802.1X **相互接続実験報告書**

第 1.0 版

2003 年 4 月 22 日

NPO 日本ネットワークセキュリティ協会
2002 年度 相互接続ワーキンググループ

目次

1　はじめに

無線 LAN は 1998 年に規格 IEEE802.11b が決定し、1999 年ごろより製品がリリースされ普及が始まっている。規格成立当初よりセキュリティの面で問題があるとの指摘があったものの、多くの無線 LAN 環境の利用者は今日にいたるまで根本的にセキュリティの問題を解決することができなかった。今まで無線 LAN のセキュリティは一般には知られずにいたものの、"AirSnort"を始め攻撃ツールとして使用可能なソフトウェアの露出により急速に危機感が広まっている。

無線での LAN 環境構築は有線での構築と異なりいくつかの現象について配慮しておかなければならない。無線の電波は壁や窓ガラスなどで弱められるもののそのまま通り抜けて広がってしまう。このためセキュリティを維持するために通信を開始する前に利用者の認証(確認)と暗号によるデータ保護の準備を済ませなければならない。

現在、無線 LAN 機器が低価格化するとともに一般企業だけでなく金融機関や政府機関でも導入されることが増えた。一方、企業や公共団体で無線 LAN の脆弱性を指摘される事件が起きるなど無線 LAN のセキュリティが注目を集めている。

現在の無線 LAN セキュリティ技術は不十分である上に運用が難しい。無線 LAN の機器レベルで解決するプロトコルがそろっていないためである。WEP による暗号処理は現時点でセキュリティを構成する重要な要素であるが、全ての AP とクライアント PC に設定を実施しなければならないなど、その運用は大きな作業コストを伴う。さらに WEP はその技術に問題が指摘されている。このため単純な固定鍵の WEP ではある程度の攻撃スキルを持つ者の前には無意味なものであるとされている。

WEP の問題はおおよそ次の 3 点である。WEP には RC4 が使われているが、これに使われる IV の脆弱性があること。WEP は AP ごとに設定するため、複数の PC、複数のユーザが知ることができ鍵の情報を漏洩しやすいこと。WEP の鍵は固定鍵であり更新されることはまず無いために、鍵を計算によって解く時間が十分にあること。これらの問題は十分に長い鍵すなわちより強固であるはずの WEP 鍵でも同様の結果となる。

根本的なセキュリティの問題を解決するためには新しい技術の導入が必要だ。現在、セキュリティのキーワードとして IEEE802.1X、WPA、IEEE802.11i などが注目を集めている。しかし、これら新しい技術はどれも決定されたばかりか検討中のものが多く、現時点で実際に採用されている新しいセキュリティ技術は IEEE802.1X かメーカ独自の方式である。この IEEE802.1X はセキュリティの基礎となる個人(機器)認証を実現するプロトコルである。さらに、IEEE802.1X による認証手順のうち X.509 電子証明書を使う EAP-TLS、PEAP、EAP-TTLS では認証データを暗号により保護する機能を持ち、その暗号鍵を WEP 鍵の配信に使用することができる。IEEE802.1X を取り入れた無線 LAN 環境の構築では PKI、RADIUS サーバ、アクセスポイント、サーバとアクセスポイントのネットワーク構成、クライアント PC の無線 LAN カード、クライアント PC で動作する認証クライアント(サプリカント)の組み合わせとなる。

このように 802.1X では構成要素が複数あり、各要素で実装の基準(基準とした文書のバージョンや解釈)が異なると組み合わせによっては障害となる可能性がある。同一種類の機器同士の差(異なるメーカの AP)、推奨外の組み合わせによる問題(特定の RADIUS サーバと AP)などの組み合わせ問題が考えられる。実際の機器を使って接続開始動作、想定される環境下での継続動作を検証することによりこれらの問題の有無を確認し、解決策を図ることを目的とした。本実験では IEEE802.1X を構成する各要素を実際の構築を通じて理解し、実際の運用で問題となりうる項目について実機を用いて検証を行った。また、本報告書はそれぞれの要素となった IEEE802.11 無線 LAN、無線 LAN における 802.1X とその RADIUS サーバを解説するとともに、実験結果より各問題への考察を行った。

本報告書ではこれ以降、それぞれの用語を以下のように省略して記載する。

IEEE802.1X → 802.1X
IEEE802.11 → 802.11
IEEE802.11i → 802.11i
IEEE802 → 802

また、802.1X では「認証サーバ」を用語として定めているが、実際に「RADIUS サーバ」がその役割を果たすため「RADIUS サーバ」と表記を統一した。

[関]

2　IEEE802.11 無線関連技術

2.1　IEEE802.11 基本

IEEE802.11-1999 (802.11)は、IEEE802 LAN/MAN 委員会の無線 LAN Working Group で標準化されている無線 LAN の仕様で、大きく分けて MAC 層と物理層から構成される。ISM(Industrial, Scientific and Medical)バンドと呼ばれる国際的に免許不要での使用が許されている 2.4GHz 帯の電波を用い[1]、1Mbps と 2Mbps の通信速度を実現している。802.11b は、802.11 の中の一つのタスクグループとして策定された拡張仕様で、2.4GHz 帯を用いて 5.5Mbps と 11Mbps の通信速度を実現している。また同様に、802.11a も拡張仕様のひとつであり、5GHz 帯の電波を用いることによって 54, 48, 36, 24, 18, 12, 9, 6Mbps と、802.11b と比べてより高速でフレキシブルな通信速度を実現している[2]。802.11 では、802.11a 仕様策定済のタスクグループも含めて現在(2003/01 時点)、以下のようなグループが存在する。

Task Group A：5GHz 帯を用いた 6～54Mbps の速度を実現
Task Group B：2.4GHz 帯を用いた 5.5～11Mbps の速度を実現
Task Group C：IEEE802.1D (ブリッジ) への対応の追加
Task Group D： 各国対応のための IEEE802.11 仕様への追加的な要求事項
Task Group E：QoS 拡張
Task Group F：Access Point 間の通信
Task Group G：2.4GHz 帯を用いた 6～54Mbps の速度を実現
Task Group H： 欧州の Regulation 対応
Task Group I：TKIP/AES を用いたセキュリティ強化
Task Group J： 日本の Regulation 対応
Task Group K： 無線リソース管理
Task Group L: N/A
Task Group M：IEEE802.11-1999 の技術的・記述上の修正のためのメンテナンス

2.1.1　ネットワークモデル

802.11 の LAN の一つの島のことを BSS(Basic Service Set)と呼ぶ。802.11 のネットワークモデルは大きく分けて 2 つあり、俗にいうアドホックモードとインフラストラクチャーモードである。アドホックモードは、IBSS(Independent BSS)を利用する形態のネットワークを指し、最少 2 つのステーションから構成される。一方、インフラストラクチャー

[1] 仕様では物理層として赤外線も標準化されている。
[2] ただし、通信速度は物理層/MAC 層等とのオーバヘッドにより実際に使用可能な速度はこれらよりも低くなる。

モードでは、AP[3]と呼ばれるステーションがいることを前提とした BSS である。AP は、自分が運用しているゾーンに対してビーコンを送出し、AP を利用するステーションは AP との間でアソシエーションを行なう必要がある。どちらのモデルにせよ、BSS 内での通信には共通の BSSID が必要であり、IBSS では各々のステーションで任意の共通の BSSID を用い、インフラストラクチャーモードでは AP の MAC ID を BSSID とする。両者を、図 2-1 図 2-2 に示す。

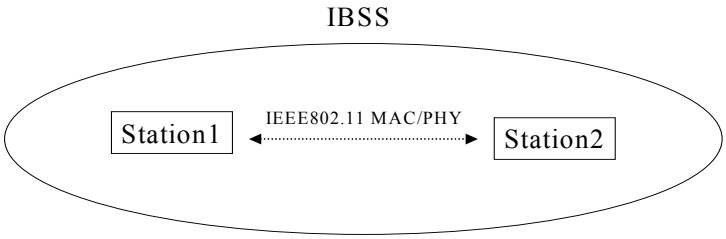

図 2-1 アドホック(IBSS)ネットワーク

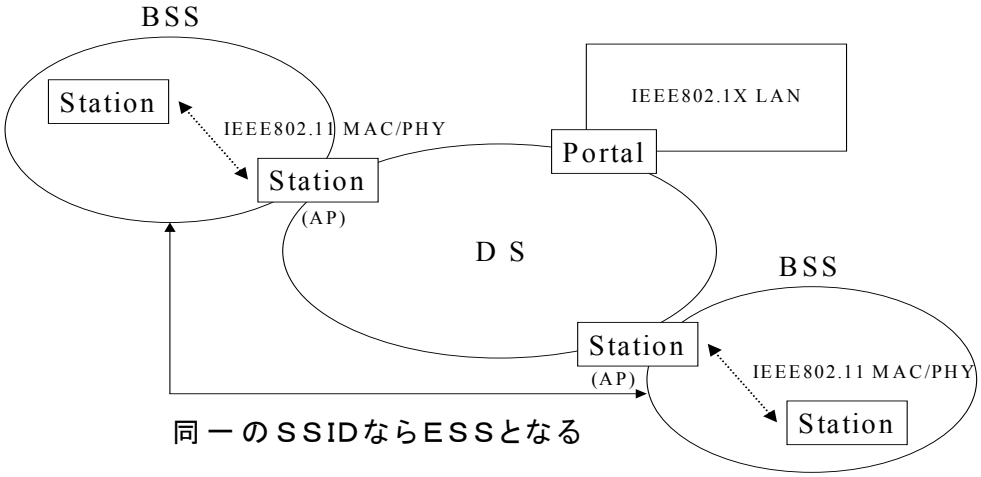

図 2-2 インフラストラクチャネットワーク

2.1.2 物理層

802.11 の物理層には、

2.4GHz 帯の電波を用いたスペクトル拡散（DS）

2.4GHz 帯の電波を用いた直交周波数分割多重（OFDM）

5GGHz 帯の電波を用いた直交周波数分割多重（OFDM）

2.4GHz 帯の電波を用いた周波数ホッピング（FH）

赤外線（IR）

---

[3] Access Point: 製品で言うところのアクセスポイントは、IEEE802.11 では Portal と呼ぶ。Portal とは、IEEE802.X で定義される LAN と接続するための機能(ブリッジ)のことである。

の仕様がある。ここでは、今回実験で使用したスペクトル拡散の方式のみについて説明し、その他は割愛する。

スペクトル拡散では、MAC 層から入力されたデータを、G(z) = z^{-7} +z^{-4} + 1 でスクランブルした後、2412MHz から 2472MHz までを 5MHz 毎に均等に割った 13 チャネルに 2484MHz を加えた総勢 14 チャネルの一つを搬送波として、DBPSK(Differential Binary Phase Shift Keying) または DQPSK(Differential Quadrature Phase Shift Keying)による変調を行なう。次に、11-chip Barker シーケンスを拡散信号として掛け合わせる 2 次変調により 11 倍の帯域（22MHz）に拡散して送出する。全チャネルで拡散符号が共通であるため、互いの干渉を完全に避けるにはスペクトル上での衝突を避ける必要があり、メインローブの衝突を完全に避けるには、少なくとも 5 チャネルのずれが必要である。また、802.11b では従来の 802.11 との互換性を保つために CCK(Complementary Code Keying)という変調を行なう。CCK では、5.5Mbps および 11Mbps において、4 ビットまたは 8 ビットの MAC から入力されるデータをかたまりとして扱い、2 ビットを DQPSK し、残りの 2 ビットまたは 6 ビットを拡散信号の選定に使う。受信側では、4 または 64 個の拡散信号との相関を求めてどの信号が使われたかを判断し、送信側で拡散信号の選定に使用した 2 または 6 ビットのデータを特定する。

2.1.3 MAC 層

802.11 の MAC の基本アクセス方式は、CSMA/CA(Carrier Sense Multiple Access w/Collision Avoidance) に ACK(Acknowledgment,) を加えた DCF(Distributed Coordination Function)である。また、802.11 の仕様としての DCF には、PCF(Point Coordination Function)も含まれるが、ここでは割愛する。802.11 の Carrier Sense には、物理的なものと仮想的なものがあり、物理的な Carrier Sense は物理層で実現されており、Carrier を検出すると MAC 層に知らせるようになっている。一方、仮想的な Carrier Sense は MAC 層で実現されており、RTS/CTS フレームを利用して実現され、主に隠れステーションへの対処に有効である。送信されたフレームを Carrier Sense で検出すると、DIFS(DCF Interframe Space)待ち、さらにステーション個別のランダムな Backoff Window スロット分待った後に、別のフレームを送出する。一方、フレームを受信したステーションは、送信元のステーションに SIFS(Short Interframe Space)の間に ACK を応答する必要があり、送信元のステーションは SIFS の間に ACK を受信しない場合は、そのフレームの再送信のスケジュールに入る。RTS/CTS を用いたアクセスでは、送信元のステーションはデータの送信に先立って RTS を送出し、受信側のステーションは CTS を応答する。これにより、そのほかのステーションは、受信側のステーションが ACK を送るまで(ACK が送られなければならない SIFS まで)メディアが busy であることを理解できる。

また、802.11 の MAC レイヤにおいて送受信するフレームには、マネージメント・コントロール・データの 3 種類が存在する。それぞれのフレームは図 3 に示すフレーム構成を

その基本とし、Frame Control フィールド中の Type および Subtype のサブフィールドをもとにフレームを一意に決定する。表１にそれぞれのフレームタイプ毎のサブタイプ一覧を示す。

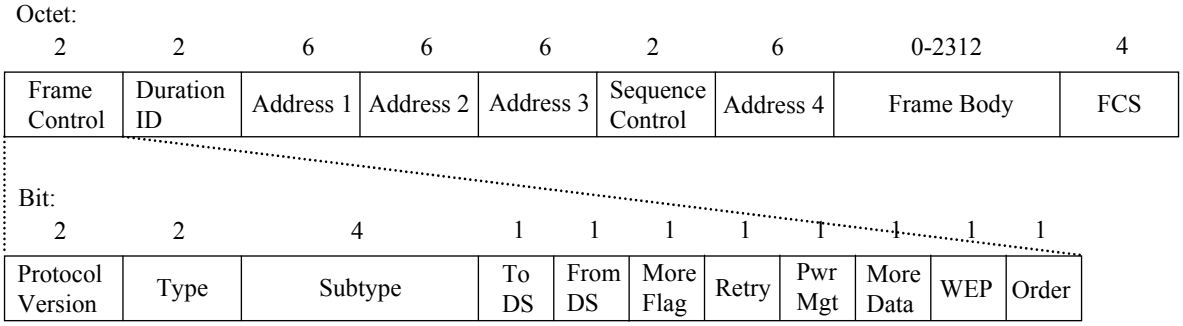

図 2-3MAC フレームのフォーマット

表１：フレーム一覧

マネージメントフレーム (00)	コントロールフレーム(01)	データフレーム(10)
Association Request（0000）	Power Save Poll（1010）	DATA（0000）
Association Response（0001）	RTS（1011）	DATA + CF-ACK（0001）
Re-Association Request（0010）	CTS（1100）	DATA + CF-POLL（0010）
Re-Association Response（0011）	ACK（1101）	DATA + CF-ACK + CF-POLL（0011）
Probe Request（0100）	CF-END（1110）	NULL Function（0100）
Probe Response（0101）	CF-END + CF-ACK（1111）	CF-ACK（0101）
Beacon（1000）		CF-POLL（0110）
ATIM（1001）		CF-ACK + CF-POLL(0111)
Disassociation（1010）		
Authentication（1011）		
Deauthentication（1100）		

それぞれのフレームはクラス 1～3 のいずれかのクラスに属しており、図４に示す状態遷移(状態 1～3)の中で使用される。状態 1 では Class 1 のフレーム Probe Request, Probe Response, Beacon, Authentication, Deauthentication, RTS, CTS, ACK, CF-END, CF-END + CF-ACK, Data (STA・STA 間通信)のみ送信可能で、状態 2 ではクラス 1 および 2 のフレーム(Association Request, Association Response, Reassociation Request,

Reassociation Response, Disassociation)を、状態 3 ではクラス３（Deauthentication, PS-Poll, Data (AP・STA 間, AP・AP 間通信)）を含むすべてのクラスのフレームが送信可能である。インフラストラクチャーモードでは、STA は Beacon または Probe Request の送信によって得られた Probe Response からアソシエーションすべき AP を決定し、認証フェーズ(状態 2)に入る。認証フェーズでは Authentication のやり取りによって認証を行い、アソシエーションフェーズ(状態 3)へと進む。アソシエーションフェーズでは Association Request と Association Response のやり取りを行い、アソシエーションを完了する。

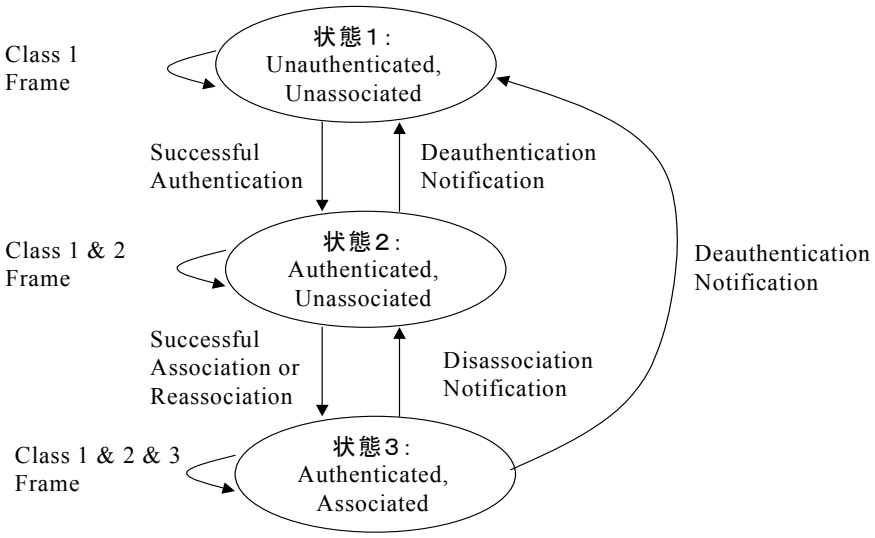

図 2-4 状態遷移

2.2 802.11 のセキュリティ機能

2.2.1 認証と暗号化

802.11 では、Open System と Shared Key の 2 種類の認証方法を規定している。認証処理はマネージメントフレームを用いたユニキャスト通信で行われ、インフラストラクチャーモードであればステーション[4]・AP 間で、IBSS モードであればステーション間で実行される。

2.2.1.1 Open System 認証

Open System 認証は最も単純な認証方法で、デフォルトの認証処理手順として規定されているが、実質認証処理を省略するための手順である。ステーションは、Open System 認証を要求する Authentication フレーム１を送信し、それ受け取った AP が Result コード "Success"を含む Authentication フレーム２を送信し、認証が完了する。

[4] PC など AP 以外の無線 LAN 接続機器

2.2.1.2 Shared Key 認証

Shared Key 認証は、それぞれのステーションが WEP で使用する共有鍵を持っているかを確認することによって機能する認証処理で、WEP が実装されている場合にのみ使用可能となる。共有鍵は予め安全な方法でそれぞれのステーションに設定されていることが前提とされ、認証処理では乱数から生成したチャレンジとそれを WEP で暗号化した暗号文の正当性により認証を行うチャレンジ・レスポンス型の認証処理である。ステーションは、WEP を使用するように設定されている場合のみ、共有鍵による認証処理を開始し、他の場合は Open System 認証を行う。ステーションは、Shared Key 認証を要求する Authentication フレーム 1 を送信し、それ受け取った基地局（ステーション）は自身が WEP をサポートしていれば、チャレンジテキストを含む Authentication フレーム 2 送信する。これを受信したステーションは、受信したチャレンジテキストを含む Authentication フレーム 3 を WEP で暗号化し、送信する[5]。これを受信した基地局は WEP で平文への復号化を行なった後 WEP ICV 値の正当性を確認し、正しければ"Success"の Result コードを含む Authentication フレーム 4 を送信する。

2.2.2 暗号化

802.11 では WEP(Wired Equivalent Privacy)と呼ぶ秘匿性機能を標準化している。WEP は、共通化鍵方式の暗号化である。データフレームまたはマネージメントフレームのうちサブタイプが Authentication であるフレームのみに適用可能であり、暗号化を行う場合は MAC ヘッダの WEP フィールドのビットを"1"に設定する。WEP により暗号化されたフレームのフォーマットの一例を図 5 に示す。Data(PDU)部分が暗号化の対象となる。

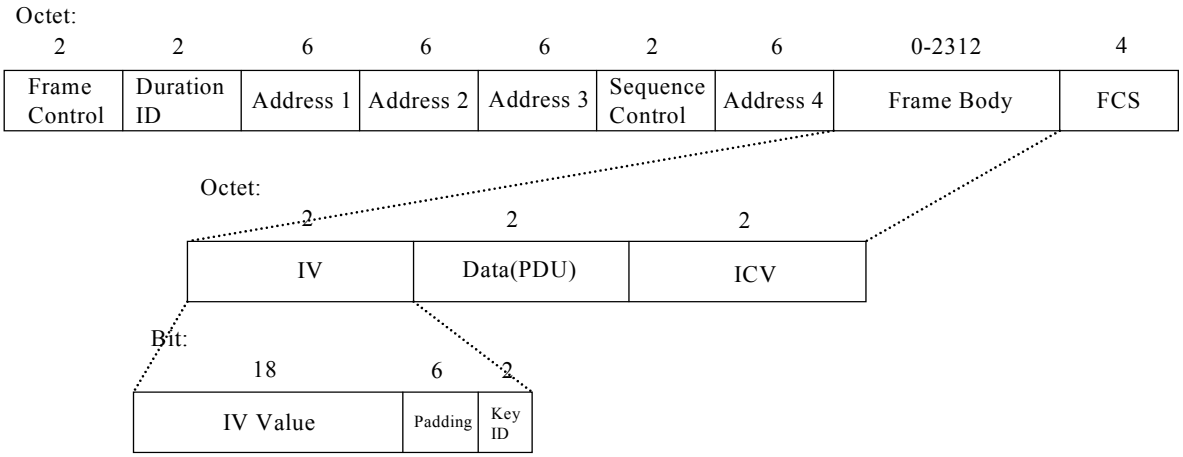

図 2-5WEP 使用時のフレームフォーマット

IV フィールドは送信側で使用した Initialization Vector の値と、使用している秘密鍵の

[5] ただし、暗号化されるのは Information items 部分のみで、MAC ヘッダは暗号化されない。

インデックス番号の情報を含む。Initialization Vector の値は基本的にフレーム毎に変更される。また、秘密鍵は最大 4 個まで送受信側で共有可能であり、Key ID フィールドによりどの鍵を用いてフレームを暗号化したかを送信先に明示する。Initialization Vector, Secret Key(秘密鍵), Plaintext(平文)から Ciphertext(暗号文)を生成する具体的手順を図 6 に示す。

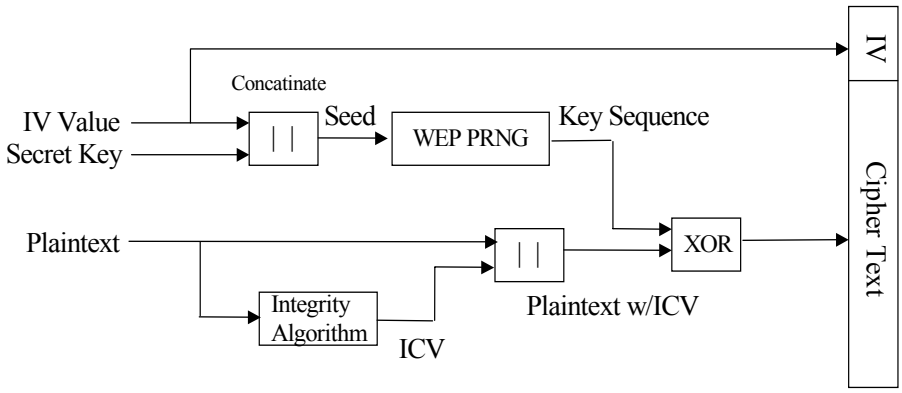

図 2-6WEP による暗号文生成手順

Initialization Vector は Secret Key と連結され、計 64bit の値が WEP PRNG(Pseudo Random Number Generator)への入力(Seed)となり、その出力として平文長の乱数を出力する。一方、Plaintext は Integrity Check Algorithm により ICV(Integrity Check Value)が生成・連結された後、WEP PRNG の出力との間で排他的論理和が取られ、暗号文が生成される。WEP PRNG は相対的に短い鍵から平文長の出力(乱数)を得るために用いられる最も重要なコンポーネントのひとつである。WEP PRNG への入力となる鍵が静的であると、多くの場合パケットのヘッダ等平文の一部分がある程度推測可能であるため、鍵が容易に解読されてしまう可能性がある。そのため、WEP では IV をフレーム毎に変更することで WEP PRNG への入力となる Seed を定期的に変更し、解読を困難にさせている。

2.2.3 問題点

802.11 を用いた無線 LAN ネットワークでは､そのセキュリティレベルを低下させる原因として､3 つの問題が存在する。ひとつ目はセキュリティ的に最低限必要な設定を行わずに運用されているという運用上の問題、ふたつ目は 802.11 の仕様そのものがセキュリティ的に十分でないという仕様上の問題､3 つ目が仕様にはあるが実装されていないという実装上の問題である。

802.11 の認証および暗号化(WEP)の機能は、WEP 鍵の存在がその根幹をなしている。つまり、正規の無線 LAN 端末(ユーザ)と無線 LAN 基地局(管理者)との間でだけ、安全に鍵が共有されていることが前提となる。通常、この鍵はシステム管理者によって手動で基地局に設定され、また無線 LAN インフラを使用するユーザには何らかの手段を用いて配布され

る。しかし、鍵の自動配布の仕組みを規定していない 802.11 では、鍵配布の方法そのものがシステム管理者任せとなってしまっているため、配布手段の安全性が確保されていない場合はその時点で無線 LAN のセキュリティレベルを著しく低下させてしまう。また、安全な方法により鍵を配布していたとしても、何らかの理由(Dictionary Attack や人為的な理由等)により鍵が漏洩する危険性は否めない。そのため、鍵の漏洩を防止するため定期的に鍵の更新を行うことはセキュリティ的に有効な手段のひとつであるが、管理コストを増大させるという点で、実際の運用ではあまり行われていないことも問題のひとつである。一旦 WEP 鍵が漏洩してしまうと、誰にも気づかれることなくデータの盗聴やネットワークへの侵入が可能となり、またたとえ鍵の漏洩を検知できたとしても、再度新規の WEP 鍵を無線 LAN ユーザに安全に配布することは管理コストの増大を招くことになる。また、仕様上は Key-mapping key と呼ばれる仕組みによって端末毎に個別の鍵を使用することも可能であるが、基地局の実装のほとんどはすべての端末との間で共通の WEP 鍵を使用するようになっており、第 3 者への鍵の漏洩の危険性が高くなっているとともに、鍵が漏洩した場合のインパクトも非常に大きなものである。さらに、共用の鍵を使っている限りは同一無線 LAN 基地局に接続している正規の無線 LAN 端末同士はお互いの通信内容を覗見ることが可能であり、特にホットスポットのような公共の無線 LAN アクセスでは大きな問題のひとつでもある。

また、802.11 のセキュリティ機能の中核をなす WEP のアルゴリズムそのものの脆弱性がいくつか指摘されている。特に、FMS attack[6]では 802.11 のトラフィックを必要十分な量、数分～数十分間程度傍受し，多少の計算をするだけで，傍受者が WEP 鍵そのものを知ることができてしまう。さらに，この方式を実装したソフトウェアがインターネット上で配布されるなど事態は深刻であり，多少のスキルを持っている者であれば(ビット長によらず)WEP 鍵を解読することはそれ程難しくない。これにより、例え適正な鍵管理を行っていたとしても、802.11 仕様の無線 LAN では WEP の脆弱性により、セキュリティの根幹が崩壊していることになる。その他基本的なところとして、MAC ID ベースのアクセスコントロールを実装している製品もあるが、アタッカーにとって MAC ID を偽装することは容易であり、セキュリティ的な防御には至らない。また、メッセージ認証(MIC)や Replay Protection の機能の欠如、Dis-Association・De-Authentication のマネージメントフレームに対するメッセージ認証の欠如も、DoS アタックに対する耐性を低下させている要因のひとつとなっている。

[渋谷]

[6] 参考文献 [2]

3　IEEE802.1X 関連技術

3.1　概要

IEEE 802.1X - Port-Based Network Access Control は基本的には IEEE802.1D での端末とスイッチ間の point to point の物理的なポート接続や 802.11 でのステーション AP 間の Association の論理的なポート接続を利用してポートに接続されているデバイスを認証し、認証プロセスに失敗すると、ポートへのアクセス防止を行う規格である。

802.11 では AP で Supplicant の資格証明を認証するため、ネットワークにアクセスする認証機関として RADIUS サーバを利用して、Supplicant が AP の論理的な非制御ポートに接続した後、ネットワークにアクセスする資格情報を確認する。資格情報の確認で認証が有効でれば認証プロセスを利用して鍵が交換され、Supplicant と AP 間のデータを暗号化し、アクセスポイントを識別できる鍵管理プロトコルが追加された。

802.11 の 802.1X では Supplicant と AP 間では PPP の認証手順を拡張しさまざまな認証方法(EAP-TYPE)を追加できる EAP over LAN を、AP と RADIUS サーバ間では RADIUS 認証プロトコルに EAP-Message ，Message-Authenticator を属性(アトリビュート)に加えた EAP over RADIUS を使用する。

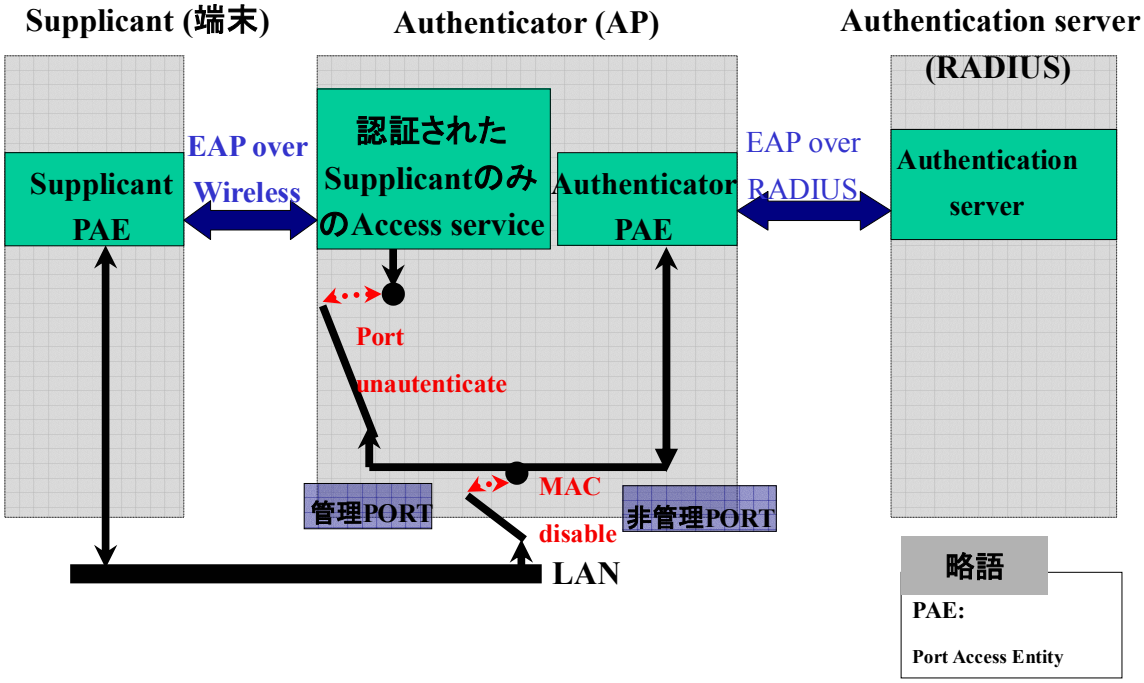

図 3-1 IEEE 802.1x - Port-Based Network Access Control

3.1.1　EAP over LAN

EAP は RFC2284(PPP Extensible Authentication Protocol)で規格化され、端末と RADIUS サーバに One Time Password、Public key、Kerberos、Smart Card などの新し

い認証方法をEAP-TYPEの拡張モジュールとして追加することによってさまざまな認証方法を使用することができる。

認証方法によってEAP-typeを下記のように分類できる。

表 3-1　パスワード交換方式

名称	特徴
EAP-MD5	ハッシュ化したパスワードを交換する。
EAP-LEAP	Cisco独自方式。 サードパーティで対応RADIUSもある。
EAP-SKE	Shared Key Exchange: 双方向認証可能で、特にローミングを目的としている。
EAP-SRP	Secure Remote Password 方式。ハッシュ化されたパスワードを格納しておき、認証に利用する特徴あり。

表 3-2　PKIを利用した方式

名称	特徴
EAP-TLS	TLS(SSL)の公開鍵証明書を利用して相互認証する。
EAP-TTLS	TLSでトンネルを作成し、トンネル上でパスワード認証する。証明書はサーバだけでよい。
PEAP	Protected Extensible Authentication Protocol: 双方向認証、ローミング利用可能なセッション鍵生成。サーバ認証と鍵生成にはEAP-TLS を使い、ユーザ認証にEAPを使う。EAP中のTLSにEAPをカプセル化している。
EAP-MAKE	Mutual Authentication Protocol: Diffie-Hellman 方式を利用。

表 3-3　GSM（Global System for Mobile Communications）

名称	特徴
EAP-AKA	UMTS AKA 認証と鍵配布方式を使う。AKA は対称鍵を利用し、UMTS SIM カード内で動作し、AKA は GSM 認証と下位互換性がある。UMTS AKA が利用できれば GSM と UMTS の認証も可能。
EAP-SIM	SIM（GSM Subscriber Identification Module）カードを使った認証とセッション鍵配布方式。

この EAP を IEEE802.3(Ethernet)、IEEE802.5(TokenRing)、802.11(無線 LAN)の各メディアで転送できるようにカプセル化したものが EAP over LAN である。

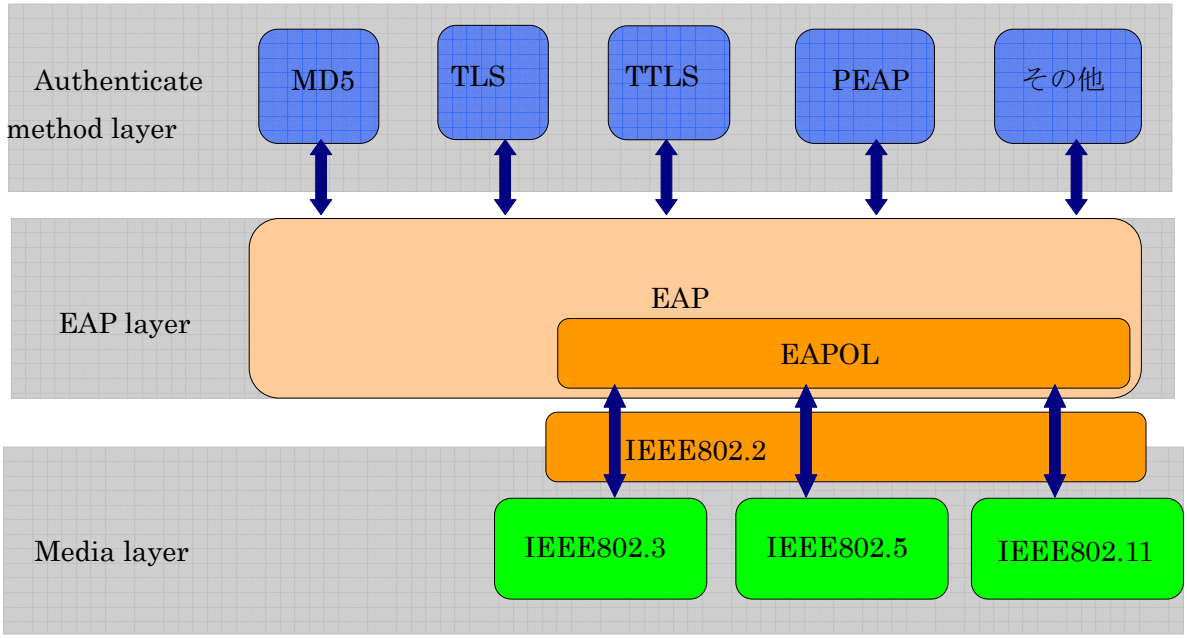

図 3-2 図 EAP のレイヤ別機能分類

EAP over LAN のデータフォマット下記のようになっている。

表 3-4 EAP over LAN のデータフォマット

Octet number	フィールド	概要
1-2	PAE Ethernet Type	PAE (Port Access Entity: ポートのアクセス管理を行うモジュールや機能) が使う Ethernet のタイプ番号 [**0x888E**]
3	Protocol Version	この EAPoL パケットのプロトコル番号 [**0x01**]
4	Packet Type	[0x03] **EAP-Packet**: [**0x00**] 次ページの EAP パケットをくるむパケット。このパケットの Body は EAP パケット。 **EAPOL-Start** [**0x01**] EAP を開始するときに使う。 EAPOL-Logoff [0x02] EAP 終了時に使う。 **EAPOL-Key** [**0x03**] 鍵配布に使う。Body の形式は別ページ参照→Key Descriptor Format EAPOL-Encapsulated-ASF-Alert [0x04] 未認証時に SNMP など管理フレームを送るのに使う。
5-6	Packet Body Length	送受信されるデータのデータ長
7-n	Packet Body	データそのもの。

Bit	Field	説明
8	Version	EAP の version を表す(802.1X は 0x01)
8	Packet Type	EAPOL の packet 種類を表す(EAPOL-KEY は 0x03)
16	Packet Body Length	Version field を除いた Type+body の長さ
8	Type	Key descriptor を表す(RC4 は 0x01)
16	Key Length	key の長さ
64	Replay Counter	64-bit NTP time stamp
128	Key IV	128-bit cryptographically random number Key field を復号する際に使用
1	F lag	Key の種類を表す
7	Key Index	WEP Key の register 番号
128	Key Signature	Version field 以降のすべての field の HMAC-MD5 による MIC 値
可変	Key	暗号済みの WEP Key か空

表 3-5　EAPOL-Key フォーマット

3.1.2　　802.1X 認証シーケンス

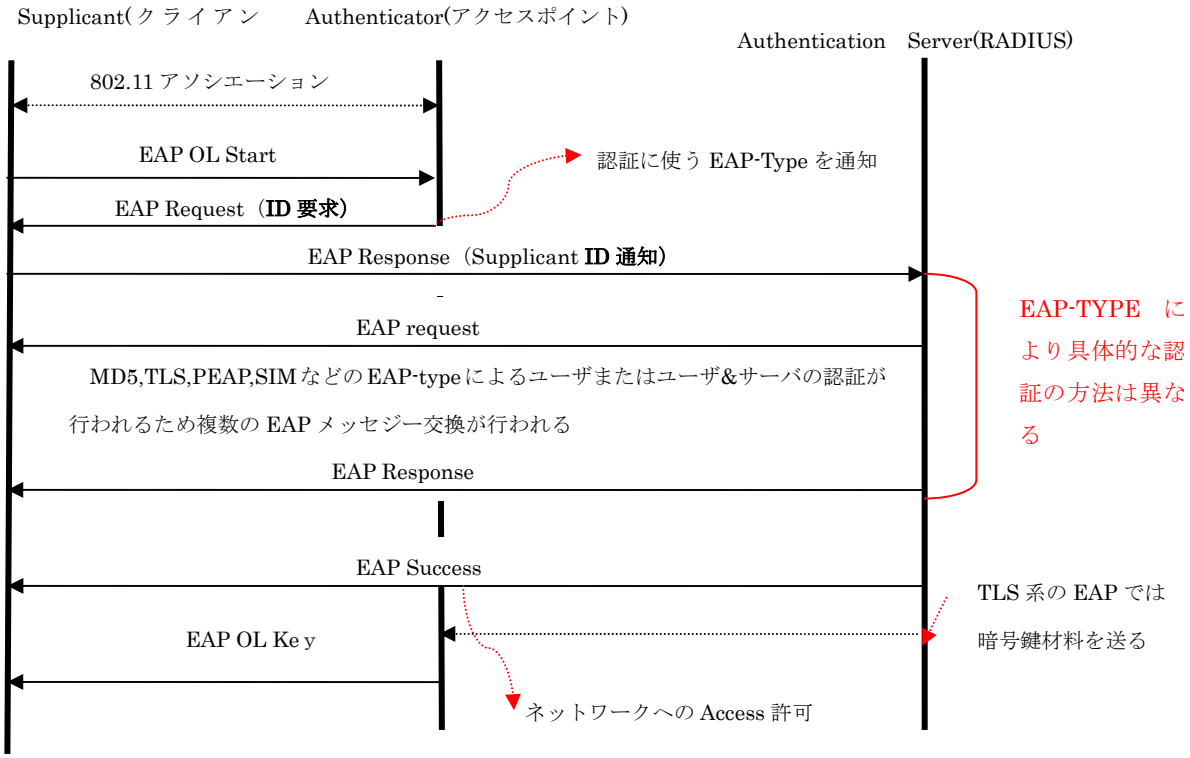

図 3-3　802.1x 認証シーケンス

[任]

3.2 EAP-TLS

3.2.1 EAP-TLS の特徴

EAP-TLS は、RFC2716(Extensible Authentication Protocol – Transport Layer Security)で規定され、TLS のハンドシェイクプロトコルを利用して、暗号アルゴリズムの選択、クライアント/サーバ間の証明書による双方向認証、暗号鍵を安全に共有するための階層的な鍵の生成などを行う方式である。特にクライアント及びサーバ双方の証明書を用いて相互に認証をおこなう点は、EAP-MD5 の様なパスワード交換方式と異なり、双方で証明書の有効性検証を実施するため、より強固なセキュリティを確保でき、EAP-TLS の最大の特徴となっている。図 3-4 に EAP-TLS における TLS のネゴシエーションシーケンスを記載する。

また、プレマスターシークレット、マスターシークレット、マスターセッションキーと階層的に鍵の生成、交換を行い、クライアントとアクセスポイント双方が保持する鍵を安全に生成することができる。

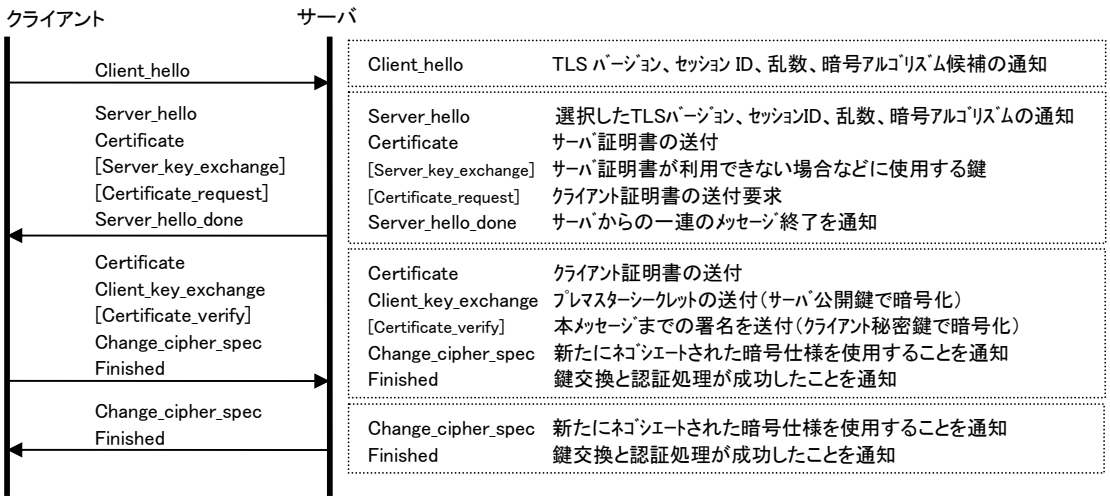

図 3-4 EAP-TLS における TLS のネゴシエーションシーケンス

EAP-TLS の特徴として、他にも以下の様なことが挙げられる。

・フラグメント、リアセンブリに対応

Flags フィールドなどを利用してフラグメントを明示する。各フラグメントパケットの Identifier 値を利用することで、リアセンブリ時のエラーに対応可能である。

・効率的な再認証が可能

短い期間内に再認証を行う場合には、クライアントは以前のセッション ID を付けて EAP レスポンスパケットを送信する。RADIUS サーバが該当セッションの継続を許可する場合

は、証明書の交換を省略することができ、認証シーケンスの一部を短縮可能である。

・パケット損失に対応可能

クライアントからのレスポンスパケットを受信できなかった場合、RADIUS サーバはその元となるリクエストを再度送信するため、パケット損失に対応可能である。

3.2.2 EAP-TLS の認証シーケンス

無線 LAN における EAP-TLS の認証シーケンスを図 3-5 に示す。認証シーケンスは、Supplicant（クライアント）と Authenticator（アクセスポイント）間における 802.11 アソシエーションに始まり、EAP ネゴシエーション、暗号仕様及び証明書の交換などを行う TLS ネゴシエーションシーケンスと続き、最後に Authentication Server（Radius）からセッションタイムアウト値及び WEP キーの元となる MPPE(Microsoft Point-to-Point Encryption)などが記載された RADIUS Access-Accept パケットが送信され、アクセスポイントから EAP-Success パケットと EAPOL-Key がクライアントに送信される。以上のシーケンスを経て、クライアント/AP 双方が WEP キーを共有することができる。

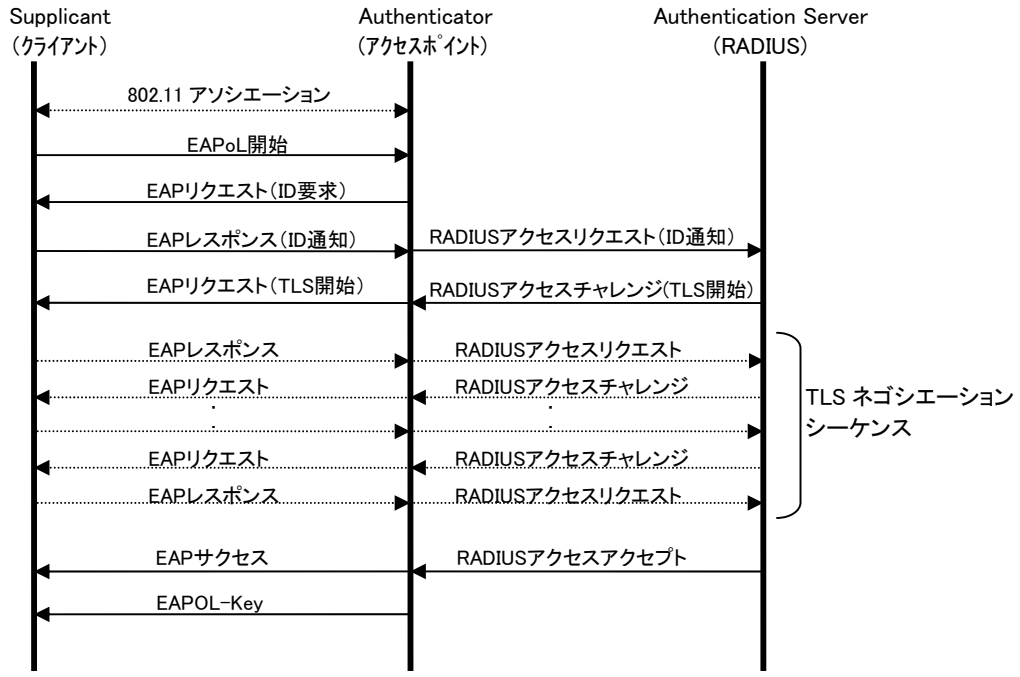

図 3-5 EAP-TLS の認証シーケンス

3.2.3 鍵の生成について

鍵の生成は、EAP-TLS において最も重要な点であるため、ここで再度触れておく。

まず初めに TLS のネゴシエーションシーケンスによる鍵生成の一例を紹介し、次にその TLS で作成したマスターシークレットを使用して、WEP キーを生成するまでの概要を紹介する。

図 3-6 に示す様に、クライアント及びサーバ双方で乱数（クライアントランダム/サーバランダム）を作成し、それぞれ Client_hello/Server_Hello の中で相手に通知する。次に、クライアントよりサーバの公開鍵で暗号化したプレマスターシークレットを Client_Key_Exchange にて、サーバへ通知する。サーバ側では、サーバの秘密鍵を用いてプレマスターシークレットを複合し、双方でプレマスターシークレットを共有する。次に、プレマスターシークレット、クライアントランダム、サーバランダム、ラベルの４つを使用して、擬似乱数関数（PRF）による演算を行い、マスターシークレットを作成する。図 3-6 のクライアントから送信する Change_cipher_spec より後は、このマスターシークレットを元に暗号化されたデータとなる。最後に、マスターシークレット、クライアントランダム、サーバランダム、ラベルの４つを使用して、擬似乱数関数（PRF）による演算を再び行い、マスターセッションキーを作成する。

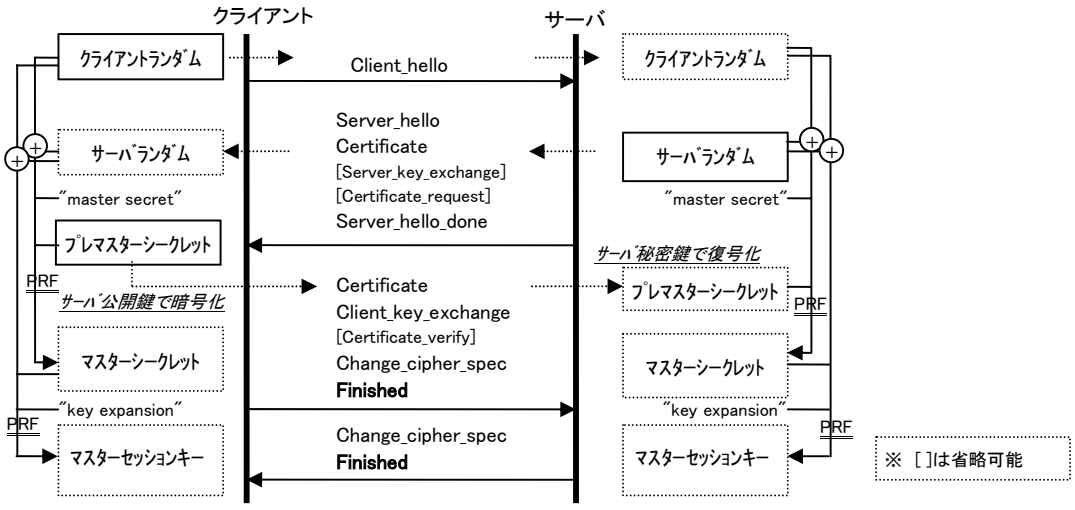

図 3-6　TLS ハンドシェイクプロトコル、レコードプロトコルによる鍵の生成(概要)

以上が TLS によるマスターセッションキー生成までの概要である。EAP-TLS を用いた実際の 802.1x の無線システムにおいては、図 3-7 に示す様に TLS のマスターシークレットを元にマスターセッションキーになる MS-MPPE(MS_MPPE_Send_key/ MS_MPPE_Recv_key)を Radius 上で生成し、Radius アクセスアクセプトパケットを利用してアクセスポイントに送信する。MS-MPPE を受け取ったアクセスポイントは、キーインデックス、初期化ベクトル、キーシグネチャー及びキーが記載された EAPOL パケット

をクライアントへ送付するといったシーケンスを経て、WEP キーを共有することになる。

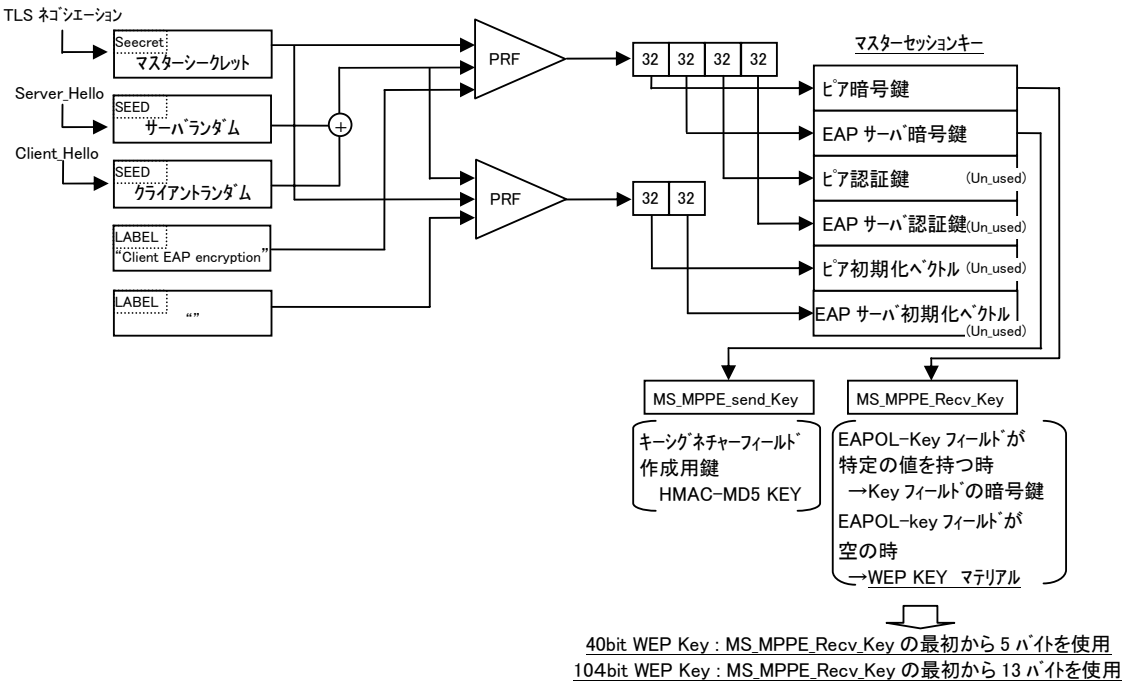

図 3-7　EAP-TLS による鍵の生成シーケンス(概要)

[岸本]

3.3 EAP-TTLS

3.3.1 EAP-TTLS の特徴

EAP-TTLS（Extensible Authentication Protocol – Tunneled Transport Layer Security）は、Funk Software 社が中心となって策定し、現在インターネットドラフトとなっている規格である。TLS ネゴシエーションで無線 LAN 上のデータ保護に利用するマスターセッションキーを生成する点やその他の主な特徴は EAP-TLS と同様であるが（3.2.1 参照）、以下の点が EAP-TLS とは異なる特徴である。

- クライアント証明書が必須ではない。

TLS ネゴシエーション内でクライアント証明書はオプション扱いとなっている。従って、EAP-TLS がクライアント証明書を必須とし、各クライアント PC で証明書を維持管理しなければならないのに対し、運用負担を軽減することができる。

- TLS トンネル内で様々なクライアント認証方式が使用可能である。

クライアント認証に TLS トンネル内でパスワードベースの認証プロトコルを利用できるため、既存の認証システムをそのまま使用することが可能である。

- ユーザ ID を TLS トンネル外では流さない。

TLS ネゴシエーション前の EAP-Response/Identity パケット内ではダミーのユーザ ID を流し、実際のユーザ ID は TLS トンネルで保護することによりセキュリティを確保している。

EAP-TTLS の構成要素は、クライアント（Supplicant）、アクセスポイント（Authenticator）、TTLS サーバ及び RADIUS サーバとなる。アクセスポイント、TTLS サーバ、RADIUS サーバは論理的な区別であり、物理的にわける必要はない。TTLS サーバとは EAP-TTLS を実装するサーバで、TLS トンネルによりクライアントとの間の認証を保護するとともに、RADIUS サーバとの間の認証をプロキシーする。このプロキシー機能があるため、RADIUS サーバには MD5-Challenge、One-Time-Password といった EAP プロトコルのほかに、PAP、CHAP、MS-CHAP、MS-CHAP-V2 といった非 EAP プロトコルを使用することができる。なお、EAP-TTLS を示す EAP タイプフィールドの値は 21 である。

また、鍵管理については、TLS ネゴシエーションで生成したマスターセッションキーの使用方法を、XXX-Data-Cipher-Suite メッセージによりクライアント及びアクセスポイントが TTLS サーバと折衝して独自に決定できる方法が提案されているが、実装としては EAP-TLS と同様に MS-MPPE が使用されている。

3.3.2 EAP-TTLS の認証シーケンス

CHAP をクライアント認証プロトコルとして使用した場合の EAP-TTLS 認証シーケンスを図 3-8 に示す。シーケンス前半の Supplicant と TTLS サーバ間の TLS ネゴシエーションはクライアント証明書が必須ではないことを除き EAP-TLS と同様であり、後半の TLS トンネル内で行われる Supplicant と RADIUS サーバ間の認証フェーズでは、使用される認証プロトコルによって RADIUS 通信のパラメータやシーケンスが異なってくる。

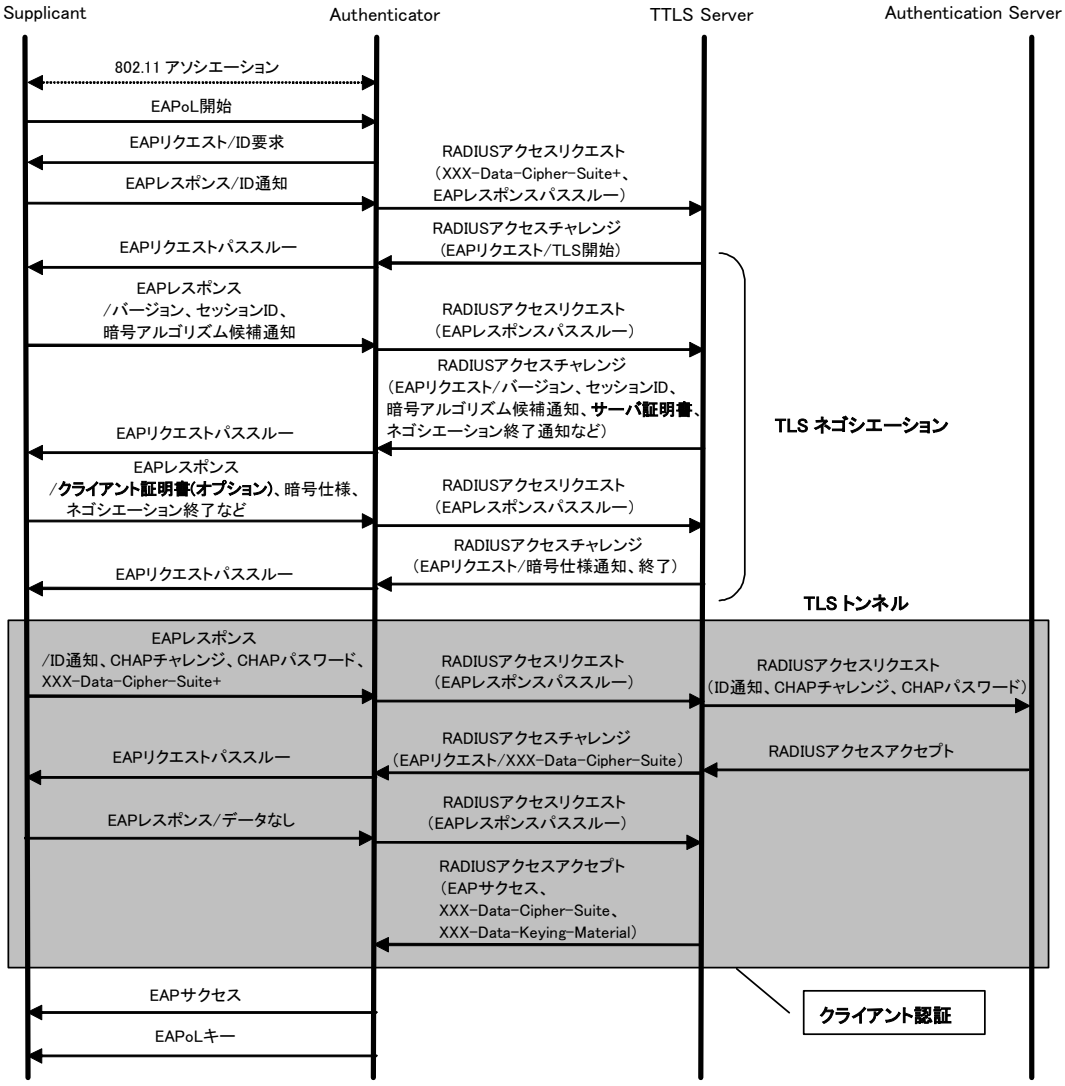

図 3-8 EAP-TTLS 認証シーケンス（クライアント認証に CHAP 使用時）

[門田]

3.4 PEAP

3.4.1 PEAP の特徴

PEAP は EAP の認証方式に TLS トンネル内の認証を利用する方式であり、強固なセキ

ュリティを確保できる方式である。PEAP クライアントと認証局の間に PEAP 認証プロセスに２つの段階がある。第１段階は PEAP クライアントと RADIUS サーバの間に TLS トンネルを確立し、 第２段階はその TLS トンネルで認証をおこなう。

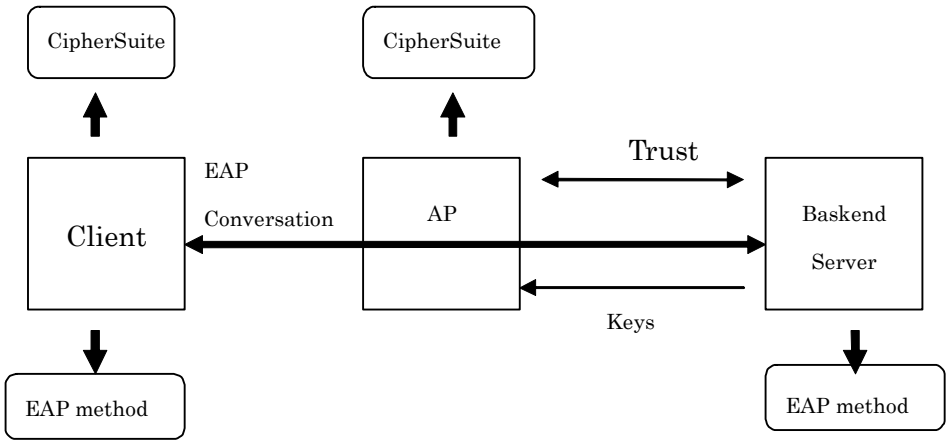

図 3-9　PEAP での各要素の機能

EAP は Authenticator(AP など)を通過し、クライアント(Supplicant)と RADIUS サーバの間で認証を行う。 Authenticator と RADIUS サーバはカンバセーションが進むために信頼を確立する必要がある。クライアントとサーバの間のカンバセーションは暗号化され、TLS トンネルを確立して、 クライアントとサーバの認証を保護する。

PEAP の特徴は以下の点で EAP-TLS と異なる。

- Flags フィールドのフォーマットが異なる。

PEAP では EAP-TLS で使用していた Flags フィールドの一部を PEAP のバージョンをあらわすビットとして利用している。

- Version　Negotiation

PEAP には version フィールドが含まれている。PEAP のバージョン 0 とバージョン 1 は互換性がないため、バージョンのネゴシエーションが必須である。クライアントが提示した version を RADIUS サーバがサポートしない場合、サーバは version を下げてクライアントに合わせることができる。

- TLS トンネル内でクライアント認証をおこなう。
- ユーザ ID は TLS トンネル内で保護される。

クライアントのユーザ ID は TLS トンネルで RADIUS サーバとの認証を保護することによって保護されている。

- クライアント証明書が必須ではない。

EAP-TLS ネゴシエーション内でクライアント証明書は必須であり、各クライアント PC

に証明書を管理が必要。しかし、PEAP ではクライアント証明書が必須ではないため、証明書によるクライアント認証を省略できる。

PEAP では EAP-TLS と同様でマスターセッションキーから階層的に鍵を生成する。マスターセッションキーを交換する部分を暗号化し、鍵を保護できる。PEAP を示す EAP タイプフィールドの値は 25 である。

3.4.2 PEAP の認証シーケンス

図 3-10 で表示している認証シーケンスの前半については EAP-TLS とほぼ同様に TLS トンネルを確立するが、クライアント証明書が必須ではない。後半については TLS トンネル内で行われる Supplicant と RADIUS サーバ間の使用される認証プロトコルによって、認証シーケンスが異なる。

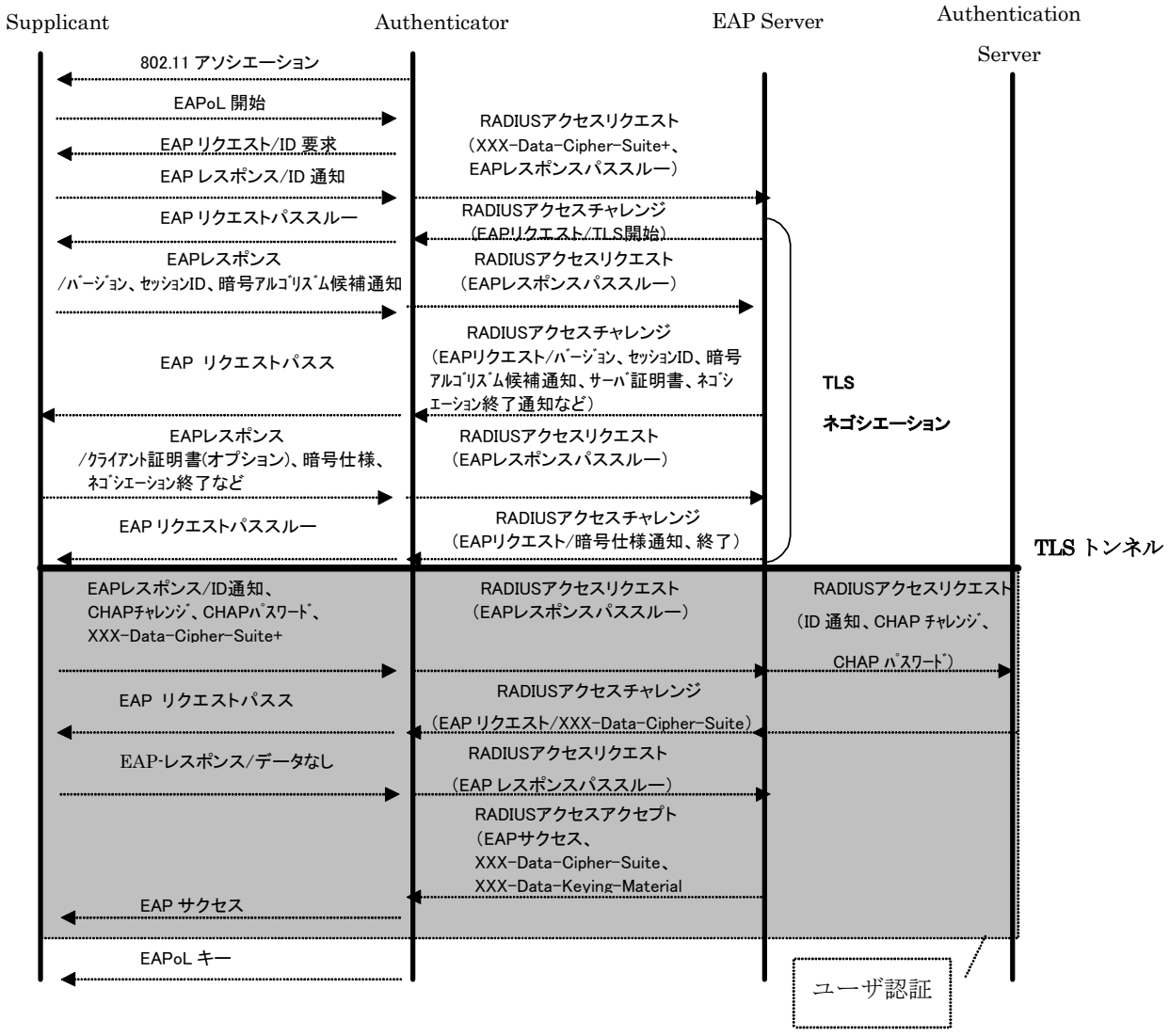

図 3-10 PEAP 認証シーケンス

[パンシット]

4　RADIUS 技術

4.1　概要

RADIUS (Remote Authentication Dial In User Service)は、RFC2865 および RFC2866 により規定されている、AAA (Authentications、Authorization and Accounting)の方式の一種で、主にダイヤルアップ接続でのユーザ認証方式として利用されている。しかし近年では、ダイヤルアップ接続でのユーザ認証のみならず、VPN、VLAN、無線 LAN などへの接続認証にも利用されるようになっている。

4.2　RADIUS プロトコル

RADIUS プロトコルは、NAS(Network Access Server)や RAS(Remote Access Server)、無線 LAN アクセスポイントなどの RADIUS クライアントと RADIUS サーバとの間で、認証、認可およびアカウンティング処理に必要な情報を伝送するための UDP ベースのアプリケーション・プロトコルである。RADIUS プロトコルは標準の UDP ポートとして、認証では 1812 番を、アカウンティングでは 1813 番を使用する。

4.2.1　RADIUS パケット

RADIUS プロトコルは UDP ベースのプロトコルであり、伝送する情報はパケット単位で扱われる。RADIUS パケットは最小 20 オクテット、最大 4096 オクテットの、以下の構造を持ったパケットである。

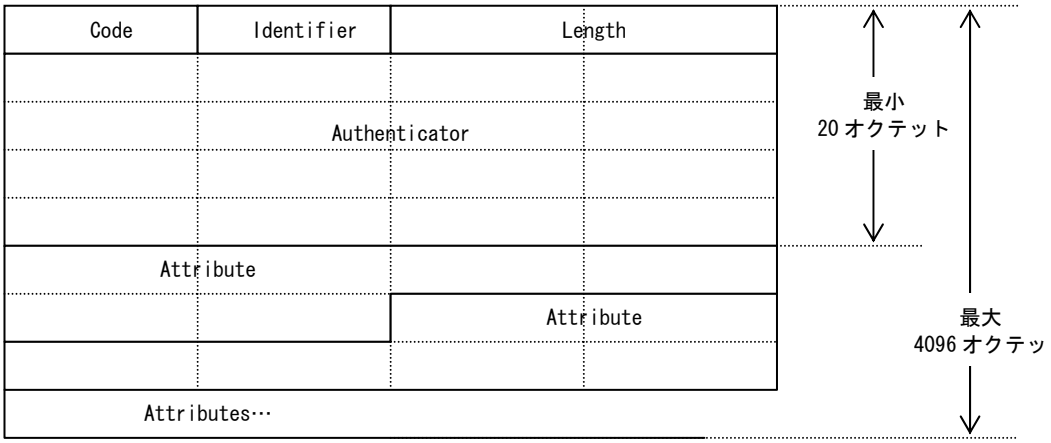

図 4-1　RADIUS パケット構造図

RADIUS パケットは Code によって用途が分けられる。一般的に使用される RADIUS パケット Code は以下の通り。

Code	名称	説明
1	Access-Request	認証処理の要求パケット。RADIUS クライアントから RADUS サーバに送られる。
2	Access-Accept	認証許可の応答パケット。RADIUS サーバから RADIUS クライアントに送られる。
3	Access-Reject	認証拒否の応答パケット。RADIUS サーバから RADIUS クライアントに送られる。
4	Accounting-Request	アカウンティング処理の要求パケット。RADIUS クライアントから RADUS サーバに送られる。
5	Accounting-Response	アカウンティング処理の応答パケット。RADIUS サーバから RADIUS クライアントに送られる。
11	Access-Challenge	認証要求に対する申し立てを行うための応答パケット。RADIUS サーバから RADIUS クライアントに送られる。

4.2.2 RADIUS 属性(アトリビュート)

RADIUS 属性(アトリビュート)は、1 オクテットの属性タイプ、1 オクテットの属性長、および可変長の属性値の 3 つの情報から構成される。以下に RADIUS 属性の構造を示す。

Type	Length	Value...

図 4-2 RADIUS 属性構成図

属性タイプ(Type)は、RADIUS 属性のタイプを番号で示す。例えばユーザ名を示す属性は 1 番(User-Name)、ユーザの PAP パスワードを示す属性は 2 番(User-Password)などである。

属性長(Length)は、RADIUS 属性の長さをオクテット単位で示す。属性長には、属性タイプを示すオクテットと、属性長を示すオクテット自身の長さを含む。従って 1 つの RADIUS 属性で保持することのできる属性値の最大オクテット数は、253 オクテットとなる。

属性値(Value)は、RADIUS 属性の値を示す。属性値の長さと形式は、それぞれの属性タイプにより異なる。

4.3 RADIUS の EAP 対応

EAP (PPP Extensible Authentication Protocol)は RFC2284 で規定されている、様々な認証手順に対応するための PPP の拡張規格である。RADIUS においても EAP に対応するための、以下の 2 つの RADIUS 属性が RFC2869 で規定されている。

4.3.1 EAP-Message

この属性は RADIUS クライアントが EAP プロトコルを理解する必要なしに、RADIUS サーバによって EAP を用いたユーザ認証を可能にするために、EAP パケットをカプセル化するための、バイナリ文字列型の属性である。RADIUS クライアントはユーザから受け取った全ての EAP メッセージを 1 つ以上の EAP-Message 属性に収納し、Access-Request の一部として RADIUS サーバに転送し、Access-Challenge、Access-Accept および Access-Reject に含まれる EAP メッセージをユーザに返送することができる。

もし複数の EAP-Message が Access-Request や Access-Challenge パケットに含まれていた場合、それらは順番になっていなければならず、またパケットの中で連続していなければならない。Access-Accept と Access-Reject パケットでは、EAP-Success または EAP-Failure を収納した、たった一つの EAP-Message 属性だけを持つようにするべきである。

RADIUS サーバが EAP メッセージを受信した際、それを解釈できない場合は Access-Reject を返すべきである。

4.3.2 Message-Authenticator

この属性は CHAP、ARAP または EAP などの認証方法を使用した RADIUS パケット Access-Request のなりすましを防ぐために Access-Request を署名するために使用する、バイナリ文字列型の属性である。通常、Access-Request における Message-Authenticator 属性の使用は任意だが、EAP-Message 属性を含む Access-Request、Access-Accept、Access-Reject または Access-Challenge では、Message-Authenticator 属性は必ず使用されなければならない。

Message-Authenticator 属性を含む Access-Request を受け取った RADIUS サーバは、Message-Authenticator の正しい値を計算し、もし送られてきた値と異なるようであれば、パケットを暗黙のうちに破棄しなければならない。

Message-Authenticator 属性を含む Access-Accept、Access-Reject または Access-Challenge を受け取った RADIUS クライアントは、Message-Authenticator の正しい値を計算し、もし送られてきた値と異なるようであれば、パケットを暗黙のうちに破棄しなければならない。

4.4 802.1X と RADIUS

802.1X は、Ethernet(IEEE 802.3)、Token-Ring(IEEE 802.5)および無線 LAN(802.11)

を含む 802 メディアのための「ネットワーク・ポート認証」を提供する。802.1X は、バックエンド RADIUS サーバの使用を要求していないため、スタンド・アロンで配備されたスイッチあるいは AP でも、集中管理によるシナリオと同様に利用することができる。しかし 802 ネットワークへの認証、認可およびアカウンティング(AAA)を集中管理することが望ましい状況では、バックエンドに認証およびアカウンティングサーバを配備すると良い。そのような状況では、802.1X Authenticator が AAA のクライアントとして機能することが期待される。

任意の AAA プロトコルのサポートが 802.1X Authenticators のオプションとして認められているが、802.1X では具体的な AAA プロトコルとして RADIUS を利用することを、規格の範囲外の付録として組み入れている。またこの付録部分は IETF ドラフト”draft-congdon-radius-8021x”として、802.1X Authenticators による RADIUS 使用法に関する提案として策定中である。

”draft-congdon-radius-8021x”では、RADIUS 属性についても、802.1X の概念に対応できるよう、その意味合いを定義しなおしている。以下に再定義された RADIUS 属性のうち、大きく意味合いが変わるものを述べる。

4.4.1 User-Name

802.1X では、通常サプリカントは EAP-Response/Identity メッセージで識別名を示す。User-Name 属性が利用可能なら、サプリカントは Access-Request の User-Name 属性にも識別名を示す。

さらに Service-Type 属性の値が Call Check(10)の場合、User-Name 属性はサプリカントの MAC アドレスに設定された Calling-Station-ID の値と同じ値を持つ。

4.4.2 User-Password、CHAP-Password、CHAP-Challenge

802.1X は PAP または CHAP 認証をサポートしないので、User-Password、CHAP-Password および CHAP-Challenge 属性は RADIUS クライアントとして動作する 802.1X Authenticator で使用されることはない。

4.4.3 Reply-Message

Reply-Message 属性はユーザに示されるであろうテキストを示す属性だが、EAP の Notification タイプメッセージが同様の働きをする。従って 802.1X Authenticator にユーザに表示するメッセージを送信するには、RADIUS サーバは表示メッセージを EAP-Message/EAP-Request/Notification 属性に入れて送るべきであり、Reply-Message 属性を使用しない方が良い。

4.4.4 State、Class、Proxy-State

これらの RADIUS 属性は RFC2865 の記述と同じように使われる。特に多くの RADIUS サーバでは、State 属性が 802.1X 認証手順の状態追跡用の属性として使用されるので、802.1X Authenticator によりサポートされるべきである。

4.4.5 Vendor-Specific

Vendor-specific 属性は RFC2865 の記述と同じように使われる。ただし以下の 2 つの VSA は WEP 鍵の動的配布を行うために使用される。

- MS-MPPE-Send-Key
- MS-MPPE-Recv-Key

これらの属性の値は、EAP 認証手順でネゴシエートされたマスターシークレットから生成され、RC4 EAPOL-Key ディスクリプタの暗号化および認証に使用される。

4.4.6 Session-Timeout

Termination-Action 属性が無かったり、値が Default(0)である Termination-Action 属性を持つ Access-Accept が送られた場合、Session-Timeout 属性はセッション切断までにサービスを提供する最大秒数を示す。

値が RADIUS-Request(1)である Termination-Action 属性を持つ Access-Accept が送られた場合、Session-Timeout 属性は再認証までにサービスを提供する最大秒数を示す。この場合、Session-Timeout 属性の値は、802.1X の再認証タイマーとして使用される。

値が RADIUS-Request(1)の Termination-Action 属性と共に、値が 0 の Session-Timeout 属性が送られた場合、最初の認証が成功後、直ちに異なる認証手順を実行の要求することを示す。

RFC2869 での記述通り、Access-Challenge にて Session-Timeout 属性が送信された場合、この属性の値は 802.1X Authenticator が EAP-Response を再送信するまでの、最大待ち秒数を示す。

4.4.7 Termination-Action

この属性の値が RADIUS-Request(1)の場合、Session-Timeout 属性の値は再認証までの最大秒数を示す。この属性の値が Default(0)の場合、Session-Timeout 属性の値はセッション切断までの最大秒数を示す。

4.4.8 Called-Station-Id

802.1X Authenticator では、この属性はブリッジまたはアクセスポイントの MAC アドレスを ASCII 形式で保持するのに用いられる。メディアが 802.11 で SSID が分かる場合、MAC アドレスにコロン':'で区切って SSID を付加すべきである。

例："00-10-A4-23-19-C0:AP1"

4.4.9 Calling-Station-Id

802.1X Authenticator では、この属性はサプリカントの MAC アドレスを ASCII 形式で保持するのに用いられる。

[納村]

5　PKI 技術

5.1　PKI の基本的な仕組み

PKI とは、公開鍵暗号を応用した電子証明書(公開鍵証明書)を用いて、電子データの暗号や署名、認証などにおいて使用される技術である。証明書のフォーマットは ITU-T/X.509 で規定されており、さらにこれをインターネット上で利用するためのサブセットとして IETF/PKIX WG が証明書プロファイルを定義している。EAP-TLS や EAP-TTLS, PEAP などにおいても、RFC3280 に準拠した証明書を用いて認証が行われる。

5.1.1　PKI における認証の考え方

ここでは認証を例に、PKI の具体的な使われ方を説明する。認証は、片方向認証(unilateral authentication)と双方向認証(mutual authentication)に分類される。SSL/TLS 認証などでよく知られているサーバ認証は、クライアント(ブラウザ)が Web サーバなどを認証する片方向認証であり、クライアント認証とは例えば Web サーバとクライアント(ブラウザ)がお互いを認証する双方向認証である。

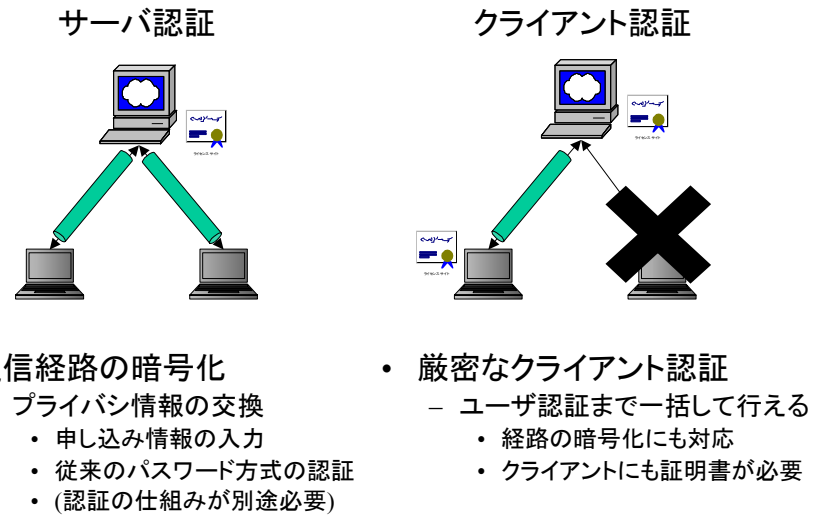

図 5-1 サーバ認証とクライアント認証

ここで認証が成立する大前提として、以下が挙げられる。

a) 認証されるエンティティは鍵ペアを所有している。

b) 秘密鍵は唯一エンティティ自身だけが所有している。

c) 公開鍵は信頼される第三者(TTP: Trusted Third-Party)－例えば認証局など－によって署名されている。これが公開鍵証明書である。

これらをもとに、例えば SSL/TLS 認証などは、以下の情報のやりとりによって認証が成

立する。

1) 認証されるエンティティ(Subscriber)は、あるデータに秘密鍵を用いて署名する。

2) Subscriber は、1)の署名データを自身の公開鍵証明書と共に Relying-Party へ送信する。

3) 認証するエンティティ(Relying-Party)は、受信した Subscriber 証明書を信頼できるか確認する。

4) Relying-Party は、受信した署名データを、3)の Subscriber 証明書を用いて検証する。

このやりとりによって、片方向認証が成立し、これを相互に行うことで双方向認証も成立する。

5.1.2 証明書の失効検証とリポジトリ

ここまでは、一般的な SSL/TLS 認証の流れで比較的よく知られている話だが、あまり知られていない話として、証明書の失効が挙げられる。

前述のステップ 3)において、Subscriber 証明書が信頼できるかどうかを確認する項目として、RFC3280 では以下を最低限の要件としている。

- 発行者公開鍵による証明書の署名検証
- 有効期限の確認
- 失効検証
- 発行者名の確認

失効検証には、その証明書が失効しているかどうかを確認するために失効情報が必要となる。この失効情報は、証明書失効リスト(CRL)として公開されているものを入手するか、あるいは OCSP レスポンダへ問い合わせて入手することになる。一般に CRL は、証明書を発行する認証局が運営するリポジトリ上で公開されていることが多い。リポジトリとは、証明書や CRL を公開するもので、X.500 ディレクトリや LDAP などのディレクトリサーバが用いられることが多い。しかし、これらのリポジトリにアクセスするには、X.500 や LDAP などのクライアント機能を実装する必要があるため、リポジトリとして Web サーバを用いるなどして代替する場合もある。

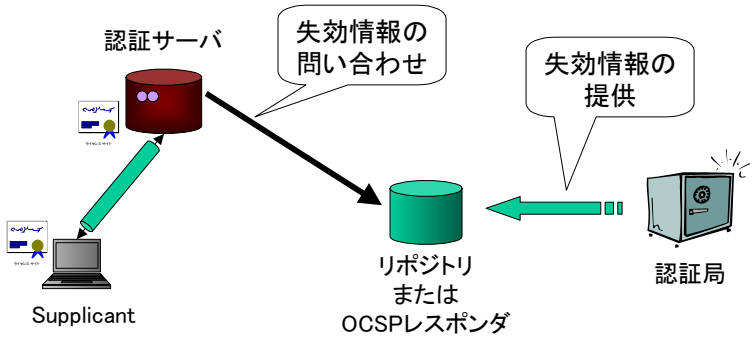

図 5-2　証明書の失効検証

5.2　X.509 証明書拡張と認証の関係

前節で証明書を検証するための最小要件を挙げた。しかし、X.509 や RFC3280 では、証明書に様々な項目を定義している。本節では特に、TLS 認証において証明書検証の際に参照されるべき項目について簡単に説明する。

5.2.1　証明書拡張とは

X.509 証明書は、v1 から始まり今日では v3 に至っている。v1 では、まさに署名者と主体者の関係や主体者の持つ鍵ペアを証明する基本領域を定義するのみだったが、v3 では、さらにいくつかの証明書拡張が追加され、より現実的な証明書となっている。

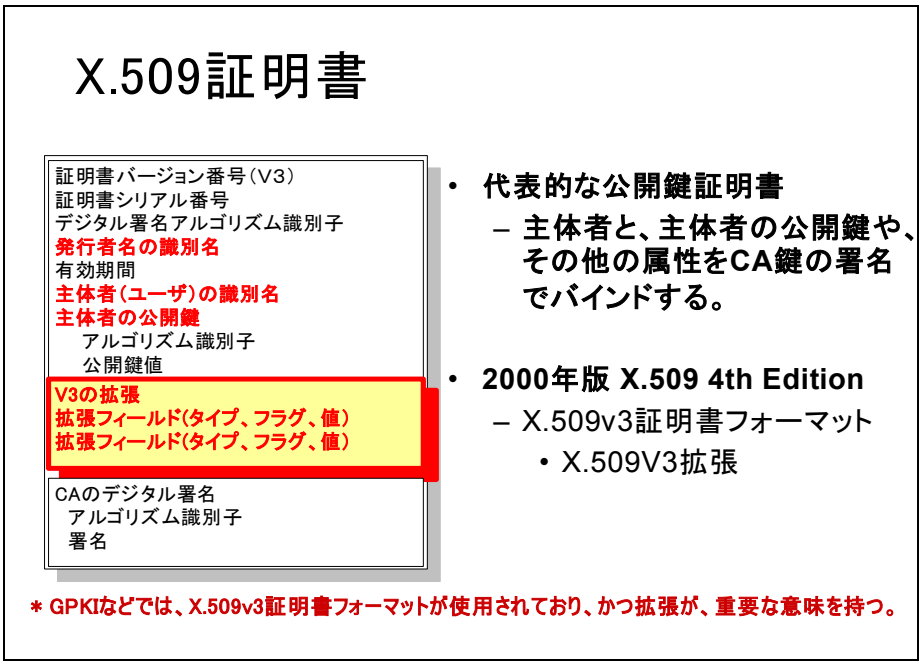

図 5-3　X.509 の証明書構造

証明書拡張には、ITU-T/X.509 が規定した標準拡張領域と、証明書発行者が任意に拡張できるプライベート拡張領域とがある。PKIX/RFC3280 では、インターネット上での利用を考慮し、このプライベート拡張領域にインターネット拡張をいくつか規定している。

一般に証明書では、これらの拡張を全て記載する必要はない。これら定義済みの拡張のうち、各 PKI ドメインにおいて必要とする拡張を選択し、記載すればよい。このように、RFC3280 などで定義された拡張のうち、利用する拡張を選別し、記載する情報を明確化したものを(狭義には)証明書プロファイルと言う。

このような証明書の構造の中で、基本領域、拡張領域をあわせて TLS 認証で必要かつ注意が必要と思われる項目はおよそ以下の通りと考えられる。

- issuer DN
- subject DN
- serialNumber
- validity
- keyUsage 拡張
- subjectAltName 拡張
- issuerAltName 拡張
- extKeyUsage 拡張
- cRLDistributionPoints 拡張
- authorityInfoAccess 拡張

5.2.2 issuer/subject DN

PKI では、実世界のエンティティを識別するための方法として識別名(DN:DistinguishedName)を用いる。このため証明書の主体者や発行者は全て識別名によって表記される。エンティティ同士の信頼関係を確認する最も容易な方法は、証明書の発行者と、発行者証明書の主体者が一致するかどうか確認することである。

5.2.3 serialNumber と CRL

識別名は、実世界のエンティティと証明書を結びつけるものだが、証明書自体を区別する方法として serialNumber が用意されている。この serialNumber は、ある発行者が発行する全ての証明書において一意でなければならない。

一般に CRL は、その証明書の発行者が発行するものであり、従って CRL においては serialNumber を記載することで失効した証明書を特定している。

5.2.4 validity

証明書が証明する対象は、ある長さ(鍵長)を持った鍵ペアであるが、一般に公開鍵暗号に

はいわゆる(暗号強度に対する)寿命が存在する。このため証明書においても、鍵強度に適切な寿命を設けるべく有効期間(notBefore,notAfter)が記載されている。

有効期間外の証明書は、既に十分な鍵強度を失っているなど信頼性に問題があるものとみなすべきである。

5.2.5 keyUsage 拡張と extKeyUsage 拡張

証明書は鍵ペア(特に秘密鍵)の所有を通じてエンティティの本人性を証明するものだが、この鍵ペアの利用用途を限定したい場合が考えられる。主な鍵用途を規定したものが keyUsage 拡張であり、さらに具体的な用途を規定するための extKeyUsage 拡張がある。

keyUsage 拡張では、既に定義済みの以下の 8 種類の用途のみを記載できる。

- digitalSignature
- nonRepudiation
- keyEncipherment
- dataEncipherment
- keyAgreement
- keyCertSign
- cRLSign
- encipherOnly
- decipherOnly

extKeyUsage 拡張では、既に定義済みの以下の用途の他、各 PKI ドメインが独自定義した用途を記載することも可能である。

- anyExtendedKeyUsage
- serverAuth
- clientAuth
- codeSigning
- emailProtection
- timeStamping
- OCSPSigning

例えば TLS 認証ではサーバ証明書の keyUsage 拡張には digitalSignature と keyEncipherment が、クライアント証明書の keyUsage 拡張には digitalSignature が少なくともセットされている必要がある。また、各証明書を TLS 認証用途に限定したい場合、認証局は、それぞれの extKeyUsage 拡張に serverAuth や clientAuth のみをセットして発行することで、他の用途には使えないようにすることができる。(複数の用途を記載することも可能)

5.2.6 subjectAltName 拡張/issuerAltName 拡張

インターネットには、既存の識別子として IP アドレスや FQDN、URI などが存在する。このため認証においても、鍵ペアを所有するエンティティとして識別すべきは IP アドレスであったり FQDN である場合がほとんどである。例えば Web サーバを認証する場合、Web サーバの FQDN または URI を証明書と関連づけられる必要がある。しかし、一般に証明書のエンティティ名として用いられる DN は directoryName であり、FQDN や URI などインターネットで用いられる識別子を直接記載するには無理がある。[7]

このため、FQDN や URI、IP アドレスなどを記載できる項目として issuerAltName 拡張や subjectAltName 拡張が規定された。両拡張で記載できる項目は以下の通りである。

- otherName
- rfc822Name(*)
- dNSName(*)
- x400Address
- directoryName
- ediPartyName
- uniformResourceIdentifier(*)
- iPAddress(*)
- registeredID

(*) インターネット上での認証に有用な項目

インターネット上での認証においては、これらの拡張を活用することで、より適切なエンティティの識別を行うことができる。

5.2.7 cRLDistributionPoints 拡張

前項 5.1.2 にて、失効検証に必要な CRL はリポジトリから取得する旨を記述した。しかしそもそも X.509 は、X.500 プロトコルを意識して策定された仕様であり、CRL は、原則として X.500 ディレクトリの証明書発行者エントリで公開されることになっていた。しかし、発行者エントリ以外で CRL を公開する場合や、インターネット上のように、アクセス方法とアクセス場所を一意に明記(URI など)しなければならない場合も想定されるため、CRL へのアクセス方法や場所を明記する cRLDistributionPoints 拡張が定義された。

---

[7] NOTE: 今日の多くの Web サーバ証明書は、下位互換に配慮して subject の commonName に Web サーバの FQDN を入れている場合が多いが、directoryName の一部に FQDN を含めることは無理がある。

インターネット上で CRL を取得するほとんどの場合には、この cRLDistributionPoints 拡張を参照する必要がある。言い換えれば、失効検証するためには、多くの場合この拡張を参照して CRL を取得する必要がある。[8]

5.2.8 authorityInfoAccess 拡張

前項 5.1.2 にて、失効情報を OCSP レスポンダから取得するケースがある旨を記述した。これは X.509 ではなく RFC3280 による独自のインターネット拡張である。CRL による失効検証は、CRL の肥大化や、CRL の更新周期など、いくつかの問題があるため、これを解消するために、低トラフィックとリアルタイムな失効情報の提供を実現したのが RFC2560 に基づく OCSP レスポンダである。

RFC3280 では、この OCSP レスポンダから失効情報を取得するために必要な拡張として、OCSP レスポンダへのアクセス方法と場所を明記する authorityInfoAccess 拡張を定義している。

[島岡]

[8] NOTE: 一部の PKI アプリケーションでは、正しく失効検証機能を実装してないためにこの拡張を無視するものもあり、注意が必要である。

6　ネットワーク構成と実験機材

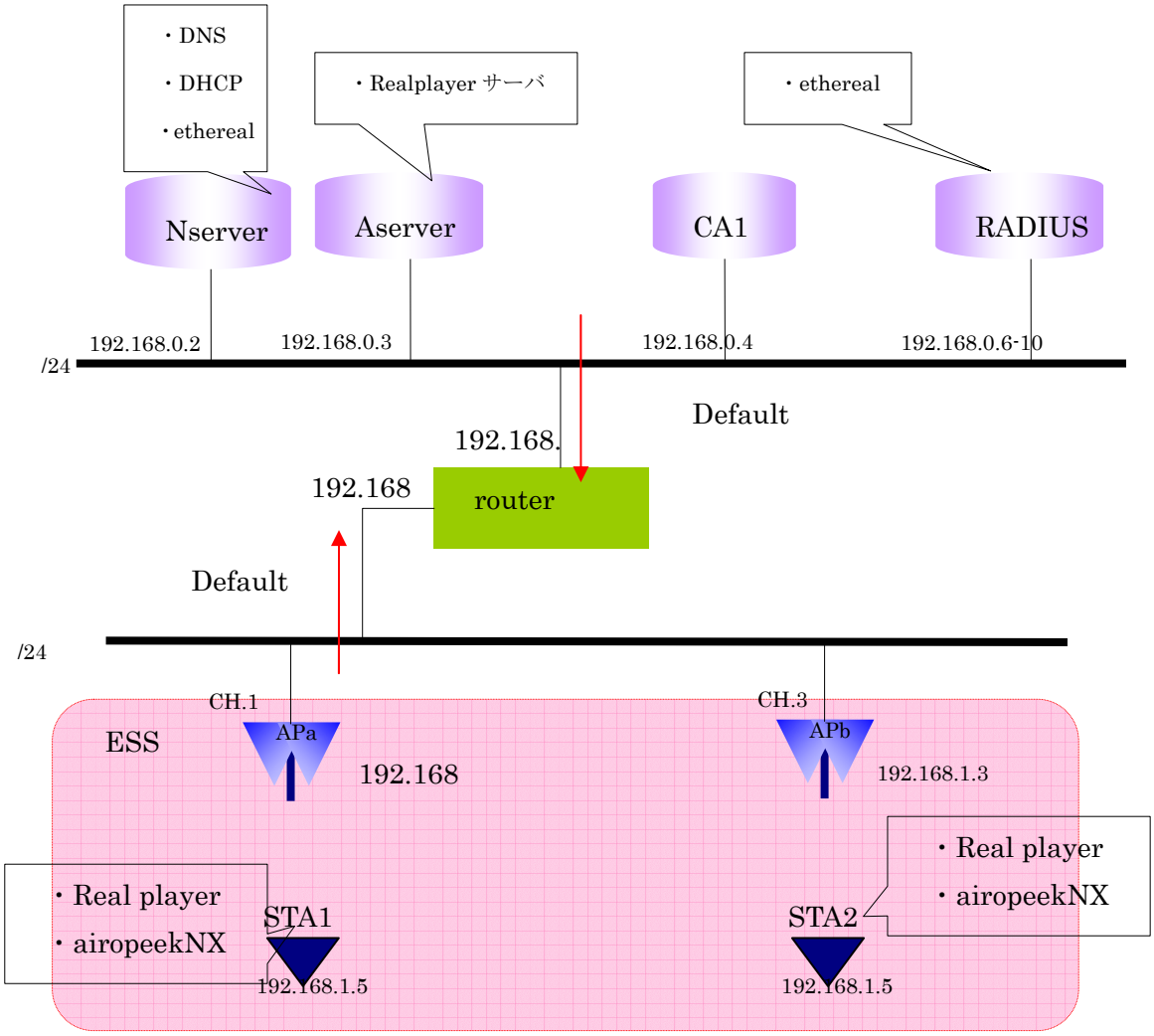

図 6-1 実験ネットーワークの基本トポロジー

機材	製造元	製品
RADIUS	Cisco Systems	ACS
	FreeRADIUS Project	FreeRADIUS
	アクセンス・テクノロジー	fullflex wireless
	Microsoft	Windows Server2003
	Funk software	Odyssey
AP	Cisco	Aironet1200
	Intel	Intel PRO Wireless AP
	ORiNOCO	AP1200
	メルコ	Air Station
Supplicant	Microsoft	Windows XP 標準機能：SP 無し
	Cisco Systems	ACUS
	Funk software	Odyssey Client
	Microsoft	Microsoft 802.1X Authentication Client
	UDTech Japan	MPWorks
	メルコ	Air Station (付属ソフトウェア)

表 6-1　実験機材リスト

7　実験項目目的と実験

7.1　概要

この章では802.1X機能及び相互接続性について検証するため行ったさまざまな実験の目的と方法を示す。802.1X の相互接続性を確かめるため、数種類の Supplicant、アクセスポイント、RADIUS を用意し、EAP-TLS、EAP-TTLS、PEAP の順で実験を行うこととした。

802.1X の相互接続実験の実施に先立ち、802.11b での接続の実験を行い、各機器の操作方法の確認、接続性の確認を十分に確認した。通常の 802.11b での接続や挙動を確認しておくことで 802.1X の実験中に不要な要因でトラブルを避けるためにもなった。

図 6-1 に示すように実験ネットワークトポロジーが構成されており、各構成機材は表 6-1 のようになっている。

7.2　EAP-TLS 実験

●Supplicant、RADIUS に依存する実験

7.2.1　最小構成の証明書プロファイル

■実験目的

特定の Supplicant、RADIUS サーバを選び TLS ネゴシエーションができる最小構成の証明書プロファイルを見つける。

■実験方法

1.　サーバ証明書プロファイルの評価

サーバ証明書(正常系)についてプロファイルセット A～D まで変化させて、RADIUS サーバの動作を確認することで、実験に適切なプロファイルセットを選択する。

各プロファイルセットについて RADIUS サーバの動作検証を行った。RADIUS サーバに設定するサーバ証明書について、プロファイルセット A～D までの各サーバ証明書を用いて、クライアントとの EAP-TLS 認証ができるかどうか確認した。

	keyUsage (critical)	subject AltName	extend KeyUsage
プロファイルセットA	○	○	○
プロファイルセットB	○	×	○
プロファイルセットC	○	×	×
プロファイルセットD	×	×	×

表 7-1 サーバ証明書プロファイルセット

2. クライアント証明書プロファイルの評価

クライアント証明書(正常系)についてプロファイルセットA～Dまで変化させて、クライアント(Supplicant)の動作を確認することで、実験に適切なプロファイルセットを選択する。

各プロファイルセットでクライアント(Supplicant)が正常に動作するか確認した。Supplicantに設定するクライアント証明書について、プロファイルセットA～Dまでの各クライアント証明書を用いて、RADIUSサーバとのEAP-TLS認証ができるかどうか確認した。

	keyUsage (critical)	subject AltName	cRL DistPoint	extend KeyUsage
プロファイルセットA	○	○	○	○
プロファイルセットB	○	×	○	○
プロファイルセットC	○	×	○	×
プロファイルセットD	×	×	○	×

表 7-2 クライアント証明書プロファイルセット

■ 実験結果

Supplicant SU2 以外は C もしくは D の証明書プロファイルセットを使った認証ができなかった。

Radius：AS1

プロファイル		Supplicant		
クライアント	サーバ	SU1	SU2	SU3
A	A	○	○	○
B	B	○	○	○
C	C	×（注1）	○	×（注1）
B	C	×（注2）	○	×（注2）
D	D		○	

注 1: 認証失敗クライアントが証明書を伝送しない

注 2: 認証は成功するが通信は不可

表 7-3 実験結果：証明書プロファイル１

Radius : AS2

プロファイル		Supplicant		
クライアント	サーバ	SU1	SU2	SU3
A	A	○	○	
B	B	○	○	○
C	C		○	
B	C	×（注2）	○	×（注2）
D	D		○	

表 7-4 実験結果:証明書プロファイル 2

Radius : AS4

プロファイル		Supplicant		
クライアント	サーバ	SU1	SU2	SU3
A	A	○		
B	B	○		
C	C	×(注3)		
B	C	×(注3)		
D	D			

注 3: 認証不可

表 7-5 実験結果:証明書プロファイル 3

7.2.2 RADIUS サーバでの CRL の認識

■実験目的

実験 7.2.1 で選択したクライアント証明書プロファイルセットを利用して、CRL に関するテストを行う。RADIUS サーバが CRL を正しく認識できるかどうか実験する。

■実験方法

クライアントから失効系証明書を用いて AccessPoint へアクセスし、TLS 認証が失敗することを確認する。

■実験結果

Supplicant はすべて SU1、プロファイルはすべて A を使用した。

RADIUS	RADIUS CRL指定あり		RADIUS CRL指定なし	
	クライアント証明書CRLDP指定		クライアント証明書CRLDP指定	
	あり	なし	あり	なし
AS1	認証失敗	認証失敗		
AS2			認証成功	認証成功

注: AS1 では CRL 指定はスタティック、CRLDP チェックは未実装

AS2 では CRL 指定は未実装

表 7-6 実験結果

7.2.3 Supplicant が CRL を認識するのか

■実験目的

Supplicant が CRL を正しく認識できるかどうかの確認。

■実験方法

サーバ証明書 CRLDP 指定をするときとしないときで各 Supplicant が CRL を認識するか確認した。

■実験結果

Radius はすべて AS1、プロファイルはすべて A を使用した。サーバ証明書 CRLDP 指定をするときは、SU1、SU2 ともに CRL を正しく認識できたが、CRLDP 指定がないときは SU2 では CRL の認識ができなかった。また SU3 は SU1 と同様の性質を持っていると判断したためこの実験以降では SU3 は特に実験しなかった。

Supplicant	サーバ証明書CRLDP指定	
	あり	なし
SU1	○	○
SU2	○	

注: OS に CRL をインストールした。

表 7-7 実験結果

7.2.4 期間切れの証明書

■実験目的

クライアント環境の証明書が期限切れになった際に、TLS 認証が成功するか確認した。

■実験方法

期限切れのクライアント証明書を用いてクライアントからの TLS 認証が失敗することを確認した。

■実験結果

SU1 ではクライアントが証明書を送信しないために認証失敗となった。 SU2 では RADIUS が Alert メッセージを出すので認証は成功しないが、クライアントは TLS ハンドシェイクを継続してしまうという現象が見られた。AS3 は実験用に用意した証明書を取り込む作業ができなかったため実験を行わなかった。

Supplicant	RADIUS			
	AS1	AS2	AS3	AS4
SU1	○	○		
SU2	○	○		

注: ○は認証失敗（期待したとおり）

表 7-8 実験結果

7.2.5　　信頼できない CA

■実験目的

次の図のように RADIUS が TLS ハンドシェイク中に、Supplicant の信頼していない CA 証明書を含めた不正な証明書チェーンを送信した場合、認証ができるのかどうか確認する。

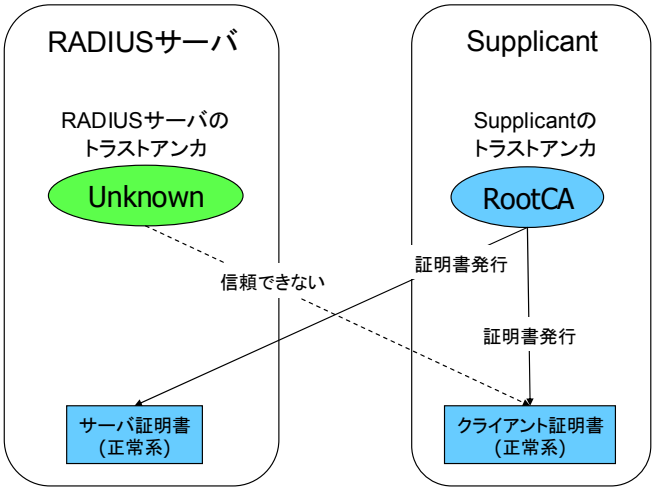

図 7-1 信頼できない CA

■実験方法

以下の手順で実験を行った。

1. サーバ、クライアントそれぞれについて以下の通りに用意する。

	サーバ環境	クライアント環境
トラストアンカ証明書	Unknown CA 証明書	RootCA 証明書
サーバ証明書	RootCA が発行した証明書	
クライアント証明書		RootCA が発行した証明書
CRL	Unknown CA が発行した CRL	RootCA が発行した CRL (使用しなくてもよい)

表 7-9 サーバ、クライアントに入れる証明書

2. 上記環境で、クライアントからの TLS 認証が失敗することを確認する。

■実験結果

SU1、SU2 で期待通り認証が失敗することを確認できた。しかしながら、先ほどの実験と同様に SU2 では RADIUS が Alert メッセージを出すので認証は成功しないが、クライアントは TLS メッセージを継続してしまうという現象が見られた。

Supplicant	RADIUS			
	AS1	AS2	AS3	AS4
SU1	○	○		
SU2	○	○		

注: ○ 認証失敗（期待した結果どおり）

表 7-10 実験結果：信頼できない CA

7.2.6 セッションタイムアウトの動作確認

■実験目的

セッションタイムアウトのトリガが AP に設定されたタイムアウト値なのか RADIUS のセッションタイムアウトのアトリビュートなのかを調べ、期待されるセッションタイムアウト時に再認証が行われることを確認する。

■実験方法

以下の手順で実験を行った。

1. AP を 1 台だけ使用する。
2. AP もしくは RADIUS サーバで再認証要求間隔設定する。
3. STA1 を無線 LAN に接続。
4. RADIUS サーバの Ethereal で 1812 ポートに対してパケット Capture を行う。2. で設定した再認証要求間隔経過後に再認証要求が発せられるか確認する。
5. 再認証の手順がリセッション手順となっているか確認する。

■実験結果

実験したすべての AP でセッションタイムアウトの動作が確認できた。AP1 では Radius の session timeout attribute による再接続要求が有効であり、AP 自体には設定することができなかった。AP2 では RADIUS、AP 両方で再認証要求間隔が設定可能だったが、AP3 では AP での設定のみ有効だった。

AP	RADIUS			
	AS1	AS2	AS3	AS4
AP1	R	R	R	R
AP2	RA	RA	RA	RA
AP4	A	A	A	A

注: R: Radius の session timeout attribute 設定によって制御できた。

A: AP の設定で session timeout 設定によって制御できた。

表 7-11 実験結果:セッションタイムアウト

7.2.7 複数の CA

■実験目的

次の図のように１つの無線 LAN ネットワークの中に複数の CA から発行された証明書が混在しているときに正常に認証ができるかどうか確認する。

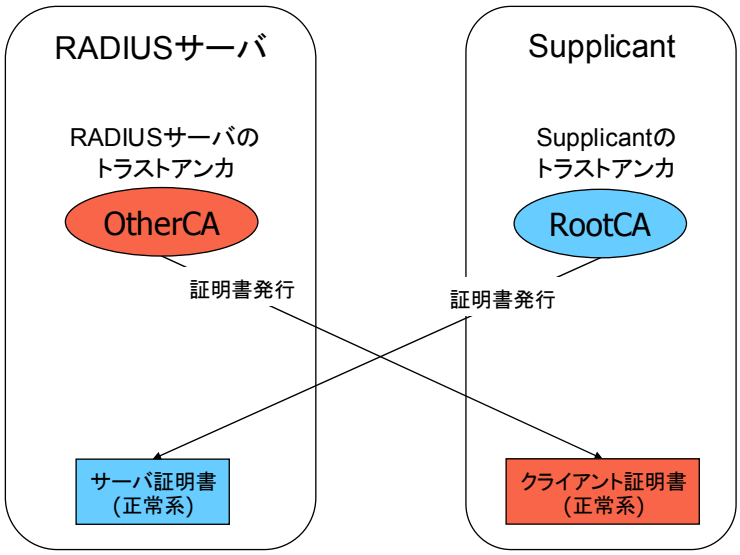

図 7-2 複数の CA

■実験方法

以下の手順で実験を行った。

1. サーバ、クライアントそれぞれについて以下の通りに用意する。

	サーバ環境	クライアント環境
トラストアンカ証明書	(表 7-1 を参照)	RootCA 証明書
サーバ証明書	r<X>sv-normal.p12 (実験 7.2.1 で選択したもの)	
クライアント証明書		o<X>cl-normal.p12 (X:実験 7.2.1 で選択したもの)
CRL	(表 7-1 を参照)	RootCA が発行した CRL (使用しなくてもよい)

表 7-12 サーバ、クライアントに入れる証明書

2. RADIUS サーバのトラストアンカを以下のように指定して、クライアントからの TLS 認証が成功するか確認する。

	サーバ環境	
	トラストアンカ 証明書	CRL
シングル トラストアンカ	OtherCA 証明書	OtherCA が発行した CRL
マルチ トラストアンカ	RootCA 証明書 OtherCA 証明書	RootCA が発行した CRL OtherCA が発行した CRL

表 7-13 トラストアンカ証明書と CRL

■実験結果

1) シングルトラストアンカの場合

Supplicant	RADIUS			
	AS1	AS2	AS3	AS4
SU1	○	○		
SU2	○	○		

表 7-14 実験結果:シングルトラストアンカ

2) マルチトラストアンカの場合

Supplicant	RADIUS			
	AS1	AS2	AS3	AS4
SU1	○	○		
SU2	○	○		

表 7-15 実験結果:マルチトラストアンカ

7.2.8 サブジェクトによる認証

■実験目的

TLS 認証に加えて、RADIUS がクライアント証明書のサブジェクト DN を用いたアクセス制御を行えるかどうか確認する。

■実験方法

以下の手順で実験を行った。

1. AP を 1 台だけ使用する。
2. RADIUS サーバにてサブジェクト認証を有効にする。
3. RADIUS サーバにて、クライアント証明書のサブジェクト DN のみに対するアクセス許可を設定する。
4. STA1 が無線 LAN に接続できることを確認する。 3. で登録したサブジェクト情報を RADIUS サーバから削除する。
5. STA1 がサブジェクト認証によるアクセス拒否されることを確認する。

■実験結果

AS1 のみサブジェクト認証に対応していたので、ほかの RADIUS での実験は行わなかった。

1) サブジェクト認証確認

Supplicant	RADIUS			
	AS1	AS2	AS3	AS4
SU1	○			

表 7-16 実験結果:サブジェクト確認

2) サブジェクト削除（変更）後の認証確認

Supplicant	RADUIS			
	AS1	AS2	AS3	AS4
SU1	接続不可			

表 7-17 実験結果:サブジェクト削除後

●Supplicant、RADIUS サーバ、AP 依存実験

7.2.9 Dynamic な WEP キーの更新

■実験目的

再認証時に WEP キーが動的に変更されるかを確認する。

■実験方法

以下の手順で実験を行った。

1. AP を 1 台だけ使用する。
2. STA1 を無線 LAN に接続した状態で Nserver に ping をうったまま無線パケットキャプチャーツールで Capture しつづけ、同時に RADIUS サーバでも Ethereal でパケット Capture を行う。
3. ping のパケットが流れる際に EAP-TLS の再認証が行われ、EAPOL-Key が流れるのかを確認する。

■実験結果

どの組み合わせでも動的に WEP キーが変更されることを確認した。

AP	RADIUS	動的な WEPキー
AP1	AS1	○
	AS2	○
	AS3	○
	AS4	○
AP2	AS1	○
	AS2	○
	AS3	○
	AS4	○
AP4	AS1	○
	AS2	○
	AS3	○
	AS4	○

注：AP4 に関しては認証と鍵の更新は連動せず

表 7-18 実験結果

7.2.10 WEP キー更新時の通信の安定性

■実験目的

動的に WEP キーが更新されるときの通信の安定性について確認する。

■実験方法

以下の手順で実験を行った。

1. AP を 1 台だけ使用する。
2. RADIUS サーバでは Ethereal で再認証が行われるか確認する。
3. RADIUS で再認証のパケットが流れる時に STA1、STA2 で ping をうって、どれだけ欠落するか確認する。 ping は ExPing を用いて実行間隔が最小としできるだけ早く ping をうつように設定する。

■実験結果

ping の欠落結果は以下のようになった。機器の組み合わせによって ping の欠落があまり違いがでたようには感じられなかった。あくまでも ping の欠落結果は参考程度にとどめておいたほうがよい。

AP2 では ping で負荷をかけた場合再認証時 Supplicant の EAP-Response を AP が RADIUS に返さず何度も試行するが失敗することが多かったため実験を取りやめた。 また、AP4 においては認証と鍵は通信切断をしない設計のため実験を行わなかった。

AP	RADIUS	Supplicant	パケットの欠落
AP1	AS1	SU1	Pingは2発落ちた
		SU2	Pingは4発落ちた
	AS2	SU1	Pingは3発落ちた
		SU2	pingが30個落ちた(注)
	AS3	SU1	ログなし
		SU2	Pingは1発落ちた
	AS4	SU1	Pingは3発落ちた
		SU2	Pingは2発落ちた

注：パケットログを見ると怪しいため結果の信憑性は低い

表 7-19 実験結果

7.2.11 Unicast key の配布形態

■実験目的

接続時に Unicast key が配信されるかどうかの確認と Unicast key がユーザごとに異なるかを確認

■実験方法

以下の手順で実験を行った。

1. AP を 1 台だけ使用し、STA1、STA2 を接続する。
2. 無線のパケットキャプチャーツールを用いて EAPOL-Key の中身を確認し、デバッグ用サプリカントを用いて WEP キーを表示させ設定される Unicast key を確認する。
3. STA1 の鍵と STA2 の鍵を比較する。

■実験結果

AP1 と AP2 ではユーザ毎に異なる Unicast key を生成されることを確認した。AP4 では AP 全体で 3 つの Unicast key しか持てないため、接続クライアントが増えると同じ Unicast key を使う可能性が確認できた。

AP	ユーザ別の暗号鍵
AP1	ユーザ毎に認証による鍵作成
AP2	ユーザ毎に認証による鍵作成
AP4	ユーザ毎の鍵ではない (3つの鍵をすべてのユーザが共有する。)

表 7-20 実験結果

7.2.12 Broadcast key の配布形態

■実験目的

接続時に Broadcast key が配信されるかどうかの確認

■実験方法

以下の手順で実験を行った。

1. AP を 1 台だけ使用し、STA1、STA2 を接続する。
2. デバッグ用サプリカントを用いて WEP キーを表示させ設定される Broadcast key を確認する。
3. STA1 の鍵と STA2 の鍵を比較する。

■実験結果

実験したどの AP においても Global キーが配信されることが確認できた。

AP	EAP-Keyの種類と状態
AP1	Broadcast keyは配布 Unicast keyは鍵をゼロで配布
AP2	Broadcast keyは配布 Unicast keyは鍵をゼロで配布
AP4	Broadcast key、 Unicast keyともに配布

表 7-21 実験結果

7.2.13 アカウンティング処理機能

■実験目的

AP のアカウンティング機能を有効にしたときにアカウンティング処理が行われるか確認する。

■実験方法

以下の手順で実験を行った。

1. AP を 1 台だけ使用する。
2. AP のアカウンティング機能を有効にする。
3. RADIUS サーバにて Ethereal で 1813 ポートに対してパケット Capture を行う。
4. STA1 を無線 LAN に接続する。
5. アカウンティングパケットが送受信されていることを確認する。
6. STA1 の無線 LAN 接続を解除する。

■実験結果

AP がアカウンティング機能を持っていれば、アカウンティング処理が確認できた。

AP	Accountingの有無
AP1	Accounting機能あり
AP2	Accounting機能なし （設定方法不明）
AP4	Accountingは可能 （GUIでのAccounting設定は出来ない）

表 7-22 実験結果

7.2.14 認証にかかる時間

■実験目的

以下の環境で認証を行った際に認証が終了するまでの時間がどれくらいかかるのか確認する。

■実験方法

以下の手順で実験を行った。

1. AP を 1 台だけ使用する。
2. 無線 LAN キャプチャーツールで Capture する間、STA1 から無線 LAN に接続する。
3. Capture 結果からセッションタイムアウトによる再認証で EAP-Request/Identity が流れてから EAPOL-key の時間の差分を計算し、実験結果に記録する（この時サーバ、クライアントの証明書が交換されることを確認）。
4. 1 回目は通常の接続開始の認証にかかる時間、2 回目は ping の負荷をかけた状態での再認証にかかる時間、3 回目はストリーミングコンテンツを受信した状態での再認証にかかる時間をそれぞれ計測する。

■実験結果

意外なことにほとんどの組み合わせにおいて通常の環境下での認証が一番時間がかかるという結果が得られた。RADIUS に AS3 を使った時は ping による負荷をかけた結果がもっとも時間がかかったという結果が得られた。これらの値は無線キャプチャーツールを用いて計測したため、時間を保証するものではなくログを採取した条件によって変動するためあくまで参考程度として頂きたい。

AP	RADIUS	Supplicant	再認証にかかった時間
AP1	AS1	SU1	1回目:0.99 2回目:0.01（注1） 3回目:-（注2）
		SU2	1回目:2.23 2回目:1.32 3回目:-（注2）
	AS2	SU1	1回目:0.93 2回目:0.86 3回目:0.81
		SU2	1回目:1.41 2回目:0.52 3回目:0.60
	AS3	SU1	1回目:1.60 2回目:3.48 3回目:1.33
		SU2	1回目:0.91 2回目:5.81 3回目:0.73
	AS4	SU1	1回目:0.91 2回目:0.21 3回目:0.42
		SU2	1回目:0.74 2回目:0.62 3回目:0.57

注１： 一方的な Request のみ

注２： 測定失敗

表 7-23 実験結果

7.2.15 Fast Reconnect による再認証

■実験目的

Fast Reconnect 認証での再認証と通常の再認証とでのかかる時間や挙動の違いについて確認する。

■実験方法

以下の手順で実験を行った。

1. APa 電源 ON、APb 電源 ON とし、STA2 を APb に接続する。
2. 無線用キャプチャーツールで Capture した状態で、STA1 を APa に接続した後 STA1 の接続を切って APb に接続を行う。
3. 再認証の際 RADIUS サーバ STA1 の証明書が流れているか確認し実験結果に記録する。
4. 再接続の EAPOL-START が流れてから EAP-Success が流れるまでの時間の差分も

記録する。

■実験結果

Fast Reconnect 認証に対応する Supplicant はなかったため実験を行わなかった。

7.3 EAP-TTLS 実験

実験を行う時点で対応する製品が少なく、相互接続実験は取りやめた。

7.4 PEAP 実験

実験を行う時点で対応する製品が少なく、相互接続実験は取りやめた。

[寺島]

8　結果と考察

8.1　PKI 関連

8.1.1　EAP-TLS 用証明書プロファイル

本実験では、各実装で要求される最小限の証明書プロファイルを特定するために、いくつかの証明書プロファイルを予め用意し、どの証明書プロファイルが利用できるか確認した。なお、これら証明書プロファイルに依存するのは RADIUS サーバと Supplicant であり、アクセスポイントなどは証明書プロファイルに依存しないため、実験対象外としている。

本実験では、EAP-TLS に重要と思われる四つの証明書拡張について、その有無と証明書利用との依存関係を調べた(表 7-1、表 7-2 参照)。

a) keyUsage 拡張

b) subjectAltName 拡張

c) extendKeyUsage 拡張

d) cRLDistributionPoint 拡張(クライアント証明書のみ)

最初にこれらの拡張について、関連する PKI アプリケーションなども含めた各種 RFC での記述を簡単に説明する。

a) keyUsage 拡張

TLS(RFC2246)では、各証明書が keyUsage 拡張を用いるのであれば少なくとも digitalSignature が必要で、用途に応じて keyEncihperment や keyAgreement が必要だと述べている。このため本実験ではサーバ、クライアントともに digitalSignature(認証用途)と keyEncipherment(暗号用途)の二点を明記した証明書と、keyUsage 拡張自体を含めない証明書を用意してサーバ、クライアントの動作をそれぞれ比較した。

b) subjectAltName 拡張

EAP-TLS では全く規定されていないが、S/MIME や HTTPoverTLS などでは、送信者メールアドレスや、接続元(クライアント)ホスト名などが subjectAltName に記述された rfc822Name や dNSName に一致していなければならない、といった規定がある。仕様には規定されてないものの、EAP-TLS の実装においても同様な制約があるかどうかについて検証した。

具体的には subjectAltName 拡張に rfc822Name を含めたクライアント証明書と、dNSName にサーバ名を含めたサーバ証明書を用意し、subjectAltName 拡張を含めない証明書との挙動を比較した。

c) extendKeyUsage 拡張

この拡張の用法について、TLS も EAP-TLS も何ら規定していない。一方 RFC3280 を読

む限り、extendKeyUsage 拡張を用いる証明書は、明記されている用途以外に使ってはいけない、と述べられている。extendKeyUsage 拡張で規定されている用途の中に clientAuth/serverAuth といった TLS 認証の用途が明記されている。このため、本実験ではクライアント証明書の extendKeyUsage 拡張に clientAuth を、サーバ証明書の extendKeyUsage 拡張に serverAuth を含めたものと、両証明書に extendKeyUsage を含めない証明書との挙動を比較した。

d) cRLDistributionPoints 拡張(クライアント証明書のみ)

インターネット上で各証明書の失効検証をするためには、ほとんどのアプリケーションが cRLDistributionPoints 拡張を必要とする。このため、本実験でも、cRLDistributionPoints 拡張を含めたクライアント証明書と、cRLDistributionPoints 拡張を含めない証明書とで挙動を比較した。

表 7-3、表 7-4、表 7-5 の結果にあるように、一部の Supplicant を除き、ほとんどの組み合わせにおいて証明書プロファイルセット A～B が通用することが確認できた。

プロファイルセット B では各証明書の subjectAltName 拡張を外しており、RADIUS サーバ、Supplicant ともに subjectAltName 拡張は不要であると考えられる。また、S/MIME や HTTP over TLS などで求められているような、subjectAltName 拡張を用いたソースアドレスの検証といったことも特に必要ではない[9]と思われる。一方プロファイルセット C では、各証明書の extendedKeyUsage 拡張を外していることから、これらの RADIUS サーバ、Supplicant ではともに extendedKeyUsage 拡張を必須としていることが確認できた。

しかし、これは EAP-TLS 認証用の証明書を、他の用途に使ってはいけないことを意味する。例えば TLS 認証は、clientAuth/serverAuth の用途に相当するので利用できるが、S/MIME などでの電子署名や暗号用途には使えないことになる。関連 RFC を読む限りでは本来 EAP-TLS 証明書には extendKeyUsage は「必須ではない」のだが、このような実装が多い現状はユーザに誤解を与えかねず、また利便性を損なう可能性がある。

証明書の用途を限定することはセキュリティを高める要素ではあるものの、それは本来証明書を発行する認証局が制御するものであり、認証局の発行ポリシと関係なくアプリケーション側が要求すべきことではないと考える。

8.1.2 802.1X における証明書検証

(1) RADIUS サーバにおける認証要件

[9] 一部の Supplicant では、任意の追加設定により subjectAltName との一致を要求することも可能。

EAP-TLS(RFC2716)では、証明書の失効について考察している。その中では、RADIUS サーバについてはインターネット接続を確立している状態なので失効検証をすることができる、としている。この時、サーバがインターネット上からCRLを取得するにはcRLDistributionPoints拡張を参照する必要がある。

表 7-6からわかる通り、一部のRADIUSサーバではcRLDistributionPoints拡張を参照して失効検証を行うことができたが、失効検証を実装していない RADIUS サーバもいくつかあり、全体としてPKI を用いた厳密な認証・暗号化を行っているメリットが損なわれてしまっているように感じられた。失効検証を行わない限り証明書は単に公開鍵を交換するためのフォーマットでしかなく、結果的に暗号化目的でしか使用されていないことになる。もともとEAPを含めた802.1Xは認証フレームワークであることから、現状の実装状況は RADIUS サーバとして不十分であると思われる。認証に必要な機能は正しく実装・提供されるべきであり、今後 RADIUS ベンダによる対応改善が望まれる。

ここで、RADIUSサーバが実装すべき失効検証の手法について考察してみる。RADIUSサーバ、特に無線 LAN のように認証/再認証が頻繁なケースでは、失効検証に必要な CRL を認証の都度取得していては、RADIUS サーバへの負荷(遅延時間についても触れる)が高まってしまう。また、CRL には肥大化問題や定時更新などの特徴があることから、一度取得した CRL を RADIUS サーバ内部で(次回更新時まで)キャッシュしておくのが効果的であると考えられる。

なお証明書の失効検証には、CRL 以外にも OCSP レスポンダを用いるモデル(RFC2560)も存在する。OCSP レスポンダは、CRL のような肥大化問題を解消し、リアルタイムな失効情報の提供を目的としたシステムだが、RADIUSサーバのような用途では失効検証の頻度(遅延時間)が圧倒的なボトルネックになってしまうため、おそらく OCSP レスポンスについてもキャッシュせざるを得ないだろう。

しかしOCSPレスポンスは実はキャッシュに向いていない。何故ならOCSPレスポンスには、CRLのような肥大化問題を解決するために、OCSP リクエスタから要求された証明書に関する失効情報のみが記述されている。このため、OCSPレスポンスのキャッシュは、特定の証明書にしか再利用できない。大規模なRADIUSサーバなどでは、CRL肥大化を回避できてもOCSPレスポンスのキャッシュが肥大化するという副作用が発生し、OCSP プロトコルへの対応の手間などを考えるとメリットは薄いと思われる。

このような点も含めると、RADIUS サーバという用途には CRL キャッシュを用いるのが最もリーズナブルな解と考えられる。ただし、認証の頻度や厳密性によっては、OCSP モデルやオンライン CRL などによる失効検証も有効であるので、利用者は、導入用途に合った機能を実装しているRADIUSサーバを選択する必要がある。

(2) Supplicant での認証要件

EAP-TLS(RFC2716)では、クライアントにおける失効検証についても考察している。クライアント

では PPTP や L2TP の場合を別として、PPP では NCP ネゴシエーションが完了するまでインターネット接続を確立できないために失効検証ができないかもしれない、と考察している。そのため、クライアントはインターネット接続後に失効チェックをするべきである、と述べている。しかしこれは、認証フローと失効検証フローが切り離されるため、多くの Supplicant にとっては難しい実装であると考えられる。

8.1.3 802.1X(認証)における信頼モデル

PKI ドメインの典型例として階層モデルが知られているが、世の中の実情は単一の階層モデルで構築されているわけではない。例えば A 社をルート認証局とする PKI ドメインで発行されたサーバ証明書と、B 社をルート認証局とする PKI ドメインで発行されたクライアント証明書を用いて認証できる技術が確立されているべきである。信頼できる必要があるかどうかは、各ドメイン間の信頼関係による。このためには、複数の PKI ドメインにまたがった認証パスの構築と検証を行える必要がある。

複数の PKI ドメインにまたがった認証パスは、現在の実装状況では大きく二種類に分類することができる。一つは GPKI などに用いられている、相互認証証明書を用いる相互認証モデルであり、もう一つは Web ブラウザなどでの SSL/TLS 認証に用いられているマルチトラストアンカモデルである。前者の相互認証モデルは、認証パスの構築が難しく、必ずしも世の中の多くのアプリケーションがサポートできているわけではない。このため、EAP-TLS においても、Web ブラウザなどと同様マルチトラストアンカモデルへの対応が期待される。

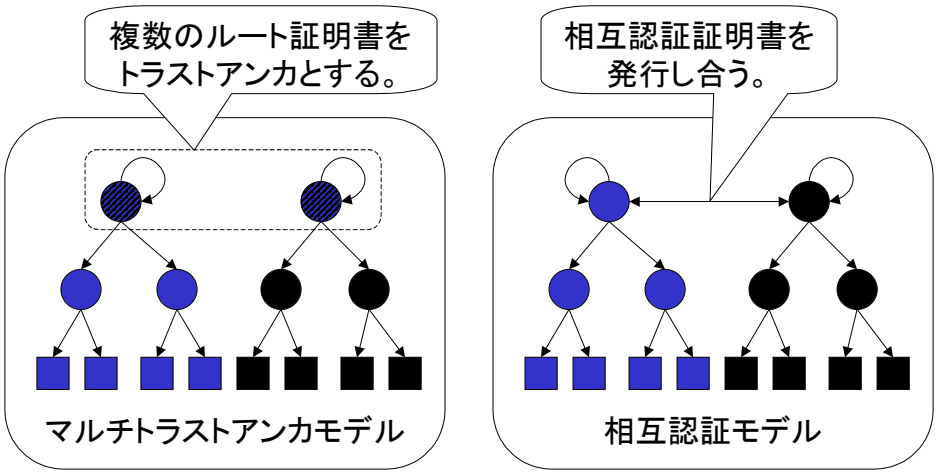

図 8-1 マルチトラストアンカモデルと相互認証モデル

そこで本実験においても、マルチトラストアンカモデルへの対応を検証する実験を行った。マルチトラストアンカモデルにおいて重要なことは、単に他ドメインのエンドエンティティを認証するのでなく、信頼関係に基づいた認証が行えるかどうかである。このため本実験では、信頼関係を持たない他ドメインの証明書について認証できないことも合わせて検証した。

本実験では、いくつかの RADIUS サーバにおいてマルチトラストアンカモデルへの対応が確認できた(表 7-10、表 7-14、表 7-15 参照)。これらの RADIUS サーバでは、いずれも信頼関係のあるドメイン(トラストリストにトラストアンカが指定されているドメイン)についてのみ認証が成功し、信頼関係のないドメイン(トラストリストにトラストアンカが指定されてないドメイン)については認証できないようになっていた

しかし、一部の RADIUS サーバでは、信頼関係を設定できるドメイン数が(おそらく実装上の制約などにより)限定されているなど、より一層の実装の拡充が求められる。

残るRADIUSサーバでは、マルチトラストアンカモデルへの対応が確認できなかったので、これらの製品では単独 PKI ドメイン(同一ルート認証局)下で発行された証明書同士でしか認証することはできない。これらの RADIUS サーバも EAP-TLS クライアント認証するための最低限の機能は実装しているとは言え、複数の PKI ドメインが混在するようなグローバル環境では、運用上必要とされる機能を実装していく必要がある。

8.1.4　　実際のモデルケースにおける 802.1X に対する機能要件

実際に 802.1X と PKI を用いた認証システムを運用する際に、どのような点に注意すべきか考察してみる。まず、このようなシステムを導入する場合には、通常の無線 LAN 環境に加えて、RADIUS サーバと、更に PKI を利用するための認証局やリポジトリが必要となる。このため導入にあたっては、これらの追加コンポーネントをどう扱うかがポイントである。

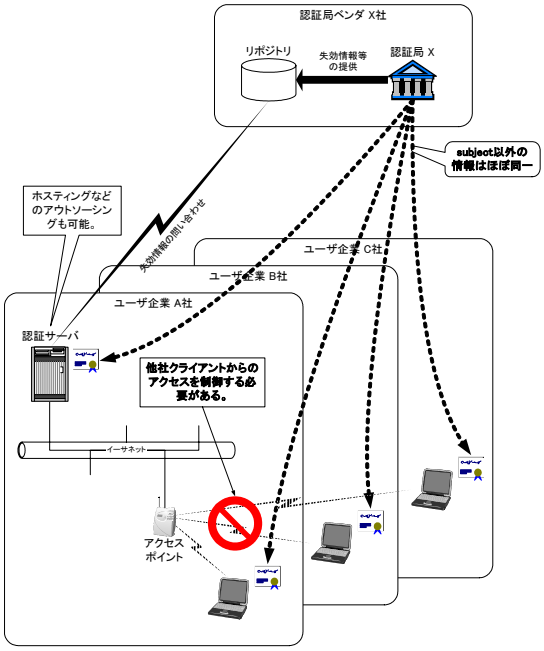

図 8-2　認証局ベンダから証明書を取得

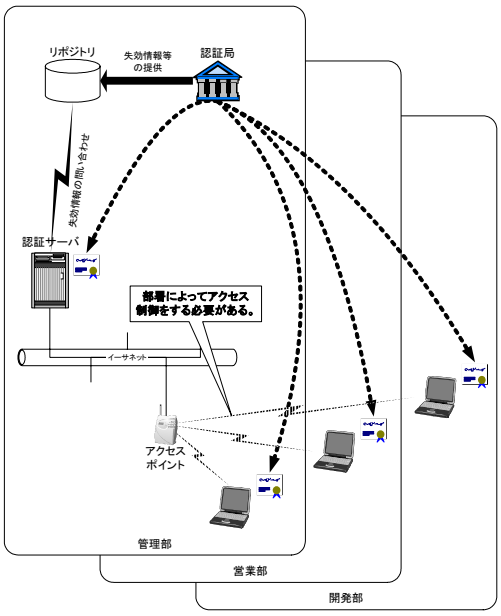

図 8-3　自前で PKI 環境を構築

8.1.4.1　　RADIUS サーバ

RADIUS サーバは、アクセスポイントと RADIUS プロトコルを用いて通信を行うので、必ずしもアクセスポイントと物理的に近接している必要はない。また、(無線 LAN)クライアントの認証には公開鍵証明書を用いるので、RADIUS サーバにユーザの秘密情報を格納する必要がない。このため、外部データセンタのホスティングサービスを利用するなど、アウトソーシングすることも可能である。

ただし、認証頻度やトラフィックを十分考慮しないと、クライアントにとって十分なパフォーマンスを得られない可能性があるので注意が必要である。また、EAP-TTLS や PEAP などクライアント認証に公開鍵暗号を用いないケースでは、RADIUS サーバに各ユーザの秘密情報を格納することになるので、セキュリティ面での配慮が重要となる。

8.1.4.2 認証局とリポジトリ

EAP-TTLS や PEAP のサーバ認証で必要となるサーバ証明書や、EAP-TLS のクライアント認証で必要となるクライアント証明書は、認証局から発行してもらうことになる。加えて、認証の際に証明書の失効検証を行うためには、リポジトリなどから失効情報を取得する必要がある。これらを実現する方法として、第三者的な認証局ベンダなどからサーバ証明書やクライアント証明書を取得する場合と、自社で認証局やリポジトリなど必要なシステムを構築する場合が考えられる。以下、両者の違いについて考察する。

(ア) 認証局ベンダから証明書を取得する場合

認証局ベンダから証明書を取得するメリットは、より安全な証明書を発行してもらえる点にある。一般に認証局ベンダは、認証局を運用するにあたり、認証局運用規定(CPS)を定め公開している場合が多い。CPS の内容は認証局ベンダによって様々であるが、認証局を如何に厳密に運用しているかを示すものである。例えば電子署名法で認定された特定認証業務の認定認証局などは、認証局運用にツーマンルール(常に二人で行動することによって不正を防ぐ)を要求するなど、きわめて厳密である。このような運用ノウハウを持つ認証局ベンダから証明書を取得することで、容易に本人以外の第三者が不正に証明書を利用してしまう事態を防ぐことができる。

また、厳密な証明書であることにより、第三者からの信頼を得やすい。このため電子署名や暗号データを交換する対象が広がるというメリットもある。例えば、社外に電子署名したデータを送る際に、自社認証局から発行された証明書を用いて署名するよりも、信頼ある認証局ベンダから発行された証明書を用いて署名した方が信頼されやすい。

一方で、注意すべき点もいくつかある。TLS 認証では、信頼する側(Relying-Party)が信頼している認証局(Trust Anchor)から、信頼される側(Subscriber)の証明書までの証明書チェーンをたどることができれば、基本的に信頼関係が成立する。図 8-2 で言えば、RADIUS サーバ(Relying-Party)が認証局 X を信頼している(Trust Anchor)場合、この RADIUS サーバは認証局 X が発行する全ての証明書を信頼できてしまう、ということである。

認証局ベンダは、勿論様々なユーザに証明書を発行しているため、このように意図していない認証が成立してしまう可能性がある。そこで、認証局ベンダからクライアント認証に用いる証明書を取得するような場合には、何らかの形で「特定の証明書だけに認証を許可する」仕組みが必要と考えられる。ここで認証を許可したい証明書と、許可したくない証明書の違いは認証局ベンダに依存するが、確実に利用可能な識別子として、subjectDN が挙げられる。このため本実験においても、IEEE や IETF で明確に要件として挙げられてはいないものの、このような subjectDN を用いたアクセス制御が実装されているかどうか検証してみた（表 7-16、表 7-17 参照)。

結果として、この subjectDN によるアクセス制御を実装・機能している RADIUS サーバはごく一部だけであった。これは IEEE や IETF で明確に要件として定義されていないた

めと考えられるが、実運用にあたっては、各標準で定義されるような単に認証に必要な機能だけでなく、このような付加機能が選定のポイントになると思われる。

（イ）　自前で認証局やリポジトリを構築する場合

証明書発行に必要な認証局やリポジトリなどの環境を自前で用意するにあたって、上述の認証局ベンダから発行される証明書と同等の厳密さを要求することは、現実的でない。同等の厳密さを確保するには、認証局ベンダと同等の運用が必要であり、電子署名法の例に挙げたように、相当な運用コストが発生するからである。

逆に証明書の厳密さを求めないのであれば、自前で PKI 環境を構築することには、それなりのメリットがある。一つには証明書の管理が挙げられる。例えば認証局ベンダから証明書を取得していた場合、証明書に対応する鍵ペアを紛失したりした場合、再発行や失効には認証局ベンダとのやりとりが必要となる。場合によっては、再発行や失効に数日かかる可能性もある。しかし、これらの証明書発行環境を自前で運用していれば、再発行や失効なども容易に行うことができる。勿論これらの運用にはコストが発生してしまうが、運用に厳密さを要求しなければ、ある程度は低減させることが可能と考えられる。

もっとも、クライアント証明書に厳密さを求めないようであれば、EAP-TLS よりも EAP-TTLS や PEAP のように、そもそもクライアント証明書を用いない認証手順を採用した方が適切である。その場合には、必要となるのはサーバ証明書だけとなるので、少数であれば認証局ベンダなどから取得してしまう方が現実的であるかも知れない。

[島岡]

8.2　鍵生成の主体について

Broadcast Key またはグローバル認証キーと呼ばれる WEP キーは、基本的にアクセスポイントがランダムに生成する場合が殆どで、製品によっては管理者が静的に設定することもできた。実験で使用した製品の中にはなかったが、鍵を自動的に更新できないものも存在するかもしれない。Broadcast Key はその使用目的上、さほど高いセキュリティを要求されるものではないが、やはり自動的に更新されることが望ましい。

一方、Unicast Key または Key-mapping Key と呼ばれる WEP キーの生成方法には

a. TLS でネゴシエートした鍵マテリアルから生成する
b. アクセスポイントがランダムに生成する

という全く異なるアプローチによる 2 つの選択肢がある。b の場合は a の鍵でそれを暗号化した実体を EAPOL-Key フレームで送信することになる。IEEE Std 802.1X-2001 と関連文書に準拠しても、その選択は最終的には実装に委ねられており、どちらがセキュリティ的に堅牢であるか等については意見の分かれるところである。

この Unicast Key の生成方法に着目すると、実験で使用したアクセスポイントは大きくこの 2 つのグループに分けることができた。以下、それぞれのグループで実際にどのような特徴を持っていたか分析結果を述べる。

8.2.1　TLS の結果から Unicast Key を生成するタイプ

TLS でネゴシエートした鍵マテリアルから Unicast Key を導き出すアクセスポイントは AP1 と AP2 であった。

WEP キーの内容

いずれのアクセスポイントも、TLS でネゴシエートした鍵マテリアルから導き出されるものを Unicast Key として使用するため、Unicast Key は完全にステーション毎の鍵である。そして、EAPOL-Key フレームで Key Index と Key Length だけを指定するだけでよく、WEP キーの実体を配布する必要はない。

なお、実験中 AP2 は Broadcast Key を自動的に更新することがなく、その設定方法を見つけ出すこともできなかった。もし仕様上、何らかのタイミングで一度生成した鍵を、以後長期間に亘り使い続けるのであるならば、問題である。

WEP キーの配布形態

WEP キーの配布時期、Key Index、配布順序等をまとめると、以下のようになる。

AP1：

Unicast Key の配布時期	1X 認証完了直後
Unicast Key の Index（0～）	3
Broadcast Key の配布時期	1X 認証完了直後と更新時
Broadcast Key の Index（0～）	鍵を自動的に更新する場合は 1, 0, 1, 0, ... と変化 鍵を自動的に更新しない場合は管理者が選択
配布順序（() 内は Index）	Broadcast Key を自動的に更新する場合は 1X 認証完了後に Broadcast(1), Unicast(3) 以後 Broadcast Key の更新の度に Broadcast(0) ↓ Broadcast(1) ↓ Broadcast(0) ↓ を繰り返す Broadcast Key を自動的に更新しない場合は 1X 認証完了後に Broadcast(n), Unicast(3)

Broadcast Key を自動的に更新する場合は、各ステーションは最終的に１つの Per-station unicast key と２つのグローバル認証キーを持つことになる。

AP2：

Unicast Key の配布時期	1X 認証完了直後
Unicast Key の Index（0～）	2
Broadcast Key の配布時期	1X 認証完了直後
Broadcast Key の Index（0～）	0
配布順序（() 内は Index）	Broadcast(0), Unicast(2)

特にこれといった特徴はなく、オーソドックスな配布形態である。

8.2.2 アクセスポイントが Unicast Key を生成するタイプ

Unicast Key を自ら生成するアクセスポイントは AP4 と参考 AP1 であった。

WEP キーの内容

いずれのアクセスポイントも TLS でネゴシエートした鍵マテリアルは、アクセスポイントがランダムに生成した Unicast Key を暗号化するためにだけ使用される。そして、暗号化した Unicast Key は、EAPOL-Key フレームで実際に送信される。基本的に鍵の更新と配布はタイマーによって実行され、それは 802.1X の認証のタイミングとは無関係である。

このアプローチ自体は賛否両論あるものの、Unicast Key がステーション毎の鍵でありさえすれば、セキュリティ強度に問題があるというものではない。そもそも IEEE Std 802.1X-2001 によると、Authenticator 内で認証を司るステートマシンと、鍵の配布を司るステートマシンが連動しなければならないという規定はない。

ただし、この２つのアクセスポイントの場合、問題は Unicast Key として配布される WEP キー内容が、全ステーションで共通のいわゆるグローバル認証キーになっているところにある。ステーション毎の鍵ではないため、ネットワーク内でステーションのプライバシーを守ることはもちろんできず、１つの鍵が破られるとネットワーク全体に被害が及ぶ可能性がある。とはいえ、鍵自体はランダムに生成されているようであるし、TLS で生成したステーション毎の鍵で暗号化して安全に配布されるため、イントラネット等に用途を限定すれば、セキュリティ強度的に必ずしも問題があるというわけではなさそうである。

ただ、奇しくもこの２つの製品は共に、無線デバイスとしてクライアント側と共通の PC カードを流用するという形態をとっており、何らかの因果関係があると思わざるを得ない。このタイプのアクセスポイントは既存の無線 LAN カード（PC カード）を利用することで、汎用的で拡張性に富む設計ができ、開発期間も短縮することができる。しかしその反面、何らかの理由でパフォーマンスが犠牲になる、あるいはそれを解決するには長い開発期間と高度な技術が必要になるのではないかと推測される。そこでセキュリティとのトレードオフにより、このような特徴を持つに至ったのではなかろうか。

なお、参考 AP1 は、設定によっては Broadcast Key を Unicast Key と同じ値にするか、異なる値にするかを選択することができるが、その仕様の意図は不明である。

WEP キーの配布形態

WEP キーの配布時期、Key Index、配布順序等をまとめると、以下のようになる。

AP4：

Unicast Key の配布時期	更新時と、Association に続けて 1X 認証を完了した直後
Unicast Key の Index（0～）	更新のたびに 1, 2, 3, 1, 2, 3, ... と変化する
Broadcast Key の配布時期	Unicast Key を配布するとき
Broadcast Key の Index（0～）	1, 2, 3 のうち Unicast Key の Index 以外
配布順序（() 内は Index）	Unicast(1), Broadcast(2), Broadcast(3) ↓ Broadcast(1), Unicast(2), Broadcast(3) ↓ Broadcast(1), Broadcast(2), Unicast(3) ↓ Unicast(1), Broadcast(2), Broadcast(3) というパターンでローテーション

Unicast Key を更新したときは、全ステーションに対して EAPOL-Key フレームを送信する。このとき、接続しているステーションの数に応じて時間的にずらしながら送信する。したがって、ある瞬間には全ステーションの約３分の１が、同じ Unicast Key をデフォルトキーとして共有していることになる。

また、更新されるのは３つの WEP キーのうち１つだけなので、Unicast Key 以外は Unicast Key の３倍の長さの周期で更新されることになる。

参考 AP1：

Unicast Key の配布時期	更新時と、1X 認証完了直後
Unicast Key の Index（0～）	0 または 2
Broadcast Key の配布時期	更新時と、1X 認証完了直後
Broadcast Key の Index（0～）	Unicast Key が 0 のときは 1 Unicast Key が 2 のときは 3
配布順序（() 内は Index）	Unicast(0), Broadcast(1), Unicast(0), Broadcast(1) または Unicast(2), Broadcast(3), Unicast(2), Broadcast(3)

Index を切り替える条件は不明である。
Unicast Key と Broadcast Key のセットを 2 回送信する意図も不明である。

8.2.3 その他のタイプ

参考 AP2 は、参考までに接続性の検証だけを行ったアクセスポイントで、次のような特徴を持っていた。

- Unicast Key として、静的な全ステーションで共通の鍵を使用する。
- 802.1X 認証後、Broadcast Key だけを配布する。

このアクセスポイントの動作は実装上の問題か、仕様であるのか調査できていない。

[中島、関]

8.3 認証のポリシ(AuthenticationServer の動作と AP の反応)

8.3.1 Session-Timeout 属性の取り扱い

4.3 節で示した通り 802.1X における Session-Timeout 属性には、WEP 鍵再配布のための再認証を行うタイマー値としての意味がある。再認証の場合、アクセスポイントとの接続は保たれたままで認証が行われるため、ストリーミング処理など行っている最中でも、認証処理中による一瞬の途切れを除けば、連続して通信を行うことができる。

しかし一部のアクセスポイントにおいて、従来通り RFC2865 に記述された意味合いでしか Session-Timeout 属性を扱わないものが存在した。そのようなアクセスポイントの場合、Session-Timeout 属性で指定された秒数経過すると、接続が切断されてしまうため、連続した通信を保つことができない。802.1X に対応した意味付けへの変更が望まれる。

8.3.2 アカウンティング処理について

実験に使用したアクセスポイントでは、AP2 以外のアクセスポイントには RADIUS Accounting の機能が備わっていることが確認できた。RADIUS Accounting の機能がある各アクセスポイントとも、Accounting パケットに含まれる RADIUS 属性は、通常ダイヤルアップ接続で使用される RADIUS 属性と遜色ないものを含んでいるため、通信データ量に従った課金を行うなどの用途に利用することは可能である。

しかし接続時間については、無線 LAN の特性上いつ切断されたか判別しづらく、実験においても、サプリカント側で切断を行ってから Accounting-Request(Stop)が発せられるまでに、相当時間(１時間程度)経過することも見られたので、接続時間を基にした課金を行うには注意が必要と思われる。

また AP3 においては、以下のような現象が見られた。

- 一連の認証手順後、RADIUS サーバが Access-Accept を返す
- しかしまた通信は行えない
- AP3 から Accounting-Request 送信。RADIUS サーバが Accounting-Response を返信
- 通信が行えるようになる

これは認証処理の一部としてアカウンティング処理を捕らえてしまっているためであり、AS5 など一部のアカウンティング処理機能のない RADIUS サーバと相互接続を行った際に問題となる動作である。

8.4 Fast Reconnect (Session Resumption)

実験で使用した Supplicant は、いずれも EAP-TLS においては Fast Reconnect (Session Resumption) をサポートしていなかったため、今回は、認証に要する時間がどの程度短縮されるかを検証することができなかった。

TLS の機能によって実現される Fast Reconnect では、クライアント証明書・サーバ証明書を必要としない。以前に確立した各種セッション情報（セキュリティパラメータ）が有効であるかどうかを端点どうしで確認し合うだけなので、認証と新しい WEP キー（および WEP キーを暗号化する鍵）の生成を高速に行うことができる。

802.1X の再認証中は、それまでの通信が継続できなければならないので、単一のアクセ

スポイントに接続している限り Fast Reconnect の高速性はさほど大きな恩恵とはならない。Fast Reconnect は、アクセスポイント間を移動する際に最もその力を発揮する。この場合、バックエンドでアクセスポイント間通信が行われるか否かに関わらず、セキュリティを保ちつつ、できる限り省略された認証プロトコルを実行しなければ、利便性を著しく欠くことになるからである。

Fast Reconnect には、実装上注意すべき点もある。例えば、取り外し可能なトークン等のデバイスを証明書の格納場所としているような場合、証明書が不要であることを理由にその存在を確認せずに Fast Reconnect を行ったり、証明書の使用前にデバイスからシステムの記憶装置に証明書をコピーしたまま放置するようなことは、避けなければならない。Fast Reconnect をサポートする場合は、プロトコル以外の部分でのセキュリティに対する配慮が必要となる。

8.5 AP 間のローミング

ローミングの理想は、アクセスポントと RADIUS サーバとの連携やアクセスポイント間通信によって、ステーションとアクセスポイントとの間の認証プロトコルを極力省略することであろう。しかし本実験では、ごく単純なローミングによって、アクセスポイントを移動したときの再認証の挙動を検証した。ローミングの際は、移動元のアクセスポイントの電源を切ることにより、アクセスポイント間通信の実行を確実に禁止し、残された省略プロトコルが EAP-TLS の Fast Reconnect だけとなるようにした。

その結果、アクセスポイントは、ステーションからの Reassociation Request に応答した後、EAPOL よって EAP-Request/Identity パケットを送信して EAP を開始するが、実験で使用した Supplicant はいずれも Fast Reconnect をサポートしないため、通常の認証が行われただけであった。

実験結果を見ると、ローミングができなかった機器の組み合わせがある。これは、実験で使用した全てのアクセスポイントは 802.1X に対応しているものの、802.1X の使用を前提にしたときの 802.11 の Authentication および Association に必要なパラメータ（認証アルゴリズム、WEP の有効性 等）はアクセスポイントによって異なるためである。Supplicant SU1 は 802.11 のレイヤまでを制御し、この差異を吸収するが、Supplicant SU2 は 802.11 のレイヤを無線 LAN カードドライバや無線 LAN クライアントツールの能力に委ねており自分では制御しないため、移動先のアクセスポイントに接続できない（Association ができない）という事態が起こる。

Supplicant ソフトウェアというものは本来、制御される「ポート」を有するデバイスの

種類に左右されるべきではないが、実際に実装しようとすると、デバイスドライバそしてシステムとの間に密接な関わりがあるため、互いに協調が必要になるということが言える。

[納村]

8.6　鍵変更時の通信の安定性

WEP キー変更の際の、通信の安定性に関する実験結果を見ると、再認証が開始すると、ほとんどの場合 ICMP ECHO Reply のいくつかが、STA 側でタイムアウトとなることが分かる。しかし、残念ながら WEP キーの配信方法に関わるような興味深い原因を示すデータを収集するには至らなかった。

例えば WEP キーの更新時に、EAPOL-Key フレームには同じレイヤでのアクノリッジが規定されていないため、アクセスポイントは EAPOL-Key フレームを送信した直後から新たな WEP キーを使用するであろう。しかし受信する側は EAPOL-Key を処理している間に次のデータを受信するかもしれない。このような場合にどう対処しているかといったことは観察できなかった。

いくつか判明した原因として、

- ECHO Request の受け取り先である NServer が、認証完了後の DHCP のリクエストや DNS のクエリー等の処理中に、有線上に流れている ECHO Request を取りこぼしていた
- RADIUS プロトコルの開始直前に有線上に流れてきた ECHO Request がルータに捨てられていた（このときは STA からの Ping はルータに対して発せられていた）

等が判明したという程度である。恐らくその他に、ECHO Reply を受信したが単に Ping ツールのシビアな設定のためにタイムアウトとなったケースもあると考えられる。

[中島]

9　WPA と IEEE802.11i

IEEE ではデータ保護、ユーザ認証機能に問題が多い 802.11 のセキュリティ機能を強化した RSN(Robust Security Network)という規格の検討を TGi で行っている。802.11i は 2002 年 11 月に draft3 が発表されたが、その後もいくつかの機能の見直しや VirtualAP など新しい機能の検討が進んでいて 2003 年末は標準化される予定である。一方無線 LAN 機器業界団体である Wi-Fi Alliance では 2002 年 10 月に IEEE802.11i-draft3 で決まった機能の中で、現在の無線 LAN 機器の hardware でもサポートできる機能や市場的に要求が高い機能を中心に実装する Wi-Fi Protected Access(WPA)規格を発表した。本章では WPA での対応機能を中心に RSN での新しい機能について説明する。

9.1　WPA でサポートする 802.11i の主な機能

WPA は IEEE802.11i-draft3 の subset で実際 WPA の仕様書の大半が 802.11i の draft3 を参照している。

9.1.1 802.11,WPA,802.11i 機材の混在運用機能

WPA は 802.11i 規格との上位交換性を持ちながら、既存の 802.11 機材との混在運用ができるようにするため AP と Supplicant では下記の設定項目を持つように指定している。

01. WPA、WEP、802.1X の EAPOL-key を利用した rekey WEP(AP のみ）での一つかそれ以上の Associate 方法
02. WPA モードでの有効な Unicast cipher list として TKIP と AES
03. WPA モードでの Pre-shared key 入力書式は ASCII 文字列と 256bit key
04. WEP モードでの 40 か 104bit の static WEP key

Beacon と Association において RSN IE(RSN Information Element : 上記設定に対応する認証機能やデータ暗号方法のリスト)を通知することによって、AP と Supplicant の認証方法とデータ保護の方法についてネゴシエーションが行われる。

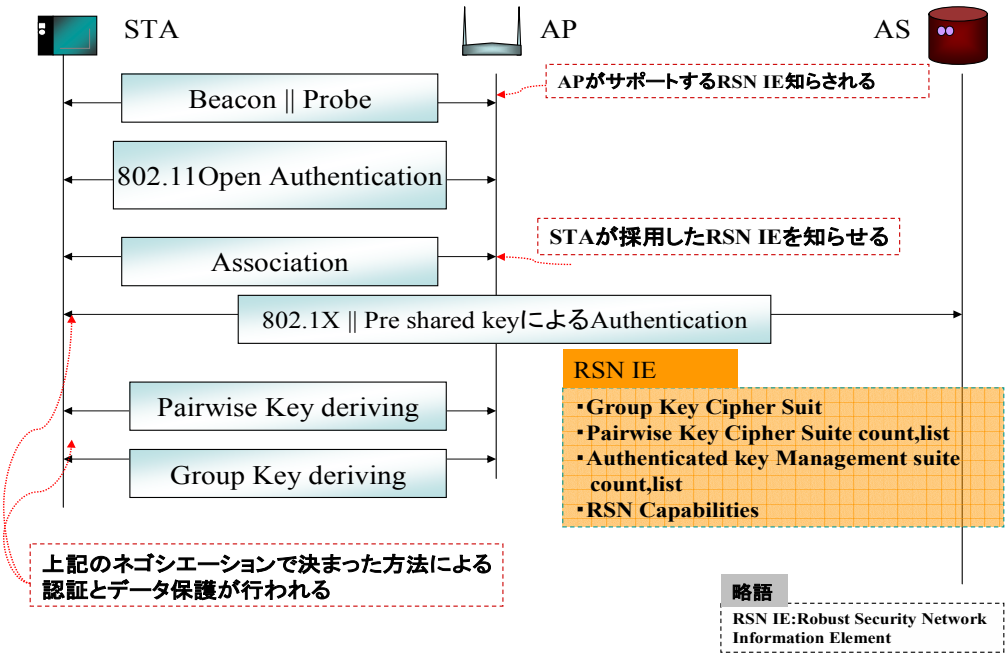

図 9-1　RSN IE の交換

9.1.2　認証機能

802.11i,WPA では 802.1X と static に決められたパスワードによる Pre Shared Key を利用して認証機能を提供する。

802.1X は EAP 対応の RADIUS を追加する必要があるがユーザの集中管理を行うことができるため、多くのユーザを管理する必要がある企業や ISP 事業者には有効である。

一方 Pre Shared Key による認証は 802.1X のような高度な認証機能やユーザの集中管理ができないが RADIUS サーバの追加導入が必要ないため、多くのユーザを管理する必要がない SOHO や個人ユーザに有効である。

9.1.3 鍵管理機能提供

802.11i,WPA では認証の際使われた鍵を利用し、下記のようにデータ保護用の鍵管理を行う

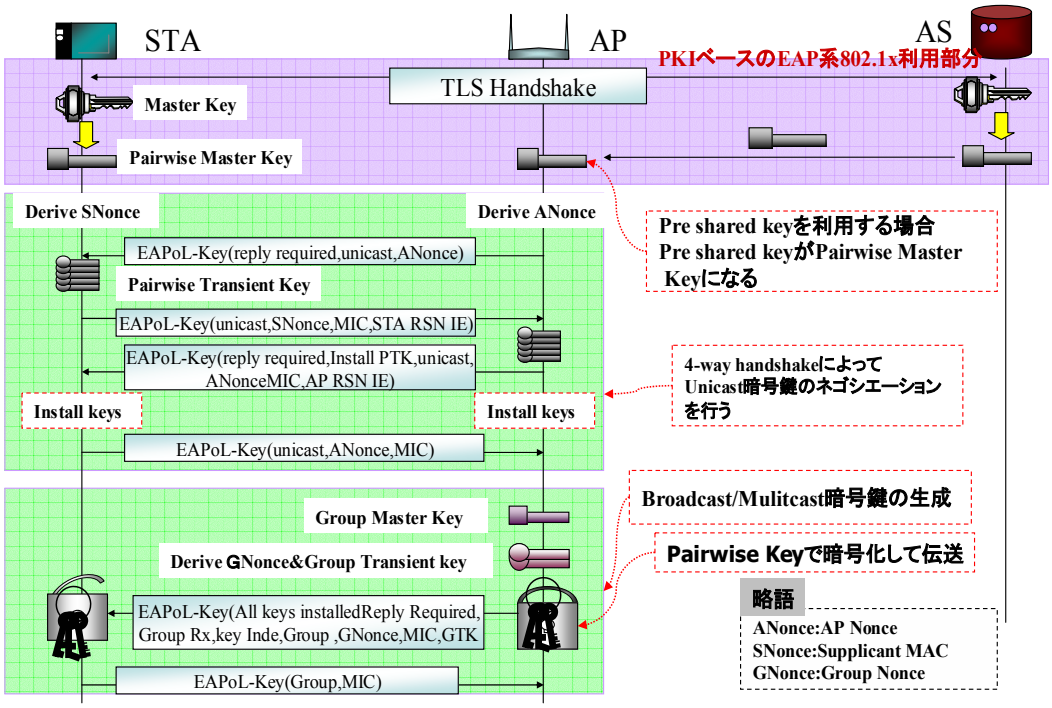

図 9-2　802.11i,WPA の鍵管理機能

9.1.4 データ保護機能

■データ暗号化

WPA の設計目標の一つは Wi-Fi 規格のハードであれば Software の upgrade のみで対応させることである。そのため、802.11i では必須としている AES によるデータ暗号の代わりに 802.11i では任意としている Temporal Key Integrity Protocol(TKIP)を必須とし、AES を任意とした。

WPA での TKIP によるデータ暗号化は下記のように行われる

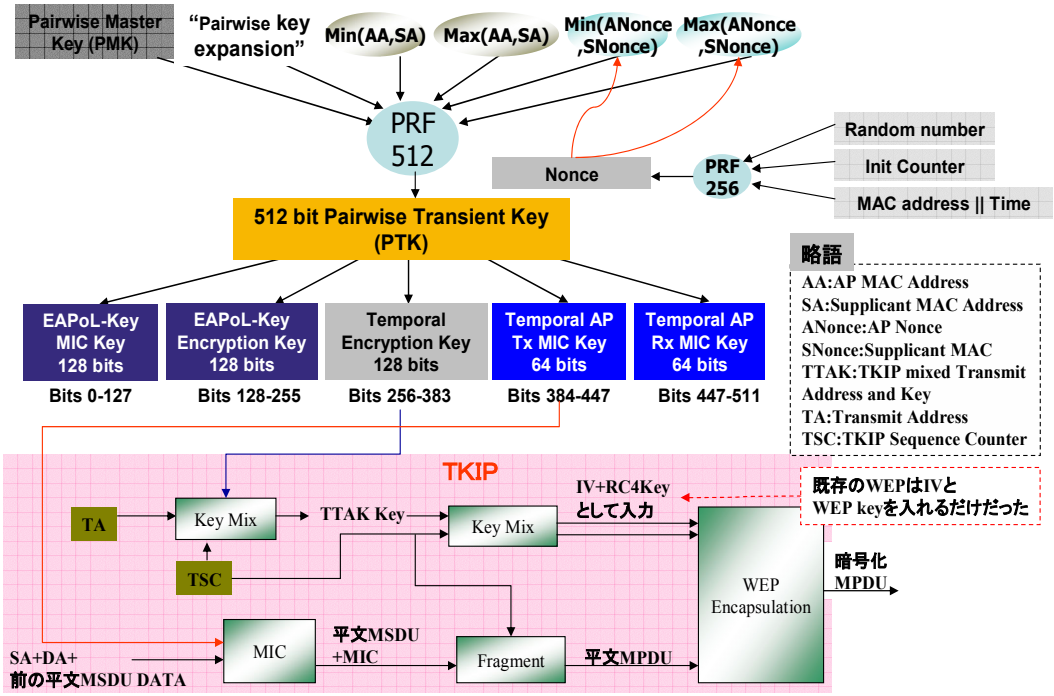

図 9-3　WPA での TKIP によるデータ暗号化(Unicast 用)

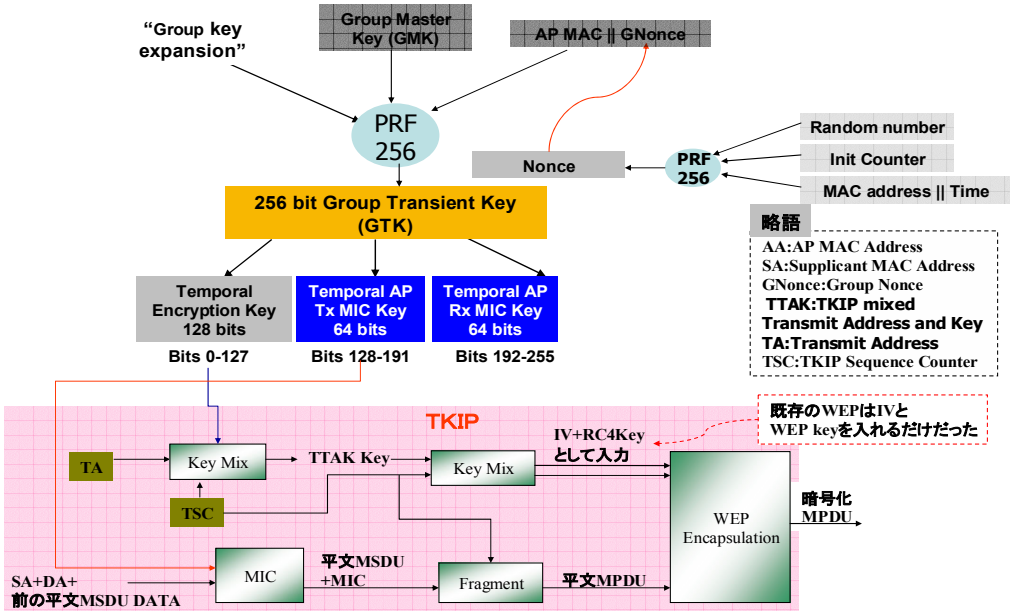

図 9-4　WPA での TKIP によるデータ暗号化(Broadcast/Multicast 用)

■データ改ざん防止

TKIP ではデータ改ざん防止に CRC32 の代わりとして下記の Message Integrity Check for TKIP(Michael)を利用している

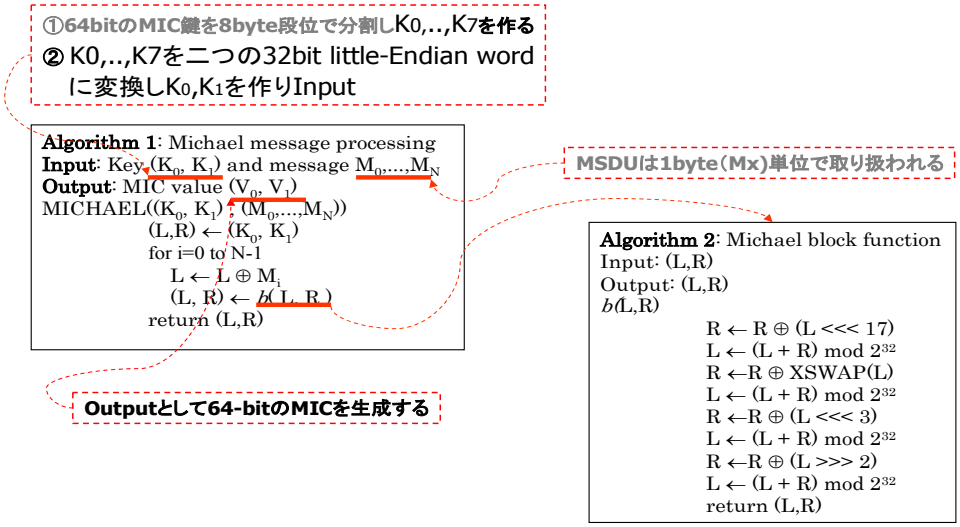

図 9-5　TKIP(Michael)

■MIC エラー監視と暗号鍵 Update

WPA では Bit-flipping 、Fragmentation 、Iterative　guessing attack などの積極的な攻撃を防ぐのを目的として AP、Supplicant では下記の MIC エラー監視と暗号鍵 Update 機能を持つ

- AP
  - ➢ Broadcast/Multicast 暗号鍵保護
    - ✧ Supplicant から EAPOL－Key Request により鍵の変更要求があった場合

    Broadcast/Multicast 暗号鍵を破棄し鍵の変更を行う
    - ✧ MIC failure が起こってから 60 秒以内に再度発生したらすべての station の GTK を update する
    - ✧ Log に MIC failure を残す
  - ➢ Unicast 鍵保護

✧ Supplicant から EAPOL－Key Request により鍵の変更要求があった場合

その Supplicant に対応する Unicast 暗号鍵を破棄し鍵の変更を行う

- ✧ Unicast 鍵を削除するか変更するまで 802.1X メッセージ以外の受信したすべてのメッセージを破棄する
- ✧ MIC failure が起こってから 60 秒以内に再度発生したら該当の station の unicast 暗号鍵を 4-way handshake によって初期化する

- ✧ Log に MIC failure を残す

- Supplicant
  - Broadcast/Multicast 暗号鍵保護
    - ✧ 要求があれば Broadcast/Multicast 用暗号鍵を破棄する
    - ✧ EAPoL-key を利用して AP に Broadcast/Multicast 暗号鍵の update を要求する
    - ✧ Log に MICfailure を残す
  - Unicast 暗号鍵保護
    - ✧ Unicast 用暗号鍵を削除するか変えるまで 802.1X メッセージ以外の受信したすべてメッセージを落とす
    - ✧ EAPoL-key を利用して AP に Unicast 用暗号鍵の update を要求する
    - ✧ Log に MICfailure を残す

9.2 WPA でサポートしない 802.11i の主な機能

WPA では 802.11i の機能のうち、以下のようなものはサポートしない。

- 将来変更の可能性がある機能
- 現在マーケットでは需要が少ない機能
- 新たに hardware の update が必要となる機能

具体的には以下の機能がサポートされていない。

① Secure IBSS
アドホックモードでの 802.1X による peer 認証機能の強化

② Secure fast handoff
Station が AP 間を移動する際、認証手順を簡略化することにより認証による遅延を減らす機能

③ Secure Reassociation, Disassociation, Beacon
Reassociation, Disassociation, Beacon message の暗号化を行う機能

④ AES（hardware update が必要）
CCMP(Counter-Mode/CBC-MAC Protocol) :AES-CTR＆CBC-MAC によるデータ暗号化。802.11i では CCMP が必須になっている。WRAP(Wireless Robust Authenticated Protocol):AES-OCB によるデータ暗号化

⑤ Virtual AP
2003 年 3 月から議論がはじまっている機能で一つ。AP が複数の Virtual AP として振舞うことにより公共無線 LAN サービスで一つの AP で複数の ISP が同時利用可能になる

[任]

10　PKIX における無線 LAN の動向

現在 IETF/PKIX では、PPP と無線 LAN に対する証明書拡張と属性の追加について、Internet-Draft が投稿されている。本章では、この I-D について簡単に解説する。この I-D では、EAP 用の extKeyUsage 拡張と、無線 LAN に用いる SSID に関する証明書拡張を定義している。

10.1　EAP extKeyUsage Value

5.2.5 で述べたように RFC3280 では、extKeyUsage について任意の用途を追加定義できる、としている。そこでこの I-D では EAP 認証を目的とした extKeyUsage の値を 2 つ定義している。

eapOverPPP

この値を extKeyUsage 拡張に含んだ証明書は、EAPoverPPP 認証に用いることができる。

eapOverLAN

この値を extKeyUsage 拡張に含んだ証明書は、EAPoverLAN 認証に用いることができる。

前述の通り、証明書がこれらの値を含んでいないからと言って、EAPoverPPP 認証や EAPoverLAN 認証に利用できないわけではない。これらを新たに定義した背景には、従来の clientAuth だけでは広義になってしまい、認証用途を特定できなくなってしまうことへの懸念があったと考えられる。

例えば、従来のような clientAuth を明記せずに、この eapOverPPP や eapOverLAN のみを extKeyUsage 拡張に明記したクライアント証明書とすることで、Web のクライアント認証には利用できずに、EAP 認証にのみ利用可能なクライアント証明書を発行することが可能になると考えられる。

10.2　wlanSSID 拡張

これも authorityInfoAccess 拡張と同様、X.509 にはなく PKIX が独自に定めるインターネット拡張である。この拡張には、複数の SSID を記述することができる。この拡張を含んだクライアント証明書を発行することで、その証明書を用いて接続可能な無線 LAN を制限することができると考えられる。

また、サーバ証明書に含めることで、その RADIUS サーバが認証可能な無線 LAN を制限することも可能と考えられる。これは、RADIUS サーバ用証明書が、不正に他の無線 LAN を認証することがないようにする防止策と考えことができる。

10.3　wlanSSID 拡張(属性証明書)

X.509 では、公開鍵証明書とは別に属性証明書というものが定義されている。これは、公

開鍵証明書と連携して機能するものだが、特に認証用途において重要なものである。

属性証明書を利用する場合には、公開鍵証明書にはなるべく不変な情報のみを記載し、認証に必要な権限などの可変情報を属性証明書に記載する。この属性証明書は、極めて短い期間で再発行することも可能であり、またエンティティに発行した公開鍵証明書を更新しないまま、エンティティの権限情報を変更可能なため、認証には非常に有効な技術である。

前述の wlanSSID 拡張も、接続先無線 LAN を制限するようなアクセス制御情報であるので、本 I-D では、これを属性証明書でも記載できるよう定義している。これにより、クライアントに配布した公開鍵証明書を変更することなく、アクセス可能な無線 LAN の SSID を記述した属性証明書のみを更新することで、クライアントをアクセス制御することが実現可能と考えられる。

[島岡]

Appendix A EAP-TTLS と PEAP の実装状況

802.1X機能	製品	製造元	対応EAP-type						
			EAPMD5	LEAP	EAPTLS	EAPTTLS	EAPSIM	PEAP	EAP-SPEKE (Simple Password Exponential Key Exchange)
RADIUSサーバ	Odyssey Server	Funk software	○	○	○	○			
	Steel-Belted Radius	Funk software	○	○	○	○		○	
	ステラクラフト Enterpras	ステラクラフト	○		○	○		○	
	Cisco ACS	Cisco Systems	○	○	○		○	○	
	.NET server	Microsoft	○		○			○	
	fullflex wireless	アクセンス・テクノロジー	○		○			○	
	AEGIS Server	Meeting house	○	○	○	○		○	
	Secure.XS	INTERLINK NETWORK	○	○	○	○		○	○
	FreeRADIUS	FreeRADIUS Project	○		○				
	Radiator Radius	Open System Consulta	○		○	○		○	
	NavisRadius	Lucent Technologies	○	○	○	○	○	○	
Supplicant	Odyssey client	Funk software	○	○	○	○			
	AEGIS Clinet	Meeting house	○	○	○	○		○	
	WindowXP	Microsoft	○		○			○	
	ACUS	Cisco		○				○	
	MPWorks	UDTech	○		○	○		○	
	Alfa & Ariss	SecureW2				○			
	XSupplicant	Freeware	○		○	○			

2003 年 4 月調査

近日対応予定も含む

アクセスポイントの対応

製品名	メーカ	802.1X 対応	WPA 対応予定
Aironet シリーズ	Cisco Systems	○	○
Intel PRO Wireless AP	intel	○	-
ORiNOCO,AP2500	プロキシム	○	-
ORiNOCO AP1000、AP2000,AP600	プロキシム	○	○
Air Station	メルコ	○	不明
CN3000	Colubris Networks	○	○
RoamAbout R2	Enterasys	○	○
AVAYA Wireless AP-3	AVAYA	○	○
WL-1154	NTT 東日本	○	不明

2003 年 4 月調査

近日対応予定も含む

Appendix B 参考文献

[1]N.Borisov et al, ``Intercepting Mobile Communications: The Insecurity of 802.11",MobiCom2001, July 2001

[2]S.Fluhrer et al, ``Weakness in the Key Scheduling Algorithme of RC4", 8th Annual Workshop on Selected Areas of Cryptography, August 2001

[3]RFC2284
"PPP Extensible Authentication Protocol (EAP)"

[4]RFC2246
"The TLS Protocol Version 1.0",

[5] IEEE802.1X
"Port-Based Network Access control"

[6]draft-ietf-pppext-eap-ttls-02 － EAP Tunneled TLS Authentication Protocol
Funk Software 社 － http://www.funk.com/

[7] draft-josefsson-pppext-eap-tls-eap-05.txt "Protected EAP Protocol (PEAP)"

[8] "Protected EAP" http://www.ietf.org/proceedings/02mar/slides/eap-3/sld008.htm

[9]RFC2865 - Remote Authentication Dial In User Service (RADIUS)

[10]RFC2866 - RADIUS Accounting

[11]RFC2869 - RADIUS Extensions

[12]draft-congdon-radius-8021x - IEEE 802.1X RADIUS Usage Guidelines

[13]P802.11i
Draft Supplement to Standard for
Telecommunications and Information Exchange Between Systems LAN/MAN Specific Requirements Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications:Specification for Enhanced Security

[14]Wi-Fi Protected Access (WPA) Version1.2

[15]ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8:
INFORMATION TECHNOLOGY - OPEN SYSTEMS INTERCONNECTION - THE DIRECTORY: PUBLIC-KEY AND ATTRIBUTE CERTIFICATE FRAMEWORKS

[16]RFC3280
Internet X.509 Public Key Infrastructure Certificate and CRL Profile

[17]RFC2560
X.509 Internet Public Key Infrastructure
Online Certificate Status Protocol - OCSP

[18]RFC2818
HTTP Over TLS

[19]RFC2632
S/MIME Version 3 Certificate Handling

Appendix C 相互接続実験作業参加者

株式会社アクセンス・テクノロジー
　納村　康司、中川　孝志、吉田　伊津子
株式会社インターネット総合研究所
　任(Im)　俊学、佐藤　めぐみ、佐藤　友治、法林　浩之
SSH コミュニケーションズ・セキュリティ株式会社
　古川　徹
オムロンフィールドエンジニアリング株式会社
　粕谷　宗史
新日鉄ソリューションズ株式会社
　寺島　弘明
セコムトラストネット株式会社
　島岡　政基、門田　剛
セコム山陰株式会社
　平本　耕造
株式会社ディアイティ
　関　義和
株式会社東芝
　渋谷　尚久
富士通エフ･アイ･ピー株式会社
　ニンスワンコシット　パンシット(Punsit)
松下電工株式会社
　岸本　英治
ユーディテック・ジャパン株式会社
　中島　聡

(50 音順)

工学院大学
　橋本　明、松木　和彦

○アドバイス他(以下敬称略)
シスコシステムズ
　上原子　茂樹
ネットマークス
　夏目　雅好
東日本電信電話株式会社
　森山　浩幹
マイクロソフト
　及川　卓也、古川　勝也

(50 音順)

Appendix D 機材および各種の御協力

機材協力
株式会社アクセンス・テクノロジー
RADIUS サーバ(fullflex wireless)
株式会社インターネット総合研究所
AP(Cisco Aironet350)
シスコシステムズ株式会社
AP(Aironet1200)、無線 LAN カード（Aironet350)、
RADIUS サーバ(Cisco ACS)、ルータ(Cisco7000)
セコムトラストネット株式会社
AP(Intel)、認証局(NEC CertWorker)
株式会社ディアイティ
ワイヤレスパケットキャプチャ(Airo Peek NX)、
AP(ORiNOCO AP-2000)
マイクロソフト株式会社
OS(Windows 2000 Server)、OS および RADIUS(Windows Server 2003 RC2)
プロキシム株式会社
AP(Harmony Access Point 802.11a)
株式会社東芝
AP(Avaya Wireless Access Point II)
ユーディテック・ジャパン株式会社
サプリカント(MPWorks)
株式会社メルコ
AP(Air Station)
(50 音順)

そのほかの協力
工学院大学