Univerza v Ljubljani

Fakulteta za računalništvo in informatiko

Magistrski študij Računalništvo in informatika

Projekt pri predmetu Kriptografija in računalniška varnost
pri doc. dr. Aleksandru Jurišiću

JERNEJ BARBIČ

AES - KRIPTOSISTEM ZA 21. STOLETJE

(AES - CRYPTOSYSTEM FOR THE 21st CENTURY)

Ljubljana, marec 2001

Kazalo

1 Uvod **3**

2 Izbor kriptosistema AES **4**
2.1 Natečaj inštituta NIST 4
2.2 Kriteriji pri izboru kriptosistema AES 4
2.3 Kratek opis kandidatov, ki so prišli v ožji izbor za standard AES 5
2.3.1 MARS 5
2.3.2 RC6 6
2.3.3 Rijndael 6
2.3.4 Serpent 6
2.3.5 Twofish 7

3 Rijndael - zmagovalec natečaja **7**
3.1 Velikosti ključev in blokov 7
3.2 Uporabnost Rijndaela v posameznih ciljnih okoljih (pametne kartice, osebni računalniki) 7
3.3 Opis algoritma Rijndael 8
3.3.1 Ponovitev več krogov šifriranja - analogija z DES-om 8
3.3.2 Štiri transformacije, ki sestavljajo posamezen krog šifriranja 9
3.3.3 Operacije s ključem 10
3.4 O implementaciji Rijndaela v praksi 11

4 Varnost Rijndaela **11**
4.1 Velikost prostora ključev - primerjava z DES-om 11
4.2 Diferencialna in linearna kriptoanaliza 12
4.2.1 Diferencialna kriptoanaliza 12
4.2.2 Linearna kriptoanaliza 14
4.2.3 Diferenčne in linearne sledi kriptosistema AES 15
4.3 Ocena skupne varnosti, upoštevajoč vse znane napade 17

5 Zaključek **17**

Literatura **18**

Opombe o literaturi **18**

1 Uvod

AES je kratica za Advanced Encryption Standard in predstavlja nov kriptografski standard, ki ga bodo na najrazličnejših področjih uporabljali milijoni ljudi. Izdelava standarda AES je za kriptografijo zelo pomemben dogodek, saj je plod večletnega sodelovanja vodilnih kriptografov z vsega sveta. AES je simetričen kriptosistem in neformalno velja za naslednika kriptosistema DES (Data Encryption Standard), ki je, skupaj s svojimi izpeljankami, verjetno najbolj pomemben simetričen kriptosistem vseh časov. Kriptosistem DES je bil zasnovan v 60. letih 20. stoletja, torej v obdobju računalniške prazgodovine, ko so računalniki bili bistveno manj zmogljivi kot danes in ko se je zdelo DES-ovih 56 bitov varnosti nepredstavljivo veliko. DES je ostal v veljavi izredno dolgo in šele v zadnjem desetletju 20. stoletja so se pojavili prvi praktično uspešni napadi nanj. Zaradi razvoja računalnikov je DES postajal vedno manj varen kriptosistem in je trenutno v fazi ukinjanja. Še nekaj časa pa bo ostala v uporabi njegova izboljšana različica trojni DES, ki danes še vedno velja za varen kriptosistem. Problem trojnega DES-a je, da uporablja kar 48 krogov šifriranja (angl. rounds), kar je bistveno več kot drugi simetrični kriptosistemi, ki ponujajo enak nivo varnosti. Zaradi tega je trojni DES prepočasen za prenos digitalnih videoposnetkov in drugih podatkov, ki potrebujejo hiter podatkovni kanal.

Avtor standarda AES je ameriška organizacija NIST (National Institute for Standards and Technology), ki deluje v okviru ameriške vlade. Gre torej za civilno inštitucijo, saj je AES civilni in ne vojaški standard. NIST je pri razvoju standarda sodelovala tudi z ameriško agencijo za nacionalno varnost NSA (National Security Agency). AES bo postal glavni simetrični kriptosistem, ki ga bo uporabljala ameriška vlada, poleg tega pa ga bodo uporabljale tudi mnoge inštitucije, organizacije in posamezniki izven ameriške vlade, bodisi v ZDA bodisi drugod po svetu. Znotraj ameriške vlade je prišlo do ostrih polemik o vprašanju internacionalizacije kriptografije, saj ima kriptografija strateški vojaški pomen. Znano je, da je šele pred kratkim prišlo do sprostitev embargov za izvoz kriptografskih produktov iz ZDA v druge države. Zaradi nezadržnega napredovanja globalizacije je namreč bilo kriptografsko znanje vse težje zadržati znotraj ZDA. Če bi AES omejili le na ZDA, bi s tem izgubili možnost, da ga preverijo kriptografi z vsega sveta. Ravno v primeru DES-a so namreč vodilno kriptoanalitično vlogo odigrali kriptografi z Japonske in Izraela.

Pri vsem tem se seveda postavlja vprašanje, kateri kriptografski algoritem naj bo izbran za AES. NIST je leta 1997 objavil javni razpis za izbiro algoritma, ki bo postal uradni standard AES. S tem je ameriška vlada povabila kriptografe z vsega sveta, da sodelujejo pri nastanku novega kriptografskega standarda. Po temeljitih analizah je NIST dne 2. oktobra 2000 objavil ime končnega zmagovalca. Standard AES bo postal algoritem Rijndael, ki sta ga razvila belgijska kriptografa Dr. Joan Daemen iz podjetja Proton World International in Dr. Vincent Rijmen, postdoktorski raziskovalec na oddelku za elektrotehniko katoliške univerze v Leuvenu v Belgiji. Ime algoritma je skovanka iz priimkov avtorjev. Avtorja predlagata naslednje tri alternativne različice izgovorjave imena algoritma: "Reign Dahl", "Rain Doll"in "Rhine Dahl". Algoritem je bil izbran na podlagi več kriterijev. Navedimo le glavne: varnost, preprostost implementacije, prozornost uporabljenih metod in hitrost izvajanja.

Ena od osnovnih razpisnih zahtev v natečaju za AES je bila, da morajo algoritmi biti

javno dostopni vsakomur brez plačil licenčnih. Uporaba algoritmov ne sme biti omejena z državnimi mejami ali drugimi patentnimi zahtevki. Na Internetu je tako na naslovu [Implementations] možno najti izvorno kodo več različnih implementacij algoritma Rijndael.

NIST je dne 28. februarja 2001 objavil uraden vzorčni predlog standarda AES (draft standard). Vzorčni standard bo do 28. maja 2001 na ogled splošni javnosti. V tem času lahko kdorkoli sporoči svoje komentarje ali zadržke o standardu AES. Več o tem je možno najti na strani [NIST]. AES bo postal uradni standard ameriške vlade predvidoma avgusta 2001. Takrat bo objavljen kot uradni ameriški zvezni FIPS standard (Federal Information Processing Standard) in začela se bo njegova uradna pot uporabe v ameriški vladi. Od standarda AES se pričakuje, da bo ostal v uporabi vsaj še 20 let, dokumenti, zašifrirani s pomočjo njega pa naj bi ostali varni še mnogo časa zatem, ko bo AES nekoč dokončno umaknjen iz uporabe.

2 Izbor kriptosistema AES

2.1 Natečaj inštituta NIST

NIST je dne 2. januarja 1997 objavil razpis za izbiro kriptografskega algoritma, ki bo postal standard AES. Izčrpno poročilo o razpisu je možno najti v [AES]. Glavni podatki o natečaju so bili povzeti tudi v [Dobbs]. Na razpis se je prijavilo 21 kandidatov iz različnih držav. Izbor algoritma AES je potekal v treh fazah. Prva faza je bila zaključena avgusta 1998. V njej je bilo določenih 15 algoritmov, ki so zadostili razpisnim pogojem. V drugi fazi, ki je bila zaključena marca 1999, je bilo izbranih 5 finalistov. To so algoritmi Rijndael (Belgija), MARS (ZDA), RC6 (ZDA), Serpent (Velika Britanija, Izrael, Norveška) in Twofish (ZDA). V zadnji fazi pa je oktobra 2000 izmed finalistov bil izbran zmagovalec (Rijndael).

Standard AES naj bi bil primeren za uporabo v čim širšem okolju. Specifikacija algoritma je morala biti javno dostopna po vsem svetu, algoritem sam pa ni smel biti zaščiten s patenti. NIST je postavil naslednje tri glavne zahteve za algoritem, ki bo implementiral AES:

1. Algoritem mora implementirati simetričen kriptosistem.
2. Algoritem mora biti bločna šifra (block cipher).
3. Velikost bloka mora biti 128 bitov. Algoritem mora podpirati velikosti ključev 128, 196 in 256 bitov.

2.2 Kriteriji pri izboru kriptosistema AES

NIST je pri izboru kriptosistema AES upošteval naslednje tri glavne kriterije:

1. varnost,
2. cena,

3. algoritmične lastnosti implementacije.

Pri tem je bila varnost najpomembnejši kriterij. Izbrani algoritem je moral biti varen pred vsemi znanimi napadi, moral je imeti solidno matematično ozadje, dobljeni tajnopis pa je moral biti čim bolj naključen.

Cena algoritma bistveno določa razmah uporabe algoritma v praksi. NIST je želel, da bi AES postal splošno uporabljan šifrirni algoritem po vsem svetu in to na čim več različnih računalniških sistemih. V ta namen uporaba algoritma ne bi smela biti omejevana s plačilom licenčnin. Druga pomembna faktorja, ki vplivata na ceno izvedbe algoritma, sta hitrost algoritma in prostorska zahtevnost algoritma. NIST je želel algoritem, katerega izvorna koda bi bila čim krajša, njegovo zahteve po pomnilniku pa čim skromnejše.

Zahteva o algoritmičnih lastnostih implementacije se nanaša na fleksibilnost algoritma, primernost za izvedbo pri obstoječi strojni in programski opremi in preprostost algoritma. Fleksibilnost algoritma pomeni, da je mogoče algoritem uporabljati tudi pri velikosti ključev in blokov, večjih od minimalno zahtevanih, da ga je mogoče varno in učinkovito implementirati v čim več različnih računalniških okoljih, da ga je mogoče uporabiti tudi kot tokovno šifro, z njim konstruirati zgoščevalno funkcijo ter ga uporabiti za druge kriptografske algoritme.

Med izbirnim postopkom se je pokazalo, da se trije glavni kriteriji pogosto prepletajo. Zato sta bila cena in algoritmične lastnosti implementacije obravnavana skupaj, kot sekundarni kriterij za varnostjo.

2.3 Kratek opis kandidatov, ki so prišli v ožji izbor za standard AES

V tem podrazdelku bomo opisali pet finalistov natečaja NIST. Pripomnimo, da gre za robustne in zrele algoritme, od katerih bi po mnenju NIST vsak lahko zadostno služil kot standard AES. Vsi algoritmi veljajo za zelo varne. Ni znano, da bi obstajal uspešen napad na katerega koli od njih. Vsi obstoječi napadi se nanašajo le na močno poenostavljene različice algoritmov. Več podatkov o finalistih je možno najti v [AES] in [Landau].

2.3.1 MARS

Algoritem MARS je predlagal IBM. Podobno kot kriptosistem DES ima tudi MARS ima *Feistelovo strukturo*. Feistelova struktura je ohlapen pojem, ki pomeni, da se biti trenutnega stanja razkosajo na več blokov. Nova vrednost nekaterih (enega ali pa tudi več) blokov je kar prepisana vrednost nekaterih starih blokov. Nova vrednost preostalih blokov pa je neka funkcija starih blokov in podključa kroga. Večina kriptosistemov, ki šifriranje opravlja v več krogih, za šifriranje ne uporablja neposredno glavnega šifrirnega ključa, ampak najprej s pomočjo glavnega šifrirnega ključa določi za vsak krog šifriranja posebej poseben ključ, ki mu rečemo *podključ kroga*. Pri DES-u se Feistelova struktura kaže v tem, da se $2t$-bitni vhodni niz na začetku i-tega kroga razdeli na t-bitna niza L_{i-1} in R_{i-1}. Novi polovici izračunamo po formulah $L_i := R_{i-1}$ in $R_i := L_{i-1} \oplus f(R_{i-1}, K_i)$, kjer je f ustrezna enkripcijska funkcija, K_i pa ustrezen *podključ kroga*.

MARS razdeli celoten blok dolžine 128 bitov na štiri besede dolžine 32 bitov. Pri vsakem od 32 krogov šifriranja ena od teh besed s pomočjo S-škatel modificira ostale

besede. S-škatle so bile izbrane s pomočjo zgoščevalne funkcije SHA-1 in s pomočjo decimalnega zapisa konstant e in π. Srednjih 16 krogov ima drugačno strukturo kot prvih in zadnjih 8.

MARS je kompleksen algoritem, kar je povzročalo težave pri ocenjevanju njegove varnosti. Vseeno velja prepričanje, da ima MARS še veliko rezerve pri varnosti. MARS je razmeroma neprimeren za uporabo v omejenih okoljih, ker ima velike zahteve po pomnilniku. Njegova zapletenost tudi otežuje neposredno izvedbo v strojni opremi.

2.3.2 RC6

RC6 je 20-krožni Feistelov kriptosistem, ki ga je predlagalo podjetje RSA Security Inc. Med vsemi finalisti je kriptosistem RC6 najbolj preprost. RC6 je izboljšava kriptosistema RC5, katerega varnost je bila v preteklosti že precej raziskana. Del teh analiz je možno ponovno uporabiti tudi za analizo kriptosistema RC6, kar je prispevalo k večjemu zaupanju v njegovo varnost. Glavni operaciji v algoritmu RC6 sta množenje celih števil in rotacija bitov celega števila, zato je njegova hitrost močno odvisna od specifičnega procesorja in programskega jezika. RC6 je v povprečju najhitrejši algoritem izmed finalistov na 32-bitnih računalnikih. Njegova glavna slabost je, da dešifriranje zahteva hranjenje vseh podključev naenkrat v pomnilniku, zaradi česar RC6 ni optimalen algoritem za tiste pametne kartice, kjer je potrebno izvajati tudi dešifriranje. Zaradi uporabe operacij množenja, rotacij in seštevanj, ki jih je v splošnem težko maskirati brez večjega padca hitrosti implementacije algoritma, je RC6 tudi precej občutljiv na napade z uporabo merjenja časa izvajanja algoritma in merjenja porabljene moči med izvajanjem.

2.3.3 Rijndael

Rijndael je zmagovalec natečaja, zato mu posvečamo celoten naslednji razdelek.

2.3.4 Serpent

Serpent so predlagali trije kriptografi iz Velike Britanije, Izraela in Danske. Serpent je izmed finalistov še najbolj podoben DES-u po svoji zgradbi. Sestavlja ga 32 krogov šifriranja. V vsakem izmed krogov šifriranja trenutnim podatkom bitno prištejemo podključ kroga, zatem podatke transformiramo s pomočjo S-škatel, na koncu kroga pa izvedemo še linearno transformacijo (ta vključuje operacijo XOR in rotacije bitov). Namen linearne transformacije je *difuzija bitov*, s katero dosežemo, da vsak vhodni bit v enaki meri vpliva na vse izhodne bite. V Serpentu že po treh krogih vsak vhodni bit vpliva na vsak bit vmesnega stanja.

Serpent uporablja 8 različnih fiksnih S-škatel. Vsaka S-škatla transformira 4 bite v 4 bite. V posameznem krogu blok podatkov dolžine 128 bitov razdelimo na 32 zaporednih skupin po 4 bite in na vsaki od 32 skupin uporabimo eno in isto S-škatlo. Vsaka od osmih S-škatel je uporabljena štirikrat in sicer v krogih $i, 8+i, 16+i, 24+i$ za nek $i \in \{1, 2, \ldots, 8\}$. S-škatle so bile zasnovane s pomočjo DES-ovih S-škatel in decimalnega razvoja števila $(\sqrt{5}-1)/2$.

Serpent velja za zelo varen kriptosistem z veliko varnostne rezerve. Njegova zgradba je razmeroma preprosta, njegove pomnilniške zahteve pa dovolj skromne za uporabo v

pametnih karticah. Glavna pomankljivost Serpenta leži v počasnosti izvajanja šifriranja in dešifriranja , saj je v splošnem v tem pogledu najslabši med finalisti. Manjši problem predstavlja tudi medsebojna programska nekompatibilnost šifriranja in dešifriranja, saj algoritma za nobenega od njiju ni mogoče niti delno uporabiti za drugega.

2.3.5 Twofish

Twofish je predlagalo ameriško kriptografsko podjetje Counterpane Systems. Twofish je Feistelov kriptosistem s 16 krogi šifriranja. Posebnost Twofisha je uporaba S-škatel, ki so odvisne od ključa, kar je nekoliko neobičajno in sproža pomisleke o varnosti pred napadom s pomočjo diferencialne kriptoanalize. S-škatle v Twofishu so bijektivne transformacije na bitnih nizih dolžine 8. V posameznem krogu šifriranja se 128-bitni vhod razdeli na štiri 32-bitne besede. Na prvih dveh besedah izvedemo določene rotacije bitov, nato pa vsako od njih transformiramo s pomočjo štirih S-škatel, ki so, kot že rečeno, odvisne od ključa. Vsako od obeh besed nato še transformiramo s pomočjo linearne transformacije, prištevanja konstante po modulu 2^{32} in bitnega prištevanja podključa. Zatem prvi besedi prištejemo tretjo besedo, drugi pa četrto. Prvo in drugo besedo še bitno rotiramo in tako dobimo novo prvo in novo drugo besedo. Nova tretja in četrta beseda pa sta originalni prva in druga beseda.

Twofish ima veliko stopnjo varnosti, bil pa je kritiziran zaradi prevelike kompleksnosti. Generiranje podključev je sicer precej počasno, ni pa potrebno vseh ključev hkrati hraniti v pomnilniku. Twofish je primeren za uporabo v pametnih karticah. Njegova prednost je, da sta šifriranje in dešifriranje praktično enaki operaciji. Njegovi slabosti pa sta povprečne možnosti implementacije v strojni opremi in velika občutljivost na napade z merjenjem časa izvajanja in merjenjem porabljene moči.

3 Rijndael - zmagovalec natečaja

V tem razdelku bomo podrobneje opisali algoritem Rijndael. Pri tem se bomo opirali na originalno specifikacijo [Rijndael], ki sta jo napisala avtorja sama.

3.1 Velikosti ključev in blokov

Rijndael podpira velikosti ključev 128, 196 in 256 bitov in velikosti blokov 128, 196 in 256 bitov. Velikost ključa in bloka je možno izbrati neodvisno.

3.2 Uporabnost Rijndaela v posameznih ciljnih okoljih (pametne kartice, osebni računalniki)

Rijndael je posebej primeren za uporabo na 8-bitnih računalnikih (pametne kartice) in 32-bitnih računalnikih (moderni osebni računalniki). V tem pogledu je Rijndael najbolj fleksibilen izmed vseh finalistov. Uporabnost v teh dveh ciljnih okoljih je bila namenoma izboljšana s posebno zasnovo algoritma.

3.3 Opis algoritma Rijndael

V tem podrazdelku bomo podali popolno specifikacijo algoritma Rijndael. Vsi izračuni v Rijndaelu potekajo v obsegu $GF(2^8)$. Vsak element tega obsega lahko opišemo z enim bajtom informacije, kar je ugodno za implementacijo. Povsod v Rijndaelu uporabljamo za predstavitev obsega $GF(2^8)$ polinomsko bazo, podano z nerazcepnim polinomom $X^8 + X^4 + X^3 + X + 1$. Algoritem na več mestih operira tudi s polinomi nad obsegom $GF(2^8)$, predvsem s polinomi stopnje kvečjemu 3. Vsak tak polinom nosi 32 bitov informacije, kar je ugodno za 32-bitne računalnike.

3.3.1 Ponovitev več krogov šifriranja - analogija z DES-om

Rijndael nima standardne Feistelove strukture, kot jo poznamo iz DES-a. Podobno kot pri DES-u pa šifriranje poteka v več zaporednih krogih. Število krogov Rijndaela je odvisno od velikosti bloka in velikosti ključa.

	$B = 128$	$B = 196$	$B = 256$
$K = 128$	10	12	14
$K = 196$	12	12	14
$K = 256$	14	14	14

Tabela 1: Število krogov šifriranja v algoritmu Rijndael v odvisnosti od velikosti bloka (B) in velikosti ključa (K).

Posamezen krog šifriranja je sestavljen iz naslednjih bijektivnih transformacij: ByteSub, ShiftRow, MixColumn, AddRoundKey. Vsak od krogov šifriranja ima enako strukturo z izjemo zadnjega, kjer transformacijo MixColumn izpustimo. To je podobno nezamenjavi levega in desnega bloka v zadnjem krogu šifriranja pri DES-u in služi enotnosti zgradbe dešifriranja in šifriranja. Pred prvim krogom šifriranja podatkom bitno prištejemo prvi podključ. V nasprotnem primeru bi napadalec lahko transformacije ByteSub, ShiftRow in MixColumn iz prvega kroga šifriranja enostavno "olupil stran", saj te transformacije niso odvisne od ključa.

Naj B označuje velikost bloka čistopisa. V Rijndaelu vhodni čistopis razdelimo na zaporedne bajte $b_1, b_2, \ldots, b_{B/8}$, te bajte pa potem organiziramo v *matriko vmesnega stanja*. Elemente te matrike lahko interpretiramo kot elemente obsega $GF(2^8)$, njena dimenzija pa je $4 \times (B/32)$. Matrika vmesnega stanja seveda vsebuje $4 \cdot (B/32) \cdot 8 = B$ bitov informacije. Ker B lahko zavzame le vrednosti 128, 196 in 256, so ustrezne velikosti matrike vmesnega stanja 4×4, 4×6 in 4×8. Bajte $b_1, b_2, \ldots, b_{B/8}$ razporedimo v matriko vmesnega stanja po vrsti na položaje $a_{11}, a_{21}, a_{31}, a_{41}, a_{12}, \ldots, a_{4,B/32}$. Organizacija vmesnega stanja v matriko je pomembna pri transformacijah ShiftRow in MixColumn. V psevdo-C notaciji ima celoten algoritem Rijndael naslednjo zgradbo:

```
Rijndael (State,CipherKey) // State = blok čistopisa
                           // na koncu Rijndaela je v State shranjen tajnopis
                           // CipherKey = ključ
{
KeyExpansion (CipherKey,Subkeys); // določimo Nr+1 podključev
```

```
                                    // Nr = število krogov
AddRoundKey (State,Subkeys[1]); // začetno prištevanje ključa
For ( i = 1; i < Nr ; i++ )
  Round (State, SubKeys[i+1]); // ponovitev Nr-1 krogov
FinalRound (State,SubKeys[Nr+1]); // zadnji krog šifriranja
} // tajnopis je vrnjen preko reference State
```

3.3.2 Štiri transformacije, ki sestavljajo posamezen krog šifriranja

V psevdo-C notaciji posamezen krog (razen zadnjega) izgleda takole:

```
Round (State,SubKey) // State = referenca na vmesno stanje šifriranja
                     // SubKey = podključ kroga
{
ByteSub(State); // analogija S-škatel, ki zagotovi nelinearnost
ShiftRow(State); // difuzijska plast
MixColumn(State); // difuzijska plast
AddRoundKey(State,SubKey); // prištevanje ključa
} // rezultat kroga je vrnjen preko reference State
```

Transformacija **ByteSub** na vsakem od elementov matrike vmesnega stanja neodvisno izvede po vrsti naslednji operaciji:

1. izračun inverza v $GF(2^8)$ (element 0 se pri tem preslika sam vase),
2. izračun afine transformacije nad $GF(2)$, podane z izrazom

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Transformacija **ShiftRow** predstavlja cikličen premik v levo vrstic trenutne matrike vmesnega stanja. Skupaj s transformacijo MixColumn imata vlogo difuzijske plasti. Vsako od štirih vrstic matrike vmesnega stanja premaknemo za različno število mest. Specifične vrednosti so odvisne od velikosti bloka, podaja pa jih naslednja tabela.

	C0	C1	C2	C3
$B = 128$	0	1	2	3
$B = 196$	0	1	2	3
$B = 256$	0	1	3	4

Tabela 2: Velikost cikličnega premika vrstice matrike vmesnega stanja pri transformaciji ShiftRow v odvisnosti od velikosti bloka (B). Pri tem Ci označuje premik $(i+1)$-ve vrstice matrike stanja, $i = 0, 1, 2, 3$.

Transformacija **MixColumn** je druga polovica difuzijske plasti. Vsak stolpec matrike trenutnega stanja se interpretira kot polinom stopnje kvečjemu 3 nad obsegom $GF(2^8)$. Polinom pomnožimo s fiksnim polinomom $c(X) = '03'X^3 + '01'X^2 + '01'X + '02'$ in reduciramo po modulu X^4+1. Štirje koeficienti dobljenega polinoma predstavljajo rezultat transformacije MixColumn. Pri tem oznaka *'ab'* pomeni tisti element obsega $GF(2^8)$, katerega bitna reprezentacija glede na v Rijndaelu izbrano polinomsko bazo ima v šestnajstiškem sistemu obliko *ab*. Ker je polinom $c(X)$ tuj proti polinomu X^4+1, je polinom $c(X)$ obrnljiv element kolobarja $GF(2^8)[X]/(X^4+1)$. Zato je transformacija MixColumn obrnljiva.

V zadnjem delu kroga izvedemo operacijo **AddRoundKey**, pri kateri trenutnemu stanju bitno prištejemo podključ kroga. To je edina transformacija v celotnem krogu, na katero vpliva ključ.

3.3.3 Operacije s ključem

Rijndael za šifriranje ne uporabi neposredno ključa, ampak najprej iz ključa izračuna podključe kroga. Teh je za ena več, kot je vseh krogov šifriranja. V vsakem krogu namreč potrebujemo en podključ kroga, poleg tega pa potrebujemo še en dodaten podključ za začetno šifriranje. Podključi kroga so definirani rekurzivno s pomočjo prejšnjih podključev kroga in ključa samega (procedura KeyExpansion). V grobem poteka ta procedura tako, da iz ključa generira tabelo razširitve ključa (key expansion table), nato pa to tabelo razkosa na potrebno število podključev krogov. Pripomnimo, da ima podključ kroga enako dolžino kot blok čistopisa, ki je v splošnem lahko različna od velikosti glavnega ključa. Procedura KeyExpansion ima v psevdo-C notaciji naslednjo obliko:

```
KeyExpansion (byte Key[4 * Nk], word W[Nb * (Nr + 1)])
{
  //začetek tabele razširitve ključa je kar glavni ključ
  for (i = 0; i < Nk; i++)
    W[i] = (key[4 * i], key[4 * i + 1], key[4 * i + 2], key[4 * i + 3]);

  // rekurziven izračun preostanka tabele razširitve ključa
  for (i = Nk; i < Nb * (Nr + 1); i++)
  {
    temp = W[i - 1];
    if (i % Nk == 0)
      temp = SubByte(RotByte(temp)) ^ Rcon[i/Nk];
    else if ((i % Nk == 4) && (Nk > 6))
      temp = SubByte(temp);
    W[i] = W[i - Nk] ^ temp;
  }
}
```

Procedura KeyExpansion prebere glavni ključ Key in vrne tabelo razširitve ključa W, ki je tabela 4-bajtnih besed. Pri tem *Nr* označuje število krogov šifriranja, *Nk* je enako bitni dolžini ključa, deljeni z 32, *Nb* pa je enako bitni dolžini bloka, deljeni z 32. Operacija RotByte predstavlja cikličen premik v levo bajtov v dani 4-bajtni besedi. Operacija

SubByte vsak bajt 4-bajtne besede transformira z Rijndaelovo S-škatlo. Elementi tabele Rcon so konstante kroga (round constants). Podane so s pravilom

$$\text{Rcon}[i] = (\text{RC}[i], \text{`00'}, \text{`00'}, \text{`00'}),$$

kjer je RC$[i]$ tisti element obsega $GF(2^8)$, ki ga glede na v Rijndaelu izbrano polinomsko bazo predstavlja polinom X^{i-1}. Iz tabele W dobimo podključe kroga, tako da tabelo W razrežemo glede na velikost podključev. Prvih $32 \cdot Nb$ bitov predstavlja prvi podključ kroga, naslednjih $32 \cdot Nb$ bitov drugi podključ kroga in tako naprej vse do zadnjega podključa kroga. Velikost tabele W je ravno prav velika, da je v njej shranjenih vseh $Nr + 1$ podključev kroga.

3.4 O implementaciji Rijndaela v praksi

Rijndael je bil zasnovan z namenom, da bi učinkovito deloval na 8-bitnih in 32-bitnih procesorjih. V obeh primerih je implementacija preprosta. V primeru 8-bitnih procesorjev izkoristimo bajtno zasnovo Rijndaela ($GF(2^8)$) in dejstvo, da ima polinom-množitelj $c(X)$ iz transformacije MixColumn razmeroma preproste koeficiente. Ker imajo pametne kartice razmeroma malo spomina, je ugodno dejstvo, da lahko podključe kroga z minimalno količino dodatnega računanja računamo sproti in nam jih torej ni potrebno vseh hkrati hraniti v pomnilniku.

V primeru 32-bitnih procesorjev pa izkoristimo dejstvo, da je edina nelinearna transformacija (ByteSub) izvedena prva, zato lahko celoten krog šifriranja enostavno realiziramo s pomočjo vnaprej izračunanih tabel. Skupna velikost teh fiksnih tabel je 4 Kb. Pri dešifriranju moramo vse operacije izvesti v obratnem vrstnem redu, zato tak pristop na prvi pogled ni mogoč. S preprostim premislekom pa lahko vidimo, da je struktura dešifrirnega kroga pravzaprav enaka kot struktura šifrirnega kroga. Ta lastnost je posledica podrobnega načrtovanja algoritma s strani avtorjev. Zato lahko za izvedbo dešifrirnega kroga uporabimo kar algoritem za šifrirni krog z manjšimi spremembami.

Rijndael je bil implementiran na več 8-bitnih procesorjih in več 32-bitnih procesorjih. Programska implementacija na 32-bitnem 200 MHz Pentiumu je pri velikosti ključa in bloka 128 bitov dosegla hitrost enkripcije 8.8 Mb/s.

4 Varnost Rijndaela

4.1 Velikost prostora ključev - primerjava z DES-om

Velikost prostora ključev kriptosistema AES je ogromna v primerjavi z velikostjo prostora ključev pri DES-u. Pri velikosti ključa 128 bitov je na voljo približno 3.4×10^{38} ključev, pri velikosti ključa 196 bitov približno 6.2×10^{57} ključev in pri velikosti ključa 256 bitov približno 1.1×10^{77} ključev. Za primerjavo navedimo, da je velikost prostora ključev kriptosistema DES 7.2×10^{16} ključev, kar pomeni, da je 128-bitnih AES-ovih ključev reda velikosti 10^{22} več, kot je DES-ovih ključev. Konec 90. let prejšnjega stoletja so se pojavili posebni namenski računalniki za iskanje DES-ovega ključa s pomočjo požrešne metode. Če bi takšen računalnik uporabili za iskanje 128-bitnega AES-ovega ključa, bi za to potrebovali 150000 milijard let, kar je 10000-krat več, kot je ocenjena starost vesolja.

Moč računalnikov se pri današnjem tempu razvoja tehnike tipično podvoji vsakih 18 mesecev. Moderni računalniki so danes ravno že dovolj hitri, da zmorejo v sprejemljivem času poiskati 56-bitni DES-ov ključ. Razumno je predpostaviti, da podaljšanje ključa za 1 bit pomeni dvakrat več dela za njegovo iskanje. Pod predpostavko, da je AES na enoto ključa enako varen kot DES na enoto ključa, bo varnost kriptosistema Rijndael s 128-bitnim ključem torej ogrožena šele po $(128 - 56) \times 1.5 = 108$ letih. Več podobnih primerjav in zaključkov o velikosti ključev je mogoče najti v zanimivem poročilu [Blaze], ki ga je napisalo več vrhunskih kriptografov.

4.2 Diferencialna in linearna kriptoanaliza

Diferencialna in linearna kriptoanaliza sta dva izmed najučinkovitejših znanih napadov na kriptosistem DES. V tem podrazdelku bomo pokazali, da je kriptosistem AES varen pred tema dvema napadoma. Pri tem bomo sledili specifikaciji [Rijndael] in doktorski disertaciji [Daemen] enega od avtorjev Rijdaela.

4.2.1 Diferencialna kriptoanaliza

Diferencialna kriptoanaliza je napad z izbranim čistopisom. Idejo diferencialne kriptoanalize je možno uporabiti za napad na vsak simetrični kriptosistem, ki pri šifriranju uporablja več krogov. Naj bosta x_1 in x_2 bloka čistopisa, y_1 in y_2 pa ustrezna tajnopisa. Predpostavimo, da poznamo vmesna tajnopisa z_1 in z_2 pred šifriranjem v zadnjem krogu, ali pa vsaj njuno bitno razliko $z_1 \oplus z_2$. Ker je en sam krog šifriranja razmeroma enostaven, je mogoče iz znane razlike $z_1 \oplus z_2$ sklepati o vrednosti določenih bitov podključa zadnjega kroga in s tem določiti neko množico podključev, med katerimi je tudi pravi podključ zadnjega kroga. Če bi ta postopek lahko ponovili za dovolj veliko število parov čistopis/tajnopis, bi lahko enolično rekonstruirali podključ zadnjega kroga. To je pri kriptosistemu DES dovolj za rekonstrukcijo glavnega ključa. Pri kriptosistemu AES za rekonstrukcijo glavnega ključa potrebujemo K bitov tabele razširitve ključa, kjer je K bitna dolžina glavnega ključa. Ker so podključi pri AES enako dolgi kot bloki, pri nekaterih kombinacijah dolžina ključa/dolžina bloka poznavananje posameznega podključa ne zadostuje za razkritje glavnega ključa. V tem primeru si lahko pomagamo tako, da s pomočjo zadnjega podključa tajnopis za en krog dešifriramo in s pomočjo enakega postopka kot prej poiščemo še predzadnji podključ. To v vsakem primeru zadostuje za rekonstrukcijo celotnega ključa.

Problem pri opisanem pristopu je, da je samo s pomočjo znanih čistopisov x_1, x_2 in tajnopisov y_1, y_2 zelo težko ali nemogoče določiti razliko vmesnih tajnopisov z_1, z_2. Diferencialna kriptoanaliza izkoristi dejstvo, da je pri nekaterih kriptosistemih za nekatere (redke) vrednosti $z' = z_1 \oplus z_2,\ x' = x_1 \oplus x_2$ verjetnost $P(z'|x')$ razmeroma zelo velika. Pripomnimo, da so v praksi te verjetnosti v absolutnem smislu precej majhne. Če imata čistopisa x_1 in x_2 razliko x' in je ustrezna razlika pred zadnjim krogom šifriranja res enaka z', lahko uporabimo sklep iz zgornjega odstavka, ki nam da neko (majhno) množico podključev, med katerimi je tudi pravi podključ. Tak par x_1, x_2 imenujemo *pravilen par*. Če pa razlika pred zadnjim krogom ni enaka z' (tak par x_1, x_2 imenujemo *napačen*) dobimo sledeč algoritmu iz prejšnjega odstavka napačno množico možnih podključev. Pokazati je

možno, da so elementi te množice enakomerno porazdeljeni med vsemi možnimi vrednostmi podključev [Daemen]. Pri kriptoanalizi za dane x_1, x_2, y_1, y_2 ne vemo, kateri od obeh primerov nastopi. Z dovolj velikim številom parom čistopis/tajnopis pa se statistično pravi podključ loči od drugih. V ta namen lahko definiramo razmerje signal/šum (signal/noise ratio). Predpostavimo, da imamo na voljo M parov čistopisov z razliko x' in ustreznih M parov tajnopisov. Potem je v povprečju $P(z'|x') \cdot M$ parov pravilnih in $M - P(z'|x') \cdot M$ parov napačnih. Ker je tipično $P(z'|x') \ll 1$, je pravilnih parov veliko manj kot napačnih parov. Predpostavimo, da je povprečno število predlaganih podključev enako γ. Vseh predlogov skupaj je tako γM. Naj bo podključ dolg ν bitov. Pri kriptosistemu AES je tako $\nu = B$, kjer je B bitna dolžina bloka, pri kriptosistemu DES pa je $\nu = 48$. Za vsak podključ preštejemo, kolikokrat je bil predlagan. Število predlogov S pravilnega podključa s pomočjo pravilnih parov imenujemo *signal*. Velja $S \approx P(z'|x') \cdot M$. Preostalih $\gamma M - P(z'|x')M$ predlogov pa je enakomerno porazdeljenih med vsemi 2^ν možnimi vrednostmi podključa. Vsaka izmed vrednosti tako prejme povprečno $N = (\gamma M - P(z'|x')M)/2^\nu$ predlogov. Količino N imenujemo šum, razmerje

$$S/N = \frac{P(z'|x')2^\nu}{\gamma - P(z'|x')} \approx \frac{P(z'|x')2^\nu}{\gamma}$$

pa razmerje signal/šum. Pravilna vrednost ključa tako dobi v povprečju $S+N$ predlogov, preostale napačne vrednosti pa v povprečju N predlogov. Za uspešnost kriptoanalize mora biti razmerje S/N dovolj veliko, po možnosti vsaj večje od 1, tako da dobi prava vrednost dvakrat več predlogov kot preostale vrednosti. V primeru, da je $P(z'|x') < 2^{1-\nu}$, je $S/N < 2/\gamma$. Ker je γ tipično precej večji od 2, od tod sledi, da je v tem primeru razmerje šum/signal majhno in je kriptosistem varen pred napadom z diferencialno kriptoanalizo.

Kako za dan kriptosistem ugotoviti, ali obstajajo kakšne vrednosti z', x' z veliko verjetnostjo $P(z'|x')$? Večina kriptosistemov je prezapletenih za neposredno iskanje takih vrednosti z', x'. Pomagamo pa si lahko s pomočjo *diferenčnih sledi*. Posamezni krogi šifriranja namreč niso tako zapleteni kot vsi krogi skupaj. Zato je lažje konstruirati pare razlik u', v', ki se raztezajo le čez en krog in ki imajo dovolj veliko verjetnost. Če nam uspe razliko v' preko ponovnega kroga (z dovolj veliko verjetnostjo) nadaljevati do nove razlike w' in tako naprej vse do končne vrednosti z', dobimo diferenčno sled $x' = x'_0, x'_1, x'_2, \ldots x'_{Nr-1} = z'$. Pri tem Nr označuje število krogov. Verjetnost diferenčne sledi $P(z'|x')$ je enaka

$$P(x'_1|x')P(x'_2|x'_1) \cdot \ldots \cdot P(z'|x'_{Nr-2}).$$

Ker je verjetnost $P(z'|x')$ enaka vsoti verjetnosti vseh diferenčnih sledi, ki vodijo od x' do z', velja da je $P(z'|x')$ večja od verjetnosti vsake diferenčne sledi. Za varnost kriptosistema je torej potreben pogoj, da ni diferenčnih sledi z visoko verjetnostjo. Ker so diferenčne sledi edini praktično izvedljiv način iskanja razlik x', z' z veliko verjetnostjo, v praksi ta pogoj štejemo tudi za zadosten.

V [Daemen] je možno najti rezultat, da je verjetnost diferenčne sledi približno enaka produktu razširitvenih razmerij vseh aktivnih S-škatel dane diferenčne sledi. Pri izdelavi kriptosistema AES je bila izbrana taka S-škatla, ki ima maksimalno razširitveno razmerje 2^{-6}. Vse te pojme bomo natančneje obravnavali v podrazdelku o diferenčnih in linearnih sledeh kriptosistema AES, kjer bomo pokazali (izrek (4.3)), da ima vsaka 4-krožna

diferenčna sled vsaj 25 aktivnih S-škatel. Iz teh dveh rezultatov potem sledi, da je verjetnost vsake diferenčne sledi dolžine 4 kvečjemu 2^{-150}. Vsaka 8-krožna diferenčna sled ima tako verjetnost manjšo od 2^{-300}. Največja uporabljena dolžina podključa pri AES je $\nu = 256$ bitov. Tako je izpolnjen pogoj varnosti $P(z'|x') < 2^{1-\nu}$ in je razmerje signal/šum že za osem krogov kriptosistema AES veliko manjše od 1. Zaradi večje varnosti pa so avtorji dodali še nekaj dodatnih krogov.

4.2.2 Linearna kriptoanaliza

Linearna kriptoanaliza je napad s poznanim čistopisom in tajnopisom. V tem smislu je linearna kriptoanaliza splošnejša od diferencialne kriptoanalize. Linearna kriptoanaliza je možna, če obstaja večkrožna korelacija med vhodnimi podatki, izhodnimi podatki in ključem. Razliko med verjetnostjo korelacije in 1/2 imenujemo *korelacijski koeficient*. Naključne (neuporabne) korelacije imajo verjetnost blizu 1/2 in korelacijski koeficient blizu 0. Korelacija se mora raztezati čez vse kroge šifriranja razen enega ali dveh. *Enokrožna korelacija* je linearna bitna enakost oblike

$$In[i_1, i_2, \ldots, i_r] \oplus Out[j_1, j_2, \ldots, j_s] = K[k_1, k_2, \ldots, k_t].$$

Pri tem oznaka $A[l_1, l_2, \ldots, l_u]$ pomeni $A[l_1] \oplus A[l_2] \oplus \ldots \oplus A[l_u]$. In označuje vhod kroga, Out izhod, K pa podključ kroga. Pri diferencialni kriptoanalizi večkrožne diferenčne razlike poiščemo s pomočjo enokrožnih diferenčnih razlik. Tako tudi pri linearni kriptoanalizi *večkrožne korelacije* sestavimo s pomočjo enokrožnih korelacij. Pravimo, da konstruiramo *linearno sled*. Enokrožne korelacije lahko sestavimo v linearno sled, če je izhod posamezne enokrožne korelacije enak vhodu naslednje. Poiskati moramo torej enokrožne korelacije naslednje oblike:

$$\begin{aligned}
I_0[i_{01}, i_{02}, \ldots, i_{0r_0}] &\oplus I_1[i_{11}, i_{12}, \ldots, i_{1r_1}] = K_1[k_{11}, k_{12}, \ldots, k_{1t_0}], \\
I_1[i_{11}, i_{12}, \ldots, i_{1r_1}] &\oplus I_1[i_{21}, i_{22}, \ldots, i_{2r_2}] = K_2[k_{21}, k_{22}, \ldots, k_{2t_1}], \\
&\vdots \\
I_{R-1}[i_{R-1,1}, i_{R-1,2}, \ldots, i_{R-1,r_R}] &\oplus I_R[i_{R1}, i_{R2}, \ldots, i_{Rr_R}] = K_R[k_{R1}, k_{R2}, \ldots, k_{Rt_R}].
\end{aligned}$$

Pri tem I_k označuje izhod k-tega kroga šifriranja, ki je hkrati tudi vhod $(k+1)$-vega kroga šifriranja. Oznaka K_k pomeni podključ k-tega kroga. Vidimo, da morajo indeksi bitov, ki nastopajo v izhodu posamezne enokrožne korelacije, biti enaki indeksom bitov, ki nastopajo v vhodu naslednje enokrožne korelacije. Enokrožne korelacije lahko seštejemo in tako dobimo večkrožno korelacijo

$$I_0[i_{01}, i_{02}, \ldots, i_{0r_0}] \oplus I_R[i_{R1}, i_{R2}, \ldots, i_{Rr_R}] = \sum_{i=1}^{R} K_i[k_{i1}, k_{i2}, \ldots, k_{it_i}]. \tag{1}$$

Predpostavimo, da imajo enokrožne korelacije korelacijske koeficiente $k_1, k_2, \ldots, k_R$. Preprosto je pokazati, da ima potem večkrožna korelacija (1) korelacijski koeficient

$$k = 2^{R-1} k_1 k_2 \cdot \ldots \cdot k_R.$$

Pokažimo zdaj, kako bi lahko večkrožno korelacijo z dovolj veliko absolutno vrednostjo korelacijskega koeficienta uporabili za napad na kriptosistem AES. Kot vemo, je kriptosistem AES sestavljen iz začetnega prištevanja ključa, ki mu sledi določeno število ponovitev krogov šifriranja. Opisali bomo različico napada, v kateri uporabimo korelacijo, ki se razteza čez vse kroge šifriranja brez začetnega prištevanja ključa. Pri tem bomo napadli ravno to začetno prištevanje ključa. Možna je tudi različica, kjer uporabimo korelacijo, ki se razteza čez vse kroge, razen čez zadnjega, in pri kateri napademo zadnji krog šifriranja.

Predpostavimo torej, da imamo korelacijo oblike (1), ki se razteza čez vse kroge šifriranja brez začetnega prištevanja ključa in ki velja z verjetnostjo p. S pomočjo te korelacije lahko določimo bite $i_{01}, i_{02}, \ldots i_{0r}$ tabele razširitve ključa. To storimo tako, da pretečemo vseh 2^r možnosti za te bite, pri čemer za vsako možnost M izvedemo naslednje izračune. Za vsak poznani par čistopis/tajnopis izračunamo bite vhoda I_0, tako da bitom $i_{01}, i_{02}, \ldots, i_{0r_0}$ čistopisa prištejemo bite, kot jih podaja trenutno preiskovana možnost M. Biti tabele I_R pa so kar biti tajnopisa. Tako lahko preštejemo, za kolikšen del poznanih parov čistopis/tajnopis ima leva stran enakosti (1) vrednost 0. Pri pravilni vrednosti bitov $i_{01}, i_{02}, \ldots, i_{0r_0}$ tabele razširitve ključa bo ta delež bodisi blizu vrednosti p, bodisi blizu vrednosti $1 - p$. Pri nepravilnih vrednosti bitov ključa pa bo precej bližje vrednosti $1/2$. Na ta način lahko določimo bite $i_{01}, i_{02}, \ldots i_{0r_0}$ razširitvene tabele ključa. Pripomnimo, da pri tem desne strani enakosti (1) nismo poznali, ovrednotili ali kako drugače obravnavali. Če je ta desna stran za konkretne vrednosti podključev (ki jih ne poznamo) enaka 0, bo zgoraj omenjeni delež težil k vrednosti p, v nasprotnem primeru pa k vrednosti $1 - p$. Z uporabo drugih korelacij lahko potem določimo še druge bite, dokler jih ni dovolj za rekonstrukcijo glavnega ključa.

Za kriptosistem AES bomo pokazali, da ni linearnih sledi z absolutno vrednostjo korelacijskega koeficienta večjo od $2^{B/2}$, kjer je B dolžina bloka čistopisa. To je splošno sprejet pogoj varnosti pred napadom z linearno kriptoanalizo [Rijndael]. V delu [Daemen] je pokazano, da je korelacijski koeficient linearne sledi približno enak produktu vhodno-izhodnih korelacij njenih aktivnih S-škatel. S-škatla kriptosistema AES ima maksimalno vhodno-izhodno korelacijo enako 2^{-3}. Iz glavnega izreka (4.3) naslednjega podrazdelka sledi, da je maksimalna korelacija 4-krožne linearne sledi enaka 2^{-75}. Korelacija poljubne 8-krožne linearne sledi je tako manjša od 2^{-150}. Vrednosti dolžine ključa B pri AES zasedajo vrednosti $128, 196, 256$, zato je varnostni pogoj $B/2 < 150$ v vsakem primeru izpolnjen.

4.2.3 Diferenčne in linearne sledi kriptosistema AES

V tem razdelku bomo ocenili dolžine in verjetnosti diferenčnih in linearnih sledi kriptosistema AES. Najprej definirajmo nekaj novih pojmov. Pri kriptosistemu AES so vmesna stanja med šifriranjem predstavljena z matriko vmesnega stanja, ki je matrika bajtov. Beseda sled naj v tem podrazdelku označuje bodisi diferenčno bodisi linearno sled. Za sled lahko za vsako vmesno stanje določimo *aktivne bajte*. Za diferenčno sled so to tisti bajti, kjer trenutna razlika ni enaka 0. Za linearno sled pa so to bajti, ki jih določa izbirni vektor vhodne tabele In. To so torej bajti, ki nastopajo v vhodu posamezne enokrožne linearne korelacije. S-škatle, ki delujejo na aktivnih bajtih, imenujemo *aktivne S-škatle*.

Razširitveno razmerje S-škatle je razmerje med tistimi vhodnimi pari s podano razliko x', ki jih S-škatla transformira v izhodne pare s predpisano razliko z', in vsemi vhodnimi pari z razliko x'. Podobno je *vhodno-izhodna korelacija* S-škatle razmerje med tistimi vhodnimi podatki, za katere dana linearna korelacija velja, in vsemi vhodnimi podatki. Skupek vseh aktivnih bajtov posameznega vmesnega stanja imenujemo *aktivnostni vzorec*. Vsak stolpec matrike vmesnega stanja, ki vsebuje vsaj en aktiven bajt, pa imenujemo *aktiven stolpec*. *Teža aktivnostnega vzorca* naj bo število vseh aktivnih bajtov, ki ga sestavljajo. *Stolpčna teža* pa naj pomeni število vseh aktivnih stolpcev aktivnostnega vzorca. Končno, imenujmo *teža sledi* vsoto tež aktivnostnih vzorcev na začetku vseh krogov dane sledi. Pokazali bomo, da ima vsaka sled dolžine 4 vsaj 25 aktivnih škatel. Najprej pokažimo naslednji dve lemi.

Lema 4.1 *Teža vsake sledi dolžine* 2, *ki ima vsaj* Q *aktivnih stolpcev na začetku drugega kroga, je vsaj* $5Q$.

Dokaz: Označimo z a_0 število aktivnih bajtov pred prvim krogom, z a_1 število aktivnih bajtov pred drugim krogom in z b_0 pa število aktivnih bajtov po operaciji ShiftRow prvega kroga, torej pred operacijo MixColumn. Pokazati je potrebno, da je $a_0 + a_1 \geq 5Q$. Ker je očitno $a_0 = b_0$, moramo torej pokazati $b_0 + a_1 \geq 5Q$. Ker ima sled vsaj Q aktivnih stolpcev na začetku drugega kroga, ima tudi vsaj Q aktivnih stolpcev pred operacijo MixColumn prvega kroga, saj je operacija MixColumn linearna. Operacija MixColumn ima razvejitveno število enako 5, kar po definiciji pomeni, da za poljuben neničeln stolpec (polinom) velja, da je vsota Hammingove teže originalnega stolpca in transformiranega stolpca vsaj 5. Za vsakega od Q aktivnih stolpcev zato velja, da je vsota aktivnih bajtov pred transformacijo MixColumn in po njej vsaj 5. Zato velja $b_0 + a_1 \geq 5Q$, kar smo želeli pokazati. ■

Lema 4.2 *Za vsako sled dolžine* 2 *velja, da je vsota števila aktivnih stolpcev pred prvim krogom in števila aktivnih stolpcev po koncu drugega kroga večja ali enaka* 5.

Dokaz: Naj A_i označuje aktivnostni vzorec pred $(i+1)$-vim krogom (torej ob koncu i-tega kroga), $i = 0, 1, 2$, B_i pa naj označuje aktivnostni vzorec po operaciji ShiftRow $(i+1)$-vega kroga (torej pred operacijo MixColumn $(i+1)$-vega kroga), $i = 0, 1$. Operacija ShiftRow zamakne različne vrstice za različno število mest. Zato je stolpčna teža vzorca A_0 večja od teže vsakega stolpca vzorca B_0. Iz enakega razloga je stolpčna teža vzorca B_1 večja od teže vsakega stolpca vzorca A_1. Ker je stolpčna teža vzorca B_1 enaka stolpčni teži vzorca A_2, iz tega sledi, da je vsota števila aktivnih stolpcev pred prvim krogom in števila aktivnih stolpcev po koncu drugega kroga večja od vsote teže poljubnega stolpca g pred transformacijo MixColumn prvega kroga in po transformaciji MixColumn prvega kroga. Označimo to zadnjo vsoto, ki je odvisna od stolpca g, z G. Aktivnostni vzorec vedno vsebuje vsaj en bajt, zato vedno obstaja nek aktiven stolpec. Ker ima operacija MixColumn razvejitveno število enako 5, velja za (vsak) aktiven stolpec g neenakost $G \geq 5$, s čimer je lema dokazana. ■

Izrek 4.3 *Vsaka sled dolžine* 4 *ima vsaj* 25 *aktivnih škatel.*

Dokaz: Če lemo (4.1) uporabimo posebej za prva dva kroga in posebej za druga dva kroga, dobimo, da je teža sledi večja ali enaka vsoti $5(Q_1 + Q_3)$, kjer sta Q_1 in Q_3 števili aktivnih stolpcev na začetku prvega in tretjega kroga. Če zdaj uporabimo lemo (4.2) za drugi in tretji krog, dobimo, da je $Q_1 + Q_3 \geq 5$, kar smo želeli pokazati. ■

4.3 Ocena skupne varnosti, upoštevajoč vse znane napade

Poleg diferencialne in linearne kriptoanalize je AES odporen tudi na vse druge znane napade. To vključuje napad s kvadratom (Square Attack), napad z odrezanimi diferenciali (Truncated Differentials), interpolacijske napade (Interpolation Attacks), iskanje šibkih ključev (Weak keys) in napade, ki izkoriščajo podobnost ključev (Related-key attacks). V primeru, ko je število krogov večje od 6, za nobenega od teh napadov ni znana izvedba, ki bi bila hitrejša od preiskovanja vseh možnih ključev.

S preprostim kombinatoričnim razmislekom je možno premisliti, da je število vseh različnih bločnih kriptosistemov z dolžino bloka v in dolžino ključa u enako $((2^v)!)^{2^u}$. Za praktične vrednosti parametrov u in v (vsaj 40) predstavljajo kriptosistemi, ki imajo kriptoanalitsko pomembne slabosti, zanemarljiv delež.

Definirajmo dva koncepta varnosti bločnih kriptosistemov. Pripomnimo, da ne gre za matematično ekzaktne koncepte. Za bločni kriptosistem pravimo, da je *K-varen*, če imajo vse strategije napada nanj enake ali večje časovne in prostorske zahtevnosti kot najučinkovitejši napad na veliko večino bločnih kriptosistemov enakih dimenzij. Podoben koncept varnosti je hermetičnost. Za kriptosistem pravimo, da je *hermetičen*, če nima slabosti, ki je ne bi mogli najti tudi pri drugih kriptosistemih enakih dimenzij.

Avtorja kriptosistema Rijndael sta si zadala cilj, da bi Rijndael bil K-varen hermetičen kriptosistem. V tem primeru bi bil Rijndael tako varen, kot je teoretično sploh mogoče. Do danes vsa empirična dejstva in teoretični rezultati kažejo na to, da je Rijndael res K-varen in hermetičen.

5 Zaključek

AES vzbuja velika pričakovanja v računalniški industriji. Algoritem Rijndael naj bi po zatrjevanju ameriške vlade ostal v uporabi vsaj še 20 let. Pričakujemo lahko, da se bo na trgu kmalu po uradni objavi FIPS standarda o AES pojavilo veliko število izdelkov, ki bodo imeli vgrajen kriptosistem AES. AES naj bi postal splošno razširjen standard, ki naj bi bistveno pripomogel k še večjemu razmahu digitalnih komunikacij.

Od AES-a veliko pričakuje tudi kriptografska javnost. Ta pozdravlja odpravljanje carinskih omejitev na kriptografske izdelke in podpira internacionalizacijo kriptografije. Razpis za izbiro standarda AES predstavlja velik korak v tej smeri. AES je ambiciozen projekt, ki ne bi mogel obstajati brez neizmerne količine kriptografskega znanja raziskovalcev po vsem svetu. Če se zavemo, da je kriptografija še pred tridesetimi leti bila v domeni vojske in obveščevalnih služb, v znanstvenih krogih pa se je z njo ukvarjala le peščica raziskovalcev, je to res velik dosežek.

Literatura

[NIST] *Domača stran standarda AES*, dosegljiva na straneh NIST na naslovu:
`http://csrc.nist.gov/encryption/aes`

[Algorithm] *Informacije o algoritmu, ki implementira AES (Rijndael)*, dosegljive na straneh NIST na naslovu:
`http://csrc.nist.gov/encryption/aes/rijndael`

[Rijndael] *J. Daemen, V. Rijmen: AES Proposal: Rijndael*, popolna specifikacija algoritma Rijndael, ki sta jo napisala njegova avtorja, dosegljiva na naslovu:
`http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf`

[AES] *J. Nechvatal et al.: Report on the Development of the Advanced Encryption Standard (AES)*, uradno poročilo NIST-a o izboru standarda AES, dosegljivo na naslovu:
`http://csrc.nist.gov/encryption/aes/round2/r2report.pdf`

[FIPS] *Vzorčni standard FIPS o AES (draft standard)*, dosegljivo na naslovu:
`http://csrc.nist.gov/publications/drafts/dfips-AES.pdf`

[Daemen] *J. Daemen: Cipher and hash function design strategies based on linear and differential cryptanalysis*, poglavje 5, doktorska disertacija, Katholieke Universiteit Leuven, Belgija, dosegljivo na naslovu:
`http://csrc.nist.gov/encryption/aes/rijndael/PropCorr.pdf`

[RijndaelHome] *Domača stran algoritma Rijndael*, ki jo urejata avtorja algoritma Rijndael, dosegljivo na naslovu:
`http://www.esat.kuleuven.ac.be/~rijmen/rijndael`

[Implementations] *Implementacije algoritma Rijndael za različne operacijske sisteme*, podstran domače strani algoritma Rijndael, dosegljiva na naslovu:
`http://64.129.7.53/www.rijndael.com/implementations.html`

[Landau] *S. Landau: Communications Security for the Twenty-first Century: The Advanced Encryption Standard*, članek iz revije Notices of the AMS, Volume 47, Number 4, April 2000

[Dobbs] *AES and the Twofish Encryption Algorithm*, članek iz revije Dr. Dobbs's Journal, Volume 292, December 1998

[Blaze] *M. Blaze et al.: Minimal key lengths for symmetric ciphers to provide adequate commercial security: A report by an ad hoc group of cryptographers and computer scientists*, dosegljiva na naslovu:
`http://www.crypto.com/papers/keylength.txt`

Opombe o literaturi

AES je mlad kriptografski standard, zato je večina literature o njem na voljo v elektronski obliki na Internetu. Glavni strani o AES sta uradna stran [NIST] inštitucije NIST,

ki je vodila izbor algoritma za AES, in domača stran [RijndaelHome] avtorjev algoritma Rijndael. Ti dve strani skupaj vsebujeta vse potrebne informacije za implementacijo Rijndaela in bosta zagotovo zadovoljila vse, ki želijo AES uporabljati. Poleg tega je na teh dveh straneh možno najti tudi veliko gradiva za raziskovalce, ki želijo izvedeti več o teoretičnem ozadju algoritma Rijndael in se ukvarjati z njegovo varnostjo.

Člankov o AES v strokovnih matematičnih in računalniških revijah je še razmeroma malo. V nekaj računalniških revijah so bile v obliki krajših poljudnih člankov povzete glavne informacije o AES (npr. v Dr. Dobb's Journal). Bodisi v matematični knjižnici bodisi elektronsko na Internetu sem pregledal vse kriptografske revije (razen revije Cryptologia) s spiska na domači strani tečaja o kriptografiji in računalniški varnosti (`http://valjhun.fmf.uni-lj.si/~ajurisic/tecaj1/lit.html`) in v nobeni nisem našel strokovnega članka o AES ali o njegovi varnosti. Za kaj takega je najbrž še prezgodaj, saj uradni standard o AES pravzaprav sploh še ni bil uradno obljavljen.