組込みシステム向けIPsecの実装と評価

堀　武司，堤　大祐，吉川　毅，山本　寧

Implementation and Evaluation of IPsec for Embedded System

Takeshi HORI, Daisuke TSUTSUMI, Takeshi KIKKAWA, Yasushi YAMAMOTO

抄　録

近年のインターネットの普及などにより，組込みシステムにおいてもTCP/IPによるネットワーク接続機能のニーズが高まっている。それに伴い，組込みシステム上でもネットワークセキュリティが重要な課題となる事が予想される。特に，組込みシステムが扱う情報には，個人のプライバシや機械装置等の管理に関する情報が含まれる場合が多く，これらの通信の保護は重要な課題である。我々は，リソース制約の厳しい小規模な組込みシステムをターゲットとして，μITRON OS上で動作するTCP/IPプロトコルスタック TINETに対して，IPsec 規格 に基づくセキュリティプロトコルの実装を試みた。数種類のマイコンによる試験の結果，扱うデータ量が大量とならない場合には，ソフトウェア暗号処理でも実用的な通信が可能である事が確認された。また，メモリ使用量の増加は従来の30%程度であり，リソース制約の厳しい環境でも適用可能なコンパクトな実装が実現出来た。

キーワード：組込みシステム，TCP/IP，ネットワークセキュリティ，IPsec

Abstract

As the needs for TCP/IP network on embedded systems are increasing, network security problem becomes important. Embedded network systems often handle sensitive information such as personal privacy or control signal of machinery, which should be protected by a proper security protocol.

We developed an implementation of IPsec security protocol for TINET TCP/IP protocol stack, which runs on μITRON real-time operating system. In our evaluation on several embedded CPUs, it is proved that our IPsec protocol stack without hardware cipher engines had sufficient performance on small network traffic. The memory usage of the TINET with IPsec extension is only about 30% larger than that of the original TINET, so that it is usable on embedded systems with severe resource restrictions.

KEY-WORDS : TCP/IP, Embedded Systems, Network Security, IPsec

１．はじめに

近年のインターネットの普及などにより，PCやワークステーションなどの汎用コンピュータに限らず，家電製品や産業機器などの組込みコンピュータシステムにおいてもネットワークによる外界との連携が重要な機能要素の一つとなってきており，そのための基盤技術であるTCP/IPによる通信機能のニーズが高まっている。また，総務省の「u-Japan 戦略」(平成16年～）では，あらゆる電子機器がネットワークに接続され，あらゆる場所で情報サービスが利用可能な「ユビキタスネットワーク社会」の実現を目標に掲げており，ネットワークに対応した組込みシステムの需要は，今後更に拡大

事業名：重点領域特別研究
課題名：組込みシステム向けネットワーク接続ソフトウェア群の開発

していくと予想される。

組込みシステムのネットワークで重要となる課題の一つとして，セキュリティの確保の問題がある。インターネット上などでのセキュリティ確保の問題は既に広く社会に認知されているが，組込みシステムにおいてもネットワーク対応製品の増加につれてセキュリティが重要な課題となる事が予想される。特に，ユビキタスネットワーク的なアプリケーションで扱われる情報には，個人のプライバシ等を含む情報（例：家庭用防犯システムなど），機械装置等の制御や管理に必要な情報（例：ロードヒーティング制御のための気象センサなど）などが多くなる事が予想される。これらの情報をインターネットや無線ネットワークでやりとりする場合には，暗号化などによる通信保護は必須の機能である。

筆者らは，組込みシステムの中でも主にリソース制約の厳しい小規模な組込みシステムをターゲットとし，オープンソースソフトウェアとして開発が進められているμITRON 仕様OS（TOPPERS/JSPカーネル)およびTCP/IPプロトコルスタック（TINET）の二つのソフトウェアを技術基盤として活用し，組込みシステムのネットワーク対応に関して取り組んできた。現在のTINETは，TCP/IPの基本機能およびWWWなど一部の応用層機能を含んでいるが，通信の暗号化などセキュリティ機能への対応はまだ進められていない。そこで本研究では，TINETを基盤として，組込みネットワークアプリケーションに必要となるセキュリティ関連機能の検討と，それらを実現するためのソフトウェア開発を行った。

２．セキュリティ規格の検討

TCP/IPネットワーク上でセキュアな通信を実現するための技術は様々なものが存在し，用途に応じて使い分けがなされている。そこで，現在用いられている主要なセキュリティ技術を比較し，小規模組込みシステムでの利用に適した技術の検討を行った。

2.1 SSL（Secure Socket Layer)

SSL（Secure Socket Layer)，TLS（Transport Layer Security）は，トランスポート層で通信保護機能を提供するプロトコルである。現在のインターネットでは，Web サーバとブラウザとの間の通信保護のための技術（HTTP over SSL）として広く用いられている。

SSL/TLSはHTTPに限らずFTP(ファイル転送)やPOP(電子メール受信）など，TCP（Transmission Control Protocol）を使用する全ての通信で利用可能である。しかし，トランスポート層としてTCPではなくUDP(User Datagram Protocol)を用いる通信の保護には利用出来ない。

2.2 SSH（Secure Shell)

SSH（Secure Shell）は，遠隔ホストに対する操作（遠隔ログイン，コマンド実行，ファイル転送など）を暗号化通信によって実現する技術である。UNIX系ＯＳのrsh，rloginコマンドなどの代替として開発され，現在では遠隔ホストへのリモートログイン手段の標準として用いられている。

SSHはトンネル機能を有しており，SSHで保護された通信路を介して他のアプリケーションの通信を中継するためにも用いられる。

2.3. IPsec

IPsec は，ネットワーク層（TCP/IP におけるIP層）において通信保護などの機能を実現するプロトコルである。そのため，IP層の上位に実装されたプロトコル（TCP，UDP，ICMP等）は，IPsecによって区別無く統一的に取り扱う事が可能である。また，IPsecに関する処理はIP層で透過的に行われるため，アプリケーションプログラム側でIPsecを意識せずに導入できるメリットがある。

IPsec は，現行のインターネットプロトコル（IPv4）ではオプション機能であるが，次世代のIP v6では必須機能と指定されており，標準セキュリティプロトコルとして位置づけられている。

2.4 対応するセキュリティ規格の選択

これらの技術の長所，短所を比較検討した結果，本研究では，実装対象とするセキュリティ規格としてIPsec を選択した。その理由を以下に示す。

- IPsecはIP層で全ての処理を行うため，幅広い応用層プロトコルをカバーする機能を効率的に提供可能である事。
- 組込みシステムではTCP以外の通信を扱う場合が多いため，TCPのみにしか適用出来ないSSLは用途が限定され効率が悪い事。

３．IPsecの概要

IPsecを構成する技術要素の基本概念について，簡単に説明する（図１)。

セキュリティポリシー（SP）は，通信に対してIPsecを適用する際のルールを定義する。SPでは，IPアドレスやポート番号で指定したパケットの条件毎に，Discard(パケット破棄)，Bypass(IPsecを適用しない)，Apply（IPsecを適用する)などのルールを指定する。

セキュリティアソシエーション(SA)は，IPsecで保護された仮想的な通信路である。SAは，終点アドレス，セキュリティプロトコル，SPI(インデックス)によって一意に識別され，SAD(SAデータベース)で管理される。SAには適用す

る暗号の種類や暗号鍵などの情報が登録されており，それを用いてパケットの暗号，復号処理が行われる。

IPsecによる通信を行う場合は，通信相手との間にSAを確立しなければならない。SAの設定は，手作業で行う場合もあるが，一般的にはIKE(Internet Key Exchange)[5]などを用いて自動的に行う場合が多い。

IPsecで用いられるセキュリティプロトコルには，ESP (Encapsulating Security Payload) とAH (Authentication Header)の2種類が存在する。ESPは，IPパケットの暗号化と認証（改竄防止）を行うプロトコルである。一方，AHはパケットの暗号化は行わず認証機能のみを提供する（AHの認証機能はESPよりも範囲が広く，IPヘッダを含むパケット全体を対象とする)。

図１　IPsec を構成する技術要素

４．組込みシステム向けIPsecサブセット仕様の検討

IPsecの規格は様々な機能を含んでおり，大規模なものである。そのため，メモリ等の資源への制約が厳しい小規模組込みシステムでは，全ての機能に対応した実装を行なう事は困難である。そこで本研究では，IPsecの仕様全体を実装するのではなく，小規模組込みシステムを対象として必要な機能の取捨選択を行ったサブセット仕様を検討し，それに基づいて実装を行うものとした。

INTAP情報家電安全性技術委員会「情報家電向けIPv6最小仕様案」ではIPsecに関するサブセット仕様に関しても一部検討されており[2]，我々のサブセット仕様作成時もこの内容を参考としている。表 1は，今回検討したサブセット仕様の概略をまとめたものである。以下、個々の項目について説明する。

4.1　通信モード

IPsecの通信モードには，IPデータグラムのペイロード部分のみを処理対象とするトランスポートモードと，ヘッダを含むIPデータグラム全体を対象とするトンネルモードがある。前者はEnd-to-Endの通信に，後者はネットワーク間の通信をゲートウェイ上で一括して処理する場合に用いられる。

TINETは，終端ノードとして利用に必要な機能のみをサポートしており，ゲートウェイのような機器での利用は想定していない。そのため，本研究のIPsec実装ではトランスポートモードのみに対応し，トンネルモードに関しては実装しないものとした。

4.2　セキュリティプロトコル

パケット内容の認証機能はESP，AHの両方でサポートされている。また，ESPとNULL暗号を組み合わせる事により，暗号化を行わない認証のみの処理を実現出来るため，AHの機能の大部分はESPによって代替する事が可能である。そこで，セキュリティプロトコルとしてESPのみを実装し，AHについてはサポートしない事とした。

4.3　暗号及びハッシュのアルゴリズム

IPsecでは，通信を行う両者の合意に基づき，暗号，ハッシュのアルゴリズムを選択する仕組みとなっている。具体的にどのアルゴリズムに対応するかは処理系依存であるが，RFC4305では，暗号は3DESおよびNULLが必須(MUST)，AES3)が強く推奨(SHOULD+)と規定されており，ハッシュはSHA1が必須とされている。

不特定多数との通信を行うシステムでは相互接続性向上のために多種のアルゴリズムに対応する事が望ましいが，通信相手が限定されている場合は必要最小限のものに絞り込んでも問題ない。本研究では，暗号としてAES-CBC-128，ハッシュとしてHMAC-SHA1-96をそれぞれサポートする事とした。

4.4　鍵交換プロトコル

前述の通り，IPsec における標準的な鍵交換プロトコルとしてはIKE（Internet Key Exchange）が用いられている。

IKEに関しては，構造が複雑でありリソース制約の厳しい組込みシステム向きではない，例外処理等の仕様が曖昧で相互接続性に問題がある，などといった問題点が指摘されている。しかし，現時点ではIKEに代わる有力な鍵交換プロトコルが存在しないため，本開発ではIKE version 1の，必要最小限の機能について実装を行った。

	対応する項目	対応しない項目
Internet　Protocol	IPv4 (IPv6)	
モード	トランスポートモード	トンネルモード
セキュリティプロトコル	ESP	AH, IPcomp
暗号アルゴリズム	AES-CBC-128 NULL	
ハッシュ	HMAC-SHA1-96 NULL	
鍵交換プロトコル	手動鍵交換 IKE v1(一部)	

表１　IPsecサブセット仕様

５．TINETに対するIPsec機能の実装

４章で検討したサブセット仕様に基づき，TINETをベースとしてIPsec機能の実装を行った。

5.1　TINETの概要

開発のベースとして使用したTINET1)の概要を説明する。TINETは，苫小牧高等工業専門学校 情報工学科の阿部教授によって開発された小規模組込みシステム向けのTCP/IPプロトコルスタックであり，μITRON 4.0仕様の組込みOSであるTOPPERS/JSPカーネル上で動作する。

TINETの設計は，BSD Unix（FreeBSD)のTCP/IP実装に基づいて行われている。ただし，リソース制約の厳しい組込みシステム上で動作させるため，以下に示す方針に従い大幅な修正が加えられている。

- ROM128Kバイト，RAM32Kバイト程度の環境での動作を想定している。
- アプリケーションプログラムインターフェース(API)として，抽象度の高くオーバーヘッドの大きなソケットインターフェースではなく，組込みシステム向けに設計されたITRON TCP/IP APIを採用している。
- 単一のインターフェースを持つ終端ノードでの利用を前提とし，経路選択などの処理を単純化している。
- 動的なメモリ管理を極力排除している。BSD Unixでは通信パケットのデータをmbufと呼ばれる複雑なデータ構造で管理しているが，TINETではより単純な構造を持つnet_bufによるバッファ管理を行っている。

今回のIPsec機能の拡張においても，TINETの持つこれらの特長を大きく損なわないように留意して作業を行った。

5.2　SP・SAデータベースの管理

SPで定義されるルールセットは，ネットワークを設計する段階で固定的に決められる場合が多く，システム動作中に変更が必要となる場合は少ない。そこで，SPの設定はITRONの静的APIを用いて行い，必要なデータ構造をコンパイル時に生成する設計とした。

一方，SAの内容は実行時に設定される場合がある（鍵交換プロトコルによる自動鍵設定を用いる場合など)。そこで，SAについては実行時に設定可能な動的APIを用意した。ただし，SAデータベース全体のエントリ数に関しては，静的に指定する必要がある。SAはそれぞれの通信相手に対して個別に必要となるため（通常は送受信に各1個，計2個のSAが必要)，IPsecによって通信可能な相手の数はSAデータベースのエントリ数によって制限される。

5.3　暗号・ハッシュアルゴリズムの実装

暗号(AES)，およびハッシュ(SHA1)の処理に関しては，FreeBSDカーネル内に含まれる実装をほぼそのまま利用した。呼び出し部分のAPIは共通化されているため，他のアルゴリズムへの変更や追加の作業は容易である。

5.4　パケットの書き換え処理とnet_buf

図２は，IPsec ESPが適用される前後のIPパケットの構造である。暗号化処理では，IPヘッダ中のプロトコル番号は本来の値からESP（50番）に置き換えられる。IPヘッダとペイロードの間にはESPヘッダ(SPI値とシーケンス番号で計８オクテット)が挿入される。ESPヘッダの後ろには，元パケットのIPペイロード部およびESPトレーラが暗号化されて格納される。ESPトレーラ部には，元パケットのプロトコル番号や認証に使用するハッシュ値が格納される。

図２　ESP適用前後のIPパケット

このように，ESPの処理では，データの挿入，削除，移動などの処理が頻繁に発生する。BSDの実装では，パケットを格納するバッファとしてmbufと呼ばれるデータ構造を用いており，無駄なコピーを行わずに複雑なバッファ操作を効率よく実行する事が可能である。

一方，TINETではnet_bufと呼ばれる，よりシンプルなデータ構造を用いている。net_bufは図 3 に示すようにいくつかのヘッダを付加されただけの単一の線形バッファであるが，これでIPsec ESPのパケット書き換え処理を実装出来るかどうかを検討した。

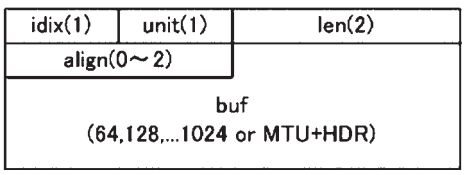

図３　net_buf の構造

送信処理時には，パケット長がESPヘッダ+トレーラ長だけ大きくなる。net_bufの物理バッファサイズは２のべき乗単位であり後方に空き領域が存在する可能性もあるが，保証は出来ない。そのため，ESPパケットを格納するための

net_bufを新規に割り当て，元のnet_bufからコピーしながら暗号化とパケット構築を行う処理とした。

受信処理時には，パケット長は元のパケットよりも小さくなるため，バッファサイズの問題は発生しない。しかし，暗号アルゴリズムがCBCモード（隣接するブロックとXORを取りつつ暗号・復号を行う）で行われる関係で，同一メモリ領域内で復号する事が困難である。そのため，受信時の処理に関しても，新しいnet_bufに対してコピーしながら行う事とした。

このように，送受信処理ともにゼロコピー処理を実現出来ておらず，余分なnet_buf消費（バッファ一つ分）が発生している。ただし，バッファ間のコピー操作の大半は暗号・復号処理と同時に実行されているため，これによって余分なバッファコピー操作が発生している訳ではなく，速度性能に関しては実質的な影響は少ないと言える。

5.5 IPsecパケット送受信時の処理

IPsec ESP パケットの受信時，送信時の処理フローを，それぞれ図４，図５に示す。これらの処理は，主にTINETのIP層入出力処理であるip_input()およびip_output() 関数の内で行なわれている。

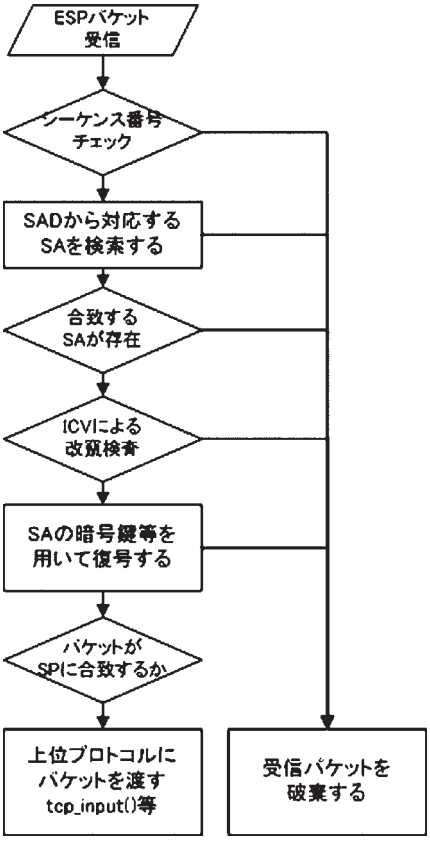

図４　ESP受信パケットの処理

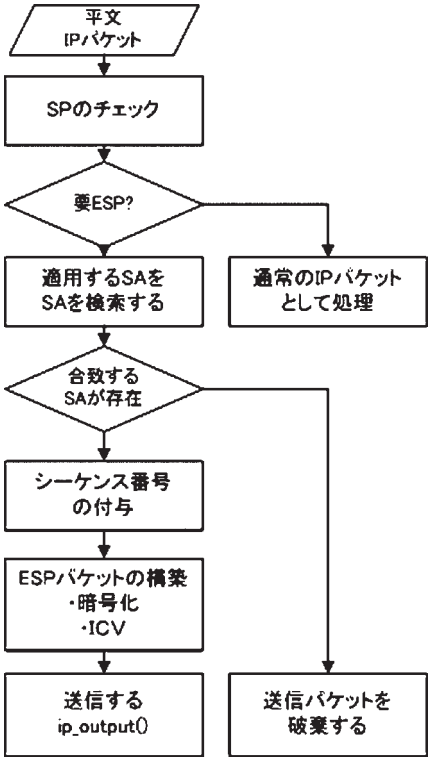

図５　ESP送信パケットの処理

６．鍵交換プロトコルの実装

本章では，TINET上におけるIKE 鍵交換プロトコルの実装について説明する。

6.1 IKEの概要

IKEによる鍵交換は，IPsec による通信保護が確立する以前に行わなければならない。そのため、IKEの処理は，IKE自身のための保護された通信路(ISAKMP SA)を確立する処理（Phase1）と、その上でIPsec SAに必要な情報の交換を行う処理(Phase 2)の二段階で行われる。

Phase 1では，ISAKMP SA で用いる暗号鍵などの情報を安全でない通信路を用いて交換するために，公開鍵暗号の一種であるDiffie-Hellman アルゴリズムが用いられる。

6.2 Diffie-Hellman の実装

Diffie-Hellman アルゴリズムの詳細な説明は省略するが，その処理は，

$$g^x \bmod p$$

という形の計算（冪乗剰余計算）で構成されている。ただし，g，x，p は非常に大きな数（768～2048ビット程度）であるため，計算負荷は非常に大きい。

本研究では，Diffie-Hellmanアルゴリズム及びそれに必要な多桁長演算処理を実装するために，オープンソースソフトウェアOpenSSL に含まれるコードの一部を利用した。

OpenSSLはSSL/TLSを実現するためのライブラリの集合体であるが，DH(Diffie-Hellman)，BN(多桁長演算)などのサブモジュールを含んでいる。BNモジュールは汎用の多桁演算ライブラリであるため，Diffie－Hellmanに必要な冪乗剰余算，およびそれと依存関係のある関数のみを抽出し，不要なコードを削除する事で，コードサイズの縮減を図った。

OpenSSLはPOSIX環境向けのソフトウェアであるため，メモリ管理のためにmalloc()，free()などのヒープメモリ操作を多用している。しかし，μITRON上のプログラムでは，動的なメモリ管理機構の使用は好ましくない。今回の実装では，可能な部分は静的メモリ割り当て処理に置き換え，単純な置き換えが難しい部分については簡略化されたmalloc()，free()実装を作成して対応した。

6.3 IKEプロトコルの実装

6.2節で実装したDiffie-Hellmanを用い，IKEプロトコルの処理を実装した。実装範囲はIPsec の鍵交換処理に必要な最小範囲とした。具体的な実装項目は以下の通りである。

- Phase 1 メインモード
- Phase 2 クイックモード
- 暗号，ハッシュは，IPsecと同じAES-CBC-128，HMAC-SHA1-96のみ対応
- 認証手段は，共有秘密鍵のみ対応

実装したIKEによって行われる鍵交換処理の実行例を，図６に示す。

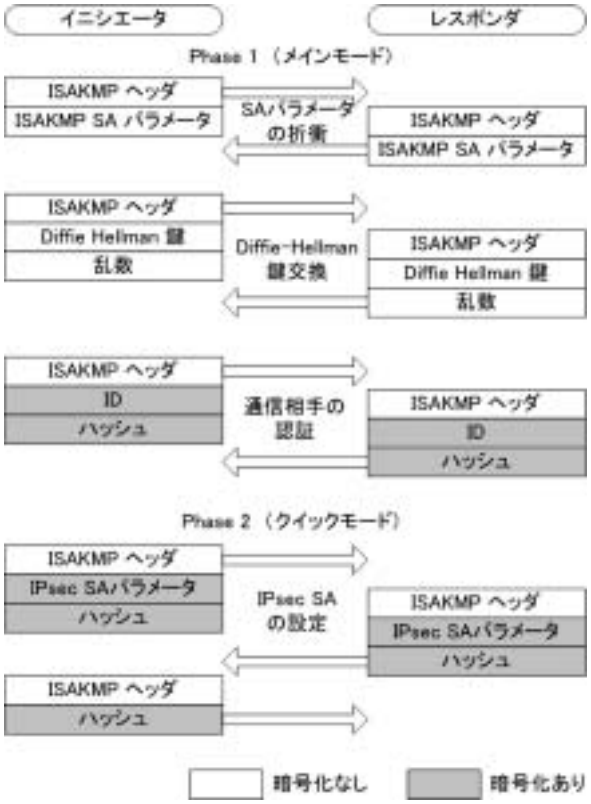

図６　IKEによる自動鍵交換処理の例

７．開発したソフトウェアの性能評価

本研究で試作したIPsecソフトウェアを，複数のマイコンシステム上に搭載し，実行速度やメモリ使用量などに関して評価を行った。

7.1 評価に用いたハードウェア環境

評価実験に使用した組込みマイコンシステムの仕様を表２に示す。いずれも，ネットワークインターフェースとしてEthernetを搭載した，小規模～中規模の組込み向けマイコンである。以下，評価用マイコン1～4と呼ぶ。

また，マイコンボードと対向するPC側は，FreeBSD 5.1を搭載したノートPCを使用した。

表２　評価用マイコンシステム

No.	CPU	clock	Ethernet NIC
1	ルネサステクノロジ H8S2638	20 MHz	NE2000互換 (ne)
2	ルネサステクノロジ SH7615 (Cache off)	60 MHz	SH-Ether (she)
3	ルネサステクノロジ SH7727	98 MHz	LAN91C111 (sn)
4	ザイリンクス PowerPC405 (Cache off)	64 MHz	LAN91C111 (sn)

図７　評価マイコン4（PowerPC405）

7.2 暗号・復号の処理速度

各マイコンにおける暗号・復号処理の速度を評価した(表３)。暗号アルゴリズムは鍵長128bitのAES-CBCであり，暗号，復号処理の時間は，いずれも１ブロック(128bit/16 octet)あたりの処理時間である。測定は，暗号・復号処理部分の前後に外部ポート出力を埋め込み，オシロスコープにより出力波形を計測する事で行った。

表３　暗号・復号の処理速度

マイコン	暗号処理(μsec).	復号処理(μsec)
1	1489	1403
2	214	214
3	69	67
4	125	127

7.3　IPsec ESPの通信速度

IPsec ESPの処理全体に関する速度を評価するため，PCとマイコンボードの間でTCP ECHOサービスによる通信を行い，その通信時間を評価した。

試験環境としては，評価用マイコン２及び４を使用した。

測定する時間は，コネクションの確立からECHOメッセージの交換，コネクション開放までに要した時間とした。ECHOのメッセージの大きさは64，128，256，512，1024オクテットと変化させ，IPsec ESPを適用した場合と適用しない場合についてそれぞれ1000回の通信を行い，その平均時間を評価した。時間の計測は，PC側でプロトコルアナライザによってパケットの送受信を記録し，そのタイムスタンプ値から計算した。

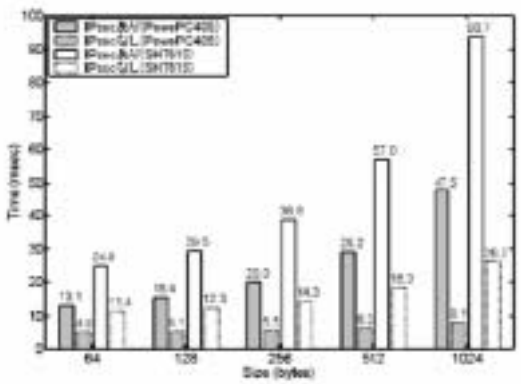

図８　TCP ECHOの通信速度

実験の結果を図 8に示す。送信メッセージサイズに対する通信時間の増加は，平文の場合はそれほど大きくないのに対し，ESPを適用した場合は非常に大きくなっている。これは，処理時間の大部分が暗号・復号処理に費やされているためである。

組込み向けCPUでは暗号のソフトウェア処理は負荷の高い処理であり，大量のデータを高速に送受信するようなアプリケーションでの利用は困難と考えられる。しかし，通信データサイズが小さい場合は，低速なCPUを用いたシステムでも十分実用的な処理が可能である。遠隔監視などの用途では、扱うデータ量が少なく高速通信も必要ない場合も多く，活用が期待される。

音声や画像の伝送など，高速大容量の通信が必要な製品の場合には，暗号処理のハードウェア化が必要であろう。最近の組込みマイコンには暗号アクセラレータをオンチップで搭載する製品も増えており，それらのデバイスを利用する事が考えられる。また，評価マイコン４の様なFPGA混載型のシステムの場合は，FPGA上に暗号エンジンを実装する事も可能である。

7.4　鍵交換の処理速度

試作したIKE実装を使用し，鍵交換に要する処理速度を評価した。IKEの処理はDiffie-Hellmanの計算負荷が非常に高いため、試験環境にはCPU性能の高い評価マイコン3を使用し、IKE Phase1について計測を行った。速度の計測は、IKE通信のパケットをPC側のプロトコルアナライザで観測し、そのタイムスタンプによって行なった。

表４　IKE Phase1 の所要時間

パケット	方向	時刻（sec）
1	PC→マイコン	0.00
2	マイコン→PC	0.12
3	PC→マイコン	5.27
4	マイコン→PC	5.46
5	PC→マイコン	12.81
6	マイコン→PC	13.11

計測結果を表４に示す。対向するPC側と比較すると処理が遅く、通信の一部で相手側のタイムアウトによる再送処理が発生する場合もあったが、鍵交換処理自体は正常に完了する事が出来た。

より低速のCPU環境（H8Sなど）では，IKEによる鍵交換を行うのは困難と思われる。これらの環境では，自動鍵交換機能を使用せず，IPsec SAを手動で設定する運用方式が現実的と考えられる。

7.5　メモリ使用量

実装したIPsec のRAMおよびROMのメモリ使用量を評価した。評価環境は，評価マイコン２を使用した。

結果を表５に示す。IPsecを利用しない場合と比較して，RAMが約15%，ROMが約30%程度の増加に止まっており，TINETが動作する程度の小規模組込みシステムであれば，十分実用的に利用できる範囲と考えられる。

表５　IPsec 実装のメモリ使用量

	RAM (byte)	ROM (byte)
IPsec	22060	40772
暗号	44	24360
OS+TINET	126070	224698

８．応用事例

開発したIPsecソフトウェアを用いた応用事例を，いくつか紹介する。

図９は、TINET付属のWWWサーバ(nserv)へのアクセスにIPsecを適用した例である。IPsecを用いた場合でも，通常の全く同等な結果が得られている。IPsecに関する処理は全てプロトコルスタックの内部で行なわれるため、アプリケーション側ではコードの変更を一切行わずにIPsecを利用する事が可能であった。

図10は，開発成果物を展示会等へ出品する際に，IPsec の動作を視覚的にわかりやすく説明するためのデモシステムである。マイコンボード側ではTCP ECHOサービスが動作しており，PC画面の右上部のボックスに入力してテキストがマイコン側へ送信され，エコーバックされた結果が右下に表示される。PC側のネットワークインターフェースではパケットキャプチャを行っており，TCPのペイロード部（ESPの場合はそれに相当する部分）の内容を取り出し表示している。IPsecを用いない平文通信の場合は入力した通信内容がそのまま表示されてしまうが，IPsec を適用した場合は暗号化されたデータが表示され，通信が保護されている事がわかる。

図９　WWWサーバの動作例

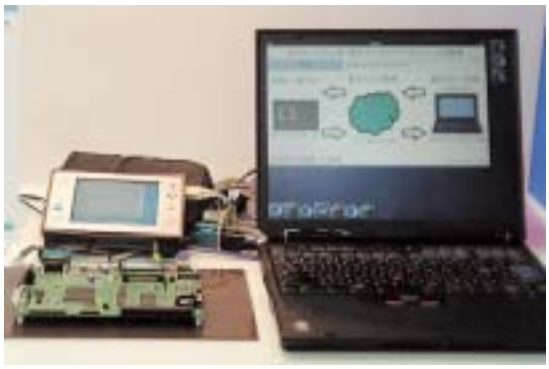

図10　展示会向けデモシステム

９．まとめ

μITRON OS用のオープンソースTCP/IPプロトコルスタックであるTINETに対して，IPsec 及びIKEの機能を実装し、暗号化によるセキュアな通信機能を実現した。

性能評価試験の結果では，PC等と比較して低性能な組込み向けCPUによるソフトウェア実行では高速処理は期待できないものの，通信量が大量とならなければ十分実用的なシステム構築が可能である事が示された。

今後は，より高速な暗号処理が必要な分野での利用を目指し，ハードウェア暗号エンジン等との協調動作などに関して研究を進める方針である。また，今回の実装では不完全な部分に関する機能強化も併せて進めたいと考えている。

なお，本研究で開発したIPsec関連ソフトウェアは，TINET本体と同様にTOPPERSライセンスによるフリーソフトウェアとして公開，配布する事を予定している。

謝辞

今回の研究開発にあたっては，TINET TCP/IPプロトコルスタックの開発者である苫小牧高等工業専門学校 情報通信科　阿部　司 教授，TOPPERSプロジェクトを主宰される名古屋大学大学院情報科学研究科　高田広章教授には，初期の開発方針から実装の細部に関する事項で幅広くご指導を頂きました。この場を借りて感謝の意を表します。

引用文献

1) 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 No.6，pp.1583-1592，(2003)
2) INTAP 情報家電安全性技術委員会：情報家電向けIPv6最小仕様案，http：//www.tahi.org/lcna/docs/IPv6-min-spec/IPv6-min-spec-ver42.htm
3) Announcing the ADVANCED ENCRYPTION STANDARD (AES)，Federal Information Processing Standards Publication 197，(2001)
4) Kent，S.・R. Atkinson：“Security Architecture for the Internet Protocol”，RFC2401，(1998)
5) Harkins，D.・D. Carrel：“The Internet Key Exchange(IKE)”，RFC 2409，(1998)