

# Zotrvalé stavy prúdovej šifry RC4

## 1 Úvod

V roku 1987 navrhol známy kryptológ Ron Rivest pre firmu RSA Data Security Inc. prúdovú šifru RC4<sup>1</sup>. Táto šifra bola implementovaná v produktoch firmy RSA Data Security, ale jej presná špecifikácia bola udržiavaná v tajnosti. Zmena nastala v roku 1994, keď neznámy autor zverejnil jej zdrojový kód na Internete. Správnosť tohto popisu nepriamo potvrdila firma RSA v roku 2001 vo svojej reakcii na útok na údajnú RC4.

V súčasnosti je šifra RC4 najrozšírenejšia softvérovo založená<sup>2</sup> prúdová šifra na svete. Pretože RC4 je jednou z najpoužívanejších šifier v SSL/TLS, jej najčastejšie využitie práve je ochrana prenášaných dát na Internete. RC4 bola taktiež implementovaná do množstva komerčných produktov ako napríklad Microsoft Office, Lotus Notes alebo Oracle SQL.

## 2 Popis RC4

Šifra RC4 priťahovala a priťahuje pozornosť kryptológov nielen svojim rozšíreným využitím, ale predovšetkým svojou jednoduchosťou. V porovnaní so zložitými popismi blokových šifier sa pri pohľade na popis šifry RC4 vynára otázka, ako až jednoduchá môže byť šifra, aby ešte bola bezpečná.

Šifra RC4 je v skutočnosti skupina algoritmov parametrizovaná veľkosťou binárnych slov nad ktorými šifra pracuje. Označme tento parameter  $n$  a príslušnú šifru ako  $RC4_n$ . Popíšme si šifru  $RC4_n$ .

Nech  $n$  je prirodzené číslo a nech  $N = 2^n$ . Vnútorňý stav šifry  $RC4_n$  pozostáva z permutácie  $S \in \mathcal{S}_N$  všetkých  $N$  bitových slov a dvoch indexov  $i, j \in \{0, \dots, N - 1\}$ . Šifra  $RC4_n$  (rovnako ako väčšina prúdových šifier) môže byť rozdelená do dvoch častí. V prvej, nazvanej Key Scheduling Algorithm (KSA), sa z kľúča (zvyčajne 40 až 256 bitov dlhého) odvodí permutácia  $S \in \mathcal{S}_N$  (KSA je jediná časť  $RC4_n$ , ktorá je závislá na kľúči). Výstupná permutácia z KSA je následne použitá ako vstup do druhej časti RC4, nazvanej Pseudo Random Generation Algorithm (PRGA). Táto v každom svojom cykle vyprodukuje  $n$  pseudonáhodných bitov, tzv. prúd kľúča (anglicky keystream). Tento sa bit po bite XOR-uje s prúdom otvoreného textu, čím získame šifrovaný text. RC4 je teda binárna aditívna synchrónna prúdová šifra.

---

<sup>1</sup>písmeno 'R' v názve RC4 je pravdepodobne skratka od mena autora 'Ron' respektíve 'Rivest' a písmeno 'C' skratka od anglického slova 'code' alebo 'cipher'

<sup>2</sup>Šifra RC4 nieje na rozdiel od väčšiny používaných prúdových šifier založená na hardvérovo orientovaných lineárnych posuvných registroch. Tieto sa ľahko implementujú v hardvéri, ale sú pomalé v softvérovej implementácii.

Nech  $K$  je kľúč a  $S$  permutácia na množine  $\{0, \dots, N - 1\}$ . Označme  $K[i]$   $i$ -ty  $n$ -bitový blok kľúča  $K$ . Pri reprezentácii permutácie  $S$  ako tabuľky s jedným riadkom a  $N$  stĺpcami, kde v  $i$ -tom stĺpci je hodnota  $S(i)$ , môžeme zloženie permutácie  $S$  s transpozíciou  $(i, j)$  popísať ako výmenu (swap) stĺpcov  $i$  a  $j$ . V súlade s touto reprezentáciou môžeme hodnotu  $S(i)$  označiť aj ako  $S[i]$ .

Číslo  $l \in \mathbb{N}$  bude označovať najmenšie prirodzené číslo rovné alebo väčšie ako podiel bitovej dĺžky kľúča  $K$  a čísla  $n$ , je teda  $l = \lceil |K|/n \rceil$ . Algoritmus 1 a algoritmus 2 popisujú obe časti šifry RC4.

---

**Algorithm 1** KSA - The Key Scheduling Algorithm of  $RC4_n$

---

**Vstup:** kľúč  $K$   
**Výstup:** permutácia  $S \in S_N$

- 1:  $N \leftarrow 2^n$
- 2:  $l \leftarrow$  počet  $n$ -bitových blokov  $K$
- 3: **for**  $i = 0$  to  $N - 1$  **do**
- 4:    $S[i] = i$
- 5: **end for**
- 6:  $j \leftarrow 0$
- 7: **for**  $i = 0$  to  $N - 1$  **do**
- 8:    $j \leftarrow (j + S[i] + K[i \bmod l]) \bmod N$
- 9:   Swap( $S[i], S[j]$ )
- 10: **end for**

---



---

**Algorithm 2** PRGA - The Pseudo Random Generation Algorithm of  $RC4_n$

---

**Vstup:** permutácia  $S \in S_N$   
**Výstup:** prúd kľúča

- 1:  $N \leftarrow 2^n$
- 2:  $i \leftarrow 0$
- 3:  $j \leftarrow 0$
- 4: **loop**
- 5:    $i \leftarrow i + 1 \bmod N$
- 6:    $j \leftarrow (j + S[i]) \bmod N$
- 7:   Swap( $S[i], S[j]$ )
- 8:   **Output**  $S[(S[i] + S[j]) \bmod N]$
- 9: **end loop**

---

Z popisu KSA vidíme, že vstupný kľúč môže mať dĺžku až  $n \cdot 2^n$  bitov. Pretože je ale tento kľúč použitý na generovanie permutácie  $2^n$  prvkov, je efektívna dĺžka kľúča rovná maximálne  $\log_2(2^n!)$  bitov.

Všetky známe implementácie RC4 majú dĺžku slova 8 bitov, ide teda o šifru RC4<sub>8</sub>. Vnútorý stav tejto šifry pozostáva z permutácie  $S$  256 prvkov a dvoch indexov  $i, j \in \{0, \dots, 255\}$ . Máme teda  $256!$  možností pre  $S$  a  $256^2$  možností pre  $i$  a

$j$ . Spolu je to približne  $2^{1700}$  možných vnútorných stavov, čo natľko veľké, že útoky typu Time/Data/Memory Tradeoff sú na RC4<sub>8</sub> nepoužiteľné. Navyše sa vnútorný stav RC4 vyvíja nelineárnym spôsobom a je teda mimoriadne zložitú použiť čiastočné informácie o aktuálnom vnútornom stave na určenie vnútorného stavu v budúcnosti.

### 3 Stručné zhrnutie výsledkov diplomovej práce

Ako bolo popísané vyššie, vnútorný stav šifry RC4<sub>8</sub> má entropiu približne 1700 bitov. Je teda prirodzená otázka, či neexistuje iba čiastočne určený vnútorný stav, ktorý by správanie šifry RC4 "výrazne" ovplyvňoval a v priebehu algoritmu PRGA sa vyvíjal predpovedateľným spôsobom.

Ďalej v texte sa budeme venovať šifre RC4 obecné. T.z. budeme skúmať stavy šifry RC4 <sub>$n$</sub>  pre ľubovoľné  $n \in \mathbb{N}$ . Na úvod si uvedme definíciu parciálneho vnútorného stavu šifry RC4 <sub>$n$</sub> . Pre jednoduchosť zápisu položíme  $N = 2^n$ .

**Definition 1.** *Nech  $i \in \mathbb{Z}_N$ ,  $j \in \mathbb{Z}_N$  a nech  $\sigma$  je parciálna permutácia na  $\mathbb{Z}_N$  s  $|\text{Dom}(\sigma)| = k > 0$  (teda prosté zobrazenie zo  $\mathbb{Z}_N$  do  $\mathbb{Z}_N$  definované na  $k$  prvkoch). Trojicu  $T = (i, j, \sigma)$  nazývame  $k$ -stavom.*

$k$ -stav je teda parciálny stav šifry RC4, pri ktorom poznáme oba indexy  $i$  aj  $j$  a poznáme hodnoty permutácie  $S$  na  $k$  prvkoch.

Jedným z typov, v literatúre doteraz popísaných stavov, sú takzvané *predpovedajúce* stavy (anglicky predictive states). Tieto nám umožňujú predpovedať niekoľko najbližších výstupov algoritmu PRGA.

V tomto texte uvádzam pre jednoduchosť iba neformálne definície konkrétnych pojmov. Presné definície je možné nájsť v plnej verzii diplomovej práce.

**Definition 2.** *Nech  $T$  je  $k$ -stav.  $T$  nazveme  $l$ -predpovedajúcim, ak každé zúplnenie  $k$ -stavu  $T$  na plný RC4 stav má zhodných nasledujúcich  $l$  výstupov algoritmu PRGA.*

Aj keď nám  $l$  predpovedajúci  $k$ -stav pri jeho výskyte umožní predpovedať nasledujúcich  $l$  výstupov, využitie tejto triedy stavov v kryptoanalýze je obmedzené tvrdením, že pre každý  $l$  predpovedajúci  $k$ -stav je nutne  $l \leq k$ .

Druhým dôležitým, v literatúre popísaným typom parciálnych stavov RC4 sú takzvané *výhodné* stavy. Ide o stavy, pri ktorých je možné sledovať hodnotu indexu  $j$  najbližších niekoľko cyklov PRGA.

**Definition 3.** *Nech  $T$  je  $k$ -stav.  $T$  nazveme  $l$ -výhodným, ak každé zúplnenie  $k$ -stavu  $T$  na plný RC4 stav má nasledujúcich  $l$  cyklov algoritmu PRGA rovnakú hodnotu indexu  $j$ .*

Na rozdiel od predpovedajúcich stavov, nie je na prvý pohľad zrejmé žiadne kryptoanalytické využitie výhodných stavov. V literatúre týmto stavom nebola venovaná žiadna pozornosť.

Nás naopak zaujala práve určitá podtrieda výhodných stavov, ktoré sme nazvali stavy zotrvalé.

**Definition 4.** *Nech  $T$  je  $k$ -stav.  $T$  nazveme zotrvalým, ak každé zúplnenie  $k$ -stavu  $T$  na plný RC4 stav má rovnakú hodnotu indexu  $j$  v celom nasledujúcom priebehu algoritmu PRGA.*

Zotrvalý  $k$ -stav je teda  $\infty$ -výhodný  $k$ -stav. Otázka nájdenia zotrvalého stavu nie je zaujímavá iba z kryptologického, ale aj z matematického hľadiska. Znamenalo by to nájdenie stabilného prvku v dynamickom systéme, ktorého vývoj je popísaný algoritmom PRGA.

Z kryptologického hľadiska bude pre nás dôležitý taktiež pojem dosiahnuteľnosti stavu.

**Definition 5.** *Nech  $T$  je  $k$ -stav.  $T$  nazveme dosiahnuteľným, ak existuje kľúč  $K$  taký, že  $(0, 0, KSA(K))$  je zúplnením stavu  $T$ .*

Dosiahnuteľné sú teda také  $k$ -stavy, pre ktoré existuje aspoň jedno zúplnenie  $(i, j, S)$  také, že  $i = 0$ ,  $j = 0$  a  $S$  je výstupom algoritmu KSA pre nejaký kľúč  $K$ .

Úvod výskumnej časti diplomovej práce pozostával z vybudovania základov teórie zotrvalých stavov. V tejto časti sa nám podarilo popísať chovanie týchto stavov a dokázať niekoľko tvrdení o vlastnostiach týchto stavov.

Ďalej sme svoju pozornosť zamerali na istú podskupinu zotrvalých stavov, ktoré sme nazvali monotónne. Tieto stavy by sa dali neformálne popísať ako stavy, pri ktorých je index  $j$  "väčší" ako index  $i$ . Táto skupina stavov sa vyznačuje vlastnosťou, že každý takýto stav je invariantný na zmenu parametru  $n$ . Teda kým inému typu útoku na RC4 by sa s veľkou pravdepodobnosťou dalo predísť zväčšením hodnoty parametru  $n$  z v súčasnosti používaného  $n = 8$  na napríklad  $n = 16$ , útoky pomocou monotónnych stavov ostávajú účinné pri ľubovoľnej hodnote parametru  $n$ .

Prvým našim krokom pri hľadaní monotónneho zotrvalého stavu bolo hľadanie vhodného matematického modelu pre popis týchto stavov. Potrebovali sme čo najjednoduchší popis parciálneho stavu RC4, ale taktiež jednoduchosť popisu akcie algoritmu PRGA na tento parciálny stav.

Prirodzená, vyššie uvedená reprezentácia stavu ako trojice pozostávajúcej z dvoch indexov a parciálnej permutácie je sídce prímociará reprezentácia vnútorného stavu RC4, avšak pridávanie transpozícií k permutácii nieje príliš prehľadné. Tento popis je navyše dosť obecný. Dajú sa ním totiž popísať všetky možné stavy RC4, kým my sme potrebovali popísať iba zotrvalé stavy.

V práci navrhujeme a popisujeme takzvaný tabuľkový model pre monotónne zotrvalé stavy. V úvode kapitoly o tabuľkovom modeli sme dokázali jednoznačnú súvislosť medzi týmto modelom a zotrvalými stavmi. Následne sa nám podarilo pomocou tohto modelu výrazne rozvynúť teóriu zotrvalých stavov a dokázať niekoľko hlbších tvrdení o týchto stavoch.

V kapitole 5 (nazvanej Quest for persistent states) sme pomocou vybudovanej teórie popísali sústavu lineárnych rovníc, ktoré musí každý monotónny zotrvalý stav spĺňať. Následne sa nám podarilo dokázať regulárnosť matice zadanej týmito rovnicami a teda existenciu jednoznačného riešenia tejto sústavy pre daný vektor pravých strán (riešením sústavy je práve zotrvalý stav).

V práci je toto tvrdenie dokázané pre obecnější prípad a to ako dôkaz regulárnosti matice zadanej ľubovoľnou ekvivalenciou na lineárne usporiadanej množine. Taktiež sme pomocou regulárnosti tejto matice ukázali jednoznačnú súvislosť medzi monotónnymi zotrvalými stavmi a ekvivalenciami na lineárne usporiadanej množine. Ďalšie dosiahnuté výsledky pre ekvivalencie sme následne aplikovali na teóriu zotrvalých stavov.

Ako náš výskum zotrvalých stavov pokračoval, postupne sme sa preorientovali od hľadania zotrvalého stavu na hľadanie dôkazu neexistencie dosiahnuteľného monotónneho zotrvalého stavu. V práci sa nám nakoniec podarilo dokázať neexistenciu monotónneho zotrvalého  $k$ -stavu pre  $k$  rovné 2, 3, 4. V priebehu práce sme taktiež objavili niekoľko tried zotrvalých stavov, všetky sú však nedosiahnuteľné (ich využitie v kryptoanalýze je však možné pomocou techník *fault injection*).

Na záver diplomovej práce popisujeme využitie zotrvalých stavov v kryptoanalýze. Nasleduje stručný popis tohto využitia.

Predstavme si, že poznáme nejaký dosiahnuteľný zotrvalý  $k$ -stav  $T = (i_0, j_0, \sigma_0)$  a predpokladajme, že aktuálny vnútorný stav šifry RC4  $(i, j, S)$  je zúplnením tohto stavu. Je teda  $i = i_0$ ,  $j = j_0$  a  $\sigma_0$  restrikcia permutácie  $S$ . Pretože je aktuálny vnútorný stav zúplnením zotrvalého stavu, má predpísané zmeny indexu  $j$ , a má teda do určitej miery monotónne správanie (v práci je dokázaná periodičnosť zotrvalých stavov). Táto monotónnosť môže byť následne detekovaná na prúde kľúča (teda pri znalosti otvoreného textu). Pretože pre konkrétne zotrvalé stavy môžeme detailne popísať ich vplyv na zvyšok permutácie v priebehu algoritmu PRGA, je možné v niektorých krokoch PRGA pomocou týchto stavov určiť hodnoty permutácie na výstupe a ich vzory. Pretože sú zotrvalé stavy zo svojej podstaty "večné" (nezanikajú v priebehu cyklov PRGA), mohli by sme z popisu konkrétneho stavu a aktuálneho výstupu RC4 spätne zrekonštruovať celý vnútorný stav RC4 (pri nedostatku otvoreného textu je možné zrekonštruovať pomocou zotrvalého stavu iba časť vnútorného stavu a použiť jeden z v literatúre popísaných algoritmov na rekonštrukciu zvyšku vnútorného stavu). Celý tento postup je v diplomovej práci taktiež demonštrovaný na konkrétnom príklade nami nájdeného, nedosiahnuteľného zotrvalého stavu.

Vidíme teda, že pokiaľ by sme našli nejaký dosiahnuteľný zotrvalý stav RC4, bolo by možné detekovať výskyt tohto stavu a následne previesť known-plaintext attack na šifru RC4. Pritom na úspešné prevedenie útoku by stačila znalosť iba malej časti otvoreného textu, pomocou ktorej by sme zrekonštruovali celý vnútorný stav RC4 a následne boli schopní dešifrovať celý nasledujúci šifrovaný text. Pretože táto známa časť otvoreného textu môže byť napríklad HTTP hlavička alebo počiatočné informácie o dokumente napísanom vo Worde, dal by sa tento útok pri reálnom

použití považovať aj za ciphertext-only attack.

## 4 Záver

V tomto dokumente som sa snažil stručne popísať svoju diplomovú prácu. Pretože ale táto obsahuje 24 strán nami popísaných a následne dokázaných lemmat, tvrdení a viet, tento popis nemôžno považovať ani zďaleka za úplný.

Napriek tomu, že sa RC4 venovala v literatúre značná pozornosť, žiadna doteraz publikovaná práca neobsahovala matematickú teóriu vnútorných stavov RC4. Predložená diplomová práca je v tomto zmysle inovátorská; vybudovaná teória sa nepodobá žiadnej doteraz publikovanej práci o RC4.

## 5 Copyright

Predložená práca doteraz nebola publikovaná. Autor v súčasnosti pracuje na jej rozšírení a bude sa snažiť o publikáciu na niektorej z medzinárodných konferencií.