

WEPの安全な運用方法とその実装について

Secure WEP operation and its implementation

渡辺 優平 † 入山 敬大 † 森井 昌克 †
Yuhei Watanabe Takahiro Iriyama Masakatu Morii

1 はじめに

スマートフォンなどの携帯端末の発展や普及にともない、無線LANの利用が急速に拡大している。無線LANは電波を用いて通信を行うため、常に盗聴の危険性にさらされている。安全に通信を行うために情報の暗号化を行い、情報の漏洩を防ぐ必要がある。無線LANの暗号化方式の一つにWired Equivalent Privacy(WEP)[1]がある。WEPはストリーム暗号であるRC4をもとに開発された共通鍵暗号方式である。WEPは深刻な脆弱性が指摘されており、それを利用した攻撃が多数提案されている。

2001年にS. Fluhrer, I. Mantin, A. ShamirらによってFMS攻撃が提案された[2]。FMS攻撃はweak IVと呼ばれる特定のIVを用いてWEP鍵の導出を行う。FMS攻撃を拡張した攻撃が2004年にKorekによって提案された[3, 4]。Korek攻撃ではFMS攻撃より多くのIVをweak IVとして利用することができる。これらの特定のIVに依存する攻撃はフィルタリングによってweak IVを取り除くことで対策が可能である。一方で特定のIVに依存しない攻撃が提案されている。2006年にA. KleinによってKlein攻撃が提案された[5]。これはIVとキーストリームを用いてWEP鍵を先頭から逐次的に導出する攻撃である。2008年にE. Tews, R. Weinmann, A. PyshkinらによってPTW攻撃が提案された[6]。この攻撃はKlein攻撃を改良した攻撃であり、RC4の内部状態の近似を用いることでWEP鍵の和を並列して導出することが可能である。複数の攻撃関数を用いる攻撃として2010年に寺村, 朝倉, 大東, 桑門, 森井らによりTeAM-OK攻撃が提案された[7]。この攻撃はKlein攻撃, PTW攻撃, OKM攻撃[8]の三つの攻撃関数を用いてWEP鍵の導出を行う。2013年にP. Sepehrdad, P. Suisil, S. Vaudenay, M. VuagnouxらによってTornado攻撃が提案された[9]。この攻撃は改良したKlein攻撃やKorek攻撃などから導出した22個の攻撃関数を利用して投票を行い、WEP鍵を推定する。この攻撃は22,500パケットの盗聴により確率0.5でWEP鍵の導出が可能である。

以上のようにWEPに対する様々な鍵回復攻撃が提案されている。そのためより安全性の高いWi-Fi Protected Access2(WPA2)[10]などの暗号化方式への移行が推奨されている。しかしこれらの方式への移行は様々なコストを要することから、大規模な施設などでは現在でもWEPが利用されている。WEPに対する従来の鍵回復攻撃を防ぐ運用方法として、一つのWEP鍵を使用するパケット数を制限し、一定間隔ごとにWEP鍵を更新する方法がある。従来の鍵回復攻撃は10,000パケットの盗聴において成功確率は0である。そのため10,000パケットの通信を行うごとにWEP鍵を更新する方法が一般的である。しか

し10,000パケットは非常に少数であり、頻繁な鍵の更新は通信のスループットに多大な影響を与える。

WEPに対する鍵回復攻撃の対策として2011年に塚畝, 藤堂, 森井らによりStrong IVが提案された[11]。Strong IVはKlein攻撃が成立しないIVの集合を指す。このIVのみを利用することで100,000パケットの盗聴において鍵回復攻撃を困難にし、WEP鍵の更新間隔を拡大することが可能となる。しかし塚畝らのStrong IVは生成確率が低く、100,000個のIVを生成するために多大な時間を要してしまう。そのため本来の目的であるスループットの向上を達成することが困難であった。2014年に入山, 渡辺, 森井らにより生成確率を向上したStrong IVが提案された[12]。入山らはStrong IVの定義を修正し、従来よりも多くのIVをStrong IVとして利用できることを示した。Strong IVの生成確率の向上により100,000個のStrong IVを高速に生成可能となる。さらに入山らはStrong IVのみを利用した場合のKlein攻撃の成功確率について評価を行い、Strong IVとその他のIVとを混合して用いることで100,000パケットの通信においてWEPを安全に運用できることを示した。

本稿ではWEPのより安全な運用方法を提案する。提案方式を用いた場合にWEPの暗号化に要する時間の評価を行う。提案方式では入山らのStrong IVに加えてTornado攻撃におけるKorek攻撃, キーストリームのバイアス(SVV_10[13])による攻撃関数に対応したStrong IVを用いる。Tornado攻撃では投票条件が全て成立している場合に攻撃関数によるWEP鍵の候補値に対する投票が行われる。そこで任意のIVとWEP鍵を用いた場合に投票条件が成立しているかどうかを効率的に判定する方法を提案する。100,000パケットごとに鍵を更新することを想定したWEPの運用方法としてKlein攻撃, Korek攻撃, SVV_10のバイアスによる攻撃関数を全て無効にするStrong IVを50,000個, Klein攻撃を除いた攻撃関数のみを無効にするIVを50,000個混合して利用する。このときStrong IVの判定と並行して暗号化を行うことで通信への負荷を軽減する。Strong IVの判定と暗号化を並行して行った場合に50,000パケットの暗号化に要する時間の評価を行う。Strong IVの判定と同時に1472バイトの平文を50,000個暗号化するのに要した時間は700ミリ秒である。したがって一回の暗号化あたりに要する時間は14マイクロ秒となる。提案手法によりWEPにおける従来の鍵回復手法を困難にし、高速な通信が可能となる。

2 WEP

電波を用いて通信を行う無線LANは常に盗聴の危険性にさらされている。安全に通信を行うために共通鍵暗号を用いた暗号化方式が利用されている。無線LAN暗号化方式の一つであるWEPはRC4をベースにした共通鍵暗号方式である。本章ではRC4とWEPの概要について述

† 神戸大学大学院工学研究科, Graduate School of Engineering, Kobe University

Algorithm 1 KSA

```

KSA( $K[0, \dots, \ell - 1]$ ):
  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( $K$ ):
   $i \leftarrow 0$ 
   $j \leftarrow 0$ 
   $S' \leftarrow \text{KSA}(K)$ 
  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

```

べる。

2.1 RC4

RC4 は 1987 年に Ronald Rivest により開発されたストリーム暗号である。RC4 はソフトウェアによる高速処理性能に優れているため、商用アプリケーションや SSL/TLS、無線 LAN 用のプロトコルである WEP などで広く利用されている。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 バイトの秘密鍵 K を利用して $\{0, 1, \dots, N - 1\}$ の数列からなる内部状態 S を初期化する。一般的に $\ell = 16$, $N = 256$ が利用される。PRGA は初期化された内部状態 S からキーストリーム $Z_1, Z_2, \dots, Z_r, \dots$ 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 との排他的論理和をとることで暗号文 C_r が得られる。KSA と PRGA のアルゴリズムを Algorithm 1, 2 に示す。アルゴリズム中の $K[x]$ は秘密鍵の x バイト目を、 $S[x]$, $S'[x]$ はそれぞれ KSA と PRGA における内部状態の x バイト目を表す。ここで $+$ は N を法とした算術加算である。

2.2 WEP

WEP は無線 LAN の暗号化方式の一つであり、RC4 をベースにした共通鍵暗号方式である。RC4 の要素数 N は 256 とする。WEP において秘密鍵 K は 24 ビットの初期化ベクトル (IV) と 40 ビットあるいは 104 ビットの WEP 鍵の連結により構成される。本稿では 104 ビットの WEP 鍵を想定して議論する。WEP による暗号化通信において、IV の値はパケットごとに変更される。パケットごとに異なる秘密鍵を用いることで常に同じキーストリームが生成されるのを防ぐためである。IV は通信を行う二者間で共有する必要があるため、暗号文とともに送信される。このとき IV は暗号化されておらず、WEP 鍵を共有していない第三者にも既知となる。IV の長さは 24 ビッ

トであり十分に大きい領域を確保しているとはいえない。以上のような IV の性質が WEP の脆弱性の要因となっている。

3 WEP に対する鍵回復攻撃

WEP に対する鍵回復攻撃は IV の値に依存する攻撃と依存しない攻撃がそれぞれ提案されている。IV に依存しない攻撃として Klein 攻撃 [5], PTW 攻撃 [6], TeAM-OK 攻撃 [7], Tornado 攻撃 [9] が挙げられる。本章ではこれらの鍵回復攻撃について述べる。

3.1 Klein 攻撃

2006 年に Klein は特定の IV に依存しない攻撃を提案した [5]。Klein 攻撃は RC4 の内部状態において特定の遷移が起こりやすいことを利用して内部状態の遷移を予測し、その内部状態とキーストリームとの関係から WEP 鍵の推測を行う。内部状態の予測に推測する WEP 鍵の一つ前のバイトの情報をを用いるため、WEP 鍵を先頭から逐次的に復元する攻撃である。Klein 攻撃の関係式を次式 (1) で表す。ここでは y バイト目の WEP 鍵 $K[y]$ を導出する場合を想定する。

$$K[y] = f_{Klein}(K[0], K[1], \dots, K[y-1], Z_y) = S_y^{-1}[y - Z_y] - j_y - S_y[y] \quad (1)$$

式 1 を用いて IV、キーストリームおよび $y - 1$ バイトまでの鍵の値から WEP 鍵の候補値を導出する。Klein 攻撃が成立する内部状態の遷移は二つの条件から構成される。一つ目の条件は KSA の y ラウンドにおいてスワップされた要素 $S[j]$ が PRGA の $y - 1$ ラウンド目までスワップされないことである。条件式は以下のように表される。

$$S_{y+1}[y] = S'_{y-1}[y]$$

これを条件 1 とする。二つ目の条件は $S'_{y-1}[y]$ の値がキーストリームにより一意に決定することである。条件式は以下のように表される。

$$S'_{y-1}[y] = y - Z_y$$

これを条件 2 とする。条件 1, 2 が同時に成立した場合に Klein 攻撃は成功する。Klein 攻撃の成功条件を図 1 に示す。多数のパケットに対して式 (1) から鍵の候補値の推測と投票を行い、最も投票数の多かった候補値を WEP 鍵の 1 バイトとする。Klein 攻撃は WEP 鍵を先頭バイトから逐次的に導出していく。あるバイトで WEP 鍵の推定を誤った場合、それ以降の WEP 鍵の推測にも影響する。その結果、鍵の導出に必要な計算量が大幅に増加する可能性がある。

3.2 PTW 攻撃

2008 年に E. Tews, R. Weinmann, A. Pyshkin らによって PTW 攻撃が提案された [6]。この攻撃は Klein 攻撃と同様に IV に依存しない攻撃である。PTW 攻撃は IV 以降の秘密鍵により遷移する内部状態について近似を行い、IV とキーストリームのみから WEP 鍵の導出を行う。近似を用いることで WEP 鍵の和を並列に求めることができる。そのため一つの WEP 鍵バイトを誤って推測した場合でも他のバイトの推測に影響を与えない。PTW 攻撃の

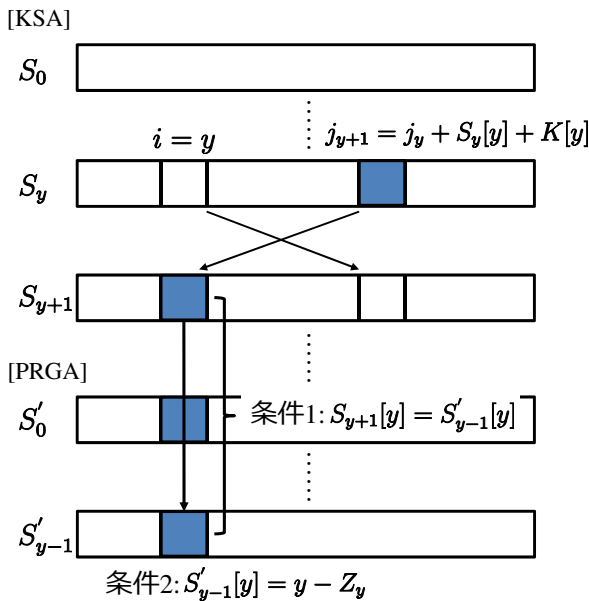


図1 Klein 攻撃の成功条件

成立に必要な条件は Klein 攻撃と同様である。PTW 攻撃の関係式は以下のように表される。

$$\begin{aligned}\sigma_y &= f_{PTW}(K[0], K[1], K[2], Z_y) \\ &= S_3^{-1}[y - Z_y] - j_3 - \sum_{l=3}^y S_3[l]\end{aligned}$$

ここで σ_y は WEP 鍵の先頭バイトから $K[y]$ までの和を表す。PTW 攻撃は近似を用いているため、キーストリームの先頭バイトから順に攻撃成功確率は低下していく。

3.3 TeAM-OK 攻撃

2010年に寺村, 朝倉, 大東, 桑門, 森井らにより TeAM-OK 攻撃が提案された [7]。この攻撃は Klein 攻撃, PTW 攻撃, OKM 攻撃 [8] の三つの攻撃関数を利用する。OKM 攻撃の関係式を以下に示す。

$$K[y - 16] = f_{OKM}(K[0], K[1], \dots, K[y - 17], \sigma_{15}, Z_y)$$

OKM 攻撃ではキーストリーム $Z_{19}, Z_{20}, \dots, Z_{31}$ と WEP 鍵の総和 σ_{15} を利用して鍵の復元を行う。 σ_{15} の導出に PTW 攻撃の関係式を用いる。OKM 攻撃は WEP 鍵の総和が既知という条件のもとで、キーストリーム $Z_{19}, Z_{20}, \dots, Z_{31}$ から高い確率で WEP 鍵を導出することが可能である。OKM 攻撃は Klein 攻撃の成功条件が成立していた場合に成功する。TeAM-OK 攻撃は Klein 攻撃, PTW 攻撃, OKM 攻撃をキーストリームの位置によって使い分け、鍵の候補値への投票を行い、WEP 鍵を導出する。

3.4 Tornado 攻撃

2013年に P. Seperdad, P. Susil, S. Vaudenay, M. Vuagnoux らによって Tornado 攻撃が提案された [9]。この攻撃では 22 個の攻撃関数を用いて投票を行い、WEP 鍵を推測する。利用する攻撃関数は Klein 攻撃を改良したものを、MP 攻撃 [14] を改良したものを、Korek 攻撃のバイアス

を改良したものを、キーストリームの 16 バイト目のバイアス (SVV_10[13]) を利用したものである。それぞれの攻撃関数に対する投票条件が成立した場合に投票を行う。改良した Klein 攻撃の攻撃成功条件は従来の Klein 攻撃と同様である。Tornado 攻撃は 22,500 パケットの盗聴により確率 0.5 で WEP 鍵の復元に成功する。

4 Strong IV

WEP に対する様々な鍵回復攻撃が提案されている一方で、攻撃を困難にし、安全に WEP を利用するための議論が行われている。本章では鍵回復攻撃への対策と、その一つである Strong IV について述べる。

4.1 鍵回復攻撃を困難にする WEP の運用

FMS 攻撃などの IV に依存する攻撃に対しては weak IV をフィルタリングする手法が考えられている。フィルタリングにより weak IV を取り除き、WEP の通信に利用しないことで攻撃を困難にすることができる。Klein 攻撃などの IV に依存しない攻撃に対しては一つの WEP 鍵を使用するパケット数を制限し、定期的に WEP 鍵を更新する方法が提案されている。現在のところ 20,000 パケット以下の盗聴によって鍵回復を行う方法は提案されていない。また 10,000 パケットの盗聴において既存の鍵回復攻撃の成功確率は 0 である。そのため 10,000 パケットの通信を行うごとに WEP 鍵を更新する方法が一般的である。しかし 10,000 パケットは非常に少数であり、通信のスループットに多大な影響を与える。

4.2 Strong IV

鍵の更新間隔を延長しても WEP を安全に利用するための方法として、2011年に塚畝, 藤堂, 森井らによって Strong IV が提案された [11]。Strong IV は Klein 攻撃が成功しない IV の集合を指す。Strong IV を用いると Klein 攻撃の成功条件が成立しないため、鍵回復が困難になる。同様の攻撃成功条件を持つ PTW 攻撃, OKM 攻撃, Tornado 攻撃における改良 Klein 攻撃に対しても有効である。Klein 攻撃が成功するためには条件 1, 2 の両方が成立している必要がある。塚畝らは Klein 攻撃の条件 1 が成立しない IV を Strong IV と定義している。Strong IV は利用する WEP 鍵の値によって異なるため、RC4 を動作させて判定を行う。このとき WEP 鍵 13 バイトそれぞれについて条件 1 の成否を確認する必要がある。ランダムに選択した IV が 1 バイトの WEP 鍵を保護する Strong IV となる確率は以下のように表される。

$$Prob_{StrongIV_1} = 1 - \left(\frac{255}{256}\right)^{254} \approx 0.63$$

WEP 鍵 13 バイト全てを保護する Strong IV となる確率は

$$Prob_{StrongIV_{13}} = \left(1 - \left(\frac{255}{256}\right)^{254}\right)^{13} \approx 2.46 \times 10^{-3}$$

となる。IV の総数が 2^{24} であるため、WEP 鍵 13 バイトを保護する Strong IV の生成確率は非常に低い。そのため塚畝らは WEP 鍵 13 バイト中 12 バイト以上を保護する Strong IV を用いることで生成確率を向上している。この Strong IV を利用した場合、100,000 パケットの盗聴においても鍵回復攻撃を困難にすることが可能である。しか

しながらこの Strong IV を 100,000 個生成するためには多大な時間を要してしまう。そのため本来の目的であるスループットの向上を達成することが困難であり、実用的であるとはいえない。

4.3 定義を修正した Strong IV

塚畝らの提案した Strong IV は生成確率が低く、生成に時間を要してしまうため実用的でないという問題点がある。2014 年に入山、渡辺、森井らにより Strong IV の定義を修正し、生成確率を向上した Strong IV が提案された [12]。入山らは Klein 攻撃の攻撃成功条件 1 だけでなく条件 2 が成立しない IV も Strong IV として利用することで Strong IV の生成確率を向上させている。定義を修正した場合に WEP 鍵 13 バイトを保護する Strong IV の生成確率を以下に示す。

$$Prob'_{StrongIV_{13}} = \left(1 - \left(\frac{255}{256} \right)^{254} \cdot \frac{2}{256} \right)^{13} \approx 0.96$$

この生成確率は従来の Strong IV と比較して極めて大きく、短時間での Strong IV 生成が可能となる。

定義を修正した Strong IV を用いた際の Klein 攻撃の成功確率を以下に示す。

$$Prob'_{Klein} = \left(1 - \left(\frac{255}{256} \right)^{254} \right) \cdot \frac{254}{256 \cdot 255} \approx \frac{0.63}{256} \quad (2)$$

式 (2) より攻撃成功確率がランダム (1/256) より低くなっていることがわかる。このため WEP 鍵の推測方法を変更することで攻撃が成立する可能性がある。この問題への対策として入山らは Strong IV とその他の IV を混合して利用することを提案している。Strong IV とその他の IV とを併用することで IV の集合全体として Klein 攻撃の成功確率を 1/256 にすることで Klein 攻撃の成功条件を持つ攻撃を困難にする。入山らの手法では Strong IV 50,000 個とその他の IV 50,000 個を混合して 100,000 個の IV の集合として利用する。この IV の集合を利用することで一つの WEP 鍵に対して 100,000 パケットの通信を安全に行うことができる。

5 Strong IV を用いた WEP の運用とその実装

本章では入山らの Strong IV に Tornado 攻撃の対策を加えた Strong IV を提案する。さらに既存の鍵回復攻撃全てを無効にする Strong IV を適用した WEP の運用方法を示す。

5.1 Tornado 攻撃に対応した Strong IV

3.4 節で述べた通り、Tornado 攻撃は Klein 攻撃を改良した攻撃など 22 個の攻撃関数から構成されている。改良 Klein 攻撃は定義を修正した Strong IV により困難になる。さらに Korek 攻撃と SVV_10 のバイアスによる攻撃関数の投票条件を無効にすることで Strong IV の安全性を向上させる。Korek 攻撃および SVV_10 バイアスによる攻撃関数と投票条件を表 1 に示す。表 1 の攻撃関数はそれぞれの投票条件が全て成立した場合にのみ WEP 鍵の投票に利用できる。そのため投票条件が一つでも成立しない場合は攻撃関数による WEP 鍵の候補値への投票が行われない。この投票条件の性質を利用して Korek 攻撃および SVV_10 のバイアスによる攻撃関数に対応した

Strong IV を構成する。それぞれの投票条件を無効にする Strong IV の生成方法を以下に示す。

Step1 RC4 の実行

判定を行う IV を生成し、WEP 鍵と合わせて秘密鍵として RC4 を実行する。

Step2 投票条件の判定

RC4 の処理から得られる内部状態およびキーストリームに対して投票条件の成立状況を判定、記録する。

Step3 投票の可否の判定

Step2 で記録した投票条件の成立状況からそれぞれの攻撃関数に対する投票条件が全て成立しているかを確認し、攻撃関数による投票の可否を判定する。

Step4 Strong IV の判定

全ての攻撃関数による投票が不可能であれば、IV を Strong IV と判定する。攻撃関数による投票が可能なものがあれば Step1 に戻り、IV を再生成したのち判定を行う。

上記と同時に Klein 攻撃の攻撃成功条件に関する判定を行うことで Klein 攻撃と同様の成功条件を持つ攻撃および Tornado 攻撃が成立しない IV が生成される。このようにして生成された Strong IV は WEP に対する既存の鍵回復攻撃全てを困難にすることができる。

5.2 Strong IV を用いた WEP の運用

5.1 節で提案した Strong IV を用いた WEP の運用方法について述べる。従来の場合と同様に 100,000 個のパケットごとに鍵を更新することを想定する。100,000 個のうち 50,000 個は Klein 攻撃および Korek 攻撃、SVV_10 のバイアスによる攻撃関数を無効にする IV (Strong IV1) を利用し、残りの 50,000 個は Klein 攻撃を除く攻撃関数を無効にする IV (Strong IV2) から構成する。100,000 個の IV 全体で Klein 攻撃の成功確率を 1/256 にするためにこのような構成とする。次に Strong IV の判定と暗号文の生成を並行して行う方法を述べる。Strong IV の判定には一定の長さのキーストリームを生成する必要がある。そこで判定時に生成するキーストリームの長さを平文の長さと同じにし、Strong IV の判定と同時に生成されたキーストリームを用いて暗号化を行う。Strong IV の判定を通過した場合は生成された暗号文を送信する。Strong IV でないと判定された場合は暗号文を破棄し、新しい IV に対して同様にして Strong IV の判定と暗号文の生成を行う。上記のようにして Strong IV の判定と同時に暗号文の生成を行うことで、一回の通信あたりにかかる負荷を軽減する。

Strong IV の判定と並行して暗号文を生成するのに要する時間の評価を行う。RC4 のみを実行した場合、Klein 攻撃および Korek 攻撃、SVV_10 のバイアスによる攻撃関数を無効にする場合、Korek 攻撃および SVV_10 のバイアスによる攻撃関数を無効にする場合の三つに対して 50,000 個の暗号文を生成するのにかかる時間を計測する。平文の長さを 256 バイト、1024 バイト、1472 バイトとし、異なる 256 個の WEP 鍵に対して実験を行い、暗号化に要した時間の平均値を算出する。1472 バイトはイーサネットおよび無線 LAN などにおける 1 パケットの最大サイズからヘッダなどの情報を除いた場合の長さである。実験は計算機上のシミュレーションとして行う。使用した計算機の CPU は Core i3 530 である。実験結果を表 2 に示す。表 2 より平文の長さが 1472 バイトの場合に

表 1 Korek 攻撃および SVV_10 の関係性と投票条件

名称	攻撃関数	投票条件
SVV_10	$S_t^{-1}[0] - \sigma_{16}(t)$	$S_t^{-1}[0] < t + 1$ or $S_t^{-1}[0] > 15$, $Z_{16} = -16, j_2 \notin \{t + 1, t + 2, \dots, 15\}$
A_u15	$2 - \sigma_y(t)$	$S_t[y] = 0, Z_2 = 0$
A_s13	$S_t^{-1}[0] - \sigma_y(t)$	$S_t[1] = y, (S_t^{-1}[0] < t + 1$ or $S_t^{-1}[0] > y - 1)$, $Z_1 = y$
A_u13_1	$S_t^{-1}[Z_1] - \sigma_y(t)$	$S_t[1] = y, (S_t^{-1}[Z_1] < t + 1$ or $S_t^{-1}[Z_1] > y - 1)$, $Z_1 = 1 - y$
A_u13_2	$1 - \sigma_y(t)$	$S_t[y] = y, S_t[1] = 0, Z_1 = y$
A_u13_3	$1 - \sigma_y(t)$	$S_t[y] = y, S_t[1] = 1 - y, Z_1 = 1 - y$
A_s5_1	$S_t^{-1}[Z_1] - \sigma_y(t)$	$S_t[1] < t + 1, S_t[1] + S_t[S_t[1]] = y$, $Z_1 \neq \{S_t[1], S_t[S_t[1]]\}$, $(S_t^{-1}[Z_1] < t + 1$ or $S_t^{-1}[Z_1] > y - 1)$
A_s5_2	$S_t^{-1}[S_t[1] - S_t[2]] - \sigma_y(t)$	$S_t[2] + S_t[1] = y, Z_2 = S_t[1]$, $S_t^{-1}[S_t[1] - S_t[2]] \neq \{1, 2\}$, $(S_t^{-1}[S_t[1] - S_t[2]] < t + 1$ or $S_t^{-1}[S_t[1] - S_t[2]] > y - 1)$
A_s5_3	$S_t^{-1}[Z_2] - \sigma_y(t)$	$S_t[2] + S_t[1] = y, Z_2 = 2 - S_t[2]$, $S_t^{-1}[Z_2] \neq \{1, 2\}$, $(S_t^{-1}[Z_2] < t + 1$ or $S_t^{-1}[Z_2] > y - 1)$
A_u5_1	$S_t^{-1}[S_t^{-1}[Z_1] - y] - \sigma_y(t)$	$S_t[1] = y, S_t^{-1}[Z_1] < t + 1, S_t^{-1}[S_t^{-1}[Z_1] - y] \neq 1$, $(S_t^{-1}[S_t^{-1}[Z_1] - y] < t + 1$ or $S_t^{-1}[S_t^{-1}[Z_1] - y] > y - 1)$ $Z_1 \neq \{y, 1 - y, S_t^{-1}[Z_1] - y\}, S_t^{-1}[Z_1] \neq 2y$
A_u5_2	$1 - \sigma_y(t)$	$S_t[y] = 1, Z_1 = S_t[2]$
A_u5_3	$1 - \sigma_y(t)$	$S_t[y] = y, S_t^{-1}[Z_1] \neq 1, S_t^{-1}[Z_1] < t + 1, Z_1 = S_t[S_t[1] + y]$
A_s3	$S_t^{-1}[Z_2] - \sigma_y(t)$	$S_t[1] \neq 2, S_t[2] \neq 0, S_t[2] + S_t[1] < t + 1$, $S_t[2] + S_t[S_t[2]] + S_t[1] = y, S_t^{-1}[Z_2] \neq \{1, 2, S_t[1] + S_t[2]\}$, $S_t[1] + S_t[2] \neq \{1, 2\}, (S_t^{-1}[Z_2] < t + 1$ or $S_t^{-1}[Z_2] > y - 1)$
A_4_s13	$S_t^{-1}[0] - \sigma_4(t)$	$S_t[1] = 2, S_t[4] \neq 0$, $(S_t^{-1}[0] < t + 1$ or $S_t^{-1}[0] > y - 1), Z_2 = 0$
A_4_u5_1	$S_t^{-1}[254] - \sigma_4(t)$	$S_t[1] = 2, Z_2 \neq 0$, $Z_2 \neq 254, (S_t^{-1}[254] < t + 1$ or $S_t^{-1}[254] > 3)$
A_4_u5_2	$S_t^{-1}[255] - \sigma_4(t)$	$S_t[1] = 2, Z_2 \neq 0$, $(S_t^{-1}[255] < t + 1$ or $S_t^{-1}[255] > 3), Z_2 = S_t[2]$
A_neg_1	$1 - \sigma_y(t)$ or $2 - \sigma_y(t)$	$S_t[2] = 0, S_t[1] = 2, Z_1 = 2$
A_neg_2	$2 - \sigma_y(t)$	$S_t[2] = 0, S_t[1] \neq 2, Z_2 = 0$
A_neg_3	$1 - \sigma_y(t)$ or $2 - \sigma_y(t)$	$S_t[1] = 1, Z_1 = S_t[2]$
A_neg_4	$-\sigma_y(t)$ or $1 - \sigma_y(t)$	$S_t[1] = 0, S_t[0] = 1, Z_1 = 1$

表 2 50,000 個の暗号文の生成に要した時間

平文の長さ (バイト)	256	1024	1472
RC4(ミリ秒)	104	232	306
Strong IV1(ミリ秒)	493	618	700
Strong IV2(ミリ秒)	470	593	669

一回の暗号化にかかる時間はそれぞれ 6.12 マイクロ秒, 14.00 マイクロ秒, 13.38 マイクロ秒となる. Strong IV1 と Strong IV2 を混合して 100,000 パケットの通信を行う場合, Strong IV を用いない場合と比べて 2.2~2.3 倍の時間を要するが既存の鍵回復攻撃を全て困難にして安全な通信を行うことができる.

6 まとめ

本稿では WEP のより安全な運用方法を提案した. 計算機シミュレーションにより, 提案方式を用いた場合における WEP の暗号化に必要な時間の評価を行った. 提案方式では入山らの Strong IV に加えて Tornado 攻撃における Korek 攻撃, SVV_10 のバイアスによる攻撃関数に対応した Strong IV を用いる. Tornado 攻撃の攻撃関数による投票が投票条件の成立に依存する性質を利用して, 任意の IV に対して投票条件の成立を効率的に判定する方法を提案した. 100,000 パケットごとに鍵を更新することを想定した WEP の運用方法として Klein 攻撃, Korek 攻撃, SVV_10 のバイアスによる攻撃関数を全て無効にする Strong IV を 50,000 個, Klein 攻撃を除いた攻撃関数を無効にする IV を 50,000 個混合して利用する場合に通信へ

の負荷を軽減する方法として Strong IV の判定と暗号化を並行して行うことを提案した。また IV の判定と暗号化とを並行して行った場合に要する時間の評価を行った。結果として Strong IV の判定を行う場合は従来と比べて一回の暗号化に約 2.3 倍程度の時間を要することを示した。提案手法により WEP における従来の鍵回復手法を困難にし、高速な通信が可能となる。

参考文献

- [1] IEEE Computer Society, “Wireless lan medium access control(MAC) and physical layer(PHY) specifications”. IEEE Std 802.11, 1999.
- [2] S.R. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4,” Selected Areas in Cryptography, 8th Annual International Workshop, SAC 2001, pp.1–24, 2001.
- [3] Korek, “Next generation of WEP attacks?,” 2004. <http://www.netstumbler.org/showpost.php?p=93942&postcount=35>.
- [4] R. Chaabouni, “Break WEP Faster with Statistical Analysis,” Technical report, EPFL, LASEC, June 2006.
- [5] A. Klein, “Attacks on the RC4 stream cipher,” Des. Codes Cryptography, vol.48, no.3, pp.269–286, 2008.
- [6] E. Tews, R. Weinmann, and A. Pyshkin, “Breaking 104 Bit WEP in Less Than 60 Seconds,” Information Security Applications, 8th International Workshop, WISA 2007, pp.188–202, 2007.
- [7] R. Teramura, Y. Asakura, T. Ohigashi, H. Kuwakado, and M. Morii, “Fast WEP-Key Recovery Attack Using Only Encrypted IP Packets,” IEICE Transactions, vol.93-A, no.1, pp.164–171, 2010.
- [8] T. Ohigashi, H. Kuwakado, and M. Morii, “A Key Recovery Attacks on WEP with Less Packets”. Technical Report of IEICE, ISEC, 2007.
- [9] P. Sepehrdad, P. Susil, S. Vaudenay, and M. Vuagnoux, “Smashing WEP in a Passive Attack,” Fast Software Encryption - 20th International Workshop, FSE 2013, pp.155–178, 2013.
- [10] I.C. Society, “802.1x-port based network access control”. IEEE Std 802.11, 2001.
- [11] 塚畝翼, 藤堂洋介, 森井昌克, “既存鍵回復攻撃を無効にする wep 運用の提案”. 電子情報通信学会技術研究報告, LOIS, 2011.
- [12] 入山敬大, 渡辺優平, 森井昌克, “WEP における Strong IV の評価とその実装”. 暗号と情報セキュリティシンポジウム (SCIS), 2014.
- [13] P. Sepehrdad, S. Vaudenay, and M. Vuagnoux, “Discovery and Exploitation of New Biases in RC4,” Selected Areas in Cryptography - 17th International Workshop, SAC 2010, pp.74–91, 2010.
- [14] S. Maitra and G. Paul, “New Form of Permutation Bias and Secret Key Leakage in Keystream Bytes of RC4,” Fast Software Encryption, 15th International Workshop, FSE 2008, pp.253–269, 2008.