付録 F：Camellia の高階差分について

1 はじめに

本資料は、電子政府事業に用いる暗号方式に関する応募資料 (暗号技術仕様書 Camellia 及び自己評価書 Camellia[NTT,Mitsubishi]) に基づき高階差分／補間攻撃の立場で詳細評価を行うために作成したものである。アルゴリズムに関する詳細は省略したので、必要に応じ、仕様書を参照して頂きたい。

2 Camellia のアルゴリズム

Caqmellia の仕様は下記のとおりである.

- ブロック長:128bits
- 鍵長:128,192,256bits
- 段数:18, 24(ただし，18 段は鍵長 128bit．24 段は鍵長 192 及び 256bit)

2.1 データ攪拌部

図 1 に Cammelia のデータ攪拌部を示す。基本構造は、Feistel 型であり、鍵長に応じ、18 段又は 24 段の処理が行われる。6 段毎に補助関数（FL/FL^{-1} 関数）の挿入され、第 1 段の直前及び最終段の直後において、拡大鍵が排他的論理和加算されている。図において添え字の括弧内の数字はビット幅を表す。k で始まる変数は、鍵生成部から供給される拡大鍵である。

3 Camellia の構成要素

3.1 F 関数

F 関数は 64 ビット入出力の関数であり、拡大鍵 64 ビットを用いて入力データを攪拌する。8 ビット入出力の S 関数 8 個を並列に並べ、その出力をバイト単位の P 関数で攪拌する。(図 2)。

3.2 FL/FL^{-1} 関数

FL/FL^{-1} 関数は Feistel 構造の 6 段毎に挿入される補助関数であり、64 ビット入出力、拡大鍵 64 ビットの関数である (図 3)。128 ビット幅のデータは、左右半分づつ別の FL/FL^{-1} 関数を通る。関数の設計目標は、F 関数の差分／線形特性を損なうことなく、将来の未知の攻撃を防ぐこ

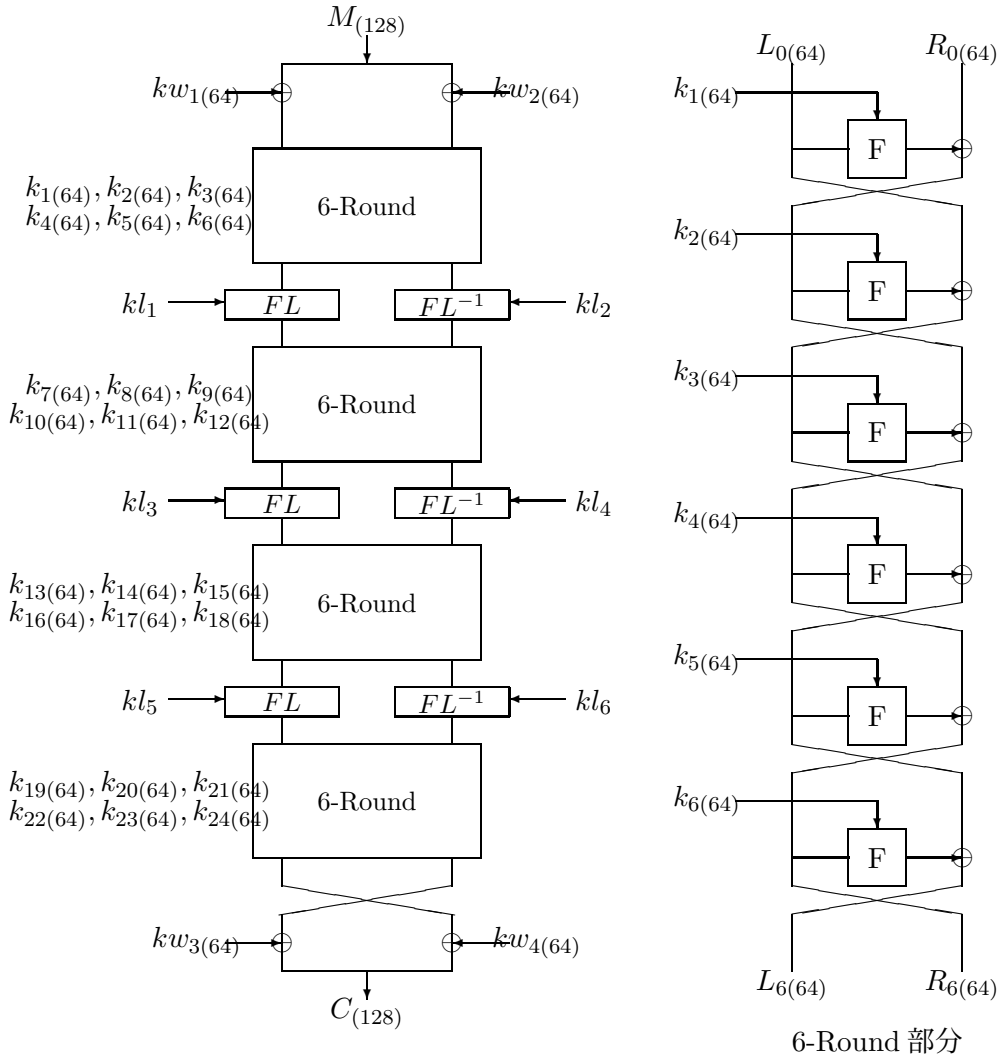


図 1: Camellia の暗号化過程

とであり、MISTY の FL 関数の設計方針を踏襲している。FL/FL^{-1} 関数は、入力に関し線形である。

3.3 S 関数及び s-box

S 関数は、8 ビット入出力の全単射関数であり、$GF(2^8)$ 上の逆数関数をアフィン変換して得られる、一つの S-box テーブル s_1 を使い廻しして、4 種類の S 関数 s_1, s_2, s_3, s_4 を構成している。次式の関数 f は、拡大体の逆数関数、g は多項式基底による 8 次元ベクトルへの変換 h は線形変換であり、定数は 16 進定数を表す。入力は $x_{(8)}$、出力は $y_{(8)}$ である。

$$
\begin{aligned}
s_1 &: \ y_{(8)} = h(g(f(0xc5 \oplus x_{(8)}))) \oplus 0x6e \\
s_2 &: \ y_{(8)} = s_1(x_{(8)}) <<< 1
\end{aligned}
$$

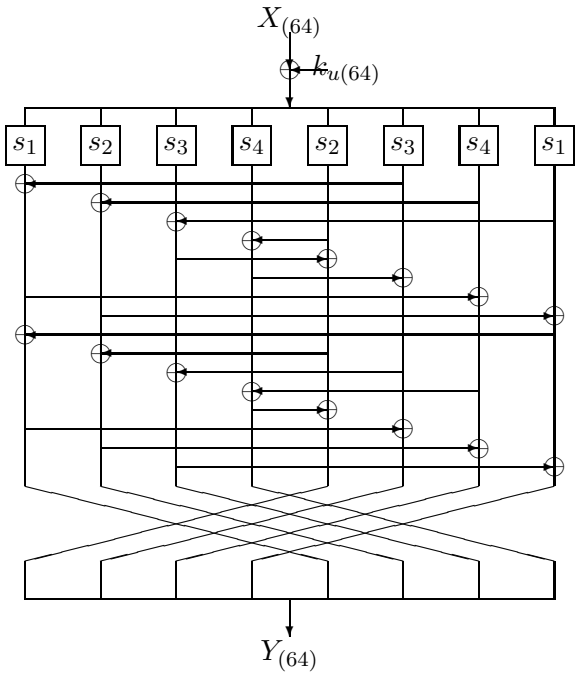


図 2: F 関数

$$
\begin{aligned}
s_3 &: y_{(8)} = s_1(x_{(8)}) >>> 1 \\
s_4 &: y_{(8)} = s_1(x_{(8)} <<< 1)
\end{aligned}
$$

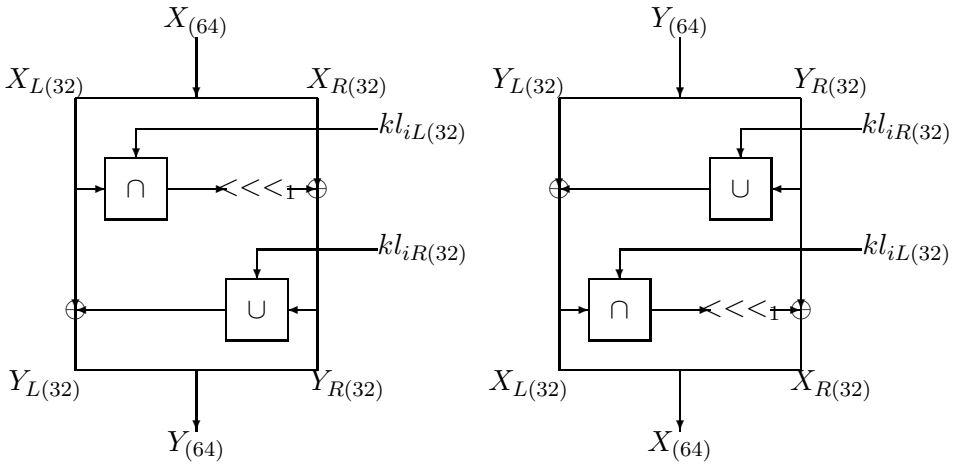
3.4 P 関数

P 関数は F 関数において、S 関数通過後適用される線形拡散層であり、バイト単位の排他的論理和で構成される。入力 8 バイト $(x_1, x_2, \cdots, x_8)$ に対し出力 8 バイト $(y_1, y_2, \cdots, y_8)$ は次式である。

$$
\begin{aligned}
y_1 &= x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7 \oplus x_8 \\
y_2 &= x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7 \oplus x_8 \\
y_3 &= x_1 \oplus x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_8 \\
y_4 &= x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \\
y_5 &= x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_8 \\
y_6 &= x_2 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_8 \\
y_7 &= x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_8 \\
y_8 &= x_1 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7
\end{aligned}
$$

3.5 自己評価書による高階差分解読法耐性

高階差分攻撃では、暗号化変換の中間変数のブール多項式が平文に関し d 次であれば、d+1 階差分が 0 となる性質を利用している [JK97]。Camellia では、S ボックスを作成する際に、$GF(2^8)$



FL 関数　　FL^{-1} 関数

図 3: FL 関数

上の逆数関数を利用しており、その次数は 7 である。実際の S ボックスの次数も 7 次であることを確認しており、データが多くの S 関数を通過するに従い、指数関数的に次数の上昇が期待できる。例えば、3 個の置換表を通過した後、次数は $7^3 > 128$ になり、仕様どおりの Camellia に対して高階差分攻撃は成功しないものと考える。

堀と金子は 1 階差分を使うことによって IT 関数と FT 関数を除いた 5 段 E2 が攻撃可能であることを示した [HK98]。しかし、その攻撃は高階差分攻撃というよりも従来の差分攻撃と同様であると考える。それはその技法の主要な点は、ある中間段で非自明な定数となるような平文と暗号文の組を見つけることにある。したがって、堀らの攻撃に対する Camellia の耐性は従来の差分攻撃に対する耐性と同等であると考える。

4　S 関数の評価

4.1　代数次数及び項数

$s_1 \sim s_4$ の 4 種類すべての s 関数の各出力ビットに対し、ブール多項式展開したところ提案者の自己評価書通り 7 次である。s_1 の次数及び項数を表 1 に示す。最大次数項の欄は、例えば、x_6 の入っていない 7 次項 $x_1x_2x_3x_4x_5x_6x_8$ を $x^7_{[6]}$ と表現してある。他の s 関数は、この入出力ビットの巡回置換で与えられるので、その次数、項数は、同一である。各出力ビット 122 項から 133 項と、平均的項数が偏り無く出ている。

4.2　S 関数の補間多項式表現

Camellia で用いられている S 関数 $s_1 \sim s_4$ を $GF(2^8)$ 上で多項式表現した時の次数、項数を 8 次の原始多項式すべてに対し調査した。次数は全て 254 次であり、項数は原始多項式に依存する。. 結果を表 2 に示す。原始多項式は、各次数の係数を高次項から並べ、それを 16 進表現してある。

即ち、原始多項式 $0x11d$ は $x^8+x^4+x^3+x^2+1$ を表す。補間多項式の項数は、最小 251、最大 255 であり、8 ビット入力の全単射関数としての最大値 255 に近い項数である。原始多項式の選択により特別に項数が少なくなる事は無く、その選択で、補間攻撃の効率を上げる事は期待薄である。

5 F 関数の評価

5.1 F 関数の次数

Camellia の F 関数は図 2 で示される。F 関数は S 関数及び P 関数の 1 段構造からなる。S 関数は 4 種類の s-box がそれぞれ 2 個ずつの計 8 個の s-box が並列に並び、その出力が P 関数を通過する構造である。P 関数は各要素の排他的論理和のみで表現されており、P 関数の次数は 1 次である。F 関数の次数は S 関数の次数 7 と P 関数の次数 1 の積として、7 次である。

F 関数入力 8 バイトのうち 1 バイトのみを変数に取った場合、出力 8 バイトのうち最低 5 バイトが 7 次となり、残り 3 バイトが 0 次となる。同じく入力 2 バイトを（鍵の値のみ異なる）同一変数として取るならば、F 関数出力の左半分 4 バイトは 6 次、右半分は 7 次となる組み合わせが存在する。これらは、P 関数に起因し、後者は、P 関数の構造による S 関数の最高次数項のキャンセルに起因する。

5.2 $FL(FL^{-1})$ 関数の次数

$FL(FL^{-1})$ 関数は図 3 で示され、論理積、論理和、排他的論理和及び 1bit ローテーションからなる。$FL(FL^{-1})$ 関数の次数は 1 次である。

6 高階差分攻撃及び補間攻撃の評価

6.1 形式的代数次数評価

Camellia の F 関数の次数は 7 次であることより、F 関数を 2 回通過後は、$7^2=49$ 次となる。従って、平文右半分を変数、左半分を固定値とするならば、3 段目 F 関数出力において 50 階差分が 0 になる。4 段目出力左半分が 0 であり、1 段消去型攻撃を考えるならば、5 段まで攻撃が可能である。FL 関数は 1 次で、左右半分のブロック内でのデータの攪拌を行っているだけであるので、FL 関数があってもこの結果は変わらない。しかし、5 段目以降の出力では、次数が $7^3>64$ であり、その出力の高階差分を使った攻撃は不可能である。

6.2 8 階差分を用いた攻撃

平文入力 16 バイト $(x_1,x_2,\cdots,x_{16})$ 中 1 バイトに着目し、その入力 1 バイトを変数として、1 ～ 8 階差分を調査した。ここでは、FL 関数無しとして評価している。その範囲で効果的であったのは、8 階差分を用いたもの (後述の SQUARE 型攻撃) であり、平文右半分 1 バイト $x_j,(8\leq j\leq 16)$ を変数とすると 4 段目出力左半分 64 ビットの 8 階差分値が 0 となる。例として x_{16} を変数として、1 ～ 4 段までの各段出力の高階差分値を調べた結果を表 3 に示す。x で表示しているところは変数となるバイトであり 0 は、常に 0 となるバイトである。この 8 階差分を用いて、1 段消去型攻撃

出力 bit	最大次数	総項数	最大次数項
y_1	7	126	$x^7_{[3]}$,$x^7_{[5]}$,$x^7_{[7]}$
y_2	7	129	$x^7_{[1]}$,$x^7_{[3]}$,$x^7_{[5]}$,$x^7_{[6]}$,$x^7_{[7]}$,$x^7_{[8]}$
y_3	7	133	$x^7_{[2]}$,$x^7_{[4]}$,$x^7_{[6]}$
y_4	7	129	$x^7_{[2]}$,$x^7_{[7]}$,$x^7_{[8]}$
y_5	7	122	$x^7_{[3]}$
y_6	7	131	$x^7_{[2]}$,$x^7_{[3]}$,$x^7_{[4]}$,$x^7_{[5]}$,$x^7_{[8]}$
y_7	7	125	$x^7_{[4]}$
y_8	7	127	$x^7_{[1]}$,$x^7_{[3]}$,$x^7_{[4]}$,$x^7_{[6]}$,$x^7_{[7]}$,$x^7_{[8]}$

表 1: S-box(s_1) の代数次数と項数

原始多項式	項数			
	s_1	s_2	s_3	s_4
0x11d	254	254	253	255
0x169	254	251	253	255
0x1e7	250	255	254	255
0x12b	254	254	254	255
0x165	255	254	255	255
0x163	255	255	253	253
0x15f	255	253	253	253
0x1c3	254	255	253	255
0x171	255	255	253	255
0x12d	253	254	253	253
0x1cf	254	254	255	255
0x1a9	253	251	254	253
0x14d	254	255	253	254
0x18d	254	255	255	255
0x1f5	253	255	253	254
0x187	252	255	255	255

表 2: 原始多項式と補間多項式項数

階数	1Round	2Round	3Round	4Round
1 階	000000x000000000	xxx0xxx000000000	xxxxxxxxxxx0xxx0	xxxxxxxxxxxxxxx
2 階	0000000000000000	xxx0xxx000000000	xxxxxxxxxxx0xxx0	xxxxxxxxxxxxxxx
3 階	0000000000000000	xxx0xxx000000000	xxxxxxxxxxx0xxx0	xxxxxxxxxxxxxxx
4 階	0000000000000000	xxx0xxx000000000	xxxxxxxxxxx0xxx0	xxxxxxxxxxxxxxx
5 階	0000000000000000	xxx0xxx000000000	xxxxxxxxxxx0xxx0	xxxxxxxxxxxxxxx
6 階	0000000000000000	xxx0xxx000000000	xxxxxxxxxxx0xxx0	xxxxxxxxxxxxxxx
7 階	0000000000000000	xxx0xxx000000000	xxxxxxxxxxx0xxx0	xxxxxxxxxxxxxxx
8 階	0000000000000000	0000000000000000	xxxxxxxx00000000	xxxxxxxx00000000

表 3: 1 ∼ 4Round　1 ∼ 8 階差分値（変数 x_{16}）

を行うならば、5 段まで攻撃可能である。なお、5.1 節 F 関数の次数を意識し、これ以外に、バイト単位で 4 つまで、同一変数に取る組み合わせで 8 階差分まで調査したが、結果として 1 バイトを変数に取るものが最も効果的であった。

6.3 SQARE 型攻撃

Cammellia の F 関数において、全単射性を持っている部分は、$s_1 \sim s_4$ の 4 種類の S 関数及び F 関数そのものである。従って、S 関数単位の 8 階差分及び F 関数単位の 64 階差分が、この攻撃の対象となる。

暗号系が Feistel 構造であり、F 関数の全単射を使って、F 関数単位の高階差分ならば、4 段目出力左半分 64 ビットの高階差分が 0 となる。1 段消去型攻撃であれば、64 階差分で 5 段攻撃が可能である。同じく、S 関数単位でも 8 階差分で、5 段攻撃が可能なことは、前節で述べた。二つの攻撃の違いは、前者は FL 関数付きでも、攻撃可能であり、後者は、FL 関数無しの場合に適用できる手法である。

6.4 補間解読法 (Interpolation Cryptanalysis)

補間攻撃を拡張した概念として線形和攻撃がある。付録 J で探索した範囲において、Camellia は線形和攻撃に対しても安全であり、5 段で線形和攻撃の攻撃方程式が意味を持たなくなる。

7 高階差分攻撃耐性の評価

以上の探索結果より、FL 関数無しの場合、S 関数単位に変数を取るのが効果的であり、鍵によらず 4 段目左半分 64 ビットの 8 階差分が 0 となる。FL 関数の付いた通常の Camellia では、6.1 節の形式的代数次数評価に基づく 50 階差分が適用でき、同じ場所の 50 階差分が 0 となる。この 2 つのケースについて、攻撃に必要な計算量を以下で見積もる。

7.1 FL 関数無しの場合

Cammellia の F 関数では、F 関数 1 段当たり 64 ビットの拡大鍵が用いられる。1 段消去型攻撃では、拡大鍵の総当たりでは無く、本文 3.2 節の線形化攻撃が効果的である。S 関数の次数が 7

であること及び P 関数は線形であることより、未知項の数は高々$2^8 * 8 = 2^{11}$ である。攻撃方程式は、64 ビット幅であるから、必要な高階差分組数は、$M_2 = \lfloor \frac{2^{11}}{64} \rfloor = 2^5$、即ち、必要平文組数 $2^8 * 2^5 = 2^{13}$ である。各項は、どれか 1 つの S 関数の鍵にのみ関わっており、複数の S 関数にまたがる鍵の項は存在しない事、また F 関数には 8 個の S 関数があることから、計算量は本文 (3.9) 式より

$$T_{5\,段} = \frac{2^8 * 2^5 * 2^{11}}{8} = 2^{21} \tag{1}$$

の F 関数計算量となる。2 段以上の消去型攻撃では、最初の 1 段のみ線形化し、残りの段は拡大鍵の総当たりをすれば、段当たり推定すべき拡大鍵ビット数は 64 ビットづつ増加する。この場合の計算量を本文 (3.10) に従い求めれば、表 4 となる。この 8 階差分攻撃で、8 段まで攻撃可能性があるが、9 段以上は安全である。

段数	平文組数	計算量（段関数＝1）
5	2^{13}	2^{21}
6	2^{14}	2^{86}
7	2^{14}	2^{150}
8	2^{14}	2^{214}

表 4: 8 階差分による攻撃 (FL 関数無し)

7.2 FL 関数有りの場合

形式的次数評価に基づく、50 階差分の攻撃を考える。平文右半分 64 ビット中の任意の 50 ビットを変数にすればよい。前節と同様にして、1 段消去型攻撃では、必要な 50 階差分組は、$2^5 = 32$ 組であり、2^{51} 組の平文から 50 階差分組が 51 組得られるので、必要平文組数は 2^{51} である。計算量は $T_{5\,段} = 2^{63}$ となる。2 段消去型で、最初の 1 段のみ線形化し、残りの段は拡大鍵の総当りを行うとして、計算量と平文組数を求めれば表 5 である。この 50 階差分攻撃で、6 段＋ FL まで、攻撃可能である。なお、ここでは、FL 関数に入る拡大鍵 128 ビット分は総当たりするとして評価した。

段数	平文組数	計算量（段関数＝1）
5	2^{51}	2^{63}
6	2^{51}	2^{127}
$6 + FL$	2^{51}	2^{255}

表 5: 32 階差分による攻撃 (FL 付き)

8 結論

Camellia に対する詳細評価として, 形式的代数次数解析を含む高階差分解読法及び補間解読法耐性を検討した。その結果、FL 関数無しの場合、8 階差分を用いて 8 段まで、攻撃が可能であ

り、FL 関数付きの場合、50 階差分を用いて 6 段+FL まで攻撃が可能と推定される。しかし、それ以上の段数を攻撃する有効な高階差分を得ることはできなかった。

これより、暗号技術仕様書で指定されている 18 段 (24 段) に関し、高階差分攻撃の耐性は十分にあるといえる。また、他の解読法については, 自己評価書において、差分攻撃、線形攻撃、丸め差分攻撃、丸め線形攻撃、不能差分利用攻撃、ブーメラン攻撃、等価鍵不存在性、スライド攻撃、関連鍵攻撃、統計量情報による評価、実装攻撃及びブルートフォース攻撃に関連する自己評価を行っており、その記述内容も信頼でき、Camellia は十分安全な暗号と考える。

参考文献

[NTT,Mitsubishi] 日本電信電話, 三菱電機, ”暗号技術応募書／暗号技術仕様書 Camellia ／自己評価書 Camellia”, IPA 提出資料

[JK97] T.Jakobsen and L.R.Kunudsen, ”The Interpolation Attack on Block Cipher.”, IN E.Biham,editor, Fast Software Encryption-4th International Workshop,FSE’97,Volume 1267 of Lecture Notes in Computer Science, pp.28-40,Berlin, Heidelberg, New York, 1997.Springer Verlag.

[HK98] Y.Hori and T.Kaneko, ”A study of E2 by higher order differential attack.”, Technical Report ISEC98-38, The Institute of Electronics, Information and Communication Engineers,1998.(1998)

[A00] K.Aoki, ”Practical Evaluation of Security against Generalized Interpolation Attack.”, IEICE Transactions Fundamentals of Electronics, Communications and Computer Science,Vol. E83-A, No. 1, pp.33-38,2000(2000)