C-004

Camellia 暗号回路の鍵固定によるハードウェア特殊化
Key specific Hardware specialization for Camellia Encryption

松岡 俊佑† 日野善規‡ 市川 周一§
Shunsuke Matsuoka Yoshiki Hino Shuichi Ichikawa

1. はじめに

論理回路の入力の一部を定数に固定すれば，入力値に応じて論理回路を最適化することで，回路規模を削減し動作速度を改善することができる．こうした技術をハードウェア特殊化と呼ぶ．こうして得られた回路は，ある入力値に対して特化した回路となるため，異なる入力値に対しては新しいハードウェアを生成しなければならない．従ってハードウェア特殊化回路を実装するには FPGA 等の再構成可能な論理デバイスが適している．ハードウェア特殊化技術の応用例として，ディジタルフィルタの係数を固定した回路[1]や，DES 暗号や AES 暗号の鍵を固定した回路[2][3]などが報告されている．

本研究では，共通鍵暗号の一つである Camellia 暗号を対象として，鍵を固定したハードウェア特殊化回路について検討し，FPGA による実装評価を行った．

2. Camellia 暗号アルゴリズム

Camellia暗号[4]は，2000 年に三菱電機とNTTで共同開発され，電子政府推奨暗号(CRYPTREC)や，ISO/IEC国際標準暗号，インターネット標準暗号(IETF)に選定された世界標準の共通鍵暗号の一つである．ブロック長は，AES暗号と同様に,128bit,192bit,256bitから選択できる．基本構造はDES暗号と同じくFeistel構造を採用しており，データを左右に半分ずつ分けて処理していく．また，暗号化と複合化は順序を逆に処理していくことで，AES暗号のように逆関数を用意することなく，同じルーチンで実現できる．ブロック長 128bitのCamellia暗号アルゴリズムを図１に示す．F関数と排他的論理和からなる全 18 ラウンドの処理と，6 ラウンドと 12 ラウンドの後に挿入されるFL⁺/FL⁻関数，および最初のラウンド前の排他的論理和処理(Pre-Whitening)と最後のラウンド後の排他的論理和(Post-Whitening)，および副鍵生成部から構成される．F関数は次のような式で定義される．

$$Y_{(64)} = P\left(S\left(X_{(64)} \oplus k_{(64)}\right)\right) \qquad (1)$$

上式のP関数は排他的論理和処理と，８ビット入力８ビット出力の非線形変換であるS関数からなる．鍵拡張部では，入力暗号鍵Kを基にして，暗号生成部のラウンド処理と同じくF関数と排他的論理和を表１に示した定数を用いて繰り返し処理していくことにより中間鍵KAが生成される．入力暗号鍵Kと中間鍵KAをローテーションシフトすることで 26 個の 64bit副鍵（kw_1～kw_4, kl_1～kl_4, k_1～k_{18}）が生成される．

表１　鍵生成部の定数

$\Sigma_{1(64)}$	0xA09E667F3BCC908B
$\Sigma_{2(64)}$	0xB67AE8584CAA73B2
$\Sigma_{3(64)}$	0xC6EF372FE94F82BE
$\Sigma_{4(64)}$	0x54FF53A5F1D36F1C

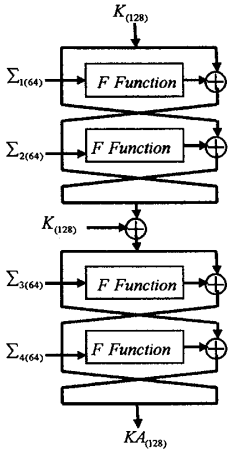

図２　鍵生成部

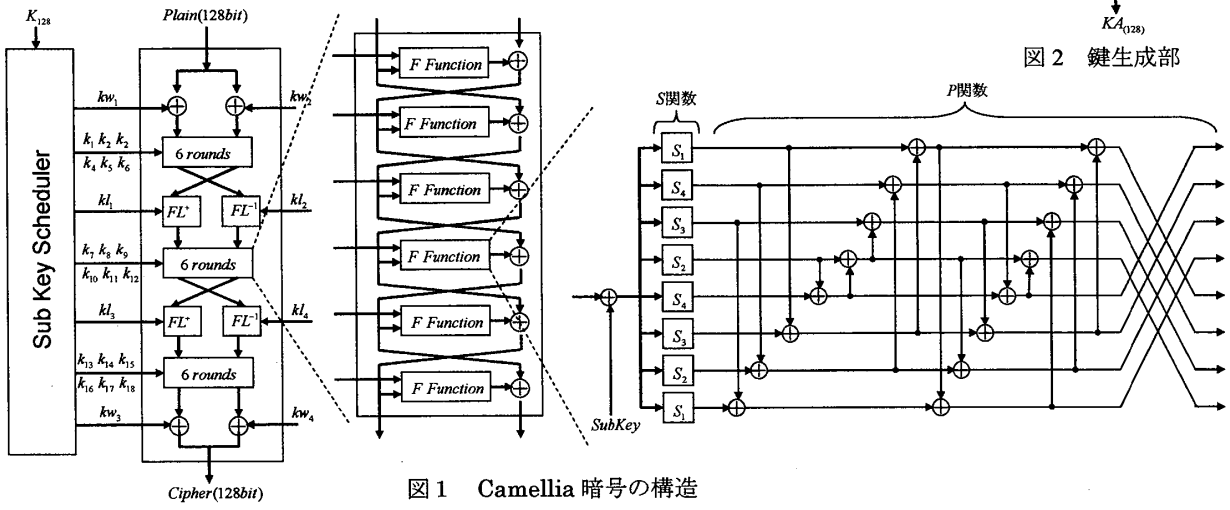

図１　Camellia 暗号の構造

† 旭川工業高等専門学校
Asahikawa National College of Technology
‡ 中部電力 Chubu Electric Power Co.,Inc
§ 豊橋技術科学大学　Toyohashi University of Technology

3. ループ型 Camellia 暗号回路

図 1，2 に示した Camellia 暗号アルゴリズムをハードウェア回路で実現するためのアーキテクチャ方式として，Pre-Whitening と Post-Whitening 処理するための EX-OR，FL 関数回路，全 18 ラウンドを処理する回路全てを，間にレジスタを配置して実装するアンロール型アーキテクチャ[5]や，全 18 ラウンドでは同じ処理が繰り返されることから，一回分のラウンドを処理する回路と，FL 関数回路，および Pre-Whitening と Post-Whitening 処理するための EX-OR，および副鍵を生成するための鍵拡張回路を用意し，レジスタに中間データを保存しながら繰り返しループさせるループ型アーキテクチャがある[6][7][8]．本研究では，ループ型 Camellia 暗号アーキテクチャ方式のオーソドックスな回路として、青木らによって設計された Camellia 暗号回路[9]を評価の基本(original)として用いることにした．図 3 は，original 回路のブロック図を示している．データレジスタ(Data_reg)への入力をマルチプレクサで切り替え，EX-OR，ラウンド処理回路，FL 関数回路で処理を順次行うことにより暗号化と複合化がそれぞれ 23 クロックで実行される．各処理の入力として用いられる副鍵は，鍵拡張部(Key Scheduler)で生成される．鍵拡張部は，入力暗号鍵を格納するレジスタ KL と，鍵拡張処理によって生成される中間鍵を格納するレジスタ KA，および各レジスタの出力を 15bit,または 17bit 左右にローテンションシフトさせるシフターからなどから構成される．中間鍵を生成するさいに，ラウンド処理が繰り返し実行されるが，これは，暗号化・複合化で用いられるラウンド処理回路が共用されている．

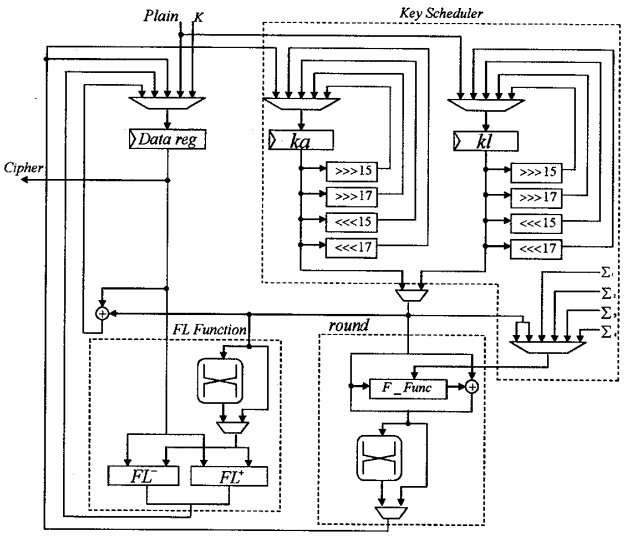

図 3　ループ型 Camellia 暗号回路（original）

4. Camellia 暗号回路のハードウェア特殊化

4.1 副鍵固定回路(fixed_subkey)

前節で述べたループ型Camellia暗号回路(original)を基にして，入力暗号鍵を定数に固定し場合の，ハードウェア特殊化回路を２種類提案する．まず，一つ目の回路として，入力暗号鍵が決まれば，26 個の副鍵は事前に計算できることから，すべての副鍵を定数に固定した副鍵固定回路(fixed_subkey)を提案する(図 4)．original回路の鍵拡張部の換わりに副鍵（kw_1～kw_4, kl_1～kl_4, k_1～k_{18}）を選択するマルチプレクサを配置する．

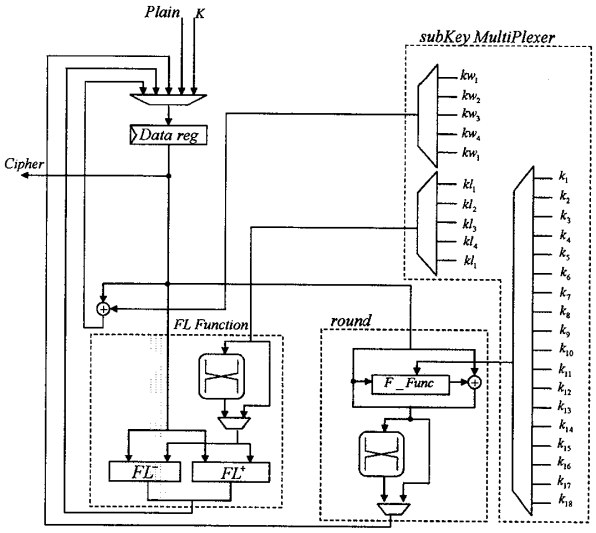

図 4　**副鍵固定回路**（fixed_subkey）

4.2 F 関数 Ex-OR 簡単化回路(F_Func_xor_collapse)

排他的論理和処理(Ex-OR)は，一方の入力が'0'ならばもう一方の入力の値がそのまま出力され，一方の入力が'1'ならばもう一方の入力の値が反転されて出力される．すなわち，片方の入力を定数に固定することによって，EX-OR素子を省略またはNOT素子に置き換えることができ，論理回路が簡単化される．Camellia暗号回路では図 1 に示すとおり，F関数の中で副鍵との排他的論理和処理が行われている．ここでは，全ての副鍵を定数に固定し，F関数内の副鍵との排他的論理和(Ex-OR)を簡単化した回路(F_Func_xor_collapse)を提案する．Pre-Whitening・Post-Whitening処理と，FL関数への入力副鍵(kw_1～kw_4, kl_1～kl_4)は，fixed_subkeyと同じく定数に固定しマルチプレクサで選択する．F関数回路の内の排他的論理和(Ex-OR)に置き換えて，定数に固定した副鍵（k_1～k_{18}）ごとに回路を簡単化し，マルチプレクサで選択する(図 5).

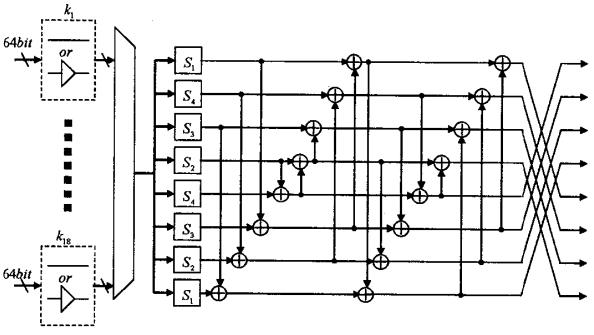

図 5　F_Func_xor_collapse の F 関数回路

5. FPGA による実装評価

前節で述べた Camellia 暗号回路を，Xilinx 社製 EDA ツールの ISE11.1 で論理合成および配置配線した．ターゲットデバイスは XC3S400A とした．論理合成オプションの設定項目の一つである Optimization Goal は Speed 優先と Area 優先の 2 つの条件にについてそれぞれ評価を行い，その他の設定項目はデフォルトのままとする．EDA ツールの動作環境は，CPU が Intel Core2Duo T7250，メモリが 1GB，OS が Microsoft Windows XP Professional Service Pack2 の仕様の PC を用いた．

他の共通鍵暗号との比較のために，AES 暗号回路についても，論理合成および配置配線を行った．AES 暗号回路のハードウェア特殊化については，[3]ですでに報告されているが，ここでは，インターネット上で公開されている Megarajan Perk のループ型 AES 暗号回路 [10]を評価の基本(original)として用いることにした．ただし，AES 暗号は暗号化と複合化で別な回路が必要となるため複合化回路は実装せず暗号化回路のみとする．Megarajan Perk の暗号化回路(original)を基に，Camellia 暗号回路のハードウェア特殊化と同様な方法で，ラウンド鍵を固定した回路(fixed_roundKey)，とラウンド鍵との排他的論理和処理を簡単化した Ex-OR 簡単化回路(xor_collapse)を設計した．

Camellia 暗号回路および AES 暗号化回路の鍵を固定したハードウェア特殊化回路は，鍵の定数値ごとに異なる回路が生成される．そこで，ランダムな暗号鍵を 20 個生成し，それぞれの暗号鍵ごとにハードウェア特殊化を行う.生成された回路ごとに論理合成・配置配線を行い，その平均値を評価結果とする．評価項目は，論理規模(Logic Scale)，最大動作周波数(Max.Freq)，Througput および，ハードウェア特殊化回路では，暗号鍵を入れ替えるたびに再実装する必要があることから，論理合成・配置配線にかかる時間(Gene.time)とした．

以上の評価環境で AES 暗号化回路について論理合成・配置配線した結果を表 2 に示す．評価の基本として用いた Megarajan Perk 暗号化回路(original)と比較して，鍵を固定したハードウェア特殊化回路 fixed_roundKey，および xor_collapse は，配置配線生成時間，論理規模，最大周波数，throughput がともに改善されることが確認できた．ただし，xor_collapse については fixed_roundKey に対して，いずれの評価項目について改善がみられなかった．

次に，Camellia 暗号回路の評価結果を表 3 に示す．評価の基本として用いた original 回路と比較して，fixed_subkey は，Optimization Goal が Speed 優先の場合は論理規模は 27%減少，最大周波数は 72%増加，throughput は 72%増加した．Area 優先の場合は論理規模が 30%減少，最大周波数は 57%増加，throughput は 56%増加した．original 回路と比較して，F_Func_xor_collapse は，Optimization Goal が Speed 優先の場合は論理規模は 23%減少，最大周波数は 57%増加，throughput は 56%増加した．Area 優先の場合は論理規模が 17%減少，最大周波数は 77%増加，throughput は 77%増加した．F_Func_xor_collapse は fixed_subkey と比較して，論理規模は，Speed 優先の場合は 5%増加，Area 優先の場合は 16%増加しているが，最大周波数は Speed 優先の場合は 13%，Area 優先の場合は 12%増加している．論理合成・配置配線にかかる時間は，original と比較して，fixed_subkey および F_Func_xor_collapse は 14%～25%程度短縮された．以上の結果から，提案した fixed_subkey と F_Func_xor_collapse は，original 回路に対して，論理合成・配置配線生成時間，論理規模，最大周波数，throughput がともに改善されることが確認できた．また，F_Func_xor_collapse は，fixed_subkey に比べて論理規模はやや増加するものの，最大周波数が改善されることが確認できた．

表 2　AES 暗号の評価結果

	Design	Gene.Time (sec)	Logic Scale (slice)	Max. Freq. (MHz)	Throughput (Mbps)
speed	original	138	2327	154.9	1803
	fixed_round_key	122	1690	131.3	1527
	xor_collapse	122	1753	124.5	1449
Area	original	129	1828	124.6	1450
	fixed_round_key	111	1484	121.7	1416
	xor_collapse	117	1626	108.9	1267

表 3　Camellia 暗号の評価結果

	Design	Gene.Time (sec)	Logic Scale (slice)	Max. Freq. (MHz)	Throughput (Mbps)
speed	original	177	1909	71.7	399
	fixed_sub_key	117	1406	123.5	687
	F_func xor_collapse	118	1476	141.3	786
Area	original	141	1751	66.1	368
	fixed_sub_key	107	1227	104.1	580
	F_func xor_collapse	116	1458	117.4	653

6. おわりに

本研究では，Camellia 暗号回路について，暗号鍵を定数に固定したハードウェア特殊化回路を提案した．EDA ツールにより，論理合成・配置配線した結果，ハードウェア特殊化する前の回路と比較して，論理合成・配置配線に要する時間，論理規模，最大周波数，throughput ともに改善されることが確認できた．

参考文献

[1]手塚康瑛,市川周一,野田善之,“ディジタルフィルタのハードウェア特殊化と制振制御への応用,” 電子情報通信学会技術研究報告, vol. 109, no. 395, pp.83-88 , (2010).

[2] J.Leonard and W.H.Mangione-Smith, “A case study of partially evaluated hardware circuits: Key-specific DES”, Proc.Field Programmabel Logic and Applications(FPL’97), LNCS1304, pp151-160, (1997).

[3] Ryoichiro Atono, Shuichi Ichikawa,: "Design and Evaluation of Data-dependent Hardware for AES Encryption Algorithm",IEICE Transactions on Information and Systems, Vol. E89-D, No.7, pp. 2301-2305 (2006).

[4] K.Aoki, T.Ichikawa, M.Kanada, M.Matsui, S.Moriai, J.Nakajima, and T.Tokita, ”Specification of Camellia a 128bit Block Cipher”, https://info.isl.ntt.co.jp/crypt/camellia/specifications.html/01jspec.pdf , (2000).

[5] Daniel Denning, James Irvine, Malachy Devlin, ``A HIGH THROUGHPUT FPGA CAMELLIA IMPLEMENTATION," PhD Research In Micro-Electronics & Electronics, (2005).

[6] Huiju Cheng and Howard Heys, " Compact Hardware Implementaiton of the Block Chiper Camellia with Concurrent Error Detection", Canadian Conference on Electrical and Computer Engineering, pp1129-1132, (2007)

[7] Xianwei Gao, Erhong Lu, LiLi Kun Lang, "LUT-based FPGA Implementation of SMS4/AES/Camellia", Fifth IEE International Symposium on Embedded Computing pp73-76, (2008),

[8] Panasayya Yalla and Jens-Peter Kaps, "COMPACT FPGA IMPLEMENTATION OF CAMELLIA", International Conference on Field-ProgrammableLogic andApplications, pp658-661, (2009)

[9] 東北大学青木研究室，”Camellia IP Core”，http://www.aoki.ecei.tohoku.ac.jp/crypto/web/cores.htm

[10] Megarajan perk ,http://islab.oregonstate.edu/koc/ece575/03Project/Megarajan-Park/