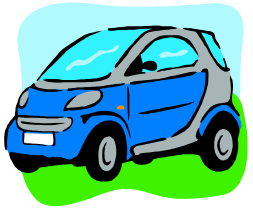
FIVEFRONT
CORPORATION

間違いだらけの無線LANセキュリティー

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



無線LANのセキュリティー

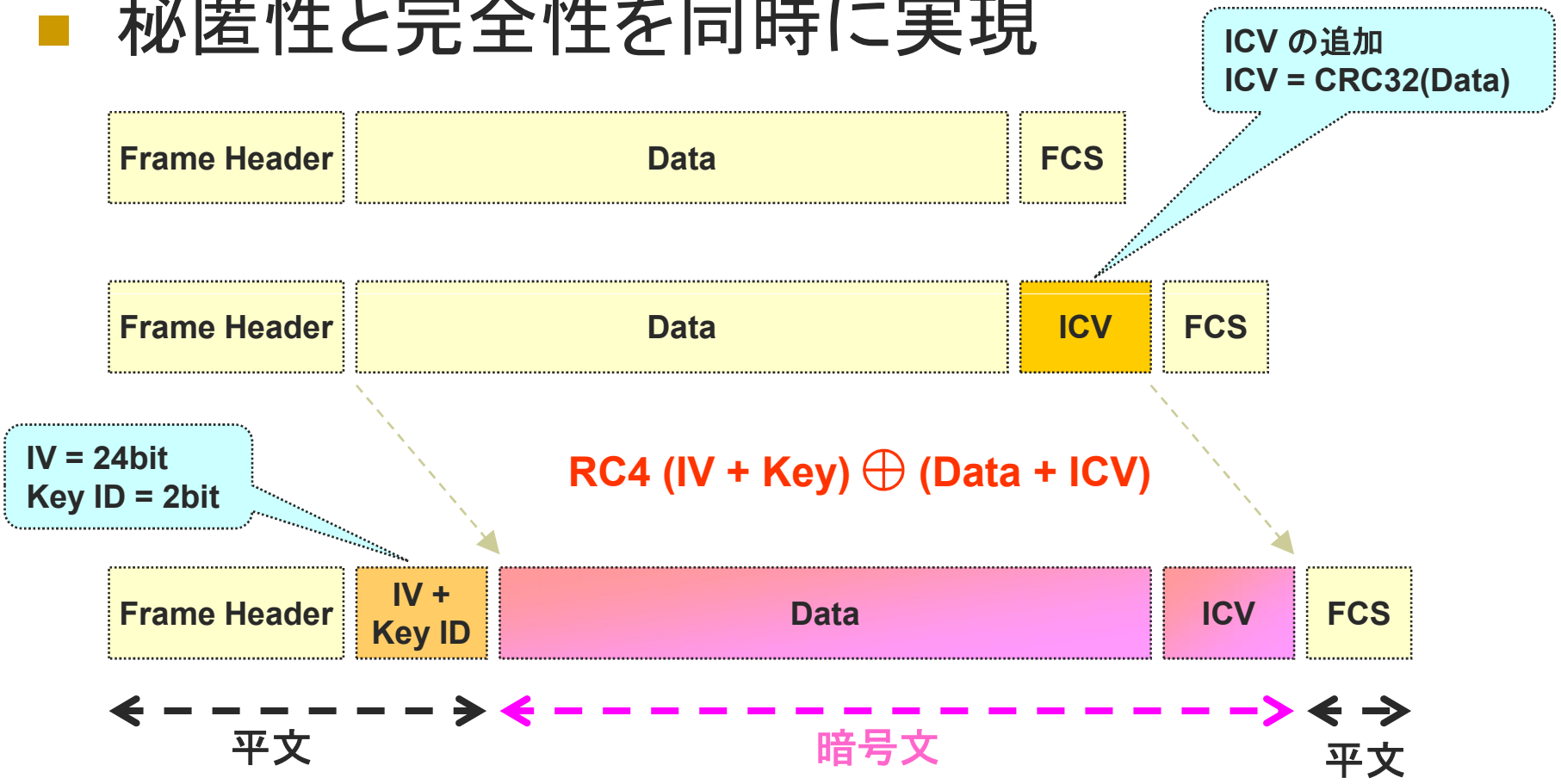
- SSIDの秘匿
- MACアドレスの制限
- WEP
- 802.1X
- WPA
- WPA2 / 802.11i
- …

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）

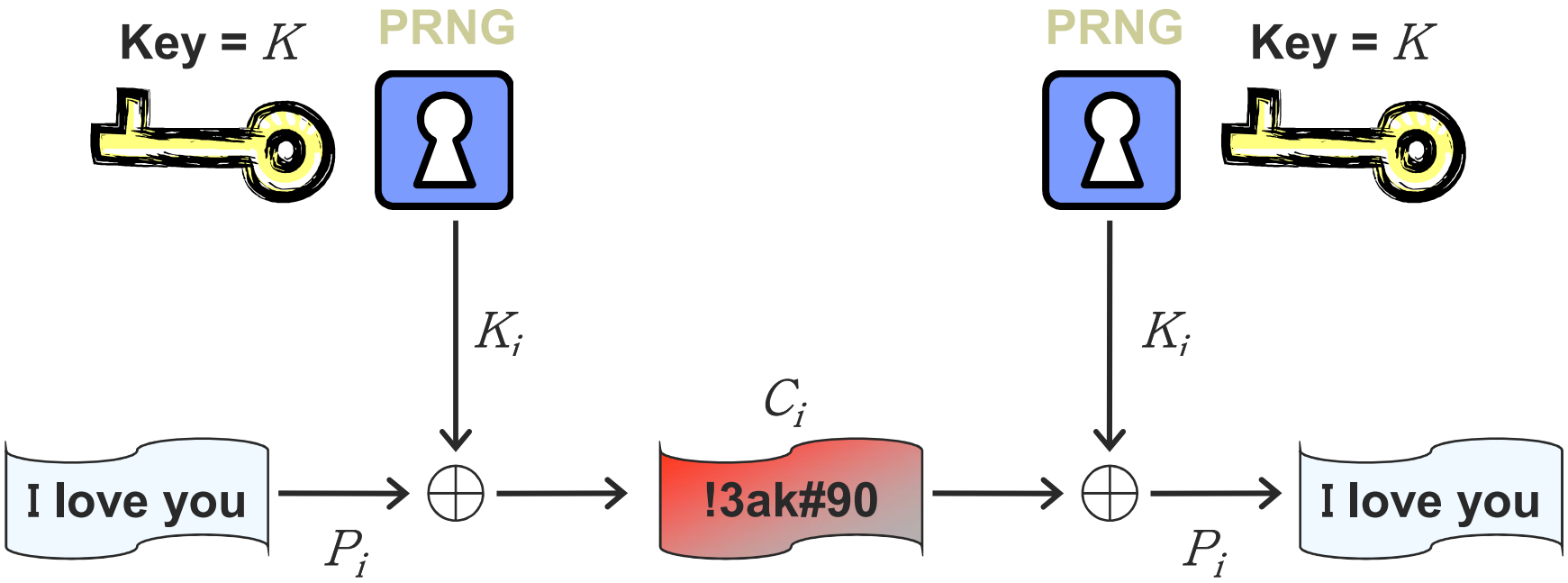
ところが、ここに落とし穴があった。IVは24ビットしかなく、連続して通信を行っていると早くて数時間で1巡してしまう。また、無線LANで送信されるパケットの最初の部分はつねに同じパターンが使われているのである。つまり、IVが何巡かするまでパケットを監視しつづけていれば、暗号鍵が解読できてしまうのだ。

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

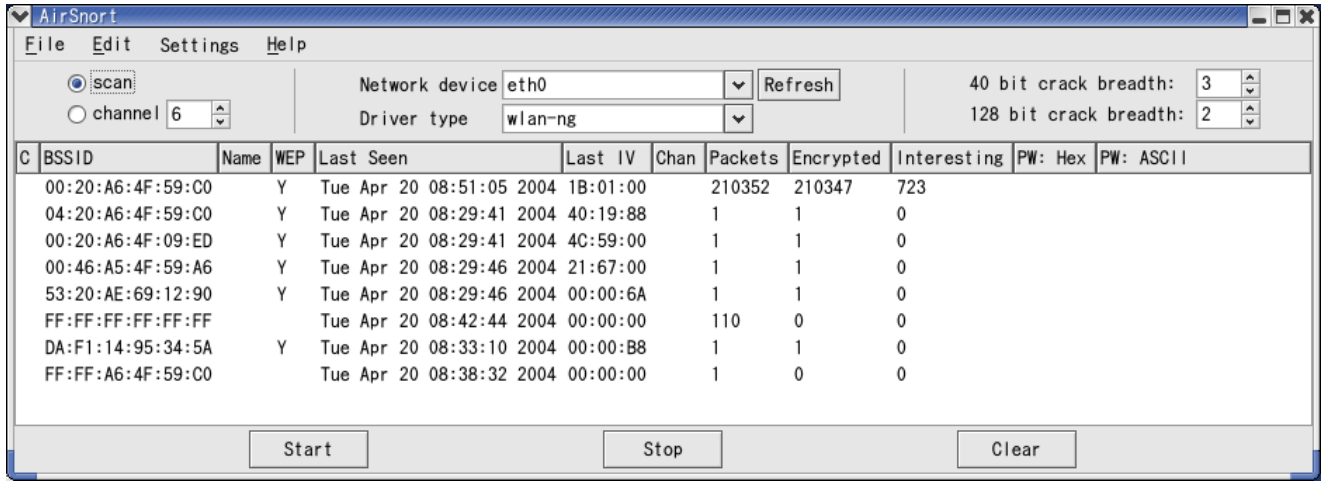
同じIVを使うと何が起こるか？

- 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

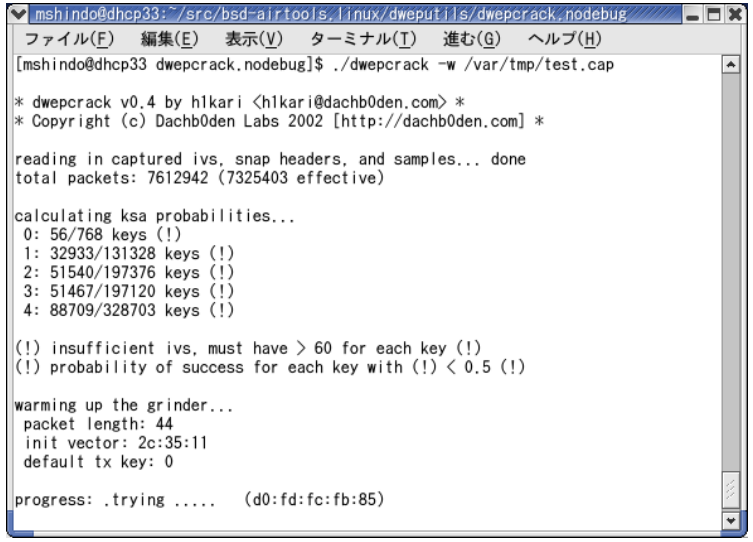
本当の脅威　～ キーリカバリー攻撃 ～

- S. Fluhrer, I. Mantin, A. Shamir, *Aug. 2001*
- 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！

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

- FMS を最適化した「統計的攻撃」
- FMSより遙かに良い性能を示す！
  - FMS は数百万パケットを集める必要性がある
  - Korek では50万パケット程度で済む
- 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

2007年4月、さらに衝撃が！！

- 『104bit WEPが60秒で破れる？！』
  - ドイツ、ダムシュタット工科大学の研究者（Eric Tews, Ralf-Philipp Weinmann, Andrei Pyshkin）による発表
  - http://eprint.iacr.org/2007/120.pdf
  - 通称「PTW攻撃」
- 40,000フレーム収集すると、50%の確率でWEP鍵をリカバーできる
  - 802.11a/gでactive attackすれば60秒以下
  - 802.11n（MIMO）なら30秒を切る可能性も！

PTW攻撃の実装

- aircrack-ptw
  - PTWのアルゴリズムを入れたツール（PoC）
  - ARP injectionにはaircrack-ngを使用
  - 近々aircrack-ngに取り込まれる予定

ということで・・・

- まじめにGood Bye WEP!
  - まさにWhat on Earth does this Protect?
  - 802.1X の rekey さえ追いつかない
  - 無線が高速化すると、さらに短い時間で破れるようになる
- 本気でHello WPA / WPA2（aka 802.11i）

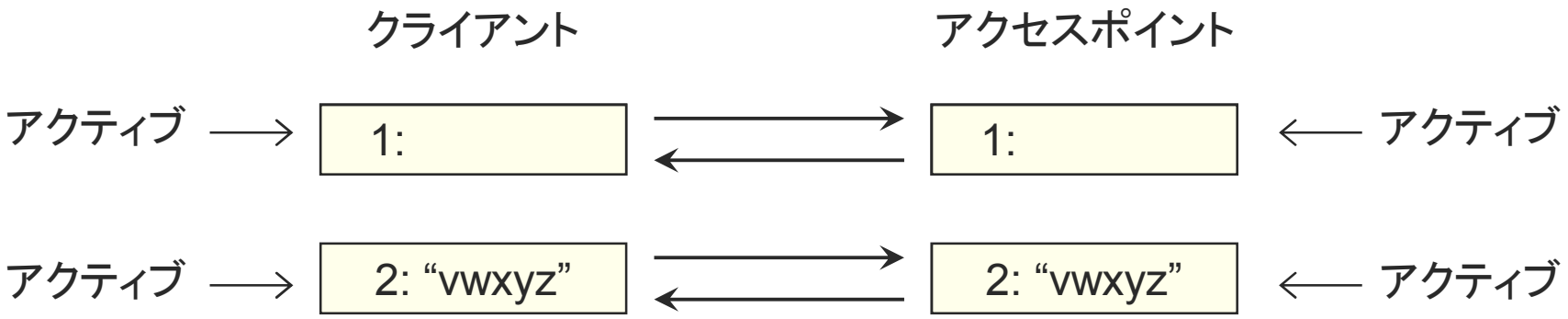
と、言いたいところですが、

- WEPを使っている皆さん、まだWEPを使っている理由は何ですか？
  - WEPがそんなに弱いなんて知らなかった？
  - 知ってはいたけど、そんなに気にはしていない？
  - ニンテンドーDSを持っているから？？☺

なぜ WEP 鍵は4つあるの？

- 多くのアクセスポイントは WEP 鍵を4つ設定することができる
  - 1つでも動くの？
  - 4つ設定したほうがより安全？

答： 鍵の変更をしやすくするため

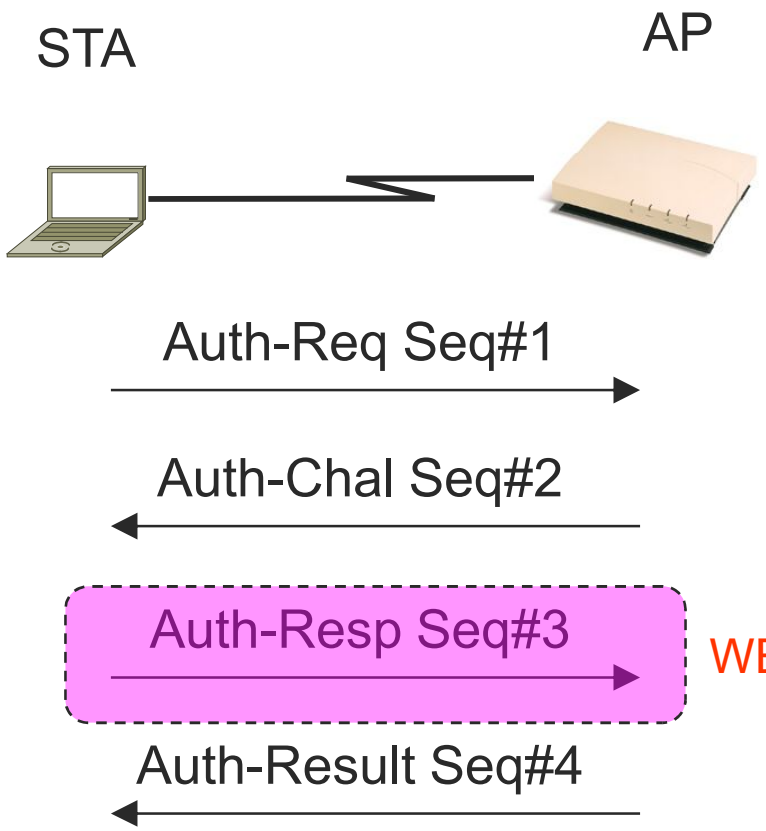


① クライアント、アクセスポイント共に 1: “abcde” で通信している。
② アクセスポイントに 2: “vwxyz” を追加（ただし、まだ、アクティブな鍵は 1: のまま）。
③ クライアントに 2: “vwxyz” を追加し、アクティブな鍵を 2: に変更。
④ 全てのクライアントで 3) までの設定が終了したら、アクセスポイントのアクティブな鍵を 2: に変更。
⑤ アクセスポイント＆全てのクライアントが 2: “vwxyz” で通信しているので、鍵 1: “abcde” を削除（この時点で鍵 1: “abcde” から鍵 2: “vwxyz” への変更が完了！

アクセスポイントでの認証

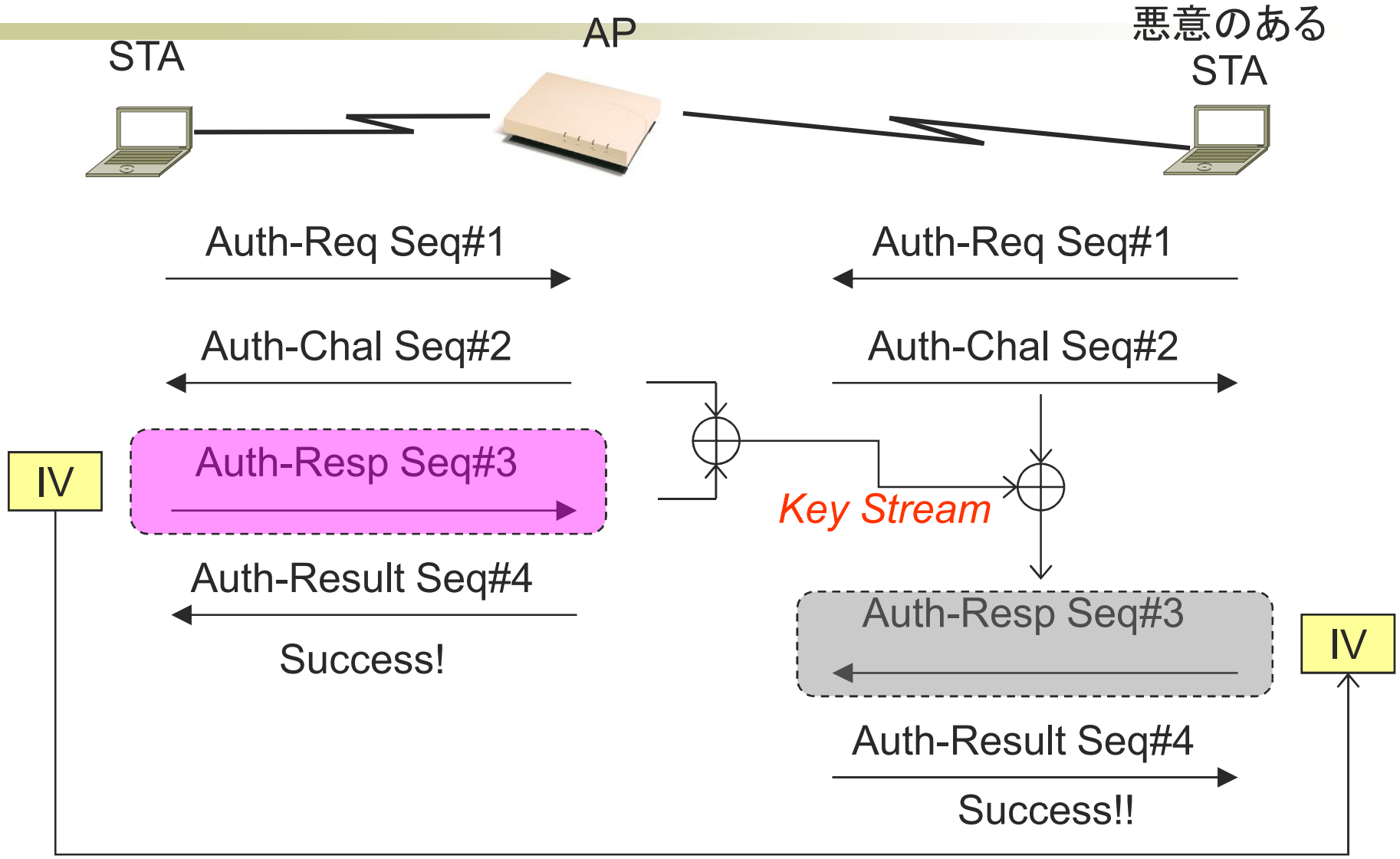
どれを使うべきなの？

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

こらあかん・・・



結論

- “する(シェアードキー認証)” より “しない(オープン認証)” ほうが安全！
- WPA / 802.11i ではオープン認証を使うことになっている

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

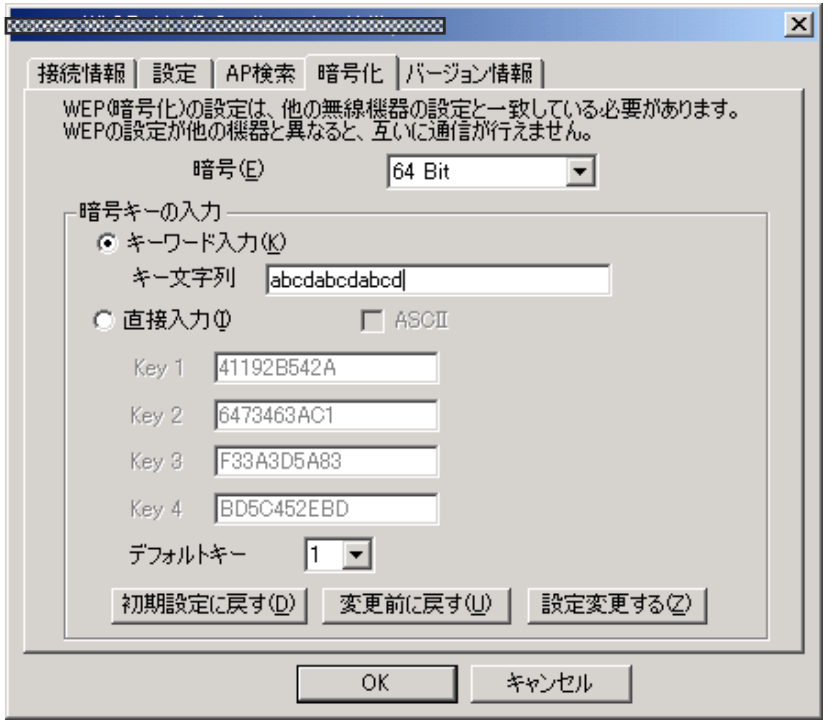
（B誌、2004年9月）

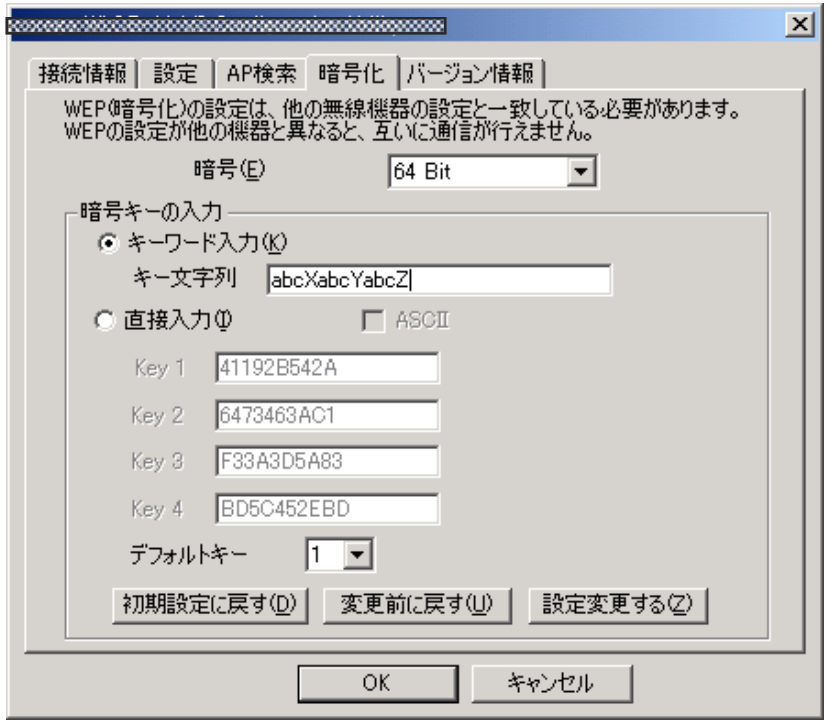
そりゃないっしょ！

- 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
      - 104 ビットの強度を得るには36.8文字必要
  - 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

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





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、PTW等の統計的攻撃

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

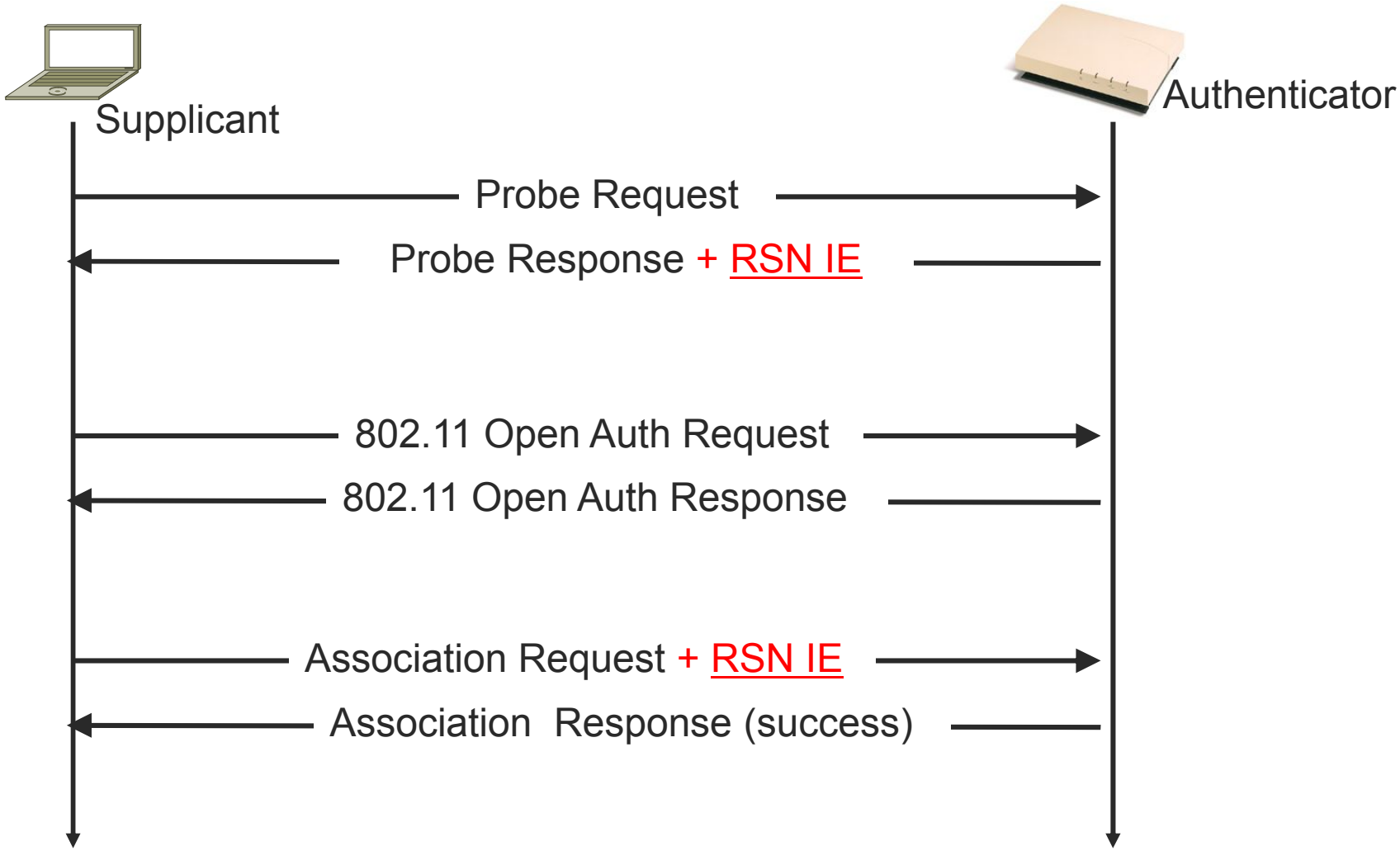
WPA (Wi-Fi Protected Access)

- 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”

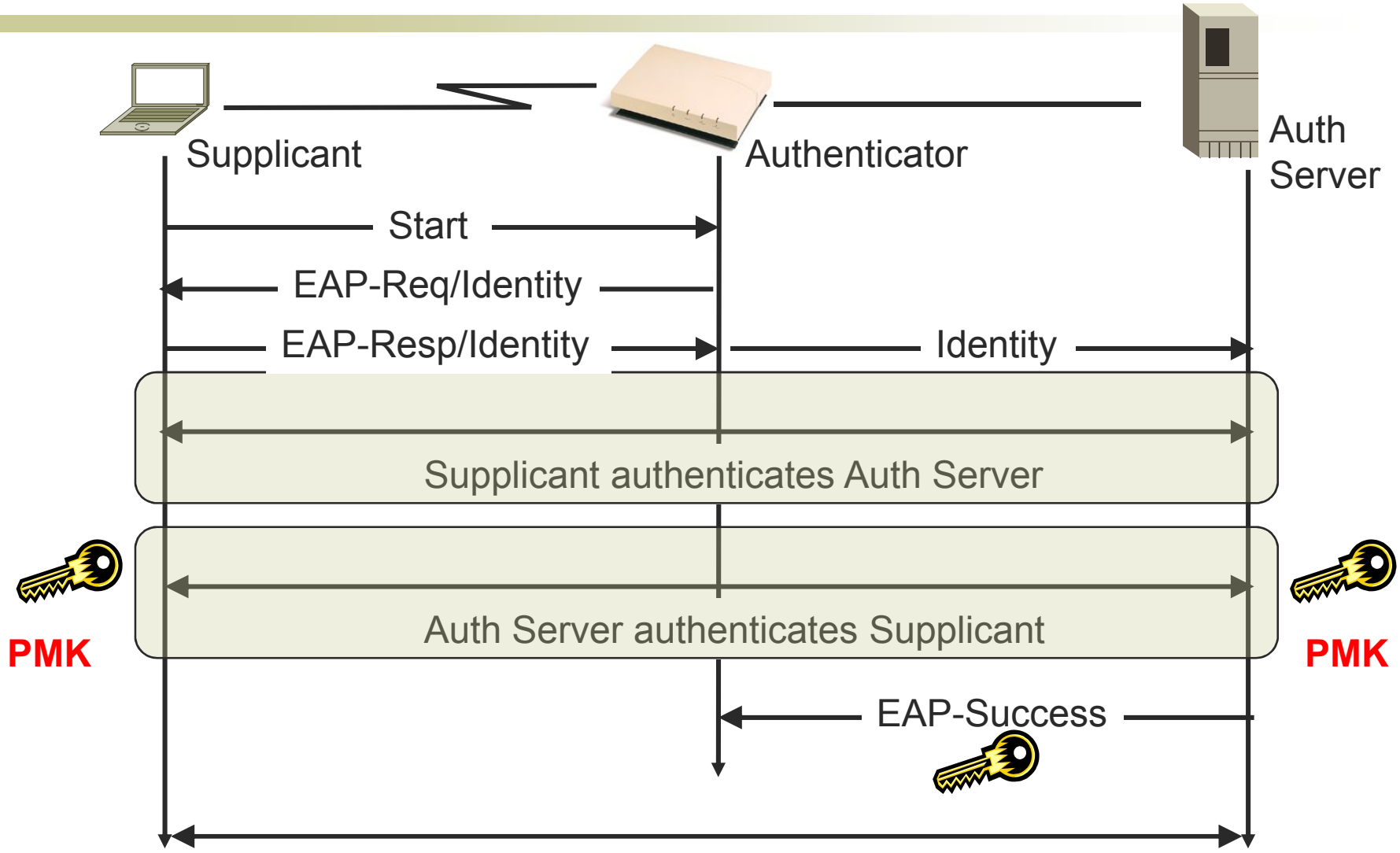
WPA ステップ

- 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

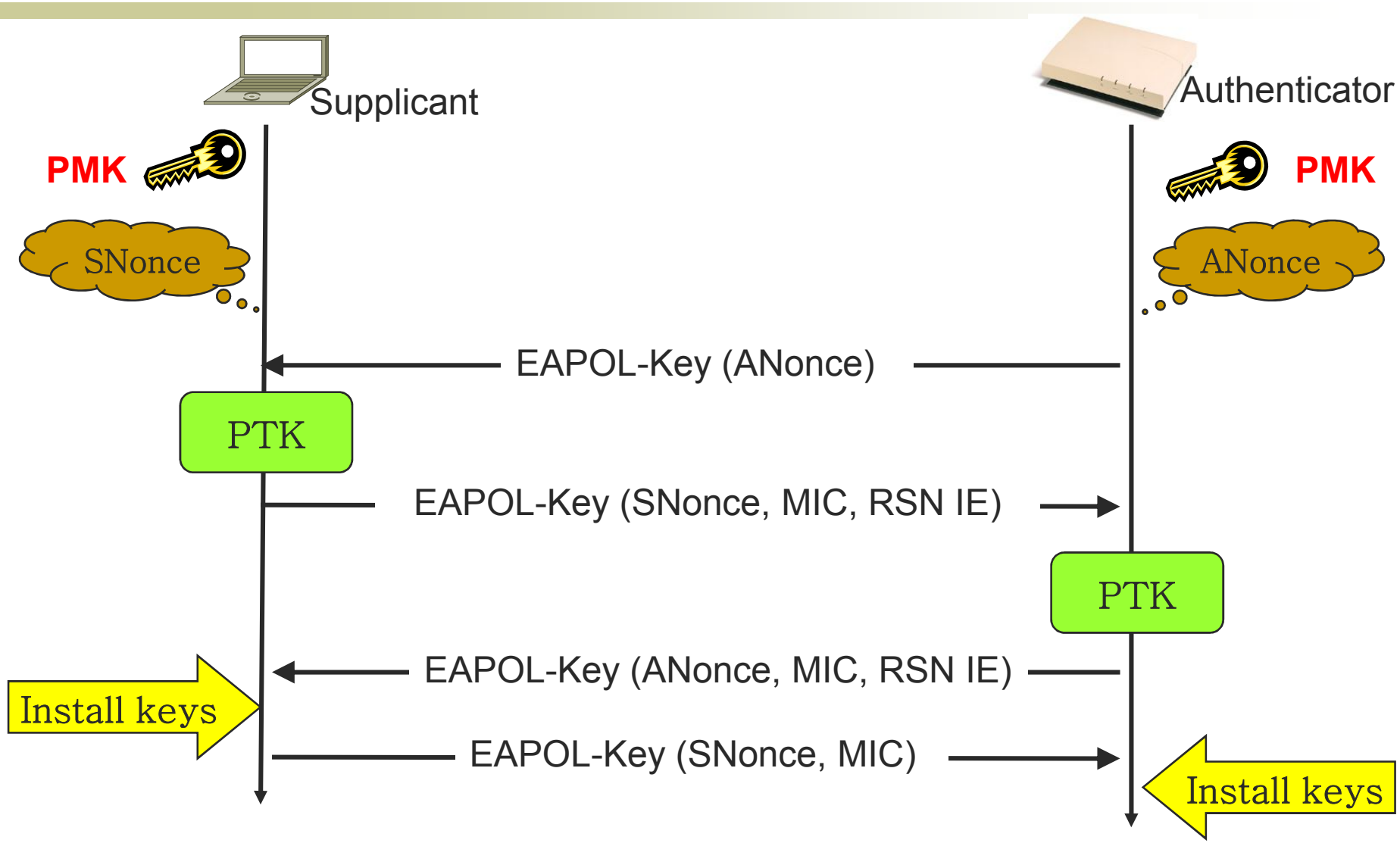
アソシエーションとケーパビリティーの確認



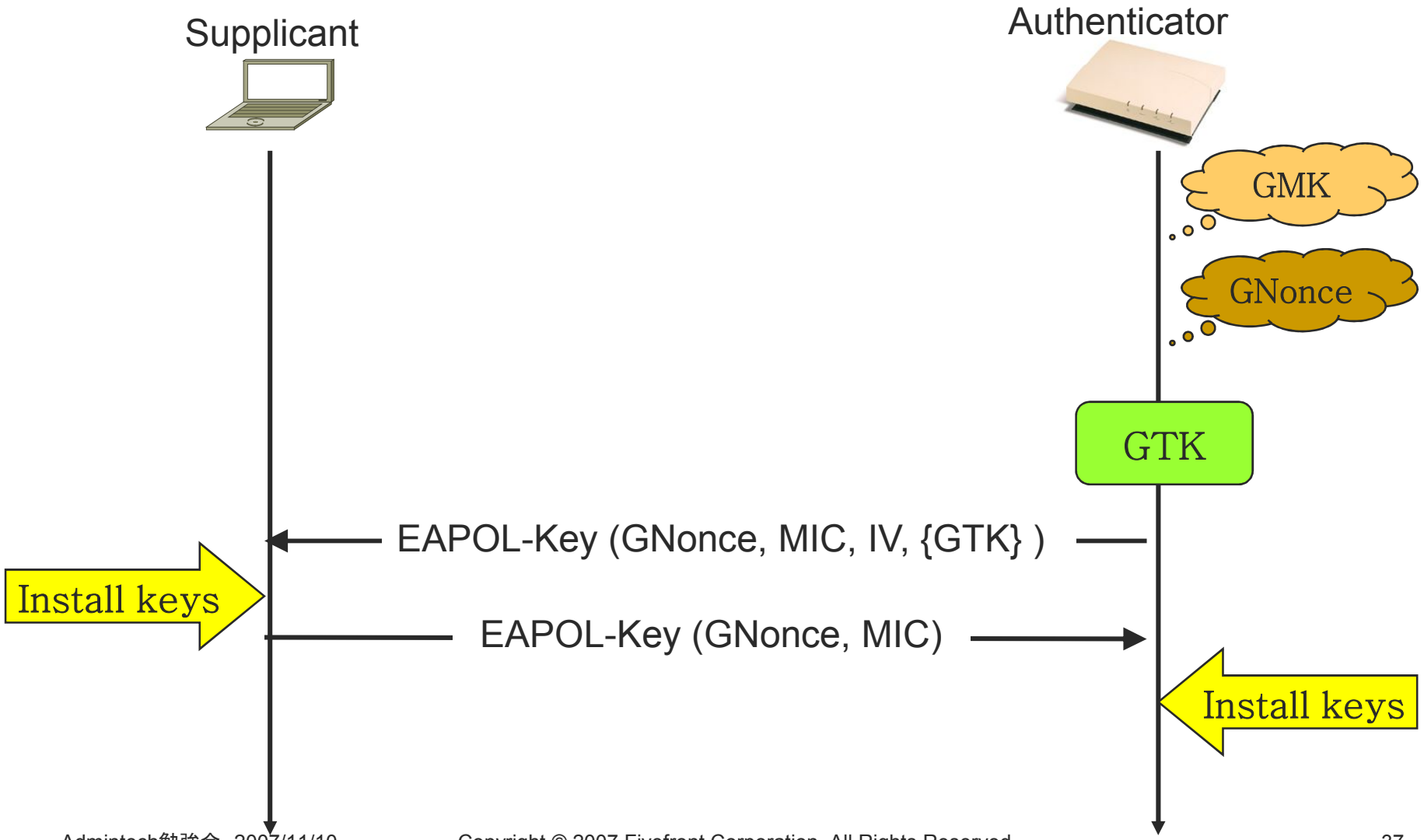
802.1X 認証と PMK の配布



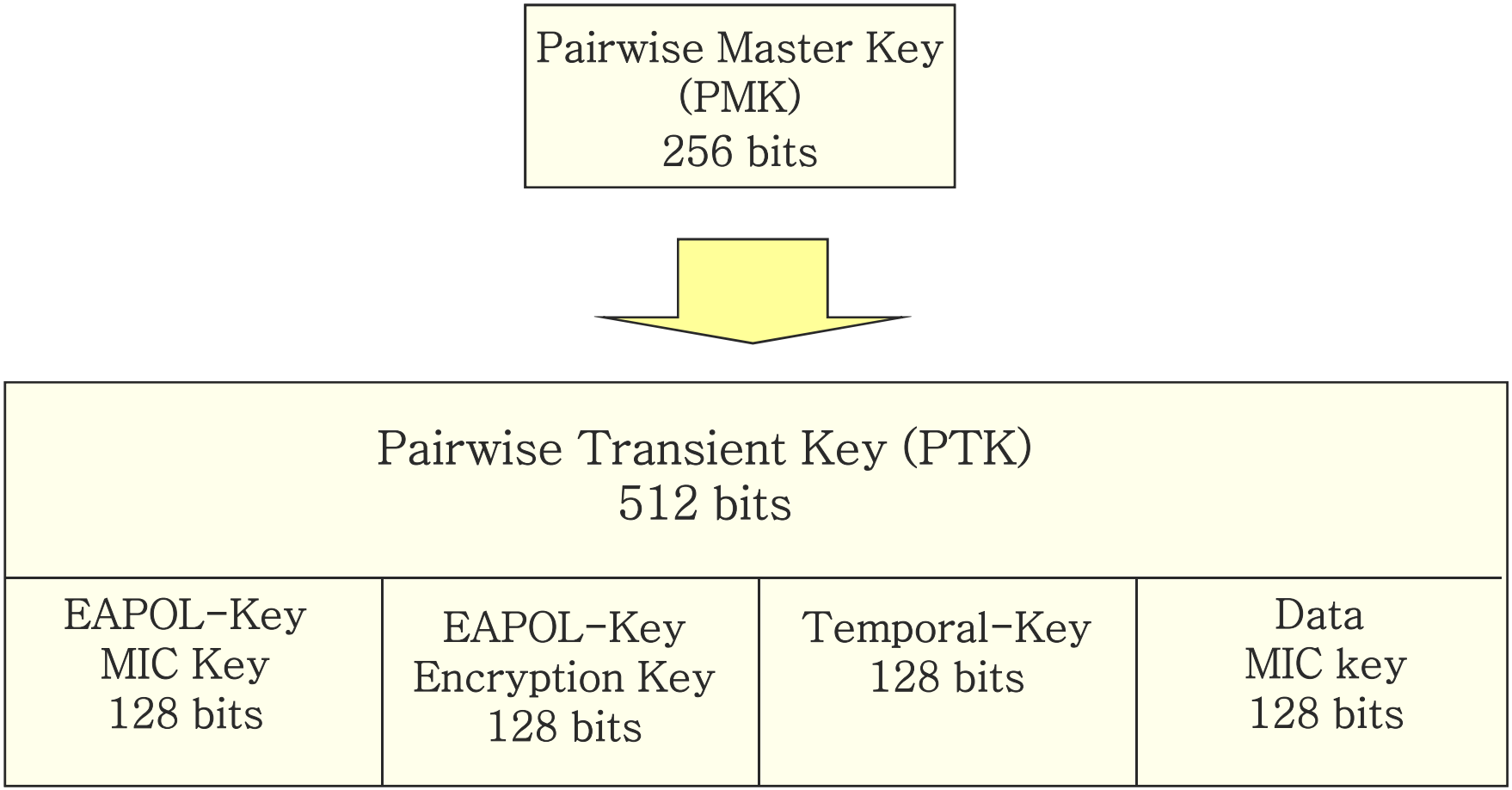
Temporal Key の導出 ～ 4 way handshake ～



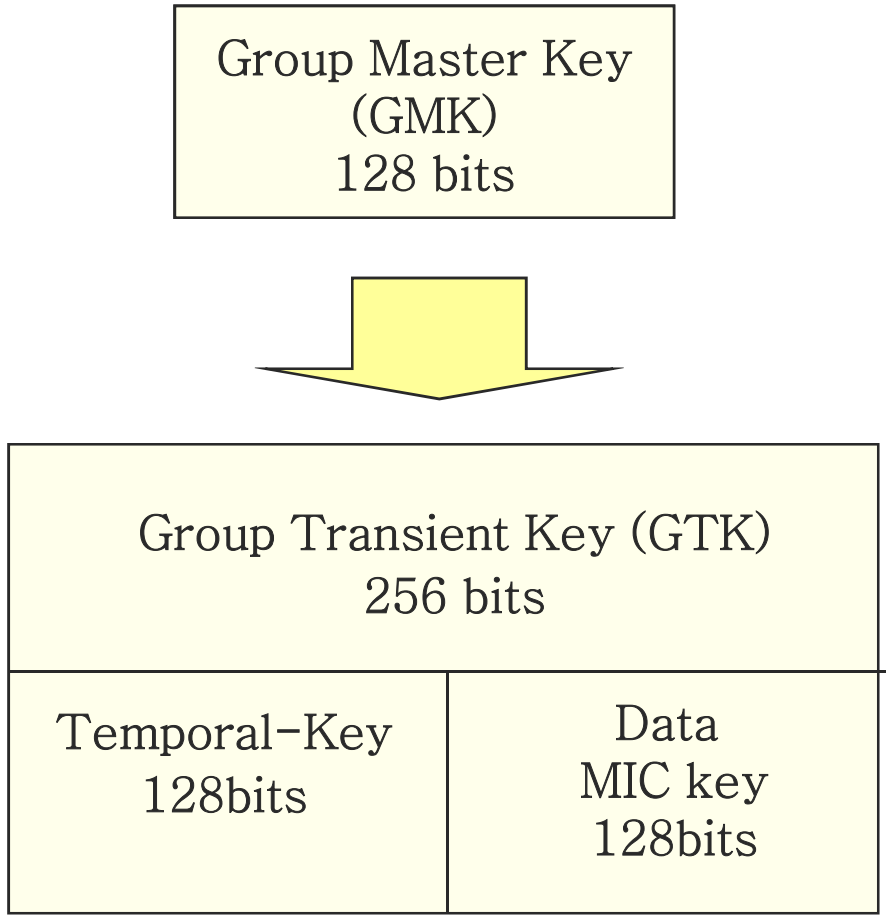
Group Key の導出 ～ 2 way handshake ～



Pairwise Key Hierarchy (for TKIP)



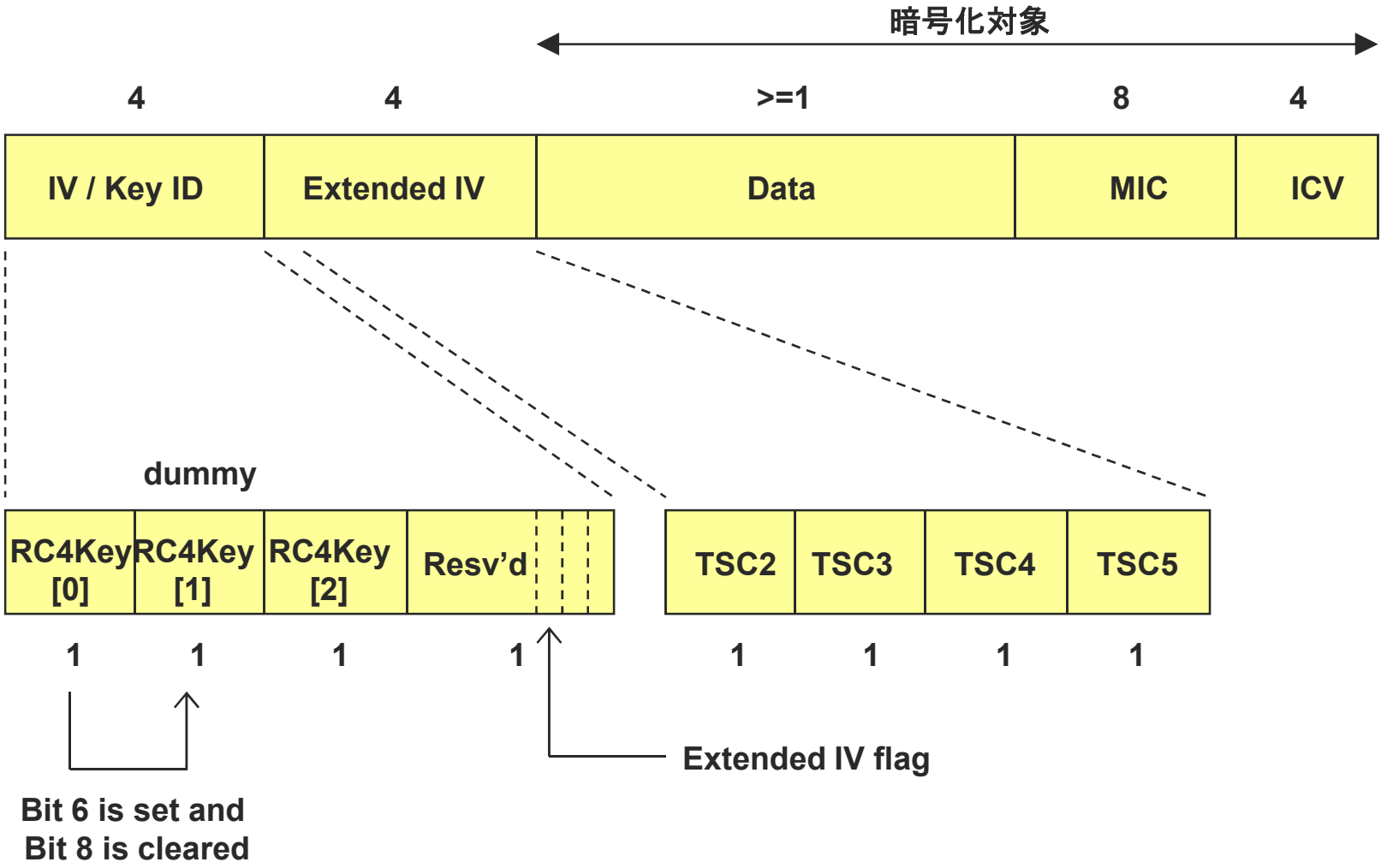
Group Key Hierarchy (for TKIP)



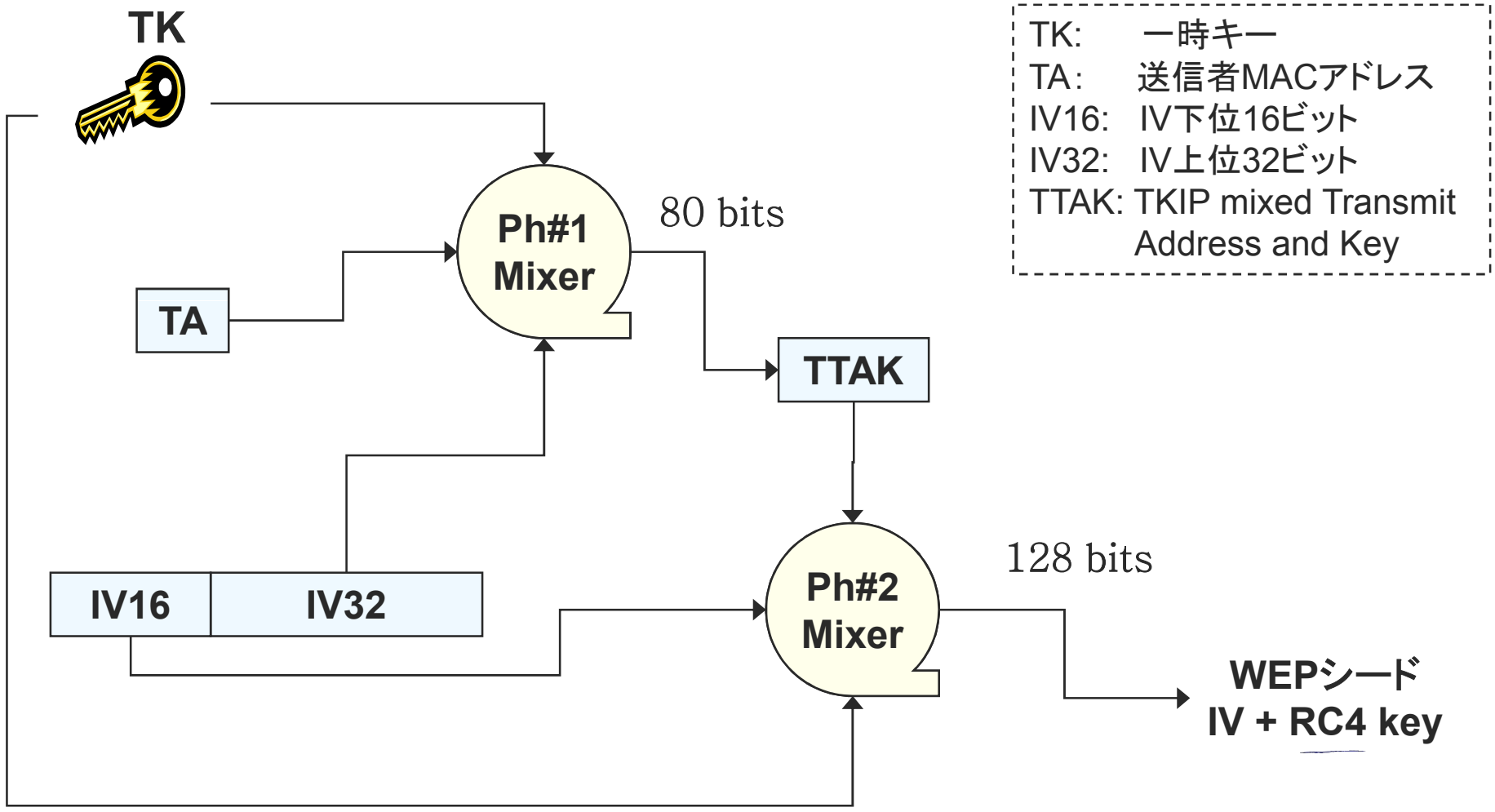
TKIP (Temporal Key Integrity Protocol)

- 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)

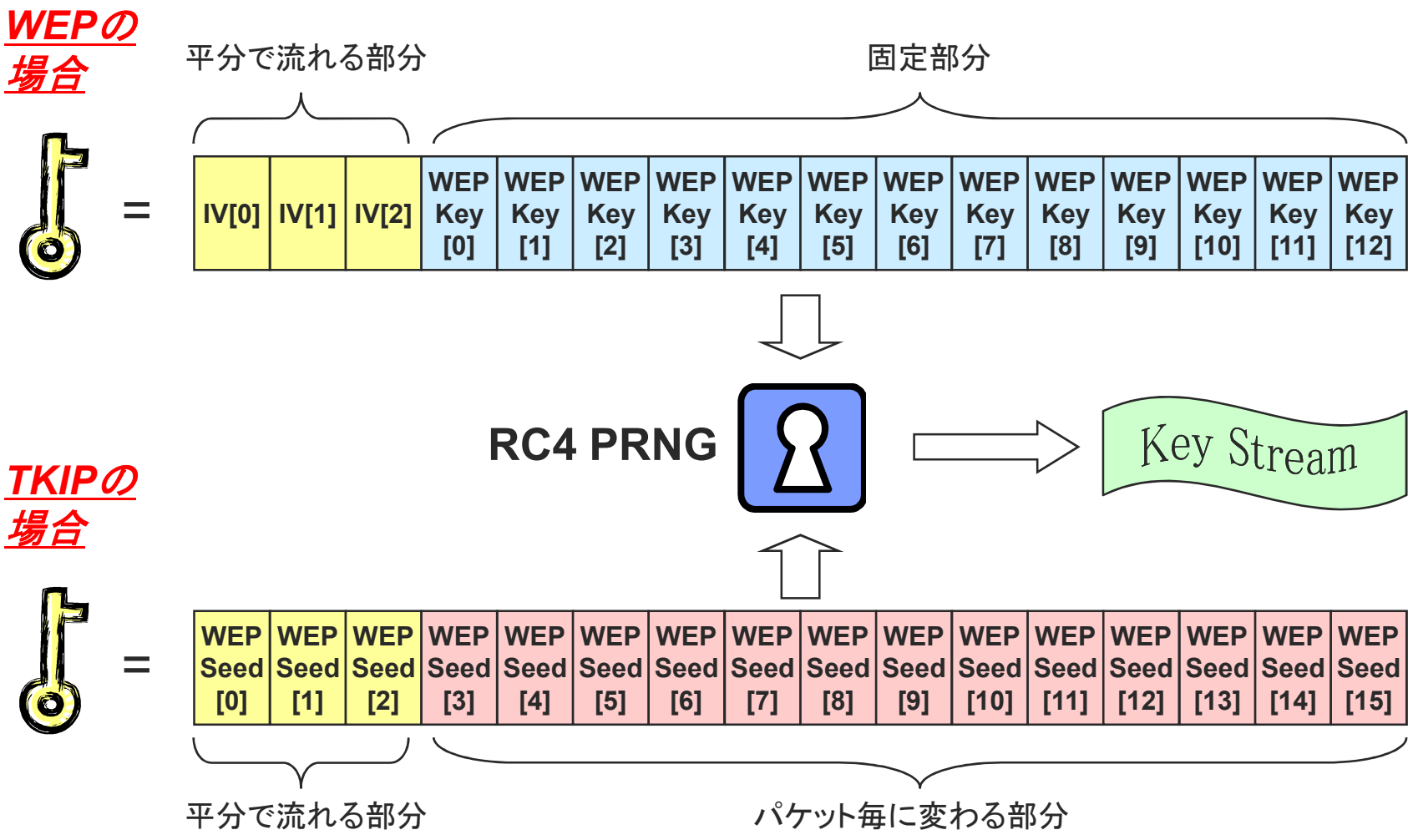
TKIP Frame Format



Per-packet-mixing function



WEP vs TKIP with RC4



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)

PBKDF2

- PSK = PBKBF2(*PassPhrase*, *SSID*, *SSIDLength*, *n*, *b*, *alg*)
  - where *n* = 4096, *b* = 256, *alg* = HMAC_SHA1
- RFC2898でも規定

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

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

Michael に DoS 攻撃は可能？

Michael の counter-measure を利用した DoS 攻撃は可能？？

答： 可能です。でも・・・

- 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

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)

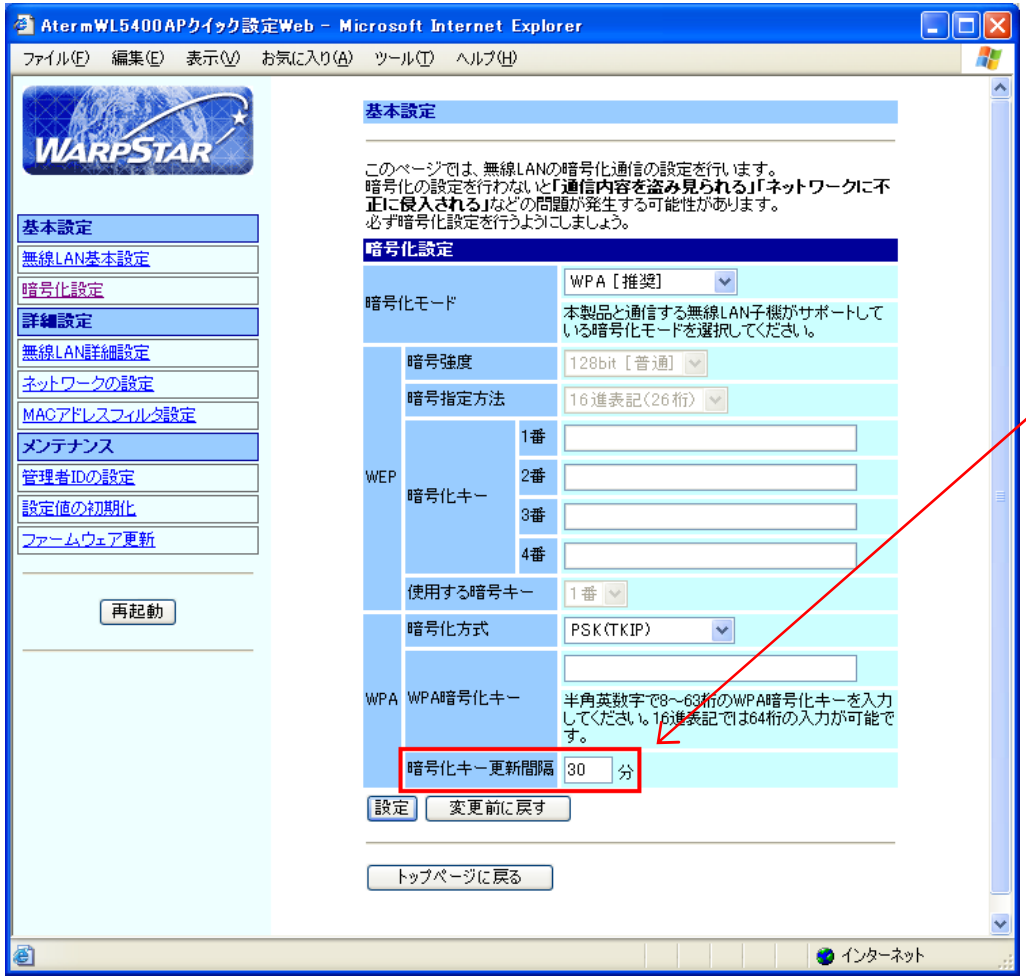
- 鍵の配布メカニズムがない
  - 802.1X による鍵配布
  - PSK では未解決
- IV の空間が小さい（i.e. 24bit）
  - 48 ビットに拡張された
- リプレイ攻撃に無力
  - IV の増やし方が規定され、検出できるようになった
- FMS、Korek、PTW 等の統計的攻撃
  - TKIP の key mixing function で IV と生成されるKey Streamの関連性を（ほぼ？）なくした

典型的WPA(TKIP)の説明

名称	方式	強度	キーの長さ	概要
WEP	---	64bit	5文字/10桁(16進数)	古い機械などとの互換性があるが解読の恐れがある
		128bit	13文字/26桁(16進数)	解読方法は知られているがキーが長いため時間がかかる
WPA	TKIP	---	任意	キーを一定間隔で変更することで安全性を確保
	AES	---	任意	最新の暗号化アルゴリズムを利用した強固な方式

(Web記事B、2007年7月)

TKIPの設定画面

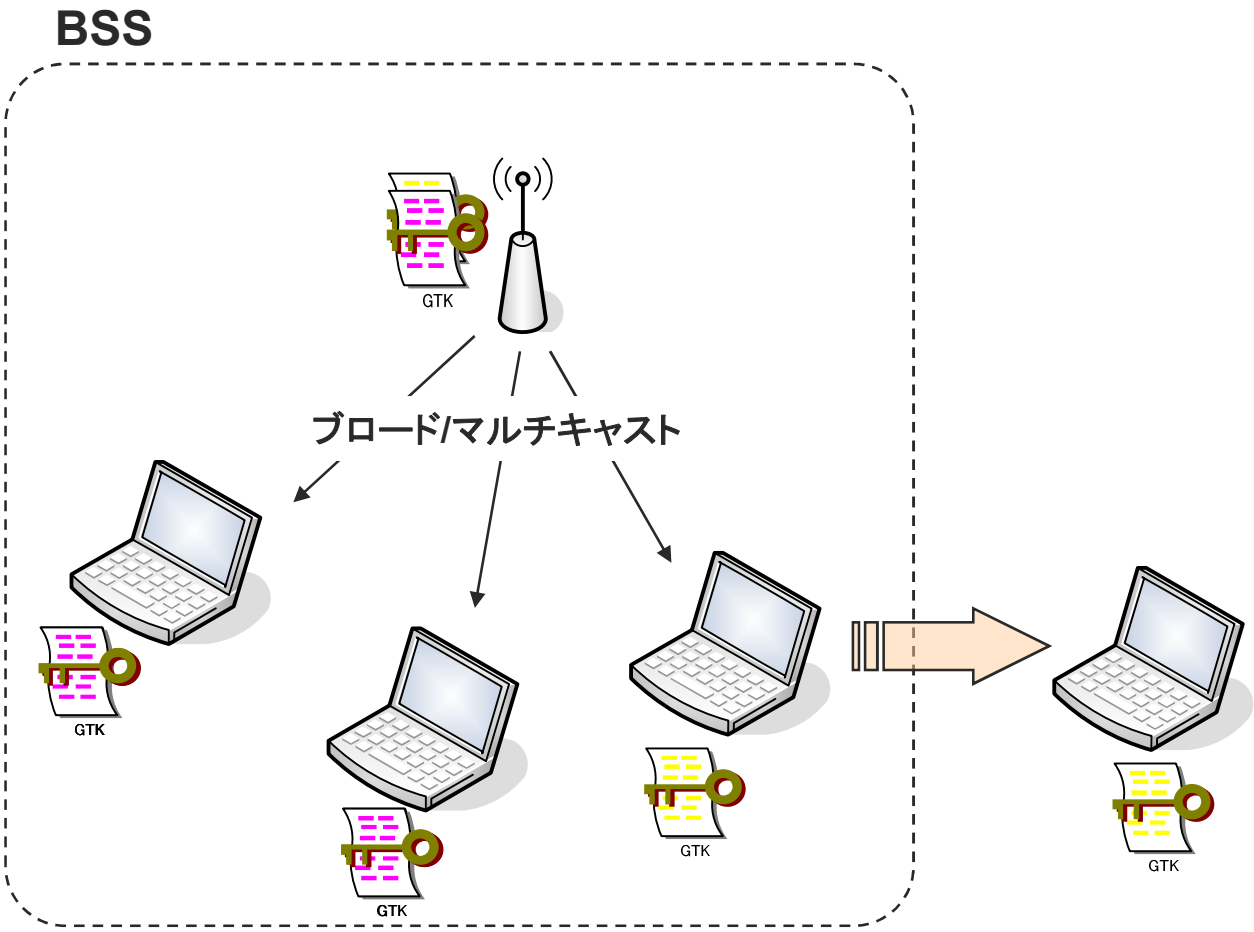


確かにWPA(TKIP)には鍵の更新間隔に関するパラメータが追加されている！

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！

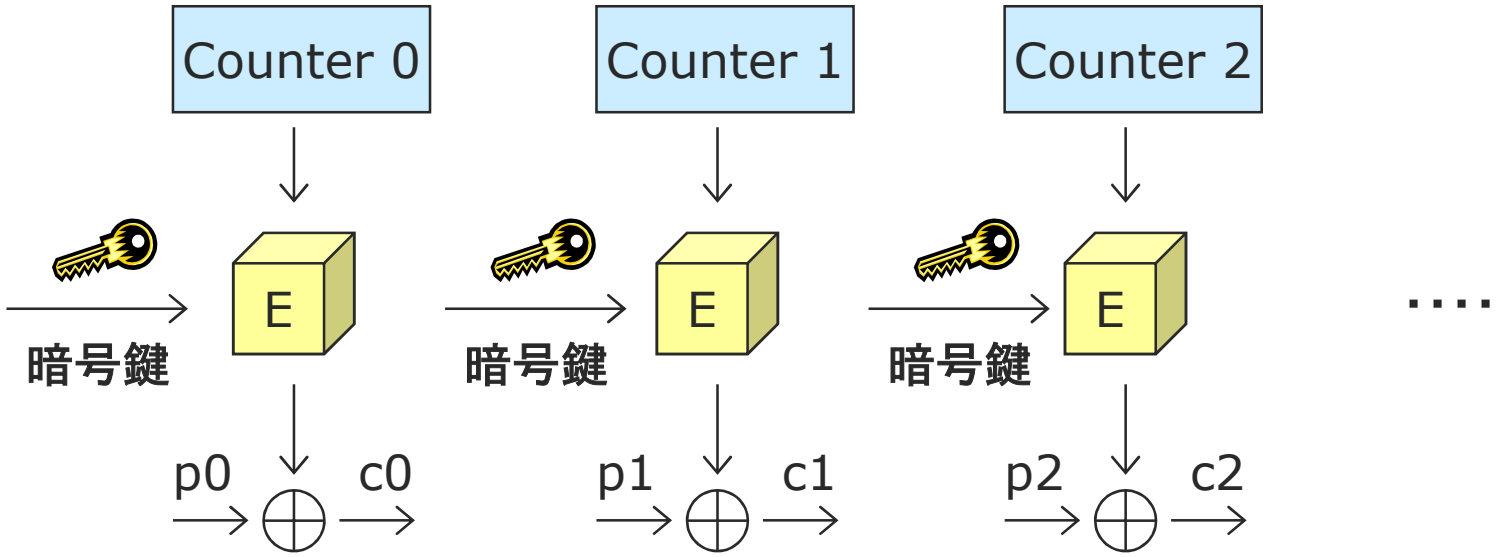
IEEE 802.11i

- 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

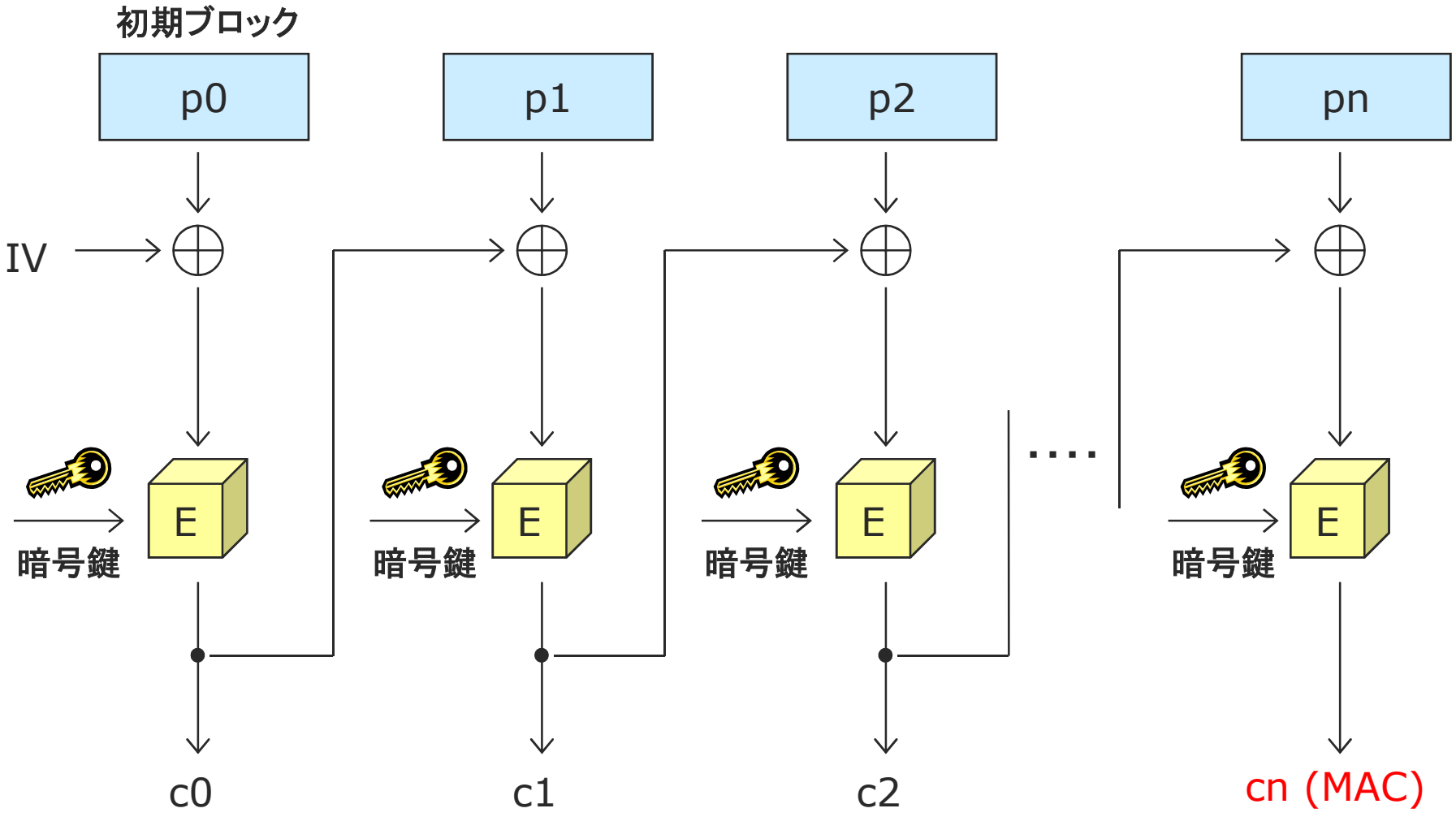
Counter-Mode



- 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



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社以上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

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

チェック

- あなたの正解率はどれくらい？
  - 70点以上： 

  - 50点～70点： 

  - 50点以下：