri katerih je verjetnost, da bodo razbiti v omejenem času z omejenimi sredstvi, dovolj majhna.

1.3 Kerckhoffov zakon

Že leta 1883 je nizozemski kriptolog Auguste Kerckhoffs pod predpostavko 'Nasprotnik pozna sistem šifriranja' definiral za resno kriptografijo osnovno pravilo: Varnost sistema šifriranja ne sme biti odvisna od načina šifriranja. Za tajnost je edina garancija tajnost ključa.

Pri načrtovanju sistemov varnosti je pametno predvidevati, da so podrobnosti kritpografskega algoritma napadalcu že poznane - samo tajnost ključa zagotavlja varnost.

Zgodovina kriptografije je dokazala, da je ohranjanje tajnosti podrobnosti široko uporabljanega algoritma lahko zelo težko. V primerjavi s celotnim kodirnim algoritmom je ključ je lažje varovati, saj je navadno le majhen del informacije, in lažje spremeniti, če pride do zlorabe. Zato velja, da varnost nekega kodirnega sistema sloni na tajnosti nekaterih ključev.

1.4 Ocenjevanje uspeha pri kriptoanalizi

Uspeh pri kriptografiji ni enoznačno določen. Poznamo več vrst uspehov:

- popolno razbitje (ang. Total break): napadalec pridobi skrivni ključ;
- globalna dedukcija: napadalec odkrije funkcionalno enakovreden algoritem za kodiranje in dekodiranje, ne izve pa ključa;
- lokalna dedukcija: napadalec odkrije dodatne čistopise (ali šifropise);
- informacijska dedukcija: napadalec pridobi nekaj novih informacij;
- algoritem prepoznave (ang. Distinguishing algoritm): napadalec lahko izloči šifro iz naključne permutacije.

1.5 Zgodovina kriptoanalize

Kriptoanaliza in kriptografija sta se razvijali druga ob drugi. Novi šifrirni postopki so zamenjali stare razbite, razvite so bile nove metode kriptoanalize, ki so razbile izboljšane sheme. Praktično gledano sta kriptoanaliza in kriptografija kot dve strani istega kovanca. Za ustvarjanje varne kriptografije je potrebno načrtovati proti mogoči kriptoanalizi.

Čeprav je dejanska beseda 'kriptoanaliza' relativno nova (William Friedman, 1920), so metode razbijanja kodirnih in šifrirnih postopkov precej starejše. Prva znana zapisana razlaga kriptoanalize sega v 9. stoletje in med drugim opisuje metodo frekvenčne analize.

Dandanes je moderna kriptografija visoko presegla kriptoanalizo. Po mnenju Davida Kahna je kriptoanaliza na nek način mrtva, saj obstaja veliko

sistemov, ki jih ni mogoče zlomiti. Po drugi strani pa ne smemo pozabiti na druge učinkovite metode: prestrezanje, uporaba fizične sile, bočni napadi, kvantni računalniki itd.
Trditev je mogoče preuranjena: šibke šifre še niso povsem izkoreninjene. V akademskih sferah so vsak dan predstavljeni novi načrti, ki pa so pogosto razbiti. Blokovno šifriranje Madryga (1984) so razbili z napadom na osnovi enega ali več šifropisov (COA) leta 1998. Nadomestilo za DES, FEAL-4, je bilo uničeno z več napadi, večina je bila popolnoma praktičnih. Tudi v gospodarstvu šifre niso brez napak. Algoritmi A5/1, A5/2 in CMEA, ki so uporabljeni pri mobilni telefoniji, so lahko razbiti v urah, minutah in z zmogljivo opremo celo v realnem času. Leto 2001 je dokazalo dovzetnost protokola WEP, ki varuje brezžična omrežja Wi-Fi, na praktični napad s pripadajočim ključem.

Uspešna kriptoanaliza je nedvomno zaznamovala zgodovino. Možnost prebiranja misli in načrtov, ki naj bi bili tajni, je velika prednost, ki do svojega izraza najbolj pride med vojno. Med 1. svetovno vojno je razbitje Zimmermannovega telegrama Združene države popeljalo v vojno. Podobno je med 2. svetovno vojno kriptoanaliza nemških šifrirnih postopkov vplivala na dejanski konec vojne in morda celo odločila izid.

1.5.1 Enigma

Enigma je električna naprava za šifriranje sporočil. Uporabljale so jo nemške oborožene sile med 2. svetovno vojno. Beseda izhaja iz grščine in pomeni uganko.

V času do konca vojne 1945 in še kasneje so bili v uporabi številni različni modeli Enigme. Najbolj razširjena je bila ENIGMA I, ki so jo od leta 1930 uporabljale nemške oborožene sile (nem. Reichswehr). Enigma je bila med 2. svetovno vojno najbolj uporabljan sistem šifriranja sporočil. Na prvi pogled videti kot pisalni stroj, ki ga sestavljajo tipkovnica, zbirka zamenljivih valjev in polja z lučkami za prikaz. Valji so bistveni za šifriranje. Vsak valj ima na obeh straneh po 26 električnih spojev, ki so na tajen način povezani med seboj. Važno in za šifriranje izredno pomebno je, da se za razliko od enoabecedne zamenjave, kjer se ista črka odprtega besedila vedno prevede v isto črko šifriranega besedila, pri ENIGMI po vsaki vneseni črki spremeni sistem zamenjave (poligrafska substitucija).

Število možnih šifer za standardno ENIGMO I z izbranimi tremi vrtljivimi valji (izmed petih), enim od dveh povratnih valjev, uporabi desetih povezav se lahko izračuna kot produkt 120 položajev vrtljivih valjev, 676 nastavitev obročev, 17.576 osnovnih nastavitev in 150.738.274.937.250 možnih povezav, kar je: 120·676·17.576· 150.738.274.937.250 = 214.917.374.654.501.238.720.000, približno $2{,}149 \cdot 10^{23}$ možnosti (77 bitov).

Število možnih šifer je izredno veliko in se lahko primerja s sodobnimi sistemi kriptografije. Sistem šifriranja *DES* ima velikost števila ključev samo 56 bitov, njegov naslednik AES (Rijndael) pa praviloma 128 bitov in velja za popolnoma varnega.

Velikost ključa in s tem število možnih ključev je nujno potreben pogoj za varnost sistema šifriranja, ni pa zadosten.

Uporabniki ENIGME so bili prepričani, da se njihovo strojno izdelano šifriranje besedil ne da zlomiti z ročnimi metodami, kar je bilo možno za praktično vse sisteme šifriranja do leta 1918. Kar so spregledali, je bilo, da se strojno šifriranje lahko poskusi zlomiti s strojnimi metodami.

Skupina poljskih matematikov, zbranih okrog Mariana Rejewskega, je že pred drugo svetovno vojno dosegla velike uspehe pri dešifriranju besedil. Rejewski je uporabil zakone teorije permutacij, jezikovna pravila nemščine in odstopanje od pravil slučajnosti, ki so jih povzročili uporabniki. Tako ljudje radi za ključ sporočila uporabljajo kombinacije kot AAA, BBB, ABC…Dejstvo, da so Nemci poslali na začetku vsakega sporočila, tako imenovani ključ sporočila. Ta je bil zašifriran z dnevnim ključem in je bil na začetku sporočila poslan dvakrat. Vseboval je začetno nastavitev vrtljivih valjev in to sporočilo je bilo namenjeno zmanjšanju napak pri sprejemanju/oddajanju sporočil. Sprejemnik sporočila je najprej dešifriral začetnih 6 črk sporočila z dnevnim ključem in nato postavil vrtljive valje v tako določen začetni položaj in dešifriral dejansko sporočilo. Rejewski je razpolagal tudi s kodnimi knjigami Nemcev, navodili za uporabo in drugimi materiali. Nemci so tudi delali napake, ko so na zahtevo po pošiljanju testnega signala oddali na primer 50 krat črko A, kar je bilo enostavno dešifrirati. Tako je bil napor za »lomljenje« šifer bistveno zmanjšan.

S pomočjo elektromehanskih naprav, je bilo možno v nekaj urah odkriti dnevni ključ, ki se je uporabljal za nastavitve valjev in so ga Nemci menjali vsak dan ob polnoči. Leta 1939 so Nemci izboljšali sestavo ENIGME, tako da so namesto treh uporabljali pet vrtečih valjev, od katerih so bili istočasno v uporabi vedno le trije. Prav tako so uvedli povezavo vtičnice za 10 parov namesto dotedanjih štirih parov. Tako nastala komplikacijo so Poljaki rešili z uporabo 60 bomb.

Britanski kriptoanalitiki so delali v Bletchley Parku. Projekt je potekal pod kodnim imenom Ultra. Nadaljevali so tam, kjer je moral nehati Rejewski in zlomili tudi šifro leta 1939 izboljšane ENIGME. Pri delu jim je pomagala površnost nemških šifrantov: ponavljajoči in slabo izbrani ključi sporočil, standardna oblika vremenskih podatkov in podatkov o legi.

Kriptografska zaščita ENIGME je bila osnovana na tajnosti žičnih povezav v valjih, kar je v nasprotju s Kerckhoffsovim načelom. S tem so postale te povezave del algoritma in ne ključa. Pomebno je, da se te povezave od leta 1920 do konca uporabe leta 1945 niso spremenile. Pri normalni uporabi tako razširjene naprave ne bi smeli domnevati, da bo tako važen del naprave ostal tajen, kljub temu da so si Nemci za to zelo prizadevali.

2 Napadi v kriptoanalizi

Cilj kriptoanalize je poiskati slabosti ali razpoke v varnosti kriptografskih shem. Izvajalci kriptoanalize so bodisi sovražni napadalci, ki si želijo sistem podrediti, bodisi načrtovalci (ali drugi), ki želijo zgolj oceno varnosti sistema.

Vdiralce delimo na pasivne vdiralce (prisluškovalce) in na aktivne vdiralce (ponarejevalce). Pasivni vdiralec mora zbati sporočilo zgolj dekriptirati, aktivni pa mora poleg tega obvladati tudi enkripcijo.

Pasivni napadalec le opazuje komunikacijski kanal, zato ogroža zaupnost podatkov. Aktivni napadalec pa poskuša zbrisati, dodati in na kakršenkoli način spremeniti prenos na kanalu, zato poleg zaupnosti ogroža tudi celovitost in overitev podatkov.

Napadi v kriptoanalizi se razlikujejo glede na to, kaj ima napadalec ob napadu pri roki. Model napada na osnovi enega ali več šifropisov (COA) vključuje napadalca in nekaj šifropisov. Napad na osnovi para čistopis-šifropis (KPA) je napad s pomočjo čistopisa in njegove kodirane verzije, šifropisa. Napad na osnovi (CPA) je model kriptoanalitičnega napada, ki predpostavlja, da ima napadalec možnost izbrati poljuben čistopis in nato pridobiti njegovo kodirano različico – šifropis. Pri napadu s pomočjo izbranega šifropisa (CCA) si napadalec izbere šifropis in povzroči, da je dešifriran z neznanim ključem, izbori si dostop do naprave za dešifriranje. Napadalec, ki lahko dešifrira izbrana sporočila, ogrozi zaupnost sistema in v skrajnem primeru z izdajanjem skrbno izbranih šifropisov in analiziranjem dešifriranih rezultatov pridobi skrivni ključ. Napad s pripadajočim ključem (ang. Related-key attack) je oblika kriptoanalize, kjer napadalec opazuje delovanje šifrirnega postopka z več različnimi ključi. Vrednosti ključev so napadalcu na začetku nepoznane, pozna pa neko matematično povezavo, ki jih povezuje.

2.1 Napad na osnovi enega ali več šifropisov (COA)

Model napada sestavlja napadalec in nekaj šifropisov. Napad je popolnoma uspešen, če je iz šifropisa mogoče izpeljati čistopis ali, kar je še boljše, ključ. Za uspešno dejanje velja tudi pridobitev nekaj informacij o osnovnem čistopisu. Primer: nasprotnik za namene varovanja prometnega toka (ang. Traffic-low securtiy) nenehno pošilja šifrirana sporočila. Zelo uporabno bi bilo torej ugotoviti, kaj so resnična sporočila in kaj so nule. Že samo informativna ocena bi bistveno olajšala analizo prometa (ang. Traffic analysis).

V zgodovini kriptografije so bili zgodnji šifrirni postopki rutinsko razbiti z uporabo šifropisov. Kriptografi so razvili statistične tehnike za napad na šifropis, na primer frekvenčno analizo. Mehanske kodirne naprave, na primer

Enigma, so bile trši oreh, čeprav je bila slednja razbita s pomočjo šifropisov in ne-varnega protokola za vzpostavitev nastavitve sporočila.

Vsaka modernejša šifra poskuša vpeljati zaščito pred napadi COA. Proces vpeljevanja načrta za novo šifro traja več let in je podvržen preizkušanju obsežnih količin šifropisov. Kljub vsemu se slaba uporaba šifer ali naslanjanje na algoritme, ki niso bili podvrženi temeljitemu pregledu, običajno konča pri kodirnih sistemih, ki so predmet napada na osnovi šifropisa.

2.1.1 Primeri uspešnih napadov

Šifrirni postopek s premajhnim ključem je lahko predmet napadov surove sile (ang. Brute force) s pomočjo šifropisov in preizkušanja vseh mogočih ključev. Potrebno je le ločiti veljaven čistopis od naključnega šuma. Primer je DES, ki ima le 56-bitni ključ.

Vse preveč pogosti so komercialni varnostni izdelki, ki izpeljujejo ključe, za drugače trdne šifrirne postope kot AES, iz uporabniško izbranih gesel. Ker uporabniki redko izberejo geslo vsaj približno blizu entropije območja ključa (ang. Key space), so takšni sistemi pogosto preprosto razbiti s pomočjo le širfopisov.

Nekateri moderni šifrirni postopki so ranljivi na napade COA. Primer je Akelarre, 128-bitni bločni šifrirni postopek, predlagan leta 1996, ki je bil podvržen napadom na osnovi šifropisov leta 2000.

RC4

Zgodnja verzija Microsoftove programske opreme PPTP za navidezna privatna omrežja (VPN) je uporabljala isti ključ RC4 (ARCFOUR) za pošiljatelja in prejemnika. V vsakem primeru, ko je tak pretočni šifrirni postopek z istim ključem uporabljen dvakrat, je pot napadom COA odprta.

RC4 je najbolj uporabljan softverski pretočni šifrirni postopek, ki je vključen pri popularnih protokolih kot so: protokol SSL (za varovanje internetnega prometa) in protokol WEP (za varovanje brezžičnih omrežij).

Čeprav je poseben zaradi svoje preprostosti, ne zadosti visokim zahtevam za varnost, ki so jih postavili kriptografi. Posledica tega je, da nekateri načini uporabe RC4 vodijo do zelo ne-varnih sistemov. Ni priporočljiv za uporabo v novih sistemih, čeprav so nekateri sistemi, osnovani na RC4, dovolj varni za praktično uporabo.

WEP

Za protokol WEP, prvi varnostni protokol za brezžično vernost (Wi-Fi), je bilo dokazano, da je ranljiv na napade, večina je bila na onovi šifropisov. Ker brezžično omrežje oddaja sporočila s pomočjo radia, je zelo dovzetno na

prisluškovanje (ang. Eavesdropping). S pomočjo kriptoanalize je bilo odkritih več pomembnih slabosti, zato je WEP leta 2003 zamenjal standard WPA in leta 2004 standard IEEE 802.11i (WPA2).

Kodiranje uporablja algoritem RC4, pretočni šifrirni postopek, pri katerem je bistvenega pomena, da se isti ključ nikoli ne ponovi. Da do tega ne pride, WEP v vsako sporočilo vključi 24-bitni vektor IV. Ključe je potrebno menjavati ročno in to se tipično dogaja neredno. Napadalec lahko sklepa, da so vsi ključi, uporabljeni za kodiranje paketov, povezani s poznanim IV. Ta predpostavka je WEP odprla seriji napadov, tudi napad s pripadajočim ključem (ang. Related key attack).

Veliko sistemov WEP zahteva ključ v heksadecimalni obliki. Nekateri uporabniki si izberejo ključe, ki so oblikovani z omejenim heksa naborom 0-9, A-F. Takšne ključe je lahko uganiti.

2.2 Napad na osnovi poznanega čistopisa (KPA)

Napad na osnovi para čistopis-šifropis (KPA) je napad s pomočjo čistopisa in njegove kodirane verzije, šifropisa. Napadalec lahko oba uporabi za pridobivanje nadaljnih informacij, tipično je to ključ.

Med drugo svetovno vojno so potekali poskusi na Bletchey Park, lokaciji delovanja razbijanja kod, v Angliji. Izvajal se je pritisk na Nemce, da bi proizvedli sporočila, za katere so poznali čistopis. Poznani čistopisi so se imenovali *cribs*, poskusi prisile Nemcev, da bi jih proizvedli, pa *gardening*.

2.2.1 Primeri uspešnih napadov

Na ta tip napada so zelo ranljivi kodirani arhivi datotek (primer ZIP). Napadalec s kodirano datoteko ZIP za napad potrebuje le eno nekodirano datoteko iz arhiva, ki oblikuje poznan čistopis ali prvi del para čistopis-šifropis. Z uporabo javno dostopne programske opreme je nato hitro mogoče izračunati ključ in dekodirati celotno sporočilo.

Klasični šifrirni postopki so tipično ranljivi na napad na osnovi para čistopis-šifropis. Primer je cezarjeva šifra (ang. Caesar cipher). Če napadalec pozna eno črko pripadajočega čistopisa in šifropis, lahko popolnoma dekodira sporočilo.

2.3 Napad s pomočjo izbranega čistopisa (CPA)

Napad na osnovi (CPA) je model kriptoanalitičnega napada, ki predpostavlja, da ima napadalec možnost izbrati poljuben čistopis in nato pridobiti njegovo kodirano različico – šifropis. Cilj napada CPA je zbrati informacije, kar

zmanjša varnost kodirne sheme. V najslabšem primeru lahko izbrani čistopis razkrije skrivni ključ.

Na prvi pogled je to precej nerealističen model napada. Precej neverjetno je, da bi napadalec lahko prepričal kriptografa naj zanj kodira velike obsege čistopisov, ki si jih je poprej sam izbral. Res pa je, da je moderna kriptografija vgrajena v programsko in strojno opremo in prikladna za široko paleto aplikacij. Velikokrat je napad CPA kaj lahko izvedljiv. Napad CPA je pomemben v okviru kriptografije z javnim ključem, kjer je kodirni ključ javni, zato lahko napadalec kodira katerikoli šifropis želi.

- **Serijski** CPA (ang. Batch). Napadalec izbere vse čistopise preden so kodirani.
- **Prilagodljivi** CPA (ang. Adaptive)**.** Napadalec izvede serijo interaktivnih poizvedb, kjer izbira čistopise na podlagi informacij iz predhodnih poizvedb.

2.4 Napad s pomočjo izbranega šifropisa (CCA)

Pri napadu s pomočjo izbranega šifropisa si napadalec izbere šifropis in povzroči, da je dešifriran z neznanim ključem – izbori si dostop do naprave za dešifriranje. Naprava, ki omogoča dešifriranje izbranih šifropisov se splošno imenuje deširfrirni orakelj (ang. Decryption orcle). Napadalec, ki lahko dešifrira izbrana sporočila, lahko ogrozi zaupnost sistema, posledice pa so lahko hujše. V skrajnem primeru lahko napadalec z izdajanjem skrbno izbranih šifropisov in analiziranjem dešifriranih rezultatov pridobi skrivni ključ. Uspešno izveden napad CCA lahko slabi varnost sheme tudi ko dešifrirni orakelj ni več na voljo.

Če je kriptosistem ranljiv na napade te vrste, se morajo razvijalci izogibati situacij, kjer bi napadalec lahko dešifriral izbrane šifropise. To je lahko težje kot se zdi, saj tudi delno izbrani šifropisi ponujajo dovolj možnosti za subtilne napade. Še več, nekateri šifrirni sistemi (na primer RSA) uporabljajo isti mehnizem za podpisovanje in dekodiranje sporočil. To omogoča napade, če na sporočilu, ki bo podpisano, ni bilo opravljeno zgoščevanje.

Zaščita pred napadi te vrste je šifrirna shema, ki omejuje zmožnost spreminjanja šifropisa. Izmed vseh predlaganih shem je za RSA najbolj pogost standard OAEP. OAEP velja za varnega v okviru *random oracle* modela. To je matematična abstrakcija, ki se uporablja pri kriptografskih dokazih. Tipično nastopajo pri dokazih, kjer ne obstaja vgradljiva funkcija, ki bi zagotavljala zadostne matematične zmožnosti in bi zadostila dokazu varnosti. Dokazi, ki se poslužujejo te abstrakcije, so varni v okviru modela.

Proti prilagodljivemu napadu na osnovi izbranega šifropisa je varen vsak kriptografski sistem z zavedanjem čistopisa (ang. Plaintext aware). Zavedanje čistopisa je del varnosti šifriranja z javnim ključem. Šifrirni sistem se zaveda čistopisa, če je za zmogljiv algoritem težko izvedljivo, da pridobi veljaven šifropis brez zavedanja pripadajočega šifropisa.

2.4.1 Neprilagodljivi napad CCA

CCA1 je poznan tudi kot indiferentni napad s pomočjo izbranega šifropisa. Napadalec ima dostop do dešifrirnega oraklja preden izbere šifropis, s katerim napada. Cilj napada je pridobiti čimveč informacij in oslabiti sistem s široko paleto ciljnih šifropisov. V najboljšem primeru lahko napad razkrije skrivni dešifrirni ključ in popolnoma razbije shemo.

2.4.2 Prilagodljivi napad CCA

CCA2 razširja prejšnji napad. Napadalec uporablja orakelj po tem, ko je izbral širopis napada. To je interaktivna oblika CCA, kjer napadalec pošlje kopiso šifropisov, ki so dešifrirani, nato pa izbira šifropise pise na podlagi informacij iz predhodnih poizvedb.

Napadi so izvedeni na celo kopico shem z javnimi ključi, tudi RSA. Mogoči so le, če so šifropisi lahko spremenjeni na posebne načine, ki bodo imeli predvidljiv učinek na dešifriranje tega sporočila. Preprečiti jih je mogoče s pravilno uporabo kriptografskega bitnega zapolnjevanja (ang. padding) ali preverjanjem redundance.

Prilagodljivi napadi na osnovi izbranega šifropisa so veljali za teoretičen problem vse do leta 1998, ko je Daniel Bleichenbacher iz Bell Laboratories demonstriral praktičen napad na sistem, ki je uporabljal kodiranje RSA in kodirno funkcijo PKCS #1 v1, vsebujoč verzijo protokola SSL. Napad je izkoristil slabosti funkcije PKCS da je po stopnjah razkril vsebino z RSA kodiranega sporočila. Potrebno je bilo poslati več milijonov testnih šifropisov dekodirni napravi (z SSL opremljeni spletni strežnik). Praktično to pomeni, da je sejni ključ SSL lahko razkrit v razumnem časovnem obdobju, dnevu ali manj.

2.5 Napad s pripadajočim ključem

Napad s pripadajočim ključem (ang. Related-key attack) je oblika kriptoanalize, kjer napadalec opazuje delovanje šifrirnega postopka z več različnimi ključi. Vrednosti ključev so napadalcu na začetku nepoznane, pozna pa neko matematično povezavo, ki jih povezuje. Na primer: napadalec ve, da je zadnjih 80 bitov vedno isto, čeprav ne pozna vrednosti.

Napad na prvi pogled izgleda nerealističen. Bilo bi zelo neverjetno, da bi napadalec prepričal kriptografa, naj mu šifrira izbrana sporočila z različnimi ključi, ki jih povezuje neka matematična zakonitost.

Pomemben primer šifrirnega protokola, ki je podlegel tej vrsti napada je protokol WEP, ki se uporablja v brezžičnih omrežjih WiFi. Vsak odjemalec in dostopovna točka v omrežju, ki ga varuje WEP, si delijo isti ključ WEP. Bistvenega pomena je, da se pri pretočni kodi isti ključ ne ponovi dvakrat. V ta namen WEP vključuje 24-bitni vektor IV v vsak paketek sporočila.

Ključe je potrebno menjavati ročno in to se tipično dogaja neredno. Napadalec lahko sklepa, da so vsi ključi, uporabljeni za kodiranje paketov, povezani s poznanim IV. To dejstvo je WEP odprlo seriji napadov.

Zakaj je do tega prišlo: 24-bitni vektor IV omogoča le nekaj manj kot 17 milijonov možnosti. Zaradi paradoksa rojstnega dne, je mogoče, da bosta dva paketa izmed 4096 paketkov imela isti IV in zato isti ključ RC4.

Zamenjava za WEP, WPA, preprečuje takšne napade tako, da uporablja tri nivoje ključev: glavni ključ, delovni ključ in ključ RC4. Glavni ključ souporabljajo vsak odjemalec in dostopovna točka. Uporabljen je pri protokolu TKIP za ustvarjanje novih delovnih ključev dovolj pogosto, da ustavlja poznane metode napada. Delovni ključi in daljši, 48-bitni, IV so kombinirani v ključ RC4 za vsak paket.

3 Metode v kriptoanalizi

3.1 Klasična kriptoanaliza

Klasične šifrirne postopke po načinu zakrivanja ločimo na substitucijske metode zakrivanja, kjer vsak znak ali skupino znakov nadomestimo z drugimi, in transpozicijske metode zakrivanja, ko spreminjamo vrstni red znakov.

Kriptoanaliza klasičnih šifrirnih postopkov ali klasična kriptoanaliza se posveča preučevanju pogostosti pojavljanja posameznih črk ali skupin črk v šifropisu.

Dandanes težko delo štetja in analiziranja znakov opravlja računalniška programska oprema, ki delo opravi v sekundah. Če klasičnim šifrifnim postopkov ob bok postavimo moderne izračunske moči, je jasno, da v teh časih ne morejo ponujati prave zaščite za zaupne podatke.

3.1.1 Frekvenčna analiza

Pri kriptoanalizi gre za preučevanje pogostosti pojavnjanja posameznih črk ali skupin črk v šifropisu. Osnovana je na dejstvu, da se v vsakem odseku pisanega besedila določene črke ali skupine črk pojavljajo različno pogosto, z različnimi frekvencami. Še več, obstaja karakteristična porazdelitev šrk, ki je približno enaka v skoraj vseh vzorcih določenega jezika.

Osnovna uporaba frekvenčne analize je ugotavljanje pogostosti znakov v šifropisu in nato povezovanje z mogočimi znaki v čistopisu. V angleškem jeziku, na primer, je E zelo pogost, X pa se pojavlja zelo redko. Podobno so ST, NG TH in QU pogosti pari črk (bigrami). 12 najbolj pogostih črk je ETAOIN SHRDLU. Preprosti substitucijski šifrirni postopek zamenja vsak znak čistopisa za nek drug, določen znak pa bo vedno zamenjan z istim znakom. Na primer: vsak E bo zamenjan z X. Čistopis z veliko znakov X, bo ponujal napadalcu sklep, da X predstavlja nek pogost znak, na primer E. Bolj kompleksna uporaba statistike ponuja več informacij napadalcu: preučevanje več parov, treh znakov (trigramov) itd.

Na primer: Q se skoraj vedno pojavlja skupaj u U, čeprav se Q sam po sebi redko pojavlja. Pri nekaterih širfrirnih postopkih so te lastnosti naravnega jezika ohranjene v šifropisu. Ti vzorci so lahko uporabljeni v napadu na osnovi šifropisa.

Prvo znano zapisano razlago frekvenčne analize (katerekoli oblike kriptoanalize) je verjetno podal arabski polimat v 9.stoletju. Po podrobni analizi Korana je bilo jasno, da ima arabski jezik karakteristično pogostost

pojavljanja črk. Uporaba se je zelo razmahnila, zato so kriptografi razvijali veliko shem, da bi se jim postavili po robu.

- **Homofoni**. Več alternativ najbolj pogostim besedam. Primer: v X in Y v šifropisu lahko pomenita E v čistopisu;
- **Večabecedna substitucijske metode** (ang. Polyalphabetic substitution) ali uporaba več abeced. Primer: Vigenerejeva šifra;
- **Poligrafična substitucijske metode**, ki obravnavajo pare ali trojice znakov kot enoto zamenjave. Primer: šifra Playfair.

Slabost vseh the poskusov je, da povišujejo stopnjo zapletenosti tako šifriranja kot dešifriranja, kar zlahka pripelje do napak.

3.1.2 Kasiskijev pregled

Leta 1863 je Kasiski izdal knjigo, ki velja za prvo na področju napadov na večabecedne substitucijske metode, še posebno na Viegenerejo šifro. Metodo sta neodvisno razvila Babbage in Kasiski.

Z metodo je mogoče ugotoviti dolžino ključne besede (ang. Keyword) uporabljene pri večabecednem substitucijskem šifrirnem postopku. Po tem se besedilo – šifropis razporedi v stolpce dolžine n, kjer je n dolžina ključne besede. Vsak stolpec je obravnavan kot enoabeceden substitucijski šifrirni postopk, na primer napaden s frekvenčno analizo.

Metoda pregleduje nize znakov, ki se v šifropisu ponavljajo. Nizi naj bi bili za uspešno analizo dolgi vsaj tri znake. Razdalje med dvema zaporednima pojavoma niza so verjetno večkratniki ključne besede. Iskanje več ponavljajočih nizev zmanjšuje možnosti dolžine ključne besede, saj poiščemo največji skupni delitelj vseh razdalij.

Primer:

Kripto_je_kratko_za_**kripto**analizo

Kripto je niz, ki se ponavlja. Razdalja med dvema zaporednima pojavoma je 20 znakov. Čistopis bomo primerjali z ključem, ki ima šest znakov '**abcdef'** (6 ne deli 20), in ključem, ki ima pet znakov **'abcde'.**

abcdef	abcdef	abcdef	ab**cdef**	**ab**cdef	abc
kripto	_je_kr	atko_z	a_**krip**	**to**anal	izo

Tabela 1: Kasiskijev pregled, prvi ključ, [3]

abcde	**a**bcde	abcde	abcde	**abcde**	**a**bcde	abc
kript	**o**_je_	kratk	o_za_	**kript**	**o**anal	izo

Tabela 2: Kasiskijev pregled, drugi ključ, [3]

Težava je iskanje nizov, ki se ponavljajo. To je ročno zelo težavna operacija, ki jo računalniki zelo poenostavijo. Vseeno pa je potrebna človeška interakcija, saj je ponavljanje nekaterih nizov zgolj slučajnost – razdalje bodo imele največji skupni deitelj 1.

Moderni napadi na večabecedne šifre so v osnovi identični tem, z dodatkom štetja slučajnosti (ang. Coincidence counting).

3.1.3 Štetje slučajnosti

Tehnika je primerjava dveh besedil, ki sta postavljena skupaj. Beleži se število slučajev, kjer se znak pojavi na istem mestu v obeh besedilih. To je lahko v pomoč pri določanju slučaja, ko sta obe besedili napisani v istem jeziku, z isto abecedo. Takrat bodo seštevki občutno višji kot pri besedilih v različnih jezikih, šifriranih z različno abecedo ali pri nesmiselnih besedilih.

Predstavljajmo si abecedo s samo dvema črkama A in B. Naj se v našem jeziku A pojavlja z verjetnostjo 75% in B z 25%. Če postavimo dve besedili drugo ob drugega, lahko pričakujemo naslednje pare, ki so zapisani v tabeli na levi.

Par	Verjetnost
AA	56.25%
BB	6.25%
AB	18.75%
BA	18.75%

Par	Verjetnost
AA	18.75%
BB	18.75%
AB	56.25%
BA	6.25%

Tabela 3: Štetje slučajnosti, [3]

Verjetnost slučajnosti je: 62.5% (56.25% za AA in 6.25% za BB).

Sedaj pa vzemimo, da uporabimo dv besedili: eno v tem jeziku in drugo šifrirano s substitucijsko metodo, ki zamenja A z B in obratno. Pričakujemo lahko naslednje pare v desni tabeli.

Verjetnost slučajnosti je sedaj samo 37.5% (18.75% za AA in 18.75% za BB), kar je očitno vidno nižje kot pri uporabi istega jezika in iste abecede.

Isti princip drži za resnične jezike, na primer za angleščino. Določeni zanki, recimo E, se pojavljajo veliko bolj pogosto kot drugi – dejstvo, ki je uporabljeno pri frekvenčni analizi substitucijskih metod. Slučajnosti, ki vključujejo E so precej pogoste. V slučaju primerjave dveh angleških tekstov je štetje slučajnosti višje kot pri primerjavi angleškega in nekega drugega jezika.

Lahko si je predstavljati, da je ta učinek subtilen. Podobni jeziki imajo višje štetje slučajnosti kot jeziki, ki si niso podobni. Ni težko generirati naključnega besedila, ki ima podobno frekvenčno distribucijo kot resnično besedilo, in tako umetno dvigniti štetja slučajnosti.

Matematičen zapis I (indeks slučajnosti) je podan v nadaljevanju. Dolžina besedila je označena z n, z $n_1 ... n_m$, so označene pogostosti ali frekvence posameznih znakov abecede.

$$I = \sum_{i=1}^{m} \frac{(n_i - 1)n_i}{(n-1)n}$$

Slika 1: Indeks slučajnosti, [3]

Indeks slučajnosti je 6.6% za angleški jezik in 7.6% za nemški jezik. Če bi bilo vseh m črk abecede enako pogostih, bi bil indeks enak 1/m.

3.2 Kriptoanaliza simetričnih postopkov

3.2.1 Diferenčna in linearna kriptoanaliza

Linearna in diferenčna kriptoanaliza šifrirnih sistemov ne napadata direktno. Namesto tega preučujeta bločne šifrirne postopke in ugotavljata slabosti zasnove. Rezultat tega je, da so bločni šifrirni postopki zasnovani z zavedanjem tega dejstva. Popolnoma razumeti principe zasnove bločnih postopkov pomeni razumeti linearno in diferenčno kriptoanalizo.

Vpliv standarda DES na moderno kriptografijo ne more biti precenjen. Tako diferenčna kot tudi linearna kriptoanaliza sta bili zasnovani za napad na DES. Ne predstavljajo praktičnega napada, pač pa kažejo na slabosti zasnove bločnih šifrirnih postopkov. Tehnike so postale osnovno orodje za analizo vseh bločnih šifirinih postopkov danes.

Diferenčno analizo sta leta 1990 predstavila Shamir in Biham. Velja za napad na osnovi izbranega čistopisa. Linearno kriptoanalizo je leta 1993 razvil Matsui. Velja za rahlo bolj realističen napad, večinoma zato ker je napad na osnovi poznanega čistopisa, torej ne potrebujemo več čistopisov.

DES

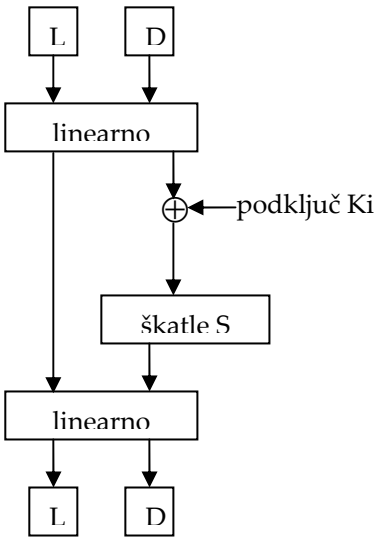
DES je eden starejših, zelo uveljavljenih standardiziranih algoritmov. Je simetričen in ima substitucijske in transpozicijske lastnosti, ki jih kombinira v kaskadi preslikav.

DES je nastal kot izpeljanka šifrirnega algoritma, ki ga je uporabljala ameriška vojska, ki ga je sicer uporabljala s 64-bitnim ključem. Teoretično ozadje algoritma ni pojasnjeno, zato obstaja sum, da vojska pozna bližnjico za razbijanje civilne različice DES.

Danes navadnega algoritma DES nihče več nima za varnega, uporablja se le še zaradi združljivosti s starejšimi sistemi. V praksi se pogosto uporablja trojni DES, pri katerem 64-bitni blok trikrat spustimo skozi postopek, le da v vsaki ponovitvi uporabimo drugačen 56-bitni ključ. Način je reda 10^{17} varnejši od navadnega DES, je pa tudi trikrat počasnejši.

Za opis tehnike ne potrebujemo vseh podrobnosti DES, zato je podan zgolj poenostavljen model. DES ima 8 škatel S, ki 6 vhodnih bitov $x_0x_1x_2x_3x_4x_5$ preslikajo v 4 izhodne bite $y_0y_1y_2y_3$. Preslikave delujejo nad 64-bitnimi bloki. V funkcionalnosti se ne razlikujejo, uporabljajo pa različne ključe K_i. Desna polovica bloka (spodnjih 32 bitov) se nespremenjena preslika na levo polovico izhoda (zgordnjih 32 bitov). Desno polovico izhodnega bloka dobimo po izvedbi nelinearne funkcije nad vhodnima blokoma in ključem K_i trenutne preslikave.

Slika prikazuje zelo poenostavljen pogled na DES, ki zadošča. Pomembno se je posvetiti nelinearnim delom DES, zato diagram poudari dejstvo, da so škatle S edina nelinearnost v sistemu. Slika prikazuje tudi način vstopa ključa v sistem.



Slika 2: Poenostavljen model DES, [1]

Pregled diferenčne analize

Diferenčna analiza je bila razvita za preučevanje DES, zato jo je smiselno razložiti v tem kontekstu. Celoten sistem DES razen škatel S je linearen. Linearni deli sicer igrajo pomembno vlogo pri varnosti, z vidika kriptoanalize pa so preprosti. Matematiki se dobro spopadajo z linearnimi enačbami, zato težavo pri DES predstavljajo nelinearni deli. Tako diferenčna kot tudi linearna analiza se posvečata nelinearnim delom DES, škatlam S.

Ideja diferenčnega napada je primerjava vhodnih in izhodnih diferenc. Zaradi preprostosti se omejimo na poenostavljeno škatlo S, preslika 3 bite v 2 bita. Predpostavimo, da za 3 vhodne bite $x_0x_1x_2$ bit x_0 predstavlja vrstico, bita x_1x_2 pa stolpec.

	stolpec			
vrstica	0 0	0 1	1 0	1 1
0	1 0	0 1	1 1	0 0
1	0 0	1 0	0 1	1 1

Tabela 4: Primer škatle S, [1]

Za primer vzemimo dva vhodna niza, $X_1 = 110$ in $X_2 = 010$, in ključ K=011. Dobimo $X_1 \oplus K = 101$ in $X_2 \oplus K = 001$. Velja:

$$\text{škatla}_S(X_1 \oplus K) = 10 \text{ in } \text{škatla}_S(X_2 \oplus K) = 01.$$

Zdaj predpostavimo, da je K neznanka, poznamo pa vhode ($X_1 = 110$ in $X_2 = 010$) in pripadajoče izhode ($\text{škatla}_S(X_1 \oplus K) = 10$ in $\text{škatla}_S(X_2 \oplus K) = 01$). Pogled v tabelo nam pove, da je $X_1 \oplus K \in \{000, 101\}$ in $X_2 \oplus K \in \{001, 110\}$. Ker poznamo X_1 in X_2, velja:

$$K \in \{110, 011\} \cap \{011, 100\}$$

Iz česar sledi, da je K = 011. Ta napad je v osnovi napad na osnovi para čistopis-šifropis na eno škatlo S za ključ K, isti pristop deluje na eno DES škatlo S.

Napadanje ene škatle S v enem ciklu DES verjetno ni preveč uporabno. Poleg tega napadalec ne bo poznal vhoda nobenem ciklu, razen prvemu, in izhoda nobenega cikla, razen zadnjega. Vmesni cikli izgledajo zunaj dosega kriptoanalize. Če želimo, da je ta napad uporaben za analiziranje DES, je potrebno napad razširiti na en zaključen cikel – torej na istočasno delovanje vseh škatel S. Ko je napad razširjen na en zaključen cikel, ga je potrebno razširiti na več ciklov.

Posvetimo se razlikam med vhodi in izhodi. Tako je enostavno nekatere škatle označiti za 'aktivno' in ostale za 'neaktivne'. Rezulat tega je, da lahko napad razširimo na en cikel. Da bi ga lahko kasneje razširili na več, moramo izbrati takšne diference, ki oblikujejo izhodne diference v uporabno obliko za naslednji cikel.

Recimo, da poznamo vhoda X_1 in X_2. Dejanski vhod v škatlo S je $X_1 \oplus K$ (za X_1) in $X_2 \oplus K$ (za X_2), ključ K je neznan.

Razlike so definirane kot operacija po modulu 2, torej sledi, da je operacija iskanja razlike enaka kot operacija izključno ali XOR ($\oplus$). Razlika vhodov v škatlo S je tako: $(X_1 \oplus K) \oplus (X_2 \oplus K) = X_1 \oplus X_2$, kar pomeni, da je vhodna diferenca neodvisna od ključa K. To je osnovna ugotovitev diferenčne kriptoanalize.

Naj bo $Y_1 = \text{škatlaS}(X_1 \oplus K)$ in $Y_2 = \text{škatlaS}(X_2 \oplus K)$. Izhodna diferenca $Y_1 \oplus Y_2$ je skoraj enaka vhodni diferenci naslednjega cikla. Cilj je previdno načrtovati vhodno diferenco, da jo lahko zasledimo skozi več ciklov. Ker je vhodna diferenca neodvisna od ključa in ker je diferenčna kriptoanaliza napad na osnovi izbranega čistopisa, si lahko vhode izberemo tako, da so izhodne diference v željeni obliki.

Drug pomemben element napada je dejstvo, da se vhodna diferenca 0 vedno preslika v izhodno diferenco 0. Vhodna diferenca 0 pomeni, da sta vhodni vrednosti X_1 in X_2 zavzeli enako vrednost, zato bosta izhodni vrednosti enaki: $Y_1 \oplus Y_2 = 0$. Pomembnost te osnovne ugotovitve leži v sposobnosti, da nekatere škatle S izločimo kot 'neaktivne' z vidika diferenčne kriptoanalize, saj je njihova vhodna diferenca enaka 0.

Zadnja ugotovitev je, da ni nujno, dase stvari zgodijo z gotovostjo. Z drugimi besedami: če se izhod zgodi le z neko netrivialno verjetnostjo, lahko še vseeno razvijemo učinkovit napad.

Za vsako škatlo S, lahko izvedemo analizo uporabnih vhodnih diferenc. Za vsako možno vhodno vrednost X poiščemo pare X_1 in X_2 tako, da je $X=X_1 \oplus X_2$, in izračunamo pripadajoče izhodne diference $Y=Y_1 \oplus Y_2$. Rezultati so v tabeli, ki sledi.

	škatlaS(X_1) ⊕ škatlaS(X_2)			
$X_1 \oplus X_2$	00	01	10	11
000	8	0	0	0
001	0	0	4	4
010	0	8	0	0
011	0	0	4	4
100	0	0	4	4
101	4	4	0	0
110	0	0	4	4
111	4	4	0	0

Tabela 5: Diferenčna kriptoanaliza, [1]

Primer: vhodna diferenca 010 se vedno preslika v 01, kar je najbolj pogost rezultat.

Pregled linearne kriptoanalize

Ironično linearna kriptoanaliza, tako kot diferenčna kriptoanaliza, temelji na nelinearnih delih bločnega šifrirnega postopka. Čeprav je bila linearna kriptoanaliza zasnovana nekaj let za diferenčno, je v osnovi preprostejša in učinkovitejša za DES. Potrebuje le en čistopis, ne več čistopisev kot diferenčna kariptoanaliza.

Pri diferenčni kriptoanalizi smo se osredotočili na vhodne in izhodne razlike. Ideja linearne kriptoanalize je izdelati linearen približek (linearne enačbe) za nelinearne dele. Matematiki so dobri pri reševanju linearnih enačb, zato je jasno, da, če je linearizacija mogoča, to tudi uporabimo za napad. Edina nelinearnost so škatle S, zato je linearna kriptoanaliza namenjena le-tem.

Vrnimo se spet k škatli S na tabeli. Tri vhodne bite označimo z $x_0x_1x_2$, dva izhodna pa z y_0y_1. Bit x_0 predstavlja vrstico, bita x_1x_2 pa stolpec. V naslednji tabeli so označene vrednosti za katere linearen približek velja. Ker obstaja 8 izhodnih vrednosti, je vsaka vrednost, ki se razlikuje od 4, odstopanje od naključne porazdelitve.

	Izhodni biti		
Vhodni biti	y_0	y_1	$y_0 \oplus y_1$
0	4	4	4
x_0	4	4	4
x_1	4	6	2
x_2	4	4	4
$x_0 \oplus x_1$	4	2	2
$x_0 \oplus x_2$	0	4	4
$x_1 \oplus x_2$	4	6	6
$x_0 \oplus x_1 \oplus x_2$	4	6	2

Tabela 6: Linearna kriptoanaliza, [1]

Rezultati prikazujejo, na primer, $y_0 = x_0 \oplus x_2 \oplus 1$ z verjetnostjo 1 in $y_0 \oplus y_1 = x_1 \oplus x_2$ z verjetnostjo ¾. S takšnimi informacijami lahko naše škatle S zamenjamo z linearnimi funkcijami. Rezultat je zamenjava nelinearnih škatel S za linearne enačbe, ki držijo le z neko verjetnostjo.

Da so te linearne enačbe uporabne pri napadu na bločni šifrirni postopek kot je DES, je potrebno pristop razširiti tako, da lahko rešujemo linearne enačbe za ključ. Kot pri diferenčni kriptoanalizi je potrebno rezultate nanizati preko več ciklov.

Kako dobro je mogoče primerjati DES škatle S z linearnimi funkcijami? Vsaka šlatla S je bila načrtovana tako, da nobena linearna kombinacija vhoda ni dober približek enemu izhodnemu bitu. Obstajajo pa linearne kombinacije izhodnih bitov, ki so približek linaernim kombinacijam vhodnih bitov. To je možnost za uspeh linearne kriptoanalize.

Zaključek

Varnosti šifrirnega postopka se ne da dokazati. Pred nepoznanimi napadi se je težko braniti, zato se kriptografi ukvarjajo s poznanimi napadi. Za bločne šifirne postopke so to diferenčna, linearna kriptoanalizo in variante le-teh. Zato je poglavitni cilj pri načrtovanju bločnih postopkov onemogočanje napadov z diferenčno in linearno kriptoanalizo.

Kako lahko kriptografi otežijo napade? Poglavitni pomislek pri iterirani bločni kodi je izbiranje med številom ciklov, količino zamenjav in stopnjo razprševanja.

Pri obeh napadih boo uspeh pri verjetnosti v enem ciklu, ki je manjši od 1, verjetno zmanjšan z vsakim ponovnim ciklom. Bločna koda, ki bo imela le povišano število ciklov, bo varnejša. Drugi način je višja stopnja mešanja. Pri DES to pomeni izgradnjo boljših škatel S. Spet, več mešanja – več varnosti.Po drugi strani pa boljša nelinearnost otežkoči vgradnjo napadov.

3.2.2 Kriptoanaliza po modulu n

Napad po modulu n se uporablja za bločne in pretočne šifrirne postopke. Pri svojem delovanju izkorišča posebnosti delovanja šifrirnega postopka preko ekvivalenčnih (kongruenčnih) razredov po modulu n. Metodo so prvič predlagali leta 1999 in uporabili na RC5P (različici RC5) in M6 (družina bločnih šifer, uporabljenih v standardu FireWire).

3.2.3 Napad XSL

Napad XSL (razširjena raztresena linearizacija) sloni na analiziranju intervalov šifre in izpeljavi sistema kvadratnih istočasnih enačb. Sistemi enačb so tipično zelo veliki, na primer za 128-bitni AES 8000 enačb z 1600 spremennljivkami. Za reševanje takšnih sistemov obstaja več metod, pri tej se za reševanje enačb in pridobivanja ključa uporablja posebni algoritem XSL.

Napad XSL vsebuje učinkovit algoritem za obvladovanje kvadratnih enačb z več spremenljivkami (ang. Multivariate quadratic equations, MQ). Leta 1999 sta Kipnis in Shamir pokazala, da je mogoče algoritem z javnim ključem, bolj natančno shemo HFE, zmanjšati na sistem kvadratnih enačb, kjer je enačb več kot spremenljivk (predefiniran sistem). Ena izmed tehnik za reševanje takšnih sistemov je linearizacija, ki zamenjuje vsako kvadratnost z neodvisno spremenljivko, pridobljeni linearni sistem pa rešuje z algoritmi kot so Gausova eliminacija. Linearizacija zahteva dovolj linearno neodvisnih enačb.

Pri kriptoanalizi HFE enačb ni bilo dovolj, zato sta Kipnis in Shamir predlagala relinearizacijo, ki v sistem enačb po linearizaciji doda nelinearne enačbe. Rezultirajoči sistem je rešljiv po drugi uporabi linearizacije. Relinearizacije se je izkazala kot dovolj splošna metoda in je uporabna tudi za druge sheme.

Leta 2000 je bil predlagan izboljšani algoritem XL (raztresena linearizacija), ki zvišuje število enačb tako, da jih množi z vsemi monimiali določene stopnje. Napad naj sicer ne bi deloval na enačbah, izpeljanih iz bločnih šifrirnih postopkov kot AES. Vseeno pa je imel izdelani sistem enačb posebno strukturo, ki jo je izkoristil algoritem XSL, izboljšava XL. Pri XSL so enačbe množene le s posebno izbranimi monomiali, predlaganih je bilo veliko variant.

Mogoče je, da XSL deluje pri nekaterih modernih algoritmih, trenutno pa ne predstavlja velike nevarnosti v smislu praktične varnosti. Čeprav je hitrejši

kot napad surove sile, so potrebni viri še vedno ogromni, zato je precej neverjetno, da bi bili sistemi realnega sveta ogroženi. Do sprememb bi lahko prišlo z izboljšavami. Ker je ta način napada nov in nepričakovan, so nekateri kriptografi izrazili nelagodje ob algebrski strukturi šifrirnega postopka kot je Rijndael.

3.2.4 Drseči napad

Napad je namenjen kriptoanalizi bločnih šifrirnih postopkov. Idejo sta prvotno objavila Edna Grossman in Byrant Tuckerman leta 1977. Ta oblika kriptoanalize je bila izdelana zaradi prevladujoče ideje, da je mogoče tudi zelo šibke šifrirne postopke utrditi s povišanjem ciklov, kar naj bi ustavilo diferenčni napad. Grossman in Tuckerman sta kasneje tudi demonstrirala napad na šibko bločno šifro NDS.

Drseči napad deluje na način, ki naredi število ciklov za nepomembno. Analizi razporeda ključev (algoritem za izračun podključev za vsak cikel) sledi izkoriščanje šibkosti za razbijanje šifrirnega postopka. Najbolj splošno je ponavljanje ključev na cikličen način.

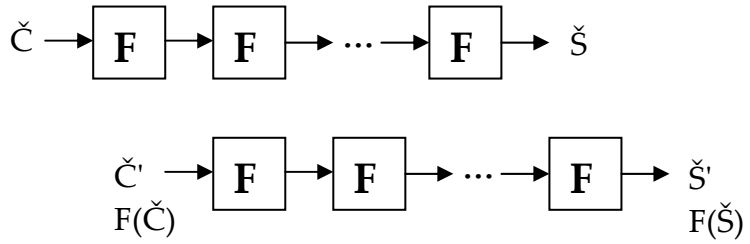
Edina zahteva za drseči napad na postopek je, da ga je mogoče razdeliti na več ciklov identične funkcije F. Funkcija mora biti ranljiva na napad na osnovi para čistopis-šifropis. Napad je tesno povezan z napadom s pripadajočim ključem.

Dejanski napad

Vzemimo, da ima šifrirni postopek bloke dolge n bitov in razpored ključev $K_1 ... K_m$, ki so poljubne dolžine. Napad deluje tako, da razbije kodo na identične permutacijske funkcije F, ki lahko vsebujejo več ciklov kode. To določa razpored ključev. Na primer: če postopek uporablja izmenjujoč razpored ključev, kjer preklaplja med K_1 in K_2 za vsak cikel, potem je funkcija F sestavljena iz dveh ciklov. Vsak K_i se bo v F pojavil vsaj enkrat.

Naslednji korak je pridobiti $2^{n/2}$ (paradoks rojstnega dne) parov čistopis-šifropis. Te pare, ki jih označimo kot (Č,Š), uporabimo za iskanje drsnega para ($Č_0$,$Š_o$) ($Č_1$,$Š_1$). Drsni par ima lastnosti: $Č_0 = F(Č_1)$ in $Š_0 = F(Š_1)$. Par označuje sporočilo, na katerem je bila uporabljena funkcija F – zdrselo je preko enega šifrirnega cikla. Po identifikaciji drsnega para je šifra razbita zaradi ranljivosti na napade na osnovi poznanega čistopisa. Iz para je mogoče preprosto pridobiti ključ.

Postopek je sledeč: izberemo katerega koli izmed parov ($Č_0$,$Š_o$) ($Č_1$,$Š_1$) in preverimo, kateri ključi pripadajo $Č_0 = F(Č_1)$ in $Š_0 = F(Š_1)$. Če se ključi ujemajo, smo našli drsni par, v nasprotnem primeru pa nadaljujemo s preverjenjem.



Slika 3: Drseči napad, [3]

3.3 Kriptoanaliza asimetričnih postopkov

Asimetrična kriptografija (kriptografija z javnim ključem) je kriptografija, ki temelji na uporabi dveh ključev: privatnega in javnega ključa. Vsi algoritmi za zakrivanje z javnim ključem so asimetrični, kar pomeni, da se šifrirni (enkripcijski, E ključ) in dešifrirni (dekripcijski, D ključ) ključ razlikujeta.

Takšni šifrirni postopki temeljijo na 'težkih' matematičnih problemih, ki pomenijo osnovo njihove varnosti. Očitna točka napada je razviti metodo, ki bo problem rešila. Varnost kriptografije z javnim ključem je osnovana na matematičnih vprašanjih, tako kot kriptografija z enim ključem splošno ni, zato povezuje kriptoanalizo s širšimi matematičnimi raziskovanji na nove načine.

Asimetrične sheme so osnovane na težavnosti reševanja različnih matematičnih problemov. Če je razvit izboljšan algoritem, ki reši poblem, posledično oslabi sistem. Shema izmenjave ključev Diffie-Hellman temelji na težavnosti izračuna diskretnega logaritma. Leta 1983 je Don Coppersmith razvil računsko izvedljiv način iskanja diskretnega logaritma in tako kripoanalitikom podal orodje, s katerim so lahko razbili šifrirni sisten Diffie-Hellman. Druga shema, algoritem RSA, še ni bil razbit. Varnost temelji na težavnosti faktorizacije. Preboj na področju razcepa bi vplival na varnost RSA.

Faktorizacija (praštevilski razcep, razcep na prafaktorje) je predstavitev števila kot zmnožka manjših števil. Po osnovnem teoremu aritmetike, ne vsako naravno število večje od ena mogoče zapisati kot praštevilo ali kot produkt praštevil. Za razcep velikih števil ni poznanega učinkovitega algoritma. Očitna težavnost problema je srce določenih algoritmov. S problemom se spopadajo področja matematike in računalništva, na primer eliptične krivulje, algebrska teorija števil in kvantno računanje. Faktorizacija števil različne dolžine je različno težavna. V tem času velja za najtežji problem faktorizacija dveh naključno izbranih praštevil približno enake velikosti.

Problem izračuna diskretnega logaritma je soroden problemu faktorizacije. Oba sta težko rešljiva, saj ni poznanih učinkovitih algoritmov. Algoritmi enega problema se pogosto prenesejo na drugega. Težavnost obeh pa se s pridom uporablja pri kriptografskih problemih.

3.3.1 Rojstnodnevni napad

Rojstnodnevni napad izkorišča matematiko, ki temelji na paradoksu rojstnega dne. Če lahko funkcija zavzame H različnih enako verjetnih izhodov in je H dovolj velik, potem po izračunu funkcije za približno $1{,}2\sqrt{H}$ različnih argumentov lahko pričakujemo par različnih argumentov z enakim izhodom – trk.

Primer: če je uporabljena 64-bitna zgoščevalna funkcija, je mogočih $1.9\ 10^{19}$ različnih izhodov. Za trk bi bilo potrebno le približno $5.1\ 10^{9}$ poskusov. V primeru, da so izhodne vrednosti funkcije razporejene neenako, so možnosti za trk še večje. Pojem 'uravnoteženosti' zgoščevalne funkcije ocenjuje odpornost na rojstnodnevni napad.

Na ta tip napada je lahko ranljiv digitalni podpis. Sporočilo m je tipično podpisano v dveh korakih. Najprej je izračunana funkcija f(m), kjer je f zgoščevalna funkcija. Nato pa se uporabi skrivni ključ za podpis f(m).

Recimo, da želi Alice pretentati Boba, da bi podpisal pogodbo. Alice pripravi dve pogodbi m in m'. Nato poišče število mest, kjer lahko m spremeni brez spremembe pomena: vstavljanje vejic, praznih vrstic, edennamesto dveh presledkov itd. Na ta način lahko ustvari veliko število variacij m, ki so vse veljavne pogodbe, in variacij m'. Nad vsemi izvede zgoščevanje. Ko najde primer veljavne pogodbe in primer neveljavne pogodbe, ki imata enako vrednost f(m) = f(m') je prišlo do trka. Pravo pogodbo ponudi Bobu v podpis, nato pa podpis prestavi na drugo pogodbo. Podpis potrjuje, da je Bob podpisal m'.

Da do tega napada ne pride, je potrebno zagotoviti, da je izhod zgoščevalne funkcije dovolj velik. Tako je rojstnodnevni napad računsko neizvedljiv. Priporočljivo je tudi, da Bob, preden podpiše pogodbo, doda nekaj kozmetičnih popravkov. To problem sicer ne reši, saj lahko sedaj Alice sumi, da želi uporabiti napad.

Rojstnodnevni napad lahko doda hitrosti iračuna diskretnega logaritma.

3.3.2 Bočni napad na RSA

Pogosto je mogoče napasti šifrirni postopek brez direktnega napada na algoritem. Veliko procesov lahko namreč proizvede neželjene stranske kanale, ki prepuščajo informacije. Razlog za to lahko tiči v izvedbi izračunov, uporabljenega medija, porabljene moči itd.

Opisan je postopek časovnega bočnega napada na algoritem RSA. Napad temelji na dejstvu, da nekatere operacije trajajo več kot druge. Z natančnim izračunom je mogoče določiti tajni ključ RSA. Podoben napad je dovolj robusten za uspešno izvedbo vgradnje RSA v verziji odprtega SSL preko omrežne povezave.

Pregled RSA

Prvi, danes tudi najbolj znani, algoritem za zakrivanju z javnim ključem je nastal leta 1978 na ameriškem institutu MIT. Algoritem, ki so ga ustvarili Rivest, Shamir in Adelman, temelji na celoštevilski algebri in izkorišča lastnosti velikih praštevil.

Postopek nima problema simetričnih algoritmov, distribucije ključa ni. Šibka točka pa je počasnost. Vzrok je izračun šifriranja in dešifriranja, saj gre za potenciranje dveh zelo velikih števil. Operacija je računsko zahtevna in zato tudi počasna. Algoritem je za sedaj primeren za skrivanje krajših sporočil, na primer kjučev za šifriranje po simetričnem postopku.

Ideja algoritma

Izberemo dve veliki praštevili: p in q. Izračunamo n in z po naslednjih enačbah: n = pq; z = (p-1)(q-1). Izberemo število D (tajni ključ), ki nima skupnih faktorjev s številom z. D in z sta si tuji števili. E (javni ključ) izberemo tako, da velja enačba $ED = 1_{\mod z}$. Šifriranje in dešifriranje potekata po naslednjih enačbah: $E(P) = P^{E} \bmod n$; $D(E(P)) = E(P)^{D} \bmod n$

Časovni napad na RSA

Vzemimo, da je M sporočilo, ki ga bo Alice podpisala, in D njen tajni ključ. Alice podpiše M z izračunom $(M)^{D} \bmod n$. Recimo, da želi Trudy dobiti Alicin tajni ključ D. Vzemimo, da je ključ dolg n+1 bitov, ki jih označimo:

$D = D_0 D_1 ... D_n$; kjer je $D_0 = 1$

Vzemimo, da je bilo za izračun $(M)^{D} \bmod n$ uporabljeno ponavljajoče kvadriranje:

```
x = M
for j = 1 to n
        x = mod (x^2, N)
        if d_j ==1 then
                x = mod (xM, N)
        end if
next j
return x
```

Slika 4: Ponavljajoče kvadriranje, [1]

```
Mod (x, N)
If x >=N
    x = x % N
end if
return x
```

Slika 5: Učinkovita funkcija mod, [1]

Vzemimo, sa je mod(x, N) tak kot na sliki. Rezultat je, da bo čas izračuna za $d_j = 1$ in $d_j = 0$ različen. Izkaže se, da lahko Trudy s pomočjo napada CPA pridobi bit ključa D_1 in s skrbno izbranimi čistopisi še naslednje bite. Seveda pa se mora zanesti na statistiko, še posebno preko omrežja, kjer se lahko pojavijo odstopanja v časovnem smislu.

3.4 Posebni napadi

3.4.1 Bočni napad

Bočni napad je vsak napad, ki namesto na teoretičnih slabostih algoritma temelji na informacijah, pridobljenih iz fizične vgradnje šifrirnega sistema. To so na primer: informacija o času, poraba moči, elektromagnetno izžarevanje in celo zvok. Vsak dodaten vir informacij je lahko uporaben pri napadu na sistem.

Za vodilnega na področju bočnih napadov velja Paul Kocher. Odkritje ranljivosti pametnih kartic na tovrstne napade, kar je ustavilo uveljavljenost za nekaj let, gre pripisati njemu.

Velik potencialen vir informacij izhaja iz tako imenaovanih nenamernih izžarevanj. Celotna veja varnosti je posvečena varnosti sevanja (EMSEC, TEMPEST). Mogoče je, na primer, s pomočjo elektromagnetnega sevanja z razdalje konstruirati računalniški zaslon. Pametne kartice so bile napadene tudi s pomočjo diferenčne analize moči (PDA), ki izkorišča dejstvo, da nekatere operacije potrebujejo višjo porabo energije.

Tovrstne napade prištevamo med pasivne, aktivni pa se pogosto imenujejo diferenčna analiza okvare (DFA). Ideja je povzročiti okvaro s ciljem pridobivanja informacij. Na pametne kartice v telefonih GSM je mogoče izvesti napad s pomočjo tehnik DFA.

Napadi na ljudi, ki se bavijo s šifriranjem, se večinoma ne imenujejo bočni. Večina bočnih napadov zahteva precejšnje tehnično znanje notranjih operacij sistema, na katerem je šifrirni postopek nameščen.

Časovni napad opazuje premikanje podatkov v in iz CPE ali spomina, na strojni opremi, kjer teče algoritem. Preprosto samo z opazovanjem koliko časa je potrebno za prenos informacije o ključu je včasih mogoče določiti, kako dolg je ključ v tem primeru. Uporabna je tudi izformacija, ki izključi dolžine določenih ključev.

Neizbežno dejstvo električnega življenja v dejanskih vezjih je, da tokovi ogrevajo materiale, skozi katere tečejo. Le-ti izgubljajo toploto v okolico, zato zaradi gretja in ohlajanja obstaja nenehno mehansko tresenje. To največ pripomore k zvokovnemu oddajanju nizkih nivojev iz delujoče CPE. Nedavna

raziskava (Shamir) je dokazala, da je tudi na ta način mogoče pridobiti nekaj znanja o delovanju sistemov šifriranja in algoritmih.

Bočni napadi lepo prikažejo kako napadalci ne igrajo po pravilih. Napadalci se bodo vedno trudili izkoristiti najšibkejši člen verige varnostnega sistema. Najboljša zaščita je misliti kot napadalec.

3.4.2 Napad surove sile

Napad naj bi porazil sistem s preizkušanjem velikega števila možnosti, na primer preizkušanja vseh mogočih ključev. V večini shemah je teoretična možnost poznana. Urejeno je tako, da je napad računsko neizvedljiv v določenem časovnem okviru. Ena izmed definicij 'lomljena' šifrirne sheme je odkritje metode, ki deluje hitreje kot napad surove sile.

Za simetrične šifirine postopke napad s surovo silo išče ključ. Preizkušanje vseh mogočih ključev se itvaja toliko časa, dokler se iz šifropisa ne da poustvariti čistopisa.

Pri napadu surove sile je pričakovano število poskusov (preden najdemo pravi ključ) enak polovici ključa. Če obstaja, na primer, 2^{64} možnih ključev, bo napad potreboval povprečno 2^{63} poskusov da pridobi ključ.

Simetrični šifrirni postopki z dolžinami ključev 64-bitov so podlegli tovrstnemu napadu. DES je bil razbit z običajno strojno opremo leta 1998, RC5 pa pred kratkim. Če so ključi generirani na šibek način, na primer izvedeni iz gesel, ki jih je lahko ugotoviti, se nabor poskusov omeji.

Postopki z dokazano perfektno varnostjo, kot je na primer enkratna prevleka, niso ranljivi na napad surove sile.

Kvantni računalniki

Ena izmed mogočih uporab za kvantne računalnike bi lahko bila kriptoanaliza. Zaradi kvantnih stanj se odpirajo nov pogled na računanje.

Če bi bil kvantni računalnik zgrajen, bi se spremenilo veliko stvari. Mogoče bi bilo izvesti izjemno hitre napade surove sile, dolžine ključev, ki zdaj veljajo zunaj dosega vsakega napadalca surove sile, bi bile premajhne. Mogoče je, da po razmahu kvantnih računalnikov noben šifrirni postopek ne bo več varen, verjetno pa bo preprosto dodajanje bitov ključu spet ustavilo napade surove sile, tudi tiste s kvantnimi računalniki.

Druga možnost je, da bi povečana izračunska moč pomagala tudi napadom, ki niso s surovo silo. Na primer: ves uspeh na področju faktorizacije se nima zahvaliti algoritemskim izboljšavam. Veliko se ima zahvaliti povečani moči in prisotnost delovnih kvantnih računalnikov bi vidno pospešila faktorizacijo in tako tudi ranljivost nekaterih sistemov.

3.4.3 Napad vmesni človek

Napad vmesni človek (ang. man-in-the-middle attack) napadalcu omogoča branje, vsavljanje in spreminjanje besedil, ki potekajo med dvema stranema. Strani se ne zavedata, da je bila povezava med njima spremenjena. Napad MITM je bil še posebno ugoden za vgradnjo v protokol izmenjave Diffie-Hellman, če je bil uporabljen brez avtentikacije.

Napad vsebuje enega ali več izmed naslednjih metod:

- Prisluškovanje, vključno z analizo prometa in napadom KPA;
- Napad CCA;
- Substitucijski napad. Če napadalec pozna točno ali del vsebine enega izmed naših besedil lahko brez ključa spremeni del vsebine;
- Ponovitveni napad;
- Napad za zavrnitev storitve (ang. Denial of service).

3.4.4 Gumijasta cev

Izraz napad z gumijasto cevjo (ang.Rubber-hose attack) označuje kriptoanalitično metodo za pridobivanje šifrirnih skrivnosti na nasilen način in ne z matematičnim ali tehničnim napadom.

V modernih šifrirnih sistemih je najšibkejši člen pogosto uporabnik. Direktni napad na šifrirni algoritem ali na uporabljene protokole bo dražji in težji kot napad na uporbnike. Zato je mnogo šifrirnih in varnostnih sistemov načrtovano tako, da je človeška ranljivost znižana na najmanjšo vrednost. Grožnje ne bodo učinkovite pri razbijanju sistema.

Podobno deluje socialni inženiring. Na nežnejši način z manipulacijo uporabnikov je mogoče pridobiti zaupne informacije. Socialni inženirji izkoriščajo naravno naravnanost ljudi k zaupanju in se ne ukvarjajo z luknjami v računalniški varnosti. Splošno velja, da so uporabniki tudi v varnosti najšibkejši člen.

4 Seznam uporabljenih virov

[1] Stamp M., Information secrity, principles and practice

[2] Vidmar T., Informacijsko komunikacijski sistem

[3] Wikipedija, prosta spletna enciklopedija, http://en.wikipedia.org/wiki/, maj 2006

[4] Tomažič S., Varnost v telekomunikacijah in kako jo zagotoviti, http://www.lkn.fe.uni-lj.si/lknpub/Clanki/2003/Varnost%20v%20telekomunikacijah.pdf

[5] Slovar telekomunikacij, http://www.ltfe.org/, maj 2006

[6] Evroterm, večjezična terminološka zbirka, http://www.gov.si/evroterm/, maj 2006

5 Seznam uporabljenih kratic

Kratica	Angleški izraz	Slovenski izraz
AES	advanced encryption standard	Napredni standard za šifriranje
COA	Ciphertext-Only Attack	Napad na osnovi šifropisa
CCA	Chosen-Ciphertext Attack	Napad na osnovi izbranega šifropisa
CPA	Chosen-Plaintext Attack	Napad na osnovi izbranega čistopisa
DES	data encryption standard	Standard za šifriranje podatkov
DFA	Differential Fault Analysis	Diferenčna analiza okvare
HFE	Hidden Field Equations	Enačbe skritega polja
IV	initialization value	Inicializacijska vrednost
KPA	Known-Plaintext Attack	Napad na osnovi poznanega
MITM	man-in-the-middle	Vmesni človek
NDS	New Data Seal	Novi podatkovni pečat
OAEP	Optimal Asymmetric Encryption Padding	Optimalno asimetrično šifrirno zapolnjevanje
PKCS	Public Key Cryptography	Šifriranje z javnim ključem
PDA	Differential Power Analysis	diferenčna analiza moči
PPTP	Point-to-Point Tunneling Protocol	Protokol za tuneliranje med dvema točkama
RSA	Rivest Shamir ADleman	
SSL	Secure Socket Layer	sloj varnih vtičnic
TKIP	temporal key integrity protocol	protokol neokrnjenosti začasnega ključa
VPN	Virtual Private Network	Navidezno zasebno omrežje
WEP	Wired Equivalent Privacy	Zasebnost kot v žičnem omrežju
Wi-Fi	Wireless Fidelity	Brezžična vernost
WPA	Wi-Fi Protected Access	Zaščiteni dostop WiFi
XSL	eXtended Sparse Linearisation	razširjena raztresena linearizacija