RC4の弱鍵の一般化

寺村 亮一† 大東 俊博‡ 桑門 秀典† 森井 昌克†

†神戸大学大学院工学研究科
657-8501 兵庫県神戸市灘区六甲台町１－１

{teramura@stu., kuwakado@, mmorii@}kobe-u.ac.jp

‡広島大学情報メディア教育研究センター
739-8511 広島県東広島市鏡山１－４－２

ohigashi@hiroshima-u.ac.jp

あらまし 本稿ではRC4における特定の出力系列から高い確率で一部を回復できる秘密鍵 (弱鍵) を提案する．新たに提案する弱鍵は，鍵空間中に高い割合で存在し，かつ過去にRC4に対して提案されてきた弱鍵をその一部として含む．すなわちそれらはRC4の弱鍵を一般化したものである．またこれら一般化した弱鍵を効率よく探索するアルゴリズムを提案し，実際に過去に提案されてきた弱鍵がその一部として導出できること，そして過去に報告がなかった多くの鍵が今回提案する弱鍵に含まれることを示す．これら一般化した弱鍵の考え方を利用することで，攻撃者はRC4の鍵の導出を効率よく行うことが可能となる．

A Generalization of the RC4's Weak Keys

Ryoichi Teramura† Toshihiro Ohigashi‡
Hidenori Kuwakado† Masakatu Morii†

†Graduate School of Engineering, Kobe University
1-1, Rokkodai, Nada-ku, Kobe, Hyogo 657-8501, Japan

{teramura@stu., kuwakado@, mmorii@}kobe-u.ac.jp

‡Information Media Center, Hiroshima University
1-4-2, Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8511

ohigashi@hiroshima-u.ac.jp

Abstract In this paper, we propose a new class of the weak key in RC4 and generalize the class of the weak keys in RC4. The generalized weak keys consists of Roos' weak key, Wagner's weak key and the numerous keys that no one have seen. In addition, we propose the algorithm that can search for generalized weak keys. Exploiting those, we can recover the secret key efficiently.

1 はじめに

RC4とは1987年にRivestにより開発され，アルゴリズムが公になった1994年から15年が経過した現在においても現実時間での解読法が提案されていない優れたストリーム暗号である．その安全性と、ソフトウェア上での高速動作性より，Secure Sockets Layer (SSL)，Transport

Layer Security (TLS)，Wired Equivalent Privacy (WEP)，Wi-Fi Protected Access (WPA) 等，様々な標準通信規格の暗号アルゴリズムとして採用されている．

RC4 の内部状態は可変の値 n をベースとした一つの置換配列と二つのポインタから構成され，多くのソフトウェアで使用される $n = 8$ の場合においては，およそ 1700 ビットもの非常に大きな値をとる．またその秘密鍵は可変の長さをとる．

RC4 はその利用規模の大きさより，様々な研究機関から解析結果が報告されている．解析手法の一つとして，弱鍵や等価鍵 [1] など鍵の性質に着目した研究がある．弱鍵に関する代表的な研究としては，1995 年に Roos により報告された弱鍵のクラスが挙げられる [2]．また同年には Wagner により，Roos による報告とは異なる弱鍵が報告されている [3]．この Wagner の弱鍵を用いた関連鍵攻撃は，Fluhrer らにより提案された WEP に対する鍵回復攻撃 (FMS 攻撃)[4] 中で利用されており，また我々も以前提案した WEP に対する鍵回復攻撃中において一部この考え方を利用している [5].

従来，これら弱鍵はそれぞれ独立のものとして扱われており，それら弱鍵となりうる鍵の組み合わせも限られていたため，RC4 単体の安全性に与える影響もまた限定されたものであった．

本稿では，新しい RC4 の弱鍵を定義し，それらを探索するアルゴリズムを提案する．提案する弱鍵は，以前我々が提案した初期予測状態を利用して定義したものであり，その集合の中に Roos の弱鍵，Wagner の弱鍵を部分集合として含んでいる．またアルゴリズムを用いた実験結果より，過去に報告されていない多くの数の弱鍵がいまだ RC4 に存在することを明らかにする．すなわち我々の提案する弱鍵は RC4 に存在する弱鍵を一般化したものと考えられる．

本稿の構成を次に示す．二章において RC4 の概要，そして過去に提案された弱鍵についての説明を，三章において今回新たに提案する弱鍵の定義を行う．四章においてそれら弱鍵を探索するアルゴリズムの説明，そして実際にそれらが過去に提案した弱鍵を含んでいることを示す．

RC4 KSA:
1: **for** $\forall x \in \{0, 1, \ldots, N-1\}$
2: $\quad S[x] \leftarrow x$
3: **end for**
4: $j \leftarrow 0$
5: **for** $\forall i \in \{0, 1, \ldots, N-1\}$
6: $\quad j \leftarrow j + S[i] + K[i \bmod l]$
7: $\quad$ Swap $S[i]$ and $S[j]$
8: **end for**

RC4 PRGA:
1: $i \leftarrow 0$
2: $j \leftarrow 0$
3: **loop**
4: $\quad i \leftarrow i + 1$
5: $\quad j \leftarrow j + S^*[i]$
6: $\quad$ Swap $S^*[i]$ and $S^*[j]$
7: $\quad$ Output $z \leftarrow S^*[(S^*[i] + S^*[j])]$
8: **end loop**

図 1: RC4 の概要

五章はまとめである．

2 RC4

本節では，RC4 の概要，そして過去に提案された弱鍵の説明を行う．

2.1 RC4 の概要

RC4 のアルゴリズムは $n \cdot l$ ビットの長さを持つ鍵 K から時刻 0 の内部状態 (初期状態) を生成する鍵スケジューリングアルゴリズム (KSA: Key Scheduling Algorithm) と，内部状態からキーストリーム z を生成する擬似乱数生成アルゴリズム (PRGA: Puseudo Randam number Generation Algorithm) の二つのパートより構成される．それぞれのアルゴリズムの詳細を図 1 に示す．RC4 の内部状態は $N(= 2^n)$ 個の要素を持つ置換配列 S と二つのポインタ i，j から構成される．以降，説明を容易にするため，以下の記号を定義する．

- S: KSA 時における置換配列
- S^*: PRGA 時における置換配列 ($S_0^* = S_{N-1}$)

- $S_i[x]$: KSA の時刻 i における x 番目のインデックスの要素に格納されている値

また本稿において全ての演算は N を法として行う．

2.2 過去に提案された弱鍵

RC4 における弱鍵として，Roos，Wagner が提案したものがそれぞれ挙げられる．本節ではそれら過去に提案された弱鍵を紹介する．

2.2.1 Roos の弱鍵

Roos の弱鍵の条件は以下に示す式で与えられる．

$$K[0] + K[1] = 0 \bmod N \tag{1}$$

この条件を満たしている場合，同様に文献中で提案されている線型方程式より，$S_0^*[1]$ に 1 が，$S_0^*[2]$ に $K[2]+3$ がそれぞれ高い確率で代入される．このとき PRGA の時刻 1 において $i=1$，$j=1$ となり，$z_1 = S_0^*[S_0^*[1]+S_0^*[1]] = K[2]+3$ が出力される．すなわち Roos の弱鍵では，ある 1 バイトの出力は $K[0]$，$K[1]$ という高々2 バイトの鍵の情報で生成されてしまい，またその出力は $K[2]$ の情報を漏らすことになる．Roos は式 (1) を満たした鍵が弱鍵のクラスに含まれると定義している．

2.2.2 Wagner の弱鍵

Wagner は文献 [3] 中で弱鍵の明確な形を定義せず，関連鍵攻撃として提案を行っている．そのため本節では，改めて Wagner の弱鍵を定義する．

Definision 1. a，b をそれらが $0 < a < b \leq l$ の関係を満たす自然数であると定義する．また $T[a]$ は以下の式より導き出される値である．

$$T[a] = S_a[0] + S_a[1] + \cdots + S_a[a] \tag{2}$$

この時，以下の条件を満たす鍵が Wagner の弱鍵となる．

$$\begin{cases} S_a[a] + S_a[T[a]] = b \\ T[x] > a, \quad x \in \{1, \ldots, a\} \end{cases} \tag{3}$$

上記の条件が満たされる場合，高い確率で $S_a^*[a] + S_a^*[T[a]] = b$ が成立する．これに Roos の線型方程式を合わせて考えることで，時刻 a の出力は以下の式であらわされる．

$$\begin{aligned} z_a &= S_a^*[S_a^*[a] + S_a^*[T[a]]] && (4)\\ &\approx S_a^*[b] && (5)\\ &\approx S_0^*[b] && (6)\\ &\approx \sum_{x=0}^{b} K[x] + \frac{b(b+1)}{2} && (7) \end{aligned}$$

これにより，出力から鍵の 1 バイト分の情報を得ることができる．

3 弱鍵の定義

本章では一般化に用いる弱鍵を定義する．本章で示す初期予測状態そして弱鍵は，我々が SCIS 2008 で提案したものと同一であるため，詳細な説明は省略する．

3.1 初期予測状態

予測状態 (predictive state) とは Mantin らにより報告された，その出力より一部が高確率で推測されうる内部状態である [6]．初期予測状態とは Mantin らの提案した予測状態の条件を，初期状態において満たしている内部状態のことを示す．キーストリームから内部状態の推測を行う際に，初期予測状態であれば従来の予測状態よりも推測成功確率が上昇する．キーストリームから初期予測状態を推測する際の成功確率は以下の式で表せる．

$$P[S_A^*|Z_B] \approx \frac{N^{b-a}}{1+N^{b-a}} \tag{8}$$

ここで S_A，Z_B はそれぞれ初期予測状態とそれが出力するキーストリームを表す．

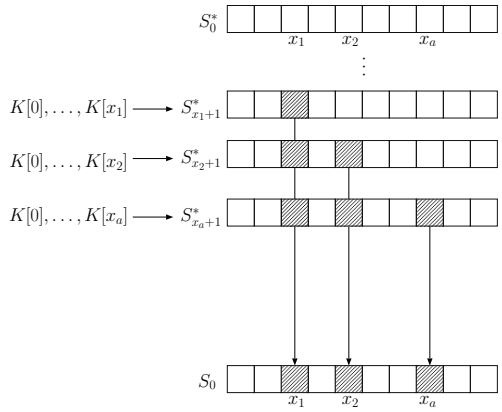

図 2: 鍵と初期状態の相関

3.2 新しい弱鍵の定義

KSA 中の内部状態 $S_1[1], S_2[2], \ldots, S_a[a]$ の値は $K[0], \ldots, K[a]$ の値のみで決定できる．また $S_x[x] = S_0^*[x] \quad x \in \{1, \ldots, a\} = A$ は式 (9) に示す確率でそれぞれの対応したインデックス以前の鍵の情報のみから生成される (図 2 参照).

$$P[S_A|S_A^*] \approx \prod_{t \notin A}^{N} \left(\frac{N - n_a}{N} \right) \tag{9}$$

ここで n_a は，t よりも小さくかつ集合 A に含まれるインデックスのを持つ要素の個数である．そして $S_0^*[0], \ldots, S_a^*[a]$ が初期予測状態ならば，その出力は必ず Z_B を出力する．すなわち鍵の要素が十分にかく拌されることなく，$K[0], \ldots, K[a]$ のみで出力が決定する．これは特定のキーストリームが出力された際に，それらを出力した鍵は式 (10) に示す確率で $K[0], \ldots, K[a]$ を有することを意味している．

$$P[K_A|Z_B] \approx \frac{N^{b-a}}{1 + N^{b-a}} \cdot \prod_{t \notin A}^{N} \left(\frac{N - n_a}{N} \right) \tag{10}$$

これらは通常の鍵よりも一部が容易に推測される弱鍵である．以上より，RC4 の弱鍵を新たに次のように定義する．

Definision 2. ある鍵の a 個のインデックスの集合を $x_1, \ldots, x_a$ とする．そしてこの鍵を用いて生成される $S_{x_a}[x_1], \ldots, Sx_a[x_a]$ が初期予測状態 $S_0^*[x_1], \ldots, S_0^*[x_a]$ と一致している場合，その鍵は弱鍵である．

4 弱鍵の一般化

本章では前章で定義した弱鍵が，従来の弱鍵と比較してどのような関係にあるのかを示す.

4.1 弱鍵探索アルゴリズム

従来の鍵バイト間における関係式で定義される弱鍵と異なり，今回新たに定義した弱鍵は，探索を行い探し出す必要がある．本節では前章で定義した弱鍵を効率よく探索するアルゴリズムを提案する．提案するアルゴリズムは以下の二つの部位に分けられる.

- 弱鍵計算部
- 初期予測状態探索部

弱鍵計算部は初期予測状態 $S_0^*[x_1], \ldots, S_0^*[x_a]$ が与えられた際に，それぞれの要素に代入されている値と同一の値を $S_{x_a}[x_1], \ldots, S_{x_a}[x_a]$ に代入していると考え，それをとりうる $K[0], \ldots, K[x_a]$ の組み合わせを導出する.

初期予測状態推測部は，初期予測状態となる配列のインデックス，そしてその初期予測状態が予測するキーストリームのバイト数の二つの値を引数にとる．初めに与えられたインデックスの要素を “既知 (known) ”としてマークする．以降，それぞれの時刻ごとに i，j が取りうる範囲を再帰的に導出していき，探索を行う．そして i，j，$S[i] + S[j]$ の指す要素がすべて既知である場合，時刻 i における出力が予測できたとする．そして予測できる出力の個数が引数として与えられた数に達したならば，そのときに配列に代入されている値を初期予測状態として出力する．このアルゴリズムの詳細を図 (3)-(5) に示す.

4.2 過去の弱鍵との関係

弱鍵探索アルゴリズムの初期予測状態探索部が導出する初期予測状態の一つとして，$S_0[1] = 1$ かつ $S_0[2] \neq 1$ ならば，その出力 $z_1 = 1$ とな

```
Search:
Input x_1, ..., x_a, b
Output predictive initial states
 1: i ← 1
 2: j ← 0
 3: for each x ∈ {x_1, ..., x_a} do
 4:    S[i] ← x
 5:    j ← x
 6:    loop
 7:       S[j] ← S[j] + 1
 8:       if S[S[i] + S[j]] is known
 9:          Swap S[i] and S[j]
10:          num ← num +1
11:          Round_S[i](i, j, S)
12:       end if
13:    loop
14:end for
15:return predictive initial states
```

図 3: 探索アルゴリズム

```
Round_S[i]:
Input i, j, S
 1: i ← i + 1
 2: if S[i] is known then
 3:    if S[i] is already assigned a value then
 4:       Round_S[j]
 5:    else if S[i] is not assigned a value then
 6:       if S[i + 1] is unknown then
 7:          S[j] ← i + 1 − j
 8:          Round_S[j]
 9:       else if S[i + 1] is known then
10:          for each x ∈ {0, ..., N − 1}
              without already set do
11:             S[i] ← x
12:             Round_S[j]
13:          end for
14:       end if
15:   end if predictive initial states
16:end if
17:return
```

図 4: i の範囲決定アルゴリズム

るものがあげられる．この初期予測状態は鍵計算部より，まず以下の状態に変換される．

$$S_2^*[1] = S_0[1] = 1 \tag{11}$$

$$S_3^*[2] = S_0[2] \neq 1 \tag{12}$$

上記の KSA 時の内部状態を生成する $K[0]$，$K[1]$，$K[2]$ の組み合わせは次のように導出できる．

- if $K[0] = 1$

$$\begin{cases} K[0] + K[1] + 0 = 0 \\ K[0] + K[1] + K[2] + 2 \neq 1 \end{cases} \tag{13}$$

- if $K[0] = 2$

$$\begin{cases} K[0] + K[1] + 1 = 1 \\ K[0] + K[1] + K[2] + 1 \neq 1 \end{cases} \tag{14}$$

- if $K[0] = x$

$$\begin{cases} K[0] + K[1] + 1 = 1 \\ K[0] + K[1] + K[2] + 3 = 0 \end{cases} \tag{15}$$

- otherwise

$$\begin{cases} K[0] + K[1] + 1 = 1 \\ K[0] + K[1] + K[2] + 3 \neq 1 \end{cases} \tag{16}$$

式 (13)-(16) より，次の弱鍵クラス $K[0]$，$K[1]$ の条件を導出できる．

$$K[1] + K[0] = 0 \bmod N \tag{17}$$

これは Roos の弱鍵クラスと同一である[1]．また Roos は文献中の付録においても，いくつかの弱鍵を提案しているが，それらもすべて提案手法で導出できる弱鍵の一つにすぎない。また Wagner の弱鍵も Roos の弱鍵と同様に弱鍵推測アルゴリズムにより導出できることを確認した．すなわち本稿で提案した弱鍵は過去に提案されてきた RC4 の弱鍵を一般化したものであると考えられる．

また提案アルゴリズムを利用することで，これら以外にも多くの弱鍵を発見することが可能である．一例を上げると $K[0] + K[1] = 3$, $K[2] = 249, K[3] = 1$ ならば，他の $K[4], \ldots, K[l]$ の値にかかわらず，その出力は高い確率で $z_2 = 3$，$z_3 = 255$ を出力する．すなわち上記の出力が得られた場合，その鍵の $K[0], \ldots, K[3]$ を高い確率で推測することが可能である．

[1]我々の提案手法では $K[2]$ にも若干の条件がつくものの，この条件を満たさない限りは Roos の弱鍵においても，キーストリームからの鍵推測は成功しない．すなわち，Roos の弱鍵をより正しく定義していると考えられる

```
Round_S[j]:
Input i, j, S
 1: j ← j + S[i]
 2: if S[j] is unknown then
 3:    Swap S[i] and S[j]
 4:    Round_S[i](i, j, S)
 5:    j ← x
 6: else if S[j] is known then
 7:    if S[j] is already assined a value then
 8:       Swap S[i] and S[j]
 9:       if S[S[i] + S[j]] is known then
10:          num ← num +1
11:          Round_S[i](i, j, S)
12:       end if
13:    else if S[j] is not assigned a value then
14:       for each x ∈ {0, ..., N − 1}
            without already set do
15:          S[j] ← x
16:          Swap S[i] and S[j]
17:          if S[S[i] + S[j]] is known then
18:             num ← num +1
19:             Round_S[i](i, j, S)
20:          end if
21:       end for
22:    end if
23: end if
24: return
```

図 5: j の範囲決定アルゴリズム

5 むすび

本稿では RC4 の弱鍵を一般化し，それらを効率よく探索するアルゴリズムを提案した．そして過去に提案された弱鍵は，アルゴリズムから導出できる弱鍵の一つであること，すなわち新たに提案する弱鍵の部分集合であることを示した．また従来は弱鍵では無いとされていた鍵の中にも，まだ多くの弱鍵が存在することを示した．

今後の予定として，更なる探索アルゴリズムの効率化，弱鍵を用いた攻撃手法の詳細な評価，そして WPA，SSL 等の現実に RC4 を用いている環境への弱鍵を利用した攻撃手法の適用が挙げられる．

参考文献

[1] M. Matsui, “Key collisions of the RC4 stream cipher,” Proc. of FSE2009, LNCS, vol.5665, pp.38–50, Springer-Verlag, 2009.

[2] A. Roos, “A class of weak keys in the RC4 stream cipher.” `http://marcel.wanda.ch/Archive/WeakKeys`.

[3] D. Wagner, “My RC4 weak keys.” `http://www.cs.berkeley.edu/~daw/my-posts/my-rc4-weak-keys`.

[4] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the key scheduling algorithm of RC4,” Proc. of SAC2001, LNCS, vol.2259, pp.1–24, Springer-Verlag, 2001.

[5] R. Teramura, Y. Asakura, T. Ohigashi, H. Kuwakado, and M. Morii, “Fast WEP-key recovery attack using only encrypted IP packets,” IEICE Transactions on Fundamentals, vol.E93-A, no.1, 2010 (in press).

[6] I. Mantin and A. Shamir, “A practical attack on broadcast RC4,” Proc. of FSE2001, LNCS, vol.2355, pp.152–164, Springer-Verlag, 2001.