組込み機器向け暗号化ミドルウェア

松山　淳　　細貝　和彦

来るべきユビキタスネットワーク社会の実現に向けて，さまざまな組込み機器がネットワークに接続されはじめている。それに伴い，これら機器におけるセキュリティの確保が重要視されるようになった。

一般的に，組込み機器に使用されるCPUやOSは製品によりさまざまであり，使用される暗号化アルゴリズムも多様である。

当社では，これらの特徴を考慮した組込み機器向けの暗号化ミドルウェア（以下，本ソフトウェア）を開発した。本稿ではその概要を述べる。

ユビキタスネットワークとセキュリティ

ユビキタスネットワークの発達につれ，従来では考えられなかった家庭内のさまざまな電化製品がネットワーク化されるようになり，これに伴って新たな製品やサービスが創出されようとしている。総務省は，ユビキタスネットワーク関連の市場規模が2007年には59.3兆円，2010年には87.6兆円になると試算している[1]（**図1**）。

一方，ユビキタスネットワークの発達は，ネットワーク化された各種組込み機器に対して，セキュリティの確保という新たな課題をもたらした。

組込み機器に求められるセキュリティ製品の特徴

従来のセキュリティ製品は，パソコンやワークステーション等の情報機器やルータやゲートウェイといったネットワーク機器に重点がおかれていた。このため，比較的処理能力の高いCPUを有し，かつ限られた機器やOS間にクローズした環境で利用されることを前提としていた。

これに対し，ユビキタス時代では，ネットワークに接続されたさまざまな組込み機器と，これら機器間通信でセキュリティの重要性が増すと考えられる。このため，さまざまなCPUやOSによって構成され，かつコスト的な制約の厳しい組込み機器に対応したセキュリティ製品が必要になると考えられる。

これらのことから，本ソフトウェアの開発では，以下の点に留意した。

- 安価で標準的なアルゴリズムを採用する。
- 複数のプラットフォームに容易に移植できる。
- サイズがコンパクトで処理が軽い。

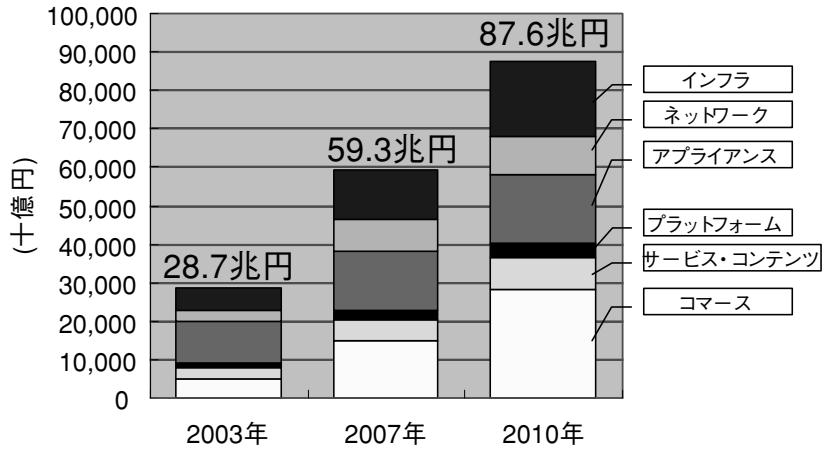

図1　ユビキタスネットワーク関連の市場規模予測
（出展：情報通信白書平成16年版）

アルゴリズムの選択

組込み機器に対して暗号化を実装する際には，組込み機器特有のメモリ容量や処理能力の制限，コスト的制約，および互換性や汎用性に配慮する必要がある。

IPA（独立行政法人 情報処理推進機構）による「国内で入手可能な暗号関連製品リスト」[2] に掲載されている230製品で使用されている暗号化方式のうち，共通鍵暗号方式として使用されている暗号化アルゴリズムについて調べた結果を**表1**に示す。

表1　暗号アルゴリズムと採用数

アルゴリズム	採用数	ロイヤリティ
DES(Triple DES含む)	125	不要
AES	52	不要
RC2，RC5 *1)	31	要
Blowfish/Twofish	11	不要
MISTY1*2)	10	条件による
SXAL *3)	4	要
IDEA*4)	4	条件による
その他	25	—

表1に示すように，NIST（米国商務省標準技術局）によって米国の標準暗号として定められたDES（Data Encryption Standard）とAES（Advance Encryption Standard）の採用数が多く，これら2つの暗号化方式で全体の67%以上を占めていることが分かる。これは，DESが既に1970年代から20年以上も利用されている暗号化方式であり，さまざまな分野においてデファクトスタンダードとして利用されているためである。一方AESは2000年に選定された比較的新しい暗号化方式であるが，既に暗号化強度が疑問視されているDESの後継暗号として今後の利用拡大が予測されているためであると考えられる。

さらに，**表1**に示したアルゴリズムの中で，特許が既に失効している，または特許権者が権利を放棄する等の理由により自由に利用できるアルゴリズムとして，DES，AESに加えてBlowfishおよびTwofishが上げられる。この2つの暗号方式は共にブルース・シュナイアー（Bruce Schneier）によって考案された共通鍵のブロック暗号アルゴリズムで，特にTwofishはAES選定の際，最終選考まで残ったことで知られている。これらロイヤリティフリーのアルゴリズムの採用は，コスト的制約の厳しい組込み機器に対して非常に有用である。

これらさまざまなアルゴリズムの特徴を考慮し，安価でかつ，十分な機能と互換性をもった暗号化機能を，組込み機器に対して実装することをねらい，本ソフトウェアでは，まずDESとBlowfishを実装した。なお，次に述べるマルチプラットフォーム化への配慮により，他のアルゴリズムも比較的容易に実装することができる。

マルチプラットフォーム化

暗号化機能をマルチプラットフォームに対応させ，移植の容易性を高めるためには，OSによる差異と，使用するCPUによる差異の2点を考慮する必要がある。

(1) OS依存部とアルゴリズム部の分離

OSによる差異は，タスク（プロセス，スレッド）管理機能とスケジューリング方法，各タスク間の同期，通信機能，およびメモリ管理，割り込み管理が主であるが，暗号化機能を実装する上で，これらOSに依存する機能を**図2**に示すように，OS対応部としてアルゴリズム部と分離する構成とした。これにより，アルゴリズム部をOS非依存として構成することができ，どのOSでも同じアルゴリズム部のコードが利用できる。さらにアルゴリズム部とOS対応部とのインタフェースを定めることにより，複数のアルゴリズムを1つのOS対応部上に容易に構築することが可能となり，アルゴリズムの追加，変更等に柔軟に対応することができる。

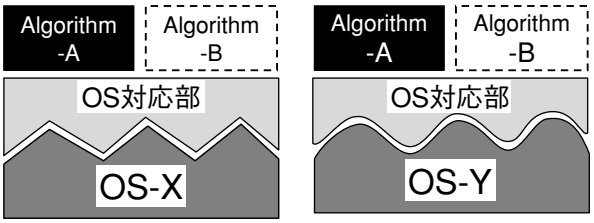

図2　OSに依存しない実装イメージ

(2) CPU非依存なアルゴリズム部の構成

CPUによる差異は，ビット幅とエンディアンの違いによるものが主であり，この違いはアルゴリズム部の実装に大きく影響する。一般に，これらの差異は定義ファイル等により吸収するが，本ソフトウェアでは，これに加えて**図2**で示したアルゴリズム部内のCPU依存処理をライブラリとして分離し，**図3**に示す構成とした。

これによってアルゴリズム部のCPU依存性が排除され，他CPUへの移植が容易になった。さらに，この構成は次項以降に述べる処理の最適化とコードサイズの縮小にも寄与している。

*1)RC2,RC5はRsa Security Inc.の登録商標です。　*2)MISTYは三菱電機(株)の登録商標です。　*3)SXALは(株) ローレルインテリジェントシステムズの登録商標です。
*4)IDEAはASCOM Systec社の登録商標です。

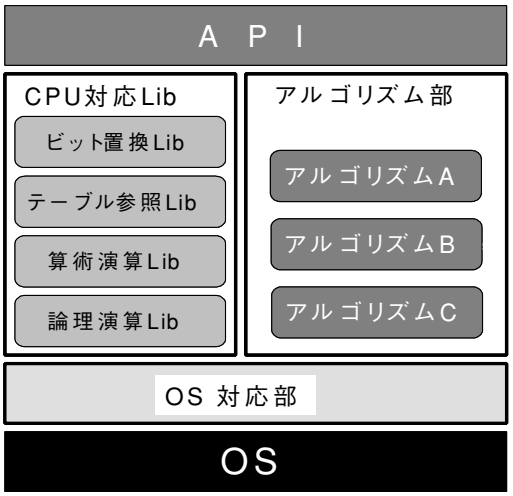

図3　暗号化ミドルウェアの構成

コンパクト化と処理の軽量化

(1) ライブラリ化によるコードサイズの縮小

暗号のための符号化，復号化処理には，算術演算，論理演算，テーブル参照，およびビット置換が主に用いられる。近年ではCPU性能向上によりソフトウェアによる実装が多くなったため，論理演算およびテーブル参照による実装が多く用いられている。さらに，ほとんどの暗号化アルゴリズムは“ラウンド”と呼ばれる繰り返し処理によって暗号化強度を高めている。たとえばDESの場合，64ビットの平文ブロックに対して同じ処理を16回（16ラウンド）行う必要があり[3]，このためプログラム中には“while”や“for”などの繰り返しを含むラインが全体の1割近く存在する。

つまり，暗号化は主に算術演算，論理演算，テーブル参照，ビット置換，およびその繰り返しによって構成されていると考えることができる。このため，本ソフトウェアでは，各アルゴリズムの類似処理をライブラリ化することにより，処理の簡素化とコードサイズの縮小を目指した。さらに，繰り返し処理中に多く使用されるコードをライブラリとして局所的に集めることは，キャッシュのヒット率向上にもつながり，結果として処理能力の改善につながっている。

(2) 処理能力改善のための最適化箇所の抽出

DESは，その生い立ちが古く当初はハードウェアによる実装に主眼がおかれていたためビット置換処理が多く含まれている。このためDESのソフトウェアによる実装は他のアルゴリズムと比較して複雑な論理となり，高い処理能力が要求される。このようなアルゴリズムを，組込み機器で使用される非力なCPU上に実装するためには，演算処理の最適化が必要になる。そのためにはまず，暗号化処理の中でどの様な演算が多く使われているかを調べ，どのような処理を最適化する必要があるかを特定する必要がある。

我々は，DES，AESおよびBlowfishの各アルゴリズムを試験的にC言語で実装し，そのコードを調査した。その結果，総ライン数の約40%において算術演算または論理演算が行われていた。さらに，これらのコードにおける各演算の出現数を調査した結果，**図4**に示すように論理演算とシフト演算が全体の64%を占めていることが分かった。

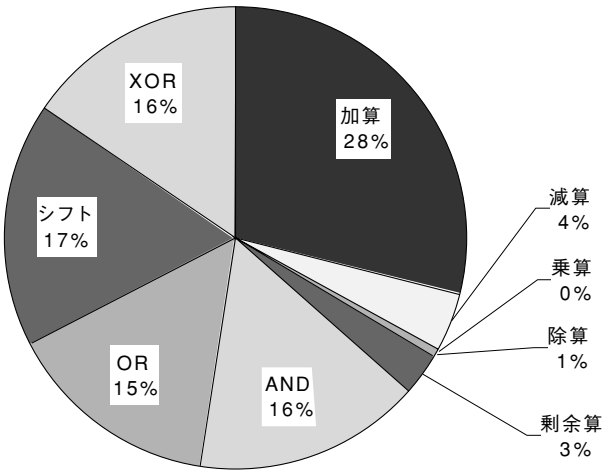

図4　暗号化で使用される演算の割合

これらのことから，本ソフトウェアではCPU依存処理を先に述べた4つのライブラリにカテゴリ化して（**表2**），その中で特に出現頻度の高い論理演算，シフト演算および，それらの繰り返し処理に着目し，これを各CPUに対応した効率の良いものに最適化することで，性能向上とコードサイズの縮小化を図った。

表2　ライブラリのカテゴリと機能概要

カテゴリ	機能概要
論理演算	連続した論理演算，配列の論理演算，長いビット長用のシフト及びローテート処理
算術演算	連続した算術演算，配列の算術演算処理
テーブル参照	S-BOX* などの処理に使用される連続したテーブル参照と演算処理
ビット置換	主としてDESで使用されるビット置換処理

*Substitution Box:置換ボックス,暗号の基本処理である置換を行う。
主に入力値に対応したテーブルを索引することにより入力値を変換する。

ま と め

ユビキタスネット社会の実現に向けて，組込み機器に対するセキュリティ要件は，今以上に多くなる。このため，当社では本ソフトウェアのさらなる高機能化，高性能化に取り組んでいる。

今回は，主として共通鍵ブロック暗号アルゴリズムの実装に焦点を当てたが，実際の製品に組み込む場合は，さらに，RSA*5) 等の公開鍵暗号アルゴリズム，ハッシュ関数，擬似乱数発生系などを実装する必要がある。今後は，これら機能の充実と対応プラットフォームの拡大を中心に開発を進めていく予定である。 ◆◆

■参考文献

1）総務省:“情報通信白書平成16年版”
http://www.johotsusintokei.soumu.go.jp/whitepaper/ja/cover/index.htm
2）独立行政法人 情報処理推進機構:“国内で入手可能な暗号関連製品リスト”
http://www.ipa.go.jp/security/fy14/crypto/cmvp/needs.html
3）ブルース･シュナイアー：暗号技術大全，初版，ソフトバンクパブリッシング株式会社，p.305，2003年

●筆者紹介

松山淳：Atsushi Matsuyama. 沖通信システム株式会社 第1ネットワークグループ ソフトウェア開発第2部
細貝和彦：Kazuhiko Hosogai. 沖通信システム株式会社 第1ネットワークグループ ソフトウェア開発第2部

TIPS 【 各暗号化方式について 】

DES（Data Encryption Standard）
1977年にNISTによって連邦情報処理規格（Federal Information Processing Standard）に採用された64ビットのブロック暗号。鍵長は56ビット。

Triple DES
DESの暗号化強度を高めるために,1つのブロックに対して56ビットの鍵を2つ（または3つ）使用して,3回の暗号化処理を行う。鍵長は112（56×2）ビットまたは168（56×3）ビット。

AES（Advance Encryption Standard）
ベルギーのJoan DaemenとVincent Rijmen によって考案されたRijndael 。2000年にNISTが公募の中からAESに選定した。鍵長とブロック長さはそれぞれ128,192,256ビットから選択。

RC2
Rsa Security Inc.のRon Rivestによって考案された,可変鍵長の64ビットブロック暗号。

RC5
RC2の後継暗号として,Ron Rivestによって考案された。ブロック長,鍵長,段数がいずれも可変という特長を持つ。

Blowfish/Twofish
双方ともBruce Schneierの考案した暗号化方式。Blowfishは,64ビットブロック暗号で,鍵長は448ビットまでの可変。Twofishは128,192,256ビットの鍵長を持つ,128ビットブロック暗号。

MISTY1
MISTYは三菱電機が1992年に発表した64ビットブロック暗号アルゴリズム。MISTYにはMISTY1とMISTY2がある。

SXAL
日本のローレルインテリジェントシステムズ社が開発した64ビットブロック暗号。1995年ISO9979に登録された。

IDEA（International Data Encryption Argorithm）
LaiとMasseyがAscom社と共同開発した暗号アルゴリズムで128ビットの鍵長を持つ64ビットブロック暗号。

*5）RSAはRsa Security Inc.の登録商標です。