Diferencialna analiza električne aktivnosti in protiukrepi

Kriptografija in računalniška varnost

Aleš Štimec
Fakulteta za računalništvo in informatiko,
Univerza v Ljubljani,
Ljubljana, SI-1001, Slovenia
Email: ales.stimec@fri.uni-lj.si

12. junij 2008

Kazalo

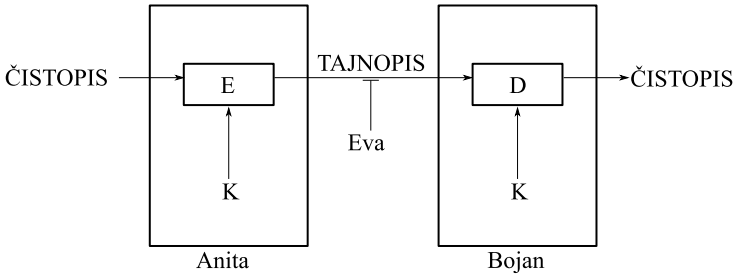
Povzetek

V pričujočem delu so podane osnove napadov na kriptosiste s pomočjo stranskega kanala s posebnim poudarkom na napadih s pomočjo analize elektrilne aktivnosti. Podan je tudi pregled protiukrepov, ki so na voljo snovalcem kriptosistemov, tako na področju prilagajanja programske kot tudi strojne opreme.

1 Uvod

Kriptografi so do nedavnega pri analizi varnosti kriptosistemov modelirali kriptografske algoritme kot idealne matematične objekte. Pri tem niso upoštevali pomanjkljivosti kriptografskih algoritmov, ki nastanejo zaradi implementacije le-teh na neki strojni opremi, saj je praktično nemogoče implementirati nek kriptografski algoritem na dani strojni opremi tako, da takšna implementacija ne bi oddajala v okolico nikakršnih informacij o svojem delovanju. Ravno to spoznanje pa je vodilo k razvoju novega razreda napadov, ki ga imenujemo *kriptoanaliza s pomočjo stranskega kanala* (angl. *side-channel cryptoanalysis*).

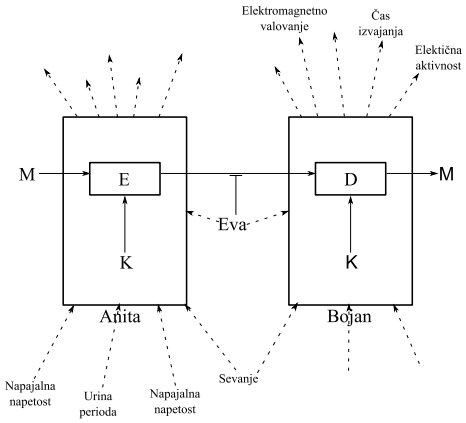
Za začetek si poglejmo klasični model kriptosistema. Spoznajmo Anito in Bojana, ki si želita izmenjati sporočilo, in radovedno Evo, ki bi rada to sporočilo prebrala ter, če bi se ponudila priložnost, sporočilo tudi spremenila (shema sistema je prikazana na sliki 1). Anita in Bojan bosta svoja sporočila šifrirala. Eva pa ima popoln dostop do vseh tajnopisov, ki si jih parček izmenja, pozna šifro, ki jo uporabljata, po možnosti ima nekaj parov (čistopis,tajnopis), ne pozna pa tajnih ključev, ki jih uporabljata Anita in Bojan (Kerckhoffov princip [20]). Če uporabljena šifra uspešno prestane vse Evine napore, da bi pridobila tajna ključa, se pri takšnem modelu kriptosistema reče, da je šifra „varna“.



Slika 1: Klasični model kriptografije, kjer si Anita in Bojan želita izmenjati sporočilo, radovedna Eva pa bi ga rada prebrala.

Takšen model kriptografskega sistema pa ne upošteva na začetku omenjenega dejstva: implementacija šifre, ki teče na neki strojni opremi, izdaja v okolje pomembne informacije o svojem delovanju, prav tako pa lahko različni faktorji iz okolja vplivajo na delovanje strojne opreme (glej sliko 2). Eva lahko z zajemom informacij, ki jih strojna oprema pri svojem delovanju oddaja v okolje (t.i. *stranski kanal*), pridobi dodatne podatke o šifri. Tako lahko meri potek porabe električne energije strojne opreme [24], na katerem teče šifriranje, elekotromagnetno valovanje [17, 19, 40], ki ga ta oprema oddaja, ali pa meri čas izvajanja posameznih operacij med šifriranjem [13, 23]. Prav tako lahko Eva vpliva na pogoje, v katerih ta strojna oprema deluje, kot so temperatura okolja,

napajalna napetost, razna sevanja in dolžina urine periode in s tem povzroči napake v delovanju strojne opreme, ki ji izdajo informacije o uporabljenem ključu [7, 8]. Zaradi izrednega razvoja kriptoanalize s pomočjo stranskega kanala, morajo snovalci novih kriptosistemov upoštevati model kriptografskega sistema, ki upošteva informacije o delovanju strojne opreme izdane v okolje, in hkrati upoštevati tudi vpliv okolja na delovanje strojne opreme, kot prikazuje slika 2.



Slika 2: Izboljšan model kriptografije

V nadaljevanju sledi kratek pregled različnih vrst napadov s pomočjo stranskega kanala s posebnim poudarkom na analizi električne aktivnosti. V poglavju 3 je podrobneje opisana diferencialna analiza električne aktivnosti na primeru DES[1]. V poglavju 4, pa sledi pregled protiukrepov, ki jih lahko uporabijo snovalci kriptografskih algoritmov skupaj z razvijalci strojne opreme, na kateri se ti algoritmi izvajajo, da bi otežili diferencialno analizo električne aktivnosti.

2 Pregled kriptoanalize s pomočjo stranskega kanala

Ideja za napad na šifro s pomočjo stranskega kanala je pravzaprav starejša od računalnikov in tiskanih vezij. Verjetno nam je vsem dobro znan prizor vlomilca, ki poskuša vlomiti v bančni sef tako da s pomočjo stetoskopa posluša delovanje notranjega mehanizma ključavnice. Izdajalski kliki, škrtanja in premiki mehanizma so prav tako stranski kanal, ki vlomilcu pomaga do pravilne kombinacije in ga pripelje do vsebine sefa.

V nadaljevanju pa poglejmo, kako lahko izkoristimo informacije, ki jih v okolje izdaja strojna oprema, na kateri teče nek kriptografski algoritem, da bi se dokopali do uporabljenih tajnih ključev.

2.1 Časovna kriptoanaliza

Idejo, da je čas izvajanja operacij pri šifriranju možno uporabiti kot stranski kanal, je prvi predlagal Koch leta 1996 [23] (prvi osnutek članka je kot opozorilo

[1] Data Encription Standard

proizvajalcem in uporabnikom kriptografskih algoritmov objavil 7.12.1995), kar je v tistem času prispelo celo na prvo stran New York Times-a [29]. Čas šifriranja se lahko razlikuje zaradi:

- uporabe predpomnilnika v pomnilniški hierarhiji,
- optimizacije, ki jo uporabljata prevajalnik in strojna oprema, s katero se med izvajanjem izognemo nepotrebnim inštrukcijam,
- pogojnih stavkov, ki vodijo v različne poti izvajanja programa,
- procesorskih inštrukcij, katerih čas izvajanja je odvisen od operandov.

Na prvi pogled se morda zdi, da je količina informacije, ki jo pridobimo z merjenjem časa izvajanja, majhna, a Kocher pokaže, da zadostuje za odkritje tajnega ključa pri DES, RSA in DSS. Leta 2005 pa je Bernstein [6] pokazal, da je mogoče s pomočjo znanih čistopisov odkriti tajni ključ AES mrežnega strežnika, ki teče na oddaljenem računalniku (za kar pravi, da je kriva sama šifra AES in ne kakšna specifična implementacija).

Da bi preprečili kriptoanalizo te vrste je možnih kar nekaj protiukrepov. Lahko, na primer, zahtevamo, da se vse inštrukcije na strojni opremi izvajajo enako dolgo, kar postane neučinkovito, saj smo odvisni od časa izvajanja najpočasnejše inštrukcije. V program lahko dodamo naključne zakasnitve, kar poveča količino podatkov, ki jih mora napadalec zajeti, da bi prišel do tajnega kluča.

2.2 Kriptoanaliza na podlagi napak

Predstavljajmo si, da Anita in Bojan uporabljata enostavno kriptografsko napravo, katere izhod je ali čistopis ali tajnopis, odvisno od nekega bita v registru. Če se vrednost tega bita slučajno spremeni (npr. če spominsko celico, ki hrani ta bit, zadene kozmični žarek), se na izhodu takšne kriptografske naprave namesto tajnopisa pojavi čistopis. Če bi Evi nekako uspelo povzročiti zamenjavo tega bita, bi lahko na izhodu naprave brala čistopise sporočil, ki si jih parček pošilja.

Seveda je zelo malo verjetno, da bi napad te vrste deloval na kakšnem resničnem kriptografskem sistemu. Še posebej zato, ker so načrtovalci sistemov že od samih začetkov skrbeli za robustnost kriptografskih naprav, ki so odporni na manjše napake, za katere niso menili, da bi lahko ogrozile varnost sistema.

Biham in Shamir [7] sta leta 1997 predstavila napad znan kot *diferencialna analiza napak* na DES, v katerem na podlagi napake na enem bitu v R_{15} najdeta tajni ključ. Istega leta je Boneh s sodelavci [8] objavil analizo napak na implementacijo RSA, Rabinov sistem podpisov in nekatere avtentikacijske protokole (kriptosistemi z javnim ključem). V članku pokažejo, da je na podlagi dveh podpisov sporočila (en pravilen in en podpis z napako) možno odkriti pošiljateljev zasebni ključ. Podrobne opise napadov s pomočjo diferencialne analize napak najdemo v [11, 33].

Proti kriptoanalizi na podlagi napak je možnih več protiukrepov. Ena izmed možnosti je preverjanje pravilnosti tajnopisa, preden se le-ta pojavi na izhodu, kar podvoji delo, ki ga mora opraviti kriptografska naprava. Seveda pa se tudi pojavi možnost, da pri samem preverjanju pravilnosti pride do napake. Druga možnost je ščitenje kriptografske naprave proti zunanjim vplivom, kot

so razna sevanja, napačna napajalna napetost, napačna dolžina urine periode, spremembe v zunanji temperaturi, itd.

2.3 Kriptoanaliza na podlagi elektromagnetnih signalov

Prve primere uporabe elektromagnetnih signalov kot stranskega kanala najdemo že med prvo svetovno vojno, kjer sta obe strani prisluškovali telefonskim pogovorom sovražnika. Poljski telefoni, ki so jih uporabljale enote na fronti so bili povezani z eno žico, medtem ko je drugi vodnik zaradi prihranka pri teži predstavljala zemlja in kmalu so opazili, da lahko s pomočjo lastnega telefona slišijo pogovore nasprotnikov, ki so bili vkopani nekaj deset metrov stran. Odkritje je vodilo k postavljanju prisluškovalnih postaj in, seveda, k pospešenemu razvoju protiukrepov.

Po drugi svetovni vojni so se širile govorice, da se tajne službe ukvarjajo z izkoriščanjem informacij, ki jih izdajajo elektromagnetni signali, katere oddajajo kriptografske naprave med svojim delovanjem. Leta 1985 je van Eck [40] pokazal, kako je možno s komercialno dostopno opremo sestaviti prisluškovalni sistem, ki lahko obnovi sliko z zaslona oddaljenega do en kilometer. Od takrat je sledilo več raziskav [17, 19, 34], med drugimi tudi IBMov članek [1] leta 2003 v katerem demonstrirajo, kako je možno elektromagnetne signale izkoristiti za napad na pametne kartice.

Pred kratkim pa je NSA [14] (angl. *National Security Agency*) umaknila oznako zaupnosti z nekaterih dokumentov programa TEMPEST[2], ki potrjujejo domneve o raziskavah tajnih služb glede izkoriščanja tega stranskega kanala.

Elektromagnetni signali lahko izdajo celo več informacij o tajnih ključih, kot električna aktivnost, katere opis sledi v naslednjem poglavju, saj obstaja v posameznem vezju več virov elektromagnetnih signalov. Informacija, ki jo nosi nihanje električne napetosti, je na voljo tudi v obliki elektromagnetnih signalov zaradi fizike delovanja polprevodniških elementov. Poleg tega pa nekateri protiukrepi, ki otežujejo napade na podlagi električne aktivnosti ne preprečujejo napadov na podlagi elektromagnetnih signalov.

Možni protiukrepi proti tej vrsti napadov vključujejo oklepljanje kriptografskih naprav (Faradayeva kletka), namerno oddajanje elektromagnetnega šuma, naključno izvajanje kriptografskega algoritma, itd.

2.4 Analiza električne aktivnosti

Ta razred napadov so leta 1998 prvi predstavili Kocher, Jaffe in Jun [25].

2.4.1 Uvod v analizo električne aktivnosti

Večina današnjih digitalnih vezij je zasnovanih na CMOS[3] tehnologiji. Zato si za začetek poglejmo karakteristike porabe električne energije pri tej tehnologiji. CMOS vezja so sestavljena iz dveh vrst tranzistorjev. Tip P, ki prevaja, ko je na vhodu logična 0, in tip N, ki prevaja, ko je na vhodu logična 1. Med najpomembnejše gradnike CMOS vezij spada inverter (NOT vrata), ki je prikazan na sliki 3. Opazimo, da pri vhodu 1 prevaja spodnji tranzistor, medtem ko zgornji

[2] Telecommunications Electronics Material Protected from Emanating Spurious Transimssions

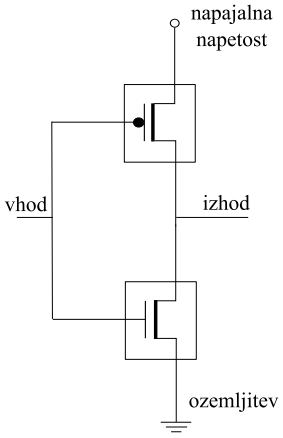
[3] Complementary metal–oxide–semiconductor

tranzistor ne prevaja in na izhodu dobimo vrednost 0. Pri vhodu 0 prevaja zgornji tranzistor in na izhodu dobimo vrednost 1. Pri zamenjavi vrednosti na vhodu pride do nedefiniranih vmesnih stanj, v katerih prevajata oba tranzistorja, in posledično do izgubnih tokov zaradi kratkega stika med napajalno napetostjo in ozemljitvijo.

Kot vidimo skozi gradnike CMOS vezij v statičnem stanju (ko je vhod ali 0 ali 1) teče zelo majhen električni tok. Prihaja pa do dinamične porabe pri preklopih stan, ko se vrata CMOS obnašajo kot majhen kondenzator (s tipično kapaciteto reda $C = 10fF$). Vendar pa je zaradi velikega števila vrat in velike frekvence pri kateri delujejo vezja dinamična poraba toka lahko relativno velika. Če si, na primer, predstavljamo manjši mikroprocesor z milijon logičnimi vrati, dolžino urine periode $dt = 5ns$ (frekvenca $200MHz$) in napajalno napetostjo $V_c = 3,3V$ izračunamo povprečno porabo toka po enačbi

$$I \sim \frac{\#vrat * (C * V_c)}{dt} = \frac{1000000 * (10pF * 3.3V)}{5ns} = 6.6A\ .$$

Podrobne informacije o CMOS vezjih in porabi električnega toka so zainteresiranemu bralcu na voljo v [42].



Slika 3: CMOS NOT vrata.

Opazimo, da je poraba električnega toka neposredno odvisna od števila preklopov, ki se opravijo v vezju, kar pa je odvisno od operacije, ki se trenutno opravlja, operandov, nad katerimi se operacija izvaja, vsebine registrov, podatkovnih vodil in spomina. Torej lahko z merjenjem porabe toka kriptografske naprave razkrijemo informacije o njenem delovanju, operacijah, ki jih naprava izvaja, operandih, nad katerimi se operacije izvajajo, dostopih do spomina...

Tipično merimo porabo toka kriptografske naprave tako da med napravo in ozemljitev vstavimo upornik (običajno $R = 47\Omega$) in s pomočjo digitalnega osciloskopa merimo nihanje napetosti na njem med šifriranjem. Nihanje porabe toka lahko nato izračunamo s pomočjo Ohmovega[4] zakona. Sled nihanja napetosti ali toka (obe sta ekvivalentni in se razlikujeta zgolj za nek konstanten faktor) nato uporabimo za kriptoanalizo.

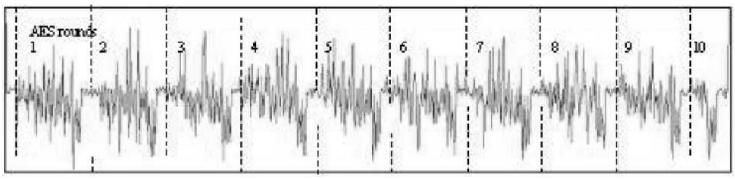
[4]Ohmov zakon pravi, da je tok $I = \frac{V}{R}$, kjer V izmerjena napetost in R upor vstavljenega upornika.

V naslednjih poglavjih sledita opisa dveh razredov metod za pridobitev tajnih informacij z analiziranjem električne aktivnosti kriptografske naprave.

2.4.2 Enostavna analiza električne aktivnosti

Enostavna analiza električne aktivnosti je metoda po kateri dobimo informacijo o tajnem ključu neposredno iz sledi nihanja napetosti med šifriranjem. Napadalec mora poznati podrobnosti implementacije šifre, da se lahko osredotoči na določen del sledi nihanja napetosti, v katerem se vrši operacija neposredno na tajnem ključu in poskuša iz sledi nihanja napetosti priti do informacije, ki bi mu razkrila tajni ključ. V splošnem velja, da je lahko vsaka operacija, ki deluje neposredno na tajnem ključu, potencialna točka za napad s pomočjo enostavne analize električne aktivnosti.

Primer sled nihanja napetosti pri AES je prikazan na sliki 4. Opazimo, da je sled zaradi preklapljanja velikega števila tranzistorjev dokaj šumna, vendar lahko kljub temu razberemo posamezne iteracije AES vključno z začetnim in končnim delom.



Slika 4: Primer sledi nihanja napetosti pri AES.

Če lahko napadalec določi kje na sledi nihanja napetosti se izvrši določena operacija nad določenim bajtom tajnega ključa, lahko s pomočjo izmerjene napetosti določi Hammingovo težo[5] tega bajta. Če napadalec pozna Hammingove teže vseh k podnizov, ki sestavljajo tajni ključ, se število možnosti, ki jih mora preizkusiti zmanjša z 2^{nk} na

$$\left[\sum_{m=0}^{n} \binom{n}{m} / 2^n\right],$$

kjer n označuje število bitov v posameznem podnizu tajnega ključa. Dokaz in podrobnejšo razlago lahko bralec najde v [31]. Vzemimo za primer implementacijo DES na 8-bitnem procesorju, kjer je tajni ključ dolžine 7 bajtov ($n = 8$, $k = 7$). Če napadalec pozna Hammingovo težo vseh sedmih bajtov ključa, potem mora pri izčrpnem iskanju namesto 2^{56} preiskati le še 2^{40} možnih ključev. Seveda je to samo enostaven primer in pri šifrah, ki uporabljajo daljši ključ ali pri trojnem DES šifriranju, nam poznavanje Hammingove teže posameznih podnizov ključa ne pomaga bistveno pri izčrpnem iskanju tajnega ključa.

Protiukrepi proti tej vrsti napadov vljučujejo zmanjšanje variacij v porabi električnega toka med posameznimi operacijami in izogibanje pogojnim skokom, ki so neposredno odvisni od vrednosti tajnega ključa. Zmanjšanje variacij v porabi električnega toka lahko dosežemo z uporabo diferencialnega CMOS vezja pri kateri se izhod vrat vedno pojavi tudi v negirani obliki ali z implementacijo šifre

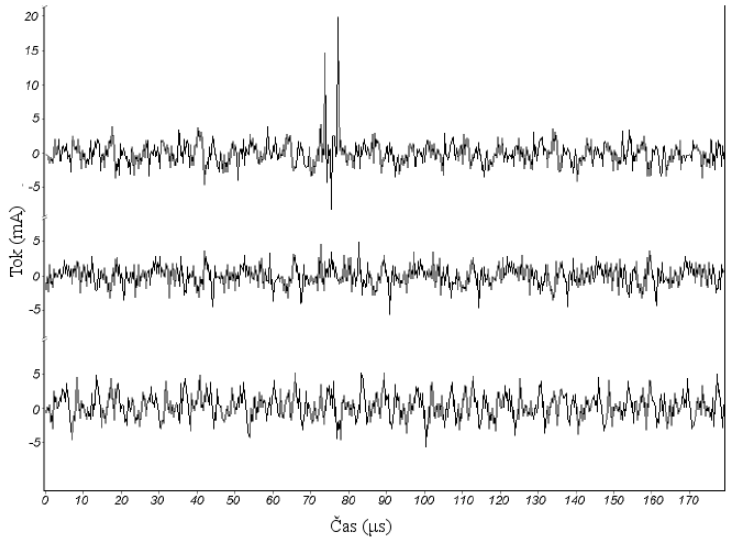
[5] Hammingova teža je enaka številu enic v binarnem nizu.

s pomočjo strojne opreme (npr. FPGA - Field Programmable Gate Arrays, PLD - Programmable Logic Devices) namesto implementacije s programsko opremo.

2.4.3 Diferencialna analiza električne aktivnosti

Medtem ko lahko napadalec s pomočjo enostavne analize električne aktivnosti enostavno razbere grobe značilnosti implementacije šifre, mora še vedno poznati podrobnosti implementacije, da bi lahko določil, v katerem trenutku se izvajajo operacije, ki mu lahko pri napadu koristijo. Pri tem pa mu delo dodatno otežuje še šum, ki je vedno prisoten pri meritvah. S pomočjo večjega števila sledi nihanja električne napetosti in statistične analize pa lahko razkrijemo bolj specifične informacije o delovanju kriptografske naprave kljub prisotnosti šuma in nepoznavanju posameznih podrobnosti implementacije.

Pri diferencialni analizi električne aktivnosti napadalec razdeli množico sledi nihanja električne napetosti s pomočjo preroka (angl. *oracle*) na dve množici in potem na podlagi razlike med množicama poskuša preveriti preroka.



Slika 5: Primer diferencialnih sledi nihanja električne napetosti. Na vrhu je diferencialna sled izračunana s pravilnim prerokom, spodnji dve sledi pa sta izračunani z nepravilnim prerokom.

Pri tej vrsti napadov ima napadalec tipično na voljo čistopise in/ali tajnopise ter množico sledi nihanja napetosti $S_1, S_2, \ldots, S_N$, ki jih je zbral z meritvami. Število zaporednih meritev k v posamezni sledi je odvisno od frekvence zajemanja, spominske zmogljivosti napadalčeve strojne opreme ter trajanja šifriranja. Prerok, ki ga napadalec uporablja, razdeli množico sledi na dve podmnožici na podlagi izbranega bita b pri izbrani vmesni vrednosti pri šifriranju

$$\mathcal{S}_0 = \{S_i : b = 0\}$$
$$\mathcal{S}_1 = \{S_i : b = 1\} \ .$$

V naslednjem koraku napadalec izračuna povprečno sled nihanja električne na-

petosti za obe podmnožici

$$\bar{\mathcal{S}}_0 = \frac{1}{|\mathcal{S}_0|} \sum_{S_i \in \mathcal{S}_0} S_i$$
$$\bar{\mathcal{S}}_1 = \frac{1}{|\mathcal{S}_1|} \sum_{S_i \in \mathcal{S}_1} S_i \ ,$$

kjer je $|\mathcal{S}_0| + |\mathcal{S}_0| = N$. Z odštevanjem obeh povprečij

$$\delta_b = \bar{\mathcal{S}}_0 - \bar{\mathcal{S}}_1 \ ,$$

pa dobi diferencialno sled nihanja električne napetosti. Nato se mora napadalec na podlagi direfencialne sledi nihanja električne napetosti odločiti, ali je prerok pravilen. Če množico sledi naključno razdelimo na dve podmnožici, se razlika med povprečnima sledema obeh podmnožic približuje ničli, ko se število izmerjenih sledi približuje neskončnosti. Torej, če je prerok napačen in je množico sledi nepravilno razdelil na dve podmnožici, se bo predvidena vrednost bita b razlikovala od dejanske vrednsti bita b v približno polovici sledi in veljalo bo

$$\lim_{N \to \infty} \delta_b \approx 0 \ ,$$

ker se bodo deli, ki niso korelirani z predvideno vrednostjo bita b zmanjševali z $\frac{1}{\sqrt{N}}$ [25]. Če pa prerok pravilno napove vrednost bita b in na podlagi tega pravilno razdeli množico sledi na podmnožici, bo diferencialna sled enaka vplivu, ki ga ima bit b na porabo električnega toka. Graf diferencialne sledi bo imel pri pravilnem preroku izrazite maksimume, kjer se operacije vršijo neposredno na izbranem bitu b.

Tu se napadalec sooča z dvema problemoma. Najprej mora na vsaki diferencialni sledi zaznati maksimume, nato pa mora med diferencialnimi sledmi vseh prerokov izbrati sled z najbolj izrazitimi maksimumi. Pri tem pa mu delo otežuje vseprisotni šum, ki ga v lahko razdelimo na pet razredov:

1. *eksterni šum*, ki je posledica šuma v napajalni napetosti, urini periodi, ...
2. *intrinzični šum*, ki je posledica premikov elektronov po vodnikih v vezju,
3. *kvantizacijski šum*, ki je posledica digitalizacije analognega signala,
4. *vzorčni šum*, ki nastane zaradi diskretnega vzorčenja signala in
5. *algoritmični šum*, ki je posledica variacij podatkovnih bajtov, nad katerimi se izvajajo operacije.

Intrinzični in kvantizacijski šum sta majhna. Eksterni šum lahko napadalec odpravi s pravilno uporabo meritvene opreme in filtriranjem. Vzorčni šum lahko zmanjša z manjšo urino preiodo zajemanja podatkov. Pri algoritmičnem šumu pa mora izbrati naključne vhodne podatke (čistopise), da se algoritmični šum s povprečenjem izniči.

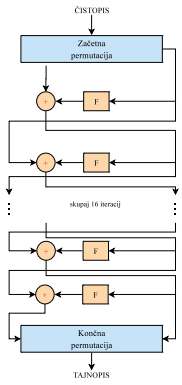
Primeri diferencialnih sledi nihanja električne napetosti so prikazani na sliki 5. Na vrhu je diferencialna sled izračunana s pravilnim prerokom, sledita pa ji dve sledi izračunani z nepravilnim prerokom. Na sledi, izračunani s pravim prerokom opazimo izrazite vrhove, kjer se operacije vršijo neposredno nad izbranim bitom b.

3 Kriptoanaliza DES z diferencialno analizo električne aktivnosti

V prejšnjih poglavjih so bile razložene osnove diferenciale analize električne aktivnosti, sedaj pa si oglejmo, kako bi jo izvedli na primeru DES.

3.1 DES

DES (angl. *Data Encryption Standard*) [32] so razvili v sedemdesetih letih prejšnjega stoletja pri IBM. Že od samega začetka je bil algoritem tarča pozornosti mnogih raziskav, predvsem zaradi relativno majhne dolžine tajnega ključa in suma o vpletenosti ameriške nacionalne agencije za varnost (angl. NSA - National Security Agency) v njegov razvoj. Kljub temu da danes DES ne velja za varno šifro, se v številnih različicah (predvsem trojni DES) še vedno uporablja v mnogih aplikacijah (npr. v slovenskih zdravstvenih karticah).



Slika 6: Osnovna struktura algoritma DES

Osnovna struktura DES je prikazana na sliki 6. Najprej se na čistopisu izvede začetna permutacija, ki ji sledi šestnajst iteracij, v katerih se vmesni tajnopis razdeli na levo (L_i, kjer indeks i označuje številko iteracije) in desno polovico (R_i), nad njimi pa se izvedejo sledeče operacije:

$$L_i = R_{i-1} \tag{1}$$

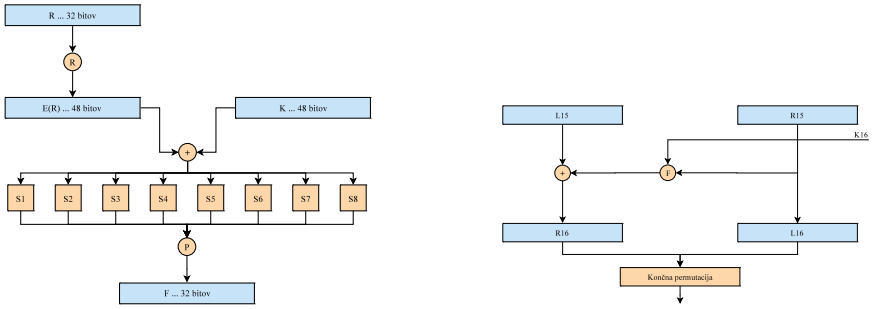
$$R_i = L_i \oplus F\left(R_{i-1}, K_i\right), \tag{2}$$

$$\text{kjer je } F\left(R_{i-1}, K_i\right) = P(S(E(R_{i-1} \oplus K_i) \text{ (glej sliko 7a)}), \tag{3}$$

kjer K_i predstavlja podključ v i-ti iteraciji. Zadnja iteracija se razlikuje od ostalih, saj je pri tej leva in desna polovica vmesnega tajnopisa ne zamenjata (glej sliko 7b)), ampak se nad njima izvrši še zaključna permutacija, da dobimo končni tajnopis.

3.2 Diferencialna analiza električne aktivnosti DES

Predstavljajmo si nepridiprava, ki je pridobil našo pametno kartico, na kateri teče algoritem DES, na njej pa je shranjuje tudi naš tajni ključ, do katerega bi se nepridiprav rad dokopal. Recimo, da ima na voljo dovolj časa, dober digitalni



Slika 7: Slika 7a) prikazuje podrobnješo zgradbo funkcije F, ki se izračuna v vsaki iteraciji. Slika 7b) pa prikazuje zadnjo iteracijo DES, ki se rahlo razlikuje od prejšnjih.

osciloskop in vso ostalo potrebno opremo ter znanje, da lahko za množico izbranih čistopisov izmeri sledi nihanja električne napetosti ter prebere tajnopise.

Sedaj mora nepridiprav izbrati, kakšne preroke bo uporabil pri napadu. Ker pri zadnji iteraciji DES velja $L_{16} = R_{15}$ (glej sliko 7b)) in ker napadalec pozna zaključno premutacijo pri DES, lahko za vsak tajnopis izračuna, kakšen je bil vhod R_{15} v funkcijo F. Iz slike 7a) je razvidno, da se v funkciji F 48-bitni ključ K_i v vsaki iteraciji razdeli na 8 6-bitnih blokov. Napadalec se torej osredotoči na določen 6-bitni podključ v zadnji iteraciji, saj bi lahko na podlagi znane vrednosti ključa izračunal vrednost nekega bita b v vmesnem tajnopisu L_{15}.

Zato sestavi 2^6 prerokov, od katerih vsak prerok na podlagi ene izmed možnih vrednosti ključa razdeli množico sledi nihanja napetosti na dve podmnožici:

1. $\mathcal{S}_0$, v kateri so sledi nihanja napetosti za katere je na podlagi izbrane vrednosti ključa izračunana vrednost bita b enaka 0 in

2. $\mathcal{S}_1$, v kateri so sledi nihanja napetosti za katere je na podlagi izbrane vrednosti ključa izračunana vrednost bit b enaka 1.

Nato za obe množici izračuna povprečno sled nihanja $\bar{\mathcal{S}}_0$ in $\bar{\mathcal{S}}_1$ ter na podlagi razlike med obema še diferencialno sled nihanja napetosti δ_b. Na koncu pregleda diferencialne sledi nihanja napetosti vseh prerokov in izbere tistega preroka, ki ima v diferencialni sledi nihanja napetosti najbolj izrazite vrhove, ter tako najde pravilno 6-bitno vrednost podključa.

Enako lahko napadalec sedaj stori še za ostalih 7 podključev in tako pridobi celoten ključ K_{16}, ki je bil uporabljen v zadnji iteraciji. Da bi pridobil celotni 56-bitni tajni ključ, ki je shranjen v pametni kartici, mora sedaj ponoviti postopek na predhodnih iteracijah šifriranja in tako poiskati manjkajoče vrednosti tajnega klju;a, ki se izpustijo v postopku generiranja ključev za posamezno iteracijo (za vsako iteracijo se iz dveh 28-bitnih zamaknjenih polovic začetnega ključa izbere po 24 bitov, ki skupaj tvorijo 48-bitni ključ v dani iteraciji).

Iz povedanega je razvidno, da lahko tak nepridirav z ustrezno opremo, ki dandanes niti ni tako draga, brez večjih težav pridobi tajni ključ iz naše nezaščitene pametne kartice. Očitno je, da bi bile takšne pametne kartice v današnjem času, ko se hitro povečuje procesorska moč dostopnih računalnikov, hkrati pa se ceni

merilna oprema, povsem neuporabne. Zato se je po razkritju teh pomanjkljivosti pojavilo veliko raziskav, kako preprečiti uhajanje informacij o tajnih ključih preko porabe električne energije in nekatere od teh protiukrepov bomo opisali v naslednjem poglavju.

4 Protiukrepi

Pri snovanju kriptosistemov moramo ves čas imeti v mislih njihovo implementacijo na dejanski strojni opremi, da bi preprečili napade s pomočjo analize električne aktivnosti. Protiukrepi vključujejo tako prilagoditev programske opreme kot tudi prilagoditev strojne opreme, na kateri bo kriptografski algoritem izvajan.

4.1 Prilagajanje programske opreme

Pri zapisu kriptografskega algoritma v obliki programa, ki bo izvajan na dani strojni opremi (npr. pametni kartici), lahko pisci z določenimi ukrepi otežijo delo napadalcu, ki bi se rad dokopal do tajnega ključa na podlagi enostavne ali diferencialne analize električne aktivnosti.

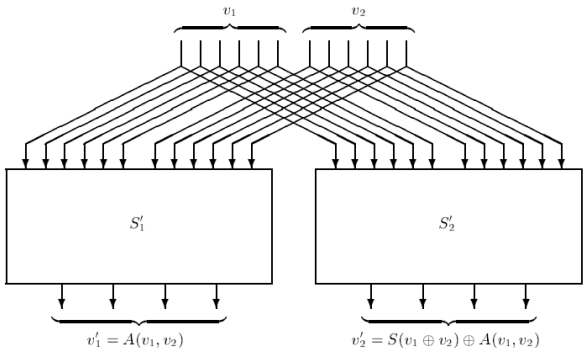
Prvi in očiten ukrep je, da omejimo število šifriranj, ki jih je na neki strojni opremi možno izvesti [10]. Tako lahko, na primer, omejimo število autentikacij, ki jih je možno izvesti na neki pametni bančni kartici, na neko dovolj veliko število, ki pa napadalcu ne omogoča, da bi zbral zadostno število meritev za uspešno izvedbo napada. Tipičnemu uporabniku bančne kartice bi tako lahko zadoščalo milijon avtentikacij v življenski dobi kartice, kar pa morda ne bi zadoščalo za uspešen napad, še posebej, če uporabimo tudi nekatere protiukrepe, ki bodo navedeni v nadaljevanju. Prav lahko pa se zgodi, da ima napadalec fizičen dostop do kartice, dovolj znanja o zgradbi kartice ter primerno strojno opremo, da lahko s fizičnim posegom ponastavi tak števec avtentikacij na ničelno vrednost.

Drugi ukrep je uvajanje naključnih zakasnitev v program [9, 12], tako da se posamezne točke v meritvah porabe električne energije ne nanašajo več na iste ukaze v algoritmu. S tem otežimo delo napadalcu, saj mora sedaj najprej poiskati točke v vseh meritvah, ki ustrezajo izvajanju istih ukazov algoritma. Seveda je ta problem za napadalca relativno enostavno rešljiv, če uporabi sodobne tehnike digitalnega procesiranja signalov, zato moramo, če naj bi bil ta protiukrep zanesljiv, naključne zakasnitve uvesti v celoten tok programa in tako napadalcu ustrezno otežimo delo. Upoštevati moramo tudi, da dodajanje naključnih zakasnitev posledično privede tudi do povečanja časa izvajanja algoritma.

Še en popularen ad-hoc protiukrep je naključno zaporedje izvajanja algoritma [10], s čimer bi dosegli podoben učinek kot pri naključnih zakasnitvah. Če naj bi bil takšen protiukrep učinovit, bi morali celotno zaporedje ukazov izvajati naključno, kar pa je zaradi same zaporedne narave algoritmov praktično neizvedljivo. V nasprotnem primeru, bi lahko napadalec, tako kot v primeru naključnih zakasnitev, rekonstruiral pravilno zaporedje izvajanj, oziroma našel točke v meritvah, ki ustrezajo izvajanju določenega ukaza, s pomočno sodobnih tehnik digitalnega procesiranja signalov.

Če nam strojna oprema omogoča, lahko pri pisanju algoritma uporabimo strojne ukaze, pri katerih je poraba električne energije neodvisna od vhodnih vrednosti ali pa so vsaj variacije v porabi majhne. Žal ta protiukrep ni vedno na voljo, več o protiukrepih na podlagi strojne opreme pa sledi v naslednjem poglavju.

Goubin in Patarin [18] sta za šifro DES predstavila metodo, pri kateri vsako vmesno vrednost V nadomestimo s k spremenljivkami $V_1, \ldots, V_k$, tako da velja $V = f(V_1, \ldots, V_k)$. Pri primeru predstavljenem v članku nadomestijo vsako vrednost V z vrednostima V_1 in V_2, tako da velja $V = V_1 \oplus V_2$, nato pa pokažeta, kako se izvajajo operacije na tako maskiranih spremenljivkah. Večina operacij je trivialno izvedljivih, zanima pa je njihova rešitev za izračun substitucije S (angl. *S-box*), pri kateri uvedejo dve novi substituciji, tako da se nemaskirane vrednosti nikoli ne pojavijo v čisti obliki (glej sliko 8). Pri tem je z A označena naključna in tajna transformacija.



Slika 8: Goubin in Patarin pri računanju substitucije na maskiranih spremenljivki $v = v_1 \oplus v_2$ vpeljeta dve novi substituciji, tako da se pri računanju nemaskirani vrednosti v in v' nikoli ne pojavita v spominu..

V [3] sta Akkar in Giraud predstavila metodo transformiranega maskiranja (angl. *trasformed masking method*) za DES. Osnovna ideja njihove metode je, da se vsi izračuni opravijo na vrednostih, ki so maskirane z naključno masko. Prav tako modificirajo substitucije S tako da so vhodi in izhodi maskirani z isto masko.

Pomanjkljivost obeh metod je povečan čas izvajanja algoritma, kot tudi povečana prostorska zahtevnost, kar je lahko odločilnega pomena pri kriptografskih sistemih, kjer so strojne zmogljivosti omejene (npr. pri pametnih karticah). Kljub zaščiti proti napadom s pomočjo diferencialne analize električne aktivnosti pa ti metodi ne zagotavljata varnosti pred napadi s pomočjo diferencialne analize višjega reda [24]

Da bi omogočila zaščito pred diferencialno analizo višjega reda sta Akkar in Giraud predstavila izboljšano metodo maskiranja [4], v kateri maskirajo vse vrednosti krajše od dvaintrideset bitov, ki so odvisne od tajnega ključa, pri tem pa poskrbijo, da se vmesne vrednosti nikoli ne maskirajo z istimi maskami. Kmalu za tem so Akkar in sodelavci [2] objavili napad na poprej omenjeno metodo maskiranja in predlagali izboljšave. Leta 2006 je Lv v [26] opisal štiri nove napade na izboljšano metodo maskiranja predlagano v [2]. V [27] pa so

avtorji podali ter dokazali pet zahtev, ki jim mora zadostiti implementacija DES, da bi s pomočjo maskiranja zagotovili varnost pred napadi s pomočjo diferencialne analize električne aktivnosti:

1. vse kritične vmesne vrednosti v algoritmu morajo biti maskirane z neko naključno masko,
2. vse xor-ane izhodne vrednosti substitucij S v pri in zadnji iteraciji morajo biti maskirane z neko naključno masko,
3. xor-ani izhodi substitucij S v prvih dveh (zadnjih dveh) iteracijah morajo biti maskirane z neko naključno masko,
4. xor-ani izhodi substitucij S v drugi in zadnji (prvi in predzadnji) iteraciji DES morajo biti maskirani z neko naključno masko,
5. xor-ani izhodi substitucij S v prvih dveh ter zadnji (zadnji in prvih dveh) iteraciji DES morajo biti maskirani z neko naključno masko.

V istem članku prav tako dokažejo, da za varno maskiranje implementacije DES zadoščajo tri 32-bitne maske ter šest dodatnih substitucij S, ki jih je potrebno ustvariti pri vsakem šifriranju.

4.2 Prilagajanje strojne opreme

Izvor vseh težav pri preprečevanju analize električne aktivnosti je lastnost strojne opreme, da preko porabe eletrične energije izdaja v okolje informacije o svojem delovanju. Torej bi bil logični protiukrep pri snovanju kriptosistemov uporaba strojne opreme, ki v svoje okolje preko porabe eletrične energije ne izdaja informacij o svojem delovanju. Žal takšna strojna oprema ne obstaja, saj se pri vsakem vezju nekaj energije nujno porabi za obdelavo podatkov [5], vendar je možno z raznimi ukrepi količino izdane informacije zmanjšati. Seveda je takšna strojna oprema kompleksnejša in posledično tudi dražja.

Tako lahko namesto CMOS tehnologije, ki je najpogosteje uporabljana pri kriptografskih napravah, uporabimo tehnologijo SABL (angl. Sense Amplifier Based Logic) [37, 38], pri kateri se v vsakih vratih zgodi en preklop ne glede na vhodno vrednost (tudi v primeru, ko se vrednost na vhodu ne spremeni). Posledično imajo ta vezja $3,5-$ do $4.5-$krat večjo porabo na enoto površine kot CMOS vezja. Zaradi dvojnih ciklov (cikel predhodnega nastavljanja vrednosti ter izvrševalni cikel) pa imajo ta vezja za faktor dva zmanjšano zmogljivost glede na CMOS vezja. Podobna je tehnologija WDDL (angl. Wave Dynamic Differential Logic style) [39], ki ima glede na CMOS 3.5 večjo porabo električne energije glede na površino vezja in približno dvakrat zmanjšano zmogljivost.

Kljub neodvisnosti porabe električne energije od vhodnih vrednosti pri navedenih tehnologijah, je napad na podlagi diferencialne analize električne aktivnosti še vedno mogoč zaradi medsebojnih vplivov posameznih elementov in povezav v vezjih, vendar pa mora napadalec zbrati bistveno več meritev kot pri tehnologiji CMOS.

Tudi asinhronska vezja [21] imajo nekatere zanimive lastnosti za snovanje varnih kriptografskih sistemov. Eksperimentalni rezultati [16] kažejo, da je količina informacij izdana na podlagi porabe električne energije manjša kot pri

ustreznih sinhronskih vezjih, vendar pa to zmanjšanje ni zadostno, da bi preprečili napade. V primeru napadov na podlagi elektromagnetnega sevanja nam odsotnost urinih signalov celo olajša napad, saj se s tem odstrani šum, ki ga ti signali generirajo. Vendar pa je prednost asinhronskh vezij v majhni porabi v primerjavi s sinhronskimi CMOS vezji.

Glede na to, da ne poznamo tehnologije, pri kateri bi bila poraba vrat neodvisna od vhodnih vrednosti, je naslednji korak, da s filtriranjem odstranimo nihanja napetosti, ki so posledica izvajanja operacij na vmesnih vrednostih kriptografskega algoritma. Pri tem lahko uporabimo pasivne ali aktivne filtre. Pri izbiri moramo upoštevati, da veljajo za pasivne filtre omejitve, ki izhajajo iz omejene velikosti kondenzatorjev, ki jih še lahko umestimo na čip [36]. Pri aktivnih filtrih, pa je težava v zamudi, ki se pojavlja pri filtriranju in pušča odkrite številne lastnosti nihanja porabe električne napetosti, ki napadalcu omogočajo uspešen napad.

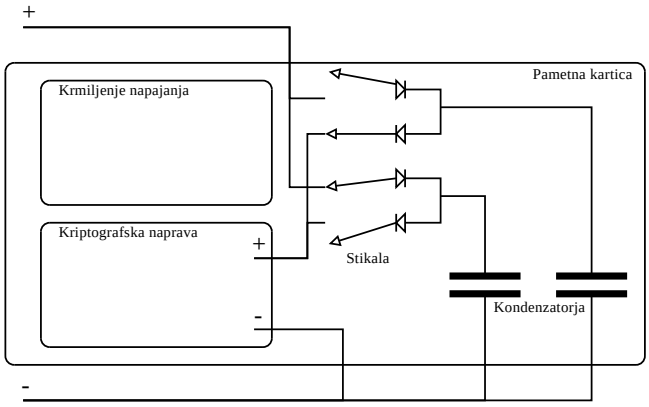
Tako je, na primer, Rakers s sodelavci [35] izdelal filter za brezkontaktno pametno kartico, ki je vstavljen med digitalno vezje, na katerem se izvaja algoritem, in oddajno anteno. S filtriranjem je uspel zmanjšati raven signala za 66dB in ker mora za vsake 3dB, za katere zmanjšamo moč izdajalskega signala napadalec zbrati dvakrat več meritev, torej mora sedaj napadalec zbrati kar 2^22 krat več meritev. To pa v praksi pomeni, da bo njegov napad namesto ene minute trajal več kot sedem let.

Zanimiva je tudi možnost, ki jo je opisal Shamir [36], pri kateri ločimo napajanje od vezja, na katerem se izvajajo kripografski algoritmi. Predlagal je rešitev, pri kateri za napajanje vezja skrbita dva kondenzatorja, tako da se izmenično eden polni, drugi pa medtem napaja kriptografsko napravo (glej sliko 9). Pri tem je predlagal uporabo kondenzatorjev velikosti 0.1 mikrofaradov, ki so bili v tistem času dostopni v izvedbi velikosti $2 \times 2 \times 0.4$ mm, kar omogoča vgradnjo v pametne kartice standardne velikosti ob zanemarljivem dodatnem strošku pri masovni proizvodnji. Kondenzatorja te velikosti bi ob povprečnem toku 5 miliamperov zadoščala za 20 milisekund napajanja, pri čemer bi napetost na kondenzatorju padla za 1 V. Tako bi pri frekvenci ure 5 MHz takšno napajanje zadoščalo za izvršitev 100 ukazov, kar bi pomenilo, da bi napadalec ob merjenju poteka porabe električne energije imel na voljo skupno porabo stotih zaporednih ukazov, kar je manj informativno od prvotnega poteka porabe električne energije, a še vedno omogoča napadalcu uspešno izvedbo napada.

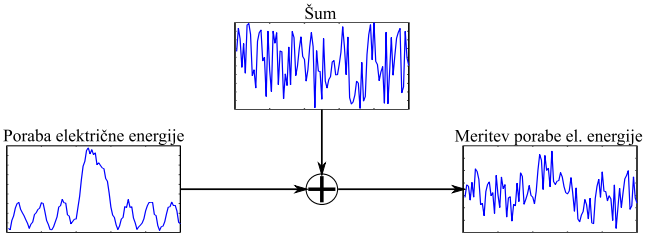
Da bi napadalcu popolnoma onemogočili napad, bi bilo potrebno v enoto za krmiljenje napajanja vgraditi vezje, ki bi napetost na kondenzatorju zmanjšalo na neko vnaprej določeno vrednost v nekem konstantnem časovnem obdobju, pred ponovnim polnjenjem. Prav tako je potrebno upoštevati, da lahko napetost na vhodno/izhodnih povezavah pametne kartice izda stanje napetosti na kondenzatorjih, kar enostavno rešimo tako, da prepovemo izvajanje vhodno/izhodnih operacij med izvajanjem občutljivih delov kriptografskih algoritmov. Na ta način bi bile meritve, ki bi jih zbral napadalec, popolnoma neuporabne za napad s pomočjo diferencialne analize električne aktivnosti.

Prav tako je možno napadalcu otežiti delo s tem, da v kriptosistem vgradimo vezje, ki porablja naključno količino električne energije, kar je bilo omenjeno že v enem prvih člankov o napadu s pomočjo analize električne aktivnosti [41]. S tem dodamo meritvam šum, ki napadalcu oteži delo, saj mora zbrati veliko več meritev, da se šum pri povprečenju izniči (glej sliko 10).

V prejšnjem poglavju je bilo opisano maskiranje na nivoju programske opreme



Slika 9: Shamir je predlagal ločitev napajalnega dela od vezja na katerem se izvajajo kriptografski algoritmi.



Slika 10: Napadalcu lahko otežimo delo tako, da v kriptosistem vgradimo vezje, ki naključno porablja električno energijo.

kot eden možnih protiukrepov. Tako se je pojavila ideja, da bi implementirali maskiranje že na nivoju strojne opreme. Tako ne bi električni vodnik nosil vrednosti, ki bi bila korelirana s kakšno vmesno vrednostjo algoritma. Očitno bi bil tak pristop splošnejši od maskiranja na nivoju programske opreme. Ko bi enkrat razvili shemo za varno maskiranje na nivoju strojne opreme, bi lahko avtomatično generirali maskirana vezja, ki bi varno implementirala poljuben kriptografski algoritem, kar bi bistveno razbremenilo snovalce kriptosistemov.

Predlaganih je bilo več shem za varno maskiranje na nivoju strojne opreme. Tako je Messerges s sodelavci [30] predlagal zamenjavo MUX vrat, ki se uporabljajo pri implementaciji nelinearnih operacij (npr. substitucije), z maskiranimi MUX vrati, ki so sestavljena iz treh navadnih MUX vrat. V [22] pa avtorji predlagajo aritmetično-logično enoto, v kateri so vse osnovne operacije zaščitene z eno ali večimi naključnimi maskami, ki pripadajo posameznim maskiranim vratom.

Večina varnostnih analiz maskirnih shem je temeljila na predpostavki, da vsi vhodni signali maskiranih vrat prispejo hkrati. Mangard in sodelavci [28] pa so pokazali, da ta predpostavka ne drži, zaradi česar pride v eni urini periodi do večih prehodov na izhodu vrat, medtem ko se vhodne vrednosti spreminjajo. Kot so pokazali v članku so ti prehodi (angl. *glitches*) korelirani z vmesnimi vrednostmi vsakič, ko vhodne vrednosti ne prispejo hkrati, zaradi česar jih je možno izkoristiti za napad na kriptosistem. Zato je nadvse pomembno, da sno-

valci kriptositemov, ki bodo implementirani na strojni opremi z maskiranjem, upoštevajo tudi informacije, ki jih ti prehodi izdajajo potencialnim napadalcem [15].

5 Zaključek

V današnjem času se skokovito povečuje naša odvisnost od prenosnih kriptografskih naprav, ki so sposobne varno shranjevati tajne informacije. Tako se v vsakodnevnem življenju zanašamo na pametne kartice, ki so prisotne na vsakem koraku od bančnih storitev, do mobilne telefonije, zdravstvenega varstva in varovanja osebnih podatkov. Zato je nujno, da so takšne kriptografske naprave varne pred zlorabami nepridipravov, ki so tipično dobro tehnično podkovani in imajo na voljo primerno strojno opremo, saj bi imela vsakršna pomankljivost lahko resne in daljnosežne posledice.

Medtem ko je v preteklosti zadostoval matematični dokaz varnosti kriptosistema, se morajo danes snovalci kriptosistemov zavedati tudi implementacije algoritmov na dani strojni opremi in pomanjkljivosti, ki jih le-ta prinese. Z razkritjem razreda napadov s pomočjo stranskega kanala je postalo še kako pomembno, da imamo pri načrtovanju kriptosistema v mislih vse informacije, ki jih bo bodoči kriptosistem izdajal v okolje in ki jih potencialni napadalec lahko izkoristi za napad.

V pričujočem delu smo napravili kratek pregled napadov s pomočjo stranskega kanala s posebnim poudarkom na analizi električne aktivnosti. Podan je bil primer diferencialne analize električne aktivnosti na primeru DES. Prav tako so bile podani tudi protiukrepi, ki jih lahko uporabijo snovalci kriptosistemov skupaj s snovalci strojne opreme. Kot je razvidno, je področje protiukrepov proti analizi električne aktivnosti zelo aktivno raziskovalno področje, kar tudi priča o pomembnosti področja.

Prav izdelovalci pametnih kartic skrbno spremljajo omenjeno področje in kolikor je le mogoče skrivajo podrobnosti o svojih produktih ter protiukrepih, ki jih njihovi produkti vsebujejo. Še vedno pa je najbolje, da se uporabniki najbolj zanašamo nase in poskrbimo, da naše pametne kartice ne pridejo v roke nepridipravom.

Literatura

[1] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The em side-channel(s). In *CHES '02: Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, pages 29–45, London, UK, 2003. Springer-Verlag.

[2] M.-L. Akkar, R. Bevan, and L. Goubin. Two power analysis attacks against one mask method. In *Proceedings of the Fast Software Encryption 2004 FSE'04*, volume 3017 of *Lecture Notes on Computer Science*, Berlin, 2004. Springer-Verlag.

[3] M.-L. Akkar and C. Giraud. An implementation of des and aes, secure against some attacks. In *CHES '01: Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, pages 309–318, London, UK, 2001. Springer-Verlag.

[4] M.-L. Akkar and L. Goubin. A generic protection against high-order differential power analysis. In T. Johansson, editor, *FSE*, volume 2887 of *Lecture Notes in Computer Science*, pages 192–205. Springer, 2003.

[5] C. H. Bennet and R. Landauer. The fundamental physical limits of computation. *Scientific American*, 253(1):48–56, 1985.

[6] D. J. Bernstein. Cache-timing attacks on AES, April 2005 . http://cr.yp.to/antiforgery/cachetiming-20050414.pdf.

[7] E. Biham and A.. Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, pages 513–525, London, UK, 1997. Springer-Verlag.

[8] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults. *Lecture Notes in Computer Science*, 1233:37–51, 1997.

[9] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. A cautionary note regarding evaluation of AES candidates on smart-cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, Mar. 1999.

[10] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, pages 398–412, London, UK, 1999. Springer-Verlag.

[11] C.-N. Chen and S.-M. Yen. Differential fault analysis on AES key schedule and some coutnermeasures. In *ACISP*, pages 118–129, 2003.

[12] J. Daemen and V. Rijmen. Resistance against implementation attacks: A comparative study of the AES proposals. In *Proc. Second Advanced Encryption Standard Candidate Conf.*, Mar. 1999.

[13] J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestré, J.-J. Quisquater, and J.-L. Willems. A practical implementation of the timing attack. In *CARDIS '98: Proceedings of the The International Conference on Smart Card Research and Applications*, pages 167–182, London, UK, 2000. Springer-Verlag.

[14] Tempest documents. http://cryptnome.org/nsa-tempest.htm.

[15] W. Fischer and B. M. Gammel. Masking at gate level in the presence of glitches. In J. R. Rao and B. Sunar, editors, *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2005.

[16] J. J. A. Fournier, S. W. Moore, H. Li, R. D. Mullins, and G. S. Taylor. Security evaluation of asynchronous circuits. In C. D. Walter, Ç.K. Koç, and C. Paar, editors, *CHES*, volume 2779 of *Lecture Notes in Computer Science*, pages 137–151. Springer, 2003.

[17] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In *CHES '01: Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, pages 251–261, London, UK, 2001. Springer-Verlag.

[18] L. Goubin and J. Patarin. Des and differential power analysis (the "duplication"method). In *CHES '99: Proceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems*, pages 158–172, London, UK, 1999. Springer-Verlag.

[19] H. J. Highland. Electromagnetic radiation revisited. *Comput. Secur.*, 5(2):85–93, 1986.

[20] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX. pp. 5−38, Jan. 1883, pp. 161−191, Feb. 1883.

[21] J. Kessels. Applying asynchronous circuits in contactless smartcards. In *Proc. ACiD-WG Workshop*, Feb. 2000.

[22] F. Klug, 0. Kniffler, and B. Gammel. Rechenwerk, verfahren zum ausführen einer operation mit einem verschlüsselten operanden, Carry-Select-Addierer und kryptographieprozessor. Technical report, German Patent DE 10201449 C1, January 2001.

[23] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, pages 104–113, London, UK, 1996. Springer-Verlag.

[24] P. C. Kocher, J. Jaffe, and B. Jun. Introduction to differential power analysis and related attacks. Technical report, 1998.

[25] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, pages 388–397, London, UK, 1999. Springer-Verlag.

[26] J. Lv. On two des implementations secure against differential power analysis in smart-cards. *Inf. Comput.*, 204(7):1179–1193, 2006.

[27] J. Lv and Y. Han. Enhanced des implementation secure against high-order differential power analysis in smartcards. In C. Boyd and J. M. González Nieto, editors, *ACISP*, volume 3574 of *Lecture Notes in Computer Science*, pages 195–206. Springer, 2005.

[28] S. Mangard, T. Popp, and B. M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In Alfred Menezes, editor, *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.

[29] J. Markoff. Secure digital transactions just got a little less secure. *New York Times*, 11. December 1995.

[30] T. S. Messerges, E. A. Dabbish, and L. Puhl. Method and apparatus for preventing information leakage on microelectronic assembly. Technical report, US Patent 6,295,606, 25. Sep. 2001.

[31] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.*, 51(5):541–552, 2002.

[32] National Bureau of Standards. Data encryption standard. *FIPS*, 46, 1977.

[33] G. Piret and J.-J. Quisquater. A differential fault attack technique against SPN structures, withapplication to the AES and KHAZAD. In *Cryptographic Hardware and Embedded Systems*, pages 77–88, 2003.

[34] J.-J. Quisquater and D. Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *E-SMART '01: Proceedings of the International Conference on Research in Smart Cards*, pages 200–210, London, UK, 2001. Springer-Verlag.

[35] P. Rakers, L. Connell, T. Collins, and D. Russel. Secure contactless smartcard ASIC with DPA protection. In *Proc. IEEE Custom Integrated Circuits Conference*, May 2000.

[36] A. Shamir. Protecting smart cards from passive power analysis with detached power supplies. In *CHES '00: Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems*, pages 71–77, London, UK, 2000. Springer-Verlag.

[37] K. Tiri, M. Akmal, and I. Verbauwhede. A dynamic and differential cmos logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Proceedings of 28th European Solid-State Conference - ESSCIRC 2002*, pages 403–406, 2002.

[38] K. Tiri and I. Verbauwhede. Securing encryption algorithms against dpa at the logic level: Next generation smart card technology. *Cryptographic Hardware and Embedded Systems - CHES 2002, (C. D. Walter, C. K. Koc, C. Paar, eds.), Lecture Notes in Computer Science*, 2779:137–151, 2003.

[39] K. Tiri and I. Verbauwhede. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In *Proceedings of Design, Automation and Test in Europe Conference - DATE 2004*, pages 246–251. IEEE Computer Society, 2004.

[40] W. van Eck. Electromagnetic radiation from video display units: an eavesdropping risk? *Computer Security*, 4(4):269–286, 1985.

[41] P. Wayner. Code breaker cracks smart card's digital safe. *New York Times*, page C1, 22 June 1998.

[42] N. H. E. Weste and K. Eshraghian. *Principles of CMOS VLSI design: a systems perspective*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1985.