社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－DPS－130 (33)
2007－CSEC－ 36 (33)
2007／3／1

Protocol for carrying Authentication for Network Access (PANA) を利用したネットワークアクセス認証システムの実装と検証

海沼 義彦 [†1] 寺岡 文男 [†1]

[†1] 慶應義塾大学大学院理工学研究科

Protocol for carrying Authentication for Network Access (PANA) は，トランスポート層で認証情報を運ぶことができ，ネットワークアクセスを提供する環境に依存せずにネットワークアクセス認証を行えるプロトコルである．PANA は標準化中であることから，API の定義がされておらず実装例も少ない．本研究では，PANA の API の仕様を定め，その API を利用して PANA のモジュールを実装した．また (株) 東芝研究開発センターが独自に実装した PANA との相互接続試験を行い，本研究で仕様を定めた API の有効性を検証した．

Implementation and Examination of Authentication System for Network Access using PANA

Yoshihiko Kainuma [†1] Fumio Teraoka [†1]

[†1] Graduate School of Science and Technology, Keio University.

Protocol for carrying Authentication for Network Access (PANA) can carry user's information over transport layer, it provides environment-independent authentication for network access. Because PANA is in standardization now, there is no API definition and there are few implementations. In my work, I defined PANA API and implemented PANA modules using it. And I verified practical effectiveness of API definition through interoperability test between my implementation and Toshiba's implementation.

1 はじめに

現在，インターネットは広く普及し，誰もが気軽に利用できるようになった．その一方で，インターネット上で提供されるサービスの不正利用やサービスを提供するサーバを攻撃するといった不正行為も増加している．そういった不正行為を防ぐためにはユーザの認証・権限委譲が不可欠とされている．ユーザの認証・権限委譲はインターネット上のすべてのサービスに行われるべきであるが，それらのサービスを利用するために必要なネットワークアクセスに対する認証・権限委譲は特に厳密に行われなくてはならない．

また近年，無線技術の発展や機器の小型化，さらに Mobility Support for IPv4/v6 (MIPv4/v6) の標準化などにより，どこからでも簡単にインターネットに接続できる環境が整いつつある．それに伴いユーザは決まった場所からではなく様々な場所からインターネットに接続する機会が増えてくる．そういった環境でも可用性を失うことなくサービスを提供するためには，ネットワークアクセスを提供する場所に関係なく認証・権限委譲を行う必要となる．そのためには，ユーザと認証・権限委譲を行うプロトコル (認証プロトコル) との間のインタフェースがネットワークアクセスを提供する環境に依存していてはならない．そこで Internet Engineering Task Force (IETF) により標準化中である Protocol for carrying Authentication for Network Access (PANA)[1] がネットワークアクセスを提供する環境に依存しないユーザと認証プロトコルとの間のインタフェースを提供するプロトコルとして注目されている．

PANA は標準化中のプロトコルであることから，Application Program Interface (API) の仕様が定まっておらず，実装例はほとんどない．そこで，本研究では PANA の API の仕様を定め，さらにその API を利用して PANA のモジュールを実装する．また，実装したモジュールをまた，実装したモジュールを異なる実装と相互接続し，相互接続性を確認することで本研究において仕様を定めた API の有効性を検証する．

2 ネットワークアクセス認証システムの要素プロトコル

まず本研究で利用するネットワークアクセス認証システムの要素プロトコルについて説明する．

2.1 Protocol for carrying Authentication for Network Access (PANA)

Protocol for carrying Authentication for Network Access (PANA) は，認証情報をトランスポート層で転送するプロトコルである．PANA では認証される側を PANA Client (PaC)，認証する側を PANA Authentication Agent (PAA) と定義しており，以下ではこれらの用語を利用して PANA について説明する．

2.1.1 PANA Session

PANA では，PaC と PAA との間に PANA Session というセッションを確立して認証を行う．PANA Session では，PANA メッセージの転送に必要な情報と，PANA の安全性を保証するための情報の 2 種類の情報を管理している．前者はメッセージのシーケンス番号，最後に送受信した PANA メッセージ，再送間隔といった情報で，送受信したメッセージの検証やメッセージの再送に利用される．後者は PANA メッセージの Message Authentication Code (MAC) 生成に利用

される鍵 (PANA_AUTH_KEY) や，PANA_AUTH_KEY や MAC を生成するためのアルゴリズムといった情報で，これらの情報により認証後に交換される PANA メッセージのメッセージ認証が行われる．

2.1.2 PANA による認証の流れ

PANA による認証は以下に挙げる 5 つのフェーズを通して行われる．このフェーズを通して PANA による認証が行われる．

- Handshake Phase: PaC と PAA との間で PANA Session を開始するフェーズ．
- Authentication and Authorization Phase: ユーザから受け取った情報を利用して認証・権限委譲を行うフェーズ．
- Access Phase: 認証が成功し，インターネットへのアクセスを認められたフェーズ．
- Re-Authentication Phase: PANA Session の期限が切れる前に再認証を行い PANA Session を更新するフェーズ．
- Termination Phase: PANA Session を終了するフェーズ．

PANA による認証は Handshake Phase から始まる．PANA は PaC からも PAA からも開始することが可能であるが，PAA から開始する場合の Bootstrap の手法に関しては PANA の仕様では考えられていない．ここでは，PaC から開始する場合について説明する．まず PaC は通信に必要な情報を取得するために利用される Dynamic Host Configuration Protocol (DHCP) により PAA の IP アドレスを取得し，その IP アドレスに PaC が PANA-Client-Initiation (PCI) を送信する．PCI を受け取った PAA は PANA-Start-Request (PSR) を PaC に送り PANA Session の開始を伝える．PaC は PSR の返事として PANA-Start-Answer (PSA) を PAA に送ることで Handshake Phase が終了する．

Authentication and Authorization Phase は PSA を受け取った PAA が PANA-Auth-Request (PAR) を PaC に送るところから始まる．PaC は PAR の返事として PANA-Auth-Answer (PAN) を PAA に送る．ここで交換される PAR/PAN には認証情報が含まれており，PaC の認証・権限委譲が完了するまで PAR/PAN の交換が行われる．認証・権限委譲が終了すると，PAA は PANA-Bind-Request (PBR) を PaC に送信することで認証の結果を PaC に伝え，Authentication and Authorization Phase が終了する．認証に成功した場合は，この時点で PaC-PAA 間で PANA_AUTH_KEY が共有され，以降に交換する PANA メッセージには MAC が付加される．

Authentication and Authorization Phase で認証が成功すると Access Phase が始まり，PaC-PAA 間で定期的に PANA-Ping-Request/Answer (PPR/PPA) を交換する．このメッセージの交換により互いの到達性を確認する．また，PANA Session の期限が切れる前に PaC が PANA-Reauth-Request を PAA に送信し Re-Authentication Phase を開始する．PANA Session を終了する際には PANA-Termination-Request を送信し Termination Phase を開始する．PANA でのメッセージ交換の様子を図 1 に示す．

また，PANA にはどの Phase においてもエラーが検出されると PANA-Error-Request を送信し相手にエラーを通知する．このメッセージにはエラーの原因を示す値が含まれており，受信者はエラーの原因を取得することができる．

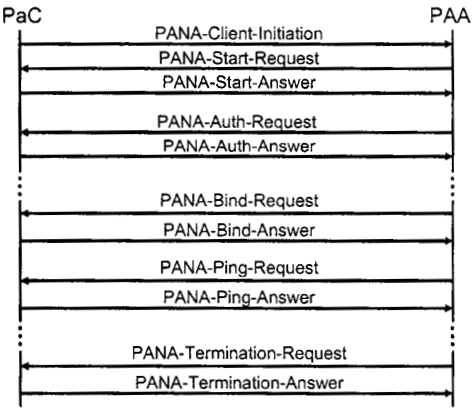

図 1: PANA でのメッセージ交換の様子

2.1.3 PANA の利点

PANA の大きな利点としては以下の 3 点が考えられる．

- 認証情報をトランスポート層で転送する．
- 認証エージェントとアクセスコントロールを行うモジュールが別々に定義されている．
- 容易に拡張が行える．

PANA はトランスポート層での認証を行うため，認証が下位層のメディア構成に依存せずに行える．さらに，アクセスポイントと認証モジュールを分離可能となり，アクセスポイント設置のコストを少なくすることができる．また認証エージェント (PAA) とアクセスコントロールを行うモジュール (Enforcement Point: EP) を分離することができるため，用途に適した柔軟なネットワーク構成が可能となる．そして PANA ではすべての情報を Attribute Value Pair (AVP) という形で運んでいるため，AVP を新たに定義することにより容易に機能拡張が行えるという利点もある．

2.2 Extensible Authentication Protocol (EAP)

Extensible Authentication Protocol (EAP)[2] は，様々な認証方式をサポートした認証フレームワークとして利用されている．PANA の中では EAP パケットが AVP として PAR/PAN に付加される形で利用される．EAP では認証される側を Peer，認証する側を Authenticator と呼ぶ．EAP では，以下に挙げる 4 種類のパケットが利用される．

- EAP Request: Authenticator から Peer へ認証に必要な情報を要求する．
- EAP Response: Peer から Authenticator へ認証に必要な情報を渡す．
- EAP Success: Authenticator から Peer へ認証が成功したことを伝える．
- EAP Failure: Authenticator から Peer へ認証が失敗したことを伝える．

EAP は，認証が成功・失敗するまで EAP Request/Response の交換を繰り返し，その結果を EAP Success/Failure で伝える．EAP Request/Response に含まれるデータと交換されるパケットの数は利用する認証方式により異なる．本研究では，パスワードベースの認証を行う EAP-MD5 と事前共有鍵 (Pre-Shared Key: PSK) による認証を行う EAP-PSK[3] を認証方式として採用する．EAP-PSK による認証手順を図 2 に示す．

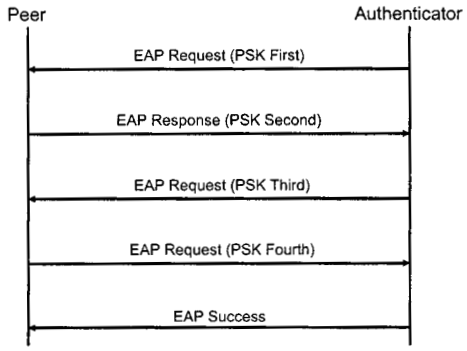

図 2: EAP-PSK による認証手順

EAP-PSK では，図 2 で示したように Peer-Authenticator 間で 4 つのパケットを交換する．全てのパケットが交換され認証が成功すると Master Session Key (MSK) が生成され，PANA Session に渡される．逆に EAP-MD5 では MSK が生成されず，PANA Session には何も渡されない．

2.3 Diameter base protocol (Diameter)

Diameter base protocol (Diameter)[4] は，認証・権限委譲にサービス利用情報収集を加えた AAA (Authentication, Authorization, Accounting) に必要な情報を Diameter メッセージとして適切なサーバまで転送するプロトコルである．本研究では，PAA が受け取った EAP パケットを適切な認証サーバまで転送するのに利用する．ここでは，Diameter メッセージを処理するサーバを AAAs，Diameter メッセージの処理を依頼するサーバを AAAc と定義する．また，Diameter メッセージの中継をするノードを AAA relay (AAAr) と呼ぶ．

Diameter は用途に合わせて様々なアプリケーションが定義されている．その 1 つが Diameter 上で EAP パケットを処理するために利用される Diameter EAP Application[5] である．Diameter EAP Application では，EAP Identity という方式を利用して AAAc がユーザ名を取得し，それを AAAs に転送する．その後は AAAs とユーザが AAAc を経由して EAP パケットのを交換し，認証が行われる．Diameter EAP Application では，Diameter-EAP-Request/Answer というメッセージが定義されており，AAAc-AAAs 間ではこのメッセージに EAP パケットを添付して EAP パケットを交換する．Diameter EAP Application による認証手順を図 3 に示す．

2.4 ネットワークアクセス認証システムの構成

本研究で構築するネットワークアクセス認証システムの全体像について説明する．本研究で構築するネットワークアクセス認証システムの構成図を図 4 に示す．流れとしてはユーザ端末上の PaC がアクセスルータ上の PAA に認証要求を出し (1)，認証要求を受け取ったアクセスルータは AAAc として AAAs に認証・権限委譲を依頼する (2)．AAAs は認証・権限委譲を行い (3)，その結果をアクセスルータを経由してユーザに伝える (4)．

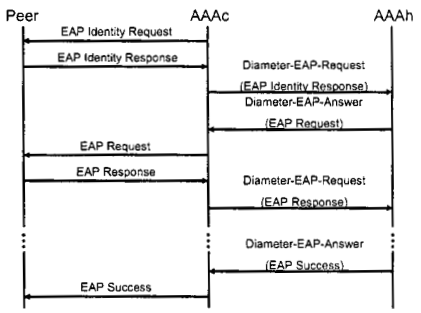

図 3: Diameter EAP Application による認証手順

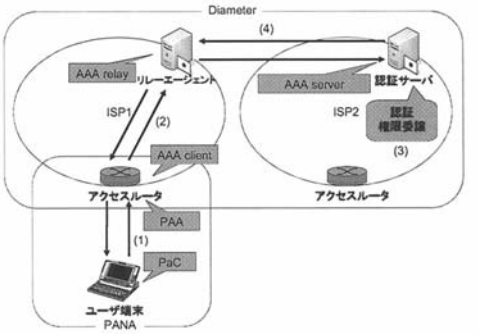

図 4: ネットワークアクセス認証システムの構成図

このシステムでは，PAA と AAAc は同一ノード上で動作する．EAP の認証によって生成された MSK を AAAs から AAAc に転送し，さらに同一ノード上で動作している PAA に渡すことで，PaC と PAA との間で PANA_AUTH_KEY の共有が可能となる．このシステム全体でのメッセージフローを図 5 に示す．

現在，ここで説明したシステムの要素プロトコルについてはそれぞれ Internet Engineering Task Force (IETF) により仕様が定められているが，プロトコル間のインタフェースなどは未定義であり，現在の仕様では 1 つのシステムとして動作させるには不十分である．さらに PANA や EAP-PSK は標準化中ということもありアプリケーションとして実装するための API も定められていない．そのため，このシステムを 1 つのシステムとして動作させた例はなく，動作検証もまったく行われていない．

3 既存のネットワークアクセス認証システム

ここでは，既存のネットワークアクセス認証システムである IEEE802.1X[6] について説明し，2 節で説明したネットワークアクセス認証システムとの性能比較を行う．

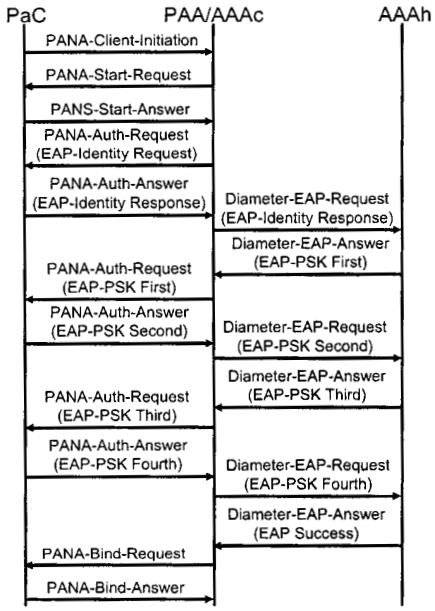

図 5: システム全体のメッセージフロー

3.1 IEEE802.1X による認証の流れ

IEEE802.1X は，Port-based access control という LAN ポートベースのアクセスコントロールを行うプロトコルである．認証要求を出す端末を Supplicant，Supplicant からの認証要求を受け付ける端末を Authenticator といい，Supplicant と Authenticator との間で EAP パケットの交換を行う．Supplicant から Authenticator に渡された EAP パケットはバックエンドの認証サーバに転送され，その処理結果が認証サーバから Authenticator を経由して Supplicant に渡される．Supplicant と Authenticator との間では EAP encupsulated Over LAN (EAPOL) というプロトコルを利用して行われる．EAPOL とは，データリンク層のフレームに EAP パケットをカプセル化することでデータリンク層での EAP パケットの交換を可能としたプロトコルで，データリンク層のメディアの種類ごとに異なる形で EAP パケットがカプセル化される．

3.2 IEEE802.1X の問題点

IEEE802.1X は，前述のようにデータリンク層で EAP パケットを交換するプロトコルである．そのため，データリンク層での認証に起因する問題点がいくつか存在する．この問題点を以下に挙げる．

- データリンク層より上位のプロトコルを利用したアクセスコントロールが行えない．
- 様々なデータリンク層の構成に対応できない．
- アクセスポイントが認証エージェントとして動作する．

IEEE802.1X は，LAN ポート単位でのアクセスコントロールには対応しているが，IP アドレスベースのアクセスコントロールのようなデータリンク層より上位のプロトコルを利用したアクセスコントロールを行うことができない．また，EAPOL ではデータリンク層のメディアごとにフレームを定義しているため，様々なデータリンク層の構成が混在するような環境には適用が難しい．さらに，IEEE802.1X ではデータリンク層での認証を行うという性質上，アクセスポイントが認証の処理を行う必要がある．しかし無線アクセスポイントが至るところに設置されどこからでも無線ネットワークアクセスが提供されるような環境を考えると，それぞれのアクセスポイントが認証の機能を持たなくてはならず，機器そのもののコストやそれらを設置するためのコストが大きくなる可能性がある．本研究で利用する PANA は，これらの問題がすべて解消されており，IEEE802.1X よりも優位性があると言える．

4 設計・実装

ここでは，ネットワークアクセス認証システムを構築した際に仕様を定義した PANA API の設計・実装について説明する．

4.1 アーキテクチャモデル

本研究で構築したネットワークアクセス認証システム全体のアーキテクチャモデルを図 6 に示す．ユーザの端末上では EAP Peer デーモンと PaC デーモンが，アクセスルータ上で PAA デーモンと AAAc デーモンが，認証サーバ上では AAAs デーモンと EAP Authenticator デーモンが動作する．PANA API はユーザの端末上で動作する PaC デーモンとアクセスルータ上で動く PAA デーモンに利用される．

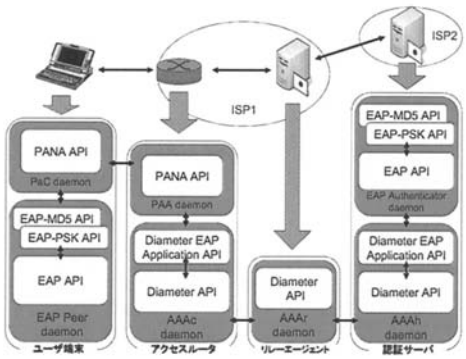

図 6: アーキテクチャモデル

4.2 設計指針

設計指針として，以下に挙げる 3 点に重点をおく．ここで挙げた 3 点を基本指針とし，設計・実装を行う．

- 動作環境に依存しない
- 拡張性を持たせる
- スケーラビリティを考慮する

動作環境に依存しない設計というのは IPv4/v6 といった利用するプロトコルによって挙動が変わることのないような設計である．また PANA のような標準化中のプロトコルは今後標準化の過程で仕様が変更する可能性もあるため，仕様の変更に柔軟に対応できるような，また用途に合わせて容易に拡張が行えるように拡張性を持たせる必要がある．拡張性を持たせるために，API として実装する部分をできるだけ基本的な処理に限定しなくてはならない．また複数のユーザが同時にシステムを利用した際にメッセージを一つ一つ処理する

ような設計ではその部分がボトルネックとなってしまい大きな遅延が発生する可能性がある．そのような事態を防ぐためにもマルチスレッド処理によりメッセージを並列処理する必要がある．また常時稼動することを考えメモリの管理も慎重に行う必要がある．

4.3 PANA APIの設計・実装

ここでは，PANA APIの設計・実装の詳細について示す．

4.3.1 コールバック

PANA APIでは，コールバックの登録によりメッセージ受信時やタイムアウト発生時の処理を指定することができる．メッセージ受信時に呼び出されるコールバックをメッセージコールバック，タイムアウト発生時に呼び出されるコールバックをタイマーコールバックと呼ぶ．

メッセージコールバックは，メッセージの種類ごとに登録することができる．同じメッセージでもRequestかAnswerかにより別々のコールバックを登録することもできる．メッセージコールバックとして登録したコールバックは，対応するメッセージを受信するとメッセージの検証を行い，正しいものであると判断された場合のみ受信したメッセージを引数として受け取った状態で呼び出される．メッセージの検証で正しいものでないと判断された場合はコールバックは呼び出されずPANA-Error-Requestが送信される．タイマーコールバックは，PANA Sessionのライフタイム，再認証，メッセージの再送，PANA-Ping-Requestの送信のそれぞれに対して登録可能で，対応するイベントのタイムアウト時にPANA Sessionの識別子を引数として呼び出される．

4.3.2 マルチスレッド

PANA APIは，4種類のスレッドを利用して処理が行われる．それぞれのスレッドについて表1に示す．

表 1: PANA APIのスレッド

スレッド名	役割
メインスレッド	PANA APIの本体となるスレッド．
リスナースレッド	メッセージを受け取るスレッド．
セッションスレッド	タイマー処理を行うスレッド．
コールバックスレッド	コールバックを呼び出すスレッド．

メインスレッドは，PANA APIが利用されると最初に作られるスレッドで，リスナースレッドとセッションスレッドを作る．リスナースレッドはselect()で常にメッセージを待ち続ける．メッセージを受け取るとコールバックスレッドを作成し処理を実行させるが，PANA-Client-Initiationだけはこのスレッド内で処理される．セッションスレッドはPANA Sessionのライフタイム，再認証，メッセージの再送，PANA-Ping-Requestの送信を常にタイマーで管理し，どれかがタイムアウトするとコールバックスレッドを作成し処理を実行させる．

4.3.3 設定ファイル

PANA APIでは，設定ファイルから初期情報を読み込み初期化を行う．設定ファイルの形式にはフリーの解析ツールが豊富なXML形式をとる．ここで設定した情報は，メインスレッド開始時に初期化され，そのメインスレッドが作成したすべてのスレッドに適用される．設定ファイルでの設定項目を表2に示す．

表 2: 設定ファイルの設定項目

	内容
pana-port	UDP ポート番号
PAA-Identify	PAA の IP アドレス
PDI-IRT	再送の初期タイムアウト時間
PDI-MRT	最大再送タイムアウト時間
PDI-MRC	最大再送回数
PDI-MRT	最大再送期間
SessionTimeout	セッションのタイムアウト時間
SessionPing	Ping の送信間隔
process	プロセスの役割 (PAA or PaC)
ReauthTime	再認証の間隔

4.3.4 構造体・関数

PANA APIでは，デーモンを実装するために操作する構造体として重要な役割を果たす構造体としてPANASession構造体，PANA_AVP構造体，PANAMessage構造体の3つの構造体が定義されている．PANASession構造体はPANA Sessionの保持する値をメンバとして持つ．PAAは複数のPaCとの間にPANA Sessionを持つことが可能であるため，PANASession構造体はリスト構造をとり，PaC-PAAの組み合わせにより一意に定義される識別子をキーとして検索することができる．PANA_AVP構造体はAVPの値を扱う構造体で，PANAでは1つのメッセージが複数のAVPを持つため，リスト構造をとる．PANAMessage構造体はPANAメッセージを表しPANA_AVP構造体のリストをメンバとして持つ．

また関数としては以下に挙げる関数が定義されている．

- PANA APIの初期化・解放を行う関数．
- コールバックを登録・削除する関数．
- PANA Sessionを管理する関数．
- PANAメッセージを管理する関数．

PANA APIの初期化・解放を行う関数にはPANAOpen(), PANAClose()が定義されており，PANAOpen()は設定ファイルの読み込み，スレッドの作成といった初期化処理を行う．PANAClose()ではスレッドの削除，メモリ領域の解放といった処理を行う．

コールバックを登録・削除する関数にはメッセージコールバックを登録・削除する関数であるPANARegisterMessageCallback(), PANADeregisterMessageCallback()と，タイマーコールバックを登録・削除する関数であるPANARegisterTimerCallback(), PANADeregisterTimerCallback()が定義されている．PANARegisterMessageCallback(), PANARegisterTimerCallback()では登録したコールバックを一意に識別する値が返り，その値を引数としてPANADeregisterMessageCallback(), PANADeregisterTimerCallback()で登録したコールバックを削除する．

PANA Sessionを管理する関数には，PANANewSession(), PANAEndSession(), PANAGetSessionAttribute(), PANASetSessionAttribute()が定義されている．PANANewSession(), PANAEndSession()はPANASession

構造体の初期化・解放を行う関数で，PANASetSessionAttribute()，PANAGetSessionAttribute() は PANASession 構造体のメンバを更新・取得する関数である．

PANA メッセージを管理する関数には，PANANewMessage()，PANAFreeMessage()，PANASendMessage()，PANACreateAndAddAVPToList()，PANAFreeAVP() が定義されている．PANANewMessage()，PANAFreeMessage() は PANAMessage 構造体を初期化・解放する関数で，PANASendMessage() はメッセージを送信する関数である．PANACreateAndAddAVPToList() は PANA_AVP 構造体を初期化し AVP リストに追加する関数で，PANAFreeAVP() は AVP リストから特定の PANA_AVP 構造体を削除し解放する関数である．

PaC/PAA デーモンは，PANAOpen()，PANARegisterMessageCallback()，PANARegisterTimerCallback() を最初に実行し，API の初期化・コールバックの登録を行う．それぞれのコールバックの中では受け取ったメッセージから PANASetSessionAttribute() により PANASession 構造体のメンバを更新し，PANANewMessage() で新しいメッセージを作成して PANASendMessage() で作成したメッセージを送信するという処理を行うことで実装可能である．

5 検証

ここでは，本研究で構築したネットワークアクセス認証システムの動作検証について説明する．動作検証では (株) 東芝研究開発センターが独自に実装した PANA である cPANA との相互接続性の確認を行った．

5.1 実験環境

相互接続性の確認は，図 7 に示すネットワーク上で行った．インターネットを介した 2 つの異なるネットワーク上にそれぞれの PaC と PAA を設置した．

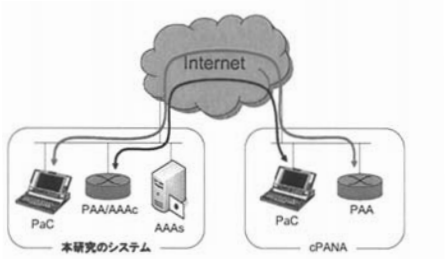

図 7: 実験に利用したネットワーク

5.2 検証結果

cPANA との相互接続性の確認は，図 5 に示したメッセージフローに基づいて実験を行った．具体的な検証項目を以下に挙げる．

- 各 Phase でのメッセージの送受信．
- Authentication and Authorization Phase での認証．
- Access Phase での PANA Session のタイマー管理．
- Re-Authentication Phase での再認証．
- Termination Phase での PANA Session の終了．
- メッセージの検証 (MAC，シーケンス番号)．
- メッセージの再送．
- エラー時の処理．

この実験では，本研究で実装した PaC と cPANA の PAA，cPANA の PaC と本研究で実装した PAA との間で PANA による認証を行い，それぞれの場合について上記の検証項目を確認した．実験は計 4 日間行い，最初は簡単なアルゴリズムしか用いない EAP-MD5 を利用し，MSK を生成しない場合の正常な動作を確認した．その後，EAP-PSK を利用し MSK を生成した場合の動作を確認したところ，EAP-PSK で認証に利用される暗号化アルゴリズムや MSK から PANA_AUTH_KEY を生成するアルゴリズムに問題が見つかったが，最終的に正常な動作を確認することができた．最終的には PANA の仕様で定義されている動作を一通り確認することができ，本研究で設計・実装した API は実用する上で充分な有効性を持つことが確認できた．

6 結論および今後の課題

本研究では，PANA を利用したネットワークアクセス認証システムを構築を目的とし，そのために必要な PANA API を定義し，実装した．本研究で仕様を定義した PANA API を利用して実装したデーモンと東芝が独自に実装したデーモンとの相互接続性の確認を行い，両者の相互接続性は充分にあることが確認できた．このことから，本研究で仕様を定義した PANA API は実用する上で充分な有効性を持っていることが証明された．

本研究では検証事項として相互接続性の確認を行ったが，システム全体の実用性を検証するためには大規模な環境での多数のユーザからの認証要求に対するスケーラビリティの測定が必要となる．今後は，このようなスケーラビリティの測定を行うことで本研究で構築したシステムの実用性を確認していきたい．

参考文献

[1] D.Forsberg, Y.Ohba (Ed.), B.Patil, H.Tschofenig and A.Yegin. Protocol for Carrying Authentication for Network Access (PANA). Internet Draft, *IETF*, Aug. 2006. draft-ietf-pana-pana-12.txt.

[2] B.Aboba, L.Blunk, J.Vollbrecht, J.Carlson and H.Levkowetz, Ed. Extensible Authentication Protocol (EAP). RFC 3748, IETF, Jun. 2004.

[3] F.Bersani and H.Tschofenig. The EAP-PSK Protocol: a Pre-Shared Key EAP Method. Internet Draft, *IETF*, Jun. 2006. draft-bersani-eap-psk-11.txt.

[4] P.Calhoun, J.Loughney, E.Guttman, G.Zorn and J.Arkko. Diameter Base Protocol. RFC 3588, IETF, Sep. 2003.

[5] P.Eronen, Ed., T.Hiller and G.Zorn. Diameter Extensible Authentication Protocol (EAP) Application. RFC 4072, IETF, Aug. 2005.

[6] IEEE. Standard for Port based Network Access Control. IEEE, 2001. http://standards.ieee.org/getieee802/download/802.1X-2004.pdf.