SC2000の安全性に関する詳細評価

1　はじめに

SC2000はブロック長64ビット、鍵長128ビットの共通鍵ブロック暗号であり、データ攪拌部はFeistel型に鍵依存線形変換FL関数を組み合わせた構造をしている。

本評価書では、最初に提案者による自己評価結果をまとめ、取り上げられている各評価項目に関する検討を行なう。

2　データ暗号化部安全性評価

差分解読法あるいは線形解読法への耐性を保証するために、証明可能安全である構造を採用する設計手法や、特性差分確率、特性線形偏差の理論的上限を評価する設計手法が知られている[1]。SC2000では、DES等の安全性評価で用いられた有意な特性差分確率、特性線形偏差を持つ近似式を探索し、有意な確率あるいは偏差を持った近似式が存在しないことをもって、これら攻撃に対する耐性を示している[3]。

一般に、攻撃者が差分解読法や線形解読法の適用に必要となる近似式の導出は、128ビットブロック暗号においては探索の計算量的にかなりの困難を伴う。設計者らは、この問題を解決するために探索対象をtruncatedベクタの差分波及パターンに置き換えて、効率化する方法を提案している[3]。truncatedベクタとは、S-box層に対する入力差分をS-box単位に分割し、入力差分が0であるか否かで0/1を割り当て、それをまとめたものである。よって、そのビット長はS-boxの個数に一致する。truncatedベクタを用いることによって、差分波及パターンの探索が大幅に効率化できる。

2.1　差分解読法

まずR関数段のみによるFeistel構造に対する2段あるいは3段繰返しの差分近似式探索を行い、R関数の活性S-box数(S5とS6)が少なくなるパターンを絞る。次に、B関数を加えたときの影響を調べ、15段の特性差分近似確率が最小になるものを求めた。ここで「近似」は、S-boxの差分確率を最大差分確率として評価することを意味する。

その結果、15段の特性差分近似確率は、3段繰り返し型を基本にする場合は2^{-134}以下、2段繰り返し型を基本にする場合は2^{-150}以下になることが分かった。つまり、両者とも2^{-128}より小さく、差分解読法に利用できる特性差分近似式が無いことを意味する。

この評価はS-boxの差分値が最大差分をとるものとして見積もられていることに注意が必要である。実際には、S-boxの差分値がすべて最大差分をとるわけではなく、最大でない差分値を含むため、一般に特性差分近似確率はこれより小さな値となる。

4段以上の繰返しが存在するか否かは検討課題としてあげられるが、差分解読法に強いS-boxを使用しているため、探索に際しては膨大は組合せの探索が必要となるため、探索の実施は困難である。

また3段繰返しの結果から推測して、たとえ4段以上の繰返しが存在したとしても現実的計算量で差分解読法は適用困難と考えられる。

2.2　線形解読法

線形解読法に対しても、差分解読法と同様にtruncatedベクタを利用した解読が可能である。ただし、ここでは差分値ではなくマスク値に対するtruncatedベクタを利用する。混乱を避けるため、ここではマスク値に対するtruncatedベクタをtruncatedマスク値と呼ぶことにする。

まずR関数段のみによるFeistel構造に対する2段あるいは3段繰返しの線形近似式探索を行い、R関数

の活性 S-box 数が少なくなるパターンを絞る。次に、B 関数を加えたときの影響を調べ、15 段の特性線形近似確率が最小になるものを求めた。ここで「近似」は、S-box の線形確率を最大線形確率として評価することを意味する。

その結果、15 段の特性線形近似確率は、3 段繰り返し型を基本にする場合は 2^{-142} 以下、2 段繰り返し型を基本にする場合は 2^{-150} 以下になることが分かった。つまり、両者とも 2^{-128} より小さく、線形解読法に利用できる特性線形近似式が無いことを意味する。

この評価は S-box の線形値が最大線形をとるものとして見積もられていることに注意が必要である。実際には、S-box の線形値がすべて最大線形をとるわけではなく、最大でない線形値を含むため、一般に特性線形近似確率はこれより小さな値となる。

4 段以上の繰返しが存在するか否かは検討課題としてあげられるが、線形解読法に強い S-box を使用しているため、探索に際しては膨大は組合せの探索が必要となるため、探索の実施は困難である。

また 3 段繰返しの結果から推測して、たとえ 4 段以上の繰返しが存在したとしても現実的計算量で線形解読法は適用困難と考えられる。

2.3 高階差分解読法

代数次数が小さい関数により構成される暗号においては高階差分解読法は大きな脅威となる [1]。SC2000 では、B 関数および R 関数 (F 関数) が 128 ビット鍵で 19 段、192 または 256 ビット鍵で 22 段使用されている。これら関数は少なくとも 2 次の係数を持つため、7 段以上で高階差分解読による必要平文数は 2^{128} 以上となる。

さらに、設計者は線形近似を組み合わせたモデルについても考察し、2,3 段のセキュリティマージンが確保されているとの評価を下している。セキュリティマージンがやや少ない印象を受けるが、さらに若干の攻撃法の進歩があったとしても 2^{128} に近い平文を要するため現実的な脅威となるとは考えにくい。

2.4 truncated 差分解読法

通常の差分解読法がビット単位で差分を分類したのに対し、truncated 差分解読法では truncated ベクタ自体に注目する [1]。つまり、truncated 差分解読法での特性差分確率は通常の差分解読法の入出力および中間状態に関し、truncated ベクタが同じ範囲で足し上げたものである。

通常の差分解読法に対するセキュリティ・マージンがそれほど大きくないことから、truncated 差分解読法に対する詳細な評価を行なう必要がある。

2.5 χ^2 解読法・分割解読法

χ^2 解読法や分割解読法は、平文と暗号文の部分情報間に統計的な相関性が存在するとき有効である [1]。SC2000 の構成とその構成要素を検討したが、部分情報間に大きな統計的相関性を起こす構造は見当たらなかった。しかし、適用可能性は否定しきれず、計算機実験等によってさらに調べることが望ましい。

2.6 不能差分解読法

通常の差分解読法では、最も大きな差分特性確率を利用して拡大鍵の推定を行なった。不能差分解読法では逆に確率 0 の差分経路、つまり、起こり得ない差分波及パターンを利用して、拡大鍵の絞り込みを行なう [1]。ここで注意すべきは特性確率の意味で確率 0 となるパターンでなく、複数経路について足し合わせたものに対し、確率 0 となる必要があることである。一般にラウンド関数が全単射の Feistel 型暗号では、必ず 5 段まで不能差分解読法で解けることが知られている。よって、B 関数抜きの SC2000 では 5 段まで解読可能であるが、6 段以上の解読に関しては不明である。B 関数は不可能差分パターンを減らす効果があると

考えられるので、SC2000 に対して不能差分解読法は有効でないと考えられる。

2.7 ブーメラン解読法

ブーメラン解読法は、適応的選択平文攻撃を使った差分解読法の一種で、平文側と暗号文側の両方から真中の段に至る差分確率が大きいとき有効となる [1]。SC2000 では 7 段で特性差分近似確率が 2^{-68} と 2^{-64} を下回るので、ブーメラン解読法は有効でないと考えられる。

2.8 法 n 解読法

法 n は、中間出力の n に対する剰余の分布に生じる偏りを利用した攻撃法であり、算術演算を基本とするアルゴリズムで有効である [1]。SC2000 のデータ攪拌部では算術演算が利用されておらず、最大平均差分・線形確率も十分小さいので、法 n 解読法は有効でないと考えられる。

2.9 非全射解読法

ラウンド関数が全射でない場合、出現しないパターンの存在を手がかりに拡大鍵の絞り込みを行なうのが非全射解読法である [1]。SC2000 の F 関数は全射であるので、この解読法は適用できない。

2.10 Luby-Racoff 流ランダム性

非線形関数がランダムだと仮定したとき、アルゴリズム全体が擬似ランダムになることを理論的に示せるとき、Luby-Racoff 流ランダム性があると言い、アルゴリズムの構造に関する安全性の指標となる [1]。ここで、擬似ランダムとは、非線形関数のサイズが無限大になる極限での漸近的性質である。擬似ランダム性は、安全であるための十分条件のようなもので、暗号アルゴリズムの構造はこの性質を満たすもので構成すべきである。

SC2000 の Luby-Racoff 流ランダム性についての研究は行なわれておらず、今後の検討が必要である。

3 鍵スケジュール部安全性評価

3.1 全数探索

全数探索は共通鍵暗号に適用されるもっとも非効率ではあるが確実な解読方法である。SC2000 は 128,196,256 ビットの鍵長であるため、最低でも 128 ビットの鍵に対する全数探索が必要である。既存の技術では 128 ビットで表現される 2^{128} 種類の鍵であっても全ての組合せを計算することは物理的に不可能に近いと思われる。

SC2000 の仕様を検討したところ、全ての鍵ビットは有意に利用されており、128,196,256 ビットのいずれかの安全性を持つ。現在のところ 128 ビットの全数探索に成功した事例は報告されていないため、いずれの鍵長であっても十分な安全性を持つと考えられる。

3.2 弱鍵

弱鍵は解読可能な弱い鍵という意味ではなく、なんらかの好ましくない暗号化処理を行う鍵を指す。SC2000 の自己評価書には、中間鍵の衝突の有無と、全ての中間鍵が一致する可能性の有無について検討が行われており、評価も妥当であった。

さらに、中間鍵に値の衝突が起こらない状況で、拡大鍵が各段で一致する状況を考察したがそのような拡大鍵は発見できなかった。

3.3 拡大鍵数

経験的あるいはヒューリスティックな評価であるが、利用する拡大鍵の量について検証する。SC2000 は 128,196,256 ビットの鍵から 32 ビットの中間鍵 12 個を生成し、この中間鍵を用いて 56 あるいは 64 個の 32 ビット拡大鍵を計算する。中間鍵は各 3 個の 4 グループに分類され、拡大鍵の演算では各グループより 1 個ずつ選択され利用される。したがって演算の組合せは少なくとも $3^4 = 81$ 通りが可能であり、SC2000 の拡大鍵計算では計算パターンの重複は見られず、鍵から中間鍵および拡大鍵の生成が有効に行われている。

3.4 統計的評価

SC2000 の鍵スケジュールはデータ暗号化部で用いる暗号関数により構成される。鍵スケジュールの出力に統計的な偏りがあるならば、暗号関数を共有しているデータ暗号化部の評価において差分解読法あるいは線形解読法に対して何らかの脆弱性を有する。また、ξ^2 検定等においても問題となる検定値は見られなかった。

4 実装に関する攻撃法の検討

SC2000 自己評価書記載の内容を元に，実装に関する攻撃の詳細な検討を行う.

4.1 タイミング攻撃 (timing attacks)

タイミング解析 (タイミング攻撃) は、データ暗号化 (復号) の中間状態などが暗号鍵に依存して処理時間が異なる場合に、その相関性を利用して処理時間から直接拡大鍵の推定あるいは中間状態の推測を行い、鍵を推定する方法である [1]。

タイミング解析は実装に依存した攻撃である為、ハードウェア、ソフトウェアいずれについても適用可能である。以下に SC2000 の実装に際して対策の必要性の有無について検討する。

4.1.1 ハードウェア

ハードウェアにおいても、暗号利用の処理時間を詳細に観測することによりタイミング攻撃の適用は可能である。SC2000 は小規模のテーブル参照あるいは論理回路で実装されるため、通常の実装においては鍵等に依存した処理時間の差異は考え難い。そのため、対策は必要ないかあるいは極めて容易であると考える。

4.1.2 ソフトウェア (PC)

PC で利用する CPU および OS は耐タンパーでないため、タイミング攻撃以外の実装に関係した攻撃方法も可能であることを考慮する必要がある。例えば実行されるコードや CPU 内部のレジスタの内容は比較的容易に観測可能である。このような直接的で強力な解析よりもタイミング解析のコストが大きければ、タイミング解析はその優位性を失うこととなる。

PC ソフトウェアは、OS 上で動作するため、OS に起因する処理時間の差異の影響を受ける。例えば、CPU のキャッシュヒットミスや、OS のメモリ管理により処理時間は変動する。この処理時間の変動よりも、鍵に依存した処理時間の変動が大きい場合にもに、タイミング攻撃は有意な情報量を取り出せるものと考える。

除算や乗算などの例外処理や処理の条件分岐の存在するとタイミング攻撃を適用できる処理時間の大きな変動が考えられるが、SC2000 は小規模のテーブル参照あるいは論理回路で実装されるため、通常の実装においては鍵等に依存して処理時間の大きな差異が発生するとは考え難い。仕様書等に述べるように、鍵スケジュールの乗算はシフト演算と加算での実装も可能であるが、CPU の命令セットにある乗算命令であっても処理時間にデータ依存の影響を与えない。したがって、処理分岐を含まず、処理時間がキャッシュミス程度の変動であり、タイミング解析に対して安全であると考えられる SC2000 コードを作成することは可能である。

4.1.3 ソフトウェア (IC カード)

IC カードは耐タンパーデバイスであるため、IC カード内部のコードやレジスタの状態を観測することは極めて困難である。すなわち IC カードの内部を直接的に観測する手法の解析は適用に困難を伴う。

一方、IC カードでは搭載されているカード OS は PC のもの程高性能ではない。そのため、詳細な評価を行い、テスト環境を整えれば特定の処理のみの時間を外界から正確に測定することは可能である。このため、除算や乗算などの例外処理や処理の条件分岐による若干の処理時間の差異であってもタイミング攻撃の適用は可能である。

したがって、SC2000 は小規模のテーブル参照あるいは論理回路で実装される。鍵スケジュールに関しては乗算演算を含むが、自己評価書に述べられているようにキャリーつきのシフト演算と加算で容易に実装でき、通常の実装においては鍵等に依存して処理時間の差異が発生するとは考え難い。処理分岐を含まず、処理時間が同じであり、タイミング攻撃に対して安全である SC2000 コードを作成することは可能である。

4.2 電力解析

IC カード実装における電力解析を検討する。電力解析は、大きく分けて SPA(Simple Power Analysis) と DPA(Differential Power Analysis) に分類される。SPA は 1 回の暗号化処理における消費電力波形を利用し、鍵の導出を試みる。あるいは、SPA による、より詳細な評価を行うならば、複数の暗号化処理の消費電力波形を観測し、同一処理箇所における消費電力波形の差異に着目し、鍵の導出を試みる。また、DPA は複数回の暗号化処理における消費電力波形を統計処理し、鍵の導出を試みる。

IC カード実装においては、SPA および DPA に対して安全な暗号処理用コードあるいは電力解析に対する対策を施したハードウェアによる実装が必要である。SC2000 は処理分岐を伴わずに実装可能であるため、タイミング解析や単純な SPA に対しては耐性が高い。

本評価では初歩的な SPA よりも一歩進んだ、複数データ間の消費電力波形の比較による SPA と DPA の適用可能性を検討する。

4.2.1 I 関数

I 関数はデータと拡大鍵との XOR 演算のみが含まれる。XOR 演算は全単射な写像であるため、一般的に SPA および DPA による解析で鍵に関する有意な情報量を取り出すのは困難であり、電力解析の適用は困難である。

4.2.2 F 関数

F 関数は S 関数 (S_func)、M 関数 (M_func)、および L 関数 (L_func) から構成される。IC カード実装においては効率の問題から S 関数と M 関数は結合関数として実装されると見られる。この S 関数と M 関数の組み合わせは、DES における S-box と P 転置の結合と同様の手段であり、すでにさまざまな対策方法が試みられている。文献 [2] の DPA 対策等を講じることにより、処理効率は低下するが、DPA についても消費

電力波形と実際の処理データとの間の相関を除去し、かなりのレベルで DPA 対策を行えるものと考える。

また、L 関数についても定数との AND 演算および XOR 演算のみで実装されるため、文献 [2] の DPA 対策に悪影響を与えないと考える。

以上により、消費電力解析に対する対策を行うのは可能であり、対策の難易度も他の暗号方式と比較して不利ではない。

4.2.3 B / B^{-1} 関数

B 関数およびその逆関数 B^{-1} 関数は 4 ビット入出力の S-box と転置 T/T^{-1} から構成される。

B/B^{-1} はビットスライス実装に適した設計であり、複数の S-box と転置 T/T^{-1} は複数の論理演算の組合せで記述できる特徴がある。PC 等に使用される現在の CPU は 32 ビットあるいは 64 ビットの演算幅を持ちこのビットスライス実装により実装の効率化がはかれる。効率的な論理演算の組合せの決定は容易ではないが、文献 [3] には必要とされる論理演算の組合せが記載されており実装可能である。

IC カードでの実装においては、ビット転置と 4 ビット入出力 S-box の参照あるいは 2 つの S-box を連結した 8 ビット入出力のテーブル参照により実装が可能である。また、各ビットをバラバラに再配置するビット転置は IC カードで使用される 8 ビット CPU では演算コストが大きくなるため、複数バイトに対する論理演算の繰返しによる実装も考慮の対象となる。

したがって、ビットスライス実装を利用した論理演算実装の場合とビットスライス実装を用いないビット転置とテーブル参照の組合せの二つの実装方法についてそれぞれ電流解析適用の可否を検討する。

ビットスライス実装は論理積 (AND)、論理和 (OR)、排他的論理和 (XOR) および論理否定 (NOT,$\overline{x}$) から構成される。B/B^{-1} 関数は任意の鍵入力について全単射な写像となる。しかし、構成部品 (最小演算単位) レベルでは、拡大鍵との論理積あるいは拡大鍵との論理和演算において構成部品の入出力が全単射とならない。具体的にはビットスライス実装の演算の一部、論理積 (AND) や論理和演算 (XOR) は表 1 の真理値表に示すように全単射ではない。

表 1: 論理演算真理値表

論理積 ($y = k \cdot x$)

k	x	y
0	0	0
0	1	0
1	0	0
1	1	1

論理和 ($y = k \mid x$)

k	x	y
0	0	0
0	1	1
1	0	1
1	1	1

表 1 より、論理積 (AND) の出力においては 3/4 の確率で 0 が、1/4 の確率で 1 が出力される。論理和 (OR) の出力においても同様に 1/4 の確率で 0 が、3/4 の確率で 1 が出力される。したがって、この出現確率の偏りに起因する情報量を SPA や DPA といった手法で取り出すことにより、B/B−1 関数の演算に用いる変数を推定することは可能である。ただし、このビットスライス処理部分から拡大鍵の情報を取り出すためには、前段の F 関数も含めた統計処理が必要である。我々の解析では、鍵の導出までは至らなかった。

一方、ビット転置と S-box のテーブル参照により実装される場合、すべての処理が全単射な写像で定義されるため、情報漏洩の要因となる確率分布の偏りは発生しない。また、拡大鍵も含まないため、この実装方法では B/B^{-1} 関数に対する電力解析の脅威は低い。

なお、さらなる評価の結果、電力解析に対する対策が必要であると判断されても文献 [2] に動的テーブル作成を用いた方法や、ビットスライス実装に適用できる論理演算のランダムマスク手法が提案されており、速度低下を代償にこれらの手法によりかなりの安全性レベルを提供することは可能であると考える。

4.3 鍵スケジュール

SC2000 の鍵スケジュールはデータ暗号化部でも利用されている S 関数と M 関数を使用する。鍵スケジュールでは乗算処理が含まれる。IC カードにおける乗算処理はデータと消費電力の相関が極めて強く電力解析の対象となる例が見られるが、SC2000 では乗算定数が 1,2,3 と小さいため、シフト演算と加算により乗算処理を回避することができる。

鍵に依存して消費電力が異なる処理は含まれないため、電力解析による鍵スケジュール部の攻撃は困難であると考える。

4.4 実装に関する攻撃法のまとめ

SC2000 の実装に関する攻撃法、タイミング攻撃 (Timing Attack), SPA(Simple Power Analysis), DPA(Differential Power Analysis) について検討を行った。これら実装に関する攻撃に対する耐性は自己評価書にはなんら記載されていない。

我々の解析によると SC2000 はタイミング攻撃に対しては安全性が高い実装が可能な設計であり、万一脅威となる場合でも対策は容易である。また、SPA や DPA といった消費電力解析に対しては、対策が困難である関数処理は見つけ出せなかった。

5 ソフトウェア実装

SC2000 のデータ暗号化および復号処理は、大きく分けて 2 個の S 関数・M 関数と 1 個の L 関数からなる F 関数をラウンド関数とする Feistel 構造、順逆のインターリブである T 関数・T^{-1} 関数で 32 並列の 4 ビット S-box を挟んだ構成した SPN 構造 (正確には PSP 型) の B 関数、鍵加算の I 関数、という 3 種類の設計要素で出来ている。

SC2000 の鍵スケジュールは S 関数と M 関数の組合せを用いる。まず 32 ビットの中間鍵の作成に、S 関数と M 関数の組で 3 組の演算と事前計算による定数が用いられる。これは 2 組の S 関数と M 関数、1 個の L 関数から構成される F 関数の 1.5 段分に相当する。SC2000 は 32 ビットの中間鍵を 12 個用いるため、中間鍵作成で F 関数 18 個分相当の演算を要する。データ暗号化部の F 関数と B 関数の演算量が同等と仮定すると、鍵スケジュール部の演算量は 1 ブロックのデータ暗号化相当である。データ暗号化部の B 関数が鍵スケジュール部の中間鍵から拡大鍵を計算する処理に相当すると仮定すると 2 倍である。実装環境に依存するが、1〜2 倍であれば適正な実装であると考えられる。

5.1 PC 実装

設計者らにより文献 [3] において Pentium III あるいは Athlon を搭載した PC での実装が報告されている。データ暗号化部、鍵スケジュール部ともに相当の最適化がはかられている。

5.2 IC カード実装

設計者らにより文献 [3] において Intel 8051 を搭載した IC カードでの実装が報告されている。

暗号化と復号を持つコードが 1751Byte の ROM で実装可能である。文献には実装内容の詳細は示されていないため、実装の正当性について若干の考察を加える。外部 RAM には 12 個の 32 ビット中間鍵 (48 バイト) と 56 個の 32 ビット拡大鍵 (224 バイト) が格納される。最後の拡大鍵の計算領域 (4 バイト) は中間鍵領域に配置できるため、$48 + 224 - 4 = 268$ バイトで拡大鍵の計算が可能である。この考察は表の内容と一致する。

暗号化速度は他の 128 ビットブロック暗号と比較してさほど高速でない。また、鍵スケジュールにデー

暗号化	8.133 msec
復号	8.609 msec
鍵スケジュール	21.666 msec
コードサイズ	1597Byte
静的データ	1154Byte
内部 RAM	24Byte
外部 RAM	270Byte

タ暗号化部の 2.5 倍の処理がかかっているがさらなる最適化の余地があると思われる。これら状況から、IC カード実装においては、データ暗号化、鍵スケジュールともにさらなる最適化の余地があると思われる。

6 おわりに

SC2000 について、データ攪拌部・鍵スケジュール部の安全性と実装攻撃に対する安全性について検討した。その結果、データ攪拌部・鍵スケジュール部に重大な欠点は見つからず、実装攻撃に対しても小さなコストで対応可能であることが分かった。また、実装性能についても考察したところ、IC カード実装ではさらなる改良の余地があった。

このように、現在のところ SC2000 の安全性上の欠陥は見つかっていないが、2 段 Feistel 型と SPN 型を交互に重ねた構造に対する解析はほとんど行なわれていない。今後、さらなる解析を重ねていくことが必要である。

参考文献

[1] 「共通鍵ブロック暗号の選択／設計／評価に関するドキュメント」, 通信・放送機構, 2000.

[2] Thomas S. Messerges, “Securing the AES Finalists Against Power Analysis Attacks”, FSE2000, 2000.

[3] 下山他, “共通鍵ブロック暗号 SC2000”, 信学技報 ISEC2000-72, 2000.