Memoirs of the Faculty of Science
Kochi University (Information Science)
Vol.29 (2008), No.12

暗号化アルゴリズムのハードウェア化手法の提案

A Proposal of Encryption Hardware Algorithm

高知大学理学部情報科学科情報コース　村岡研究室
趙　朔　　村岡　道明

1. まえがき

社会の高度情報化に伴い、データをより安全に保護するため暗号技術が重要になるが、データ量の増加に伴い計算機処理の負荷増加が問題となる。本研究では、標準暗号技術の１つである DES 暗号について高速化を目的としたハードウェア化手法を提案する。ハードウェア化手法としては、ソフトウェア言語で記述した暗号アルゴリズムをハードウェア記述言語に書き換えることにより高速化を図る。

2. 提案手法

DES は、長さ 64 ビット列を長さ 56 ビットの鍵ビット列で暗号化し、長さ 64 ビットの暗号文ビット列を出力する。このアルゴリズムを次に示し、それぞれ初期転置、鍵の生成、F 関数、最終転置である。C 言語で作成したプログラムの時間を計るにより、表１で示したように、F 関数と鍵の生成部分は長い処理時間を要する。

表１．DES 処理の工程別処理時間

各処理時間				
処理データの大きさ (KB)	初期転置 (sec)	鍵の生成 (sec)	F関数 (sec)	最終転置 (sec)
600KB	0.035	0.729	4.23	0.035
1200KB	0.068	1.416	7.261	0.068
2400KB	0.139	2.816	15.613	0.139

本論文では、最も処理時間を要する F 関数部分のハードウェア化を取り上げた。ハードウェア化に関して、パイプライン化、パラレル化などの手法があるが、この論文では、F 関数のテーブル方式による高速化をした上でさらに、ハードウェア化を行い、そして、パイプライン化を検討した。図１の暗号化アルゴリズムの手順を説明する。

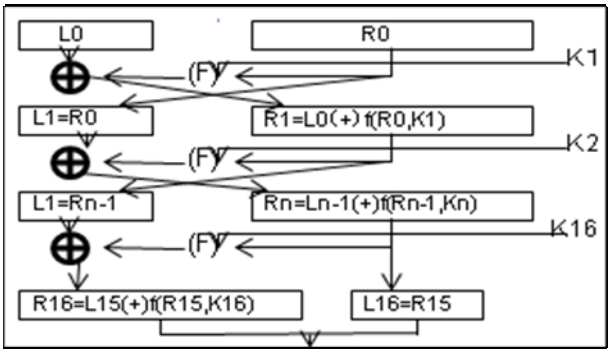

図１．暗号化アルゴリズムの流れ

(ｉ)次段左ビット L1 は上段の右ビット R0 から直接入れる。
(ii)32bit の R0 入力データを決められた関数により 48bit に拡大する。
(iii)拡大した R0 関数が K と XOR を計算し、結果 6 ビットずつ 8 個ビット列 S に分ける。
(iv)各ビット列を S-box で処理し、この値が 4 ビットの値を出力する。
(ｖ) 8 個の 4 ビットの値を一つの 32 ビット列として決められた P により転置する。得られた値と左ビットの XOR を求め、それを右ビットとする。
(vi)(ｉ)～(ｖ)まで 16 回繰り返しを行い、最終転置を行うことにより暗号化した値を得る。

F 関数の処理については、look-up table を使用した DES 暗号処理の高速化に関する方式を採用した。さらに、DES 暗号のハードウェア化を行った。図 2 で示すように、暗号化処理、鍵の生成、制御回路、入力データ制御の四つのブロックより構成された。これにより、DES の高速化が可能となる。さらに、暗号化部分のハードウェア化を行い、処理時間がより短縮される。F 関数のパイプライン化については、16 回の繰り返しにより、16 段のパイプライン構成によりさらに高速化が可能である。

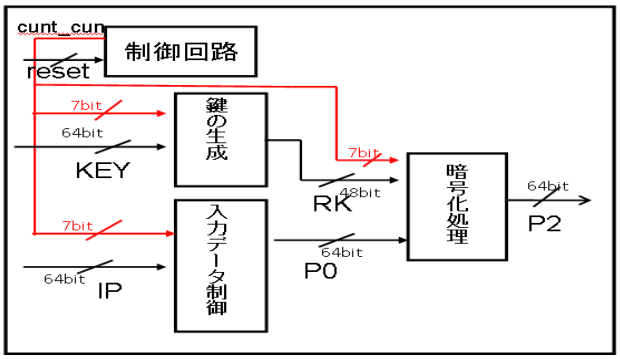

図２．DES 暗号のハードウェア化構造

3. 高速化の考察

[1]では、F 関数の look-up table を使用した方式には、テーブル方式により、DES の処理時間を削減できる。さらに、ハードウェア化をすると、そしてパイプライン化することにより、DES の処理時間が短縮される。高速化率を表す処理時間評価式(Speedup)を式(１)に示す。

$$\text{Speedup} = \frac{T}{t\max\times(n+3)} \qquad \ldots\ldots(1)$$

ここで T はソフトウェア化した DES の暗号化部分処理にかかる時間であり、 t max は各ステージの中で一番処理時間を要するステージの時間である。 n は暗号化部分について繰り返し回数で、n =16 である。ソフトウェア処理の場合には、テーブル方式による処理時間が 90.5%に改善されている。一方、ハードウェア処理の場合については、処理時間 97.7%の削減をでき、そして、パイプライン化により、処理時間 26.2%の削減を見通す。

4. あとがき

暗号化アルゴリズムの高速化を目的とした、ハードウェアアルゴリズムのパイプライン化を提案した。提案により、DES アルゴリズムの高速化ができる。今後は DES のハードウェアアルゴリズムの更なる高速化を目的として FPGA 化に向いたアルゴリズムについても改良、検討を行う。