

次世代共通鍵暗号 AES の小型回路アーキテクチャ

3G-2

高野光司 佐藤証 森岡澄夫 宗藤誠治
日本アイ・ピー・エム株式会社東京基礎研究所

1 はじめに

米国連邦政府標準暗号そしてデファクト・スタンダードとして 20 年以上にわたり使用されてきた DES(Data Encryption Standard)に代わり、2000 年 10 月にベルギーから提案された Rijndael^[1] が次期米国標準である AES (Advanced Encryption Standard) に選定された。Rijndael (以下 AES) が有する SPN 構造は並列性に優れるが、DES をはじめとする多くの共通鍵ブロック暗号が有する Feistel 構造に比べて一般に回路規模が大きくなるといわれている。これは図 1 に示したように Feistel 構造がデータの半分を F 関数で変換して、残り半分と XOR するのに対して、SPN 構造は全データを一度に変換し、また、前者は暗号化と復号化で同じ F 関数を使えるのに対して、後者は復号化において暗号化の逆変換を行う別のデータパスが必要となるためである。

本論文では、128 ビットデータを 32 ビット毎に分け 1 ラウンドあたり 5 クロックで処理し、かつ暗号化と復号化を同じデータパスで実行する小型の AES 回路アーキテクチャを提案する。

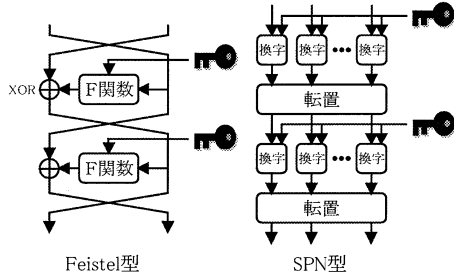


図 1 Feistel 構造と SPN 構造

2 AES のアルゴリズム

図 2 にデータ長 128 ビット、鍵長 128 ビットの AES の暗号化アルゴリズムを示す。11 箇所の AddRoundKey で、図 3 の鍵スケジューリングによって順次生成されるラウンド鍵が 128 ビットずつ入力され、データと XOR される。他の基本変換 SubBytes, ShiftRows, MixColumns においては、128 ビットを 16 ブロックに分けた 1 バイト (8 ビット) が処理の基本単位となる。

SubBytes 変換は 1 バイト入出力の非線形変換を行う S-Box を 16 個並べたもので、各 S-Box では 2 の拡大体 GF(2⁸) 上の乗法逆元演算に続き affine 変換が実行される。これと同じ 4 バイト分の S-Box が、図 3 の鍵スケジューリングでも使用される。ShiftRows は 16 バイトのデータを 4×4 行列にならべた各行を

0~3 バイト巡回シフトする。MixColumns では、各列の 4 バイトを 3 次多項式の係数とみなし、多項式

$$c(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

と乗じてから多項式 $x^4 + 1$ で剰余がとられる。

復号化では各基本演算の逆変換を順に実行する。InvMixColumns では

$$c^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$$

と各列を乗じ、InvShiftRows では暗号化と逆向きの巡回シフトを、InvSubBytes 変換では、SubBytes の affine 変換の逆変換

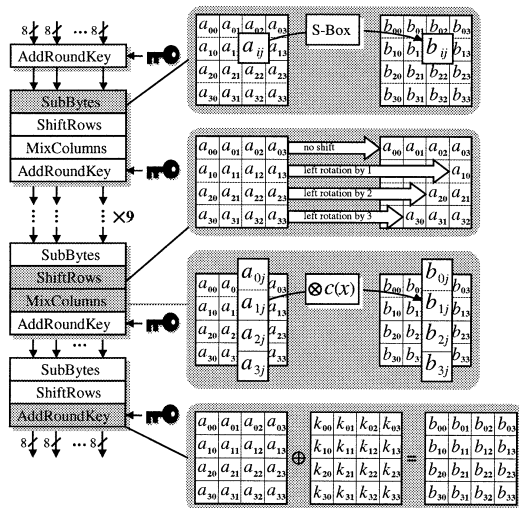


図 2 AES の暗号化アルゴリズム

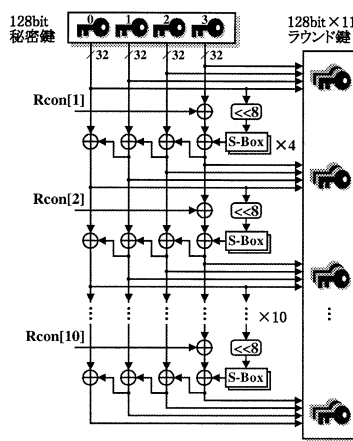


図 3 暗号化における鍵スケジューリング

A Small Hardware Architecture of the Advanced Encryption Standard
Kohji TAKANO, Akashi SATOH, Sumio MORIOKA, Seiji MUNETOH
Tokyo Research Laboratory, IBM Japan Ltd.
1623-14, Shimotsumura, Yamato-shi, Kanagawa 242-8502, Japan

が GF(2⁸)上の逆元演算の前に行われる。AddRoundKey は XOR なので逆変換も同じ XOR であるが、ラウンド鍵のスケジューリングを逆順に行う必要がある。

3 データパス・アーキテクチャ

図 4 に提案する AES ハードウェア・アーキテクチャを示す。この構成は 128 ビットのデータを一度に処理する代わりに、32 ビットずつ 4 回に分けることで回路サイズを削減するものである。AES は 32 ビット CPU における高速ソフトウェア実装を前提としているため、このように 32 ビット単位で処理を効率よく分割することができる。しかし処理のバス幅を 32 ビット以下にすると、回路削減効果は急激に低下する。これは、MixColumn や InvMixColumn の演算に用いる値を 32 ビットにそろえるためのテンポラリー・レジスタやセレクタが増加し、なおかつ制御回路のエリアがデータパス部に対して大きくなるからである。

以下、提案するアーキテクチャにおけるデータフローと、各部の詳細を説明する。暗号化においてデータは 128 ビットのシフトレジスタに保持され、1 サイクルで全ビットの ShiftRows (復号化では InvShiftRows) オペレーションが実行される。そして、5:1 セレクタを通じて 32 ビットごとに SubBytes (または InvSubBytes) 変換, MixColumns (または InvMixColumns) へと続く。ここで ShiftRows と SubBytes の順序が図 2 と異なっているが、この変更は演算結果には影響を与えない。

SubBytes 変換中の S-Box のハードウェア実装では、通常は affine 変換と同一化した 256 (=2⁸) エントリ × バイトのテーブルが用いられる。しかし、その方法では組み合わせ回路部の大半が S-Box で占められてしまい、また暗号化と復号化で別々のテーブルが必要となる。そこで、本アーキテクチャでは、逆元演算と affine 変換を切り離して実装し、逆減演算部を暗号化と復号化で共有している。そして、暗号化では

逆元演算 $x^{-1} \rightarrow$ affine 変換 \rightarrow MixColumns

のパスが、復号化 (InvSubBytes) では

affine 逆変換 $\rightarrow x^{-1} \rightarrow$ InvMixColumns

が選択される。復号化において InvMixColumns は本来 InvSubBytes の前に行われるが、上記のような演算順序に変更することでセレクタを省略し、クリティカルパスを短縮できる。この変更により鍵スケジューリング部にも InvMixColumns が必要となるが、暗号化/復号化部の InvMixColumns が MixColumns と共有されるため回路の増加はほとんどない。

なお、以上の SubBytes 変換は、鍵スケジューリングで 1 ラウンド分の 128 ビットの副鍵を生成するためにも使用される。従って 1 ラウンド分の処理には、SubBytes (または InvSubBytes) が暗号化/復号化部に 4 サイクル、鍵生成部に 1 サイクル必要となり、計 5 サイクルを要する。鍵スケジューリング部が SubBytes 変換を行っている間、暗号化/復号化部では ShiftRows (または InvShiftRows) 変換が並列に実行される。

以上が 1 ラウンド分の処理の概要である。そのほか、暗号 / 復号化の前処理として、入力データとラウンド鍵との 128 ビット XOR (AddRoundKey 処理) が行われ、最終ラウンドでは

MixColumns (または InvMixColumns) が省略される。これは図 4 の AddRoundKey 直前の 5:1 セレクタで切り替えられる。前処理の AddRoundKey も 32 ビットずつ処理されるため 4 サイクルを要するが、このときは変換前の鍵をそのまま使用するため、S-Box による変換は行われない。このことから暗号化全体のサイクル数は $4 + 5 \times 10 = 54$ となる。また、復号化も同じサイクル数であるが、最初に一度だけ 10 サイクルかけて暗号化最終段の副鍵を生成しておく必要がある。

図 4 では 4 バイト分の S-Box を用いた最小構成のみを示したが、8 バイトあるいは 16 バイト分の S-Box を用意したり、鍵スケジューリング部でも独立に 4 バイトの S-Box を持たせたりすることによって並列度をあげ、スループットを向上させることができる。0.11 μ m CMOS スタンダードセル・ライブラリによる最小実装では 6.5K ゲートで 222Mbps が、また高速実装では 20.7K ゲートで 2.3Gbps が実現された。このように本アーキテクチャは、組み込み用の小型実装からハイエンド認証サーバーなどに必要な高速実装まで幅広いアプリケーションに柔軟に対応することが可能である。

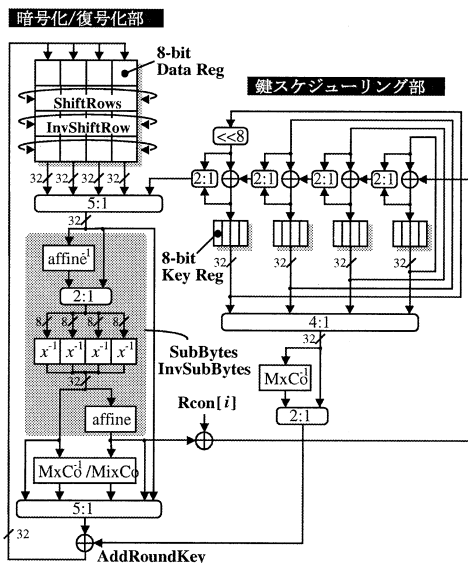


図 4 提案する AES ハードウェア・アーキテクチャ

4 おわりに

AES の小型回路実装を目的に、暗号化 / 復号化 / 鍵スケジューリングの各処理間での S-Box の共有や、MixColumns と InvMixColumns のマージ等のリソース共有を行ったアーキテクチャを提案し、このアーキテクチャによって小型実装 (6.5K gates) から高速実装 (2.3Gbps) まで柔軟に対応できることを示した。

文献

- [1] “Advanced Encryption Standard (AES) Development Effort”, <http://csrc.nist.gov/encryption/aes/index2.html>.