Vol.2011-CSEC-53 No.14
Vol.2011-IOT-13 No.14
2011/5/13

Diameter EAP Application上における EAP-TTLSの実装と評価

厚谷有輝[†1]　寺岡文男[†2]

EAP-TTLS はネットワークアクセスにおける認証方式の 1 つで，サーバの認証に電子証明書を，ユーザの認証に TLS トンネルを利用したパスワードベースの認証を行うことで相互認証をサポートする．同じく TLS ハンドシェークを利用する EAP-TLS に比べてユーザに証明書を配布する必要がない分，運用が容易である．本研究では Diameter EAP Application 上に EAP-TTLS 認証システムを実装し，EAP-TTLS 内部での認証方式には主要な 4 つの認証方式 (PAP，CHAP，MS-CHAP，MS-CHAPv2) をサポートした．検証の結果，複数のクライアントマシンで正常に認証処理を行うことができ，処理時間も運用する上で問題のない時間であることを確認した．

Implemantation and Evaluation of EAP-TTLS on Diameter EAP Application

Yuki Atsuya[†1] and Fumio Teraoka[†2]

EAP-TTLS is one of the network access authentication methods. It provides mutual authentication by using certificate for server authentication, and password transmitted in TSL tunnel for user authentication. Because distribution of certificate to users is not necessary, EAP-TTLS is easier to use than EAP-TLS. In our research, we implemented a EAP-TTLS authentication system on Diameter EAP Application. We also implemented PAP, CHAP, MS-CHAP, MS-CHAPv2 as the authentication methods inside EAP-TTLS. As a result of tests, our implementation works correctly on several operating systems, and the processing time is short enough for practical use.

†1 慶応義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
†2 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

1. はじめに

近年，小型ラップトップ PC や携帯情報端末の普及に伴い，外出先でのインターネット利用の要求が高まっている．このような環境ではユーザは自身の契約する ISP (Internet Service Provider) に接続するとは限らない．むしろ直接は契約していない ISP のネットワークに接続する機会が増えると考えられ，ユーザが自由にインターネットへ接続することを可能にするためには，ISP 間で互いの管理するユーザ情報を交換するための機能が必要となる．

こうしたマルチドメイン環境におけるユーザ情報交換の為のプロトコルとして，マルチドメイン環境での AAA システムの利用を想定して設計された Diameter Base Protocol (Diameter)[1] が挙げられる．AAA システムとは Authentication (認証)，Authorization (権限委譲)，Accounting (利用情報記録) の頭文字を取ったもので，ユーザ管理を一元的に行うためのインターネットにおけるサービス提供に関するセキュリティの概念である．現在広く AAA バックエンドとして利用されているプロトコルに Remote Authentication Dial-In User Service (RADIUS)[2,3] があるが，元々シングルドメイン用に設計されており，メッセージ転送の信頼性，セキュリティ，耐障害性などに問題点を抱えている．Diameter はこれら RADIUS の問題点を解決するために設計された後継プロトコルである．Diameter Base Protocol ではノード間のセッション管理メッセージやルーティングメッセージなど，マルチドメイン環境における AAA システムを構築するための基本的な要素を定義しているのみである．そこで具体的なネットワークサービスに Diameter Base Protocol に基づいた AAA 処理を組み込むため，Diameter Application と呼ばれる拡張を定義している．Diameter EAP Application[4] は認証フレームワークである Extensible Authentication Protocol (EAP)[5] を Diameter へ適用したプロトコルで，マルチドメイン環境におけるネットワーク認証を可能とする．

外出先での認証を考えた場合，どのようにしてユーザをセキュアに認証するかという問題がある．盗聴による危険を防ぐためにも認証情報は十分強力な暗号化によって運ぶ必要があり，またアクセスポイントが攻撃者によって故意に仕掛けられたものでないか確認するためにも，認証サーバがユーザを認証するだけでなく，ユーザも認証サーバを認証することが望ましい．EAP-TLS[6] や EAP-TTLS[7] は電子証明書を利用することによってセキュアな相互認証を行うことができる認証方式である．EAP-TLS はお互いが自身の証明書を交換して信頼する認証局の署名によってその証明書の正しさを確認する．電子証明書によって十分なセキュリティを確保することができるが，証明書の配布や管理が難しい．一方 EAP-TTLS は

ユーザには証明書を必要とせず，まずユーザはサーバの証明書を見てサーバの正当さを確認した後に暗号化された TLS トンネル内で自身の認証を行う．TLS トンネルを通るメッセージは暗号化されており第三者は見ることができないため，CHAP (Challenge Handshake Authentication Protocol) のような簡単だがその脆弱性からそのままでは使用を控えるべき認証プロトコルも安全に使用することができる．これらの点から EAP-TTLS は相互認証によるセキュリティも確保しつつ，EAP-TLS に比べて運用が容易な認証方式であるといえる．

我々は Diameter Base Protocol の実装である freeDiameter[8]，および Diameter EAP Application の実装である DiamEAP[9] をすでに公開している．DiamEAP は EAP-TLS の実装も含んでいる．本研究で実装した EAP-TTLS 認証システムは freeDiameter と DiamEAP を利用したものである．

2. AAA システム

本章ではユーザがネットワーク上でサービスを利用する際に必要となる AAA システムについて説明する．

2.1 AAA システムの概要

一般に AAA システムはユーザがサービスの要求やサービスの提供に必要な情報を AAA サーバに渡すフロントエンドと，AAA サービスに必要な情報を AAA サーバ間で交換するバックエンドとに分けられる．ネットワークアクセスサービスに AAA を適用した場合はユーザから Network Access Server (NAS) までがフロントエンド，NAS から NAS とユーザ情報を交換する AAA Server までがバックエンドとなる．

マルチドメイン環境における AAA システムではユーザの管理はユーザの所属するホームドメインにある AAA サーバが行い，ユーザが別ドメインへ移動した場合には移動先の AAA サーバとユーザ情報を交換して管理する．例えば ISP-A と契約しているユーザが ISP-B のドメイン内でサービスを利用したい場合，通常であればユーザと ISP-B と契約を結び，ISP-B がユーザの管理を行う必要がある．しかし AAA システムではユーザの情報が ISP-A の AAA サーバによって管理されており，かつ ISP-A と ISP-B が事前にローミング契約を結んでいれば，ISP 間でユーザ情報を交換することによってサービス提供を実現する．これによって ISP-A は管理領域の外にいるユーザの管理が可能となる．

2.2 Diameter

本節ではマルチドメイン環境における AAA システムプロトコルとして設計された Diameter について説明する．

2.2.1 Diameter Base Protocol

Diameter はピア・ツー・ピアモデルのプロトコルで Diameter ノード間でオーバレイネットワークを構成している．トランスポート層プロトコルには TCP や SCTP を用いる．Diameter メッセージには AVP (Attribute Value Pairs) が格納される．AVP は RADIUS でも使用されるフォーマットで，属性値とその値がペアとなって構成されるフォーマットである．Diameter のメッセージではこの AVP が複数連なって Diameter メッセージに格納される．

現在 AAA バックエンドとして広く利用されているプロトコルは RADIUS であるが，RADIUS はもともとシングルドメインでの利用を想定して設計されたため，マルチドメイン環境ではいくつかの問題点を抱える．以下にその主な問題点と Diameter による改善点を挙げる．

- Failover: RADIUS では Failover の仕組みは仕様として定義されておらず，実装依存となっている．そのため異なる実装間では Failover の仕組みが正常に機能しない可能性がある．シングルドメインでの使用ならば全て同一の実装を利用することもできるが，マルチドメイン環境では難しい．Diameter では Failover の仕組みが仕様として定義されている．
- ISP 間の通信の保護: RADIUS では End-to-End の通信を保護するには EAP などのプロトコルを使用しなければ，途中の hop でメッセージが盗聴・改竄されてしまう恐れがあった．RADIUS では通信の保護に IPsec を利用することができるが，ISP 間で事前に秘密鍵を共有することは難しいためマルチドメイン環境には IPsec は適さない．一方 Diameter は ISP 間の通信に IPSec のみでなく TLS も利用することができる．
- 通信の信頼性: RADIUS ではコネクションレスにすることで構成をシンプルにする等の目的から UDP で通信を行うため，再送処理は Failover と同様に実装依存となる．Diameter では SCTP または TCP による通信を行うので信頼性が向上している．
- 拡張性: Diameter では AVP の新たな定義が可能である．そのためより多くの AVP が定義できるよう，メッセージの種類を示す Code フィールドが RADIUS は 1 オクテットから 4 オクテットへと拡張されている．また AVP にも Flags フィールドが用意され，Vendor ID フィールドによって Vendor 独自の AVP も扱うことが可能である．

このように Diameter は RADIUS に比べよりマルチドメイン環境での AAA バックエンドインフラストラクチャとしての利用を意識した設計がなされている．また新たな AVP やメッセージ，Diameter Application の定義が可能となっており拡張性にも富んだプロトコ

Vol.2011-CSEC-53 No.14
Vol.2011-IOT-13 No.14
2011/5/13

ルとなっている．

2.2.2 Diameter EAP Application

Diameter Base Protocol は AAA 情報の転送を提供するのみで，特定のサービスに対する AAA 機能は持たない．AAA バックエンドとしての機能を果たすには Diameter Application と呼ばれる Diameter Base Protocol 上で動作するプロトコルを用いる必要がある．Diameter Application は様々なプロトコルに対して Diameter による AAA システムとしての機能を提供する．

Diameter EAP Application は Diameter Base Protocol 上で EAP による認証を可能にする Diameter Application で，RFC 4072 によって標準化されている．新たな Diameter メッセージとして Diameter-EAP-Request/Answer を定義し，EAP パケットを格納するための AVP も定義している．Diameter EAP Application での EAP-TTLS 認証シーケンスについては 2.3.3 節で詳しく述べる．

2.3 Extensible Authentication Protocol (EAP)

本節ではユーザの認証フレームワークである EAP について説明する．

2.3.1 EAP の概要

EAP では認証を行うノードを Authenticator，認証されるノードを Peer と定義する．もとは Point to Point Protocol (PPP) を拡張して認証の機能を持たせたプロトコルであるため，イーサネット上で動作させるためには EAP をカプセル化した EAP over LAN (EAPOL) を用いる．AAA システムで EAP を用いる際は AAA フロントエンドでは EAPOL で，AAA バックエンドでは RADIUS や Diameter といったプロトコルを用いて EAP パケットが運搬される．EAP 自体にも認証の機能が規定されているが，その拡張性の高さから実際には他の認証方式と組み合わせて動作させた EAP メソッドが利用されることが多い．

2.3.2 EAP-TLS

EAP-TLS は Transport Layer Security (TLS)[10] を利用する EAP メソッドである．TLS は電子証明書によってユーザとサーバの相互認証を行うプロトコルで，高度なセキュリティを保証する．ユーザとサーバは EAP Identity の交換および EAP Hello メッセージによって TLS セッションを開始し，Certificate メッセージを交換することでお互いの証明書を受け渡し確認する．そしてサーバはユーザの認証に成功すると EAP-Success をユーザへ返す．

このように EAP-TLS は電子証明書によって高度なセキュリティを確保した認証を行うことができるが，管理者は全てのユーザの証明書を作成して配布する必要がある．またユーザも自身の電子証明書を常に持ち歩かなければならず，電子証明書をインストールすることができないマシンからは利用することができない．

表 1 EAP-TLS と EAP-TTLS の比較

	認証方向	認証方法	安全性	運用の用意さ
EAP-TLS	相互認証	ユーザ・サーバともに電子証明書を利用して認証	高い	ユーザに証明書を配布する必要あり
EAP-TTLS	相互認証	TLS トンネル内でユーザを認証	高い	ユーザの証明書はオプション

2.3.3 EAP-TTLS

EAP-TTLS は EAP-TLS と同様に TLS ハンドシェークを利用した認証を行う EAP メソッドである．EAP-TTLS には Phase 1 と Phase 2 の 2 つのフェーズがあり，Phase 1 は TLS ハンドシェークによって Peer が Authenticator を認証する．このとき Peer と Authenticator の間で TLS トンネルを確立し，Phase 2 では TLS トンネル内で Authenticator が Peer の認証を行う．EAP-TLS と同様，Phase 1 の TLS ハンドシェーク時に Peer は自身の電子証明書を Authenticator に示すことによって認証を受けることもできる．この場合 Phase 2 の認証は行われず，Peer が電子証明書を渡さなかった場合に Phase 2 へ移行する．Phase 2 では PAP や CHAP のようなパスワードベースの認証方式を利用することができる．

EAP-TLS と EAP-TTLS の比較を表 1 に示す．TLS ハンドシェークを利用する点・双方向認証である点・TLS トンネルによる安全性の高さはどちらも共通している．両者の大きな違いは EAP-TTLS はサーバは必ずしも電子証明書によるユーザ認証を行わなくても良いということである．管理者は各ユーザへ証明書を配布する必要がなく，またユーザも常にそれを保持する必要がないためクライアントマシンを固定されない．つまり EAP-TTLS は電子証明書のみを利用することに比べたら安全性は劣るが，運用が比較的容易である．

Diameter EAP Application における EAP-TTLS 認証のメッセージフローを図 1 に示す．ここで NAS (Network Access Server) は EAP Authenticator として動作し，ユーザからの EAP メッセージを Diameter メッセージとして AAA サーバへ転送する．Wi-Fi アクセスポイントなどがこれに当たる．AAA サーバはユーザ情報を管理しており，ユーザの認証を行う Diameter サーバである．以下に Diameter EAP Application 上での EAP-TTLS の動作を示す．

- NAS が EAP-Request/Identity を Peer へ送信する (1).
- ユーザが EAP-Response/Identity を NAS へ送信する (2).
- これを受け取った NAS は EAP パケットを AVP へ格納し，Diameter-EAP-Request

Vol.2011-CSEC-53 No.14
Vol.2011-IOT-13 No.14
2011/5/13

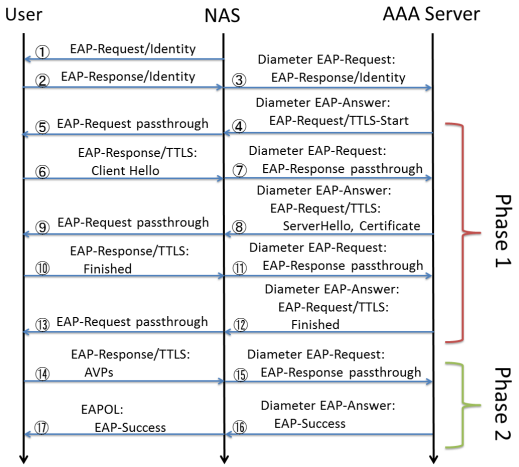

図 1 認証開始から終了までのシーケンス

メッセージを生成して AAA サーバへと送信する (3).

- Diameter-EAP-Request を受け取った AAA サーバは EAP-TTLS を開始する Start メッセージを作成し，Diameter-EAP-Answer メッセージとして NAS へ送信する (4).
- NAS は Diameter-EAP-Answer メッセージの AVP から EAP パケットを取り出し，ユーザへ送信する (5).
- ユーザは EAP の TTLS-Start メッセージを受け取ると ClientHello メッセージを送信し，NAS は Diameter メッセージにして AAA サーバへ送信する．ClientHello メッセージには Master Session Key (MSK) を生成するための乱数などが含まれている (6, 7).
- AAA サーバは ServerHello メッセージと電子証明書をユーザへ送信する (8, 9).
- Finished メッセージによって TLS トンネルを確立し，Phase 1 が終了する (10-13).
- Phase 2 へ移行すると，ユーザは認証に必要な情報を AVP 形式で送信する (14, 15).
- AAA サーバは認証処理が完了すると EAP-Success もしくは EAP-Failure を Diameter-EAP-Answer メッセージに格納して送信する (16, 17).

NAS がユーザと認証サーバの間で Diameter メッセージの処理を行っており，実際にユーザの認証を行うのはバックエンドに位置する AAA サーバとなっている．

3. freeDiameter および DiamEAP

本研究では Diameter Base Protocol の実装として開発された freeDiameter と，Diameter EAP Application の実装である DiamEAP を利用している．本章ではこれらソフトウェアについて説明する．

3.1 freeDiameter

freeDiameter は独立行政法人情報通信研究機構 (NICT) によって開発されたオープンソースソフトウェアで，RFC 3588 で定義されている Daiemter Base Protocol の実装である．2010 年 6 月に最初の安定版がリリースされ，2010 年 11 月に DiamEAP とあわせ NICT によるプレスリリースが行われた．C 言語で開発されており，プログラムの改変・商用化も可能な BSD ライセンスであるため活用や応用が容易である．2011 年 3 月 29 日にはバージョン 1.1.0 がリリースされた．

freeDiameter は主に libfreeDiameter，freeDiameterd，extension の 3 つのコンポーネントで構成されている．各 Diameter Application は extension として提供される．また Diameter メッセージと RADIUS メッセージの変換を行う Diameter/RADIUS ゲートウェイも extension として実装されている．

3.2 DiamEAP

DiamEAP は寺岡研究室によって開発された freeDiameter の extension であり，EAP による認証を可能にする Diameter Application の実装である．2011 年 1 月 28 日現在はバージョン 1.0.2 がリリースされている．freeDiameter と同様 BSD ライセンスで利用が可能である．DiamEAP では EAP メソッドは plug-in として独立して実装されており，DiamEAP の実行時に動的に読み込まれる．そのため新たに実装した EAP メソッドを DiamEAP へ組み込むことが容易なっている．現在実装されている EAP メソッドは EAP-TLS と EAP-MD5 の 2 つである．

2010 年の 9 月に行われた WIDE プロジェクトの合宿では実際のマルチドメイン環境で試験運用を行った．EAP メソッドは EAP-TLS を用い，WIDE メンバーに配布されている証明書を使用した．約 200 人が試験運用に参加した 4 日間の合宿で問題なく運用できたことを確認している．また寺岡研究室でも研究室所属のメンバーに証明書を配布しており，認証サーバとして 6ヶ月以上運用している．

Vol.2011-CSEC-53 No.14
Vol.2011-IOT-13 No.14
2011/5/13

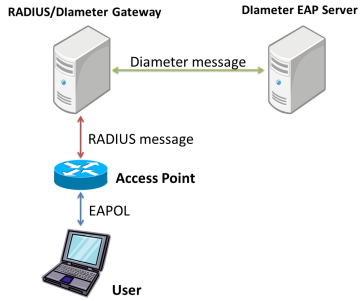

図 2 システム構成

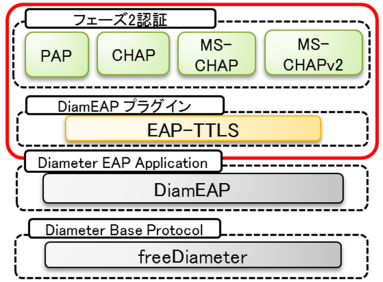

図 3 実装範囲

4. 設計および実装

4.1 設　　計

本研究で想定するシステムの構成を図 2 に示す．ユーザの認証は Diameter EAP サーバ上で行われる．今回使用したアクセスポイントは Diameter プロトコルをサポートしていないため，アクセスポイントと Diameter EAP サーバの間には freeDiameter の extention として提供されている RADIUS と Diameter のメッセージの相互変換を行う RADIUS/Diameter ゲートウェイを設置する．

EAP-TTLS の設計は RFC 5281 にのっとって行った．EAP-TTLS plug-in によってサポートされるフェーズ 2 認証方式は PAP，CHAP，MS-CHAP，MS-CHAPv2 の 4 つである．ユーザは EAP-TTLS のフェーズ 2 認証には，EAP-TTLS plug-in のサポートしている認証方式であれば自由に選択できる．あらかじめユーザがどの認証方式を使うかについては登録しておく必要はなく，例えば MS-CHAPv2 による認証をユーザが望むのであればユーザはサーバがフェーズ 2 認証に MS-CHAPv2 を利用するものとしてアクセス要求をし，認証を行うことができる．

4.2 実　　装

本研究の実装範囲を図 3 に示す．赤枠で囲った部分が本研究で新しく実装を行った箇所である．また実装環境は表 2 に示すとおりである．

DiamEAP ではプラグインで用いる関数をコールバックとして登録することで DiamEAP がそれらの関数を管理し，呼び出しを制御する．以下に本研究で利用した主なコールバックを挙げる．

表 2 実装環境

項目	内容
OS	Ubuntu 10.04
言語	C
parser	yacc/lex
Diameter Base Protocol	freeDiameter
Diameter EAP Application	DiamEAP
TLS 暗号化	GnuTLS Library

- Config Function: コンフィグファイルを読み込むときに用いる関数．DiamEAP の起動時に呼ばれる．
- Init Function: データの初期化を行う関数．DiamEAP の起動時に呼ばれる．
- Check Function: EAP パケット受信時に呼ばれる関数．本実装では Type フィールドの検査を行っている．
- Process Function: Check Function 実行後に呼ばれる関数．本実装では EAP パケットフラグメントの確認・結合と，フェーズ 1 の処理，フェーズ 2 の処理を行っている．
- BuildReq Function: EAP-Request パケットを生成する関数．EAP-Request/START の Process Function で適切な処理をした後に呼ばれ，この関数で生成した EAP パケットを DiamEAP が Diameter メッセージに格納して Peer へ送信する．
- getKey Function: Master Session Key を生成する関数．認証成功時に呼ばれる．

他にも，Phase 2 の認証方式を決定する関数である diameap_eap_ttls_AuthMethod() や，Phase 2 における認証である PAP，CHAP，MS-CHAP，MS-CHAPv2 の処理を行う関数もそれぞれ実装した．

5. 評　　価

本章では本研究で実装した EAP-TTLS plug-in の評価および考察を行う．測定項目は認証に要する全体の時間と，Phase 2 の各認証方式の処理時間である．それぞれ 5 回ずつ測定した．またすでに実装として存在する EAP-TLS plug-in を使用した場合の時間も測定し，比較した．測定に使用したマシンを 3 に示す．

5.1 動 作 確 認

実装をした EAP-TTLS サーバが正しく動作するか，パケットキャプチャを行い確認した．その結果，仕様通りのシーケンスで認証が行われており，各メッセージにも適切な値

Vol.2011-CSEC-53 No.14
Vol.2011-IOT-13 No.14
2011/5/13

表 3 実験環境

マシン	CPU	メモリ
Client	Intel Core2 Duo 1.40GHz	4GB
RADIUS/Diameter Gateway	Intel Core2 Quad 2.66GHz	4GB
Diameter EAP Server	Intel Core2 Quad 2.66GHz	4GB

表 4 実験環境

	Windows7	Ubuntu	iPad	Xperia
PAP	○	○	端末が未対応	端末が未対応
CHAP	○	○	端末が未対応	端末が未対応
MS-CHAP	○	○	端末が未対応	○
MS-CHAPv2	○	○	○	○

が格納されていることが確認できた．不正なユーザ ID やパスワードの入力があった際には Access-Reject を返し，認証を失敗させた．アクセスポイントの設定で適切に DHCP サーバを指定すれば，認証成功後にクライアントマシンは IP アドレスを取得することができた．またクライアントマシンの種類による動作結果は表 4 に示す通りとなった．この表に示すように複数の OS，クライアントマシンで正常に動作したことを確認した．

5.2 処理時間の測定

認証処理全体に要した時間と Phase 2 の処理時間について，EAP-TTLS の各認証方式と EAP-TLS でそれぞれ測定した．結果を表 5 に示す．

まず認証処理全体にかかった時間は EAP-TTLS が EAP-TLS の約半分程度の処理時間で完了した．これは EAP-TTLS ではユーザは自身の電子証明書をサーバへ渡す必要がないため，認証処理の時間が大幅に減ったと考えられる．つまり処理にかかる時間のほとんどは電子証明書の交換・検証に依存するのではないかと考えられる．また Phase 2 の処理時間も，複雑な認証方式になるに連れ多少処理時間が増加しているが，処理全体から見たら大きくない時間で Phase 2 認証を完了することが出来た．

6. ま と め

本研究では Diameter EAP Application の実装である DiamEAP の plug-in として EAP-TTLS を設計・実装し，実験によりその有用性を確かめた．認証処理にかかる時間はすでに実装として存在する EAP-TLS よりも短くなり，運用する上で問題のない範囲であることを確認した．また Phase 2 における認証方式では PAP，CHAP，MS-CHAP，MS-CHAPv2 の主要な 4 つの方式を実装し，サポートした．これによって EAP-TTLS をサポートするクライアントマシンの多くで，Diameter Base Protocol を利用した認証を行うことができるようになった．

表 5 測定結果

	Phase 2 認証方式	認証処理全体 (msec)	Phase 2 処理 (msec)
EAP-TTLS	PAP	16.39	0.184
	CHAP	16.06	0.217
	MS-CHAP	16.02	0.442
	MS-CHAPv2	15.83	0.436
EAP-TLS	-	29.08	-

参 考 文 献

1) Calhoun, P., Loughney, J., Guttman, E., Zorn, G. and Arkko, J.: *Diameter Base Protocol* (2003). RFC 3588, IETF.
2) Willens, C. R.S., Rubens, A. and Simpson, W.: *Remote Authentication Dial In User Service (RADIUS)* (2000). RFC 2138, IETF.
3) Nelson, D. and DeKok, A.: *Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes* (2007). RFC 5080, IETF.
4) Eronen, P., Hiller, T. and G.Zorn: *Diameter Extensible Authentication Protocol (EAP) Application* (2005). RFC 4072, IETF.
5) Blunk, L. and J.Vollbrecht: *PPP Extensible Authentication Protocol (EAP)* (1998). RFC 2284, IETF.
6) Simon, D., Aboba, B. and Hurst, R.: *The EAP-TLS Authentication Protocol* (2008). RFC 5216, IETF.
7) Funk, P. and Blake-Wilson, S.: *Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)* (2008). RFC 5281, IETF.
8) Decugis, S. and Teraoka, F.: freeDiameter: An Open Source Framework for an Authentication, Authorization, and Accounting Infrastructure., コンピュータソフトウェア (刊行予定).
9) Ben Ayed, S. and Teraoka, F.: DiamEAP: an Open-Source Diameter EAP Application and Its Evaluation, *Proceedings of the 16th Asia-Pacific Conference on Communication (APCC 2010)*, pp.515 –520 (2010).
10) Dierks, T. and Rescorla, E.: *The Transport Layer Security (TLS) Protocol Version 1.1* (2006). RFC 4346, IETF.