T22: 間違いだらけの無線LAN

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

間違いだらけの・・・・

- 車選び
- ゴルフクラブ選び
- ハウスメーカー選び
- 「選び」じゃないけど
  - 無線LAN

間違いを犯した（している）のは・・・

- 設計者
  - プロも間違いを犯す
  - そこから学ぶ
- 伝える側
  - 雑誌
  - Web
  - その他もろもろ
- まれにベンダーも・・・

3

セキュリティー

- 認証（Authentication）
- 許可（Authorization）
- 秘匿性（Confidentiality）
- 完全性（Integrity）
- 否認防止（Non-repudiation）
- …

4

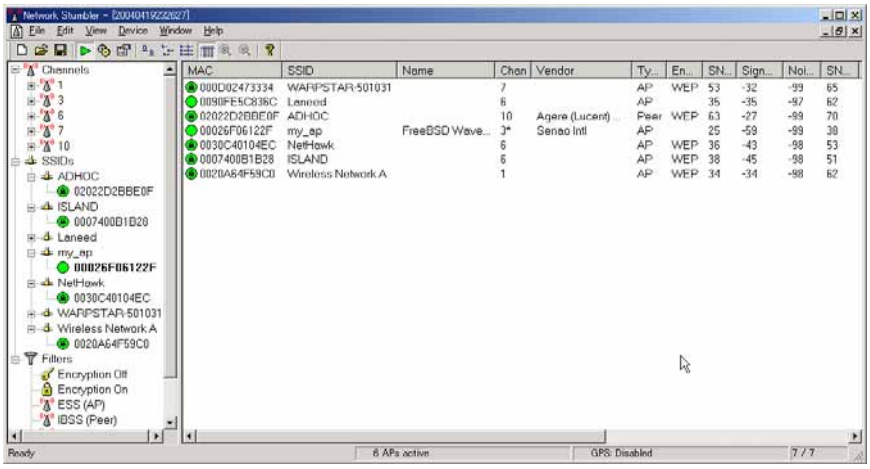
攻撃(1) ～ War Driving ～

- War Driving Tool
  - Netstumbler
  - dstumber
  - kismet
- GPS との組み合わせ
- SSID、電波強度、などを表示

http://www.wifimaps.com

5

NetStumbler



6

Kismet

7

ついに起きたか・・・

他人の無線LAN盗用…不正アクセスで逮捕の大学職員

(読売新聞：04/06/09より)

高千穂大学(東京都杉並区)のコンピューターシステムへの**不正アクセス事件で逮捕**された大職員が、調布市内の会社役員(33)が家庭で使っている無線LAN(構内情報通信網)に"ただ乗り"して、不正アクセスしていたことが9日、警視庁の調べでわかった。

パソコン通信で使われる無線LANは、**第三者に電波を"盗用"される恐れがある**と指摘されていたが、実際に不正アクセスへの悪用が表面化するのは異例。事態を重視した警視庁は、無線LANの危険性について注意を呼びかける。

調べによると、不正アクセス禁止法違反容疑で逮捕された同大職員中山良一容疑者(47)は昨年11月下旬、車に積んだパソコンを使って、無断使用防止対策が講じられていない無線LAN用電波を物色。

東京・調布市の住宅街で、会社役員宅の電波が無断で使えるのを発見した中山容疑者は、近くに車を止め、会社役員の**無線LANに"ただ乗り"**してインターネットに接続。他人のIDとパスワードを使い、同大のコンピューターシステムにアクセスしたという。

(以下略)

8

対策(1-1) ～ SSID の隠蔽 ～

- 呼び名は色々
  - ステルス機能
  - SSID ブロードキャストの無効化
  - Any 拒否
  - Closed System or Network
  - …
- 実装も色々
  - 802.11 のビーコンを止める
  - プローブリクエストに対して、
    - 応答しない
    - 応答はするが、レスポンスに SSID は入れない
    - 自分の SSID に合致する場合のみ応答
- 802.11 の仕様上、完全に秘匿するのは不可能！
  - Sniffer の類
  - Kismet

9

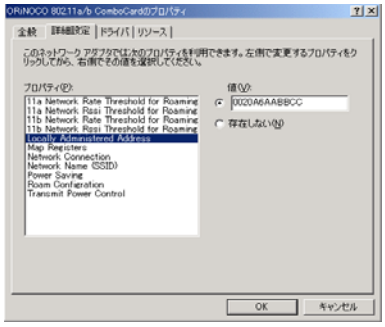
対策(1-2) ～ MAC 認証 ～

- 接続を許可する MAC アドレス (のリスト) を設定
  - AP に静的に設定する
  - RADIUS 等のサーバーに設定する

10

攻撃(2) ～ 詐称（なりすまし）～

- MAC アドレスの詐称は簡単！
- 正規のMAC アドレスはワイヤレス上で簡単に見つけることができる！



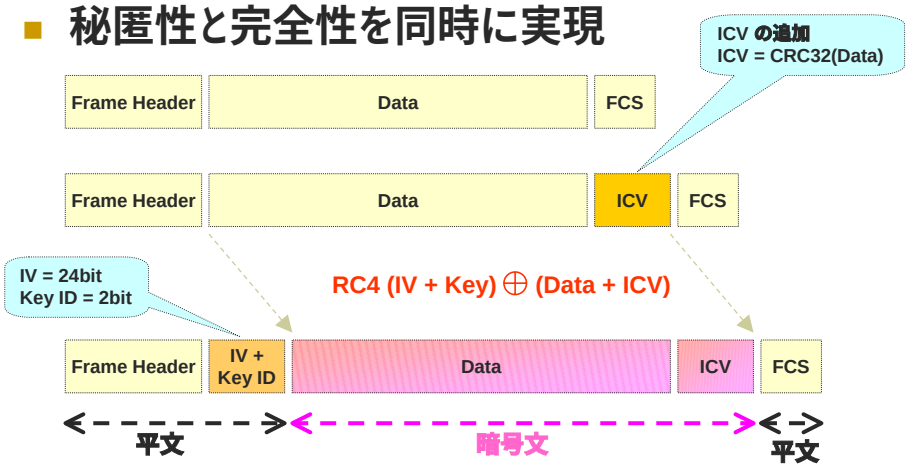

```
# ifconfig eth1 down
# ifconfig eth1 hw ether 12:34:56:aa:bb:cc
# ifconfig eth1 up
```


対策(2) WEP

- WEP (Wired Equivalent Privacy)
  - **秘匿性** (Confidentiality)
  - **完全性** (Integrity)
  - **認証** (Authentication)
- What on Earth does this Protect?

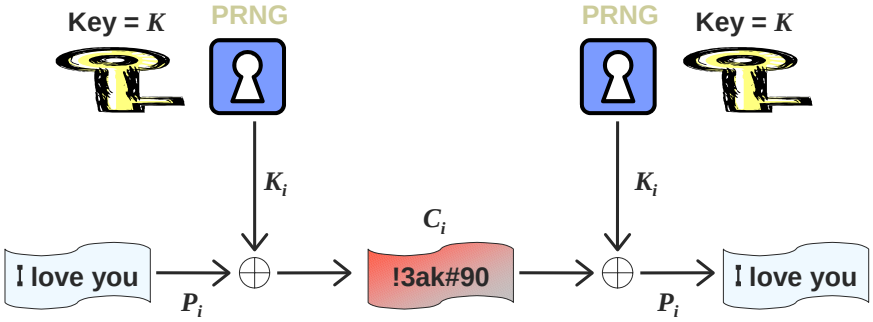
WEP 処理

- 秘匿性と完全性を同時に実現



13

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 $C_1 \oplus C_2 = (P_1 \oplus K_a) \oplus (P_2 \oplus K_a) = P_1 \oplus P_2$

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。
- 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も1つの鍵を長く使うのは望ましくない。

15

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)
  - 扱い方が規定されていない。
  - 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。
- リプレイ攻撃に無力
- FMS 攻撃

16

ほんと？？

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）

17

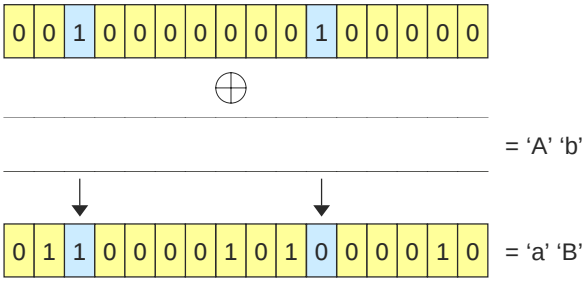
同じIVを使うと何が起こるか？

- 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

18

Bit Flipping Attack

- CRC は XOR に対して線形である！
  - CRC (M XOR ⊿) = CRC (M) XOR CRC (⊿)
- M 中の任意の bit を set したり、clear したりすることはできないが、bit を反転させることはできる！



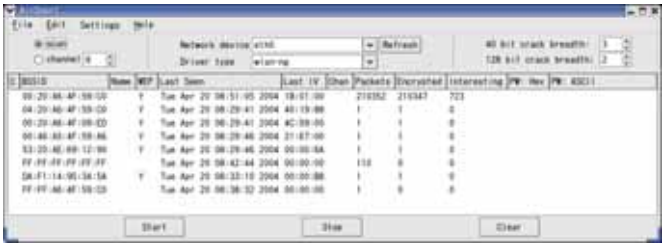
19

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

20

攻撃(3) ～ WEP Cracking ～



AirSnort
http://airsnort.shmoo.com

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

21

ほんと？？

さらに、WEPが利用しているRC4と呼ぶ関数では256バイトごとにRC4ストリームが初期化される。つまり、一つの電文内でN + (256 × c) バイト目（ただし c ≧ 0）は同じ値のキーストリームで暗号化されているのだ。従って、これらの内の1バイトでも分かれば他のバイトもすべて割り出せてしまう。

（D誌、2003年4月）

22

もちろん

- そんなことはない！
- もし、そんなことになっていたら怖くてインターネットショッピングなんてできない

SSL 保護つき (128ビット)

23

RC4 は脆弱か？

- 若干の脆弱性はあるが、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
  - 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

24

WEPを少しでも安全に使うには

- 鍵はできるだけ長いものを使う！
  - RC4はアルゴリズム上、長い鍵を使ったからといって処理が重たくなるようなことはない

25

ほんと？？

WEPの暗号鍵（WEPキー）は「文字入力」か「16進数入力」のいずれかを選択できる。文字入力の場合は半角英数字／記号（大小文字は区別される）を入力できる。16進数の数字の羅列より、意味を持たせられる文字入力のほうが間違いに気がつきやすくおすすめ

（B誌、2004年9月）

26

いやいや

- 16進数で設定できるなら16進数で設定すべき！
  - ASCII文字で設定すると1文字あたりの強度が8ビットから6ビット程度に低下する
    - 40ビット (5文字) → 30ビット
    - 104ビット (13文字) → 78ビット
  - 意味を持たせた文字列にするとさらに強度が低下する
    - パスフレーズの持つ強度は 2.5 × n + 12ビット 程度と言われている
      - 40ビット (5文字) → 24.5ビット
      - 104ビット (13文字) → 44.5ビット
  - 任意長のパスフレーズから40ビットのWEP鍵を生成するものの多くには脆弱性があり、21ビットの強度しか持っていない

27

パスフレーズからWEP鍵を生成する例

- パスフレーズの4文字目、8文字目、12文字目、・・・を変更しても同じ鍵が生成されてしまう場合は脆弱なアルゴリズムが使われている

28

これは？？

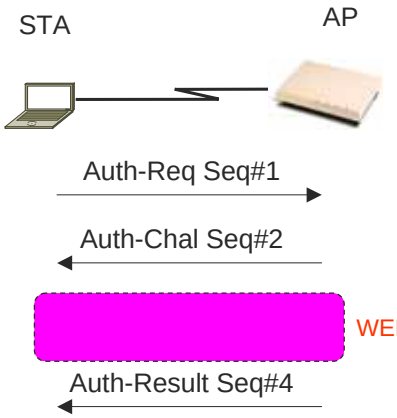
WEPキーは無線LANアクセスポイントの裏などに書いて忘れないようにしよう。

(A誌2004年9月)

論外☺

29

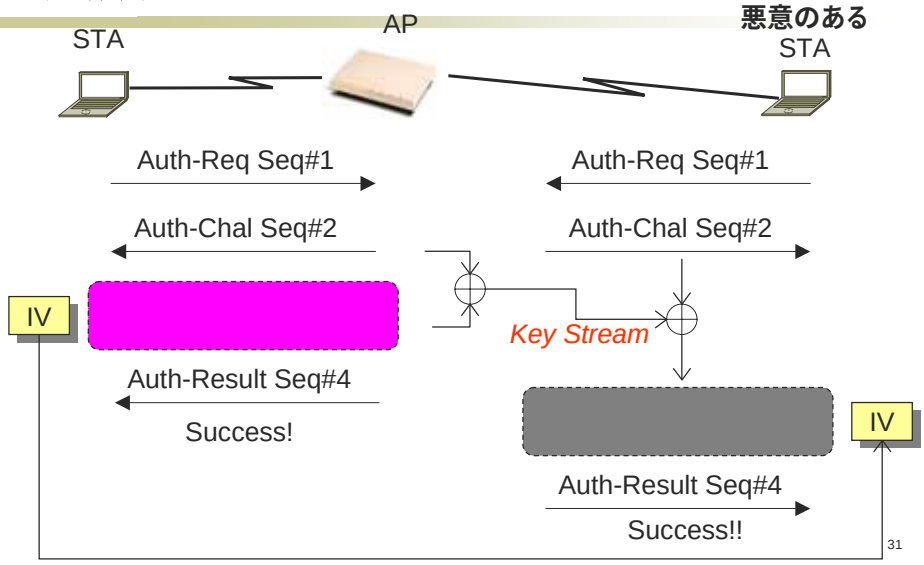
802.11 の認証



- WEP を使う！
  - AP は Challenge (128bytes) を送出
  - STA はそれを WEP で暗号化して AP へ送る
  - AP はそのフレームの整合性をチェック

30

失敗その2



31

WEP is completely broken!!

秘匿性　　**完全性**　　**認証**

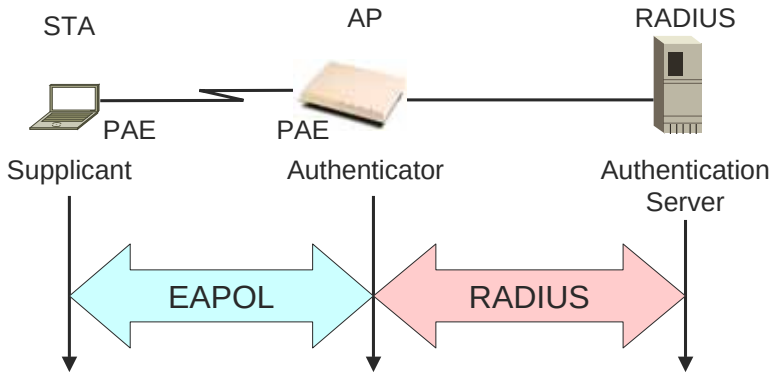
32

対策 (3-2) 802.1X

- Port-Based Network Access Control
- a.k.a 802.1aa
- **認証を「ユーザーベース」できちんとしよう！**
- **鍵配送の仕組みを提供しよう！**
  - **管理上のスケーラビリティー**
  - **暗号化方式の脆弱性を「和らげる」**

33

802.1X の構成要素



34

EAP (Extensible Authentication Protocol)

- **拡張性に富む**
  - **基本的枠組みしか規定しない**
- Authenticator **と** Back-end Server **の分離**
- **実際にどのように認証するかは** EAP Type **で決まる**
  - MD5(4)、EAP-TLS (13)、EAP-Cisco Wireles(17)、EAP-TTLS (21)、PEAP(25)、EAP-FAST(43)、**他多数**

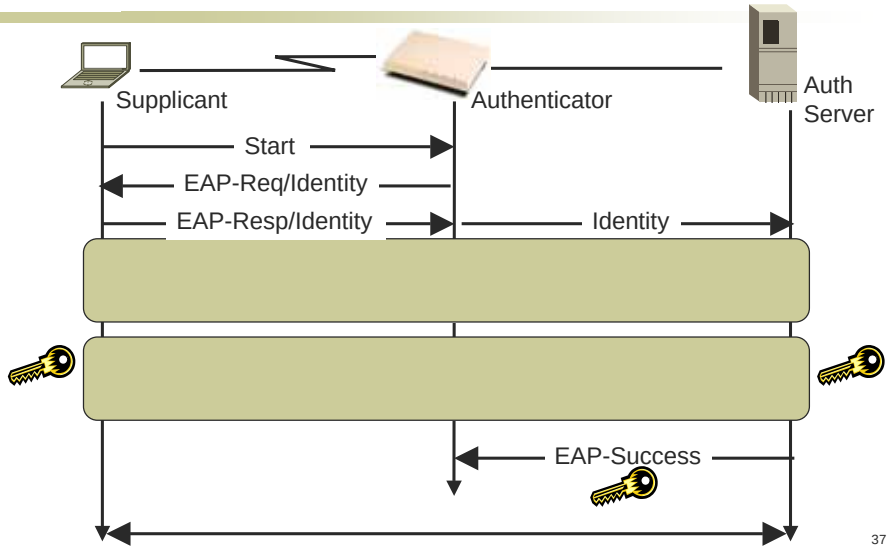
35

TLS (Transport Layer Security)

- TLS Version 1.0
  - a.k.a SSL version 3.1
- **電子証明書ベース**
- **なぜ** TLS ??
  - **相互認証**
  - **セッション鍵**
  - **広く受け入れられているから**

36

802.1X の動き



37

EAP Type

EAP Type	仕様	相互認証	認証に使うもの		鍵の配布	ユーザ名の秘匿	Windows 標準サポート
			サプリカント側	オーセンティケータ側			
MD5	公開	×	ユーザ名/パスワード	なし	×	×	△
TLS	公開	○	電子証明書	電子証明書	○	×	○
TTLS	公開	○	ユーザ名/パスワード	電子証明書	○	○	×
PEAP	公開	○	ユーザ名/パスワード	電子証明書	○	○	○
LEAP	独自	○	ユーザ名/パスワード	なし	○	×	×

38

TTLS と PEAP

- 基本的モーチベーションとアイディアは同じ
  - TLS の問題点の解決
    - サプリカント側にも電子証明書が必要
    - Identity Protection がない
  - 2段階の処理
    - まずTLSセッションを張っておいてから（この時点で認証サーバーを電子証明書で認証）、
    - そのTLSセッション上で別の認証（内部認証）を実行し、サプリカントを認証する

39

TTLS / PEAP の内部認証方式

- 理論的にはどのような認証方法を用いても良い
- TLS で守られているので、パスワードベースの認証方法を利用できる
  - TTLS : 全ての EAP に加え、PAP, CHAP-MD5 等レガシーな認証方法をサポート
  - PEAP : EAP（MS-CHAPv2 or GTC）

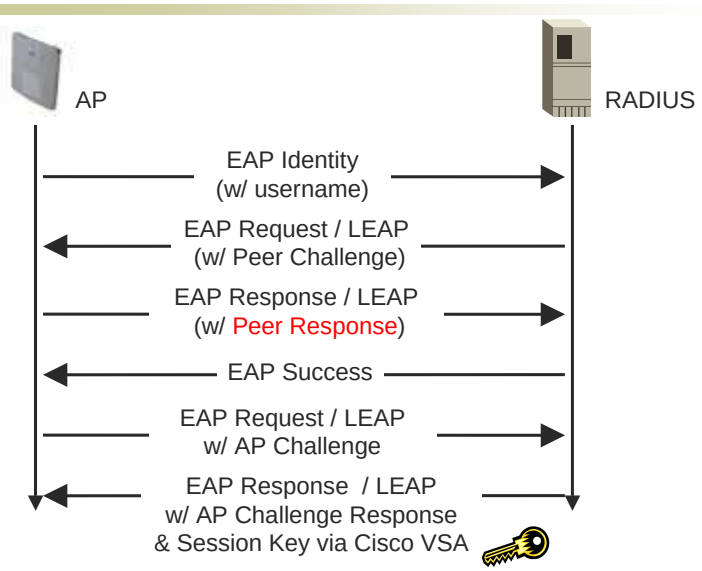
40

LEAP（Lightweight EAP）

- モーチベーション
  - TLS（サプリカント側の電子証明書呪縛）からの脱却
- TLS にかわるもの
  - でも、相互認証、セッション鍵の配布は必要
- MS-CHAP！
  - に少々手を加えたもの

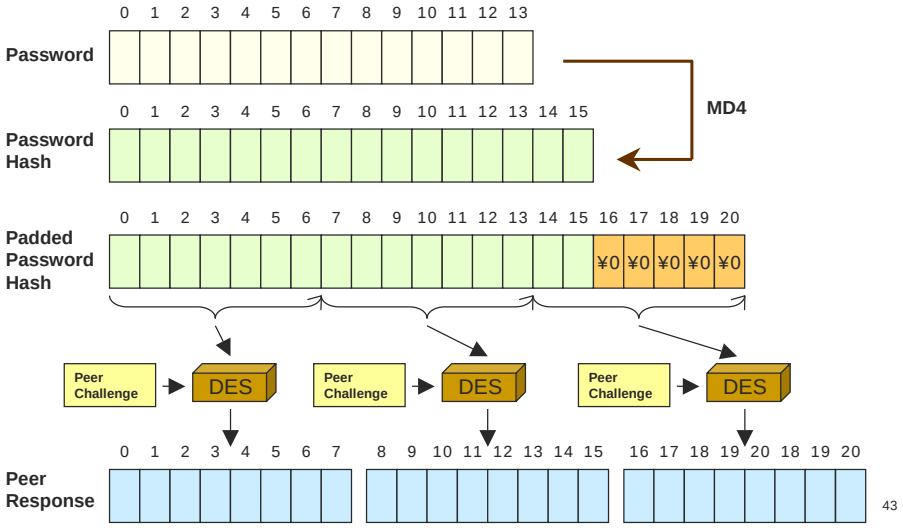
41

LEAP の動き



42

Peer Response の計算



43

LEAP の問題点

- MS-CHAP の脆弱性をそのまま引き継いでしまっている
  - ハッシュされるパスワードに“ソルト”がない
    - オフライン辞書攻撃が可能
  - Peer Response [16-23] を計算するために使われた DES の鍵は16ビットの強度しかない！
    - 簡単に brute force で最後の2オクテットの Password Hash を割り出すことができる
    - 最後の2オクテットの Password Hash がそのようになるパスワードを (辞書から探して) 試していけば良い
- LEAP cracking tool “asleap” の発表

44

FAST (Flexible Authentication via Secure Tunnel)

- Post LEAP
  - まだドラフト
  - 一部の Cisco 製品でサポートが始まっている
- TLS ベースだが、サプリカント側の証明書を必要としない
  - TLS_DH_anon_WITH_AES_128_CBC_SHA を使用
- Crypto Binding のサポート

45

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

46

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”

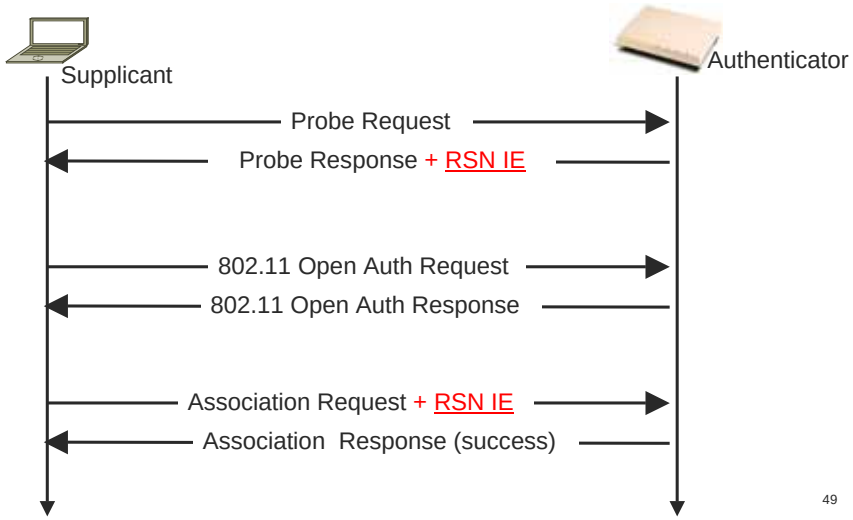
47

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

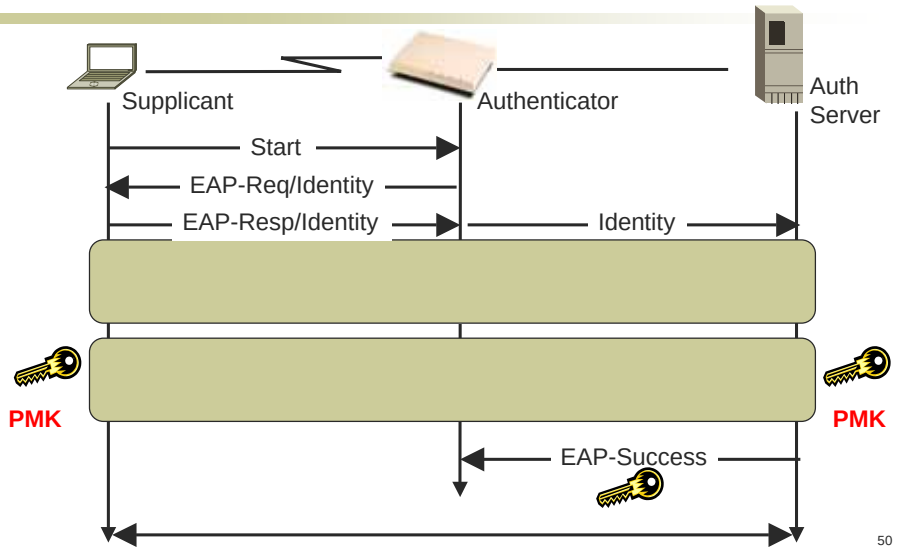
48

アソシエーションとケーパビリティーの確認



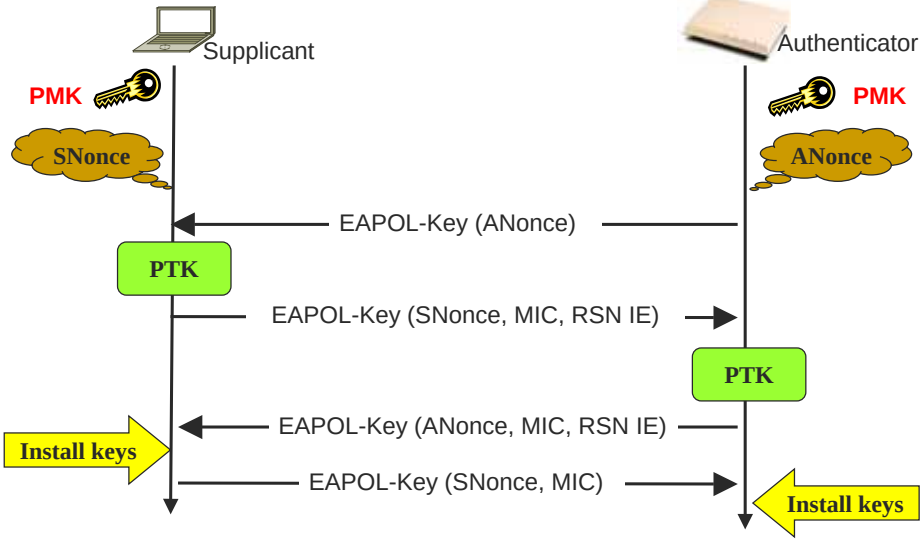
49

802.1X 認証と PMK の配布

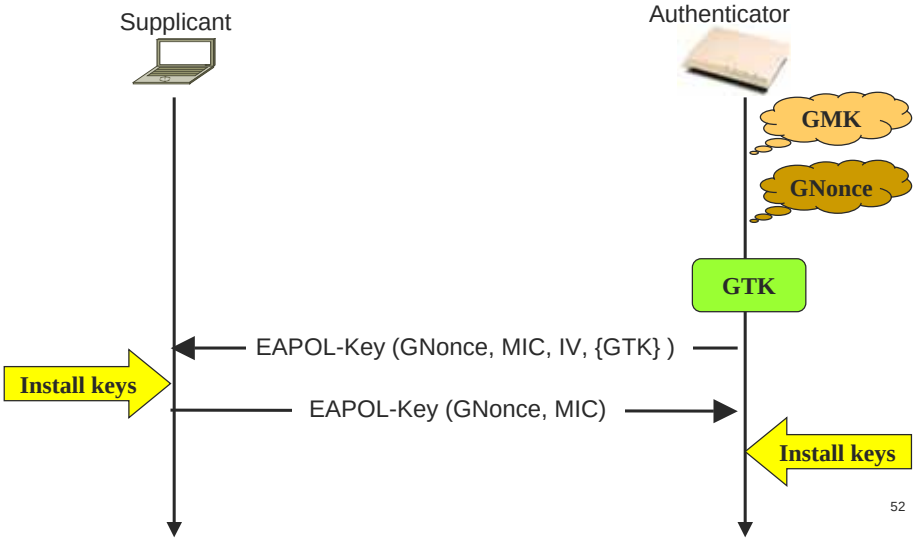


50

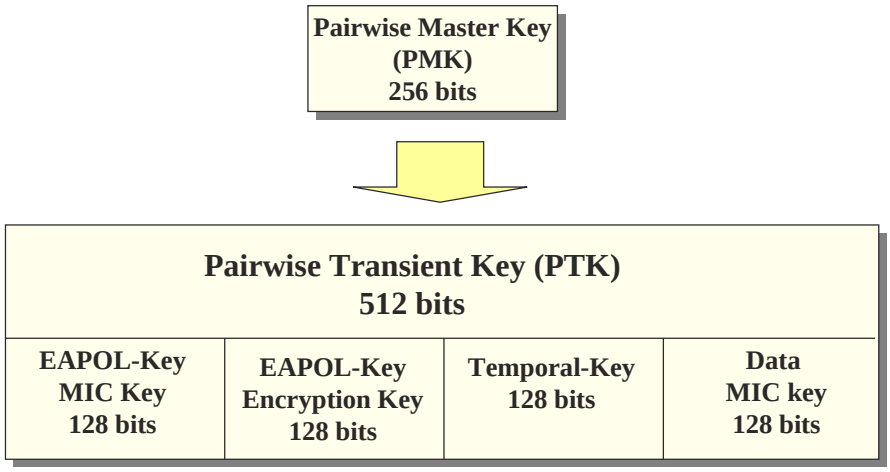
Temporal Key の導出 ～ 4 way handshake ～



Group Key の導出 ～ 2 way handshake ～

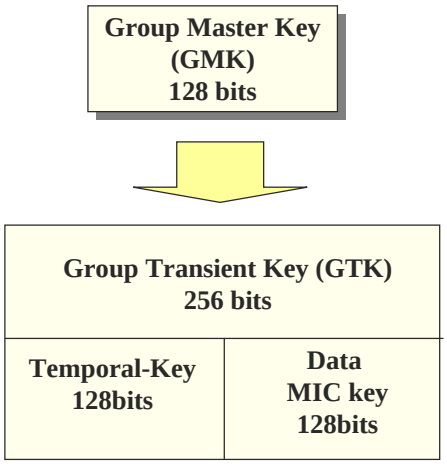


Pairwise Key Hierarchy (for TKIP)



53

Group Key Hierarchy (for TKIP)



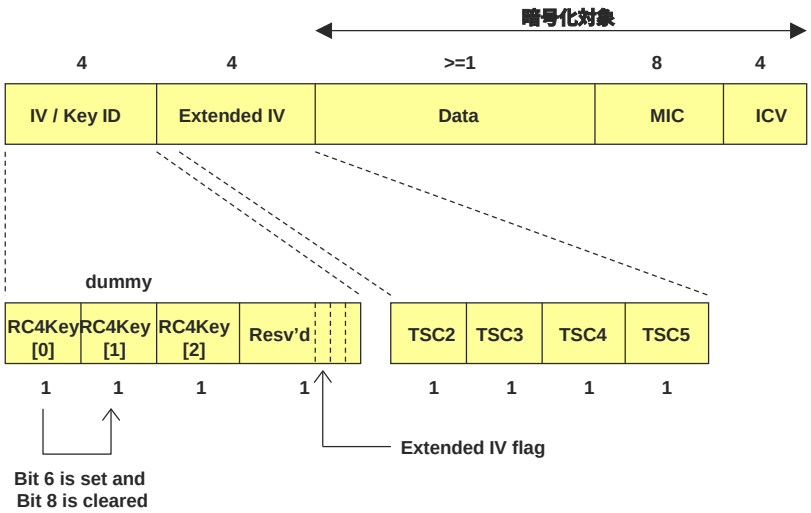
54

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)

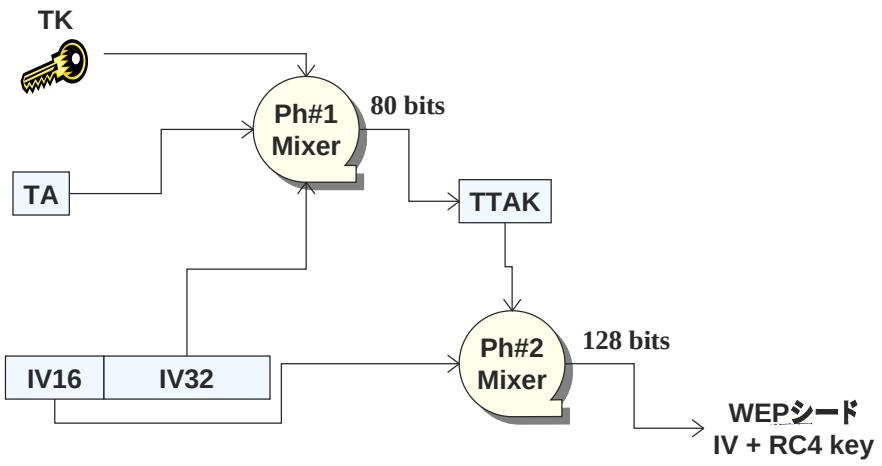
55

TKIP Frame Format



56

Per-packet-mixing function



57

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
- Pass Phrase から 256 bits PMK を生成する際の推奨方法も別途規定
  - PKCS#5 PBKDF2 (Password-Based Key Derivation Function)

58

ほんと？？

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）

説明としては分かり易いが・・・

59

Rekey (key update) in TKIP

- TKIPは「パケット毎」に暗号化に使う鍵が変わる
- ただし、マルチ（or ブロード）キャストに使われるGroup Keyはkey updateできたほうが望ましい
  - GTKは他のSTAと共有している
  - 厳密にはSTAが一つでもBSSから抜けた際にはGTKをupdateするべき
  - 定期的（時間 [秒] もしくはパケット数）でGroup Keyのupdateをすることができる
  - GMK自身もupdateすることができる
- 頻度を設定できる製品もあるし、できない製品もある
  - 設定できる製品も、それがGTKのupdateなのかGMKのupdateなのか不明な事が多い（そもそもそれが何なのか説明のない事がほとんど！）
- 802.1Xの rekey とは独立（別物）なので注意

60

これも・・・

そして、その後はTKIP（Temporal Key Integrity Protocol）という暗号化方式でやり取りする。これは先ほどのWEPとは違い、一定時間ごとに暗号化のための鍵を変更する。

（A誌2004年9月）

61

これはぎりぎり合格点？

TKIP

Temporalの名の通り、一定時間ごとに暗号鍵を変更するプロトコルです。パケットごとの鍵更新、IV鍵の48bit化などにより、従来のWEPが持っていた脆弱性を克服しています。

（ベンダーAのホームページ）

62

かなりの混乱が・・・

- WPAの特徴
  - ユーザーパスワード WPA-PSK (Pre-Shared Key) を128bitとする
  - IV を24bitから48bitとする
  - 暗号鍵は WPA-PSK と IV、MAC アドレスからハッシュ値を持って生成する
  - 1万パケット毎に暗号鍵の更新を行う

(Web上の記事A)

63

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
- MSDU に対して行われる
- 守られるのは、
  - Destination MAC address
  - Source MAC address
  - Data

64

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
- Countermeasure が必要

65

HMAC-MD5 vs Michael（参考）

	HMAC-MD5	Michael[*1]
コードサイズ	4,008バイト[*2]	1,280バイト
実行時間[*3]	5.1秒	1.8秒

<*1> Ferguson のリファレンスコード（C++）を C に書き換えたものを使用。gcc 3.2.2 で –O2 を指定してコンパイルした。
<*2> RedHat Linux 9.0 の /usr/lib/libmd5.a に含まれる md5.o のサイズ。厳密にはこの MD5 はHMAC版ではない。HMACバージョンはもう少し大きなサイズになるはずである。
<*3> 1,500 バイトの配列に対するハッシュを10万回計算するのにかかった時間を測定。

66

Is Michael subject to DoS ??

- 理論的には可能
- 実際にはちょいと面倒
  - Micheal MIC のチェックは IV counter のシーケンスチェックおよび CRC32 のチェックの後に行われる
    - IV replay protection をかいくぐり、
      - IV は Per-Packet Mixing への入力になっている！！
    - ICV のチェックをパスしなければならない。
- もっと簡単な DoS があるじゃない！
  - Disassociation or Deauthentication 攻撃
  - RF jammer

67

WEPの問題点 (再掲)

- 鍵長が 40bit と短い
- ICV に CRC32 を用いている
- 一つの鍵を使い続ける
- 鍵の配布メカニズムがない
- IV の空間が小さい (i.e. 24bit)
- リプレイ攻撃に無力
- FMS 攻撃

68

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)
  - AES が前提
  - 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

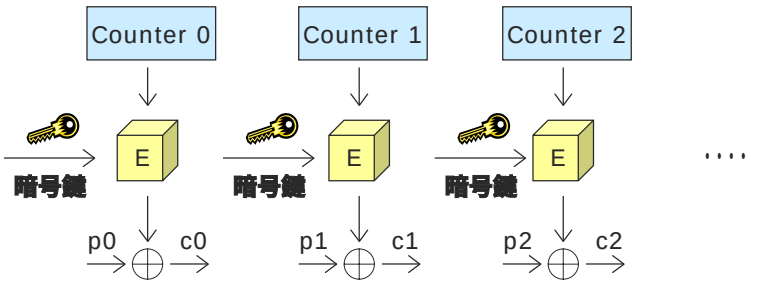
69

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

70

Counter-Mode

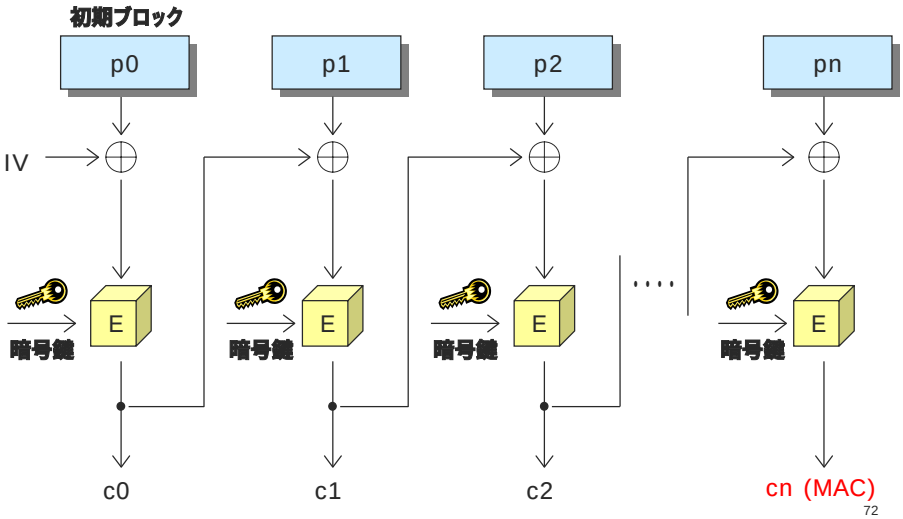


- 復号化も全く同じプロセスで良い
- 並列化可能
- ランダムアクセス
- 事前に計算しておける
- メッセージはブロックサイズに依存しない

- 暗号化だけあればよい
  - AESは暗号化と復号化は異なる

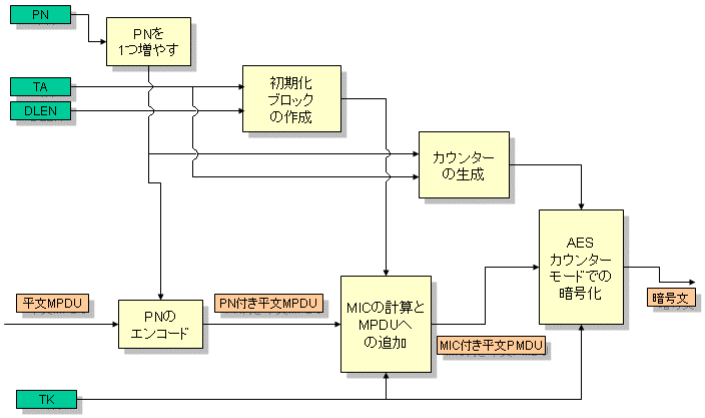
71

CBC-MAC



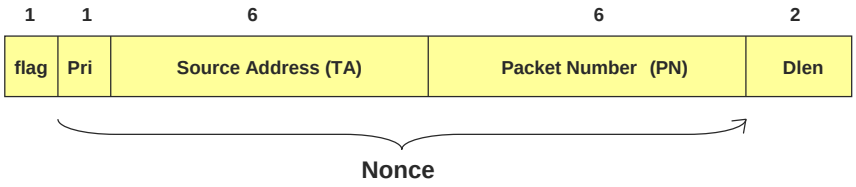
72

CCMP Encapsulation 処理の流れ



73

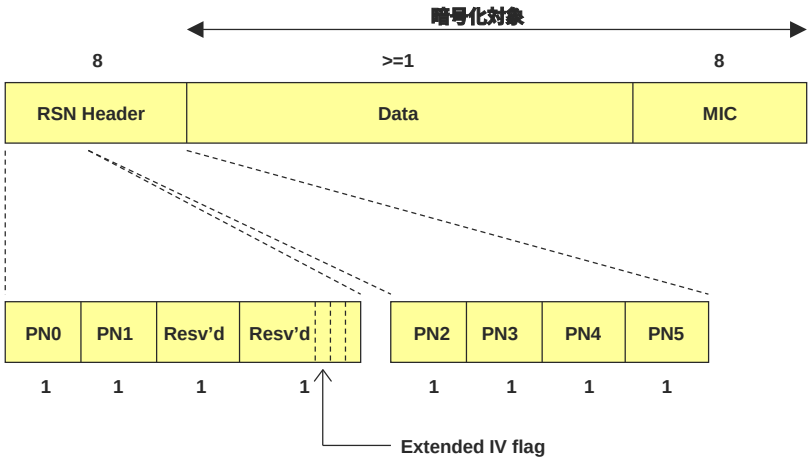
CBC-MAC の初期ブロック



- 同一内容のフレームでも異なった MAC が得られるようにするため
  - なぜ Pakcet Number だけじゃだめなの??

74

CCMP Frame Format



75

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
  - 10数社が認定をパスしている (Personal & Enterprise)
  - ぼちぼち市場に認定製品が並び始めている

76

WPA or WPA2 (802.11i)

- WPA2のほうが良いのは明白だが、実際上WPA2が必要とされる局面は
  - AES でないと受け入れることができないようなところ (e.g. 米国政府関係)
  - 高い性能が求められるところ (ハードウェアベース vs ソフトウェアベース)
- 必要に応じて使い分ければよい

77

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

78

おさらい

- 設計者たちが犯した「間違い」
  - ICVにCRC32を使った
  - RC4の使い方を間違えた
  - 簡単に破れる認証だった
  - MS-CHAPの脆弱性をそのまま引き継いだ
  - パスフレーズからパスワードを生成する方法を誤った
- 伝える側
  - 嘘まみれ！
    - きちんとWEPの脆弱性を理解していない
    - WPA/802.11iも理解していない
- ベンダー
  - 正しく伝えていない
  - 簡易性を追求するあまり、ユーザーに正確に機能を伝えられていない

79

結論

- 802.11はかなり不幸
  - 最初にミソがついてしまい、それを払拭するのに時間がかかった (or かかり過ぎた)
  - きちんと伝えられる人が (極めて) 少ない
- 誤解されたままでは可哀そう
- 正しく理解して、もっと無線を楽しもう！

80

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