RC4に対する平文回復攻撃の改良

大東 俊博† 渡辺 優平‡ 森井 昌克‡

†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

‡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

あらまし Broadcast Setting の RC4 において，暗号文のみから平文全体を復元できる平文回復攻撃が FSE 2013 で五十部らによって提案された．その攻撃は RC4 の初期の出力バイトの bias と $ABSAB$ bias を用いることで，平文の先頭 1000 テラバイトを 2^{34} 個の暗号文から復元できる．その後，USENIX Security 2013 で AlFardan らによって異なる平文回復攻撃が提案された．AlFardan らの攻撃は五十部らの攻撃とは異なる bias と効果的なカウントアップ手法を用いており，五十部らの攻撃に似た結果を得ている．本稿では五十部らの攻撃と AlFardan の攻撃を適切に組み合わせることで攻撃成功確率を向上させる．提案手法では平文バイトを復元できる確率が概ね 1 になるときの暗号文数を従来の 2^{34} から 2^{33} まで減少させることに成功している．

Improvement on a Full Plaintext Recovery Attack against RC4

Toshihiro Ohigashi† Yuhei Watanabe‡ Masakatu Morii‡

†Information Media Center, Hiroshima University.
1–4–2 Kagamiyama, Higashi-Hiroshima, Hiroshima 739–8511, Japan
‡Graduate School of Engineering, Kobe University
1–1 Rokkodai, Nada, Kobe, Hyogo 657–8501, Japan

Abstract The first full plaintext recovery attack on RC4 in the broadcast setting was proposed at FSE 2013 by Isobe et al. The attack uses the initial byte biases and $ABSAB$ bias of RC4, and can recover 1000T Byte of a plaintext from 2^{34} ciphertexts. After that, at USENIX Security 2014, AlFardan et al. proposed a variant of the plaintext recovery attack with another biases and the sophisticated count-up method. In this paper, we propose a new full plaintext recovery attack by combining Isobe et al.'s attack and the main idea of AlFardan et al.'s attack. Our proposed attack can recover all plaintext bytes from 2^{33} ciphertexts.

1 はじめに

RC4 は 1987 年に Rivest によって提案されたストリーム暗号であり，SSL/TLS [1, 2]，WEP，WPA-TKIP などのセキュリティプロトコルや，Microsoft Lotus，Oracle secure SQL などで広く使われている．RC4 は提案されて以来，様々な解読法 [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] が報告されているが，WEP などの特別な実装への攻撃を除くと，RC4 に対する現実的な攻撃は Broadcast Setting における平文回復攻撃しか知られていない．

Broadcast Setting は同じ平文が異なる複数の秘密鍵で暗号化され，その暗号文を攻撃者が観測できるという攻撃モデルである．正確には完全に同じ平文ではなくとも，その一部（攻撃の対象となる平文バイト）が暗号文毎に変化しなければ攻撃の条件を満たす．これはセッションごとにランダムな鍵を生成する SSL/TLS に合致する攻撃モデルであり，現実的な環境で実行可能な攻撃とみなせる．例えば，Web 上にある秘密のコンテンツを複数のユーザで共有することを考える．そのコンテンツを取得するために各ユーザは個別に SSL/TLS セッションを張り，異なる秘密鍵で同じ平文（コンテンツ）を暗号化して通信する．このとき，攻撃者は通信路上でそれを観測することで攻撃に必要な暗号文を取得できる．

Broadcast Setting の RC4 における最初の平文回復攻撃は FSE 2001 で Mantin らによって提案された [5]．この攻撃は，RC4 から生成される擬似乱数列 (キーストリーム) の 2 バイト目に偏り（bias）があることを利用し，異なる鍵で暗号化された 2^8 個以上の暗号文から平文の 2 バイト目をランダム探索より高確率で復元する．その後，FSE 2011 では Maitra らによって攻撃対象を平文の先頭の 2〜255 バイト目まで拡張できることが示され [7]，FSE 2013 では五十部らによって全ての平文バイトを攻撃対象とできる攻撃法が提案された [15]．五十部らの攻撃では，RC4 のキーストリームの初期のバイトにだけ生じる bias (short-term bias) とキーストリームの任意の位置のバイトで生じる bias (long-term bias) を組み合わせて使っており，その long-term bias は知られている中で最も強力な $ABSAB$ bias [4] を用いている．五十部らの攻撃は，2^{32} 個の暗号文から平文の先頭 257 バイトの各バイトをそれぞれ確率 0.8 以上で復元でき，2^{34} 個の暗号文から平文の先頭 1000 テラバイト全体を確率 0.97 程度で復元できる．

USENIX Security 2013 で AlFardan らは bias の利用方法を改善した平文回復攻撃を提案した [16]．AlFardan らの一つ目の攻撃では short-term bias についてキーストリームの各バイトで複数の bias を有効に扱えるようにする．具体的には，平文復元アルゴリズム中のカウントアップ手法に対してキーストリームの出力の生起確率を使った重み付けを施している．この攻撃は，2^{32} 個の暗号文から平文の先頭 256 バイトの各バイトを確率 0.96 以上で復元できる．また AlFardan らは，FSE 2000 で Fluhrer と McGrew によって発見された long-term bias (FM00 bias [3]) を利用し，short-term bias の場合と似たカウントアップ手法によって平文を復元する攻撃も提案している．ただし，この AlFardan らの二つ目の攻撃は特殊な条件を満たす平文への攻撃であり，五十部らの攻撃とは直接比較することはできない．しかしながら，平文バイトを復元できる確率が概ね 1 になるための暗号文のデータサイズが Broadcast Setting で言えば 2^{34} 個の程度の暗号文に相当するため，五十部らの攻撃と近い能力を有していると思われる．さらに，AlFardan らの二つ目の攻撃に含まれるカウントアップ手法や FM00 bias は五十部らの攻撃で使っている技術とは独立しており，これらを併用すれば平文回復攻撃の成功確率を向上させられると考えられる．

五十部らの攻撃で成功確率が概ね 1 になる暗号文数である 2^{34} という数は，現実の環境で影響が生じるかどうかのボーダーライン周辺にあると考えられる．そのため，従来の手法の組み合わせによってその値がどれだけ減少するかを明らかにすることは，RC4 を継続使用した場合に 258 バイト目以降の平文情報が実際に漏洩するかを考える上での重要な判断材料となる．

そこで，本稿では，まず初めに五十部らの攻撃の $ABSAB$ bias を用いたアルゴリズムを AlFardan らの攻撃のようなカウントアップ手法を利用できるように変形する．次に，FM00 bias にも同様のカウントアップ手法を適用した上で $ABSAB$ bias と併用するように改良した平文回復攻撃を提案する．さらに計算機実験により，Broadcast Setting において提案手法は 2^{33} 個の暗号文から平文バイトを概ね確率 1 で復元でき，五十部らの攻撃の半数まで必要な暗号文数を削減できていることを明らかにする．

2 RC4

RC4 は鍵スケジューリングアルゴリズム (KSA) と擬似乱数生成アルゴリズム (PRGA) の 2 つのアルゴリズムにより構成される．KSA は ℓ バイトの可変長の秘密鍵を利用して $\{0, 1, \ldots N-1\}$ を要素に持つ置換表から成る内部状態 S を初期化する．一般的に $\ell = 16$，$N = 256$ が利用され，本稿でも特に記載がない場合にはそのパラメータを用いる．PRGA は初期化された S から任意長のバイト単位の擬似乱数列 (キーストリーム) $Z_1, Z_2, \ldots Z_r, \ldots$ を生成する．ここで r は PRGA のラウンド数である．Z_r と r バイト目の平文 P_r との排他的論理和 (XOR) をとることで r バイト目の暗号文 C_r が得られる．KSA と PRGA のアルゴリズムを Algorithm 1，2 に示す．ここで $+$ は N を法とした算術加算である．

3 既存の平文回復攻撃

3.1 五十部らの攻撃

FSE 2013 で五十部らによって提案された攻撃 [15] は，short-term bias によって平文の先頭 257 バイトを復元し，long-term bias である $ABSAB$ bias を用いた逐次型のアルゴリズムによって 258 バイト目以降の平文バイトを復元する方法である．

Algorithm 1 KSA

KSA($K[0,\ldots,\ell-1]$)**:**
  for $i = 0$ to $N-1$ **do**
    $S[i] \leftarrow i$
  end for
  $j \leftarrow 0$
  for $i = 0$ to $N-1$ **do**
    $j \leftarrow j + S[i] + K[i \bmod \ell]$
    Swap $S[i]$ and $S[j]$
  end for

Algorithm 2 PRGA

PRGA(K)**:**
  $i \leftarrow 0$
  $j \leftarrow 0$
  $S \leftarrow KSA(K)$
  loop
    $i \leftarrow i+1$
    $j \leftarrow j + S[i]$
    Swap $S[i]$ and $S[j]$
    Output $Z \leftarrow S[S[i] + S[j]]$
  end loop

3.1.1 short-term bias を用いた方法

五十部らは RC4 のキーストリームの先頭の 257 バイトについて，それぞれのバイトで最も強い (最も偏りが大きい) bias を明らかにした．その bias のセットには，条件付き bias である $Z_1 = 0|Z_2 = 0$ [15] および $Z_2 = 0$ [5], $Z_3 = 131$ [15], $Z_r = 0$ $(3 \le r \le 255)$ [7, 17], $Z_r = r$ $(3 \le r \le 255)$ [15], $Z_{16} = 240$ [18], $Z_r = (256 - r)$ $(r = 32, 48, 64, 80, 96, 112)$ [15], $Z_{256} \neq 0$ [15], $Z_{257} = 0$ [15] が含まれる．

これらの bias を利用することで暗号文のみからキーストリームバイトに対応する平文バイトを復元できる．例として $Z_2 = 0$ の bias により平文の 2 バイト目 P_2 を復元する方法を述べる．$Z_2 = 0$ は確率 $2/N$ で生じることがわかっており，ランダムに 1 バイトの値が出力される場合 (生起確率 $1/N$) と比べて高確率で生じる．これは暗号文の 2 バイト目 $C_2 (= P_2 \oplus Z_2)$ が $C_2 = P_2 \oplus 0 = P_2$ を満たす確率が高いことを意味する．したがって，同じ平文バイト P_2 が異なる鍵で暗号化された場合の暗号文バイト C_2 をカウントして頻度表を作成すると，$Z_2 = 0$ に対応する値のカウント数が多くなり，その値から $C_2 = P_2$ として P_2 を復元できる．この攻撃の成功確率はカウント数の多さ (=暗号文数) と bias の強さから決まり，$Z_2 = 0$ の場合は暗号文数が 2^8 より多くなったあたりから成功確率が上昇し始める [5]．平文の先頭 257 バイトの全てのバイトの成功確率が上昇し始めるのは暗号文数が 2^{24} を超えたあたりであり，暗号文数 2^{32} で各平文バイトの復元成功確率は全て 0.8 以上まで上昇する．

3.1.2 long-term bias を用いた方法

short-term bias はキーストリームの初期にしか存在しないため，前節の攻撃ではそれ以後の平文バイトを復元できない．そこで，五十部らの攻撃ではキーストリーム全体で生じる bias である long-term bias を用いて残りの平文バイトを復元する．

五十部らの攻撃では long-term bias として $ABSAB$ bias [4] を用いている．$ABSAB$ bias はキーストリームの 2 バイト単位のシンボル (digraph) の分布の統計的な偏りのことであり，同じ digraph が短いギャップで繰り返されやすいことに注目している．digraph 間のギャップの長さを G としたとき，$ABSAB$ bias は以下の式で表される．

$$Z_r \,||\, Z_{r+1} = Z_{r+2+G} \,||\, Z_{r+3+G} \quad \text{for } G \ge 0, \ (1)$$

ここで $||$ はバイトの連結である．式 (1) の成立確率は以下の定理で与えられる．

定理 1. [4] *For small values of G, the probability of the pattern $ABSAB$ in RC4 keystream, where S is a G-byte string, is* $(1+e^{(-4-8G)/N}/N) \cdot 1/N^2$.

五十部らの攻撃では複数の G での $ABSAB$ bias を利用するが，複数の bias を束ねて利用する際の能力を表現する指標として Mantin によって示された discrimination D を用いる [4]．k 個の独立な bias を生じるイベントが存在するときの D の値に関する補題は以下のように与えられる．

補題 1. [4] *Let X and Y be two distributions and suppose that the independent events $\{e_i: 1 \le i \le k\}$ occur with probabilities* $\Pr_X(e_i) = p_i$ *in X and* $\Pr_Y(e_i) = (1+q_i) \cdot p_i$ *in Y. Then, the discrimination D of the distributions is* $\sum_i p_i \cdot q_i^2$.

さらに，bias が含まれる分布 Y とランダムな分布 X を確率 $(1-\alpha)$ で識別するために必要な暗号文数は以下の補題で与えられる．

補題 2. [4] *The number of samples that is required for distinguishing two distributions that*

have discrimination D with a success rate $1-\alpha$ (for both directions) is $(1/D)\cdot(1-2\alpha)\cdot log_2\frac{1-\alpha}{\alpha}$.

この補題は Broadcast Setting における平文回復攻撃において，正しい平文候補の分布 (bias がある分布) と 1 つの誤った候補の分布 (ランダムな分布) の区別に成功する確率と暗号文数の関係に対応しており，$1/D$ の値が小さいほど同じ成功確率を達成するのに必要な暗号文数を減らせることになる．これは，同じ暗号文数が与えられたとき $1/D$ の値が小さい (D の値が大きい) ほど平文の復元に成功する確率が高くなることを意味する．

五十部らの攻撃では $ABSAB$ bias を利用するために $(C_r \,||\, C_{r+1}) \oplus (C_{r+2+G} \,||\, C_{r+3+G})$ を暗号文から計算して用いる．$ABSAB$ bias に対応するイベントである式 (1) が成立するとき，$(C_r \,||\, C_{r+1}) \oplus (C_{r+2+G} \,||\, C_{r+3+G}) = (P_r \oplus P_{r+2+G} \,||\, P_{r+1} \oplus P_{r+3+G}) = (P_r \,||\, P_{r+1}) \oplus (P_{r+2+G} \,||\, P_{r+3+G})$ の関係が成立し，暗号文の差分の情報から平文の差分の情報を得ることができる．この関係式を用いて未知の平文バイト P_r をその直前の連続した $(G_{MAX}+3)$ バイト分の平文情報 $P_{r-G_{MAX}-3}, \ldots, P_{r-1}$ と暗号文から復元する攻撃関数 $f_{ABSAB\_F}()$ を Algorith 3 に示す．ここで，G_{MAX} は $ABSAB$ bias のためのパラメータであり，攻撃の能力を十分に引き出すことができる値として $G_{MAX} = 63$ が使われる [15]．Step 1〜4 では，$G = 0, 1, \ldots, G_{MAX}$ のそれぞれについて式 (1) に関するカウントアップをして頻度表 $T_{count}[r][G]$ を作成する．その後，異なる G の頻度表を一括して利用するために P_{r-3-G}，P_{r-2-G} の情報を使って $(P_{r-1} \,||\, P_r)$ のためのカウント数に変換した上で頻度表 $T_{merge}[r]$ に集約する．最後に平文の候補数を 2^{16} から 2^8 に減少させるために，P_{r-1} の情報を使って $T_{merge}[r]$ のカウント数を P_r のためのカウント数に変換して頻度表 $T_{guess}[r]$ に格納する．$T_{guess}[r]$ で最もカウント数が多い候補が正しい P_r として推測される．

五十部らの攻撃では，$P_1, P_2, \ldots, P_{257}$ を short-term bias を用いた方法で復元し，その平文情報を基に $f_{ABSAB\_F}()$ を用いて P_{258} から順に 1 バイトずつ逐次的に復元していく．こうすれば $f_{ABSAB\_F}()$ に必要な平文情報はそれ以前の復元処理で得られることになり，正しい平文バイトが復元されている間は攻撃の成功確率は減少しない．すなわち，1 バイト分の復元確率が 1 に達する暗号文数が得られれば，その後の全ての平文バイトを復元できる．五十部らは計算機実験により暗号文数が 2^{34} のときに 1 バイト分の復元に成功する確率が 1 に到達することを確かめた．また，理論的な解析によって平文の先頭の 1000 テラバイトまでは確率 0.97 程度で復元できることを示している．

Algorithm 3 $f_{ABSAB\_F}()$

Require: r, /* round number of a plaintext to be guessed */
G_{MAX}, /* parameter of the $ABSAB$ bias*/
$P_{r-G_{MAX}-3}, \ldots, P_{r-1}$, /* known plaintext bytes */
$(C_{r-G_{MAX}-3}, \ldots, C_r)$s of $C^{(1)}, C^{(2)}, \ldots, C^{(X)}$ /* bytes of X ciphertexts encrypted by different keys */
Ensure: P_r
1: **for** $G = 0$ **to** G_{MAX} **do**
2: Make frequency tables $T_{count}[r][G]$ of $(C_{r-3-G} \,||\, C_{r-2-G}) \oplus (C_{r-1} \,||\, C_r)$ from all ciphertexts $C^{(1)}, C^{(2)}, \ldots, C^{(X)}$, where $(C_{r-3-G} \,||\, C_{r-2-G}) \oplus (C_{r-1} \,||\, C_r) = (P_{r-3-G} \,||\, P_{r-2-G}) \oplus (P_{r-1} \,||\, P_r)$ only if Eq. (1) holds.
3: Convert $T_{count}[r][G]$ into a frequency table $T_{merge}[r]$ of $(P_{r-1} \,||\, P_r)$ by $P_{r-3-G_{MAX}}, \ldots, P_{r-2}$, and merge counter values of all tables.
4: **end for**
5: Make a frequency table $T_{guess}[r]$ indexed by only P_r from $T_{merge}[r]$ with knowledge of P_{r-1}. More precisely, using a pre-guessed value of P_{r-1}, only tables $T_{merge}[r]$ corresponding to the value of P_{r-1} are taken into consideration.
6: Regard the most frequency occurring P_r in table $T_{guess}[r]$ as the correct P_r.
7: Output P_r

3.2 AlFardan らの攻撃

五十部らの攻撃も含めて，従来の平文回復攻撃は，あるキーストリームバイト (またはその差分) に関して最も強い bias を発見し，その一つの bias の影響で頻度表のカウント数が偏る挙動に注目して平文を復元する方法であった．しかしながら，カウントする暗号文バイトに対応する bias が複数存在する場合があり，従来の単純にカウントアップする手法ではその能力を引き出すことができなかった．USENIX Security 2013 において AlFardan らはキーストリームの各バイトの最も強い bias だけを使うのではなく，各バイトに存在する他の比較的弱い bias も含めた複数の bias を利用して最尤推定により平文を求める攻撃を提案した [16].

AlFardan らの一つ目の攻撃は short-term bias を用いた方法を改良したものである．AlFardan らはキーストリームの先頭の 256 バイトについて計算機実験によってキーストリームの各バイトの出力値の生起確率を求めた．$Z_r = k$ となる確率を $p_{r,k}$ と表記する．AlFardan らの short-term bias を用いた平文回復攻撃によって r バ

Algorithm 4 AlFardan et al.'s attack using short-term bias (for r round)

Require: $\{C_{j,r}\}_{1\leq j\leq X}$, /* r-th byte of ciphertexts obtained by X independent encryptions of fixed plaintext P */
$p_{r,k}$ for $0x00 \leq k \leq 0xFF$ /* keystream distribution at position r */
Ensure: P_r^* /* estimate for plaintext byte P_r */
1: $T_{0x00} \leftarrow 0, \ldots, T_{0xFF} \leftarrow 0$
2: **for** $j = 1$ to X **do**
3: $\quad T_{C_{j,r}} \leftarrow T_{C_{j,r}} + 1$
4: **end for**
5: **for** $\mu = 0x00$ to $0xFF$ **do**
6: $\quad$ **for** $k = 0x00$ to $0xFF$ **do**
7: $\quad\quad T_k^{(\mu)} \leftarrow T_{k\oplus\mu}$
8: $\quad$ **end for**
9: $\quad \lambda_\mu \leftarrow \sum_{k=0x00}^{0xFF} T_k^{(\mu)} \log p_{r,k}$
10: **end for**
11: $P_r^* \leftarrow \arg\max_{\mu\in\{0x00,\ldots,0xFF\}} \lambda_\mu$
12: Output P_r^*

イト目の平文を推定するアルゴリズムを Algorithm 4 に示す．アルゴリズムの Step 2〜4 では r バイト目の暗号文の値をカウントして頻度表 T_{0x00}〜T_{0xFF} に格納する．これを各 bias に依存した形で重み付きのカウントをし直して最も確からしい平文候補を得る処理が Step 5〜11 である．平文バイトの候補の変数を μ，キーストリームバイトの変数を k としたとき，Step 6〜9 で平文バイトの候補 μ の尤度 λ_μ を計算していき，Step 11 で全ての候補から最も尤度が大きな候補を正しい値として推定する．Step 6〜9 の処理では平文バイトの候補 μ に対応するようにオリジナルの頻度表 T を頻度表 $T^{(\mu)}$ に並び替えた後，$\log p_{r,k}$ で重み付けをしてカウントし直してから，λ_μ に加算していく．この処理によってキーストリームバイトの全ての bias を推定に利用できる．このアルゴリズムは通常のカウントアップ手法では最も強力な bias に関するものだけを 1 ずつ数えていたのに対し，各 bias について生起確率の log を取った値をカウント値としてマージしたものに変更したものとみなせる．この攻撃によって 2^{32} 個の暗号文から平文の先頭 256 バイトの各バイトを復元できる確率は 0.96 以上まで上昇した．

AlFardan らは二つ目の攻撃として long-term bias を用いた方法も提案している．この方法では long-term bias として FSE 2000 で Fluhrer と McGrew によって発見された bias (FM00 bias) [3] を用いている．FM00 bias はキーストリームの 2 バイト単位 $Z_r||Z_{r+1}$ の値の出現確率の偏りに注目をしたものであり，index i に依存して決まる 12 個のイベントより構成されている．基本的なアイディアは，これら複数の bias を一つ目の攻撃と同じように生起確率を利用した重み付けを備えたカウントアップ手法によって正しい平文を復元するものである．ただし，この攻撃は Broadcast Setting を条件にせず，その代わりに攻撃対象の平文が特殊な条件を満たすことを要求する．具体的には，あるターゲットの平文ブロックを p としたとき，暗号化対象の平文は $P = (p \,||\, p \,||\, \ldots \,||\, p)$ のように p が何度も含まれるような条件を必要とする．なお，平文ブロック p のサイズは N バイトの倍数であり，それぞれの平文ブロックに対応する FM00 bias は index i の値が同じであるため Broadcast Setting のように攻撃に利用できる．また，p のうち攻撃対象のバイト列（実験では連続した 16 バイト) のみ未知であるとし，それ以外のバイトは全て既知である条件も必要とする．文献 [16] の実験結果によると，平文ブロックの連結数が 2^{34} 周辺で未知の平文バイトの復元確率が概ね 1 となっている．

この攻撃は長さ p の平文を異なる鍵で暗号化する Broadcast Setting に容易に変換できる．このとき，攻撃対象のバイトの前後の平文バイトを既知とする攻撃者に有利な条件が必要となるが，2^{34} 個の暗号文があれば平文バイトの復元確率は概ね 1 になると考えられる．

4 提案手法

AlFardan らの二つ目の攻撃は実行可能な条件が異なるとはいえ，五十部らの攻撃と近い攻撃の性能を有しているようにみえる．また，両者の攻撃で使っている技術はそれぞれ独立しているため，五十部らの攻撃に AlFardan らの攻撃のテクニックを応用することで更に強力な平文回復攻撃を実現することができる．本章では，AlFardan らの攻撃の中心となるアイディアである，1) カウントアップ手法への生起確率を利用した重み付けを導入，2) FM00 bias の利用，を五十部らの攻撃に加えることで平文回復攻撃を改良する．

4.1 カウントアップ手法の改善

五十部らの攻撃では $ABSAB$ bias を複数の G で利用し，その結果を集約して頻度表を作っている．定理 1 より G が異なるとき，式 (1) の成立確率は異なっているため，カウントアップ手法に bias の生起確率を利用した重み付けを加

えれば，攻撃の成功確率が向上することが期待できる．

カウントアップ手法を改善するために Algorith 3 の一部を修正する．Algorith 3 ではカウントアップの処理と既知の平文情報 P_{r-1} を用いて頻度表を 2 バイトから 1 バイトへ縮小する処理が含まれている．後者は誤った候補を事前に削除できるメリットがあるため，カウントアップの処理で重み付けをする前に移動する．具体的には Algorith 3 の Step 5 を削除し，Step 3 を以下の処理に変更する．

3: $P_{r-3-G_{MAX}}, \ldots, P_{r-1}$ を用いて $T_{count}[r][G]$ の頻度表を P_r のための頻度表 $T_{count\_1byte}[r]$ に変換する．その後，$T_{count\_1byte}[r]$ の頻度表のカウント値に生起確率を利用した重み付けを施した上で集約用の頻度表 $T_{guess}[r]$ の各候補に加算する．

まず，2 バイトの頻度表 $T_{count}[r][G]$ から 1 バイトの頻度表 $T_{count\_1byte}[r]$ を作る処理を説明する．$T_{count}[r][G]$ に含まれているのは，$(C_{r-3-G} \,||\, C_{r-2-G}) \oplus (C_{r-1} \,||\, C_r)$ の値に対するカウント数である．式 (1) より $ABSAB$ bias が生じる場合には，$Z_{r-3-G} \oplus Z_{r-1} = 0$ が成立する必要があり，そのとき $C_{r-3-G} \oplus C_{r-1} = P_{r-3-G} \oplus P_{r-1}$ が満たされる．すなわち既知の平文バイトである P_{r-3-G} と P_{r-1} の情報を使って，$T_{count}[r][G]$ から $ABSAB$ bias が生じる可能性がある 2^8 個の候補のみ抽出できる．この処理で抽出した $C_{r-2-G} \oplus C_r$ の値に対応したカウント数とラベルに対して，ラベルへ既知の平文バイトである P_{r-2-G} を XOR して並び替えてから頻度表 $T_{count\_1byte}[r]$ に格納する．$T_{count\_1byte}[r]$ は $C_{r-2-G} \oplus C_r \oplus P_{r-2-G}$ $(= Z_{r-2-G} \oplus Z_r \oplus P_r)$ をカウントした数が入る．$Z_{r-2-G} \oplus Z_r$ の生起確率の分布が得られた場合，Algorithm 4 と同じ要領で $T_{count\_1byte}[r]$ の頻度表から P_r を推定するための頻度表を重み付けをした上で作ることができる．

次に $T_{count\_1byte}[r]$ からカウント数に bias の生起確率を利用した重み付けをした上で $T_{guess}[r]$ にカウントしていく (複数の G のカウント値をマージする) 処理を説明する．$T_{count\_1byte}[r]$ は $Z_{r-3-G} \oplus Z_{r-1} = 0$ が成立する暗号文バイトのみ格納しているため，その中で $ABSAB$ bias は条件付 bias として $(Z_{r-2-G} \oplus Z_r) = 0 \,|\, (Z_{r-3-G} \oplus Z_{r-1}) = 0$ のように表現される．RC4 はキーストリームの 1 バイト同士の差分だけでは bias は生じずにランダムとなると考えられるため，この条件付確率の式の成立確率は定理 1 の確率の N 倍となり $(1 + e^{(-4-8G)/N}/N) \cdot 1/N$ となる．これは $T_{count\_1byte}[r]$ でカウントしている暗号文バイトでは $Z_{r-2-G} \oplus Z_r = 0$ は確率 $(1 + e^{(-4-8G)/N}/N) \cdot 1/N$ で生じることを意味する．また $Z_{r-2-G} \oplus Z_r$ が 0 以外になる場合のそれぞれの生起確率はランダムとみなして $1/N$ とする．このとき，Algorithm 4 と同じ要領（生起確率の log を取った値をカウント数に乗じる方法）で $Z_{r-2-G} \oplus Z_r$ の生起確率の分布を使って $T_{count\_1byte}[r]$ の頻度表からカウント数に重み付けをした上で P_r を推定するための頻度表である $T_{guess}[r]$ に加算する．

最後に $G = 0, 1, \ldots, G_{MAX}$ の全てについて同様に重み付けをした上で加算をした $T_{guess}[r]$ から，最も数値が大きい値を正しい P_r として推定する．

4.2 FM00 bias との併用

次に $ABSAB$ bias とは異なる long-term bias である FM00 bias を利用する．FM00 bias はキーストリームの 2 バイト単位 $Z_r||Z_{r+1}$ の bias であるが，そのイベントの個数やパターンは表 1 のように index i ごとに異なっている．

FM00 bias も $ABSAB$ bias と同様に条件付 bias として利用する．P_r が既知のときに未知の平文バイト P_{r+1} を復元することを考える (実際に $ABSAB$ bias と併用する場合にはターゲットの平文バイトの位置を合わせてから使う必要があることに注意する)．まず初めに暗号文から $C_r||C_{r+1}$ をカウントアップして 2 バイト単位の頻度表 $T'_{count}[r+1]$ を作成する．次に表 1 のイベントの Z_r のそれぞれについて，既知の平文バイトである P_r を使って $T'_{count}[r+1]$ からそのイベントに関係がある候補のみ抽出する．例えば，$i = 1$ のとき $Z_r = 0$ のイベントに関係するのは，$C_r||C_{r+1}$ の中で $C_r = Z_r \oplus P_r = 0 \oplus P_r = P_r$ を満たすラベルのカウント数だけであり，その情報から $(Z_r, Z_{r+1}) = (0, 0)$ または $(Z_r, Z_{r+1}) = (0, 2)$ に関係する 1 バイト単位の頻度表 $T'_{count\_1byte}[r+1]$ を作ることができる．さらに，カウント数の重み付けを行う際には $Z_{r+1} = 0$ の生起確率は $N^{-1}(1 + 2 \cdot N^{-1})$，$Z_{r+1} = 2$ の生起確率は $N^{-1}(1 - N^{-1})$，それ以外は $1/N$ のように分布を作成してから計算する．重み付けをされたカウント数はイベントの Z_r の値ごとに計算され，集約用の頻度表 $T'_{guess}[r+1]$ に加算される．表 1 の i に対応する全てのイベ

表 1: i ($= r \bmod N$) に関する FM00 bias

Index i	Z_r	Z_{r+1}	$\Pr(Z_{r+1}\|Z_r)$	D
0	0	0	$N^{-1}(1+N^{-1})$	$5/N^3$
	1	$N-1$	$N^{-1}(1+N^{-1})$	
	$N-1$	1	$N^{-1}(1+N^{-1})$	
	$N-1$	2	$N^{-1}(1+N^{-1})$	
	$N-1$	$N-1$	$N^{-1}(1-N^{-1})$	
1	0	0	$N^{-1}(1+2\cdot N^{-1})$	$9/N^3$
	2	$N-1$	$N^{-1}(1+N^{-1})$	
	$N-1$	3	$N^{-1}(1+N^{-1})$	
	$N-1$	2	$N^{-1}(1+N^{-1})$	
	$N-1$	$N-1$	$N^{-1}(1-N^{-1})$	
	0	2	$N^{-1}(1-N^{-1})$	
2	0	0	$N^{-1}(1+N^{-1})$	$8/N^3$
	0	1	$N^{-1}(1+N^{-1})$	
	3	$N-1$	$N^{-1}(1+N^{-1})$	
	$N-1$	3	$N^{-1}(1+N^{-1})$	
	$N-1$	4	$N^{-1}(1+N^{-1})$	
	$N/2+1$	$N/2+1$	$N^{-1}(1+N^{-1})$	
	$N-1$	$N-1$	$N^{-1}(1-N^{-1})$	
	0	3	$N^{-1}(1-N^{-1})$	
$3,4,\ldots,N-4$	0	0	$N^{-1}(1+N^{-1})$	$7/N^3$
	0	1	$N^{-1}(1+N^{-1})$	
	$i+1$	$N-1$	$N^{-1}(1+N^{-1})$	
	$N-1$	$i+1$	$N^{-1}(1+N^{-1})$	
	$N-1$	$i+2$	$N^{-1}(1+N^{-1})$	
	$N-1$	$N-1$	$N^{-1}(1-N^{-1})$	
	0	$i+1$	$N^{-1}(1-N^{-1})$	
$N-3$	0	0	$N^{-1}(1+N^{-1})$	$6/N^3$
	0	1	$N^{-1}(1+N^{-1})$	
	$N-2$	$N-1$	$N^{-1}(1+N^{-1})$	
	$N-1$	$N-2$	$N^{-1}(1+N^{-1})$	
	$N-1$	$N-1$	$N^{-1}(1-N^{-1})$	
	0	$N-2$	$N^{-1}(1-N^{-1})$	
$N-2$	0	0	$N^{-1}(1+N^{-1})$	$4/N^3$
	0	1	$N^{-1}(1+N^{-1})$	
	$N-1$	0	$N^{-1}(1+N^{-1})$	
	0	$N-1$	$N^{-1}(1-N^{-1})$	
$N-1$	0	1	$N^{-1}(1+N^{-1})$	$5/N^3$
	0	$N-1$	$N^{-1}(1+N^{-1})$	
	$N-1$	0	$N^{-1}(1+N^{-1})$	
	$N-1$	1	$N^{-1}(1+N^{-1})$	
	$N-1$	$N-1$	$N^{-1}(1-N^{-1})$	

ントについて集約用の頻度表 $T'_{guess}[r+1]$ に加算したあと，その頻度表と $ASBAB$ bias の (同じ平文バイトに対応する) 集約用の頻度表の値を合計して最終的な平文バイトの推測用の頻度表を作成する．

なお，平文回復攻撃の成功確率は利用している bias から計算できる D が大きいほど高くなるが，FM00 bias では i の値によってその大きさが異なる．表 1 の D の値より，$i=1$ のときが成功確率が最も高く，$i=N-2=254$ のときに成功確率が最も低くなることが予想される．

4.3 計算機実験

本稿の改良方法の有効性を確認するため計算機を用いた解読実験を行った．実験では逐次型の処理の改善を確認するため，$P_1,\ldots,P_{r-1}$ が既知のときに未知バイト P_r を復元できる確率を 256 回の試行から得ている．実験で比較するのは，五十部らの攻撃，4.1 節の攻撃，提案手法 ($i=1$ の場合)，提案手法 ($i=127$ の場合)，提案手法 ($i=254$ の場合) の 5 つの手法であり，攻撃者が得ることができる暗号文数を 2^{26} から 2^{34} まで変化させてターゲットの平文バイトの復元成功確率を得ている．ここで提案手法は 4.2 節に記載している $ABSAB$ bias と FM00 bias を併用した攻撃手法のことを指し，括弧書きの中は FM00 bias のどのパラメータで攻撃をしているかを示すものである．

攻撃のターゲットとなる平文バイトは，五十部らの攻撃，4.1 節の攻撃，および提案手法 ($i=127$ の場合) は P_{3200}，提案手法 ($i=1$ の場合) は P_{3330}，提案手法 ($i=254$ の場合) は P_{3327} とした．文献 [6] によると RC4 のキーストリームの short-term bias を取り除くために先頭の 3072 バイトを捨てることが推奨されている．そこで，long-term bias のみから構成される逐次型の処理の改善を正確に知るために，short-term bias の影響が無い上記の平文バイトを選んだ．

実験の結果を表 2 に示す．五十部らの攻撃と 4.1 節の攻撃の結果から，$ABSAB$ bias に対してもカウントアップ手法を改良することで成功確率を向上させられることがわかった．$ABSAB$ bias と FM00 bias を併用した提案手法ではどのパラメータでも暗号文数 2^{33} で確率が 1 に到達しており，五十部らの攻撃の半数程度の暗号文数で同程度の成功確率の攻撃が実現できる．また，五十部らの攻撃では効果があまり無い暗号文数 2^{27} の条件でも，提案手法では 1 バイト分の平文を復元できる確率がランダム探索と比べて 10 倍以上になっており，効果が現われる暗号文数の下限も減少している．さらに，提案手法 ($i=1$ の場合)，提案手法 ($i=127$ の場合)，提案手法 ($i=254$ の場合) の 3 つの結果と表 1 の D の値から，D の大きいパラメータでは攻撃の成功確率が向上していることも確認できた．

5 まとめ

本稿では，五十部らの攻撃に AlFardan らの攻撃のアイディアを応用することで，Broadcast Setting における RC4 に対する平文回復攻撃を改良した．まず初めに五十部らの攻撃における $ABSAB$ bias の処理に生起確率を使った重み付けを備えたカウントアップ手法を導入し，その

表 2: 計算機実験の結果

	暗号文数								
	2^{26}	2^{27}	2^{28}	2^{29}	2^{30}	2^{31}	2^{32}	2^{33}	2^{34}
五十部らの攻撃	0.0156	0.0156	0.0430	0.0859	0.1719	0.3789	0.7578	0.9688	1.0000
4.1 節の攻撃	0.0273	0.0391	0.0898	0.1406	0.2578	0.5469	0.8828	0.9883	1.0000
提案手法 ($i = 1$ の場合)	0.0117	0.0508	0.0703	0.1563	0.3828	0.7695	0.9883	1.0000	1.0000
提案手法 ($i = 127$ の場合)	0.0117	0.0508	0.0938	0.1875	0.3398	0.7109	0.9492	1.0000	1.0000
提案手法 ($i = 254$ の場合)	0.0117	0.0352	0.0664	0.1094	0.3125	0.5820	0.9453	1.0000	1.0000

成功確率を向上させた．さらに，異なる long-term bias である FM00 bias を使ったカウントアップの結果と適切にマージすることで，更に成功確率を向上させることができた．FM00 bias の影響で復元する平文バイトの位置によって解読性能は変わるものの，提案手法を用いた場合に 2^{33} 個の暗号文があれば平文バイトを概ね確率 1 で復元できることがわかった．平文回復攻撃の実行時間は暗号文を収集する時間が支配的となるため，全ての平文バイトを復元するための攻撃実行時間は半減したと考えられる．

文献 [16] では，攻撃に必要な暗号文を高速に取得するために不正な JavaScript をユーザに実行させる方法が示されている．この JavaScript には正規のサイトに繰り返し HTTPS でアクセスする命令が書かれてあり，攻撃者はそれを観測することで多数の暗号文を高速に得ることができる．この方法により 1 時間あたり 2^{21} 個の暗号文が攻撃者が取得できる．この方法で 1 台の計算機から 2^{33} 個の暗号文を得る場合には 4096 時間を要することから現時点では必ずしも過度に注意する必要はないが，そのような JavaScript を実行するユーザ数が多くなれば実行時間は短縮されることも考慮すると，今後も平文回復攻撃の発展について注視する必要がある．

謝辞

本研究の一部は JSPS 科研費 基盤研究 (C)(課題番号 26330155) および若手研究 (B)(課題番号 25730085) の助成を受けたものである．本研究を遂行するにあたり，御助言，御討論頂いたソニー株式会社 五十部孝典氏に深謝する．

参考文献

[1] Freier, A. O., Karlton, P. and Kocher, P. C.: The SSL Protocol Version 3.0 (1996). `http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt`.

[2] Dierks, T. and Allen, C.: The TLS Protocol Version 1.0 (1999). `http://www.ietf.org/rfc/rfc2246.txt`.

[3] Fluhrer, S. R. and McGrew, D. A.: Statistical Analysis of the Alleged RC4 Keystream Generator, *FSE*, pp. 19–30 (2000).

[4] Mantin, I.: Predicting and Distinguishing Attacks on RC4 Keystream Generator, *EUROCRYPT*, pp. 491–506 (2005).

[5] Mantin, I. and Shamir, A.: A Practical Attack on Broadcast RC4, *FSE*, pp. 152–164 (2001).

[6] Mironov, I.: (Not So) Random Shuffles of RC4, *CRYPTO*, pp. 304–319 (2002).

[7] Maitra, S., Paul, G. and Sen Gupta, S.: Attack on Broadcast RC4 Revisited, *FSE*, pp. 199–217 (2011).

[8] Knudsen, L. R., Meier, W., Preneel, B., Rijmen, V. and Verdoolaege, S.: Analysis Methods for (Alleged) RC4, *ASIACRYPT*, pp. 327–341 (1998).

[9] Maximov, A. and Khovratovich, D.: New State Recovery Attack on RC4, *CRYPTO*, pp. 297–316 (2008).

[10] Roos, A.: A class of weak keys in the RC4 stream cipher (1995). Two posts in sci.crypt.

[11] Nagao, A., Ohigashi, T., Isobe, T. and Morii, M.: Expanding Weak-key Space of RC4, *JIP*, Vol. 22, No. 2, pp. 357–365 (2014).

[12] Matsui, M.: Key Collisions of the RC4 Stream Cipher, *FSE*, pp. 38–50 (2009).

[13] Paul, G. and Maitra, S.: Permutation After RC4 Key Scheduling Reveals the Secret Key, *Selected Areas in Cryptography*, pp. 360–377 (2007).

[14] Biham, E. and Carmeli, Y.: Efficient Reconstruction of RC4 Keys from Internal States, *FSE*, pp. 270–288 (2008).

[15] Isobe, T., Ohigashi, T., Watanabe, Y. and Morii, M.: Full Plaintext Recovery Attack on Broadcast RC4, *FSE* (Moriai, S., ed.), Lecture Notes in Computer Science, Vol. 8424, Springer, pp. 179–202 (2013).

[16] AlFardan, N. J., Bernstein, D. J., Paterson, K. G., Poettering, B. and Schuldt, J. C. N.: On the Security of RC4 in TLS, *USENIX Security* (King, S. T., ed.), USENIX Association, pp. 305–320 (2013).

[17] Sen Gupta, S., Maitra, S., Paul, G. and Sarkar, S.: (Non-)Random Sequences from (Non-)Random Permutations - Analysis of RC4 Stream Cipher, *J. Cryptology*, Vol. 27, No. 1, pp. 67–108 (2014).

[18] Sen Gupta, S., Maitra, S., Paul, G. and Sarkar, S.: Proof of Empirical RC4 Biases and New Key Correlations, *Selected Areas in Cryptography*, pp. 151–168 (2011).