無線 LAN 暗号化方式 WEP の秘密鍵を
IP パケットから高速に導出する手法の実装と検証

島根大学　総合理工学部　数理・情報システム学科

応用情報学講座　田中研究室

S103082　前田恒太

平成 26 年 2 月 3 日

目次

第１章　はじめに

近年、無線 LAN の普及が進み、一般家庭や企業のみならず、公共施設や店舗など、様々な場所で利用されるようになった。無線 LAN はデータを電波で通信するため、有線のそれよりも盗聴などの脅威が大きくなる。そのため、データを暗号化し通信するセキュリティプロトコルの使用が必須である。

セキュリティプロトコルの中の一つである WEP は今現在も多くの無線 LAN ルータに搭載され利用されているが、これに対してはすでに多数の攻撃手法が提案され実装されている[1][2][3]。

しかし、今現在実装されている攻撃手法は、条件が限られるもので、解析が困難になるような状況をつくり出す。

だが、WEP に対してはすでに、IP パケットのみを用いて、比較的少ないパケットで、高速に WEP キーを導出するアルゴリズムが提案されている[4]。

そこで本研究では、このアルゴリズムを実装し、現実的な条件下で WEP キーが導出可能かどうかを、導出にかかったパケット数および実行時間を計測し検証する。

第 2 章　WEP

本章では、無線 LAN セキュリティプロトコルである WEP と、そのアルゴリズムに利用されているストリーム暗号 RC4 の概要及び、WEP の脆弱性について述べる。

2.1　無線 LAN

無線 LAN とは、LAN と呼ばれる、家庭内や施設内といった小さな規模で用いられるコンピュータネットワークのうち、電波の送受信によって通信を行うものである。単に無線 LAN と言った場合は、IEEE 802.11 規格に準拠した機器で構成されるネットワークを指すことが多く、本論文でもこの意味で用いる。

無線 LAN は通信ケーブルを物理的に接続する必要が無く、スマートフォンやタブレット等の普及もあり、一般家庭や企業のみならず、公共施設や店舗など、多くの場所で設置・利用されている。

図 1：無線 LAN

無線 LAN には、通信ケーブルを物理的に接続し通信を行う有線 LAN と比べて、以下のような特徴がある。

① 通信ケーブルの接続が不要
② 通信速度・安定性の面で劣る
③ 強固なセキュリティプロトコルが必要

①のようなメリットがある反面、②③のようなデメリットが存在する。②に関しては技術の進歩により、有線 LAN と比べても見劣りしないレベルの品質が出るようになってきているが、③に関しては無線で通信を行う限り、必ず必要な技術であり、プロトコルの選択によっては、セキュリティに重大な欠陥をもたらす場合がある。

2.2　無線 LAN のセキュリティプロトコル

無線 LAN 通信では、傍受が極めて容易に行えるため、送信されるパケットを暗号化し、内容を知られないようにする必要がある。その暗号化に利用されるのが各セキュリティプロトコルである。

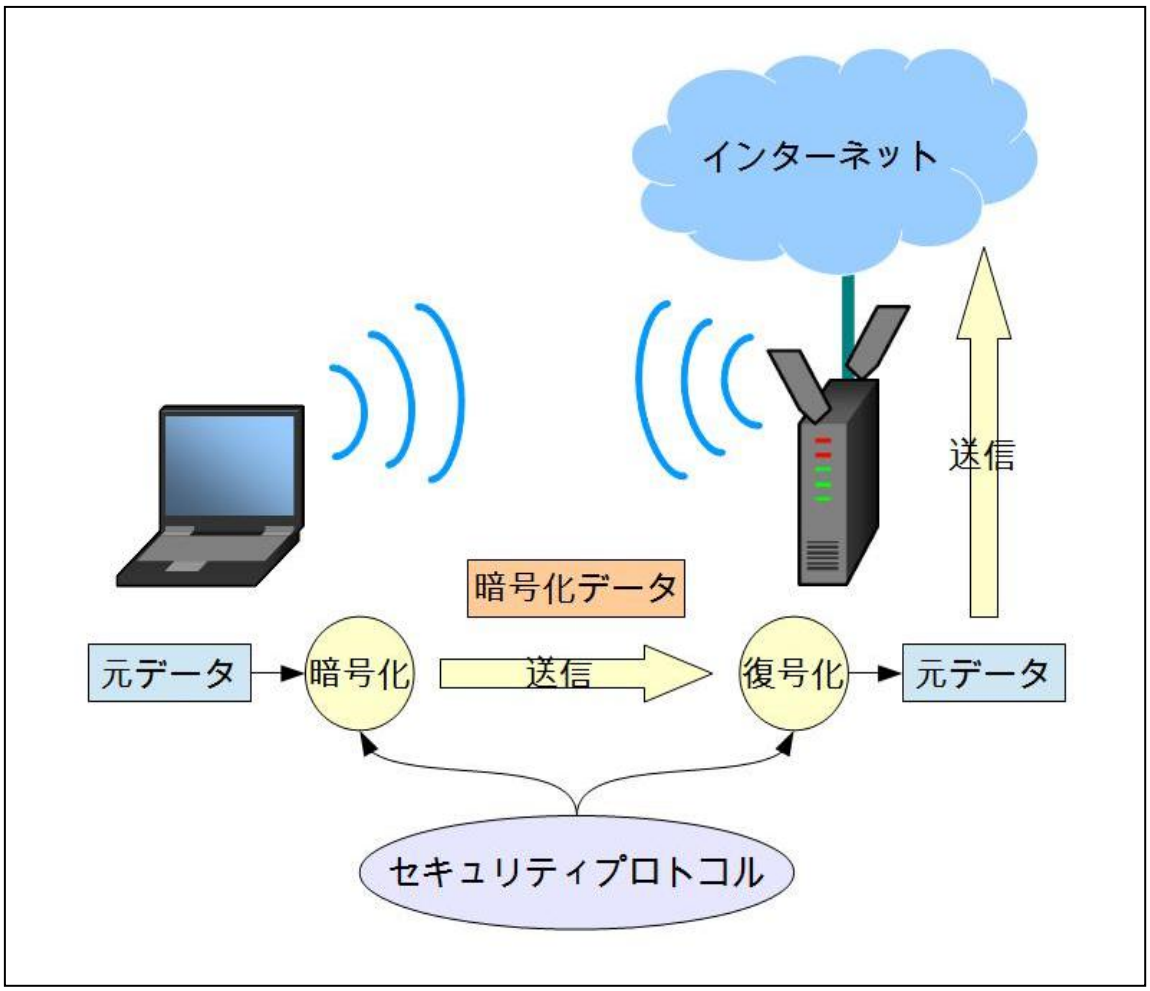

図 2　セキュリティプロトコル

この無線 LAN のセキュリティプロトコルには様々なものがあり、その中の一つである WEP は今現在も多くの無線 LAN ルータに搭載され、利用されている。

WEP の暗号化/復号化アルゴリズムには RC4 というストリーム暗号が用いられている。以下それぞれの概要を述べる。

2.3　RC4

RC4 とは、1987 年に Rivest によって提案されたストリーム暗号で、SSL や WEP など広く利用されている[3]。

RC4 の内部状態は置換配列 $S = \{S[0], S[1], \ldots, S[255]\}$ とそれに対する 2 つのポインタ i, j で構成される。なお、各要素は 1 バイトの変数である。

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。KSA では秘密鍵を用いて内部状態を撹拌し、PRGA でその内部状態からキーストリームと呼ばれる擬似乱数列を生成する。最後に平文とキーストリームの排他的論理和を取ることで暗号文を得る。復号側でも同様に、秘密鍵からキーストリームを生成し、暗号文との排他的論理和を取ることで復号を行う。

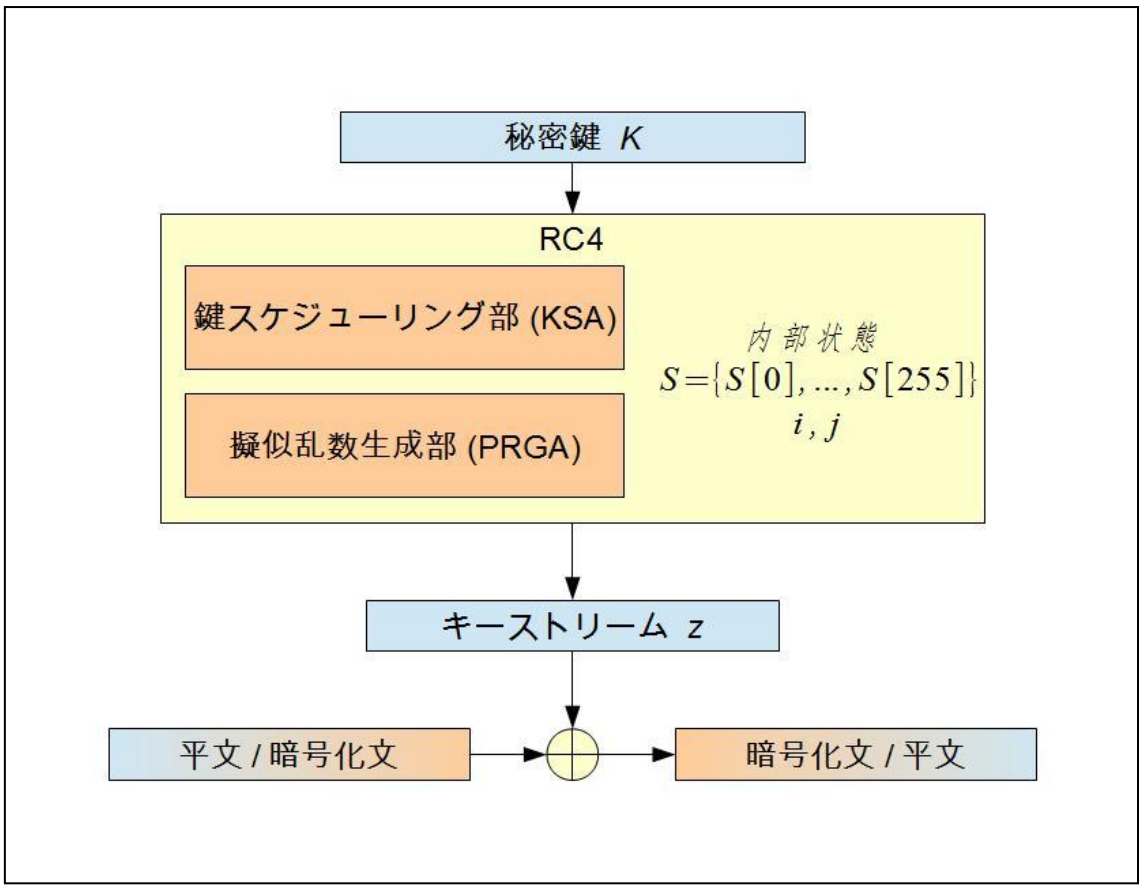

図 3　RC4

KSA と PRGA のアルゴリズムの概要を以下に示す。

RC4 KSA:
1: For $\forall x \in \{0, 1, \ldots, 255\}$
2: $S[x] \leftarrow x$
3: End for
4: $j \leftarrow 0$
5: For $\forall i \in \{0, 1, \ldots, 255\}$
6: $j \leftarrow j + S[i] + K[i \bmod K.length] \bmod 256$
7: $Swap(S[i], S[j])$
8: End for

図 4 KSA

RC4 PRGA:
1: $i \leftarrow 0$
2: $j \leftarrow 0$
3: Loop
4: $i \leftarrow i + 1 \bmod 256$
5: $j \leftarrow j + S[i] \bmod 256$
6: $Swap(S[i], S[j])$
7: $z \leftarrow S[S[i] + S[j]]$
8: End loop

図 5 PRGA

このように、KSA、PRGA はともに非常に簡潔な形で記述できる。なお、図中の K は秘密鍵、z は出力するキーストリームである。

2.4　WEP

WEP とは、有線同等機密(Wired Equivalent Privacy）の略で、IEEE 802.11b で規定されている無線 LAN セキュリティプロトコルである[5]。

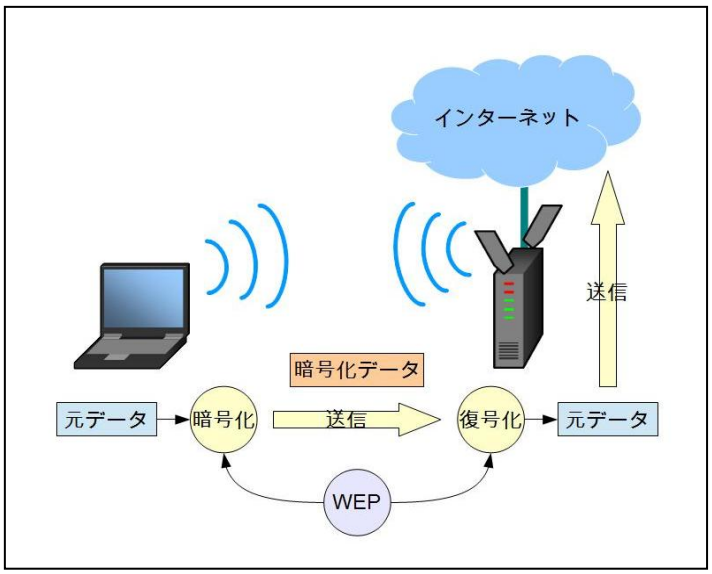

図 6　WEP

前述のように、その暗号化/復号化アルゴリズムには RC4 を用いる。また、WEP には秘密鍵の長さが異なる 2 つの種類「WEP64」「WEP128」が存在するが、本研究では、より多く利用されている WEP128 を対象とする。

WEP ではパケット毎に異なる鍵(パケット鍵)を生成し、これを RC4 の秘密鍵として利用する。具体的には、パケット鍵 K は、13 バイトの秘密鍵(WEP キーと呼ぶ) K' と、3 バイトの初期化ベクトル IV を用いて、

$$K[i] = \begin{cases} IV[i] & i = 0, 1, 2 \\ K'[i-3] & i = 3, 4, 5, \ldots, 15 \end{cases} \quad (1)$$

という形でビット結合して生成する。

パケット鍵 K によって RC4 で暗号化された暗号化文は、初期化ベクトル IV を付与して送信され、受信側は受信した IV と WEP キー K' を用いてパケット鍵 K を生成し、復号化を行う。

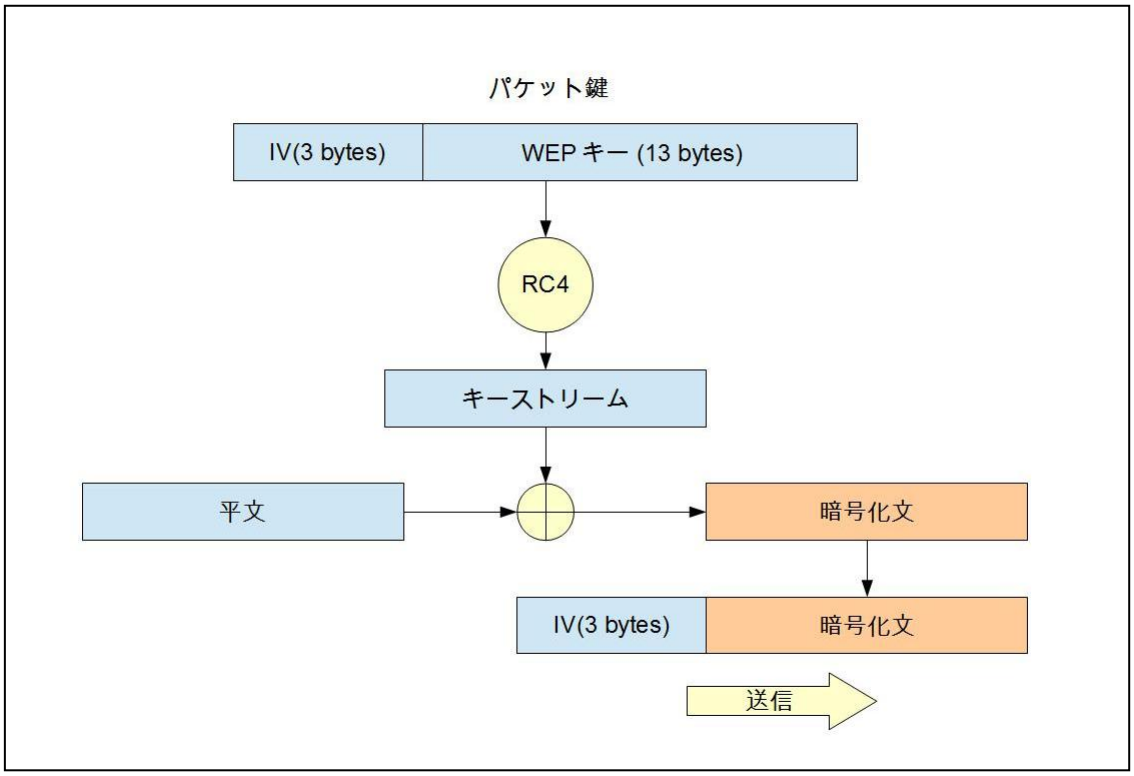

図 7　WEP の暗号化

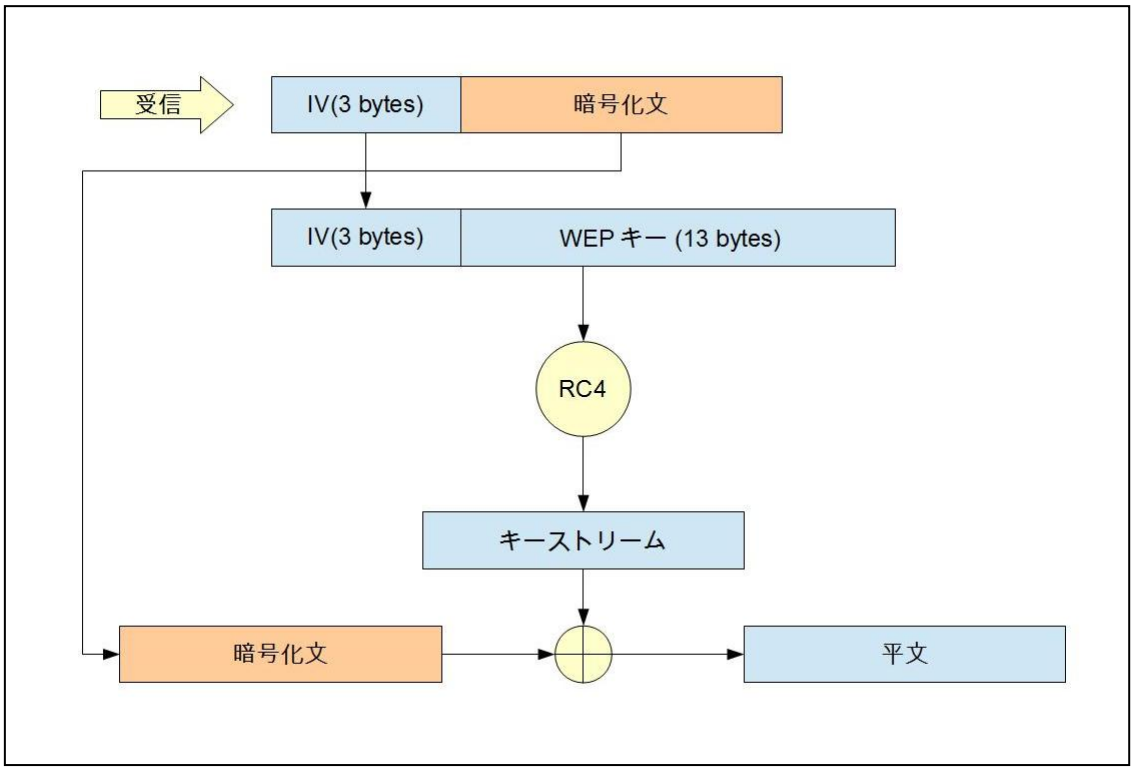

図 8　WEP の復号化

2.7　WEP の脆弱性と従来の攻撃手法

WEP は提案されて以来、様々な脆弱性を指摘されており、多数の攻撃手法が提案・実装されているが、この攻撃手法のうち多くは、条件が限られてしまうもので、解析が困難となる状況をつくりだすことができるものである。

以下に代表的な攻撃手法を 2 つあげる。

2.7.1　FMS 攻撃

代表的な攻撃手法に FMS 攻撃がある[1]。FMS 攻撃は 2001 年に Fluhrer らによって提案された、初期化ベクトル *IV* の脆弱性を利用した攻撃手法である。

IV 空間の中には、その *IV* を利用してキーストリームを生成したとき、観測したキーストリームの先頭に WEP キーの情報が大きく漏洩してしまうような値が存在し、その値は *weakIV* と呼ばれている。FMS 攻撃はこの *weakIV* で暗号化されたパケットを大量に収集し解析することで、WEP キーの導出を試みる手法である。

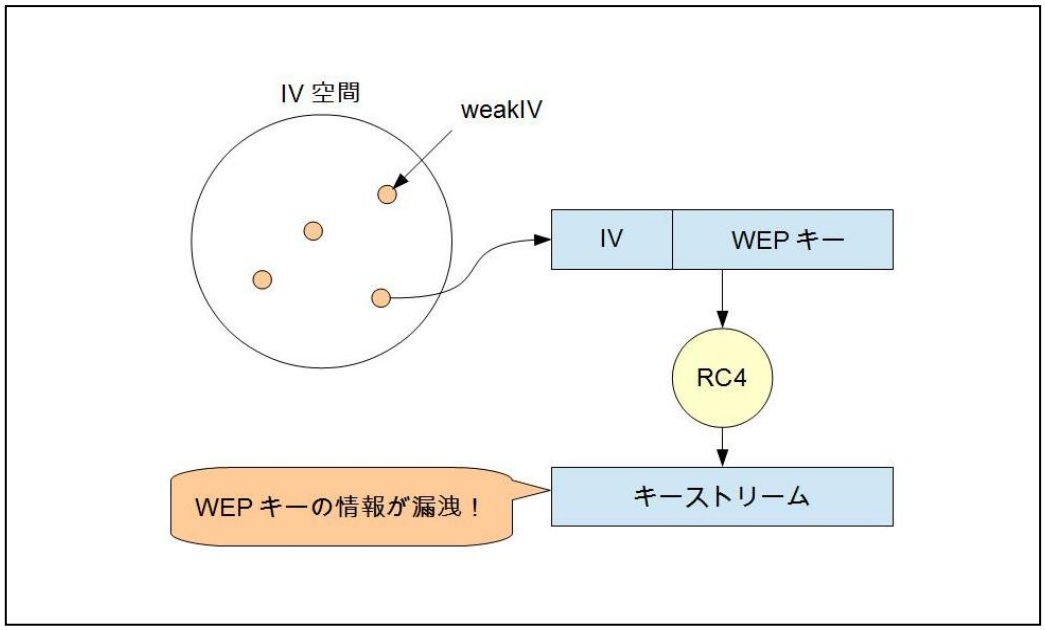

図 9　FMS 攻撃

しかし、この手法は *weakIV* によって暗号化されたパケットのみに実行可能な攻撃手法であるため、 *weakIV* を有するパケットを大量に入手するまでに多くのパケットを観測し続ける必要がある。また、 *weakIV* を利用しないことによって、この攻撃に対して耐性を持たせることが可能である。

2.7.2　PTW 攻撃

次に代表的な攻撃手法として PTW 攻撃がある[2]。PTW 攻撃は 2007 年に Tews らによって提案された、 *IV* に依らず実行可能な攻撃手法である。

PTW 攻撃は、Klein によって発表された、キーストリームから鍵を推定するという攻撃手法(Klein 攻撃)[3]を発展させたものである。WEP キーの各バイトを並列に回復することが可能で、鍵導出を非常に高速に行える手法である。この攻撃手法を実行する場合は、平文の大部分が類推することができる ARP パケットを大量に収集し利用する。

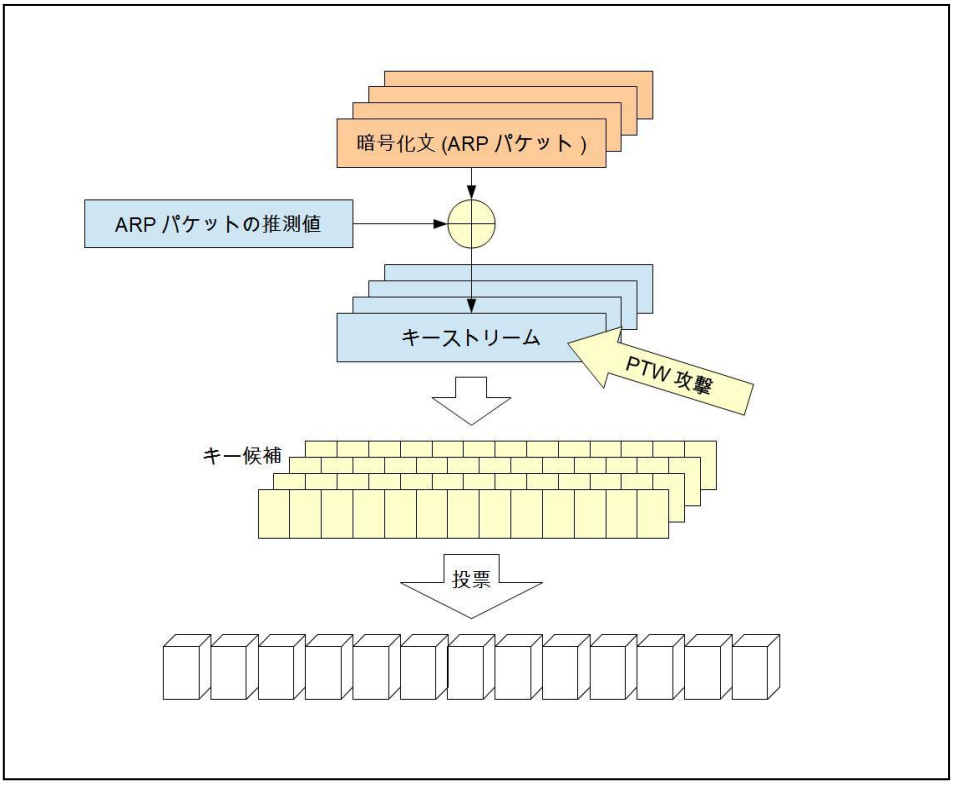

図 10　PTW 攻撃

実際には、ARP パケットは通常の通信で用いられることは少なく、大量のパケットを集めることは困難である。そのため、ARP インジェクション攻撃[6]という不正なアクセスを、アクセスポイントに対して行い、ARP パケットを強制的に送信させる必要がある。

しかし、ARP インジェクションは容易に検知可能であるため、この不正アクセスを防ぐことで攻撃自体を防ぐことが可能である。

2.8 今回実装する攻撃手法

すでに WEP に関しては、IP パケットのみを用いて、比較的少ないパケットで、高速に WEP キーを導出するアルゴリズムが、寺村らによって提案されている[4]。

この攻撃手法は PTW 攻撃と同じく、Klein 攻撃を発展させた攻撃手法であるが、攻撃には ARP パケットではなく、通常の通信で用いられる IP パケットを利用する。

通信を行う限り、IP パケットの送受信は避けられず、IP パケットのみを収集して攻撃を行うこの手法は、防ぐことができない。SSL と併用することや、MAC アドレスフィルタリングを行うことで、通信の盗聴や回線の不正利用を防ぐことはできるが、WEP キー漏洩に対する根本解決にはならない。

しかし、IP パケットは ARP パケットと違い、13,14 バイト目が固定ではなく、また容易に推定することも困難であるため、後述する理由で、PTW 攻撃で用いられる関数では、WEP キーを導出することができない。

次章ではこのアルゴリズムの実際の実装方法を、その問題に寺村らがどのように対処したか、その手法とあわせて述べる。

第 3 章　実装方法

本章では、WEP キーを IP パケットから導出する手法[4]の、具体的な実装方法について述べる。

3.1　Klein 攻撃

Klein は RC4 に対する解析を行い、RC4 の内部状態とキーストリームとの間の相関を見つけ、そこからキーストリームから秘密鍵を確率的に導出する関数(Klein 攻撃関数)を導いた[3]。その式を以下に示す。ただし、K はパケット鍵、z はキーストリーム、 S は KSA の内部状態である。

$$f_{Klein}(K[0], K[1], \dots, K[x-1], z_{x-1}) = S_x^{-1}[x - z_{x-1}] - (j_x + S_x[x]) \quad (2)$$

$$P(f_{Klein} = K[x]) \approx \frac{1.36}{256} \quad (3)$$

この関数は、$1.36/256$ という高い確率でパケット鍵の値と一致するとしている。そのため、多くのパケットを収集し投票を行うことで、逐次的に鍵を導出することができる。

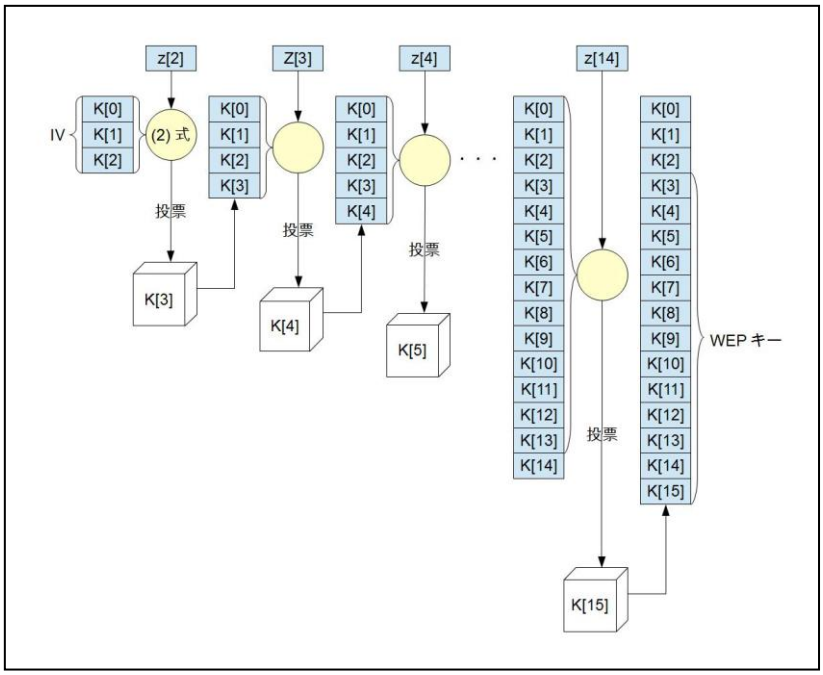

図 11　Klein 攻撃

3.2　IP パケットを用いる上での問題点

ARP パケットは先頭 15 バイトが容易に推測可能な固定値を取るため、この Klein 攻撃を用いれば逐次的にすべての WEP キーを導出する事ができる。

一方、IP パケットは 13,14 バイト目が固定ではなく、推測も困難であるため、この手法では、キーストリームの 13,14 バイト目(z_{12}, z_{13})が導出できず、その結果、パケット鍵の 14,15,16 バイト目、つまり $K[13], K[14], K[15]$ $(= K'[10], K'[11], K'[12])$ の値が導出できない。

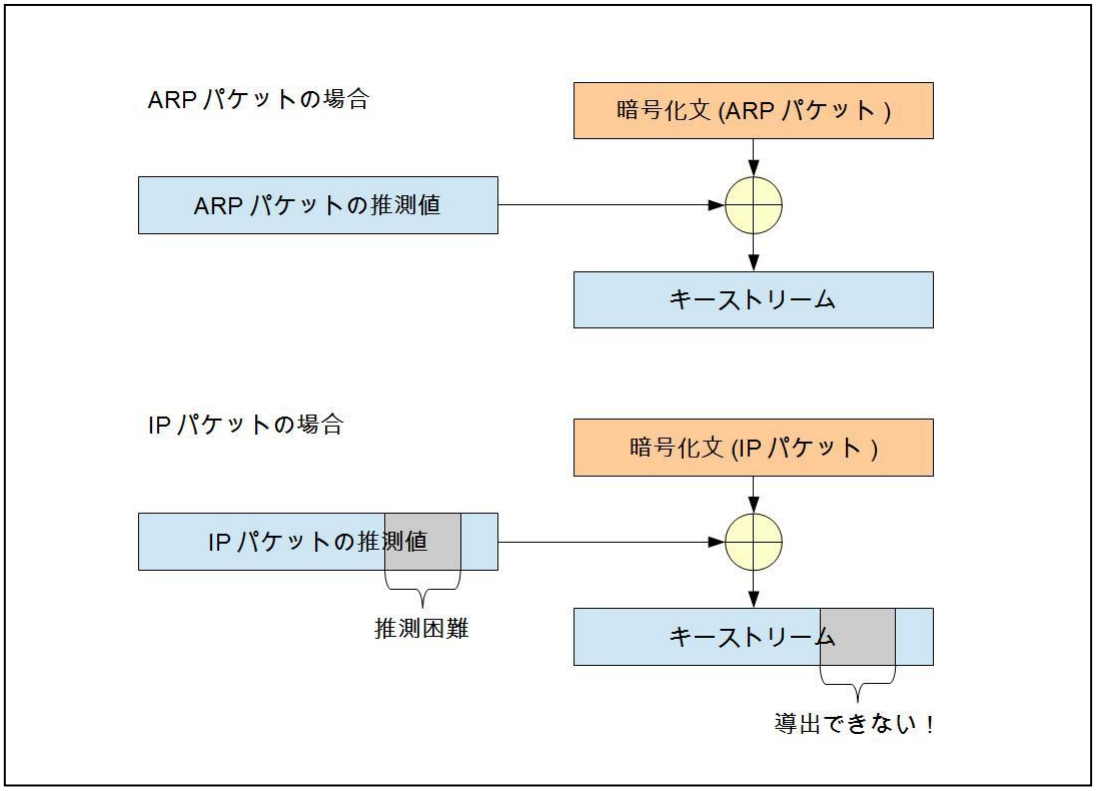

図 12　IP パケットでのキーストリーム導出

この 3 バイト($K[13], K[14], K[15]$)の全数探索を行えば WEP キー導出は可能であるが、IP パケットの 15,16,18 バイト目が固定値であることを利用することで、その問題を緩和することができる。

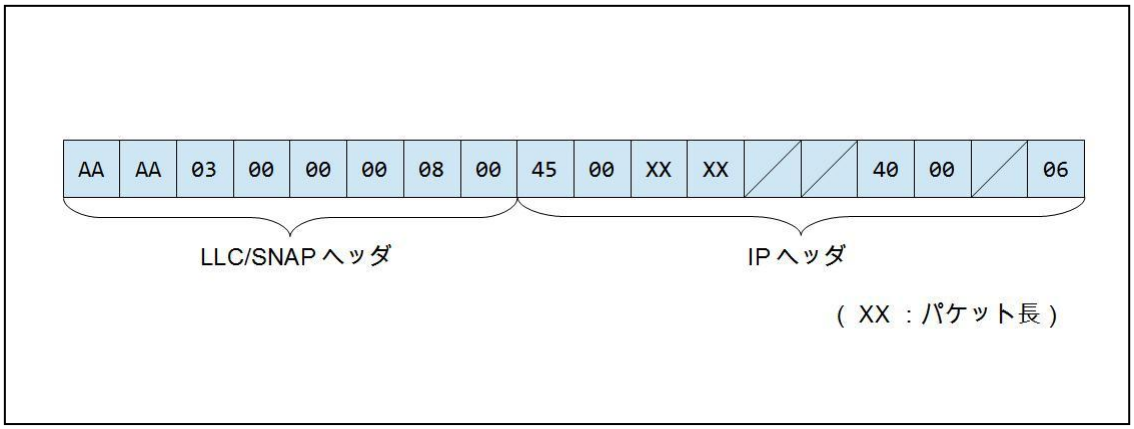

図 13　IP パケットの固定値及び推測可能値

そのためにまず、文献[2]のアイディアを拡張して用いることで、Klein 攻撃関数を以下の式に変形する。

$$\sum_{l=i}^{x} K[l] \approx S_{i-1}^{-1}[x - z_{x-1}] - (j_{i-1} + \sum_{l=i}^{x} S_{i-1}[l]) \tag{4}$$

この式を用いることで、$K[13] + K[14] + K[15](= \sigma)$の値を確率的に導出できる。この値を用いれば$K[13], K[14]$の 2 バイトのみの全数探索で WEP キーを導出することができるようになる。

3.3　実装方法

本節ではプログラム全体の構成と、各モジュールの具体的な動きについて述べる。

本プログラムの全体の構成は以下のようになっている。

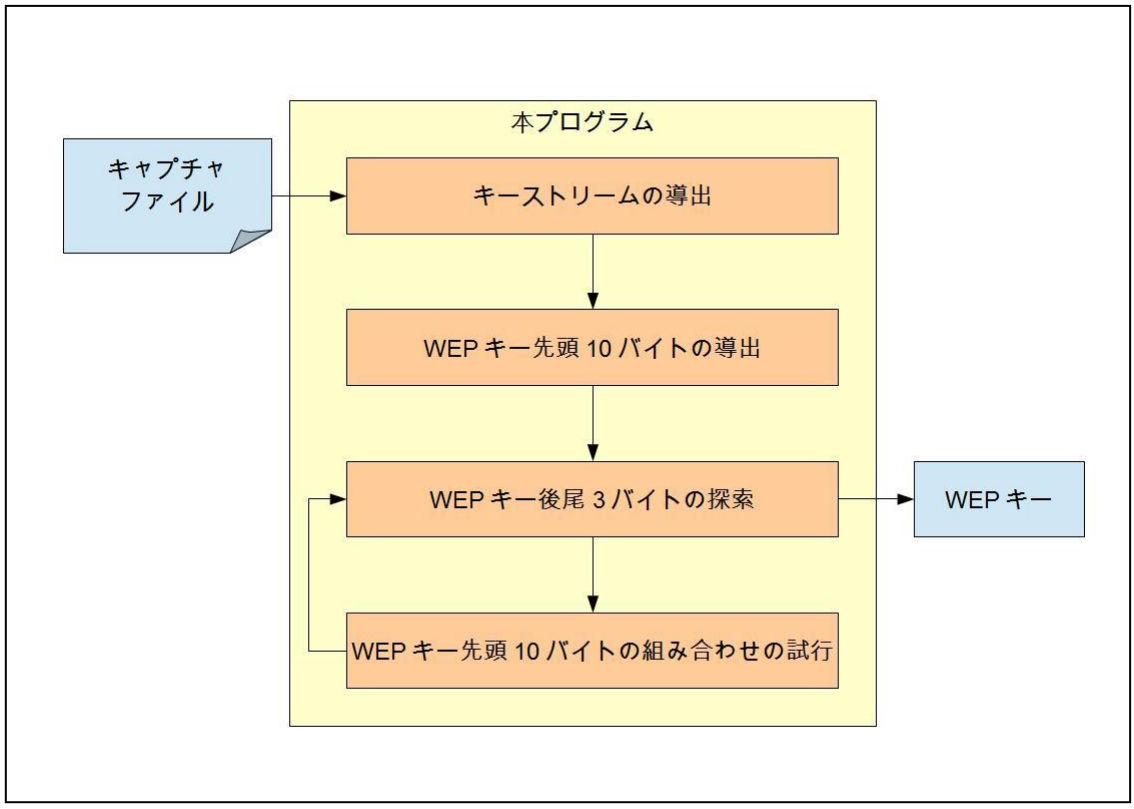

図 14　本プログラムの全体の構成

以下、それぞれのモジュールの具体的な動きを示す。

3.3.1　キーストリームの導出

このモジュールでは、IP パケットの固定あるいは推測可能値を用いて、キーストリームの導出を行う。

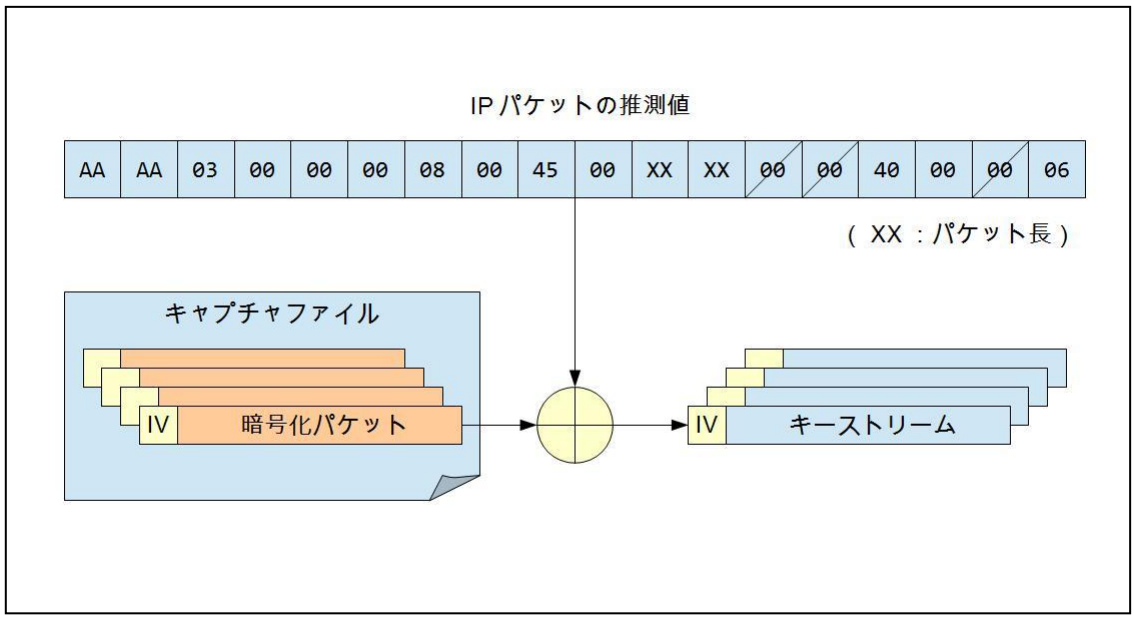

図 15　キーストリームの導出

キャプチャファイルの暗号化パケットと、IP パケットの推測値との排他的論理和を取り、キーストリームを取得する。なお、IP パケットの推測値の 13,14 バイト目はパケット長であるが、WEP で暗号化を行った場合、データ長は変わらないため、取得したパケットの長さを取得することで、そのまま値を利用できる。

最後にキーストリームの先頭に IV を付与してメモリに格納し、終了する。

3.3.2　WEP キー先頭 10 バイトの導出

このモジュールでは、3.3.1 のモジュールで取得したキーストリームに対して(2)式を適用し、WEP キー先頭 10 バイトを投票で決定する。

$$f_{Klein}(K[0], K[1], \ldots, K[x-1], z_{x-1}) = S_x^{-1}[x - z_{x-1}] - (j_x + S_x[x]) \qquad (2)$$

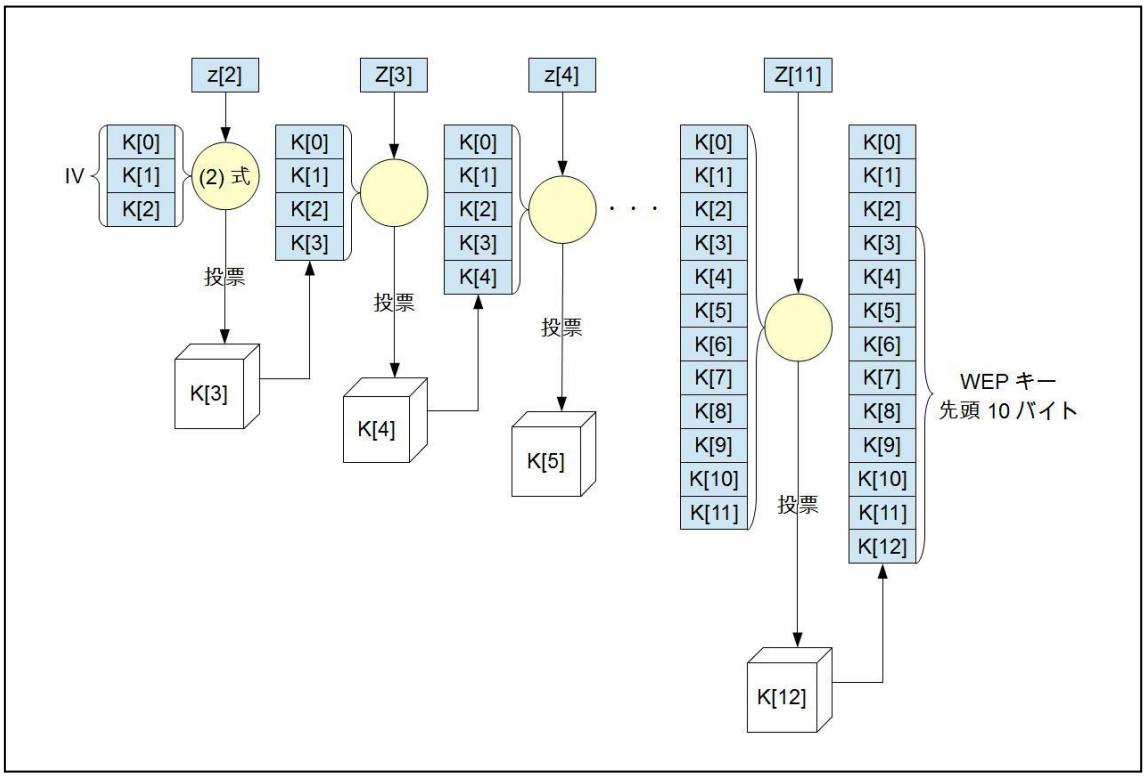

図 16　WEP キー先頭 10 バイトの導出

はじめに $x = 3$ として、すべてのキーストリーム z_2 に対し(2)式を計算する。その結果、適用したキーストリームの数だけキー候補が得られるため、その中で最も多く出現したキーを$K[3]$とする。同様の操作を $K[12]$ が得られるまで逐次的に行う。

最後に投票テーブルの得票数の多い順にキーテーブルをソートして終了する。

3.3.3 WEP キー後尾 3 バイトの探索

このモジュールでは、(4)式を用いて後尾 3 バイトの和 $\sigma\ (= K[13] + K[14] + K[15])$ を導出し、$K[13], K[14]$ の全数探索を行い、マッチングを試みる。

はじめに、(4)式を $x = 15,16,18,\ i = 13$ でそれぞれのキーストリームに対して適用し、$K[13] + K[14] + K[15])$ の値を計算し投票。得票数一位の値を σ とする。

例えば$x = 16,\ i = 13$ で適用すると以下のようになる。

$$\sum_{l=i}^{x} K[l] \approx S_{i-1}^{-1}[x - z_{x-1}] - \left(j_{i-1} + \sum_{l=i}^{x} S_{i-1}[l]\right)$$

$$K[13] + K[14] + K[15] + K[16] \approx S_{12}^{-1}[16 - z_{15}] - (j_{12} + S_{12}[13] + S_{12}[14] + S_{12}[15] + S_{12}[16])$$

パケット鍵は 16 バイトであるため、$K[16]$ には $K[0]$ が適用されるが、$K[0] = IV[0]$ であるため、両辺から $IV[0]$ を減ずることで、以下の式になる。

$$K[13] + K[14] + K[15] \approx S_{12}^{-1}[16 - z_{15}] - (j_{12} + S_{12}[13] + S_{12}[14] + S_{12}[15] + S_{12}[16]) - IV[0]$$

同様に $x = 15,16,18,\ i = 13$ で適用する。

次に σ を用いて $K[13], K[14]$ の全数探索を行い、マッチングを試みる。

$K[13]$	00	01	02	・・・	ff	00	01	・・・	ff
$K[14]$	00	00	00	・・・	00	01	01	・・・	ff
$K[15]$	$\sigma - (K[13] + K[14])$								

図 17　WEP キー後尾 3 バイトの探索

マッチングが成功した場合、その時の WEP キーを返して本プログラムは終了する。最後までマッチングが成功しなかった場合は 3.3.4 のモジュールを実行する。

3.3.4　WEP キー先頭 10 バイトの組み合わせの試行

このモジュールでは、投票テーブルを用いて、WEP キーの先頭 10 バイトを変更し、再度 3.3.3 のモジュールでマッチングを試みる。

得票率一位のキーとの得票数の差が小さいものから順に、キーを変更しながらマッチングを行う。

なお、Klein 攻撃は逐次的に値を求めているため、$K[x-1]$の変更によって、$K[x]$に誤差が発生するが、$K[x-1]$で発生した差を$K[x]$の値から減ずることで、誤差を修正することができる。([4], p.168)

以下に実行例を示す。

	キー候補(得票数)			
$K[3]$	31(100)	32(50)	33(50)	
$K[4]$	41(100)	42(99)	43(97)	
$K[5]$	51(100)	52(50)	53(50)	
$K[6]$	61(100)	62(98)	63(50)	
$K[7]$	71(100)	72(50)	73(50)	・・・
$K[8]$	81(100)	82(50)	83(50)	
$K[9]$	91(100)	92(50)	93(50)	
$K[10]$	A1(100)	A2(50)	A3(50)	
$K[11]$	B1(100)	B2(50)	B3(50)	
$K[12]$	C1(100)	C2(50)	C3(50)	

図 18　キーテーブル・投票テーブルの例

例として、図 18 のようなキーテーブル・投票テーブルが得られた場合を考える。探索の前に、探索したキーを記憶しておくための $C = \{C[3], \dots, C[12]\}$ という配列を生成し、値をすべて 0 で初期化する。

最初に、第一位との得票数差が $100 - 99 = 1$ と、最も小さい $K[4]$ のキーを変更する。$K[4]$ に変更が生じたため $K[5]$ の値を調整し、これを 1 回目のキーとする。$K[4]$ のキー候補をひとつ探索したため、$C[4]$ をインクリメントする。$(C[4] = 1)$

次に、探索していないキーのうち、第一位との得票数差が最も小さいものを探す。今回の場合は $100 - 98 = 2$ で、$K[6]$ が最も小さいため、このキーを変更する。$K[6]$ に変更が生じたため $K[7]$ の値を調整し、2 回目のキーとする。

またこれ以降は、C の値が 0 でないキー(探索したキー)が存在するため、$K[6]$ の変更はそのままに、探索したことのあるすべてのキーをたどる。今回は、$C[4] = 1$ であるため、$K[4]$ の値を変更する。$K[4]$ に変更が生じたため $K[5]$ の値を調整し、これを 3 回目のキーとする。

同様に操作を続けると、次のような順でキーをたどることとなる。なお、0 回目の部分は 3.3.2 終了時点でのキーである。

	$K[3]$	$K[4]$	$K[5]$	$K[6]$	$K[7]$	$K[8]$	$K[9]$	$K[10]$	$K[11]$	$K[12]$
(0 回目)	31	41	51	61	71	81	91	A1	B1	C1
1 回目	31	42	50	61	71	81	91	A1	B1	C1
2 回目	31	41	51	62	70	81	91	A1	B1	C1
3 回目	31	42	50	62	70	81	91	A1	B1	C1
4 回目	31	43	49	61	71	81	91	A1	B1	C1
・・・										

図 19　キー候補をたどる順の例

第 4 章　検証結果と考察

本章では、本プログラムが現実的な条件下で WEP キーを導出可能であるかどうか、その検証の結果を示す。

なお、実験は以下の条件で行う。

① WEP キーは [00: 11: 22: 33: 44: 55: 66: 77: 88: 99: AA: BB: CC] に固定
② タイムアウト時間を 30 秒に設定

現実的な条件下で導出可能かどうかの検証であるため、検証用のパケットはプログラムで生成するのではなく、実際にアクセスポイントを起動し、通信を行うことで生成したい。WEP キーを変更すると、アクセスポイントの再起動や、再接続などに時間がかかってしまうため、①の条件を指定する。
次に②についてだが、まず、WEP キーが見つからない場合の、プログラム終了の制御に関しては以下の 2 つが考えられる。

- 試行する WEP キーの上限個数を設定する。(KEY_LIMIT)
- タイムアウト時間を指定する。(TIME_LIMIT)

本プログラムはどちらも指定することが可能だが、今回はタイムアウト時間を 30 秒、上限個数を無限大とした。
以上の条件で計 50 回の試行を行った。

また、WEP キー解析には以下の PC を使用した。

OS	Windows 7 Home Premium 32 bit (6.1, ビルド 7601)
CPU	Intel® Core™2 Duo CPU E7500 (2.93GHz × 2)
メモリ	3072MB RAM

以下に検証の結果を示す。

4.1　パケット数

本節では、現実的に収集できる範囲のパケット数で、WEP キー導出が可能かどうかを検証する。

実験結果の、パケット数と成功確率の関係を以下のグラフに示す。

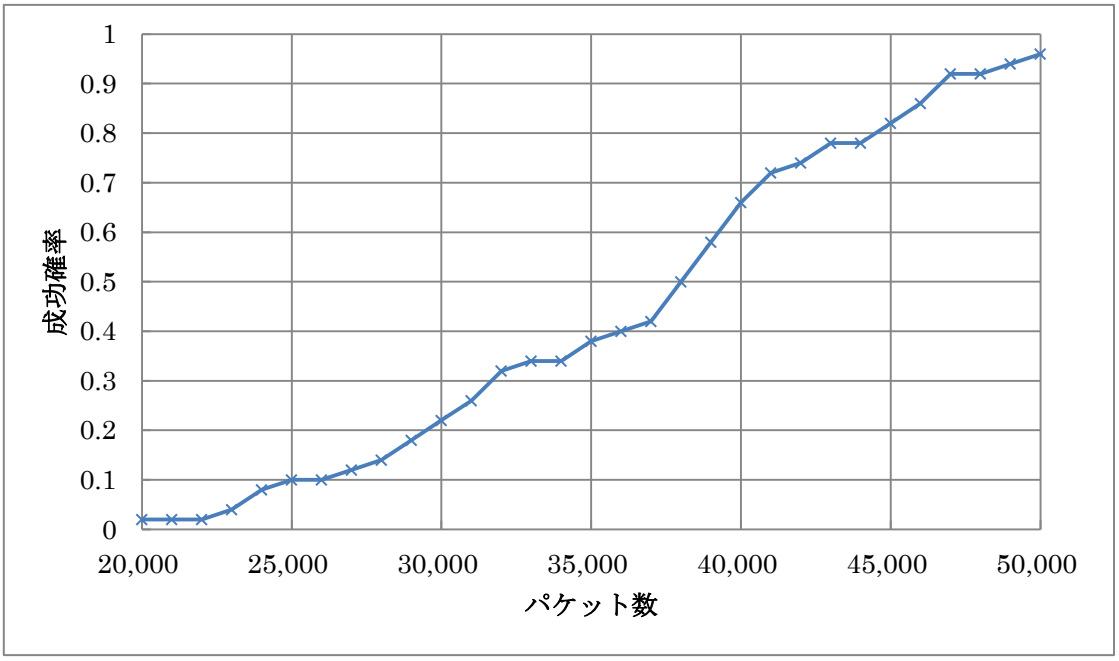

図 20　パケット数増加に伴う成功確率の変化

今回実装した攻撃手法は、所有者が対象のアクセスポイントを使って、通信を行っている場合に、送受信されるパケットを収集することで、はじめて攻撃を行うことができる。つまり、まったくそのアクセスポイントが使われていない場合、この攻撃手法では WEP キーを導出することができない。そのため、あくまで対象アクセスポイントが利用されていることを前提で考察を行う。

パケット収集時間に関しては、利用状況に応じてかなりばらつきが出るが、例えば、成功確率がほぼ 100%となる 50000 パケットについては、早ければ数十秒から数分、長くても 10 分程度の時間で収集可能である。つまり長くても 10 分程度あれば、WEP キー導出ができると考える事ができる。

結果として、対象アクセスポイントが利用されていることを前提とするならば、この手法は、現実的に収集できる範囲のパケット数で、WEP キーを導出可能であるといえる。

4.2　実行時間

本節では、パケット数増加に伴う実行時間の変化について考察する。

なお、今回はタイムアウト時間を 30 秒と設定しており、人が十分に待つことのできる時間であるため、実行時間に関しては、現実的な条件下という要件を満たしているものとする。

実験結果の、パケット数と実行時間の関係を以下のグラフに示す。

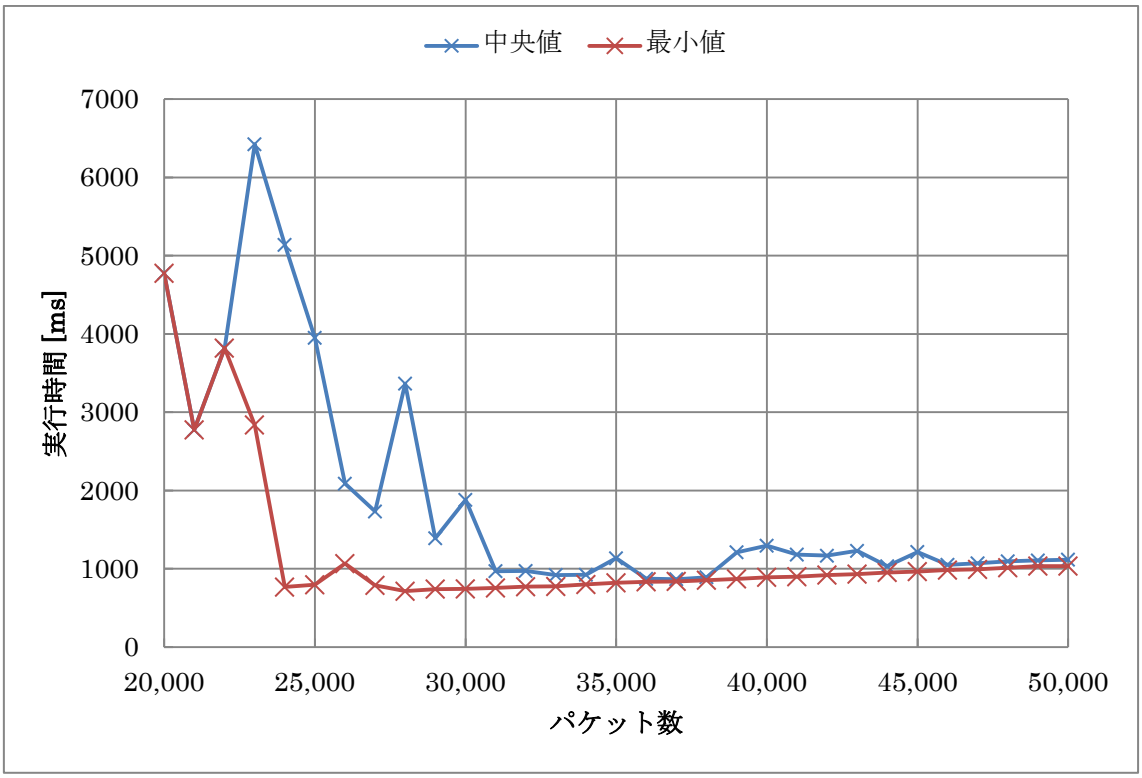

図 21　パケット数増加に伴う実行時間の変化

タイムアウト時間を設ける必要があるため、成功した試行のみのデータであり、分母はパケット数によって異なる。また、測定の結果、外れ値が出たため、平均値ではなく中央値を提示する。

パケット数が少ない段階では安定せず、値が上下しているが、パケット数が多くなると安定して速い速度で実行できている。特にパケット数が 45000 を超えると、最小値とほぼ同じ値が安定して出ていることから、第一位のキーでマッチングが成功し、3.3.4 のモジュールを実行せずに終了しているものと考えられる。なお、その場合はパケット数に比例した一定時間で完了する。

第 5 章　今後の課題

本章では、今回、時間の都合上実装できなかった仕様や、実施できなかった実験等について、今後の課題として述べる。

5.1　パケット収集と WEP キー解析を並行して実行するように改良

パケットを収集しながら WEP キー解析を行うことができれば、導出までの過程を単純化・高速化することができる。

5.2　複数の WEP キーに対して同様に導出が行えるかを検証

WEP キーを変更しての検証が、ほぼ行えていないため、課題としてあげられる。なお、多数の WEP キーを設定して実験を行う場合、プログラムで暗号化パケットを生成する方が効率的である。

5.3　試行回数を増加

試行回数が 50 回と少ないため、大量の検証用パケットを入手し、実験を行う必要がある。その場合、5.2 と同様、プログラムで暗号化パケットを生成する方法が効率的である。

5.4　パケット数・実行時間の定量的な評価

現実的な条件下で導出が行える、という抽象的な評価しかできていないため、統計の分野も踏まえて定量的な評価を行う必要がある。

謝辞

本研究を進めるにあたり、最後まで熱心に御指導して頂きました田中章司郎教授には心より御礼申し上げます。

同研究室の皆様にも、数々の御協力と御助言を頂きましたこと、厚く御礼申し上げます。

なお、本論文、本研究で作成したプログラム及び、データ、並びに関連する発表資料等の全ての知的財産権を本研究の指導教官の田中章司郎教授に譲渡致します。

引用・参考文献

[1] S. Fluhrer, I. Mntin, and A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4”, SAC2001, Lecture Notes In Computer Science, Vol. 2259, pp.1-24, Springer-Verlag, 2001.

[2] E. Tews, R. Weinmann, and A. Pyshkin, “Breaking 104 bit WEP in less than 60 seconds”, WISA2007, Lecture Notes in Computer Science, Vol.4867, pp.188-202, Springer-Verlag, 2008.

[3] A. Klein, “Attacks on the RC4 stream cipher”, Designs, Codes and Cryptography, Vol.48, no.3, pp.269-286, Sep.2008.

[4] R. Teramura, Y. Asakura, T. Ohigashi, H. Kuwakado, and M. Morii, “Fast WEP-Key Recovery Attack Using Only Encrypted IP Packet”, IEICE Trans. on Fundamentals of Electronics, Comm. And Computer Science, Vol. E93-A, no.1, pp. 164-171, Jan. 2010

[5] IEEE Computer Society, “Wireless lan medium access control (MAC) and physical layer (PHY) specifications” IEEE std 802.11, 1999

[6] The Aircrack-NG Team, “Aircrack-ng” http://www.aircrack-ng.org/