2008－CSEC－43 (9)
2008／12／5

WEPを一瞬にして解読する方法 –任意のIPパケットからWEP鍵を導出するアルゴリズムと– その実装・評価

寺村亮一 † 朝倉康生 † 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., yasakura@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 Wired Equivalent Privacy (WEP) に対する従来の鍵解読法は，特定のIV，もしくは特定のパケットを利用する，いわば選択平文攻撃である．そのため実際の環境では，それらの特定のパケットを不正アクセス等によって大量に収集する必要があった．本稿ではWEPで暗号化された任意のIPパケットから高速に秘密鍵を導出する暗号文単独攻撃法を提案する．本攻撃法によって30,000程度のIPパケットしか得られない環境においても，WEPの104ビット秘密鍵を導出できる．さらに本攻撃法を実装し，一般に流通しているパソコン環境を用いて，10秒程度で鍵を導出できることを示す．

The Method for Recovering 104-bit WEP Key Using Few IP Packets

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

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

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

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

ohigashi@hiroshima-u.ac.jp

Abstract Most attacks on the WEP protocol proposed in the past recover the key using ARP packets and can be detected by an IDS. In this paper, we proposed the key recovery attack on the WEP protocol using IP packets. Our attack recovers the 104-bit key with a probability of 0.5 when 30,000 IP packets are given and cannot be detected by an IDS.

1 はじめに

Wired Equivalent Privacy (WEP) [1] はIEEE 802.11bで規定されている，暗号化アルゴリズムにRC4ストリーム暗号を採用した無線LAN用のセキュリティプロトコルである．最初に発表されたWEPに対する鍵回復攻撃は2001年にFluhrerらによって提案された攻撃であり，FMS攻撃と呼ばれている [2][3]．FMS攻撃は特定のIV (weak IV) が用いられた際に，キーストリームに秘密鍵の情報が高い確率で漏えいすることを利用しており，この攻撃を行えばおよそ4,000,000から6,000,000の暗号化パケットを観測することで104ビット鍵の推測が可能になる．FMS攻撃はweak IVをフィルタリングして取り除くことで無効化できたが [4]，2004年にIVのフィルタリングでは防げない攻撃がKoreKにより提案された [5]．

WEPの解読研究に関して重要な成果が2006年にKleinにより提案された関連鍵攻撃である [6]．これはある関数を利用することでRC4のキーストリームから秘密鍵の値を高い確率で推測し，これを繰り返すことで鍵の値を決定する攻撃である．そして2007年にTewsらはKleinの関連鍵攻撃をWEP解読用に拡張した関連鍵攻撃を提案した [7]．Tewsらの手法は40,000 ARPパケットを観測することで確率0.5

で，また85,000 ARPパケットを観測することで確率0.95でそれぞれ104ビットの秘密鍵を導出できる．

しかしネットワーク上を流れるパケットの多くはIPパケットであり，観測のみで大量のARPパケットを収集するには多大な時間を要する．この問題を解決する方法としてARPインジェクションの利用が挙げられるが，これは直接アクセスポイント等を攻撃するため侵入検知システム (IDS) 等により容易に防ぐことが可能である．

本稿では特定のパケットによらず，任意のIPパケットから高速にWEPの104ビットの秘密鍵を導出する手法を提案する．IPパケットから秘密鍵を導出する場合，ARPインジェクション等のアクセスポイントへの攻撃を行わないためIDSを導入した環境であっても攻撃を防ぐことが困難となる．提案手法では任意のIPパケットから秘密鍵を導出するために，Kleinの関連鍵攻撃をTewsとは異なる形に拡張し，文献[8]–[10]での結果も利用することによって，より少数のIPパケットから高い確率で鍵を推測する方法となっている．さらに提案手法を実装し，30,000 IPパケットを観測することで確率0.5で，50,000 IPパケットを観測することで確率0.95でそれぞれ104ビットの秘密鍵を導出できることを確認した．

2 WEPの概要

Wired Equivalency Privacy (WEP) はIEEE 802.11bに規定されている無線LANのセキュリティプロトコルであり，その暗号化/復号にはRC4ストリーム暗号を利用している．本節ではRC4，WEPの概要を述べる．

2.1 RC4

RC4は1987年にRivestによって提案された，非常に広く利用されているストリーム暗号である．RC4の内部状態は置換配列 $S = (S[0], S[1], \ldots, S[255])$ と二つのポインタ i，j で構成され，そのアルゴリズムは鍵スケジューリング部 (KSA : Key Scheduling Algorithm) と擬似乱数生成部 (PRGA : Pseudo Random Generation Algorithm) で構成される．ここでKSAは秘密鍵を用いて内部状態を初期化するアルゴリズムであり，PRGAは初期化された内部状態 (初期状態) からキーストリームと呼ばれる擬似乱数系列を生成するアルゴリズムである．RC4のKSA，PRGAはともにスワップ処理を中心として構成されており，図1，2のようにそれぞれ非常に簡潔な形で記述できる．尚，図中の l は鍵のバイト長を，z はキーストリームを表す．また本稿では全ての演算は N を法として行われる．

Step 1 $S[x] \leftarrow x$ for $\forall x \in \{0, 1, \ldots, N-1\}$

Step 2 $i \leftarrow 0$, $j \leftarrow 0$

Step 3 For $\forall i \in \{0, 1, \ldots, N-1\}$

1. $j \leftarrow (j + S[i] + K[i \bmod l])$
2. Swap $S[i]$ and $S[j]$

図 1: KSAの概要

Step 1 $i \leftarrow 0$, $j \leftarrow 0$

Step 2 Loop

1. $i \leftarrow (i+1)$
2. $j \leftarrow (j + S[i])$
3. Swap $S[i]$ and $S[j]$
4. Output $z \leftarrow S[(S[i] + S[j])]$

図 2: PRGAの概要

PRGAにより得られたキーストリームと平文/暗号文の排他的論理和をとることでRC4の暗号化/復号は完了する．

2.2 WEP

WEPはIEEE802.11bで規定されている無線LAN用のセキュリティプロトコルであり，そのパケットの暗号化アルゴリズムにはRC4を用いる．しかしながら暗号プリミティブとしてのRC4と異なり，WEPではパケット毎に生成される鍵(パケット鍵)K を40ビットまたは104ビットの秘密鍵 K' と24ビットの初期化ベクトル IV を用いて $K = K' \| IV$ のように生成する．ここで $\|$ はビットの連結である．このパケット鍵と秘密鍵の関係はパケット鍵のバイト毎の値を $K[i]$ とすると式(1)で表せる．

$$K[i] = \begin{cases} IV[i] & \text{for } \forall i \in \{0,1,2\}, \\ K'[i-3] & \text{for } \forall i \in \{3,4,\ldots,15\} \end{cases} \quad (1)$$

ここで $K[0]$，$IV[0]$，$K'[0]$ はそれぞれパケット鍵，初期化ベクトル，秘密鍵の最上位バイトを表す．このパケット鍵をRC4に入力し，得られた擬似乱数

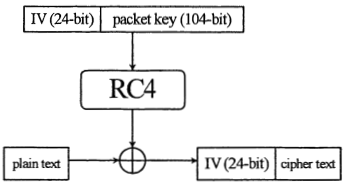

図 3: WEP の概要

とパケットの排他的論理和をとることで，WEP の暗号化/復号は行われる (図 3 参照).

3 Klein の攻撃関数

PTW 攻撃等の過去に提案された WEP への攻撃の多くは Klein により発見された RC4 における秘密鍵-キーストリームの相関を利用してパケットから秘密鍵の推測を行う [6]. この相関は以下の式 (2) に示す攻撃関数 f_{Klein} の形で表すことができる.

$$\begin{aligned} &f_{Klein}(K[0], K[1], \ldots, K[x-1], z_{x-1}) \\ &= (S_x^{-1}[x - z_x] - (j_x - S_x[x])) \\ &= \begin{cases} K[x] & \text{with Prob} = P[f_{Klein}] \\ a \neq K[x] & \text{with Prob} < \frac{1}{N} \end{cases} \end{aligned} \tag{2}$$

ここで $P[f_{Klein}]$ を式 (3) に示す.

$$\begin{aligned} P[f_{Klein}] &\approx \left(1 - \frac{1}{N}\right)^{N-2} \cdot \frac{2}{N} \\ &\quad + \left(1 - \left(1 - \frac{1}{N}\right)^{N-2}\right) \cdot \frac{N-2}{N(N-1)} \\ &\approx \frac{1.36}{N} \end{aligned} \tag{3}$$

Klein は 25,000 パケット分のキーストリームを観測した後，この攻撃関数をそれぞれのキーストリームに対して繰り返し適用することである $K[x]$ を確率的に導出できると述べている.

4 拡張した攻撃関数の提案

RC4 と同様に，WEP においても Klein の提案した鍵導出手法を順次 $K[0]$ から $K[l-1]$ まで繰り返し行うことで全ての秘密鍵の値を導出できる．しかしこの手法は途中で導出した鍵が誤っていた場合，それを訂正する度に後続の鍵を順次導出し直す必要があるため非常に計算コストがかかることが Tews らにより文献 [7] 中で指摘されている．本章ではこの問題に対処できるよう Klein の攻撃関数を拡張した新しい攻撃関数を提案する.

RC4 の KSA より，j_x は $K[x-1]$ までの値を用いて $j_x = \sum_{n=0}^{x-1}(K[n] + S_n[n])$ と表せるため，式 (2) より Klein の攻撃関数は以下に示す鍵を近似する関数として置き換えることができる.

$$K[x] \approx S_x^{-1}[x - z_x] - (\sum_{n=0}^{x-1}(K[n] + S_n[n]) - S_x[x]) \tag{4}$$

ここで鍵の $x-1$ バイト目の値に誤った値 $K'[x-1] = K[x-1] - \Delta K[x-1]$ を代入したと仮定すると，そのときの $j_x = \sum_{n=0}^{x-1}(S_n[n] + K[n]) - \Delta K[x-1]$ となり，また x バイト目の鍵の推測値 $K_g[x]$ は式 (5) で近似できる．S' は K' によって生成される内部状態を意味する.

$$\begin{aligned} K_g[x] &\approx S_x'^{-1}[x - z_x] - (\sum_{n=0}^{x-1}(S_n[n] + K[n]) \\ &\quad - \Delta K[x-1] - S_x'[x]) \\ &\approx S_x^{-1}[x - z_x] - (\sum_{n=0}^{x-1}(S_n[n] + K[n]) \\ &\quad - S_x[x]) + \Delta K[x-1] \\ &\approx K[x] + \Delta K[x-1] \end{aligned} \tag{5}$$

これより x バイト目の鍵の推測値 $K_g[x]$ は正しい鍵の値から $x-1$ バイト目の鍵の値の誤差分だけずれた値が高い確率で導出されると予測できる.

次に $K'[x-1]$ と $K'[x] = K_g[x] - \Delta K[x]$ を用いて $K[x+1]$ を導出する場合について考察する．ここで $K'[x-1] + K'[x] = K[x-1] + K[x] - \Delta K[x]$ となり $\Delta K[x-1]$ は影響しない．このため先ほどと同様に考察すると，$j_{x+1} \approx \sum_{n=0}^{x}(S_n[n] + K[n]) - \Delta K[x]$ となり $K_g[x+1] \approx K[x+1] + \Delta K[x]$ の近似を得ることができる.

以上より，Klein の攻撃関数は以下の式 (6) の形に拡張できる.

$$\begin{aligned} &f_x(K'[0], K'[1], \ldots, K'[x-1], z_{x-1}) \\ &= (S_x'^{-1}[x - z_x] - (j_x' - S_x'[x])) \\ &= \begin{cases} K[x] + \Delta K[x-1] & \text{with Prob} = P[f_x] \\ a \neq K[x] + \Delta K[x-1] & \text{with Prob} < \frac{1}{N} \end{cases} \end{aligned} \tag{6}$$

正しい鍵を導出できる確率 $P[f_x]$ は以下の定理より与えられる．

定理 1. i を集合 $\{0,\ldots,x-1\}$ に含まれる自然数と定義する．そして $\Delta K[i] \neq 0$ となる個数を e，またそのときの i の値の集合を E とする．このとき $P[f_x]$ は以下の式で近似できる．

$$P[f_x] \approx q_x \cdot \left(1-\frac{1}{N}\right)^{N-2} \cdot \frac{2}{N} + \left(1-q_x \cdot \left(1-\frac{1}{N}\right)^{N-2}\right) \cdot \frac{N-2}{N(N-1)} \quad (7)$$

式 *(7)* 中の q_x は以下の式から導出できる．

$$q_x = \left(1-\frac{1}{N}\right)^{e} \cdot \left(1-\frac{e}{N}\right) \cdot \prod_{k\in E}^{x}\left(1-\frac{x-k}{N}\right) \quad (8)$$

証明 式 (7) の導出過程は文献 [7] で説明されているため本証明では省略する．式 (6) が (1)$S'^{-1}_x[x-z_x] = S^{-1}_x[x-z_x]$, (2)$j'_x = j_x - \Delta K[x-1]$, (3)$S'_x[x] = S_x[x]$ の条件を満たすと仮定する．このとき式 (6) は式 (2) と等しくなり，正しい $K[x]$ が推測される確率は式 (3) で与えられる．条件 (1)–(3) を満たす事象が生じる確率を求めることで q_x を導出できる．

t_e を集合 E に含まれる任意の値と定義する．このとき $j_{\forall t_e} \notin \{t_e,\ldots,x\}$ ならば，$S'_i[i] = S_i[i]$ for $\forall i$ が成立する．よって条件 (3) が成立し，また $j'_x = \sum_{n=0}^{x-1}(S_n[n]+K'[n]) = j_{t-1} - \Delta K[x-1]$ より条件 (2) が成立する．j が一様にランダムに分布していると仮定すると，条件 (2)，(3) を満たす事象が生じる確率は $\prod_{k\in E}^{x}(1-(x-k)/N)$ となる．次に条件 (1) を満たす確率について考察する．条件 (1) が成立するには (i)$S'^{-1}_x[x-z_x] \neq j_{\forall t_e}$ かつ，(ii)$S'^{-1}_x[x-z_x] \notin E$ をそれぞれ満たす必要がある．(i) を満たす確率は $(1-(1/N))^e$，(ii) を満たす確率は $(1-(e/N))$ である．これら導出した確率の積より q_x が与えられる． □

5 秘密鍵導出アルゴリズムの提案

本章では前章で提案した攻撃関数を利用して，WEP を用いて暗号化された IP パケットから秘密鍵を導出するアルゴリズムを提案する．この提案するアルゴリズムは (1) 暗号化された IP パケットからキーストリームを導出する”キーストリーム導出部”，(2) キーストリームから秘密鍵のテーブルを生成する”鍵テーブル生成部”，(3) 鍵テーブルから正しい秘密鍵を決定する”秘密鍵決定部”の三つの部分から構成されている．以下、この三つの部分の詳細を説明する．

5.1 キーストリーム導出部

WEP の暗号化はパケットにおけるネットワーク層の上位に位置する部分とキーストリームの排他的論理和をとることで行われるため，ネットワーク層より上位の層に含まれる平文を決定できれば，それに対応した部分のキーストリームを導出することができる．また MAC ヘッダの情報等，ネットワーク層より下位の層に対応した平文は暗号化されないため，容易に取得することが可能である．IEEE 802.11 において MAC ヘッダには，MAC アドレスやそのパケットの宛先がアクセスポイントであるかクライアントであるか等の情報が含まれている．このうち MAC アドレスの情報を用いることで，パケットの暗号化された部分から 13，14 バイト目以外の先頭 15 バイト分のキーストリームを導出できる．この先頭 15 バイトまでのキーストリーム導出過程の詳細は文献 [7] に記されているため，本稿では省略する．

さらに IP パケットにおける 16 バイト目以降の平文に注目すると 16 バイト目のフラグオフセットには 0x00 が [7]，また 18 バイト目の使用プロトコルには TCP を意味する 0x06 が含まれていると推測できる．加えて LAN 環境では一般的にクライアントに対して以下に示すクラス A から C のプライベートアドレスを割り振り通信を行うため，宛先がアクセスポイントのパケットであれば送信元の IP アドレスが，アクセスポイントから送られてきたパケットであれば送信先の IP アドレスが推測できる．

クラス A： 10.0.0.0–10.255.255.255

クラス B： 172.16.0.0–172.31.255.255

クラス C： 192.168.0.0–192.168.255.255

例えばクラス C のプライベートアドレスを用いた環境において，MAC ヘッダにおける toDS の値が 1 であれば送信元 IP アドレスの上位 2 バイトを表す 21，22 バイト目が，fromDS の値が 1 であれば送信先 IP アドレスの上位 2 バイトを表す 25，26 バイト目が”192”，”168”をとる．

観測した全てのIPパケットに対して，上記の手法より得られるキーストリーム，パケットの暗号化に用いられたIV，toDS，fromDSの値をメモリに格納することでキーストリーム導出部は終了する．

5.2 鍵テーブル生成部

鍵テーブル生成部は"前処理部"と"本処理部"の二つに分けることができる．このうち前処理部では仮の鍵推測値であるK_{tmp}と$K[12]+K[11]+K[10]$の値を導出し，本処理部ではこれらの値を用いて$K[0]$から$K[9]+\Delta K[8]$までの鍵テーブル$KT[0]$から$KT[0]$を生成する．

5.2.1 前処理部

前章で提案した攻撃関数を利用すると，ある$K[x]+\Delta K[x-1]$の値をキーストリームZ_{x-1}と$K[0]$から$K[x-1]$までの推測値から導出できる．まず得られたキーストリームに対して$f(z_2)$を繰り返し計算し，もっとも多く得られた値を$K[0]$の仮の推測値$K_{tmp}[0]$とする．この動作を順次繰り返すことで$K_{tmp}[0]$から$K_{tmp}[9]$の値を得ることができる．このK_{tmp}の値で生成されるS_9に対して，文献[8][9]のアイディアを拡張して利用することで，z_{15}，z_{16}，z_{18}からそれぞれ$K[12]+K[11]+K[10]$の推測値を導出できる．

5.2.2 本処理部

鍵テーブルを作成する際に，$KT[x]$ for $\forall x \in \{0,1,4,5,8,9\}$に関しては前処理部と同様の処理を行い鍵テーブルを作成し，それぞれの鍵テーブルにおいて最も投票が多かった値を鍵の推測値K'として以降の鍵テーブル生成に利用する．また$KT[y]$ for $\forall y \in \{2,3,6,7\}$に関してはこの処理に加えて，文献[10]のアイディアを利用して，toDS=1のキーストリームに関してはz_{21}，z_{22}を利用して$KT[2]$，$KT[3]$に，fromDS=1のキーストリームに関してはz_{25}，z_{26}を利用して$KT[6]$，$KT[7]$にそれぞれ投票を行う．提案アルゴリズムでは文献[10]中の$\hat{S}$，$\hat{j}$をK'に加えてK_{tmp}を用いて導出ことでより鍵の推測確率を上昇させることに成功しているが，詳細は紙面の余白の都合により省略する．以上の鍵テーブル生成部のアルゴリズムの流れを図に示す．

Step 1 攻撃関数f_aと$z_2,\ldots,z_{11}$から順次$K_{tmp}[0]$から$K_{tmp}[9]$まで導出．

Step 2 $K_{tmp}[0],\ldots,K_{tmp}[9]$，$z_{14}$，$z_{15}$，$z_{17}$からそれぞれ$K[12]+K[11]+K[10]$を導出．

Step 3 $\forall x \in \{0,\ldots,9\}$において下記の1から4を試行

1. $KT[x]$に攻撃関数f_aを用いて全てのz_{x-1}から投票．
2. もし$x \in \{2,3\}$ならばtoDS=1の際のz_{21}，z_{22}から$KT[2]$，$KT[3]$に投票．
3. もし$x \in \{6,7\}$ならばfromDS=1の際のz_{25}，z_{26}から$KT[6]$，$KT[7]$に投票．
4. $K'[x]$を決定

図4: 鍵テーブル生成部の流れ

5.3 秘密鍵決定部

鍵テーブル生成部より$KT[0]$–$[9]$，$K[12]+K[11]+K[10]$の値が得られる．ここで鍵テーブルが投票率が高い候補を上位として並びかえられているとすると，大量のパケットが得られていたならばそれぞれの鍵テーブルの最上位の候補は正しい鍵の値となる．ただ十分なパケット量が得られなければ，正しい鍵の値が上位の候補とはなるものの最も投票された候補とならないことがある．この問題に対処するため，秘密鍵決定部では鍵テーブル毎に上位の複数の候補を鍵の値と仮定して探索を行う．

任意の鍵テーブル$KT[x]$の探索する候補数を$C[x]$とする．このとき$KT[x]$の最上位の値を$K_{top}[x]$，鍵と仮定した値を$K'[x]$とすると$\Delta K[x]=K'[x]-K_{top}[x]$となる．この値を次の$K'[x+1]$から引くことで，鍵テーブルを作り直さずとも正しい$K[x+1]$の導出を行うことができる．例として$K_{top}[x-1]=10$，$K'[x-1]=50$，$K_{top}[x]=70$，$K'[x]=90$がそれぞれ代入された際の$K[x]$，$\Delta K[x]$が導出される流れを図5に示す．これを利用して全ての鍵テーブルにおいて上位$C[x]$個の組み合わせを試行することで広い範囲の鍵が探索できる．

実際のアルゴリズムではまず$C[x]$には初期値として0が代入される．すなわち$K_{top}[x]$のみを鍵の値と仮定し，残りの$K[10]$，$K[11]$の値を全数探索することで104ビット鍵の探索を行う．そして正しい104ビット鍵が得られなければ，$K_{top}[x]$の投票率と$C[x]+1$番目の候補の投票率の距離が最も小さい鍵テーブルにおいて，$C[x]=C[x]+1$の計算を行い，探索する鍵の候補数を増加させる．これを繰り返すことで鍵の探索を行う．

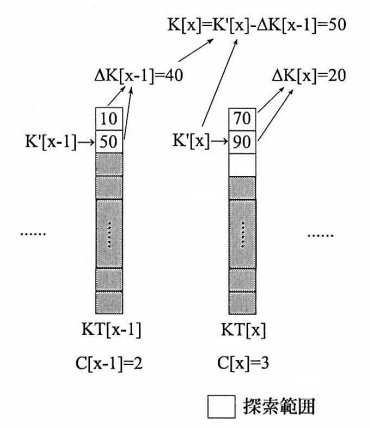

図 5: 秘密鍵決定処理

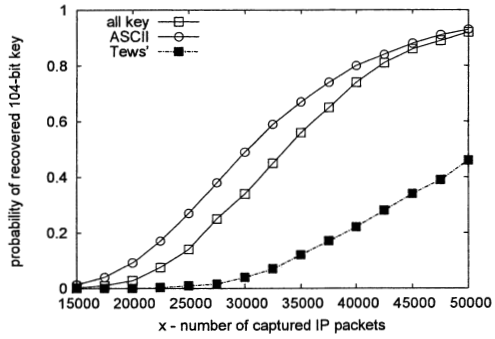

図 6: 任意の IP パケットからの鍵導出成功率

6 数値実験結果

前章で提案したアルゴリズムを実装し 104 ビット鍵から得られた IP パケットからの秘密鍵の導出実験を行った．試行回数はそれぞれの手法毎に 10,000 回行い，1 試行毎に鍵テーブルから候補として挙げる秘密鍵の上限個数は 1,000,000 個と設定した[1]．その結果を図 6 に示す．図 6 において"all key"は本稿の提案アルゴリズムを実装して IP パケットから秘密鍵の導出を行った際の攻撃成功確率を，"Tews'"は Tews らの攻撃関数のみを用いて IP パケットから秘密鍵の導出を行った際の成功確率を表す．また実際に WEP が導入されている環境において，秘密鍵は ASCII の文字列から構成されていることが多い．このため探索範囲を ASCII の印字可能文字の範囲 (0x20–0x7e) に限定する改良をアルゴリズムに施し，10,000 個の ASCII 鍵に対して同様のシミュレーションを行った．その結果を図 6 中の"ASCII"に示す．

ARP インジェクションを行うことで大量に入手できる ARP パケットと比較して，十分な数の IP パケットを得るためには時間を要するため，少ないパケット数で鍵の導出が成功することがアルゴリズムの能力としてより重要となる．図 6 より提案アルゴリズムは Tews の手法ではほぼ成功しない 20,000–30,000 といった少量の IP パケットしか得られない環境においても，十分高い確率で 104 ビット鍵の導出が可能となることが確認できる．また提案アルゴリズム単体での鍵導出に要する時間は，得られたパケットの数等の要因で若干変動するものの 1,2 秒程度であり，Tews らの手法に要する時間と同等であった．

[1] $K[10]$, $K[11]$ の値は全数探索を行うため，実際に試すことのできる鍵の組み合わせは $1,000,000 \cdot 2^{-8}$ 通りである．

7 むすび

本稿では任意の IP パケットから WEP を高速に解読するアルゴリズムを提案し，その数値実験結果を示した．本解読法によって IP パケットを観測するだけで，WEP における秘密鍵を高速に推定することが可能になった．特に 30,000–50,000 パケットを観測することにより，一般に流通しているパソコンを利用し，数秒以内に秘密鍵を導出できることを示した．

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. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," Proc. SAC2001, Lecture Notes in Computer Science, vol.2259, pp.1–24, Springer-Verlag, 2001.

[3] S. Fluhrer, I. Mantin, and A. Shamir, "Attacks on RC4 and WEP," CryptoBytes, vol.5, no.2, pp.26–34, RSA Laboratories, 2002.

[4] Orinoco, "WEPplus white paper," Oct. 2001.

[5] KoreK, "Next generation of WEP attacks?," 2004, available at http://www.netstumbler.org/showpost.php?p=93942&postcount=35

[6] A. Klein, "Attacks on the RC4 stream cipher,"submitted to Designs, Codes and Cryptography ePrint, 2007.

[7] E. Tews, R. Weinmann, and A. Pyshkin, "Breaking 104 bit WEP in less than 60 seconds," Cryptology ePrint, 2007, available at http://eprint.iacr.org/2007/120.pdf

[8] Y. Ozasa, Y. Fujikawa, T. Ohigashi, H. Kuwakado, and M. Morii, "A study on the Tews-Weinmann-Pyshkin attack against WEP," IEICE Technical Report, ISEC2007-47, pp.17–21, Jul. 2007.

[9] Serge Vaudenay and Martin Vuagnoux, "Passive-only key recovery attacks on RC4," Proc. SAC2007, Lecture Notes in Computer Science, vol.4876, pp.344–359, Springer-Verlag, 2007.

[10] T. Ohigashi, H. Kuwakado, and M. Morii, "A Key Recovery Attack on WEP with Less Packets," IEICE Technical Report, ISEC2007-109, pp.61–68, Nov. 2007.