# T12: ここまできた！無線LANセキュリティ

**進藤　資訓**
**ファイブ・フロント(株)**
**Chief Technology Officer**
**mshindo@fivefront.com**

## 無線LANのセキュリティー

- SSIDの秘匿
- MACアドレスの制限
- WEP
- 802.1X
- WPA
- WPA2 / 802.11i
- …

# WEP (Wired Equivalent Privacy)

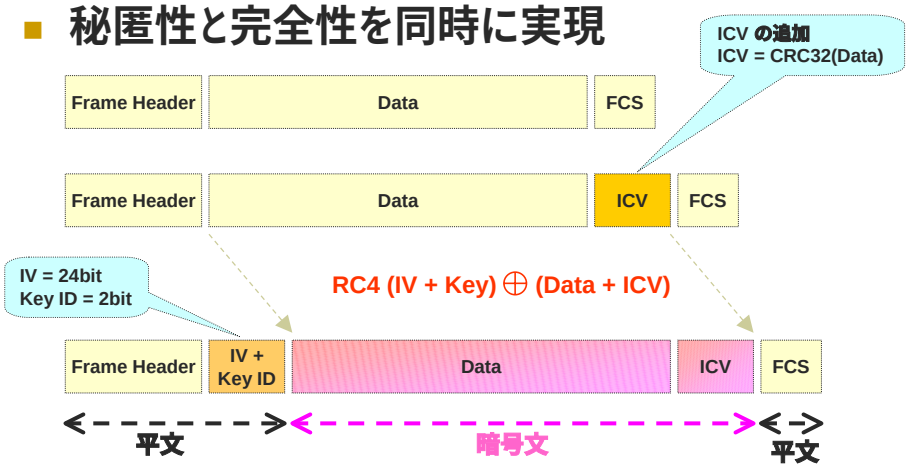
- **何をしている？**
  - **秘匿性** (Confidentiality)
  - **完全性** (Integrity)
  - **認証** (Authentication)
- **実際は？**
  - What on Earth does this Protect?



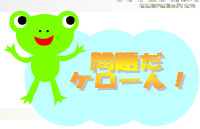


# WEP 処理

- **秘匿性と完全性を同時に実現**



# これ、ほんと？？




WEPには、同じIVで暗号化したフレームを幾つか集めると暗号鍵を解読できるという弱点がある。

（A誌、2003年9月）

ところが、ここに落とし穴があった。IVは24ビットしかなく、連続して通信を行っていると早くて数時間で1巡してしまう。また、無線LANで送信されるパケットの最初の部分はつねに同じパターンが使われているのである。つまり、IVが何巡かするまでパケットを監視しつづけていれば、暗号鍵が解読できてしまうのだ。

（C誌、2004年9月）

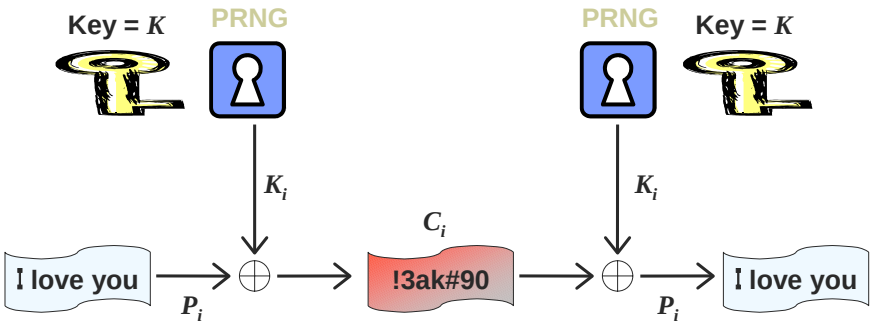




# 誤解と現実

- いわゆるIVの衝突（コリジョン）に関する誤解
- ストーリーとしては分かりやすい
  - i.e.「24ビットは短すぎたので、WPAでは48ビットにしたのさ！　だからWPAは安全なのよ。」
- 実際は、
  - IVが衝突しても壊滅的（e.g. WEP鍵を解き明かす）なことが起こるわけではない
  - ただ、衝突はできる限り起こらないほうが望ましい

## Stream Cipher




**Property 1:** If $C_i = P_i \oplus K_i$ Then $P_i \oplus C_i = K_i$

**Property 2:** If $C_1 = P_1 \oplus K_a$ and $C_2 = P_2 \oplus K_a$
Then $C_1 \oplus C_2 = (P_1 \oplus K_a) \oplus (P_2 \oplus K_a) = P_1 \oplus P_2$





## 同じIVを使うと何が起こるか？

- WEP鍵は変わらない（前提）
- 同じIVを使うと、同じキーストリーム(KS)が生成される
- $(M_1 \oplus KS) \oplus (M_2 \oplus KS) = M_1 \oplus M_2$
  - $M_1$ がわかるわけでもなければ$M_2$ がわかるわけでもない
  - ましてやWEPキーがわかるわけではない
  - 多少、$M_1$や$M_2$に関する情報は得られる

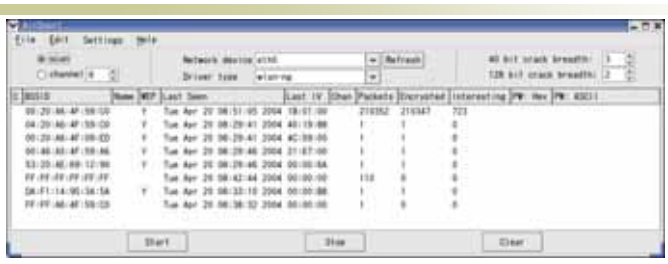
# 本当の脅威 ～ FMS 攻撃 ～

- S. Fluhrer, I. Mantin, A. Shamir, *Aug. 2001*
- Key Recovery 攻撃
- 条件
  - 生成される RC4 stream の最初のバイトが判っていて、
  - IV がある種の条件を満たす場合、Key Byte を5%の確率でguessできる
    - 代表的 Weak IV: (*B*+3, 0xff, *N*)
- key の長さに比例しかしない！
- 4,000,000 ～ 6,000,000 パケットで 40bit WEP を解読できる
- 更なる最適化で 1,000,000 パケット程度で解読可能
  - 5Mbps, 200 bytes/packet で、3125 秒





# WEP Cracking Tools



AirSnort
http://airsnort.shmoo.com

bsd-airtools (dwepcrack)
http://dachb0den.com/projects/bsd-airtools.html

## 多くの人は・・・

- 多くの人は、以下の二つの問題：
  - IVが比較的簡単に一巡してしまう
  - Weak IVを使って暗号化されたフレームを沢山集めるとWEP鍵をリカバーできてしまう
- をゴッチャに理解している！





## RC4 は脆弱か？

- ***そんなことはない！***
  - もしそうなら SSL/TLS だって WPA (TKIP) だって危ないことになる！
- 若干の脆弱性はあるものの、一般的にはほとんど問題ない
  - WEP が脆弱なのは RC4 の使い方を少々間違えたからである
- RC4 を正しく使えば安全 (と今日の時点では考えられている)
  - セッション毎に (相関関係の無いように) キーを変える
    - 例) SSL / TLS
  - 最初の数百バイト (例えば 256 バイト) を捨てる
    - 例) GTK over EAPOL

## さらなる最適化 ～ Korek 攻撃 ～

- PoC “chopper” posted by Korek in *Aug. 2004*
- **通称“Korek攻撃“と呼ばれている**
- **FMS攻撃をさらに一般化した統計的** Key Recover **攻撃**
- **すぐに他のソフトウェアに実装された**
  - Aircrack
  - WepLab
  - Airsnort
  - …





## Korek攻撃の威力

- **FMS攻撃では数百万パケット必要だったが、Korek攻撃では**10～20**万パケットで**WEP**鍵をリカバー可能！**
- **ツールも進化しており、**Active Attack**をするものが出てきた** (e.g. aircrack**の**aireplay)
- 10**分程度で**104bit WEP**を破ることができる！**
  - 802.1X**ですらもはや「安全」とは言えないレベルに達している！**

## Korek 攻撃

- FMS を最適化した「統計的攻撃」
- すばらしい性能を示す！
  - FMS は数百万パケットを集める必要性がある
  - Korek では10〜20万パケットで済む
- ツールの進化
  - Korek のアルゴリズムを実装
    - aircrack、WepLab、AirSnort、等
  - パッシブ型 (e.g. AirSnort) からアクティブ型 (e.g. aireplay) へ
    - ARP Injection等により、104ビットWEPを10分程度で破ることができる！
      - 802.1Xですらもはや「安全」とは言えないレベルに達している！
    - 高速な無線技術 (MIMO等) によるさらなる時間短縮の可能性





## 百聞は一見にしかず！！

- aircrackを使ったデモ
  - http://sid.rstack.org/videos/aircrack/whax-aircrack-wep.html

# WEPの問題点

- 鍵長が 40bit と短い
  - Brute Force で破れる
  - 最近ではほとんどの場合長い鍵 (e.g. 104 or 128 bits)が利用可能。
- ICV に CRC32 を用いている
  - ICVは暗号化対象ではあるが、CRC自体は暗号的強度はない
  - 鍵と組み合わされていない
  - MACアドレスの偽称を検出できない
- 一つの鍵を使い続ける
  - どんなに強力な暗号アルゴリズムであっても1つの鍵を長く使うのは望ましいことではない





# WEPの問題点(cont'd)

- 鍵の配布メカニズムがない
  - 管理上スケールしない
- IV の空間が小さい (i.e. 24bit)
  - フレームごとに1増やす場合、200 bytes/packet, 10% utilized で 14 時間で再利用される。
  - 扱い方が規定されていない
- リプレイ攻撃に無力
- FMS、Korek 等の統計的攻撃

## ということで・・・

- Good Bye WEP!
  - What on Earth does this Protect?
  - 802.1X の rekey さえ追いつかない可能性あり
- Hello WPA / WPA2（aka 802.11i）





## WPA の目標

- 暗号的脆弱性の排除
- ユーザーベースの認証
- 鍵の配布をサポートすること
- 動的なユーザー・セッション・パケット毎の鍵を使用
- 認証サーバーを強要しないこと
- 2003年中に利用可能になること
- ソフトウェアアップグレード可能

## WPA (Wi-Fi Protected Access)

- 802.11i のサブセット
- 認証
  - 802.1X + EAP
- 秘匿性 (暗号化)
  - 802.1X 動的鍵配布
  - TKIP
- 完全性
  - Message Integrity Check (MIC) “Michael”





## WPA ステップ

- アソシエーションとケーパビリティーの確認
- 802.1X 認証と PMK (Pairwise Master Key) の配布
- TK (Temporal Key) の導出
- GK (Group Key) の導出
- 暗号化および整合性チェック

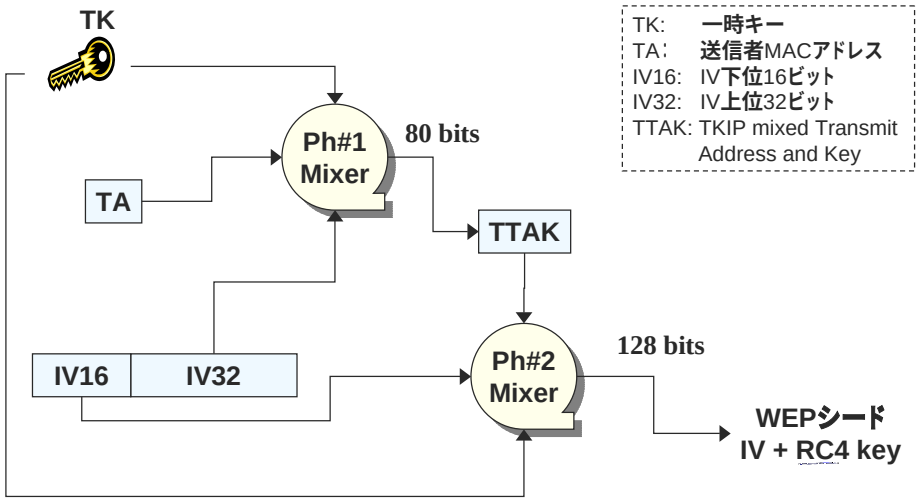
# TKIP (Temporal Key Integrity Protocol)

- IV **空間の拡張** (24 -> 48 bits)
- IV **シーケンス処理の規定**
- Per-packet-mixing Function
- Michael MIC (Message Integrity Code)

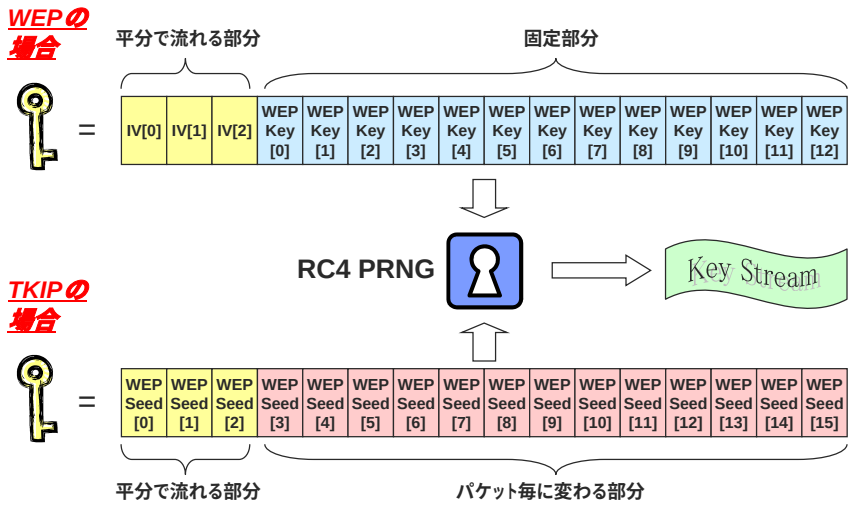




# Per-packet-mixing function



# WEP vs TKIP with RC4



# What's Michael ?

- Niels Ferguson によって考えられたメッセージダイジェスト関数の一種
- 8 octets の hash 値を生成
- 守られるのは、
  - Destination MAC address
  - Source MAC address
  - Priority
  - Data
- MIC Keyが入力される

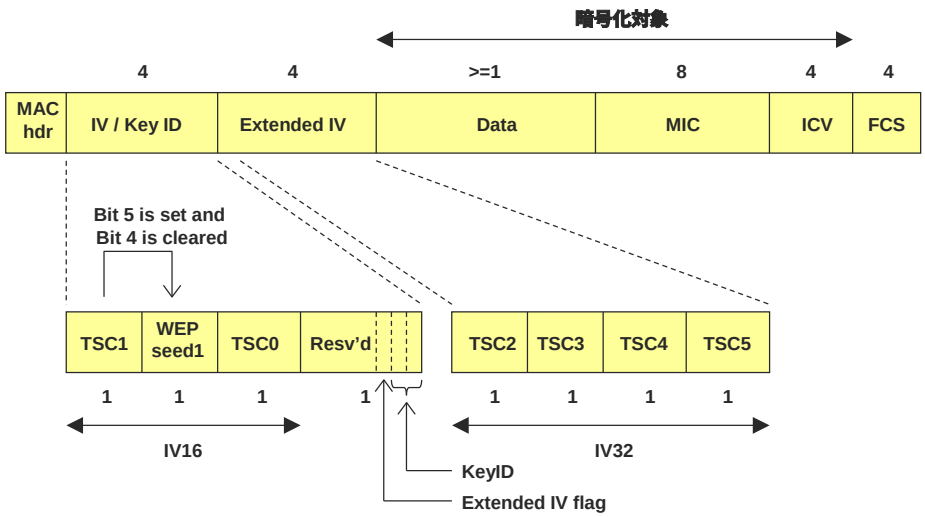
# Why Michael ?

- **与えられた** CPU **サイクルはごく僅か**
  - MD5 **や** SHA-1 **は使えない**
  - **演算を慎重に選ぶ必要あり**
- **設計上のゴールは** 20 bits **の強度を持つこと**
  - **現在知られている最も強力な攻撃は** 2^29 **個のメッセージを使った差分暗号解析**
- Counter-measure **が必要**





# TKIP **フレームフォーマット**



## PreShared Key (PSK) Mode

- RADIUS を使用しない（用意できない）場合を想定
  - ホームユース
- 802.1X で実現していた部分を手動設定で代替
  - 認証
  - PMK の配布
  - 802.1X 以降の動き(4 and 2 way handshake, 鍵の導出、TKIP、等) は non-PSK 時と同様
- PMK (256bits) を AP, STA 双方に設定





## WEPの問題点の解決

- 鍵長が 40bit と短い（もともと大きな問題ではなかったが・・・）
  - 暗号強度は104ビットとなり、（現時点では）Brute Forceでは破れなくなった
- ICV に CRC32 を用いている
  - Michael MIC による検証
  - MAC アドレスもカバー
  - MIC Key を使用する
- 一つの鍵を使い続ける
  - 802.1X による PMK の更新
  - PSK では未解決

# WEPの問題点の解決(cont'd)

- **鍵の配布メカニズムがない**
  - 802.1X による鍵配布
  - PSK では未解決
- **IV の空間が小さい** (i.e. 24bit)
  - 48 ビットに拡張された
- **リプレイ攻撃に無力**
  - IV の増やし方が規定され、検出できるようになった
- **FMS、Korek 等の統計的攻撃**
  - TKIP の key mixing function で IV と生成されるKey Streamの関連性を(ほぼ？)なくした





# 典型的WPA(TKIP) の説明


TKIPはWEPの暗号化技術をより発展させ、一定時間ごとに自動的に暗号キーが変更されるしくみを持つ。

(B誌、2004年9月)

# TKIPの設定画面

確かにWPA(TKIP)には鍵の更新間隔に関するパラメータが追加されている！

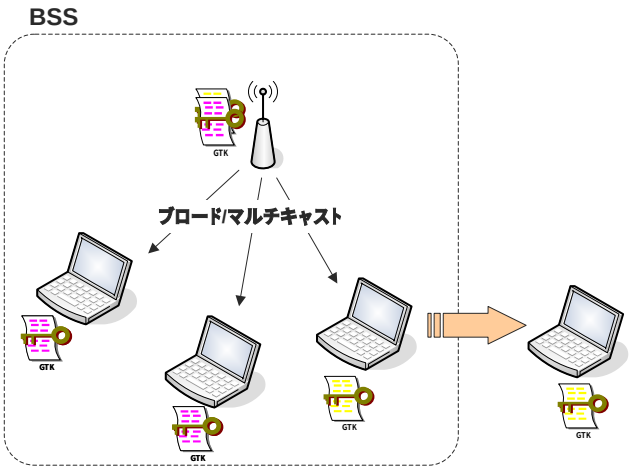




# 鍵の更新

- WPA で鍵を更新するって、どの鍵よ？？
- 暗号鍵のおおもと（i.e. PMK?）
  - **多くの人はこれであると思っているのでは！？**
  - でも、PSKの時だって鍵の更新はできるはずでしょ！
    - じゃないと、WPA-PSK はあぶない、ということになってしまうはず
- じゃ、いったい何？？

# 更新の必要があるのはGroup Key！








# Group Key 更新のタイミング

- 本来はクライアントがBSSから去ったらGroup Key を更新すべき！
- しかし、それではオーバーヘッドが大きいので、
  - 一定時間経ったら更新する
  - 一定のパケット数そのGTKを使ったら更新する

というのもアリ

## 鍵更新に関する誤解

- 多くの人は（鍵の安全性劣化を防ぐために）一定時間ごとに鍵（のおおもと=PMK）を更新すると思っている（ハズ）
  - 説明としては分かりやすい
    - FMS攻撃は沢山パケットを集めなければいけない
    - 沢山パケットを集められる前に鍵を変えてしまえ！
    - WPA（TKIP）は一定時間で鍵を更新するので安全
  - しかし、これは802.1Xで既にやっていた（できていた）ことである！





## IEEE 802.11i

- 802.11iは2004年6月に正式規格として成立
- CCMP (Counter-mode with CBC MAC Protocol) が必須
  - AES を使用
- TKIP はオプション扱い
- その他の部分はほぼ WPA と同様だが、若干の機能追加あり
  - PMK caching
  - Pre-authentication

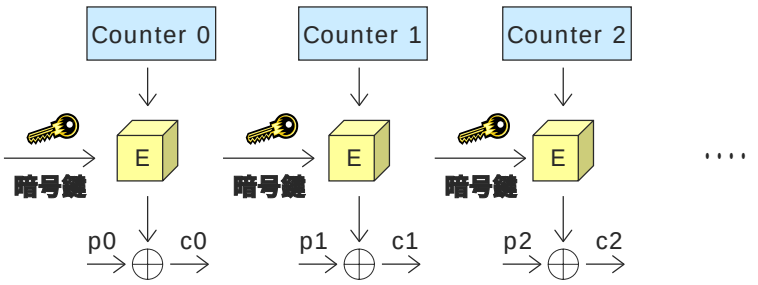
# CCMP

- Counter-mode CBC-MAC Protocol
  - AES を “Counter mode” で使用
  - AES で “CBC-MAC” も計算
- **暗号化と整合性検証を同時に実現する！**
- RFC 3610





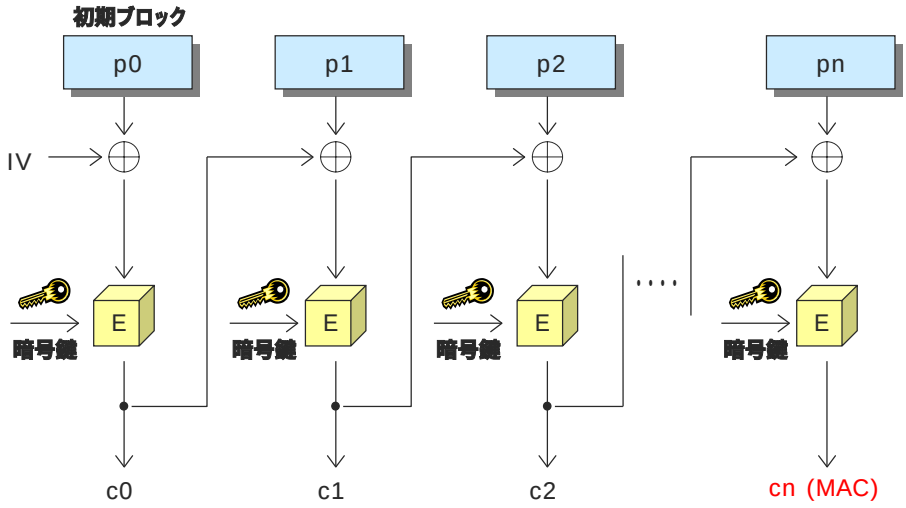
# Counter-Mode




- 復号化も全く同じプロセスで良い
- 並列化可能
- ランダムアクセス
- 事前に計算しておける
- メッセージはブロックサイズに依存しない

- 暗号化だけあればよい
  - AESは暗号化と復号化は異なる

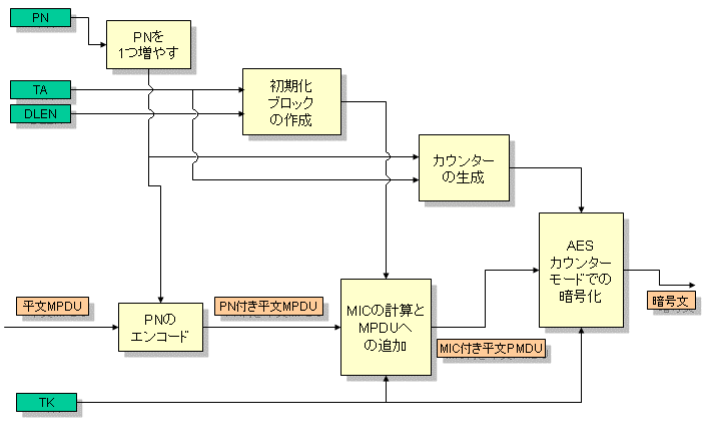
# CBC-MAC








# CCMP Encapsulation 処理の流れ



## WPA2

- WPA2 は 802.11i の相互接続性を WiFi Alliance が具体化し、認定するもの
  - WPA2 で認定されているものは 802.11i に準拠したものとなる
- 2004年9月から認定作業を開始
  - 現在、約100社弱が認定をパスしている (Personal & Enterprise)
  - ほとんどの “新” 製品はサポート





## WPAへの Korek 攻撃


- WPA は Korek 攻撃に耐性があるの？？
  - WPA は依然として RC4 を使っているけど‥。
- 答え
  - 大丈夫！ (いまのところ)
- なぜ？
  - FMS と同様、Korek も Key と Key Stream の統計的関連性 (偏り) に依存している。
  - WPA では TKIP の key-mixing によりこの問題が解決されている

# WEP, TKIP and CCMP

| | WEP | TKIP | CCMP |
|---|---|---|---|
| 暗号化アルゴリズム | RC4 | RC4 | AES |
| 暗号鍵の長さ(bits) | 40 / 104 / 128 | 104 | 128 |
| 認証鍵の長さ(bits) | N / A | 64 | 64 |
| IV の長さ(bits) | 24 | 48 | 48 |
| データ部の完全性 | CRC32 | Michael | CCM |
| ヘッダ部の完全性 | なし | Michael | CCM |
| Anti-Replay-Attack | なし | あり | あり |

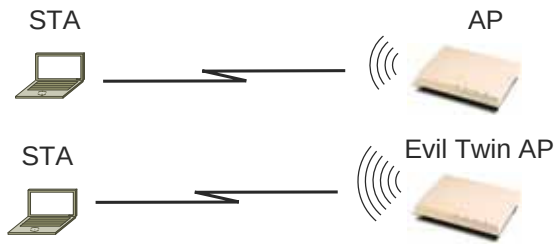




# 不正アクセスポイント（Rogue Access Point）

- 2つのタイプ
  - きちんと設定されずにネットワーク管理者に無断で設置したアクセスポイント
    - 踏み台になる可能性
  - 正当なアクセスポイントにみせかけ、クライアントを接続させ、情報を入手したり悪意のある行為をおこなう
    - 通称 “Evil Twin”（悪魔の双子）、”Access Point Phishing”、”WiPhishing”

## Evil Twin / WiPhishing




- より強い電波を出すAPを設置
  - 新しいクライアントはそちらに associate する





## 逆 war driving

- Evil Twin を搭載した車で、クライアントを “釣りに” 行く

## 何が恐怖か？

- 通信を傍受されることが恐怖ではない
- 多くの公衆無線LANサービスでは、接続時に強制ポータルで、ユーザーID / パスワードを求められる
  - ユーザーは他でも同じユーザーID / パスワードを使う傾向にある
- さらに積極的な攻撃の可能性も
  - 悪意のあるコードをダウンロードさせる、etc.





## お手軽キットも存在する

- Airsnarf
- 構成要素
  - Linux, httpd, dhcp, sendmail, iptables, Perl DNS module
- どのように動いているか？
  - 自分自身 or 自分配下の AP に associate させて、
  - dhcp で IP アドレス、DNS サーバー、デフォルト･ゲートウェイを割り当て、
  - どの URL も自分に resolve するようにし、
  - 自分の httpd（強制ポータルページ）に接続させ、ユーザーID / パスワードを入れさせ、
  - それをメールで管理者に通知する！

## 実際に起こりました！

- 2005/04、ロンドンのWireless LAN関連イベントで Evil Twin がしかけられ、誤って接続したユーザがウィルスに感染
  - http://news.zdnet.co.uk/0,39020330,39195956,00.htm
- 何が起こったか
  - SSIDを”Free_Internet_Access”、”BTOpenzone”、“T-Mobile”などと順次変え、
  - 接続してきたユーザに、ランダムに生成された45種類のウイルス、ウォーム、キーロガーをダウンロードさせた
  - 怪しい人物の目撃証言





## 他にも続々・・・

- INTEROP (May. 2005) at Las Vegas
- DefCon (Aug. 2005) at Atlanta
- ひょっとしたら、あなたが今繋いでいる AP も・・・。

## Evil Twin 攻撃をどのように防ぐか (1)

- 802.1X で、きちんとした認証を行う！
  - ほんとうに十分だろうか?
- それが出来ない場合は、ユーザーが最大限の注意を払う
  - 前回のログイン日時やユーザ名に注意する
  - 接続されたAPのMACアドレスを確認
  - 見覚えの無いポータル画面に注意する





## Evil Twin 攻撃をどのように防ぐか (2)

- もう少しクライアント側に工夫があってもよいのではないか？
  - 接続 AP の MAC アドレスのホワイトリスト
  - SSID の変化
  - AP の MAC アドレスの変化
  - デフォルトゲートウェイの MAC アドレスの変化
  - 急激な電波強度の変化
- ツールも多少出つつある
  - AirDefense Personal, etc.

# まとめ

- 「WEPは二度死ぬ！」
  - FMS 攻撃
  - Korek 攻撃
- 現時点では WPA / WPA2 に大きな脆弱性は見つかっていない
  - できる限り速やかに WEP から WPA / WPA2 に以降すべし！
  - WPA / WPA2 の安全性に対する継続した注意も必要
- Evil Twin は大きな問題！
  - もう少し積極的に防ぐ方法が必要





# 略語一覧

| | |
|---|---|
| AES | Advanced Encryption Standard |
| AP | Access Point |
| CBC | Cipher Block Chaining |
| CCMP | Counter-mode CBC MAC Protocol |
| CFB | Cipher Feedback |
| CRC32 | Cyclic Redundancy Check 32bits |
| DoS | Denial of Service |
| EAP | Extensible Authentication Protocol |
| EAPOL | EAP over LAN |
| ECB | Electronic Code Book |
| ESS | Extended Service Set |
| FCS | Frame Check Sum |
| GK | Group Key |
| GMK | Group Master Key |
| ICV | Integrity Check Value |
| IE | Information Element |
| IV | Initialization Vector |
| LCG | Linear Congruential Generator |
| LEAP | Lightweight EAP |
| MAC | Message Authentication Code |
| MD5 | Message Digest 5 |
| MIC | Message Integrity Code |
| OFB | Output Feedback |
| PAE | Port Authentication Entity |
| PBKDF | Password-Based Key Derivation Function |
| PEAP | Protected EAP |
| PKCS | Public Key Cryptographic Standard |
| PMK | Pairwise Master Key |
| PPP | Point-to-Point Protocol |
| PRF | Pseudo Random Function |
| PRNG | Pseudo Random Number Generator |
| PSK | PreShared Key |
| PTK | Pairwise Transient Key |
| RADIUS | Remote Access Dial-Up System |
| RC4 | Rivest Code (or Cipher) 4 |
| RSN | Remote Secure Network |
| SHA1 | Secure Hash Algorithm 1 |
| SSID | Service Set Identifier |
| STA | Station (client) |
| TA | Transmit (MAC) Address |
| TK | Temporal Key |
| TKIP | Temporal Key Integrity Protocol |
| TLS | Transport Layer Security |
| TTAK | TKIP-mixed Transmit Address and Key |
| TTLS | Tunneled TLS |
| WEP | Wired Equivalent Privacy |
| WPA | Wi-Fi Protected Access |
| WRAP | Wireless Robust Authenticated Protocol |
| XOR | Exclusive OR |