5A-8

組込システム向け TCP/IP プロトコルスタックにおける IPsec の実装

堤大祐、堀武司、長内研、吉川毅、山本寧

北海道立工業試験場

1. はじめに

近年、インターネットに接続できる家電製品や監視装置などの組込みシステムが多くなってきた。

このようなインターネットに接続できる組込みシステムの開発において、TOPPERS プロジェクト[1]からリアルタイム OS として TOPPERS/JSP カーネル、組込みシステム用 TCP/IP プロトコルスタックとして TINET[6]がオープンなソフトウェアとして公開されている。TOPPERS/JSP カーネルはμITRON4.0 仕様[2]、TINET は ITRON TCP/IP API の仕様[3]にそれぞれ準拠している。開発環境においても GCC(Gnu Compiler Collection)が使用可能で、インターネットに接続できる組込みシステム開発が身近になってきた。

組込みシステムがインターネットに接続するようになると、通信の安全性や IPv6 に関する要求が高まってくると思われる。IPv6 においては IPsec(Security Architecture for Internet Protocol)[4]の実装が必須となっている。TINET は IPv4 と IPv6 の両方に対応したプロトコルスタックであるが、IPsec の実装は行われていない。

現状、通信の安全性において PC では SSL(Secure Socket Layer)、基幹ネットワークではルータ間に IPsec が用いられている。SSL はアプリケーション層で、IPsec はネットワーク層で動作する。

今回、上位層に依存せず安全な通信路を提供できる IPsec をオープンなソフトウェアとして広く普及が期待できる TINET 上で実装した。IPsec の実装において以下の点に留意して行った。

- TINET の構造に対して大幅な変更は行わない。
- データのコピーは行わない。

評価にはルネサステクノロジ製 H8s(H8s2638 動作周波数 20MHz)と SH2(SH7615 動作周波数 14.7546MHz、4 倍で動作)の 2 種類の CPU を使用した。

2. IPsec の実装

IPsec はネットワーク層において、暗号技術や認証技術を用いて安全な通信を提供するものである。これにより、TCP や HTTP などより上位層において安全な通信ができる。IPsec は IPv6 において必須で、IPv4 ではオプションである。図 1 に OSI 階層モデルにおけるプロトコルの例と IPsec の位置付けを示す。

IPsec には ESP (Encapsulating Security Payload) と AH (Authentication Header) の 2 つのセキュリティプロトコルがあり、主に、ESP は暗号によるデータの機密性、AH は改竄防止・送信元の認証を行う。暗号アルゴリズムは DES(Data Encryption Standard)や AES(Advanced Encryption Standard)などがあり、認証アルゴリズムは MD5(Message Digest Five)や SHA-1(Secure Hash Algorithm)などがある。56bit DES と MD2/4/5 は実装が必須となっている。暗号処理に必要な鍵は鍵交換プロトコル IKE(Internet Key Exchange)で行う。IPsec にはトンネルモードとトランスポートモードの 2 つのカプセル化モードがあり、トンネルモードは主にセキュリティゲートウェイ間で用いられ、トランスポートモードはホスト間で用いられる。

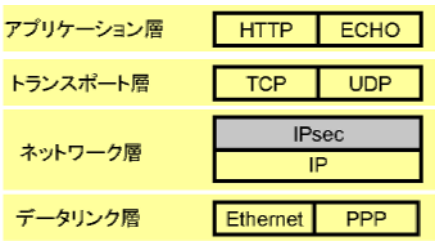

図 1　OSI 階層モデルにおける IPsec の位置付け

今回、ソフトウェアによる小型組込みシステム向けの IPsec の有効性を評価するため、情報家電向け IPv6 最小要求仕様案[5]を参考に実装する機能を選択した。選択した機能を表 1 に示す。セキュリティプロトコルは ESP を実装した。その際に必要な暗号アルゴリズムは AES を使用した。IPsec の仕様では DES が必須ではあるが、より新しい暗号アルゴリズムである AES を実装した。鍵交換においては IKE を使用せず手動で行った。カプセル化モードは小型組込みシステムではルータとして機能することは少ないと判断し、ホスト間通信を行うトランスポートモードを実装した。また、広く普及しており、試験が容易な IPv4 とした。

セキュリティプロトコル	ESP(認証機能未実装)
暗号アルゴリズム	AES-CBC(鍵長 128bit)
カプセル化モード	トランスポートモード
その他	IPv4、手動鍵交換

表 1　選択した機能

ESP を適用すると、IP パケットは図 2 のように ESP ヘッダが IP ヘッダと IP データの間に挿入され、ESP トレーラが IP データの後に付加される。ESP ヘッダは識別子など、ESP トレーラは元の IP プロトコル番号などから構成される。

IP パケットの受信は IP プロトコル番号に応じた処理を行う。ESP パケットであれば、IPsec ルールと照合し復号

Implementation of IPsec Module for Embedded TCP/IP Protocol Stack
Tsutsumi Daisuke, Hori Takeshi, Osanai Ken, Kikkawa Takeshi, Yamamoto Yasushi (HIRI)

化する。復号化後、元のプロトコル番号に応じた処理を行う。IP パケットの送信は IPsec ルールと照合し、IPsec を適用するパケットであればパケットを暗号化して送信処理を行う。

送信する際、図２のように ESP ヘッダが割り込む。TINET のネットワーク用バッファは固定長であらかじめ下位層のヘッダ領域を確保し、メモリの動的操作を行わない構成となっている。しかし、ESP ヘッダをあらかじめ割り当てておくことは IPsec の適用の有無により、バッファの操作が異なる。そこで、ESP 適用後の出力バッファを新たに設けて暗号処理をしながら出力バッファに出力した。これにより、新たにバッファを確保するメモリが必要になるが、実質的に TINET の構造を変更せずコピーなしで ESP ヘッダを割り込ませた。

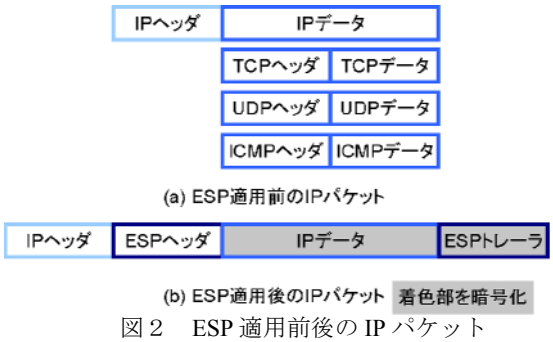

図２　ESP 適用前後の IP パケット

３．評価方法

暗号処理に要する時間を評価するため、PC からマイコンボードに ping でメッセージを送信し、その応答を受信するまでに要する時間（応答時間）を測定した。測定は IPsec を使用した場合と使用しない場合でそれぞれ 1000 回行い、応答時間の平均値を求めた。平均値は ping プログラムのレポートによる値を用いた。PC には FreeBSD4.11 を使用した。PC とマイコンボードはスイッチングハブを経由して接続した。ping の送受信データは ICMP ヘッダとデータをあわせて 64 バイトである。

プログラムサイズは開発環境が作成するレポートファイルから算出した。開発環境は Linux2.6 上で SH2 が gcc-2.95.3、binutils-2.14、H8s は gcc-3.2.3、binutils-2.14 をそれぞれ使用した。

IP パケットに対してどのように IPsec を適用するかを定めた IPsec のルールとなる SPD (Security Policy Database)、SAD (Security Association Database)は固定とし、ICMP パケットのみ ESP を適用した。暗号アルゴリズムは AES-CBC（128 ビット）、鍵交換は手動で行った。

４．評価結果

SH2 において IPsec を実装した場合の増加したプログラムサイズを表２に示す。AES の暗号処理に必要な RAM 容量がわずかなのは、呼び出し側の IPsec 部分で領域を確保しているためである。ROM の多くは暗号処理に使用する様々な定数である。カーネルやプロトコルスタック、echo サーバを含めたプログラム全体では IPsec の実装により ROM が約 30%、RAM が約 5%使用量が増加した。

	RAM	ROM
IPsec 部分	3280	39208
(AES 暗号処理部)	(44)	(24360)
カーネル、プロトコルスタック、echo サーバ	69914	166050

単位：バイト

表２　プログラムの大きさ

ping による応答時間の平均値と暗号処理に要した時間を表３に示す。暗号処理に要した時間は「暗号処理ありの応答時間」と「暗号処理なしの応答時間」との差から求め、メッセージの復号化および暗号化に要した時間を示している。その結果、SH2 で 5.2ms、H8s で 18.6ms となった。メッセージの長さは 64 バイトである。これらの値にはリアルタイム OS のオーバーヘッドが含まれている。

	SH2	H8S
暗号処理ありの応答時間(1)	6.7	22.4
暗号処理なしの応答時間(2)	1.5	3.8
暗号処理に要した時間(1)－(2)	5.2	18.6

単位：ミリ秒、ping メッセージ 56 バイト:IP データグラムとして 64 バイト(デフォルト)

表３　ping の応答時間と暗号処理に要した時間

５．おわりに

μITRON 仕様に準拠した組込みシステム開発において、リアルタイム OS と TCP/IP プロトコルスタック上で、IPsec を最小構成で実装した。その結果、暗号処理時間が許容できれば、約 40 キロバイトの ROM の追加で IPsec を小型組込みシステムに適用できる。

実装していない仕様のうち、AH、ESP のオプション、鍵交換の対応は必須と考えている。さらに、その他の暗号アルゴリズム、認証アルゴリズムの実装を通して、組込みシステム向け IPsec として完成度を高めていきたい。また、TOPPERS プロジェクトを通してオープンなソフトウェアとして公開し、利用者側の意見を取り入れながら開発を進めていきたい。

参考文献

[1] http://www.toppers.jp/
[2] 坂村健(監修)、高田広章(編)：μITRON4.0 仕様 4.02.00 、トロン協会(2004)
[3] 高田広章(編)：ITRON TCP/IP API 仕様 1.00.01、トロン協会(1998)
[4] S. Kent, R. Atkinson, “Security Architecture for the Internet Protocol”, RFC 2401, November, 1998
[5] http ://www.tahi.org/lcna/docs/IPv6-min-spec/IPv6-min-spec-ver42.htm
[6] 阿部司、吉村斎、久保洋：組込みシステム用 TCP/IP プロトコルスタックの実装と評価、情報処理学会論文誌、Vol.44, pp.1583-1592(2003)