推薦論文

IPsec と IKE を用いたユーザアクセス制御の枠組みの提案と WWW サーバへの適用

平野 学[†] 木村泰司[†] 山口 英[†]

IPsec と IKE はホスト認証の結果で得られた SA に基づき IP データグラムの暗号化を可能とする．しかし IPsec と IKE を上位層プロトコルの保護に適用しようとした場合には，ユーザの概念が IPsec と IKE に実装上欠如しているために適用が難しい．このため，IP 層に強力なセキュリティ機構が存在するにもかかわらず，アプリケーションでは独自の認証機構を実装しなければならない状況にある．この問題を解決するためには次の 2 つの課題を解決しなければならない．第 1 には IKE そのものはユーザ認証にも拡張可能であるにもかかわらず，IPsec とともに使われる現在の IKE はホスト認証だけを行うのでアプリケーションのユーザ認証にそのまま利用できないことである．第 2 の課題は IPsec は IP データグラムとユーザを対応付ける機構がないので同一ホストから複数ユーザの IP データグラムを区別できないことである．そこで本論文では認証されたユーザの識別子とトランスポート層におけるソケットペアの関連性を保証し，IKE のユーザ認証の結果をアプリケーションのアクセス制御に利用するために UADB（User Authentication Database）方式を提案する．本機構により IPsec/IKE を用いて，アプリケーションでのユーザ認証，および，アクセス制御を可能とする．さらに本方式を WWW に適用する実装を示し，その実用性の高さを示す．

Design and Implementation of User Oriented Access Control Framework Using IPsec and IKE for WWW Servers

Manabu Hirano,[†] Taiji Kimura[†] and Suguru Yamaguchi[†]

IPsec and IKE are considered as fundamental security mechanisms to protect IP layer and can secure its upper layer protocols using its host authenticated SA. However, these IPsec and IKE mechanisms basically were designed as a host oriented security mechanism so that it is difficult to use them for user oriented security mechanisms in applications. Therefore, even an underlying IP does have a strong security mechanism, still applications have to implement their own security mechanism, especially for user authentication. Specifically, the following design aspects are major reason why we can't use them for user oriented security mechanism: (1) ordinal use of IKE with IPsec does not have any concept of "user" even it can be extensible for user authentication, and (2) IPsec does not have any mechanisms that make proper mapping between IP datagrams and a user. Our challenge shown in this paper is to use IPsec and IKE framework for user authentication, without any modifications in IP layer mechanism. In this paper, we propose the UADB (User Authentication Database) system. This system enables us to use IKE for user authentications access controls for applications. This paper shows our implementation and a good example of its application for WWW service.

1. まえがき

IPsec（IP security protocol）[1]と IKE（Internet Key Exchange）[2]は VPN（Virtual Private Network）を構築するために用いることができる[3]．しかし，IPsec には上位層プロトコルの保護を目的としたトランスポートモードが存在するにもかかわらず，ホストのみを SA（Security Association）が対象とするエンティティとしてとらえた IPsec と IKE の仕様上の制約から，アプリケーションプログラムの暗号化通信路と認証には直接利用することができない．このため，IP 層に強力なセキュリティ機構が存在するに

† 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

本論文の内容は 2002 年 3 月の情報処理学会九州支部主催「火の国情報シンポジウム」にて報告され，同支部長により情報処理学会論文誌への掲載が推薦された論文である．

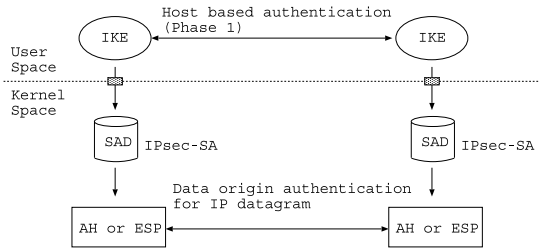

図 1 IKE のホスト認証と IPsec のデータ生成元の認証
Fig. 1 Data origin authentication in IPsec and host authentication in IKE.

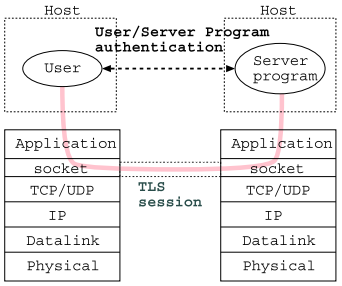

図 2 TLS におけるセッションと認証の関係
Fig. 2 Session and authentication in TLS.

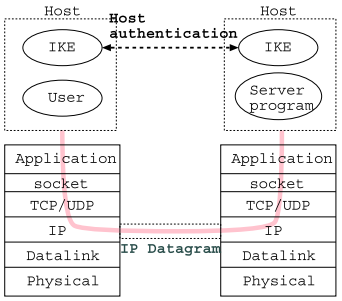

図 3 IPsec における IP データグラムと認証の関係
Fig. 3 IP datagram and authentication in IPsec.

もかかわらず，アプリケーションでは独自の認証機構を実装しなければならない状況にある．

本論文はこの課題を解決するためにユーザ認証データベース（UADB: User Authentication Database）方式を提案する．UADB 方式は認証されたユーザの識別子とトランスポート層におけるソケットペアの関連性を保証し，IKE プロトコルのユーザ認証の結果をアプリケーションのアクセス制御に利用する．本論文は提案方式を WWW（World Wide Web）に適用する実装を示す．

本論文は，まず 2 章で IPsec/IKE をアプリケーションで利用する際の制約を述べる．次に 3 章でこの制約を改善するための提案方式を述べる．4 章で提案方式を既存の WWW システムに導入する設計を行い，5 章でプロトタイプ実装を示す．6 章でプロトタイプ実装の性能測定を行う．7 章で証明書を用いた認証処理おける TLS との比較を行う．8 章でサーバプログラムに提案方式を実装するための API を示す．9 章で提案方式の WWW 以外のアプリケーションへの応用について述べる．10 章で今後の課題を述べる．11 章で関連研究について述べ，最後に 12 章でむすびを述べる．

2. IPsec/IKE をアプリケーションで利用する際の制約

IPsec と IKE をアプリケーションプログラムの暗号化通信路と認証の機能として利用するにはいくつかの制約が存在する．

2.1 IKE の認証機構を利用できない理由

SA には IKE プロトコルのフェーズ 2 のセキュリティ処理を決定する IKE-SA と IPsec のセキュリティ処理を決定する IPsec-SA がある．**図 1** に IKE のホスト認証と IPsec のデータ生成元の認証の仕組みを示す．まず IKE プログラムはフェーズ 1 でホスト認証を行い IKE-SA を確立する．次に IKE プログラムはフェーズ 2 で IKE-SA に保護された通信路を使って IPsec-SA を確立する．したがって AH（Authentication Header）[4] と ESP（Encapsulating Security Payload）[5] はホスト認証に基づく IPsec-SA を使ってデータ生成元の認証を行う．このように IKE はネットワーク層におけるホスト認証機構であるためアプリケーションプログラムのユーザ認証機構として利用することができない．

2.2 IPsec の暗号化通信路を利用できない理由

同一ホストから複数のユーザが IPsec の暗号化通信路を利用して接続する場合，サーバプログラムがユーザを識別できない問題が生じる．この問題について TLS（Transport Layer Security）[6] と比較して述べる．TLS はセッションを単位としてユーザを識別して暗号化通信路を処理する．**図 2** に TLS におけるセッションと認証の関係を示す．ユーザとサーバプログラム間の認証はセッションに対して有効である．それに対して IPsec は IP データグラムを単位としてセキュリティ処理を行うため，TLS のようにセッションを扱う機構がない．**図 3** に IPsec における IP データグラムと認証の関係を示す．IKE プロトコルのホスト認証はホスト間でやりとりする各 IP データグラムに対して有効である．また IP ヘッダ，AH ヘッダ，ESP ヘッダにはユーザを識別する情報が含まれない．したがって IPsec はユーザと IP データグラムを対応付け

表 1 UADB のデータベース形式
Table 1 UADB format.

有効期限
UNIX のユーザ ID か公開鍵証明書の識別名
終点 IP アドレス
終点ポート番号
始点 IP アドレス
始点ポート番号の数
始点ポート番号（指定した個数）
電子署名

る機構を持たない．IPsec-SA とホストの対応付けができても，IPsec-SA とユーザの識別子を対応付けることはできない．以上の理由から IPsec の暗号化通信路を利用したサーバプログラムは，同一ホストから接続してくる複数のユーザを区別することができない．

3. 提 案 方 式

本論文では 2 章で述べた制約を改善するために，ユーザ認証データベース（UADB: User Authentication Database）方式を提案する．我々は IKE プロトコルが ISAKMP ペイロードを使ってメッセージを交換することに注目し，新たにユーザ識別子とソケットペアを関連付けるための ISAKMP ペイロードを定義する．これにより IKE プロトコルの基本的な枠組みを変更することなく IKE プロトコルの認証の適用範囲をユーザにまで拡張する．

3.1 UADB のデータベース形式

UADB は，**表 1** に示すパラメータを 1 つのレコードとして扱うデータベースとして定義する．

有効期限は UADB レコードの有効期限を示す．有効期限が切れた UADB レコードは UADB から削除される．UNIX のユーザ ID または公開鍵証明書の識別名はユーザの識別子を示す．終点 IP アドレスはクライアントプログラムが接続するサーバホストの IP アドレスを示す．終点ポート番号はユーザが接続するサービスのポート番号を示す．始点 IP アドレスはクライアントホストの IP アドレスを示す．始点ポート番号の数は UADB レコードに記述する始点ポート番号の数を示す．始点ポート番号はプロキシが使用するソケットの始点ポート番号を示す．このように複数の始点ポート番号を記述することで複数の TCP コネクションを管理できる．電子署名は以上のフィールドを 1 方向ハッシュ関数で処理してユーザの秘密鍵で署名した値である．

3.2 UADB 方式の目的

UADB 方式の目的は電子署名によるユーザ認証，ソケットペアとユーザ識別子の対応付けの 2 つである．

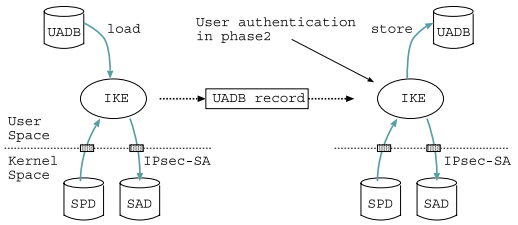

図 4 IKE のフェーズ 2 における UADB レコードの転送とユーザ認証
Fig. 4 Sending UADB record and user authentication via IKE phase2.

3.2.1 電子署名によるユーザ認証

第 1 の目的は電子署名によるユーザ認証である．これは 2.1 節で述べた IPsec と IKE をアプリケーションの認証に利用する際の制約を改善する．**図 4** にフェーズ 2 における UADB レコードの転送とユーザ認証の処理を示す．

IKE プログラムはフェーズ 2 で UADB レコードをサーバホストの IKE プログラムに転送する．受信側の IKE プログラムは UADB レコードに含まれる電子署名をユーザの公開鍵で検証する．これによりユーザを認証する．ユーザ認証に成功すれば受信側 IKE プログラムは UADB レコードを UADB に保存する．そして両ホストの IKE プログラムは IPsec-SA を確立する．保存した UADB レコードは 3.2.2 項で述べるサーバプログラムが参照する．ユーザ認証に失敗した場合，受信側の IKE プログラムは UADB レコードを破棄する．そして両ホストの IKE プログラムは IPsec-SA を確立しない．フェーズ 2 でユーザ認証を行うことで認証されたユーザのためだけの IPsec-SA を確立することができる．

3.2.2 ソケットペアとユーザ識別子の対応付け

第 2 の目的はソケットペアとユーザ識別子の対応付けである．これは 2.2 節で述べた IPsec と IKE をアプリケーションプログラムの暗号化通信路に利用する際の制約を改善する．ソケットペアは（始点 IP アドレス，終点 IP アドレス，始点ポート番号，終点ポート番号）から構成される．ユーザの識別子は UNIX のユーザ ID か X.509 形式の公開鍵証明書[7]の識別名（DN: Distinguished Name）を使用する．サーバプログラムによるソケットペアとユーザ識別子の対応付けの処理を**図 5** に示す．

クライアントホストの IKE プログラムは，サーバプログラムがクライアントプログラムからの接続を受ける前に，UADB レコードをサーバホストの IKE プログラムに転送する．クライアントプログラムと接続しているサーバプログラムがソケットペアの情報を基

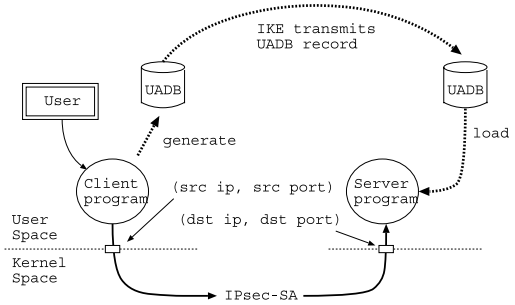

図 5 サーバプログラムによるソケットペアとユーザ識別子の対応付け
Fig. 5 Mapping socket pairs with a user identifier by a server program.

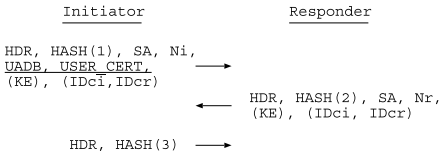

図 6 変更後の IKE プロトコルのクイックモード
Fig. 6 Modified IKE's Quick mode.

に UADB を検索し，該当する UADB レコードを見つけた場合，サーバプログラムはそのソケットペアに対応するユーザ識別子を見つけることができる．UADB レコードにおける署名の正当性が確認できれば，サーバプログラムはソケットペアとユーザ識別子を対応付けることができる．

3.3 IKE プロトコルのフェーズ 2 における ISAKMP ペイロードの追加

提案方式は IKE プロトコルのフェーズ 2 で UADB レコードを転送する．フェーズ 2 はクイックモードと呼ばれる IPsec-SA のパラメータのネゴシエーションと鍵素材の生成を行う方式で動作する[2]．本論文の実装はクイックモードの第 1 メッセージに ISAKMP ペイロードを 2 つ追加することで UADB レコードとユーザの公開鍵証明書の転送，およびユーザ認証を行う．変更後の IKE プロトコルのクイックモードを**図 6** に示す．下線の “UADB” と “USER_CERT” が追加された ISAKMP ペイロードである．“UADB” は UADB レコードをテキスト形式で転送するのに使用する．“USER_CERT” はユーザの公開鍵証明書を転送するのに使用する．IKE プロトコルは ISAKMP のフレームワークを利用してネゴシエーションを行う．よって本論文の提案方式を実装するためには，IETF IPsec DOI[8] に 2 つのペイロードを追加する必要がある．

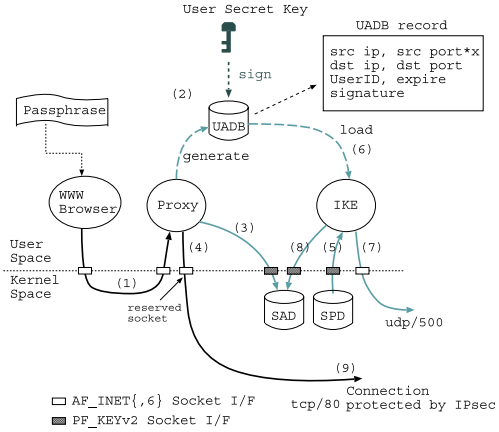

図 7 クライアントホストの処理
Fig. 7 Processes on a client host.

4. 設　　計

本章では 3 章で提案した UADB 方式を既存の IPsec システムに導入する．そして IPsec と IKE をアプリケーションプログラムの暗号化通信路と認証に利用する．本論文は実装対象のアプリケーションを WWW とした．よってクライアントプログラムはウェブブラウザであり，サーバプログラムは HTTP デーモンである．実装は IPv6 ネットワークで動作するように設計した．

4.1 初期設定

ユーザはホームディレクトリにユーザの秘密鍵，ユーザの公開鍵証明書，設定ファイルを用意する．設定ファイルには本実装で接続するホストの IP アドレスを IPv4 か IPv6 形式で記述する．両ホストの管理者はあらかじめ管理プログラムでポリシを設定する．サーバホストにはユーザの公開鍵証明書を発行した認証局（CA: Certification Authority）の公開鍵証明書を用意する．

4.2 クライアントホストの処理

図 7 にクライアントホストの処理を示す．処理の流れは次のようになる．

（1） ウェブブラウザはプロキシ経由で HTTP デーモンへ接続を試みる．

（2） プロキシは UNIX のユーザ ID または公開鍵証明書の識別名と終点 IP アドレス，終点ポート番号をキーとして UADB を検索し，該当する UADB レコードがない場合には UADB レコードの生成を開始する．プロキシは認証ページを表示し，ユーザにパスフレーズを要求する．プロキシはユーザの秘密鍵をロードしてパスフ

レーズで復号する．次にプロキシは 5.1 節で説明する方法でソケットを予約し，ユーザの秘密鍵で UADB レコードに電子署名を行い，完成した UADB レコードを UADB に保存する．

（3） プロキシはすでに確立している IPsec-SA（同じソケットペアの IPsec-SA）を SAD から削除する．

（4） プロキシは HTTP デーモンに対し予約したソケットを用いて接続する．この段階で IPsec-SA が確立していないので接続はブロックする．

（5） カーネルの鍵/ポリシ管理部は，IPsec-SA が存在しないので IKE プログラムにメッセージを送り鍵交換を要求する．

（6） IKE プログラムは UADB から該当するレコードをロードする．

（7） IKE プログラムはネゴシエーションを開始する．最初に IKE プロトコルのフェーズ 1 でホスト認証を行い IKE-SA を確立する．次に UADB 方式を実装した IKE プログラムはフェーズ 2 で UADB レコードを転送する．

（8） IKE プログラムは IPsec-SA を SAD に保存する．

（9） ブロックしていた HTTP デーモンへの接続が確立する．この接続は IPsec-SA により保護される．

（10） クライアントホストは有効期限の切れた UADB レコードを削除する．また予約ソケットを使い果たした UADB レコードも同様に削除する．

4.3 サーバホストの処理

図 8 にサーバホストの処理を示す．処理の流れは次のようになる．

（1） サーバホストの IKE プログラムは，クライアントホストの IKE プログラムから送信されたメッセージを受信する．

（2） IKE プログラムはフェーズ 1 でホスト認証を行い IKE-SA を確立する．次に IKE プログラムはフェーズ 2 でクライアントホストの IKE プログラムから送信された UADB レコードとユーザの公開鍵証明書を受信する．そして認証局の公開鍵証明書でユーザの公開鍵証明書を検証する．続いて UADB レコードの電子署名をユーザの公開鍵で検証することでユーザ認証を行う．認証に成功した場合，IKE プログラムは受信した UADB レコードの有効期限と始点ポート番号の数が制限値を超えていないか検査してから UADB に保存する．認証に失敗した場合，IKE プログラムはネゴシエーションを終了する．

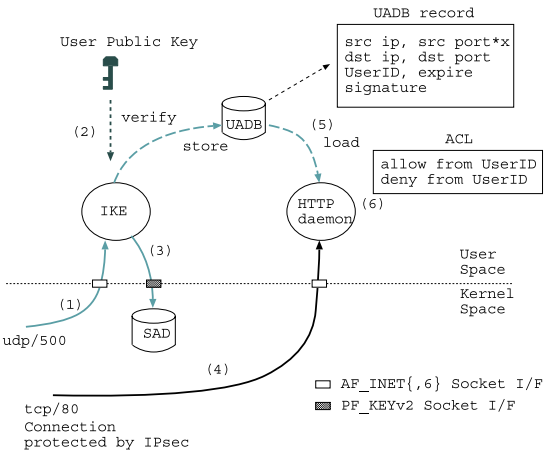

図 8 サーバホストの処理
Fig. 8 Processes on a server host.

```
order deny,allow
deny from all
allow from {bob,alice}
```

図 9 ユーザ識別子によるアクセス制御リスト
Fig. 9 Access control list using user identifier.

（3） IKE プログラムは IPsec-SA を確立する．IKE プログラムは IPsec-SA を SAD に保存する．

（4） IPsec-SA が確立した直後に，クライアントホストのプロキシから HTTP デーモンに対する接続が確立する．この接続は IPsec-SA により保護される．

（5） HTTP デーモンは接続しているソケットを検査してソケットペアを得る．そしてソケットペアをキーとして UADB レコードを検索する．HTTP デーモンは UADB レコードを参照することでユーザ識別子を特定する．

（6） HTTP デーモンはアクセス制御リストをロードする．そしてユーザ識別子を用いて URL に対するアクセス制御を行う．**図 9** にアクセス制御リストの例を示す．

（7） サーバホストは UADB レコードの有効期限が切れた時点でセッションを破棄する．また UADB レコードが予約ソケットを使い果たしたときもセッションを破棄する．

5. 実　　装

4 章で設計した UADB 方式を WWW に適用するために大きく分けて以下の 3 つの実装を行った．1 つ

表 2 実装に使用したソフトウェア
Table 2 Softwares used with the implementation.

OS	FreeBSD 4.3 Release
IPv6/IPsec スタック	KAME [9]
IKE プログラム	racoon-20011026a
プロキシプログラム	tinyproxy-1.3.3b
HTTP デーモン	apache_1.3.20, IPv6 patch
ウェブブラウザ	Mozilla 0.8.1（Gecko/20010419）
SSL ライブラリ	OpenSSL 0.9.6

目はウェブブラウザの wrapper として動作するプロキシの実装である．プロキシはユーザへのパスフレーズ要求と UADB レコードの生成処理を行う．2 つ目は IKE プログラムの改造である．UADB レコードとユーザの公開鍵証明書を転送するために新たに定義した ISAKMP ペイロードの送受信と，受信側でのユーザ認証の処理を実装した．3 つ目は UADB レコードを基にユーザ単位で URL のアクセス制御を行う HTTP デーモンモジュールの実装である．

本論文のプロトタイプ実装ではカーネルのソースコードに対する変更を行っていない．実装に使用したソフトウェアの一覧を**表 2** に示す．

5.1 始点ポート番号の問題とソケットの予約による解決策

本方式で用いる UADB レコードは複数の始点ポート番号を保持することを 3.1 節で述べた．IKE プログラムはサーバプログラムがソケットペアからユーザの識別子を特定するために，UADB レコードをあらかじめサーバホストに転送しておく必要がある．通常クライアントプログラムが UNIX の connect システムコールを呼び出すとカーネルは自動的に選択した始点ポート番号を使用する．したがって IKE プログラムはあらかじめ始点ポート番号を記述した UADB レコードをサーバホストに転送できない．

本論文で示すプロトタイプ実装はこの問題を解決するために予約ソケット管理を行うプロキシを実装する．**図 10** に処理の流れを示す．

まずクライアントプログラムはプロキシに接続する．プロキシは UNIX のユーザ ID または公開鍵証明書の識別名と終点 IP アドレス，終点ポート番号をキーとして UADB を検索し該当する UADB レコードがなければ，新たに生成する処理を開始する．プロキシはあらかじめ設定した個数のソケットを生成し bind システムコールを用いてランダムに選んだ始点ポート番号を結び付ける．そして予約したソケット識別子を後で使えるようにメモリに記憶しておく．プロキシは予約したソケットのソケットペアを基に UADB レコードを生成し UADB に格納する．その後 IKE プログラムが UADB レコードをサーバホストに転送し，IPsec-SA を確立する．クライアントプログラムはプロキシ経由でサーバホストへ接続する．プロキシは予約したソケットを使ってサーバホストに接続する．以上の処理によりサーバプログラム側で得たソケットペアと UADB レコードのソケットペアが一致する．したがって IKE プログラムはあらかじめ始点ポート番号を記述した UADB レコードをサーバホストに転送できるようになる．

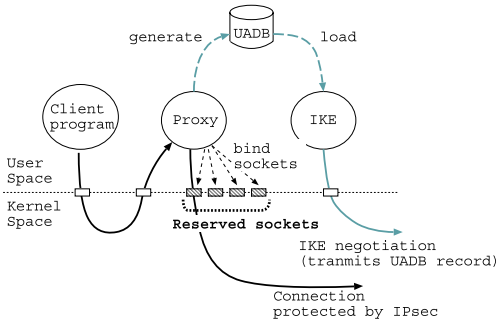

図 10 予約ソケット管理
Fig. 10 Management of reserved sockets.

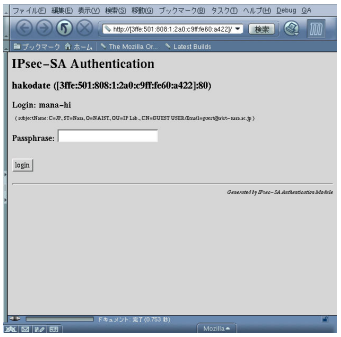

図 11 認証ページ
Fig. 11 Authentication page.

5.2 クライアントホストにおける実装

プロキシのパスフレーズの要求，UADB レコードの生成，UADB に対する入出力，予約ソケット管理および SAD の操作は新たに実装したものである．プロキシが表示する認証ページを**図 11** に示す．IKE プログラムの UADB の読み込みとフェーズ 2 における転送の機能は新たに実装したものである．

5.3 サーバホストにおける実装

IKE プログラムにおける UADB レコードとユーザの公開鍵証明書の受信，ユーザの公開鍵証明書の検証，UADB レコードの電子署名の検証，UADB レコードの有効期限と始点ポート番号の数の検査，UADB に対する入出力処理は新たに実装したものである．また

表 3 ハードウェア仕様
Table 3 Hardware specification.

	クライアントホスト	サーバホスト
CPU	Pentium II 333 MHz	Pentium 233 MHz
Memory	256 MB DRAM	96 MB DRAM
Network	100BaseT	100BaseT

表 4 プロキシの処理時間
Table 4 Process time of proxy.

処理	時間 [ms]
UADB レコードの生成	28
予約したソケットの検索	19

表 5 IKE プログラムのフェーズ 2 の処理時間
Table 5 Process time of Phase2 in IKE program.

処理	時間 [ms]
変更前	356
変更後	676

HTTP デーモンにおけるソケットペアからユーザ識別子を得る処理とユーザ識別子を用いたアクセス制御の処理は新たに追加したものである．

6. 性能測定

実装したプログラムの性能測定を行った．性能測定に使用した計算機のハードウェア仕様を**表 3** に示す．2 台の計算機は IPv6 で構築したネットワークの同一セグメントに配置して実験を行った．電子署名には 1024 bit の RSA 公開鍵を使用した．

6.1 実験結果

プロキシの処理時間を**表 4** に示す．UADB レコードの生成処理はユーザ秘密鍵のロードと復号，ソケットの予約，電子署名の生成，UADB の保存処理から構成される．予約したソケットの検索処理は UADB レコードのロード，予約したソケットの検索，UADB の更新から構成される．

IKE プログラムのフェーズ 2 の処理時間を**表 5** に示す．送信側で追加した処理は UADB レコードとユーザの公開鍵証明書のロード，ISAKMP ペイロードの生成処理である．受信側で追加した処理はユーザの公開鍵証明書の検証，UADB レコードの検証，UADB レコードの保存処理である．

HTTP デーモンがソケットペアとユーザ識別子を対応付ける処理時間を**表 6** に示す．この処理は HTTP デーモンが accept 関数で処理するソケットを getpeername システムコールと getsockname システムコールで検査する処理，UADB レコードのロードと有効期限の検査，ユーザ識別子の検索，UADB の更新処理である．

表 6 ソケットペアとユーザ識別子を対応付ける処理時間
Table 6 Process time of mapping socket pairs with a user identifier.

処理	時間 [ms]
ユーザ識別子の算出	120

6.2 従来方式との比較

従来方式とは IKE プロトコルのフェーズ 1 でホスト認証を行って IKE-SA を確立しフェーズ 2 で IPsec-SA を確立する方式である．提案方式を実現するための変更はプロキシによる UADB レコードの生成と予約したソケットの検索，IKE プロトコルのフェーズ 2 による UADB レコードの転送と電子署名の検証，HTTP デーモンによるソケットペアからユーザ識別子を算出する処理の追加である．本節は提案方式の変更点が従来方式と比べてどれくらい処理速度に影響するかを各処理時間を基に比較する．

まず IPsec-SA が確立するまでの処理時間を従来方式と比較する．提案方式の処理が要する時間はプロキシによる UADB レコードの生成時間 28 ms と予約したソケットの検索時間 19 ms，IKE プロトコルのフェーズ 2 の処理時間の増加分の 320 ms である．ゆえに従来方式と比べると 367 ms の余分な時間を要する．

次に IPsec-SA が確立してからの処理時間を従来方式と比較する．提案方式では HTTP デーモンがソケットペアからユーザ識別子を算出する処理を追加した．ゆえに従来方式と比べると accept 関数が実行されるごとに 120 ms の余分な時間を要する．

IPsec-SA が確立して最初の接続が確立するまでに提案方式が要する処理時間は 487 ms である．IPsec-SA が一度確立したあとに提案方式の HTTP デーモンが要する処理時間は 120 ms である．

7. 証明書を用いた認証処理における TLS との比較

IKE はホストの公開鍵証明書と秘密鍵を両方のホスト管理者が用意する必要がある．さらに提案方式はユーザごとに公開鍵証明書と秘密鍵を用意する必要がある．TLS はクライアントの公開鍵証明書と秘密鍵およびサーバの公開鍵証明書と秘密鍵を用意する必要がある．

提案方式は IKE が相互にネットワーク層でホスト認証を行った後に，トランスポート層におけるソケットペアとユーザの識別子を関連付けることでユーザ認証を行う．TLS はトランスポート層の上位に実装されたハンドシェイクプロトコルによりクライアントとサーバの認証を行う．

SYNOPSYS
*char** **get_uadb_user_id**(*int* sock, *int* idtype)
DESCRIPTION
accept 関数で取得したソケット sock を引数として実行すると UADB を検索してユーザ識別子の文字列を返す．UADB レコードから使用した始点ポート番号を削除して UADB を更新する．idtype で UNIX のユーザ ID か公開鍵証明書の識別名のどちらを使うか指定する．
RETURN
ユーザ識別子の文字列を返す．認証されたユーザでない場合は-1 を返す．

図 12 サーバプログラムで使用する API
Fig. 12 API for server programs.

認証処理における提案方式と TLS の違いの 1 つは認証エンティティである．提案方式は 2 段階の認証処理を行う．第 1 段階の認証エンティティは各ホストである．第 2 段階の認証エンティティはホストに接続を試みるクライアントである．対して TLS の認証エンティティはサーバとクライアントである．

8. API の提案

本論文は提案方式を WWW に適用する実装を示した．提案方式は WWW 以外のサーバプログラムに導入することが可能であるがサーバプログラムの改造を必要とする．そこで本論文では開発者が提案方式を実装するのを容易にするためにサーバプログラム向けの API を提案する．**図 12** に提案する API を示す．本論文で示したプロトタイプ実装を WWW 以外のアプリケーションに応用するためには以下の作業が必要である．(1) プロキシを WWW 以外のアプリケーションプロトコルに対応させる．(2) 提案した API をサーバプログラムに実装する．提案した API を用いることでサーバプログラムがソケット識別子からユーザを特定することができ，アプリケーションは各々のアクセス制御に利用することが可能になる．

9. 汎 用 性

提案方式はソケット識別子とユーザ識別子を対応付けるという単純な機構である．よってソケット API の処理の後に提案した API を実装するという汎用的な方法によって，WWW 以外の TCP/IP アプリケーションに提案方式を適用可能である．本論文のプロトタイプ実装は FreeBSD オペレーティングシステムで行った．IPv6/IPsec プロトコルスタックは KAME を，IKE デーモンは racoon を利用した．5 章で述べたようにプロトタイプ実装はカーネルの IPv6/IPsec プロトコルスタックに対する変更を加えていない．また 3.3 節で述べたように IKE プログラムに対する変更は ISAKMP のフレームワークを利用して行っている．よって他のオペレーティングシステムに実装された IKE プログラムに提案方式を適用することが可能である．

10. 今後の課題

提案方式はクリティカルな情報が UADB に集中するため，管理方法の安全性を検討する必要がある．本論文の実装は UADB をファイルとして管理する．よってファイルのパーミションによる保護機能に従い，管理者権限で動作している他のプロセスによる読み込み，複製が可能である．しかし UADB をカーネル内部に実装するとデータベース読み込みのためのシステムコールを隠蔽することができる．よって管理者権限で動作している他のプロセスから保護しやすい構造になると考えられる．この場合 UADB へのデータベース操作には PF_KEYv2 ソケットインタフェースを改良したものを用いる方法が考えられる．またプロキシを用いた予約ソケット管理方式のスケーラビリティの問題がある．この問題はオペレーティングシステムが作成できるソケット数に制限があるために生じる．対策としてはプロキシの機能をカーネルの API として実装し，作成できるソケット数の制限を吸収する中間層をソケット処理部に実装する方法が考えられる．

11. 関 連 研 究

ユーザエンティティとしてのクライアントをサーバが認証するシステムでアプリケーションプロトコルと独立した鍵交換プロトコルを使うシステムには Kerberos [10] および Microsoft Windows2000 以降の認証システムがある．これらのシステムと本論文で提案するユーザ認証情報を伝達する IKE との違いは，アプリケーションレベルのリソース識別子を用いるか，TCP/IP のソケット識別子を用いるかの違いである．本論文で提案したシステムをアプリケーションで利用するためには UADB のための API を開発するだけでよいという利点がある．逆にファイル入出力の権限の割当てのようなより抽象化されたエンティティには適用できない．

12. む す び

本論文の提案方式は IPsec と IKE の暗号化通信路と認証の機能をアプリケーションで利用するための枠組みである．本論文は提案方式を WWW に適用する実装を示すことでアプリケーションに独自の暗号化通信

路および認証の機能を実装しなくても IP 層の強力なセキュリティ機構を利用できることを示した．またプロトタイプ実装の性能測定を行った．最後に提案方式をアプリケーションで利用するための API を示した．

参 考 文 献

1) Kent, S. and Atkinson, R.: RFC 2401: Security Architecture for the Internet Protocol (1998).
2) Harkins, D. and Carrel, D.: RFC 2409: The Internet Key Exchange (IKE) (1998).
3) Gleeson, B., Lin, A., Heinanen, J., Armitage, G. and Malis, A.: RFC 2764: A Framework for IP Based Virtual Private Networks (2000).
4) Kent, S. and Atkinson, R.: RFC 2402: IP Authentication Header (1998).
5) Kent, S. and Atkinson, R.: RFC 2406: IP Encapsulating Security Payload (ESP) (1998).
6) Dierks, T. and Allen, C.: RFC 2246: The TLS Protocol Version 1 (1999).
7) Housley, R., Ford, W., Polk, W. and Solo, D.: RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile (1999).
8) Piper, D.: RFC 2407: The Internet IP Security Domain of Interpretation for ISAKMP (1998).
9) Jinmei, T., Yamamoto, K., Hagino, J., Sumikawa, M., Inoue, Y., Sugyo, K. and Sakane, S.: An Overview of the KAME Network Software: Design and implementation of the advanced internetworking platform, *INET'99* (1999).
10) Kohl, J. and Neuman, C.: RFC 1510: The Kerberos Network Authentication Service (V5) (1993).

(平成 14 年 9 月 6 日受付)
(平成 15 年 6 月 3 日採録)

推 薦 文

IPsec および IKE は，そのままではユーザ単位のアクセス制御に用いることができない．そこで，ユーザ認証データベース（UADB）を導入し，また IKE フェイス 2 に，UADB レコードをサーバ側に転送する機能の追加およびいくつかのプログラムの開発・改造を行うことで，サーバアプリケーションにおけるユーザのアクセス制御として利用できる方式を提案した．提案方式は必要性が高く，また実装をともなっている点で推薦に値する．

（平成 13 年度情報処理学会九州支部長　牧之内顕文）

平野　学

平成 10 年旭川工業高等専門学校電気工学科卒業．同年電気学会北海道支部長賞．平成 12 年豊橋技術科学大学工学部情報工学課程卒業．平成 14 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．IPsec と IKE を用いたアクセス制御技術の研究に従事．同年東芝（株）入社．

木村　泰司

平成 9 年芝浦工業大学システム工学部環境システム学科卒業．平成 11 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．同大学院同研究科博士後期課程在学中．インターネットアーキテクチャ，コンピュータセキュリティの研究に従事．

山口　英（正会員）

昭和 61 年大阪大学基礎工学部情報工学科卒業．昭和 63 年同大学大学院博士前期課程修了．平成 2 年同後期課程を退学し，同大学情報処理教育センター助手に着任．平成 4 年奈良先端科学技術大学院大学情報科学センター助手，同助教授を経て，平成 5 年同大学情報科学研究科助教授．平成 12 年同大学情報科学研究科教授．工学博士．インターネットアーキテクチャ，コンピュータセキュリティの研究に従事．