ハードウェアAES暗号処理とIPsecへの適用

堤　大祐

Development of Hardware AES Processor and its Application to IPsec

Daisuke TSUTSUMI

抄　録

近年，組込みシステムにおいてもインターネットに接続する機能が不可欠になってきた。また，これらのシステムは個人情報や機器の制御情報などの重要な情報も取り扱うため，通信のセキュリティがより重要な課題となっている。筆者はμITRON仕様のリアルタイムOS上で動作するTCP/IPプロトコルスタックにAES暗号を用いたIPsecの実装を行った。AES暗号処理はCPUにとって高負荷な処理であるため，組込みシステムにおいては，本来，CPUに求められている機能を損なわずに，暗号処理機能を付加することが求められている。そこで，AES暗号処理を行うハードウェアをFPGA上で開発し，IPsecに適用した。

キーワード：AES暗号，IPsec，ハードウェア，FPGA

Abstract

Recently, it is indispensable function to connect to the internet for embedded systems. Also, security of internet communications is important problem, because embedded systems work with personal information and/or the operational parameters. The author applied IPsec with AES encryption to a TCP/IP protocol stack for a μITORN real-time operating system. Processing of AES encryption requires high-performance to CPU, but, it is important to add the function of AES encryption without high-performance CPU. Therefore, the author developed AES hardware processor on an FPGA device, and applied IPsec.

KEY-WORDS : AES encryption, IPsec, hardware, FPGA

1．はじめに

近年，監視装置をはじめ，家電製品などの組込みシステムがインターネットに接続するようになってきた。組込みシステムのネットワーク接続における課題として，通信のセキュリティが挙げられる。特に組込みシステムは使用者が意図しないところで動作する場合が多く，家庭用防犯システムなどの場合における個人情報や機械制御装置の制御情報などの通信を暗号化によって保護する必要がある。

このようなインターネットに接続する組込みシステムの開発において，TCP/IPを用いた通信機能の開発が必須となっている。TCP/IPを用いた通信にはセキュリティ・プロトコルが定義されている。例えば，SSL（Secure Socket Layer）はWWWベースの通信で広く用いられているプロトコルである。一部のブラウザでは錠のアイコンが表示されるなどして，通信が暗号化されていることがわかる。このプロトコルはTCPによる通信において機能するので，UDPなどのプロトコルは暗号化されない。一方，IPsecはIPを用いる通信がすべて対象となり，上位のアプリケーション・プログラムにおいて，通信の保護について意識する必要がない。IPv6に

事業名：一般試験研究
課題名：ソフトウェア/ハードウェア協調処理による暗号通信処理システムの開発

おいてはIPsecの実装が必須である。

筆者はμITRON4.0仕様[4]のオープンなリアルタイムOSであるTOPPERS/JSPカーネル[10]と，同じくオープンなTCP/IPプロトコルスタックであるTINET[5]を用いて，AES暗号アルゴリズムを用いたIPsecの実装および評価を行ってきた[1)6)]。しかし，暗号処理はCPUに負担がかかる処理であるため，本研究ではAES暗号処理部分をハードウェア化してCPUの負荷を低減することで，IPsec通信の高速化を図った。ハードウェア化にはFPGA（Field Programmable Gate Array）を用い，FPGAを使用した試作システムを構築しIPsec通信を行った。FPGAは設計者や開発者が1個から自由に回路構造をプログラムできるLSIであり，低コストで高速なハードウェア処理機能を容易に実現できる。

2．IPsecの概要とESP処理

IPsec（IP security）はネットワーク層で機能するため，IPを用いた通信を保護できる。そのため，上位プロトコルに位置するTCPをはじめ，UDPパケットなども暗号化して保護できる。IPsecにはESP（Encapsulating Security Payload：暗号ペイロード）[9]とAH（Authentication Header：認証ヘッダ）の2つのプロトコルと，トンネルモード，トランスポートモードの2つのカプセル化モードがある。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。

IPsecはセキュリティ・ポリシー（Security Policy：SP）によって，IPパケットの取り扱いを指定する。具体的にはSPはネットワークアドレス，トランスポート層のプロトコルなどから成り，「IPsecの適用」，「IPsecを適用しない（通常の処理）」，「パケットの破棄」といったIPパケットの取り扱いを指定する。また，セキュリティ・アソシエーション（Security Association：SA）によって，ESPやAHといった使用するIPsecのプロトコル，暗号アルゴリズム，認証アルゴリズムなどを指定する。今回，試作システムに用いた機能を表1に示す。

表1　試作システムに用いたIPsecの仕様

セキュリティ・プロトコル	ESP
カプセル化モード	トランスポートモード
Internet Protocol	IPv4
暗号鍵の交換	手動
暗号アルゴリズム	AES-CBC，鍵長 128bit

トランスポートモードにおいて，ESPはIPパケットの内容を暗号化する（図1参照）。IPヘッダの次に新たにESPヘッダが挿入される。ESPヘッダはSPI値（Security Parameters Index）とシーケンス番号から構成される16バイト長のヘッダである。初期ベクトルは暗号処理時に使用する値で，ランダムな値を設定する。初期ベクトルを使用することによって，同じデータに対して暗号化処理を行っても異なる結果が得られ，より解読しにくい効果がある。ESPトレーラには元のパケットのプロトコル番号などが格納される。

図1　ESP適用後のIPパケット

IPパケットの処理フローを図2に示す。IPsecを適用しない場合（図2の点線），IP処理部とTCP/UDP/ICMP処理部との間でデータの受け渡しを行う。IPsecを適用した場合（図2の実線），IPパケットはESP処理部でSP/SAと照合され，IPsecを適用するルールに従って，指定の暗号アルゴリズムで復号化または暗号化される。本研究では暗号アルゴリズムにAESを用いた。

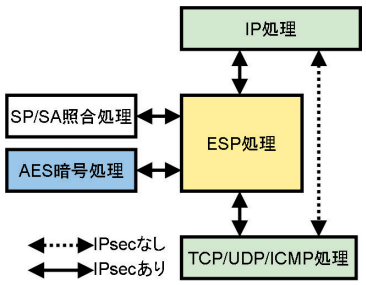

図2　IPパケットのESP処理フロー

3．AES暗号処理の概要と暗号モード

3.1　AESとは

AES（Advanced Encryption Standard）は，共通鍵方式のブロック暗号であり，次世代標準規格である[7]。従来の標準とされていたDES（Data Encryption Standard）や3-DESに代わる暗号アルゴリズムである。AESの処理の流れを図3に示す。

AESは128/192/256-bitの3種類の鍵長を有し，128-bitを単位ブロックとして暗号処理を行う。AESはこの単位ブロックをラウンドと呼ばれるAddRoundKey操作，SubBytes操作，ShiftRow操作，MixColumns操作の4操作を処理単位とし，基本的にこのラウンドを繰り返すことによって処理を行う。繰り返し数Nrは鍵長に依存し，鍵長128-bitの場合は10，192-bitの場合は12，256-bitの場合は14となっている。また，鍵はラウンド数分だけ拡張される。

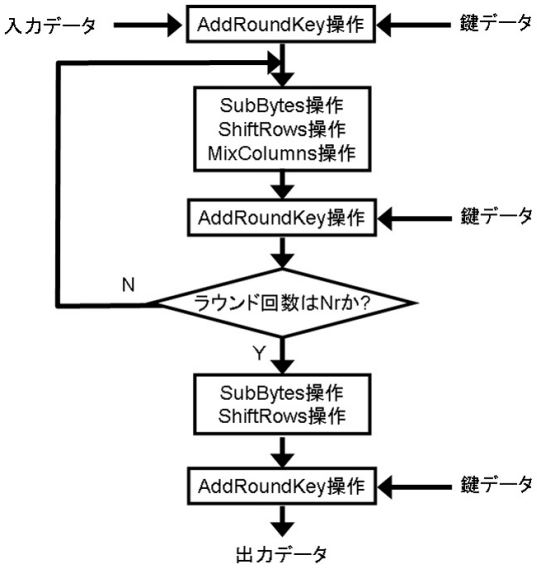

図３　AESアルゴリズム（暗号化）

3.2　暗号モード

暗号モードは単位ブロックより長い入力データを暗号処理する時に使用する手法である。主な暗号モードを以下に示す。

○ECB（Electronic Code Book）モード

入力データごとに暗号処理を行う方法であり，並列に暗号処理を行える。しかし，入力データが同じ場合，結果も同じになるため，解読される可能性があり，推奨されていない。ECBモードの処理の流れを図４に示す。

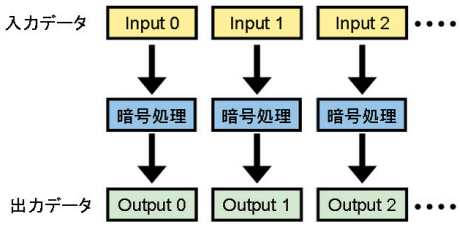

図４　ECBモード

○CBC（Cipher Block Chaining）モード

一つ前の入力データを暗号化した結果と次の入力データの排他的論理和をとり，その結果に対して暗号処理を行う。図５にCBCモードの処理フローを示す。最初の入力データを暗号化する場合，外部から与えられた初期ベクトルと排他的論理和を行う。これにより，同じ入力データを暗号化しても結果が異なり解読が難しくなる。しかし，前の暗号処理結果を使って次の暗号処理を行うため並列処理はできない。本研究では解読が難しく，実際に広く用いられているCBCモードを採用した。

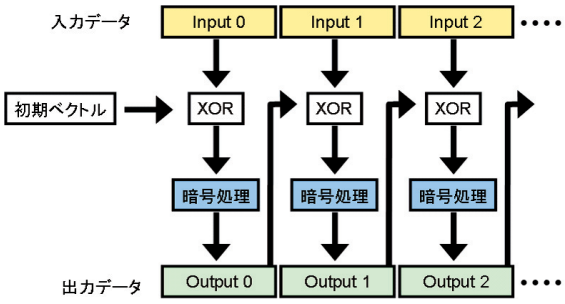

図５　CBCモード

3.3　AES処理のハードウェア化

ハードウェアの開発は図６に示すXilinx製FPGAを使用した。このFPGAはCPUを内蔵しており，開発したAES暗号処理モジュールを接続したシステムを構築しやすい利点がある。従来，ソフトウェアで行っていたIPsec通信を行いつつ，AES暗号処理モジュールを使用した動作試験を容易に行うことが可能である。使用したFPGAの仕様を表２に示す。開発環境にはXilinx製ISE10.1.03とEDK10.1.03を使用した。

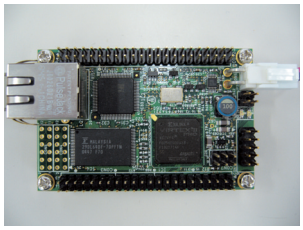

図６　試作に使用したFPGAボード

表２　FPGAボードの仕様

デバイス	XC2VP4-fg256-5 (Xilinx)
動作周波数	約 65MHz(CPU は約 260MHz)
CPU	PowerPC405
大きさ	72(W) x 47(H)　mm
メモリ	SDRAM(32M)

本モジュールを含む試作システム全体のブロック図を図７に示す。FPGA内にCPU，AES暗号処理モジュールとSDRAMコントローラ等からなり外部SDRAMと接続している。

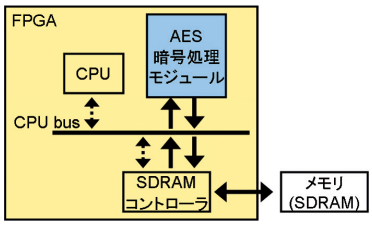

図７　試作システムのブロック図

AES暗号処理モジュールは主に以下の情報をCPUから得て処理を開始する。

・入力データのアドレス，出力データのアドレス

・暗号化/復号化の選択，処理データ数

動作の流れを図 8 に示す。処理開始後，CPUから与えられた入力データのアドレスから，バースト転送でデータを読み出す。読み出したデータは逐次暗号処理される。全ての処理終了後，CPUより与えられた書き込みアドレスにバースト転送して処理を終える。

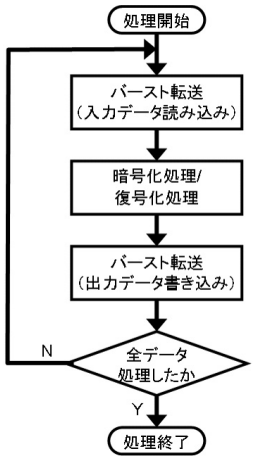

図 8　AES暗号処理モジュールの動作フロー

AES暗号処理モジュールの詳細を図 9 に示す。128-bitの単位ブロックを暗号処理するAESプロセッサ，AESプロセッサに与える入出力データ，拡張鍵を管理する入出力データ管理部，およびメモリからデータを取得，メモリへデータを書き込むためのアドレス情報を保持する制御レジスタ，およびCPUバスとインターフェースをとるバス制御から構成される。

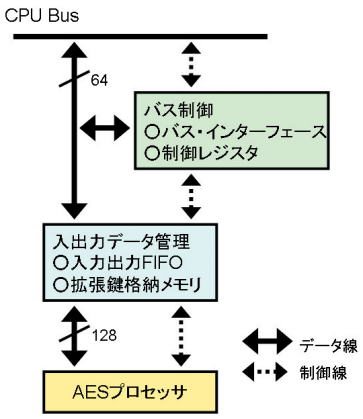

図 9　AESプロセッサと周辺回路の構成

AES暗号処理モジュールは図 3 に示すように128-bit入力データと128-bitの鍵データを用いて，AddRoundKey操作，SubBytes操作，ShiftRow操作，MixColumns操作をハードウェアで行い，処理結果を128-bitで出力する。バス制御部はマスタ機能を備え，CPUバスにアクセスしてSDRAMコントローラを経由して外部SDRAMから暗号処理に必要なデータを取得し，暗号処理結果を書き込む。その際，バースト転送を行う。試作システムにおけるCPUバスは64-bit幅[3)]であり，最大16データ分バースト転送可能である。これらの機能を使用して，メモリアクセスの効率向上を図った。一方，AESプロセッサは128-bit単位で入出力を行う。そのため，入出力データ管理部ではbit幅を調整する機能を備えた。また，64-bit幅のデータを16データバースト転送するので，それぞれ64-bit幅で16段の入力用FIFO，出力用FIFOを備えた。AESプロセッサに使用したFPGAのリソースを表 3 に，試作システム全体において使用したFPGAのリソースを表 4 に示す。LUTは4-input Look-Up Tableの略である。Block RAMはFPGA内部にあるメモリで1 blockは18-Kbyteである。

表 3　AESプロセッサの使用リソース

LUTs	1,117
Block RAMs	5
Maximum frequency	約 138MHz

表 4　試作システム全体で使用したリソース

LUTs	5,819
Block RAMs	17
DCMs	2

試作システムにおいて，AESプロセッサはCPUバスに接続した。一方，EDKはAESプロセッサをCPUバス接続なしにメモリにアクセスできるSDRAMコントローラを用意している[2)8)]。しかし，本研究ではFPGA内蔵CPU以外のシステムに適用可能にするため，CPUバスに接続した。

4．動作試験

4.1　単体試験

AES暗号処理モジュールの演算処理時間を計測した。計測方法はソフトウェアからIOポートを制御し，オシロスコープで波形の長さを計測して行った。ソフトウェア処理とハードウェア処理で条件を同じにするため，ハードウェア処理ではメモリアクセスに必要な時間も含めた。データ処理量を32バイトから1,024バイトまで 6 段階に分けて計測した。動作試験の結果を図10に示す。

図10において，“HW-AES”が開発したAES暗号処理モジュールの実行時間，“SW-AES”がCPUで処理した場合の実行時間である。図に示すようにCPU処理に比べてAES暗号処理モジュールは約10倍高速に処理することができた。なお，CPUの動作周波数は約260MHzであり，命令キャッシュ，データ

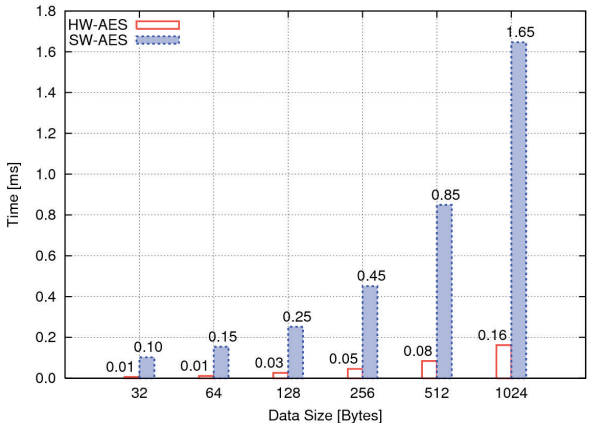

図10　AES暗号処理モジュールの処理時間

キャッシュを有効にした。SDRAMとAES暗号処理モジュールの動作周波数は約65MHzである。

4.2　IPsec通信試験

IPsec通信を行った時の，暗号処理にかかる時間を計測した。計測方法は試作システムにUDP-Echoサーバを動作させ，別に用意したPCからUDPパケットを送信する。PCではまた，プロトコルアナライザを起動させ，パケットの送信時刻と試作システムからの応答パケットの受信時刻を記録する。この2つの時刻の差が通信にかかった時間となる。IPsecを適用した時の通信時間とIPsecを適用しなかったときの通信時間との差を暗号処理にかかった時間とした。この場合，試作システムで行われた復号化処理時間と暗号化処理時間，および，IPsec通信のオーバヘッドを含んだ時間となるが，ここではそれらを含んだ時間を暗号処理時間とした。

通信試験はUDPのメッセージの大きさを32バイトから1,024バイトまで6段階に変化させて行った。まず，UDP-Echoの通信時間の実験の結果を図11に示す。図中，“Non-IPsec”はIPsecを適用しなかった場合，“Hw-Esp”はIPsecを適用し，

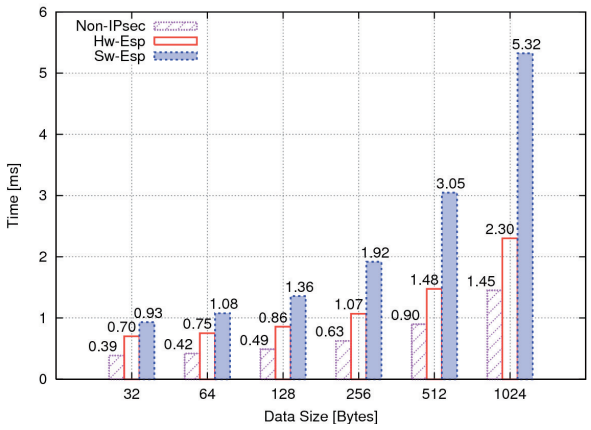

図11　UDP-Echoサービスの通信時間

AES暗号処理モジュールを使用した場合，“Sw-Esp”はIPsecを適用し，CPUで処理を行った場合を示す。メッセージの大きさが32バイトの場合，IPsecを適用しなかった場合に比べて，AES暗号処理モジュールを使用した時は約1.8倍，CPUで処理を行った場合約2.4倍通信時間を要した。また，1,024バイトの場合，それぞれ約1.6倍，約3.7倍となった。AES暗号処理モジュールを使用した方が，IPsec通信にかかる時間がIPsecを適用しない場合に比べて約1.6〜1.8倍の時間増で通信できることが確認できた。

次に，IPsec適用時にかかる暗号処理時間を図12に示す。UDPのメッセージの大きさが2倍になっても，暗号処理時間は2倍にはならない。これはIPsec適用にかかるCPUの処理時間がメッセージの大きさにかかわらず必要となるためである。32バイトのメッセージに対して，暗号処理時間はCPUで処理した場合0.55ms，AES暗号処理モジュールで処理した場合0.32msとなり，暗号処理時間が約40%改善された。また，1,024バイトの時はそれぞれ，3.87ms，0.85msとなり，暗号処理時間は約80%改善された。このように，メッセージが大きい場合，AES暗号処理モジュールがより効果的であることが確認できた。

なお，ESPプロトコルに使用する暗号アルゴリズムはAES-CBC，鍵長128ビット，鍵交換は手動で行った。動作周波数などは単体試験と同じである。UDPパケットの暗号化および復号化は正しく処理されており，IPsec適用時のPCとマイコンボード間のEchoメッセージの交換は正常に行われた。

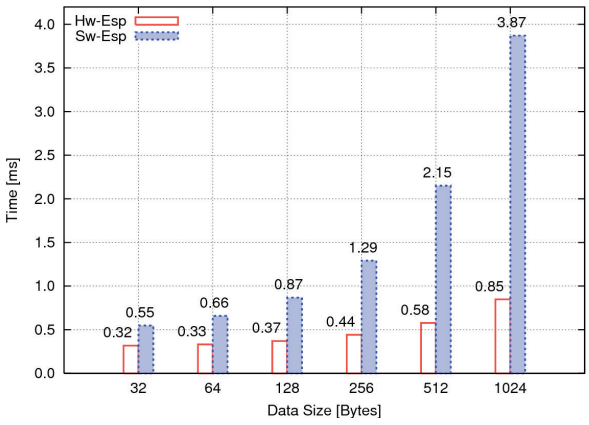

図12　UDP-Echoサービスにおける暗号処理時間

5．まとめ

AES暗号処理モジュールを開発した。本モジュールをIPsec通信に適用した結果，IPsecを適用しない場合に比べて，約1.6倍から約1.8倍の通信時間で情報を保護できた。よって，組込みシステムにおいて，CPU処理では対応できない用途にも適用可能であり，IPsec通信の適用範囲が広がった。

本モジュールは単独で動作可能であるため，IPsecへの適用の他，組込みシステム内部のパラメータ情報や動作，または，操作の履歴情報の保護，ファイルシステムと連係させ保存する情報の暗号化などに応用が可能である。また，AESプロセッサ単独では動作周波数約138MHzで動作可能なことから，高速な暗号処理が必要な用途にも適用が可能である。

引用文献

1）堀武司・堤大祐・吉川毅・山本寧：組み込みシステム向けIPsecの実装と評価，北海道立工業試験場報告，No.306，pp.1-8，（2007）

2）“MCH OPB Synchronous DRAM（SDRAM）Controller”，Xilinx Corporation，DS492，Jan.（2006）

3）64-Bit Processor Local Bus Architecture Specifications Version 3.5，IBM，（2000）

4）坂村健(監修)・高田広章(編)：μITRON4.0仕様4.02.00，トロン協会，（2004）

5）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）

6）堤大祐・堀武司・吉川毅・山本寧：組込みシステム向けTCP/IPプロトコルスタックにおけるIPsecの実装と評価，第5回情報科学技術フォーラム（FIT 2006），pp.249-250，（2006）

7）FIPS-197：Announcing the ADVANCED ENCRYPTION STANDARD（AES），NIST，Nov.（2001）

8）“Multi-Port Memory Controller”，Xilinx Corporation，DS463，Jul.（2008）

9）S. Kent，R. Atkinson：“IP Encapsulating Security Payload（ESP）”，RFC2406，（1998）

10）TOPPERSプロジェクト：http://www.toppers.jp/index.html