UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jernej Južna

TOKOVNA ŠIFRA RC4

Seminarska naloga pri predmetu
Kriptografija in računalniška varnost

Predavatelj: prof. dr. Aleksandar Jurišić

Ljubljana, 2009

Kazalo

Uvod

Šifra RC4 je najbolj razširjena tokovna šifra v programski opremi. Razloge za njeno priljubljenost gre iskati predvsem v njeni preprostosti, hitrosti in majhni strojni zahtevnosti. Uporablja jo praktično vsak današnji osebni računalnik, saj se uporablja v pomembnejših komunikacijskih protokolih WEP, WPA, TLS/SSL, PPTP... in v produktih MS Windows, MS Office, Lotus Notes, Oracle SQL ter mnogih drugih.

Šifro je leta 1987 izdelal Ron Rivest za podjetje RSA. Do leta 1994 je bila poslovna skrivnost, nato pa je bil algoritem anonimno objavljen na internetu. Po objavi se je precej strokovnjakov spustilo v kriptoanalizo in kmalu je bilo odkritih kar nekaj šibkih točk. Najdene pomanjkljivosti in predvsem napačna uporaba šifre so povzročili, da se je šifra v zadnjih letih znašla na slabem glasu. Najbolj odmeven je predvsem polom protokola WEP, ki je bil, zaradi napačne uporabe šifre RC4, uspešno razbit leta 2001.

Dodatno je med leti 2004 in 2008 potekal projekt eSTREAM, katerega cilj je bil najti ustrezne nove tokovne šifre, ki bi ob poznanih napadih nudile ustrezno varnost. Kljub temu, da projekt še ni dal dokončnega standarda, je kar nekaj kandidatov, ki izpolnjujejo vse zastavljene cilje. Zdi se, da se življenjska doba ene prvih komercialnih tokovnih šifer izteka, in da je njena prevlada končana.

Klub vsem najdenim pomanjkljivostim, trenutno še ne obstaja praktični napad, ki bi šifro razbil ob njeni pravilni uporabi. Tako šifra tudi po več kot 20 letih obstoja še vedno velja za varno in je tako primerna za večino področij, kjer potrebujemo hitro in kriptografsko močno šifriranje z uporabo cenenih preprostih mikroprocesorjev.

V prvem poglavju bomo najprej predstavili osnovne lastnosti tokovnih šifer in osnovne napade nanje. Naslednje poglavje bo podrobno predstavilo šifro RC4 in njene najpomembnejše lastnosti. V 3. poglavju se bomo posvetili kriptoanalizi omenjene šifre in specifičnim napadom nanjo. Sledil mu bo pregled uporabe šifre v praksi, na koncu pa bomo povzeli glavne lastnosti in pomanjkljivosti ter predvideli prihodnost tokovne šifre RC4.

1. Tokovne šifre

Tokovne šifre spadajo med simetrične kriptosisteme, torej sisteme, ki za šifriranje in odšifriranje uporabljajo enak ključ. S tem tokovne šifre prevzamejo splošne lastnosti te široke družine, tako dobre (hitrost in kratka dolžina ključa v primerjavi s kriptografijo z javnim ključem) kot slabe (kompleksno upravljanje s ključi).

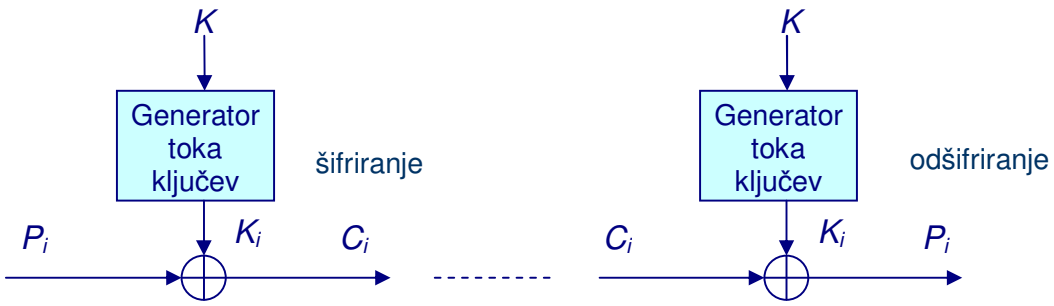
Osnovna ideja tokovnih šifer je generirati tok ključev iz enega začetnega ključa. Ta tok se uporablja za šifriranje posameznih blokov čistopisa – običajna dolžina bloka je 1 bit za tokovne šifre v strojni opremi, in 8 ali 32 bitov za šifre v programski opremi.

$$K_i = f(K, i)$$
$$Y = y_1 y_2 y_3 \ldots = e_{K_1}(x_1) e_{K_2}(x_2) e_{K_3}(x_3) \ldots$$

Y ... tajnopis
y_i ...blok tajnopisa
x_i ...blok čistopisa
K_i ...posamezni člen v toku ključev
e_{K_i} ..šifrirna funkcija s ključem K_i

Običajno se pri tem za samo šifrirno funkcijo uporablja preprosto seštevanje z *XOR* (ekskluzivna disjunkcija – $\oplus$) 1-bitnega (8/32-bitnega) člena čistopisa s trenutnim členom toka ključev.

Tipično delovanje tokovne šifre je prikazano na sliki 1.



Slika 1: Ključ (K) je vhod v generator toka ključev. Blok tajnopisa (C_i) je rezultat ekskluzivne disjunkcije bloka čistopisa (P_i) s trenutnim ključem (K_i): $C_i = P_i \oplus K_i$. Odšifriranje poteka na enak način, saj velja: $P_i = C_i \oplus K_i$.

Privlačnost tokovnih šifer je v tem, da spominjajo na Vernamov enkratni ščit. V resnici le spominjajo, saj posamezni ključi med seboj niso neodvisni in jih lahko vedno rekonstruiramo z napadom z grobo silo (angl. brute-force attack), kar pri Vernamovem enkratnem ščitu ni mogoče. Dodatno se je potrebno zavedati, da je ob istem začetnem ključu tok ključev identičen in je torej potrebno za varno šifriranje spremeniti ključ za vsak podatkovni tok.

1.1. Bločne šifre v primerjavi s tokovnimi

Bločne šifre so veliko bolj znan predstavnik simetričnih kriptosistemov. Čistopis razdelijo na enako velike bloke in nato enak ključ uporabljajo za šifriranje vseh blokov čistopisa, ki so običajno daljši (64 – 512 bitov).

$$Y = y_1 y_2 y_3 \ldots = e_K(x_1) e_K(x_2) e_K(x_3) \ldots$$

Bločne šifre lahko delujejo tudi v načinu OFB (output feedback) ali CTR (counter), kar povzroči, da se ključi za šifriranje posameznega bloka spreminjajo in v bistvu dobimo tokovno šifro. Pojavi se vprašanje, zakaj ne bi bločnih šifer tudi v praksi uporabljali za šifriranje podatkovnih tokov, saj je varnost bločnih šifer dosti bolj raziskana, razumljiva in standardizirana (npr.: DES v več kot 30 letih obstoja ni bil nikoli razbit[1]). Odgovor se skriva v večji hitrosti in manjši strojni kompleksnosti (angl. hardware complexity) tokovnih šifer, kar omogoča hitrejše prenose in njihovo uporabo v manjših/energetsko manj potratnih napravah. Vendar pa se z večjo računsko močjo meja briše in že sedaj v nekaterih primerih bločne šifre izpodrivajo »prave« tokovne šifre (npr: AES v protokolu WPA2).

1.2. Zgradba

Zgradbo tokovne šifre lahko razdelimo na dva dela – na inicializacijo in generiranje toka ključev.

Prvi del, poznan tudi kot predpriprava ključa (KSA - angl. key-scheduling algorithm), skrbi za vzpostavitev začetnega notranjega stanja na podlagi vhodnega ključa. Zahtevane lastnosti za dober KSA so enosmernost (torej, da iz notranjega stanja ni mogoče rekonstruirati ključa) in razpršenost (majhna sprememba ključa zelo spremeni začetno notranje stanje). Ta del se zažene le enkrat, pred začetkom samega kodiranja, in če kodiramo daljše čistopise nima pomembnega vpliva na hitrost.

Drugi del se imenuje tudi generator psevdonaključnih števil (PRGA – angl. pseudo-random generation algorithm) in iz notranjega stanja generira tok ključev oz. kar tok psevdonaključnih števil. V vsakem koraku PRGA vrne izhodno vrednost (šifrirni ključ za ta korak) in opravi prehod v novo notranje stanje. Lastnosti, ki jim mora dober PRGA ustrezati, obsegajo, veliko periodo, odpornost na razne statistične teste (npr: NIST), majhna korelacija z notranjim stanjem/ključem, visoka hitrost, majhne strojne zahteve, ...

[1] V smislu boljše kot napad z grobo silo, zaradi prekratkega ključa (56 bitov) DES pri današnji računski moči ne velja več za varnega.

1.3. Osnovni napadi

Poleg napada z grobo silo, ki preizkusi vse možne ključe, in napada s slovarjem, ki preizkusi le »najbolj verjetne« ključe, so se uveljavili še naslednji napadi na tokovne šifre.

1. Ponovljen tok ključev (angl. two-time pad)
V primeru uporabe istega vhodnega ključa pridobimo za šifriranje enak izhodni tok. To pomeni, da v kolikor napadalec enkrat pridobi izhodni tok (npr. z uporabo napada s poznanim čistopisom), lahko odšifrira vso komunikacijo, ki je šifrirana z istim ključem. Osnovna obramba je uporaba javno znanega inicializacijskega vektorja (IV), ki ga kombiniramo z začetnim ključem v nov vhodni ključ. S tem lahko začetni/skriti ključ uporabimo večkrat, le IV je potrebno za vsako novo šifriranje spremeniti. Pri uporabi je potrebno paziti, da če želimo ohraniti enak nivo varnosti, mora imeti IV toliko bitov kot ključ.

2. Izguba sinhronizacije (angl. synchronization attack)
Tokovna šifra lahko deluje le ob popolni podatkovni sinhronizaciji med šifrirnikom in odšifrirnikom (notranje stanje mora biti enako). V kolikor se sinhronizacija izgubi je odšifriranje od te točke dalje napačno in je potrebno postopek ponoviti. Za obrambo se uporabljata dva načina – nekatere tokovne šifre omogočajo »samo-sinhronizacijo«, kar pomeni, da se sinhronizacija lahko samodejno obnovi, drugi način, ki je uporaben za vse tokovne šifre, pa je razdelitev toka na več okvirjev (angl. frame), ki imajo predpisano (nešifrirano) glavo, ki skrbi za sinhronizacijo, hkrati pa lahko prenaša IV.

3. Spreminjanje podatkov
Če na enem mestu v tajnopisu spremenimo podatke, se to pri odšifriranem čistopisu pri tokovnih šifrah pozna samo na tistem mestu kjer je sprememba nastala (npr: pri bločnih šifrah se odšifriranje izvede, vendar že s spremembo enega bita običajno dobimo neprepoznavno zmedo). Za ugotavljanje takih napak se zato uporablja šifra za ugotavljanje avtentičnosti sporočila (MAC – angl. message authentication code), ki se doda sporočilu (običajno na koncu okvirja).

4. Razločevalni napad (angl. distinguishing attack)
Gre za »napad«, ki loči izhodni tok od popolnega naključnega niza števil. Sam po sebi napad ni škodljiv, vendar, ker običajno tokovna šifra prestane standardne statistične teste, nakazuje na korelacijo med posameznimi členi izhodnega toka, kar lahko vodi v nadaljnje napade.

Očitno je, da vsi ti napadi veljajo za vse tokovne šifre in zato je potrebno pri izbiri katerekoli zelo paziti na pravilno končno implementacijo, da je odporna vsaj na prve tri napade. V praksi se, kot že omenjeno, največkrat uporablja razdelitev čistopisa na posamezne okvirje, ki so sicer kodirani z istim ključem a drugim IV in imajo na koncu pripet povzetek MAC.

2. Tokovna šifra RC4

2.1. Zgodovina

Kot omenjeno je šifro leta 1987 izdelal znani kripto-analitik Ron Rivest (R v kratici RSA - algoritma za kriptografijo z javnim ključem) za podjetje RSA. Kratica RC v imenu uradno pomeni »Rivest Cipher«, neuradno pa se dostikrat omenja »Ron's Code«. Številka 4 da slutiti, da obstajajo tudi druge verzije in res obstajajo RC2 (1987, bločna šifra), RC3 (nikoli ni prišla v uporabo, ker je bila razbita tekom razvoja) ter kasnejše RC5 (1994, bločna šifra) in RC6 (1998, bločna šifra, kandidat za AES).

Šifra je bila do leta 1994 poslovna skrivnost, nato pa je bila anonimno objavljena na internetu (kot sporočilo na e-poštnem seznamu Cypherpunks). Z inverznim inženiringom in primerjavo izhodnih vrednosti šifre je bila kmalu zatem dokazana pravilnost objavljene kode in lov za ranljivostmi se je lahko začel.

2.2. Algoritem

Razširjenost RC4 gre pripisati predvsem njegovi preprosti implementaciji in razumljivosti same kode. Vsakega od obeh delov algoritma (KSA in PRGA) lahko v psevdokodi zapišemo v manj kot 10 vrsticah (tabela 1).

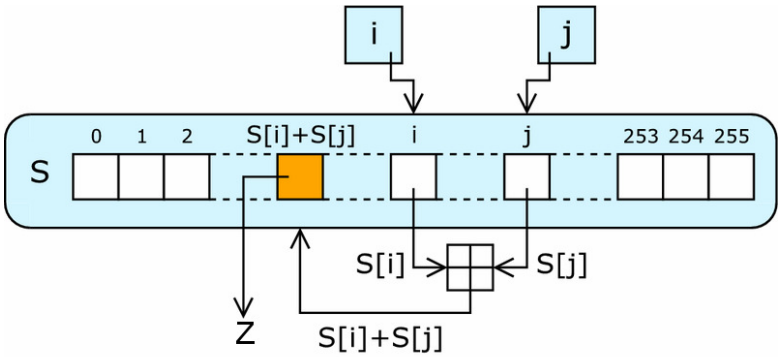
Tabela 1: Psevdokoda KSA in PRGA algoritma tokovne šifre RC4.

KSA	PRGA

`for i from 0 to 255`
`  S[i]=i`
`endfor`
`j=0`
`for i from 0 to 255`
`  j=(j+S[i]+K[i%keylenght])%256`
`  swap(S[i],S[j])`
`endfor`
	`i=0`
	`j=0`
	`while DoOutput`
	`  i=(i+1)%256`
	`  j=(j+S[i])%256`
	`  swap(S[i],S[j])`
	`  output S[(S[i]+S[j])%256]`
	`endwhile`

Vidimo, da KSA notranje stanje vzpostavi kot permutacijo števil od 0 do 255 (polje *S*), pri čimer iz začetne identične permutacije zgradi »naključno« s pomočjo vrednosti ključa *K*. PRGA nato iz tega stanja glede na vrednost kazalcev *i*, *j* izbere izhodno vrednost, ter s pomočjo funkcije *swap*, ustvari novo permutacijo. Hkrati tudi popravi vrednosti kazalcev *i*, *j*. Ker se vrednost kazalca *i* spreminja zaporedno to zagotavlja, da pri generiranju vsaka celica polja *S* sodeluje vsaj enkrat v 256 korakih.

Grafično lahko izhodno vrednost algoritma generiranja števil predstavimo kot prikazuje slika 2. Trenutna kazalca *i* in *j* kažeta na ustrezni mesti v polju *S*. Ti vrednosti se seštejeta skupaj po modulu 256 in vrednost polja, kamor kaže pridobljena vsota ($S[S[i]+S[j]]$), je izhodna vrednost *Z*.



Slika 2: Grafična predstavitev izhodne vrednosti posameznega koraka PRGA. Vir: *wikipedia common*.

2.3. Osnovne lastnosti

Že iz kode algoritma se vidi nekaj lastnosti, ki jih lahko takoj izpostavimo.

1. Preprostost
Vsekakor je to najopaznejša lastnost šifre. Nekako nismo navajeni, da lahko tako preprosta koda pomeni kriptografsko varen algoritem, še sploh če smo kdaj pogledali v kakšne implementacije bločnih šifer ali zgoščevalnih funkcij, kjer je vse polno čudnih konstant.

2. Spremenljiva dolžina ključa
Iz kode je razvidno, da šifra omogoča spremenljivo dolžina ključa – vse do 2048 bitov. Šifra v svoji zasnovi ne podpira IV, vendar je ravno zaradi spremenljive dolžine možno enostavno kombinirati IV in začetni ključ v nov vhodni ključ. V praksi se največkrat uporablja 64 ali 128-bitni skriti ključ in 24 do 48-bitni IV, ki se enostavno doda na začetku ali koncu ključa:
$K_{rc4} = K_{skriti} \parallel IV$.

3. Veliko notranje stanje
Notranje stanje algoritma obsega 2064 bitov (2048 bitov za *S* in še 16 bitov za *i* in *j*). Ker je polje *S* permutacija vrednosti med 0 in 255, je entropija celotnega stanja manjša – ocenimo jo lahko kot $\log_2(256! \cdot 256^2) \cong 1700$ bitov.

4. Hitrost
Algoritem uporablja 8-bitne operande in zelo standardne/preproste operacije, zato je zelo primeren tudi za najšibkejše 8-bitne mikroprocesorje. Na današnjih računalnikih najpreprostejša implementacija brez optimizacije doseže 200 MB/s (testirano na 2GHz enojedrnem AMD procesorju).

5. Perioda
Perioda šifre ni teoretično določena, je pa odvisna od ključa in vsekakor zelo velika – ocenjena spodnja meja (za skoraj vse ključe) je 10^{100} [13]. Ta številka je veliko manjša od teoretično največje možne $256! \approx 10^{500}$, vendar dovolj velika, da ne predstavlja realne grožnje.

6. Posplošitev

Šifra operira s standardnimi 8-bitnimi operacijami, vendar jo lahko posplošimo tudi na poljubno število bitov. V tem primeru ima *n*-bitna RC4 notranje stanje veliko 2^n, kazalca *i*,*j* sta *n*-bitna, največja dolžina ključa pa $n \cdot 2^n$. Posplošitev je pomembna za analizo nekaterih napadov na šifro, ki v dostopnih mejah delujejo le pri malih vrednostih ($n \leq 5$). V praksi se vedno uporablja *n =8*, ker manjše vrednosti ne zagotavljajo dovolj varnosti, večje vrednosti pa niso prilagojene računalniški arhitekturi ali pa zahtevajo preveliko notranje stanje (za *n=16* 64 KB in *n=32* 4 GB).

3. Kriptoanaliza

V poglavju so predstavljene šibke točke RC4, ki izhajajo iz njene zgradbe, in napadi, ki temeljijo na teh ranljivostih.

3.1. Kratki cikli

Kmalu po objavi algoritma je potekala živahna debata na temo RC4 znotraj sci.cypt novičarske skupine in Hal Finney je že leta 1994 [2] dokazal obstoj kratkih ciklov. Če velja $j=i+1$ in $S[j]=1$, potem bo v naslednjem koraku $i'=i+1=j$, $j'=j+S[i']=j+S[j]=j+1$, $S[j']=S[i']=1$ in zopet imamo ponovljeno situacijo, ker zopet velja $j'=i'+1$ in $S[j']=1$. Tako stanje je ciklično in se ponovi po 65280 korakih, kar pomeni, da ima izhodni tok tako periodo. Na srečo do takih ciklov ne more priti, saj mora biti sistem v takem stanju že na začetku, kar pa z uporabljenim PRGA algoritmom ni mogoče (vedno na začetku velja $i = j = 0$).

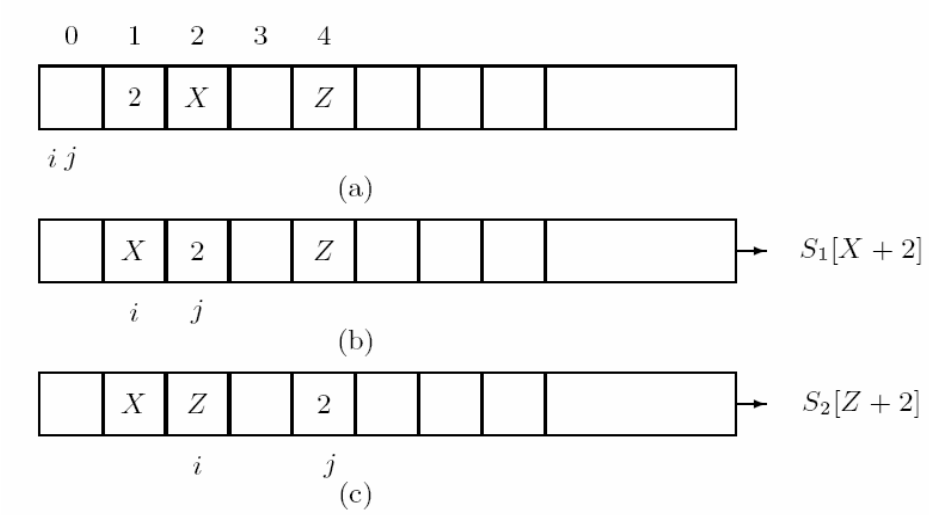
3.2. Slabi ključi

Ross je leta 1995 [14] z analizo KSA prikazal, veliko korelacijo med vhodnim ključem in začetnimi vrednostmi notranjega stanja po zaključku KSA. Na podlagi te korelacije je dokazal, da za ključe, ki ustrezajo $K[0]+K[1]=0$, s 13.8% verjetnostjo velja $Z_0=K[2]+3$ ($K[i]$... i-ti bajt ključa K, Z_i...i-ti izhod šifre). Temu pogoju ustreza 1 izmed 256 ključev in tak ključ povzroči oslabitev ključa za ~5 bitov (za določitev prvih 24 bitov takega ključa je namesto pričakovanih 2^{16} (8 bitov določa pogoj) preiskovanj dovolj, če jih opravimo ~2^{11}). V kolikor se za različne seje uporablja ključ kot števec po pravilu tankega konca (angl. little endian) se oslabitev poveča na ~18 bitov, ker se slabi ključi ponavljajo ciklično (perioda 2^{16}) po predvidljivem vzorcu (slabemu ključu *0x0000...* sledi 510 »normalnih« ključev, nato slab ključ *0xFF01...*, potem 254 normalnih, slab *0xFE02...*, 254 normalnih, ...).
RC4 omogoča spremenljivo dolžino ključev, vendar se pri KSA rabi vseh 2048 bitov ključa in v primeru krajšega ključa, ga tolikokrat ponovi, da zapolni vse bite. Grosul in Wallach sta leta 2000 [6] analizirala uporabo 2048-bitnih ključev in prišla, do ugotovitve, da taki ključi omogočajo napad s podobnimi ključi (angl. related key attack). Iz naključnega ključa sta zgradila družino 65280 zelo podobnih ključev, za katere velja $K'[i]=K[i]$, $K'[t]=K[t]+\delta$, $K'[t+1]=K[t+1]-\delta$ (ključi se razlikujejo le v dveh zaporednih bajtih). Ideja je v tem, da do koraka t velja, da je stanje KSA enako, korak t povzroči spremembo, vendar se le-ta lahko uspe v veliki meri odpraviti po koraku $t+1$. Pri analizi rezultatov sta ugotovila, da kar 18.6% podobnih ključev generira zelo podoben začetni izhodni tok dolžine vsaj 100 in 5% zelo podoben tok prvih 256 izhodov. Zaradi te podobnosti je efektivna dolžina ključa za $log_2(65280 \cdot 0.186) \approx 14$ bitov krajša. Napad pri standardnih 128-bitnih ključih ni mogoč, saj se sam ključ pri KSA 16-krat ponovi in tako onemogoča podobnost tokov ob majhni spremembi ključa.
Fluhrer, Mantin in Shamir so leta 2001 [3] objavili nov velik razred slabih ključev, za katere velja, da majhen del slabega ključa določa velik del notranjega stanja po končanem KSA in posledično prvih nekaj izhodov. Omenjeni slabi ključi so tisti, za katere velja, da je $K[0]=1$ in $K[1]>128$ ter da so b-ohranitveni (angl. b-conserve). Ključ dolžine L je b-ohranitveni, če za vsak t

med 0 in 255 velja: $K[t \bmod L] \equiv (t\text{-}1) \bmod b$. Uporabo takega 48-bitnega ključa v šifri povzroči, da 14 bitov ključa natančno določa 238 bitov začetnega stanja, 19 bitov ključa pa 472 bitov.

3.3. Razločevalni napadi

Golič je leta 1997 [5] prikazal, da za izhoda ob času t in izhod ob času t+2 prevečkrat velja $LSB(Z_t \oplus Z_{t+2})=1$ (*LSB*...najmanj pomemben bit (angl. least significant bit)). Koeficient korelacije je $\sim 2^{-20}$ oz. verjetnost pojavitve takega izhodnega toka je $0.5+2^{-21}$. Pri resnično naključnem nizu je koeficient korelacije 0 oz. verjetnost pojavitve natančno 0.5. Z uporabo teorije informacij (opisano v [4] in [5]) lahko ocenimo, da je za zanesljivo detekcijo, ali je izhodni tok rezultat šifre RC4, potrebno analizirati $\sim 2^{44}$ izhodov.
Izboljšan razločevalni napad sta leta 2000 predstavila Fluhrer in McGrew [4], ki sta analizirala pojavitev parov dveh zaporednih izhodov. Ugotovila sta, da se 8 parov pojavlja bolj pogosto kot pričakovano, 3 pa manj pogosto. Ta razlika omogoča razlikovanje po $\sim 2^{31}$ izhodih.
Pravi preboj sta leta 2001 naredila Mantin in Shamir [9], ko sta primerjala več različnih izhodih tokov med seboj. Opazila sta, da se na drugem izhodu 0 pojavi dvakrat bolj pogosto kot pričakovano (z verjetnostjo ~0.008, namesto ~0.004). To je posledica tega, da se 0 na drugem izhodu pojavi vedno, ko pri začetnem stanju velja $S_0[2]=0$ in $S_0[1] \neq 2$. Tako močna statistična anomalija seveda omogoča razločevanje že po opazovanju ~256 prvih dveh izhodov različnih izhodnih tokov. Ker je najpogostejša izhodna vrednost konstanta (in to prav 0) to seveda omogoča razkritje drugega bajta šifriranega sporočila, če je isto sporočilo kodirano z večimi različnimi ključi.
Najbolj čudno je, da omenjene anomalije ni noben odkril že davno prej, saj je znano, da poznamo t.i. šibke razločevalce (angl. weak distinguisher), ki analizirajo dolgo zaporedje enega ključa, in t.i. močne razločevalce (angl. strong distinguisher), ki medsebojno analizirajo zaporedja večih različnih ključev. Teoretično sta enaka, saj ugotovljena pristranskost enega velja tudi za drugega, vendar je v praksi lahko med njima velika razlika, kar je čudovito vidno ravno na primeru šifre RC4.
Še en razločevalni napad sta odkrila Paul in Preneel leta 2004 [12]. Dokazala sta, da je verjetnost, da sta prva dva izhoda enaka, manjša od pričakovane, ker velja, da če je pri začetnem stanju $S_0[1]=2$, potem izhoda ne moreta biti enaka (slika 3). To omogoča razlikovanje po $\sim 2^{25}$ opazovanih prvih dveh izhodov različnih tokov. Zmanjšana verjetnost pojavitve enakih vrednosti prvih dveh izhodov velja tudi, če zanemarimo prvih 256 izhodov PRGA, vendar je korelacija ustrezno manjša in moramo opazovati $\sim 2^{32}$ različnih tokov za natančno razločevanje.



Slika 3: Grafični prikaz delovanja šifre, če velja $S_0[1]=2$. (a) Začetno stanje. (b) Novo stanje in izhod $S_1[X+2]$. (c) Novo stanje in izhod $S_2[Z+2]$. Velja $S_1[X+2] \neq S_2[Z+2]$. Vir: *Povzeto po* [12].

3.4. Napad obnovitve notranjega stanja

Prvi poskus napada na RC4 je rekonstrukcija notranjega stanja šifre na podlagi izhodnega toka. Pri tem napadu sicer ne pridobimo skritega ključa, vendar je informacija o trenutnem stanju dovolj, da obnovimo celotni izhodni tok šifre.

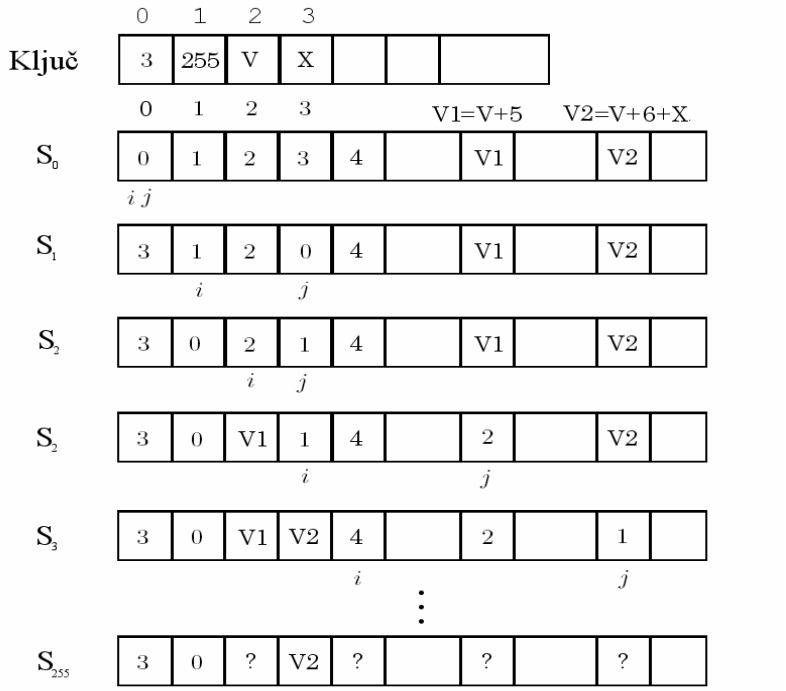
Knudsen in sodelavci so leta 1998 [8] razvili napad, ki temelji na metodi razveji in omeji (angl. branch and bound). Napad temelji na uganjevanju notranjega stanja šifre na podlagi izhodnega toka s posnemanjem delovanja šifre. Glavna ideja je omejevanje ugibanja na podlagi informacije, da je notranje stanje permutacija in tako se velik del ugibanj lahko hitro izkaže za napačne. Tak napad je kljub omejitvam izjemno neučinkovit, saj potrebuje ~2^{779} korakov za obnovitev stanja in je tako zgolj teoretičen. Maximov in Khovratovich sta leta 2008 [10] napad precej izboljšala – časovna zahtevnost sta zmanjšala na ~2^{241} korakov.

Knudsenov napad temelji na rekurzivnem reševanjem štirih spremenljivk (j_t, $S_t[i_t]$, $S_t[j_t]$, $S_t^{-1}[Z_t]$ (indeks polja, ki ima vrednost Z)) ob poznanih dveh (i_t in Z_t) s pomočjo ugibanja. Preverjanje ujemanja notranjega stanja ob protislovni rešitvi povzroči sestop rekurzije in drugačno ugibanje. Najnovejši napad še vedno poteka rekurzivno z ugibanjem in sestopanjem, vendar doseže pohitritev, ker se naslanja na iterativno reševanje sistema w-enačb z dvema spremenljivkama ($S_t[j_t]$, $S_t^{-1}[Z_t]$). Gradnja takega sistema je mogoča, ker obstajajo taka notranja stanja, ki omogočajo delno sledenje v naslednjih w korakih. Kdaj se šifra nahaja v takem stanju, pa je možno z veliko verjetnostjo predvideti na podlagi pojavitev določenih (vnaprej izračunanih) vzorcev v izhodnem toku.

Številka 2^{241} je še vedno prevelika za praktični napad, vendar postavlja zgornjo mejo varnosti, ki jo nudi RC4 – torej uporaba šifre s ključem večjim do 240 bitov nima smisla.

3.5. FMS napad

Prvi napad, ki določi ključ RC4 na podlagi izhodnega toka, so objavili Fluhrer, Mantin in Shamir leta 2001 [3]. Napad je možen le, če lahko napadalec opazuje prvi izhodni bajt večih tokov, ki jih generira ključ, sestavljen iz istega skritega ključa in dodanega IV.
Napad sloni na t.i. določenem stanju (angl. resolved condition). To je tisto stanje, v katerem po i-tem koraku KSA, velja: $i \geq max(1,X,Y)$, $X=S[1]$, $D=S[X]$ in $Y=X+D$. To stanje je z verjetnostjo večjo od e^{-3}~0.05 ohrani do konca KSA[2] (torej nobeno od polj $S[1]$,$S[X]$ in $S[Y]$ se več ne zamenja). V tem primeru bo prvi izhod PRGA enak $S[Y]$. V kolikor se zamenjava kakšnega od zanimivih polj do konca KSA zgodi, je določeno stanje uničeno, izhod pa naključen.
Najlažje je napad izvesti, če velja, da je IV ključu dodan na začetku ($K_{RC4}=IV \parallel K_{skriti}$). V tem primeru je iz prvega izhoda PRGA z verjetnostjo 5% možno izračunati $K_{RC4}[A]$, če poznamo $K_{RC4}[i];i<A$ in je IV oblike $[A,255,V]$. Grafično postopek utemeljuje slika 4. Zaradi poznanega IV poznamo prve bajte ključa, ostale pa lahko s pomočjo te relacije zaporedno poiščemo. Iskanje vsakega bajta ključa zahteva pregled ~60 ustreznih IV z različno vrednostjo V. Razbitje 128-bitnega ključa torej zahteva analizo ~960 prvih izhodov šifre z ustreznimi IV. Napad je možno izvesti tudi v primeru, ko je IV skritemu ključu dodan na koncu, ali če sta ključ in IV ekskluzivno sešteta, vendar oba primera zahteva dodatno kompleksnost.



Slika 4: Prikaz vzpostavitve notranjega stanja glede pri FSM napadu. Če želimo pridobiti vrednost 3 bajta ključa – X, potem nastavimo začetek ključa na $(3,255,V)$, kjer je V dokaj poljubna konstanta. Glede na KSA, se bo po tretjem koraku vzpostavilo notranje stanje označeno s S_3. Z verjetnostjo $p \approx 0.05$ se bodo mesta 0, 1 in 3 ohranila do konca KSA (stanje S_{255}). Tako stanje povzroči, da bo prvi izhod šifre $V2$, ki omogoča direkten izračun X.

[2] Ker je i večji od 1, X in Y se polje lahko zamenja le, če ga naslovi kazalec j. Če privzamemo, da je j naključen, potem v enem koraku ne naslovi določenega polja z verjetnostjo $(1-n^{-1})$ oz. v največ n preostalih korakih ne naslovi z verjetnostjo $(1-n^{-1})^n \approx e^{-1}$. V vseh enačbah velja n=256.

3.6. Kleinov napad

Leta 2006 je Klein [7] objavil še uspešnejši napad na šifro. Napad temelji na korelaciji med poznano vrednostjo izhoda (Z_i) pri poznanem kazalcu i in vrednostjo kazalca j ter vrednostjo polja $S[j]$. Pokazal je, da velja:

$$P(S[j]+Z_i=i)=\frac{2}{n} \text{ in } P(S[j]+Z_i=c)=\frac{n-2}{n(n-1)} \ (c\neq i, \text{n=256})$$

Dodatno po drugem koraku KSA, razen redkih izjem, velja $S[1]=K[0]+K[1]+1$ in ta vrednost se do konca KSA z veliko verjetnostjo (e^{-1}~0.37) ne spremeni. Skupaj s korelacijo lahko zaključimo, da je verjetnost, da za prvi izhod velja relacija $S[1]=1-Z_0$, 36% večja od verjetnosti naključnega izhoda. Ta razlika omogoča, da določimo iskano vrednost $S[1]$ po ~25000 opazovanjih različnih tokov. To pomeni, da če uganemo $K[0]$, lahko določimo $K[1]$. Pri določitvi drugega bajta ključa uporabimo $S[2]=f(K[0],K[1],K[2])$ in, veljavnost korelacije $S[2]=1-Z_1$. Na podlagi prej ugotovljenih $K[0]$ in $K[1]$ lahko torej določimo $K[2]$ in to na podlagi opazovanja istih tokov, ki smo jih uporabili za določitev $K[0]$ in $K[1]$, le da tokrat opazujemo drugi izhod. Podobno velja za določitev vseh naslednjih bajtov ključa K.
Napad se izkaže za uspešnejšega od FMS napada, čeprav na prvi pogled zahteva opazovanje večih izhodnih tokov. Vendar v nasprotju s FMS napadom, Kleinovem napad deluje neodvisno od IV, ne glede na to na katerem koncu je le-ta dodan (napad vseeno zahteva, da se vseh 25000 sej spreminja le IV del ključa, iskani del ključa pa ne). V praksi se izkaže, da na IV nimamo vpliva in moramo pri FMS napadu čakati na ustrezne IV, medtem ko pri Kleinovem napadu lahko uporabimo vse, kar vodi v hitrejše razbitje šifre.
Modificiran Kleinov napad deluje tudi, če zanemarimo prvih 256 izhodov šifre, vendar je v tem primeru korelacija dosti manj opazna. Napad je tudi odvisen od pravilne oblike IV za posamezen bajt ključa, kar vodi v zaporedno iskanje podobno FMS napadu. Posledično moramo za uspešno razbitje analizirati dosti več izhodnih tokov (za 128-bitni ključ ~$20\cdot 10^6$ tokov).

3.7. Ostale posebnosti

Zanimiva lastnost šifre je zmanjšanje entropije notranjega stanja glede na entropijo ključa. Vseh možnih ključev je 2^{2048}, vseh možnih permutacij pa $256!\approx 2^{1684}$, to pa pomeni, da vsaj 2^{364} ključev po KSA vzpostavi natanko enako notranje stanje. Dodatno velja, da $256!$ ne deli 2^{2048}, kar pomeni, da distribucija začetnega stanja sigurno ni enakomerna. Natančnejšo študijo distribucije je opravil Mironov [11], ki je ugotovil, da je možno z verjetnostjo 56% napovedati predznak permutacije začetnega notranjega stanja, da je najpogostejše začetno notranje stanje identiteta in da je najmanj pogosto začetno notranje stanje oblike 1,2,...,255,0.

Večina naštetih ranljivosti izhaja iz preveč preprostega KSA, ki preslabo »zmeša« bite ključa v začetno notranje stanje. Poznavanje dela začetnega notranjega stanja vodi v korelacijo med prvimi nekaj izhodi PRGA in uporabljenim ključem. Dodatno obstajajo ključi, ki to korelacijo še posebej močno izpostavijo. Pravilna uporaba, ki je odporna na vse omenjene napade, zahteva torej dobro mešanje ključa z IV, priporočeno s kakšno kriptografsko varno zgoščevalno funkcijo (npr: SHA-1), in izpustom vsaj prvih 256 izhodov (priporočeno 768 ali več).

4. Razširjenost

V tem poglavju je predstavljenih nekaj najbolj znanih praktičnih uporab šifre RC4. Predvsem je poudarek na načinu implementacije in z njo povezanih ranljivostmi.

4.1. WEP (Wired Equivalent Privacy)

WEP je protokol za zagotavljanje varnosti v brezžičnih omrežjih (IEEE 802.11), ki se uporablja od samega začetka uporabe (1997). Za šifriranje prenesenih podatkov uporablja RC4. Izmenjava poteka preko paketov (okvirjev) šifriranih s skritim ključem (40-bitni in 104-bitni) in 24-bitnim IV, ki je ključu dodan na koncu. Za podatkovno celovitost paketa je na koncu dodana ciklična redundančna koda – CRC-32 (angl. cyclic redundancy check). Skriti ključ je enak za vse udeležence v omrežju in se hkrati uporablja tudi za avtentikacijo.
Če zanemarimo prvotno 40-bitno dolžino ključa, je največja slabost 24-bitni IV, ki zaradi majhne velikosti (pre)hitro vodi v ponovljen izhodni tok. Tako se IV pri eni seji ponovi že po ~5h (maksimalen promet pri hitrosti 11Mbit/s in velikosti paketa 12000 bitov). Če lahko spremljamo več sej hkrati (možno, ker si udeleženci delijo isti ključ) pa se bo IV s 50% verjetnostjo ponovil že po $\sim 2^{12}$ paketih (slavni paradoks rojstnega dneva) oz. nekaj sekundah.
Naslednja slabost je uporaba CRC za preverjanje celovitosti podatkov, saj je CRC linearna in velja: $RC4([P, CRC(P)]) \oplus [X, CRC(X)] = RC4([P \oplus X, CRC(P \oplus X)]$. Torej lahko šifriran paket preprosto spremimo, tako da paketu XOR prištejemo poljubno vrednost z ustreznim CRC povzetkom in rezultat je zopet veljaven šifriran paket.
Po objavi pomanjkljivosti v algoritmu KSA v avgustu 2001[3] je bilo jasno, da je WEP dokončno razbit in ni več uporaben za okolja, ki zahtevajo varnost. Na podlagi odkrite korelacije je bil izdelan prosto-dostopni program aircrack-ng[3], ki je vsakomur omogočal, da po zajemu 4 – $6 \cdot 10^6$ paketov pridobi ključ. Dodatne optimizacije je leta 2004 objavil heker KoReK, kar je zmanjšalo zahtevnost na 0.5 – $2 \cdot 10^6$ paketov. Na podlagi novega napada [7] je bil leta 2007 izdelan še program aircarck-ptw[4], ki z verjetnostjo 50% pridobi 104-bitni ključ že po 40.000 paketih oz. slabi minuti spremljanja prometa.

4.2. WPA (Wi-Fi Protected Access)

Združenje Wi-Fi (Wi-Fi Alliance), ki je bilo odgovorno za WEP, je leta 2003 izdalo nov varnostni protokol WPA, ki naj bi odpravil pomanjkljivosti predhodnika. WPA nikoli direktno ne uporablja gesla, ki ga vnese uporabnik (8-63 ASCII znakov), pač pa najprej zgradi 256-bitni ključ PKS (pre-shared secret) iz gesla in identitete omrežja (SSID) po standardu PBKDF2. Za samo šifriranje pa iz ključa PKS in strojnega naslova mrežne kartice (MAC) po postopku TKIP (Temporal Key Integrity Protocol) izpelje 128-bitni šifrirni ključ, ki je skupaj z 48-bitnim IV vhod v algoritem RC4. TKIP vključuje tudi mehanizem za menjavo ključa (angl. rekeying), kar onemogoča podvajanje IV pri istem ključu. Namesto povzetka CRC-32 uporablja povzetek

[3] http://aircrack-ng.org
[4] http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/

MICHAEL, ki pri izračunu uporablja ključ (drugačen od šifrirnega) in ga tako ni možno več enostavno ponarediti.
Z vsemi opisanimi postopki noben od prejšnjih napadov ni več mogoč, zato so hekerji posegli predvsem po napadu s slovarjem (angl. dictionary attack), kjer preizkusijo vse najpogostejše besede, ki se uporabljajo za gesla. Napad je možen le pri različici WPA brez avtentikacijskega serverja, ker potem samo-avtentikacija temelji na ključu PKS, ki si ga deli celotno omrežje. S tem omogoča gradnjo vnaprej pripravljenih ključev in t.i. časovno-pomnilniški kompromis (angl. time-memory trade-off), ki se izogne zamudnemu preračunavanju 4096 SHA-1 zgostitev, ki jih sama izpeljava ključa PKS zahteva. Na internetu je prosto-dostopni program coWPAtty[5], ki ima na voljo tabele za 400.000 najpogostejših gesel za 1000 najpogostejših identitet SSID.
Novembra 2008 je bila odkrita ranljivost v TKIP v povezavi s povzetkom MICHAEL [1]. Ranljivost omogoča, da napadalec pridobi ključ, ki je uporabljen za izračun povzetka. S tem napadalec ne pridobiti glavnega ključa in tudi ne more prisluškovati ali spreminjati obstoječega prometa, lahko pa v omrežje vrine do največ 7 lastnih veljavnih paketov.
Združenje Wi-Fi je leta 2004 predstavilo tudi naslednika WPA – WPA2, ki pa ne uporablja več tekoče šifre RC4 pač pa bločno šifro AES v načinu CCM.

4.3. TLS/SSL (Transport Layer Security /Secure Sockets Layer)

TLS/SSL je zbirka kriptografskih protokolov za zagotavljanje varnosti in integritete prometa preko TCP/IP povezave (interneta). Seja TLS se najprej vzpostavi s standardnim pozdravom, ki vključuje podprte algoritme tako odjemalca kot strežnika, sledi varna izmenjava naključno generiranih ključev in opcijsko avtentikacija. Varno izmenjani ključi so osnova za izpeljavo glavnega ključa, ki se uporablja v nadaljevanju seje za simetrično šifriranje. V osnovi TLS podpira več simetričnih šifer (AES, DES, 3DES...), vendar je le RC4 tokovna in seveda najhitrejša. Integriteta sporočil je zagotovljena s povzetkom HMAC-SHA1 (HMAC-MD5), ki pri izračunu uporablja drug ključ kot za šifriranje.
Za razliko od WEP/WPA se tukaj ne uporablja IV, ampak se skozi celotno sejo uporablja isti 128-bitni ključ kot vhod v RC4. To je mogoče zato, ker TLS sloni na TCP/IP, ki zagotavlja prenos in sinhronizacijo paketov, in zato lahko šifriranje/odšifriranje poteka neodvisno od paketov za ves čas trajanja seje. Seveda pa se lahko seja prekine in ponovno vzpostavi, vendar se v tem primeru zgradi drug glavni ključ. Zaenkrat edina odkrita pomanjkljivost pri uporabi RC4 v TLS protokolu je prepogostost pojavitve 0 kot drugega izhoda šifre [9], ki pa ni posledica pomanjkljive implementacije.

4.4. Datoteka PWL v MS Windows 9x

Za hranjenje uporabniških gesel (npr: za omrežne deljene vire) se v sistemih MS Windows 95/98/Me uporablja posebna datoteka PWL (Password List). Vsak uporabnik ima svojo datoteko in v njej se lahko shrani največ 255 podatkov o virih, ki vključujejo ime in tip vira ter geslo za dostop. Datoteka je šifrirana z RC4 algoritmom z 32-bitnim ključem, ki je izpeljan iz uporabniškega prijavnega gesla. Velikost ključa je bila tudi 1995 veliko premajhna in zato je Microsoft kmalu izdal popravek, ki ključ povečal na 128-bitov.

[5] http://www.willhackforsushi.com/Cowpatty.html

Podobna baza gesel je prisotna tudi v sistemih MS Windows NT/XP/Vista. Imenuje se podatkovna baza SAM (Security Accounts Manager) in tokrat hrani le zgostitev gesla. Zaradi pojava napada s predizračunanimi tabelami zgostitev, ki omogočajo pridobitev gesla, je podatkovno bazo mogoče zaščititi s šifro RC4, ki uporablja 128-bitni ključ. Žal je bila začetna implementacija pomanjkljiva[6] in je uporabljala isti ključ za šifriranje vseh zgostitev, kar je vodilo v ponovljen izhodni tok in ni onemogočalo omenjanega napada. Microsoft je napako kmalu popravil in šifrirana podatkovna baza SAM (trenutno) velja za varno.

4.5. MS Office

Pisarniški paket MS Office že od verzije 95 dalje ponuja možnost šifriranja dokumentov. Če je verzija 95 ponujala le XOR prištevanje uporabniškega gesla čistopisu, kar ne zadosti nobenemu kriptografskemu standardu, so kasnejše verzije ponudile možnost šifriranja z RC4. Na začetku nezadostno s 40-bitnim ključem, nato pa z ustreznim 128-bitnim ključem.
Leta 2005 [15] so odkrili, da vseeno Microsoft zopet ne uporablja šifre tako kot bi jo moral, saj nove različice zaščitenega dokumenta šifrira z istim ključem in IV, kar vodi v ponovljen izhodni tok in omogoča dokaj enostavno odšifriranje. Napaka velja za vse pakete MS Office od verzije 97 do 2003.
Trenutno aktualni pisarniški paket MS Office 2007 je prinesel bločno šifriranje AES s 128 ali 256-bitnim ključem, kar odpravlja opisano pomanjkljivost.

Predstavljeni seznam ni popoln in šifra je prisotna tudi še na drugih področjih: PPTP – Microsoftov protokol za navidezna privatna omrežja (nadomestil IPsec, ki uporablja 3DES in AES), Lotus Notes, Oracle SQL ... Opazimo, da je šifra RC4 zelo pogosta in jo uporablja praktično vsak računalnik. Zato lahko trdimo, da je šifra RC4 najpogostejša tokovna šifra v programski opremi (v strojni opremi se največ uporablja A5/x pri kodiranju GSM komunikacije). Žal ima šifra nekaj ranljivosti, ki same po sebi niso zelo kritične, če se šifra sama uporablja v skladu s pravili. Vsekakor dosti več škode kot omenjene pomanjkljivosti šifre naredi njena nepravilna implementacija/uporaba. Žal je najlepši primer (preveč) preprost protokol WEP in večina implementacij s strani Microsofta. Zaradi takih zloglasnih primerov je sicer dobra šifra prišla na zelo slab glas in splošna publika se odmika od njene uporabe, čeprav do sedaj ni bilo odkritih praktičnih načinov, kako uporabiti odkrite ranljivosti za razbitje šifre RC4 pri uporabi v TLS ali WPA.

[6] http://www.mail-archive.com/bugtraq@securityfocus.com/msg01990.html

5. Povzetek

Šifra RC4 je prisotna skoraj na vsakem računalniku predvsem po zaslugi hitrosti in preprostosti. Tekom več kot 20-letnega obstoja je bila tarča veliko kriptoanaliz in odkritih je bilo kar precej pomanjkljivosti. Razširjenost, slaba implementacija in odkrite pomanjkljivosti so šifro v zadnjih letih spravile na slab glas. Posledično jo začenjajo počasi nadomeščati bločne šifre, ki so bolj raziskane, razumljive in standardizirane. Z druge strani obstaja kar nekaj konkurenčnih tokovnih šifer, še posebej, ker je med letoma 2004 in 2008 potekal projekt eSTREAM, katerega cilj je bil najti ustrezne nove tokovne šifre, ki bi ob poznanih napadih nudile ustrezno varnost.

Glavne ranljivosti šifre izhajajo iz njene preprostosti, predvsem iz preveč poenostavljene vzpostavitve začetnega notranjega stanja (KSA) na podlagi ključa. Algoritem tudi ne nudi eksplicitne podpore inicializacijskemu vektorju (IV) in za ohranitev enostavnosti se uporabi dodajanje na začetku ali koncu ključa, kar povzroča dodatne možnosti razbitja.

Pravilna uporaba, ki je odporna na vse praktične napade, zahteva dobro mešanje ključa in IV, priporočeno s kakšno kriptografsko varno zgoščevalno funkcijo (npr: SHA-1), in neuporabo vsaj prvih 256 izhodov (priporočeno 768 ali več). Dodatno mora biti ključ dovolj velik (128-bitov ali več) in »neodvisen« od ostalega sistema. Če že je odvisen od uporabniškega gesla, mora le-ta biti dovolj dolg in nevsakdanji (mešanica velikih in malih črk, številk in ločil), sam ključ pa mora biti le izpeljan iz gesla (s pomočjo zgoščevalne funkcije). Tudi IV mora biti ustrezno velik (vsaj 48-bitov), da ne pride do ponovljenega toka ključev, še boljše pa je možnost avtomatske zamenjave ključev po določenem (kratkem) času. Žal vsi omenjeni postopki vplivajo na hitrost in predvsem na večjo kompleksnost šifre.

Starost šifre se pozna tudi na tem, da je prilagojena za 8-bitne procesorje in zaradi velike zaporednosti ne zna izkoristiti paralelnosti današnjih 64-bitnih večjedrnih procesorjev. Zaradi velikega notranjega stanja tudi ni najbolj primerna za uporabo v »mikro« okoljih, kjer šteje vsak bit (npr: značke RFID).

Šifro RC4 bodo v prihodnosti sigurno zamenjale nove šifre (bločne ali tokovne), ki bodo odporne na odkrite ranljivosti, bodo imele eksplicitno podporo za IV in bodo znale izkoriščati današnje 32/64-bitne večjedrne procesorje. Vsekakor pa je lahko šifra RC4 z nekaj popravki ob pravilni uporabi vsaj še nekaj let brez problema konkurenčna in glede na to, da ima dandanes skoraj vsaka »pametna« naprava vsaj 8-bitni procesor z več kot dovolj pomnilnika za brezskrbno in hitro uporabo šifre, jo bo na tem področju zelo težko nadomestiti.

Literatura

[1] M. Beck, E. Tews. *Practical attacks against WEP and WPA*. Cryptology ePrint Archive, 2008.

[2] H. Finney. *An RC4 cycle that can't happen.* Objava na sci.crypt, 1994.

[3] S. Fluhrer, I. Mantin, A. Shamir. *Weaknesses in the key scheduling algorithm of RC4*. Selected Areas in Cryptography, 2001.

[4] S. Fluhrer, D. McGrew. *Statistical Analysis of the Alleged RC4 Keystream Generator*. Fast Software Encryption, 2000.

[5] J. Golič. *Linear Statistical Weakness of Alleged RC4 Keystream Generator*. Eurocrypt, 1997.

[6] A. Grosul, D. Wallach. *A related-key cryptanalysis of RC4*. Rice University, 2000.

[7] A. Klein. *Attacks on the RC4 stream cipher*. 2006.

[8] L. Knudsen, W. Meier, B. Prennel, V. Rijmen, S. Verdoolaege. *Analysis methods for (alleged) RC4.* Asiacrypt, 1998.

[9] I. Mantin, A. Shamir. *A Practical Attack on Broadcast RC4*. Fast Software Encryption, 2001.

[10] A. Maximov, D. Khovratovich. *New state recovery attack on RC4.* Cryptology ePrint Archive, 2008.

[11] I. Mironov. *(Not So) Random Shuffles of RC4.* Crypto, 2002.

[12] S. Paul, B. Preneel. *A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher*. Fast Software Encryption, 2004.

[13] M. Robshow. *Stream Ciphers.* RSA Laboratories, 1995.

[14] A. Ross. *A class of weak keys in the RC4 stream cipher*. Objava na sci.crypt, 1995.

[15] H. Wu. *The Misuse of RC4 in Microsoft Word and Excel*. Cryptology ePrint Archive, 2005.