共通鍵暗号を取り巻く現状と課題

― DESからAESへ ―

宇根正志／太田和夫

要　旨

共通鍵暗号は、暗号化と復号に同一の鍵を用いる暗号であり、金融分野をはじめとして幅広い分野において利用されている。共通鍵暗号の中でもブロック暗号と呼ばれる方式が主要な商用暗号として利用されており、1977年に米国政府標準暗号に認定されたDES（Data Encryption Standard）が事実上の標準として利用されてきた。

しかし、DESは、鍵長が56 bitであることから、近年のコンピューターのコストパフォーマンス向上等によって安全性が徐々に低下している。このため、現在金融分野を中心に、DESの代替暗号としてTriple DESを利用する動きが広がっている。Triple DESは、DESのアルゴリズムを３回繰り返す方式であり、①DESからの移行が比較的容易である、②全数探索法に対する安全性が向上する等の利点を有している。

一方、米国政府は、次世代の標準暗号としてAES（Advanced Encryption Standard）の標準化を進めている。AESは、標準化完了後、一般に利用可能となるまでにはさらに数年はかかるとみられており、Triple DESの次の主要な暗号方式として位置付けられている。

DESからTriple DES、さらにはAESへの移行に象徴されるように、共通鍵暗号を取り巻く環境は急速に変化している。本稿では、共通鍵暗号の機能、構造、主要な解読法のほか、主要な共通鍵ブロック暗号に関するこれまでの安全性評価結果について整理するとともに、DESからTriple DES、そしてAESへの移行の経緯と現状について説明する。

キーワード：AES、DES、Triple DES、共通鍵ブロック暗号、米国政府標準暗号

本稿は、1998年11月4日に日本銀行で開催された「金融分野における情報セキュリティ技術に関するシンポジウム」への提出論文に加筆・修正を施したものである。

宇根正志　日本銀行金融研究所研究第２課（E-mail: masashi.une@boj.or.jp）
太田和夫　日本電信電話（株）情報流通プラットフォーム研究所
（E-mail: ohta@sucaba.isl.ntt.co.jp）

Ⅰ．はじめに

共通鍵暗号は、暗号化と復号に同一の鍵を用いる暗号であり、情報の秘匿・改竄防止技術として、従来から金融分野を中心に幅広く利用されている。これまで様々な共通鍵暗号方式が提案されているが、その中でも、1977年に米国政府標準暗号に認定されたDES（Data Encryption Standard）が事実上の標準として利用されてきた。

しかし、DESは鍵長が56 bitであることから、近年のコンピューターのコストパフォーマンス向上等によって、Brute Force Methodに対する安全性低下が深刻化している。このため、現在金融分野を中心に、DESの代替暗号としてTriple DESを利用する動きが広がっている。Triple DESは、２つもしくは３つの異なる鍵を用いて、DESのアルゴリズムを３回繰り返して暗号化するという方式であり、①DESからの移行が比較的容易である、②２つもしくは３つの異なる鍵を利用することによって、鍵長がそれぞれ112 bitおよび168 bitに拡張する効果を有しており、Brute Force Methodに対する安全性が向上する等の利点を有している。米国政府は、1999年1月にTriple DESを新たな米国政府標準暗号に認定する方針を発表しており、同年4月に正式に認定される見通しである。

一方、米国政府は、次世代の標準暗号として、AES（Advanced Encryption Standard）の標準化を進めている。AESは、鍵長として128、192、256 bit、ブロック長として128 bitが利用可能な共通鍵ブロック暗号とされており、現在15の候補アルゴリズムの分析・評価が実施されている。米国政府は、AESの標準化を完了した後、AESを米国政府標準暗号として20～30年の間利用する方針を発表しており、次世代の代表的な暗号方式として幅広い分野において普及する可能性が高い。現在発表されているAESの標準化スケジュールによれば、早ければ2000年内にも候補アルゴリズムのひとつがAESとして認定される予定となっているものの、標準化完了後、AESがその安全性について高い信頼を得て、AESを利用した暗号製品が広く普及するまでには、さらに数年程度はかかるものとみられている。このため、AESが一般に利用可能になるまでの間は、Triple DESがDESの後継暗号として利用されるとの見方が多い。

DESからTriple DES、さらにはAESへの移行に象徴されるように、共通鍵暗号を取り巻く環境は急速に変化している。今後共通鍵暗号を利用するに際しては、共通鍵暗号に関する研究動向について理解を深めることが重要となっている。

本稿では、まず第Ⅱ章において、共通鍵暗号の機能、種類、構造等について説明する。続いて、第Ⅲ章において、商用暗号の主流となっている共通鍵ブロック暗号の主要な解読法について説明し、第Ⅳ章で、主要な共通鍵ブロック暗号の構造とこれまでの安全性評価結果について説明する。第Ⅴ章では、DESからTriple DESへの移行の背景について説明し、Triple DESの構造や安全性評価結果について説明する。最後に、第Ⅵ章において、次世代の主要な共通鍵ブロック暗号と目されているAESの標準化動向について説明する。

Ⅱ．共通鍵暗号の概要と機能

共通鍵暗号は、暗号化と復号に同一の鍵を利用する暗号方式である。共通鍵暗号は、ネットワーク上でやり取りされるデータや外部記憶媒体に保管されるデータを秘匿するために用いられるほか、無権限者によるデータの改竄を防止・検出するための技術としても利用されている。共通鍵暗号を利用して暗号通信を行う場合には、暗号化に利用した鍵を安全に受信者に配送する必要があるものの、鍵の配送が不要な公開鍵暗号よりも高速で暗号化・復号を行うことができる。このため、一般的に、データの暗号化手段として共通鍵暗号が利用され、共通鍵暗号の鍵を配送する手段として公開鍵暗号が利用されるケースが多い。

1. 共通鍵暗号の機能

共通鍵暗号の主な機能として、(１)守秘、(２)認証、(３)ハッシュ関数が挙げられる。

(1) 守秘機能

インターネット等オープンなネットワーク上でデータをやり取りする場合、データ送信の途中で、送受信者以外の第三者によってデータの内容を覗き見られる可能性がある。また、磁気ディスク等の記憶媒体にデータを記録・保管する場合、無権限者がその媒体から無断でデータを読み出す可能性がある。こうした無権限者によるデータの覗き見を防止する機能が守秘であり、共通鍵暗号はデータの守秘を実現する技術のひとつである。共通鍵暗号では、暗号化に利用された鍵を有する者だけが、その鍵で暗号化されたデータを復号することができる。信頼できる共通鍵暗号アルゴリズムが利用されている限り、鍵を知らない者が暗号化されたデータを解読することは計算量的に困難となる[1]。

(2) 認証機能

データの認証は、オープンなネットワーク上でやり取りされるデータの正当性を確保する機能である。共通鍵暗号によって実現される認証機能としては、主に①アクセスしてきた利用者の正当性を確認する「ユーザー認証」、②通信の途中で第三者によってデータが改竄されていないことを確認する「メッセージ認証」、が挙げられる。

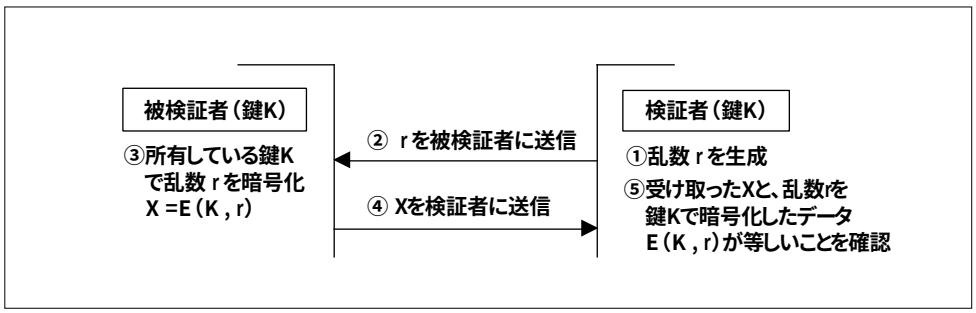
1 計算量的に困難であるとは、その計算を行うことは理論的には可能であるものの、実際にその計算を実行するには計算量が非常に大量となり、膨大な費用と時間を必要とすることから、事実上不可能であることを意味する。どの程度の計算量が「事実上不可能」であるかは、その時々の技術条件等によって左右される。

①ユーザー認証

ユーザー認証は、ネットワーク等を経由してアクセスしてきた通信相手が正当な権限を有しているかどうかを確認する機能である。共通鍵暗号では、通信を行う者同士の間で鍵を共有して安全に管理する必要があるが、いったん鍵が共有されると、その鍵を使って正しい演算を行うことが可能か否かによってユーザー認証を実現することができる。たとえば、以下の手順によってユーザー認証が実現される（図１参照）。

（ステップ１）検証者が乱数rを生成する。
（ステップ２）検証者は被検証者にrを送信する。
（ステップ３）被検証者はrを受信し、検証者との間で既に共有している鍵Kを使ってrを暗号化する。
（ステップ４）被検証者は、rを暗号化したデータE(K, r)を検証者に送信する。
（ステップ５）検証者は、鍵Kでrを暗号化したデータと受信したE(K, r)を照合し、一致した場合には被検証者が正当なユーザーであることを確認する。

図１　共通鍵暗号を利用したユーザ認証の例



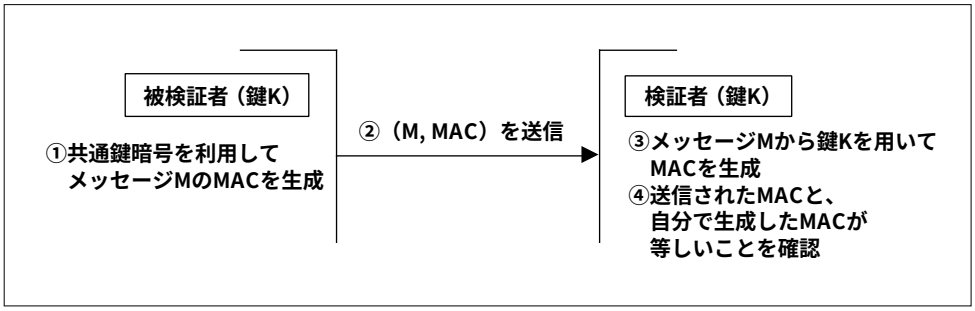
②メッセージ認証

共通鍵暗号を利用したメッセージ認証としては、メッセージの認証子MAC（Message Authentication Code）を利用する方式が一般的である[2]。MACは、メッセージを送信者の鍵を利用して圧縮変換することによって生成されるデータであり、メッセージの真正性と送信者確認を同時に行うことを目的としてメッセージに付されるものである。MACは、メッセージが1 bitでも変更されると、そのメッセージに対応するMACも変化するという性質を有しているほか、MACを生成する際には、MACから元のデータを復元することが計算量的に困難な圧縮変換が利用される。MACを利用したメッセージ認証の方法は、以下の通り（図２参照）。

2　ISO 8731-1（Banking - Approved algorithms for message authentication - Part 1: DEA）では、金融業務で利用されるMACを生成するアルゴリズムとして、CBCモードを利用したDES（図３参照）が規定されている。

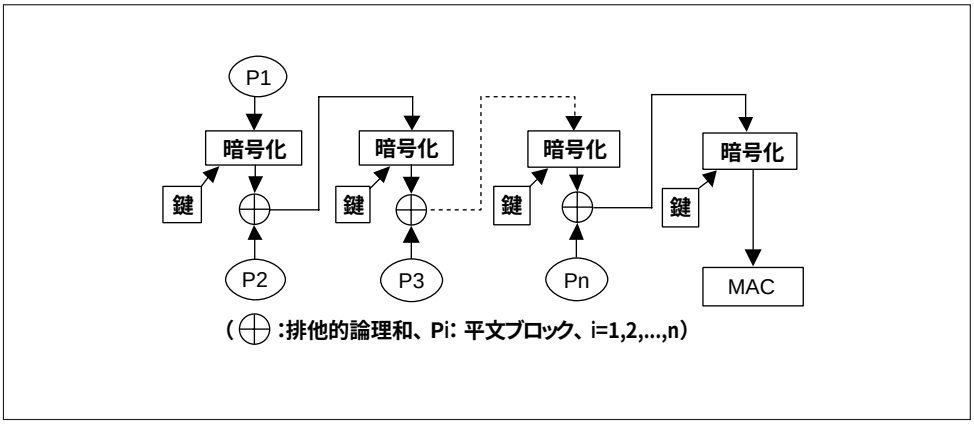
（ステップ１）被検証者は、予め共有している鍵Kを使ってメッセージMに対するMACを作成する。
（ステップ２）被検証者は、生成したMACとともにMを検証者に送付する。
（ステップ３）検証者は、受信したメッセージMから鍵Kを利用してMACを生成し、メッセージとともに受信したMACと一致するかどうかを確認する。一致した場合、メッセージの送信者がその鍵を共有している被検証者であり、かつ送信されたメッセージが改竄されていないことが確認される。万一、通信の途中で第三者によってメッセージやMACが改竄された場合、MACの検証は成功しないため、改竄の検出が可能となる。

図２　MACを利用したメッセージ認証方式



なお、共通鍵暗号を利用した典型的なMACの生成方法は、以下の図３の通り。

図３　共通鍵暗号を利用した典型的なMACの生成方法



（3）ハッシュ関数

ハッシュ関数は、任意のデータを一定のbit長のデータに圧縮変換する関数であり、前述のMAC生成のようにメッセージ認証等に利用される。ハッシュ関数としては、メッセージダイジェストタイプと呼ばれているMD5やSHA-1等が広く利用されているが、共通鍵暗号を利用する方式も提案されている。共通鍵暗号を利用する方式の中でも代表的なものとしては、Matyas-Meyer-Oseas方式が挙げられる[3]。この方式によるハッシュ値生成方法は、以下の通り（図４参照）。

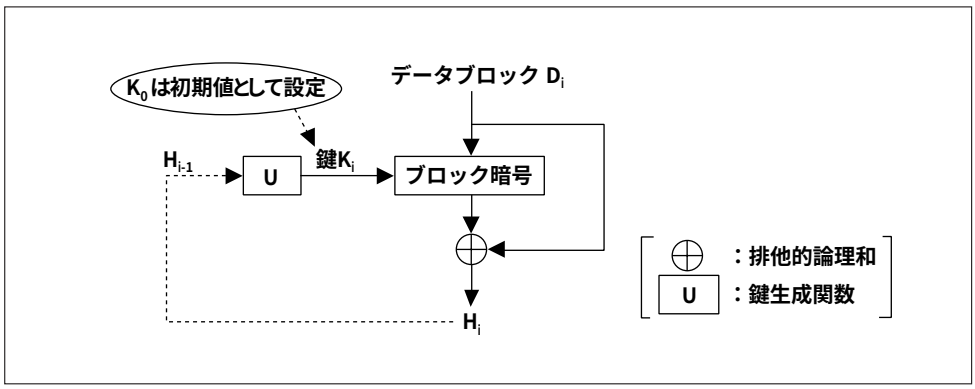
（ステップ１）ハッシュ化するデータを一定長のブロック単位（利用するブロック暗号のブロック長に対応）に分割する。分割後のデータをDi（i = 1,...,n）とする。

（ステップ２）初期値K_0を鍵としてD_1をブロック暗号によって変換し、その変換後のブロックをD_1との排他的論理和によって変換してH_1を生成する。H_1のサイズはブロック暗号のブロック長に等しい。

（ステップ３）H_1を鍵生成関数Uによって変換して鍵K_2を生成し、鍵K_2を用いて次のブロックD_2をブロック暗号によって変換する。変換後のデータは、D_2との排他的論理和によって変換され、H_2が生成される。

（ステップ４）ステップ３の手続きを最後のブロックD_nの変換が終了するまで継続し、その結果生成されるH_nがハッシュ値となる。

図４　Matyas-Meyer-Oseas方式による植生方法

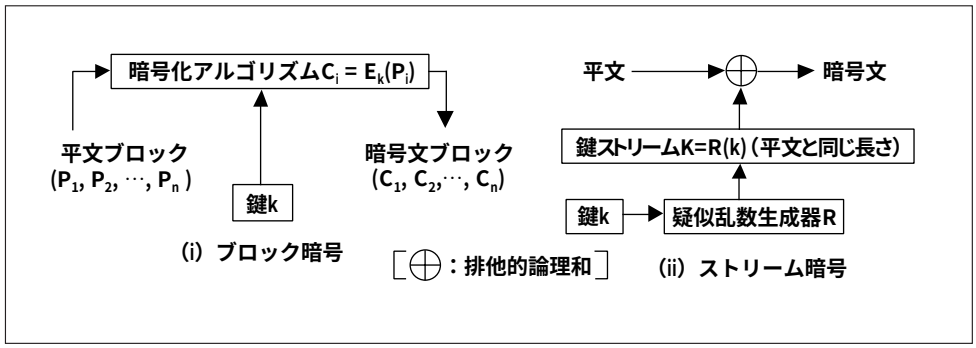


2. 共通鍵暗号の種類

共通鍵暗号は、大別すると、ストリーム暗号とブロック暗号に分類される（図５参照）。

3　Matyas-Meyer-Oseas方式は、ブロック暗号を利用したハッシュ関数の利用方法について規定されているISO/IEC 10118-2（Hash-functions Part 2: Hash-functions using an n-bit block cipher algorithm）に記載されている。本標準には、この方式のほかに、Matyas-Meyer-Oseas方式のハッシュ関数を２つ並列に配置し、ブ

図５　ブロック暗号とストリーム暗号の一般的な暗号化手順



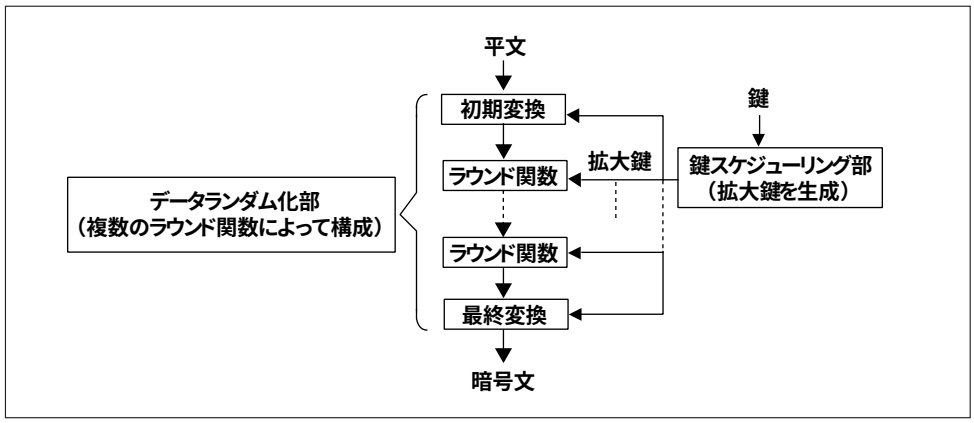
ストリーム暗号は、暗号化する平文と同じbit長の鍵ストリームを疑似乱数生成器等によって生成し、平文と鍵ストリームの排他的論理和を計算することによって暗号文を生成する方式である。ストリーム暗号は、軍事用に利用されているものが多く、具体的なアルゴリズムの仕様が公開されている暗号方式は比較的少ない。

一方、ブロック暗号は、暗号化するデータをある一定長のブロック毎に分割し、各ブロック毎に同一の鍵で暗号化する方式である。DESをはじめとして、多くのブロック暗号のアルゴリズムが公開されており、ブロック暗号が現代の商用暗号の主流となっていることから、以下ではブロック暗号について説明することとする。

3. ブロック暗号の構造

ブロック暗号は、データランダム化部と鍵スケジューリング部によって構成される（図６参照）。

図６　一般的なブロック暗号のアルゴリズムの構造



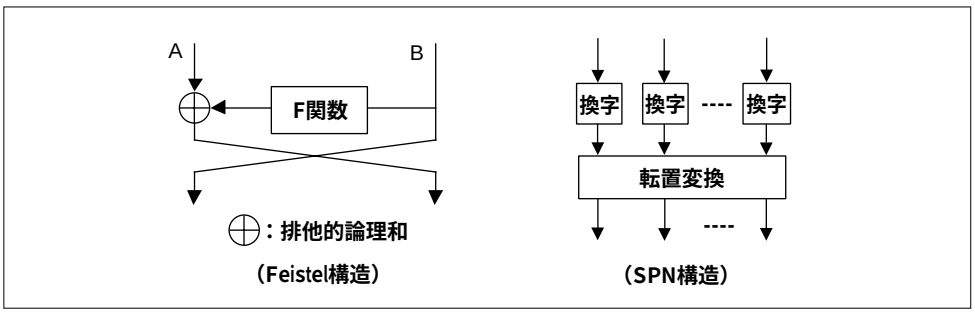
ロック長の２倍のサイズのハッシュ値を生成するMDC-2と呼ばれる方式も記載されている。なお、この標準には、Matyas-Meyer-Oseas方式に利用するブロック暗号の例として、DESが紹介されている。

(1) データランダム化部

データランダム化部は、鍵スケジューリング部において生成された拡大鍵を利用して平文ブロックを暗号化する部分である。多くの暗号方式において採用されているデータランダム化部は、①初期変換、②繰り返し利用される基本変換（ラウンド関数と呼ばれる）、③最終変換によって構成されている。暗号方式によっては、初期変換や最終変換がないものも存在する。

ラウンド関数は構造上いくつかのタイプに分類されるが、よく利用されるものとして、Feistel構造とSPN構造が挙げられる（図７参照）。Feistel構造は、米IBM社のFeistelが開発したLuciferのラウンド関数に初めて採用されたものであり、２つのサブブロックの一方をF関数と呼ばれる非線形変換によって変換した後、その変換されたサブブロックとの排他的論理和によって他方のサブブロックを変換し、２つのサブブロックの位置を入れ替えるという手順で変換が行われる。Feistel構造には、①構造が単純であり、安全性に関する分析が容易である、②暗号化変換と復号変換が同一である（同じ鍵を利用した暗号化変換で暗号文を変換すると平文が得られる）等の特徴点があり、DESをはじめとする多くの暗号方式において採用されている。

図７　Feistel構造とSPN構造



一方、SPN（Substitution-Permutation Network）構造は、換字変換（ある数値を別のある数値に規則的に置き換える変換）と転置変換（データブロックのbitの位置を規則的に入れ替える変換）が連続的に配置された構造であり、DESのF関数等に利用されている[4]。SPN構造には、①暗号化変換と復号変換が同一にはならず、暗号化変換を復号に利用するためには別の変換手段を適宜付加する必要がある、という短所が存在するものの、②データブロックの攪拌が速い（Feistel構造においてはデータブロック全体を非線形変換（F関数）するためにはラウンド関数２段が必要となる一方、SPN構造ではラウンド関数１段で変換できる）等の長所がある。

4　SPN構造は、Shannonが提案した合成関数（product cipher）の安全性に関する考え方に依拠して開発されたといわれている。合成関数は、２つ以上の異なる種類の関数を繰り返し利用することでデータを変換する関数を指すが、Shannonは「単純な変換を単独で暗号化関数に利用してもその暗号の安全性は大して向上しないが、複数の変換を組み合わせて暗号化関数を構成することで、その暗号の安全性を著しく向上させることができる」ことを情報理論的に示した。

なお、これらの構造以外にも、一般形Feistel構造や2-round SPN構造等、様々なタイプのラウンド関数が考案されている（詳細は本稿末の表９を参照）。

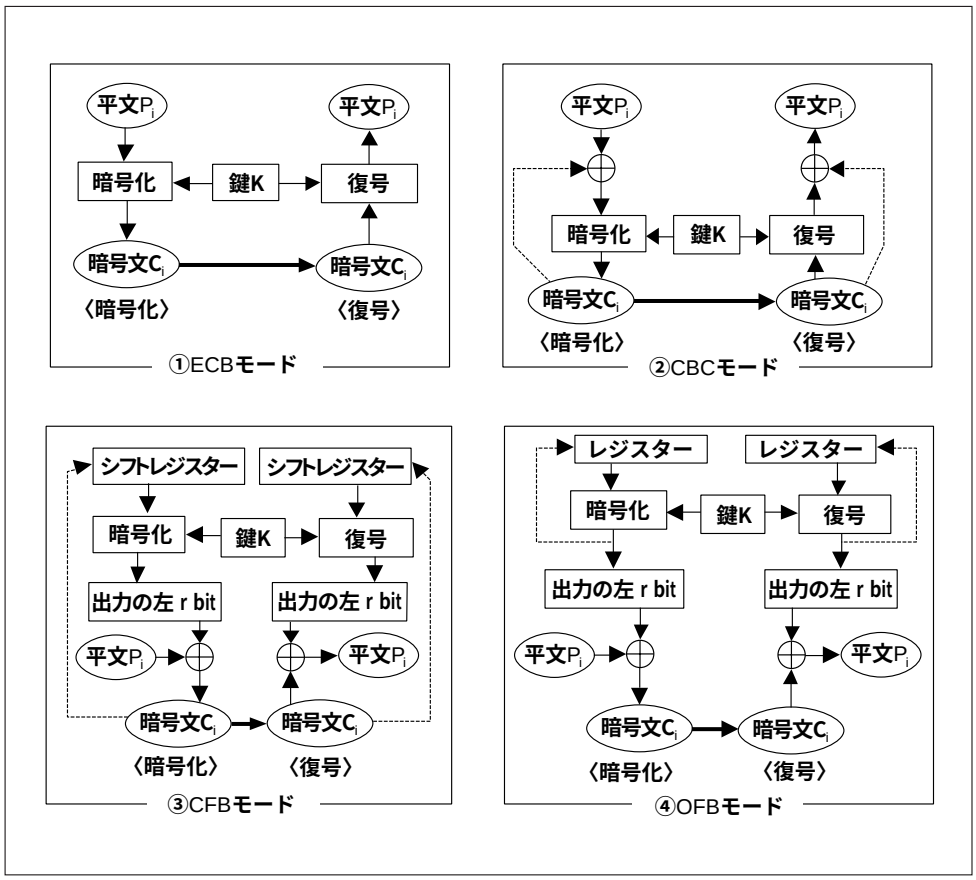
（2）鍵スケジューリング部

鍵スケジューリング部は、鍵からデータランダム化部での変換に必要な拡大鍵を生成する部分である。

4. ブロック暗号の利用モード

ブロック暗号によって暗号化・復号を行う場合、いくつかの種類の利用モードが存在する。主要な利用モードとして、①ECBモード、②CBCモード、③CFBモード、④OFBモードが挙げられる[5]（図８参照）。

図８　主要な4つの利用モード



5 これらの４つの利用モードは、ブロック暗号の利用モードに関する標準規格ISO/IEC 10116（Information technology ― Security techniques ― Modes of operation for n-bit block cipher）において規定されている。

(1) ECBモード

ECB（Electronic Codebook）モードは、各ブロックを他のブロックとは独立に暗号化・復号するモードであり、暗号化する場合には、各平文ブロックをそれぞれ暗号化アルゴリズムによって変換することで暗号文ブロックが生成される。このため、暗号文を通信相手に送信する途中で、ある暗号文ブロック１個にエラーが発生したとしても、復号の際にはその暗号文ブロックに対応した平文ブロック１個にエラーの影響が及ぶだけである。ECBモードでは、同じ鍵を利用して同じ平文ブロックを暗号化すると、同一の暗号文ブロックが生成される。

(2) CBCモード

CBC（Cipher Block Chaining）モードは、各平文ブロックを直前の暗号文ブロックとの排他的論理和によって変換した後、その変換結果を暗号化アルゴリズムによって再度変換することで暗号文ブロックを生成するモードである。最初の平文ブロックをCBCモードで暗号化する場合には、直前の暗号文ブロックとして初期値が与えられる。このため、同じ平文ブロックを同じ鍵で暗号化しても、初期値が異なっている場合には、異なる暗号文ブロックが生成される。復号の際には、まず同一の鍵を用いた復号アルゴリズムによって暗号文ブロックを変換した後、直前の暗号文との排他的論理和を計算して平文ブロックを生成する。

暗号文を通信相手に送信する途中に、ある暗号文ブロック１個にエラーが発生した場合には、復号の際、その暗号文ブロックに対応した平文ブロックと次の平文ブロックにエラーの影響が及ぶ。なお、初期値は暗号通信者間で秘密にする必要はないが、配送される初期値の真正性を確保する必要がある。

(3) CFBモード

CFB（Cipher Feedback）モードは、利用する暗号アルゴリズムのブロック長n bitよりも短い r bit（r < n）の平文ブロックを暗号化する際に利用されるモードのひとつである。CFBモードにおける暗号化の手順は、次の通り。

①n bitの初期値がシフトレジスターに入力・記録される。
②シフトレジスターから初期値が出力され、暗号化アルゴリズムによって変換される。
③変換後のデータの左からr bit分のデータが抽出される。
④１番目の平文ブロック（r bit）は、抽出されたr bitデータとの排他的論理和によって変換されて１番目の暗号文ブロックC_1（r bit）となる。
⑤１番目の暗号文ブロックC_1はシフトレジスターにフィードバックされた後、シフトレジスターに記録されているデータ（最初は初期値）の右側に結合される。結合されたデータ（n+r bit）は、左r bitが切り捨てられてn bitデータに変換され、新たにシフトレジスターに記録されるとともに出力される。
⑥シフトレジスターから出力されたn bitデータは、暗号化アルゴリズムによって

変換される。
⑦③に戻って、③～⑥の手続きが、最後の平文ブロックの変換が終了するまで繰り返される。

復号の際には、初期値と各暗号文ブロックを基にシフトレジスターからn bitデータが順々に出力され、同一の鍵による暗号化アルゴリズムによって変換された後、左r bitのデータと暗号文ブロックとの排他的論理和によって平文ブロックが生成される。このように、CFBモードでは、復号の際にも暗号化アルゴリズムが利用される。

暗号文を通信相手に送信する途中に、ある暗号文ブロック１個にエラーが発生した場合、復号の際には最大で T + １個（Tは（n/r）以上の最小の整数）の平文ブロックにエラーの影響が及ぶこととなる。

（4）OFBモード

OFB（Output Feedback）モードは、CFBモードと同様に、暗号アルゴリズムのブロック長n bitよりも短いr bit（r < n）の平文ブロックを暗号化する際に利用することができる。OFBモードにおける暗号化の手順は、次の通り。

①n bitの初期値がレジスターに入力・記録される。
②レジスターから初期値が出力され、暗号化アルゴリズムによって変換される。
③変換後のデータはレジスターにフィードバックされて記録される。
④変換後のデータの左からr bit分のデータが抽出され、１番目の平文ブロック（r bit）との排他的論理和によって変換されたデータが１番目の暗号文ブロック（r bit）となる。
⑤レジスターに記録されているデータが出力され、暗号化アルゴリズムによって変換される。
⑥変換後のデータはレジスターにフィードバックされて記録されるとともに、左からr bit分のデータが抽出され、平文ブロックとの排他的論理和によって暗号文ブロックが生成される。
⑦⑤に戻り、⑤と⑥の手続きが繰り返される。

復号の際には、通信相手から送信された初期値を暗号化アルゴリズムによって順々に変換し、各変換後のデータの左からr bit分のデータと各暗号文ブロックとの排他的論理和によって平文ブロックが生成される。OFBモードにおいても、CFBモードと同様に復号の際には暗号化アルゴリズムが利用される。

なお、暗号文を通信相手に送信する途中で、ある暗号文ブロック１個にエラーが発生した場合、復号の際には、エラーが発生した暗号文ブロックに対応する平文ブロックのみに影響が及ぶこととなる。

Ⅲ．共通鍵ブロック暗号の主要な解読法

共通鍵ブロック暗号の解読法は、Brute Force Methodと Short Cut Methodの２つに分類される。Brute Force Methodは、すべての鍵の候補をしらみつぶしに試してみる方法である。代表的なBrute Force Methodとしては、全数探索法、タイムメモリートレードオフ法、暗号文一致攻撃、辞書攻撃が挙げられる。

一方、Short Cut Methodは、暗号アルゴリズムの構造上の特徴や平文・暗号文の統計的偏りを利用して、候補となる鍵の集合全体から真の鍵の候補を絞り込む方法である。代表的なShort Cut Methodとしては、差分解読法、線形解読法、高階差分攻撃、補間攻撃、分割攻撃、関連鍵攻撃が挙げられる。最近では、差分解読法や線形解読法に対して十分な安全性を有していることが数値的に証明可能な暗号方式も提案されているが、その中のいくつかに対しては、新しく提案された高階差分攻撃や補間攻撃が有効であることが示されている。

なお、本節で取り上げる攻撃法以外にも、各暗号方式に固有の弱点を利用した攻撃法等多くの方法が存在する。

1. Brute Force Method

（1）全数探索法

全数探索法は、入手した暗号文を候補となる鍵で順々に復号することによって真の鍵を探索する方法である。したがって、全数探索法に対する安全性は鍵長のみに依存し、鍵長が長いほど安全性は高まる。鍵長がn bitの場合、候補となる鍵の個数は2^n個となることから、平均的にみて2^{n-1}回の暗号化処理によって真の鍵を発見することができる。

なお、暗号化処理回数と処理時間の関係を、DESを想定して一定の条件の下で試算した結果は、以下の表１の通り。この試算結果は、対象となるアルゴリズムや利用可能な費用等によって大きく変化する点に留意する必要がある。

表１　DESの暗号化処理回数と処理時間の関係

	2^{40}	2^{56}	2^{64}	2^{80}	2^{128}
処理時間	約12秒	約225時間（9.4日）	約80か月（6.6年）	4.3×10^5年	1.2×10^{20}年

〈前提条件〉
（1）暗号方式としてDESを想定。DESの暗号化を各回数実行するために必要な時間を試算。
（2）暗号化装置として、EFFが約25万ドルで製作したDES解読装置を利用した場合を想定（詳細は第Ⅴ章）。本解読装置の１秒間の暗号化処理回数は約888億回（ 2^{36}回）。

(2) タイムメモリートレードオフ法

タイムメモリートレードオフ法は、1980年にHellmanによって最初に提案された解読法であり、送信される平文1個を予想し得る場合に、予め暗号文と平文の対応がわかるような索引表を作成しておくことにより、暗号文を入手した後に短時間で鍵の検索を可能とする方法である。タイムメモリートレードオフ法では、予め作成しておく索引表が大きいほど鍵検索の時間を短縮することが可能となるほか、いったん検索表を作成すれば、鍵が変更されたとしても予め想定した平文が入手可能である限り適用できる。タイムメモリートレードオフ法に対する安全性は鍵長に依存し、鍵長が長いほど安全性は高まる。

(3) 暗号文一致攻撃

暗号文一致攻撃は、「n bitのサイズのデータをランダムに$2^{n/2}$個集めたときに、その中に同じデータが２個以上存在する確率が約0.5になる」という性質（バースデー・パラドックスと呼ばれる）を利用した攻撃法であり、同一の鍵によって生成された複数の暗号文の中から一致するものを見つけ出し、それらの暗号文に対応する平文の情報を入手する方法である。これまで提案されてきた多くのブロック暗号はブロック長が64 bitであるが、暗号文一致攻撃を適用すれば2^{32}個程度の暗号文を集めると約0.5の確率で同じ暗号文を見つけることができる。このように、暗号文一致攻撃に対する安全性は暗号文のブロック長に依存し、ブロック長が長いほど安全性は高まる。

(4) 辞書攻撃

辞書攻撃は、ある鍵によって暗号化された暗号文と平文のペアを予め大量に集めて適当な外部記憶媒体に記録しておき、それを辞書のように利用することによって、盗聴等によって入手した暗号文に対応する平文を得る、という攻撃法である。ブロック長がn bitの場合、この攻撃を実行するためには2^n個の平文・暗号文ペアを記録できる容量をもつ媒体が必要となる。したがって、ブロック長が長いほど、辞書攻撃に対する安全性は高くなる。

2. Short Cut Method

(1) 差分解読法

差分解読法は、1990年にBihamとShamirによって提案された解読法（Biham and Shamir［1991］）であり、ある特定の差分を有する平文のペアに対して、特定の差分を有する暗号文のペアが生じる確率が高くなる場合に、それらの平文・暗号文のペアを利用して候補となる鍵を絞り込む方法である。たとえば、まずF関数において、ある一定の入力データの差分ΔXに対して高い確率で生成される出力データの差分ΔYを予め調べておき、次に鍵をKに固定したF関数にΔXの差分を有する入力データのペアを入力して、その出力データの差分がΔYに偏っていないかどうかを

調べる。固定した鍵Kが真の値であれば出力データの差分は高い確率でΔYとなるはずであり、差分がΔYに偏っている場合、そのときの鍵Kが真の鍵である確率が高くなる。差分解読法では、このようにして候補となる鍵を絞っていく[6]。

差分解読法に対する厳密な安全性評価尺度として、平均差分確率が提案されている（Lai, Massey and Murphy［1991］）。非線形関数の入出力データにおいて各差分のペアが発生する確率をすべての鍵について計算し、その中で最も発生確率が高くなる差分のペアを見つける。このような差分ペアの発生する確率が平均差分確率である。平均差分確率が小さいほど、差分解読法に対する安全性が高くなる。平均差分確率は以下のように定義される。

【平均差分確率】 t bit の鍵 kを変数とする非線形関数 $F_k(x)$（入出力データx、yはいずれもn bit）の平均差分確率は、

$$\frac{1}{2^t}\sum_{k}\max_{\Delta x(\neq 0),\Delta y}\left(\frac{\#\{x \mid F_k(x)\oplus F_k(x\oplus\Delta x)=\Delta y\}}{2^n}\right)$$

ただし、Δxはxの差分、⊕は排他的論理和演算、#{x}はxの個数を表す。

差分解読法に対する安全性を評価する場合、データランダム化部全体の平均差分確率を計算することは困難であり、平均差分確率によって安全性が証明されている暗号方式はMISTYなど一部に限られている。そこで、厳密な意味での安全性を証明するものではないが、差分解読法に対する安全性の必要条件として、最大差分特性確率と呼ばれる指標が利用されている。たとえば、まずF関数等データランダム化部の一部となっている非線形関数の平均差分確率を計算する。次に、各非線形関数の平均差分確率がすべて独立であると仮定した上で、これらの平均差分確率を基に、データランダム化部全体もしくはその一部における確率を計算する。このような条件付き確率が最大差分特性確率となる。最大差分特性確率の計算方法は、データランダム化部の構造に依存する。

なお、差分解読法による解読に必要な平文・暗号文ペアの数は、平均差分確率の逆数として求めることができる。たとえば、ある64 bit鍵長・ブロック長のブロック暗号において平均差分確率が2^{-60}であった場合、差分解読法を利用してこのブロック暗号を解読するためには少なくとも2^{60}の選択平文・暗号文ペアが必要であることを意味する。

6 このように、攻撃者が自分にとって都合のよい平文とそれに対応する暗号文を利用できる場合の攻撃は選択平文攻撃と呼ばれている。

(2) 線形解読法

線形解読法は、1993年に松井によって提案された解読法（Matsui［1994］）であり、平文と暗号文のbit値の間に線形関係が発生する確率が1/2から乖離している場合、その線形関係を利用することによって候補となる鍵の数を絞り込む方法である。線形解読法を適用するためには、任意の平文・暗号文のペアを入手する必要がある[7]。

線形解読法に対する厳密な安全性評価尺度として、平均線形確率が提案されている（Nyberg［1995］）。非線形関数の入出力データのbit間で各線形関係が発生する確率をすべての鍵について計算し、その中で最も発生確率が高くなる線形関係を見つける。このような線形関係が生じる確率が平均線形確率である。平均線形確率が小さいほど、線形解読法に対する安全性が高くなる。平均線形確率は以下のように定義される。

【平均線形確率】 t bitの鍵kを変数とする非線形関数 $F_k(x)$（入出力データx、yはいずれもn bit）の平均線形確率は、

$$\frac{1}{2^t}\sum_k \max_{\Gamma x,\Gamma y(\neq 0)}\left(2\frac{\#\{x \mid x \bullet \Gamma x = F_k(x) \bullet \Gamma y\}}{2^n} - 1\right)^2$$

ただし、Γxはxの特定のマスク値、•は内積、#{x}はxの個数を表す。

線形解読法に対する安全性を評価する場合、データランダム化部全体の平均線形確率を計算することは困難であり、平均線形確率によって安全性が証明されている暗号方式は一部に限られている。そこで、厳密な意味での安全性を証明するものではないが、線形解読法に対する安全性の必要条件として、最大線形特性確率と呼ばれる指標が利用されている。たとえば、まず、F関数等データランダム化部の一部となっている非線形関数の平均線形確率を計算する。次に、各非線形関数の平均線形確率がすべて独立であると仮定した上で、これらの平均線形確率を基に、データランダム化部全体もしくはその一部の確率を計算する。このような条件付き確率が最大線形特性確率となる。最大線形特性確率の計算方法は、データランダム化部の構造に依存する。

なお、線形解読法による解読に必要な平文・暗号文ペアの数は、線形確率の逆数として求めることができる。たとえば、ある64 bit鍵長のブロック暗号において平均線形確率が2^{-60}であった場合、線形解読法を利用してこのブロック暗号を解読するためには少なくとも2^{60}の既知平文・暗号文ペアが必要であることを意味する。

7 攻撃者が任意の平文とそれに対応する暗号文を利用する攻撃は既知平文攻撃と呼ばれている。

(3) 高階差分攻撃

高階差分攻撃は、1994年にLaiによって最初にそのアイデアが提案され（Lai［1994］）、その後JakobsenとKnudsenや下山・盛合・金子らによって、高階差分攻撃による既存の暗号方式への攻撃に関する研究成果が発表されており（Jakobsen and Knudsen［1997］、下山・盛合・金子［1997］）、現在ではブロック暗号の有力な攻撃法のひとつとなっている。高階差分攻撃は、暗号化関数（たとえば、F関数）の入力データに関して出力データの高い階数の差分を取ると、鍵に依存しない定数が得られることを利用して候補となる鍵を絞り込む攻撃法である。たとえば、F関数に適用する場合、F関数の出力データの各bit値は入力データと拡大鍵を変数とするブール多項式によって表現されるが、そのブール多項式の次数をnとすると、出力データのn階差分値は定数となる。このn階差分値を利用して拡大鍵を変数とする線形連立方程式を立てて解くことにより、拡大鍵を求めることができる。この拡大鍵の情報を利用して、真の鍵の候補を絞り込むことが可能となる。ただし、高階差分攻撃を適用するためには、攻撃対象となるラウンド関数あるいはF関数のブール多項式次数を計算する必要がある。高階差分攻撃は選択平文攻撃のひとつである。

高階差分攻撃は、差分解読法や線形解読法に対する安全性が証明されている暗号方式のいくつかに対しても、有効となるケースが存在することが示されている。たとえば、Nyberg and KnudsenのKN暗号は、差分解読法に対して十分な安全性を有していることが証明されているが、高階差分攻撃によって攻撃に必要な選択平文数を大幅に減少させることが可能となるほか（６段のKN暗号の場合、選択平文・暗号文ペア数：$2^{60}\rightarrow 2^{8}$）、計算量も大幅に削減できる（ワークステーションを利用して0.02秒）ことが示されている（下山・盛合・金子［1997］）。こうした研究成果から、差分解読法や線形解読法に対して十分な安全性が確保されているとみられる暗号方式でも、高階差分攻撃のような新しい解読法に対して必ずしも安全であるとは言えないことが示唆される。

高階差分攻撃に対する安全性を評価するための厳密な指標はこれまでのところ提案されていないが、暗号化関数のブール多項式表現における次数が大きいほど攻撃に必要な選択平文数や計算量が増加することから、近似的な評価指標としてブール多項式の次数が利用されている。しかし、この次数が大きい値であったとしても、必ずしも高階差分攻撃に対する安全性が保証されるわけではない点には留意が必要である。

(4) 補間攻撃

補間攻撃は、1997年にJakobsenとKnudsenによって提案された解読法（Jakobsen and Knudsen［1997］）である。鍵を固定した場合に、暗号化関数は平文を変数とするn次関数によって表現されるとすれば、暗号化関数は最大（n+1）項の多項式で表されるため、異なる（n+1）組の平文・暗号文のペアを入手し、(n+1）個の係数を未知とする連立方程式を立てて解くことによって、あるひとつの鍵に対する暗号化関数を構成することができる。このようにして導出した暗号化関数を利用して真の鍵

の候補を絞っていく攻撃法が補間攻撃である。補間攻撃を適用するためには、攻撃対象となるラウンド関数等が多項式によって表現できること、そして多項式の項数が比較的少ないことが必要である。補間攻撃では、暗号化関数の次数を下げるように平文を選択して攻撃に利用することができるほか、既知平文を利用することも可能である。

高階差分攻撃と同様に、補間攻撃を利用することによって、差分解読法・線形解読法に対して安全性が証明されている暗号方式のいくつかを解読することができるとの研究成果が発表されている。たとえば、LeeとChaによって提案されたSNAKE暗号は、差分解読法や線形解読法に対して十分な安全性を有しているとされている（Lee and Cha［1997］）が、補間攻撃を利用することによって、より少ない既知・選択平文数と計算量によって解読可能であることが示されている（盛合・下山・金子［1998］）。

補間攻撃に対する厳密な安全性評価指標は、高階差分攻撃同様、これまでのところ提案されていないが、鍵を固定したときの暗号化関数を入力データの多項式で表現した場合、その項数が大きいほど攻撃に必要な既知平文や計算量が増加することから、近似的な指標として多項式の項数が利用されている。しかし、この項数が大きな値であったとしても、必ずしも補間攻撃に対する安全性が保証されるわけではない。

（5）分割攻撃

分割攻撃は、線形解読法を応用した攻撃法であり、1997年にHarpesとMasseyによって提案された（Harpes and Massey［1997］）。分割攻撃では、まず鍵を固定した上で複数の入出力データのペアを入手し、暗号化関数の入力データと出力データをそれぞれ特定の関数によって入力集合と出力集合に写像する。これらの集合を複数の部分集合に分割したときに、入力集合の特定の要素と出力集合の特定の要素との間に強い相関がないかどうかを調べる。強い相関がある場合には、その情報に基づいて真の鍵の候補を絞り込むことが可能となる。分割攻撃は既知平文攻撃のひとつである。

分割攻撃に対する厳密な安全性評価指標はこれまでのところ提案されていない。分割攻撃に対して安全であるための必要条件は、入力集合の要素と出力集合の要素との間に強い相関が発生しないことである。そのため、分割攻撃の近似的な評価指標として、入出力集合における要素間の偏りの大きさが利用されている。しかし、要素間での偏りが小さい場合であったとしても、必ずしも分割攻撃に対する安全性が保証されるわけではない。

（6）関連鍵攻撃

関連鍵攻撃は、1993年にBihamによって最初に提案された攻撃であり、ある特殊な関係を有する２つの異なる鍵による暗号化が可能な場合に、それらの鍵の関係や暗号化されたデータを手掛かりにして、真の鍵の候補を絞り込む方法である（Biham［1994］）。関連鍵攻撃は、適用可能な環境が非常に限定されていることから、他の攻撃法に比べて実現可能性は低いとみられている。

Ⅳ．主要な共通鍵ブロック暗号と安全性評価

本節では、アルゴリズムが公開されている方式のうち、従来から安全性に関する研究が盛んに行われてきた主要な共通鍵ブロック暗号として、DES、FEAL、IDEA、SAFER、LOKI、RC5、MISTYの７つを取り上げ、それらのアルゴリズムの概要や安全性に関する分析結果について整理する（安全性評価結果については、本章末の表２参照）。

1. DES

（1）アルゴリズムの概要と構造

DES（Data Encryption Standard、NIST［1993］）は、初めての商用暗号として1977年に米国政府標準暗号に制定された共通鍵ブロック暗号であり、DESの特許の所有者であるIBM社が標準化に際してロイヤリティ・フリーでの使用を認めたこと等によって、世界で最も幅広く利用されてきた。DESは、金融業務に利用される情報セキュリティ技術として、国際標準化機構金融専門委員会ISO/TC68で制定されているいくつかの国際標準において参照されている[8]ほか、米国では金融業務用の暗号アルゴリズム（ANSI X3.92）として標準化されており、多くの欧米の金融機関がDESを利用して自行システムのセキュリティ対策を講じてきた。これまでFedwire、CHAPS、日銀ネット等の世界の主要な大口決済ネットワークにおいても、DESが事実上の標準暗号として利用されてきた。

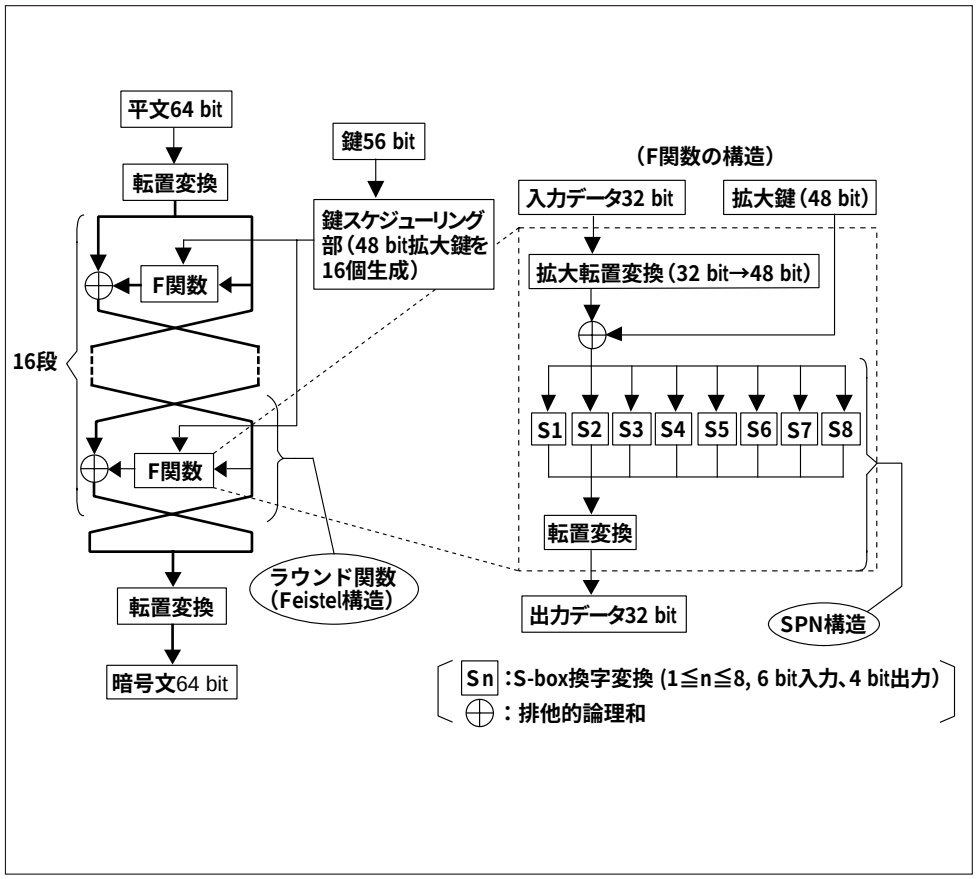
①データランダム化部

DESは、平文を64 bit単位のブロックに分割し、各ブロックを56 bitの鍵[9]を利用して暗号化する構造となっている。暗号化は以下の手順で実行される（図９参照）。

（ステップ１）64 bit平文ブロックを転置変換する。
（ステップ２）64 bitの平文ブロックを２つの32 bitサブブロックに分割する。
（ステップ３）サブブロックと48 bit拡大鍵を変数とするF関数を含むラウンド関数を16回繰り返す。
（ステップ４）32 bitのサブブロックを結合して再び64 bitのデータブロックとし、最初の転置変換の逆変換を行う。

8 DESが参照されている金融業務において利用される情報セキュリティ技術に関するISOの標準規格としては、ISO 8731-1、ISO 8732、ISO 9564-2、ISO/IEC 10116、ISO/IEC 10118-2等が挙げられる。
9 実際にアルゴリズムに利用される鍵は64 bitであるが、このうち8 bitはパリティビットであり、鍵として利用可能なのは実質56 bitであることから、一般的にDESの鍵長は56 bitとされている。

図９　DESのデータランダム化部とF関数の構造



復号は、同じ鍵の下で暗号化の手順をちょうど逆に辿ることにより実現される。DESのラウンド関数にはFeistel構造が採用されている。

ラウンド関数に含まれるF関数は、DESのアルゴリズムにおける唯一の非線形変換手段であり、内部構造にはSPN構造が採用されている。F関数の変換手順は以下の通り。

（ステップ１）32 bitサブブロックに対し、一部の bit値をコピーしてサブブロックを48 bitに拡大するとともに転置変換を行う。

（ステップ２）48 bitサブブロックを48 bit拡大鍵との排他的論理和によって変換する。

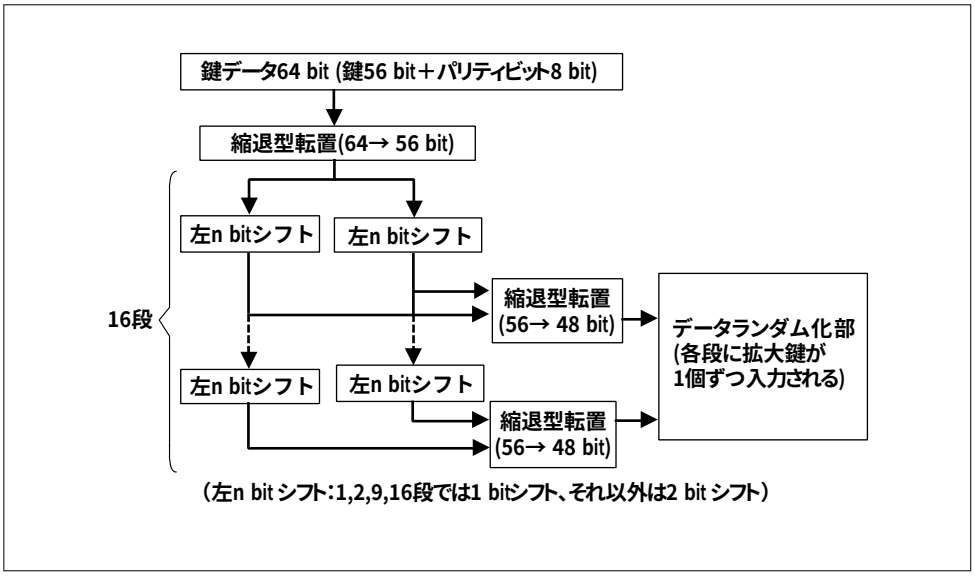
（ステップ３）48 bitサブブロックを８個の6 bitデータに分割し、８種類のS-boxと呼ばれる換字変換（換字表を利用）によって4 bitデータに変換する。

（ステップ４）８個の4 bitデータを結合して32 bitサブブロックとし、転置変換を行う。

②鍵スケジューリング部

鍵スケジューリング部では、56 bitの鍵から48 bitの拡大鍵が16個生成され、データランダム化部の各ラウンド関数に１個ずつ入力される（図10参照）。

図10　DESの鍵スケジューリング部の構造



拡大鍵は以下の手順で生成される。

（ステップ１）当初56 bitの鍵には8 bitのパリティビットが付加されているが、縮退型転置変換によってパリティビットが削除されて56 bitに圧縮されると同時に、転置変換が行われる。

（ステップ２）56 bitの鍵は28 bitずつに分割され、それぞれ左 bitシフト変換が行われる。bitのシフト数は段数に依存し、1、2、9、16段目では左1 bitシフト、それ以外の段では左2 bitシフトとなる。

（ステップ３）２つの28 bitデータは結合され、縮退型転置変換によって48 bitに圧縮されると同時に、転置変換が行われる。

（ステップ４）48 bitデータは拡大鍵となり、データランダム化部のF関数に入力される。

（2）安全性に関する評価結果

これまで、様々な解読法に対するDESの安全性が研究・評価されてきたが、現在最も脅威となっている攻撃法はBrute Force Methodである。1998年7月に行われたRSA社主催のDES解読コンテストでは、約25万ドルの予算で製作されたDES解読専用装置によって、56時間で解読されている（詳細は第Ⅴ章）。

Short Cut Methodに対する安全性の理論的研究も数多く行われており、差分解読

法を適用する場合、2^{47}個の選択平文と2^{37}回の暗号化処理に相当する計算量が必要である（Biham and Shamir［1993］）ほか、線形解読法を適用する場合には、2^{43}個の既知平文と2^{42}回の暗号化処理に相当する計算量が必要であることが示されている（Matsui［1994］）。また、分割攻撃によって13段のDESを解読するためには、1.34×2^{40}個の既知平文が必要であるとの試算結果も発表されている（浜出他［1998］）。

2. FEAL

（1）アルゴリズムの概要と構造

FEAL（Fast Data Encipherment Algorithm）は、1988年にNTTが開発したブロック長64 bitのブロック暗号（Miyaguchi, Shiraishi and Shimizu［1988］）であり、64 bitの鍵を利用するFEAL-Nと128 bitの鍵を利用するFEAL-NXが提案されている（図11参照）。現在では、CAFIS[10]のデータ暗号化手段等に利用されているほか、ANSWER-WEBにも利用されている[11]。

①データランダム化部

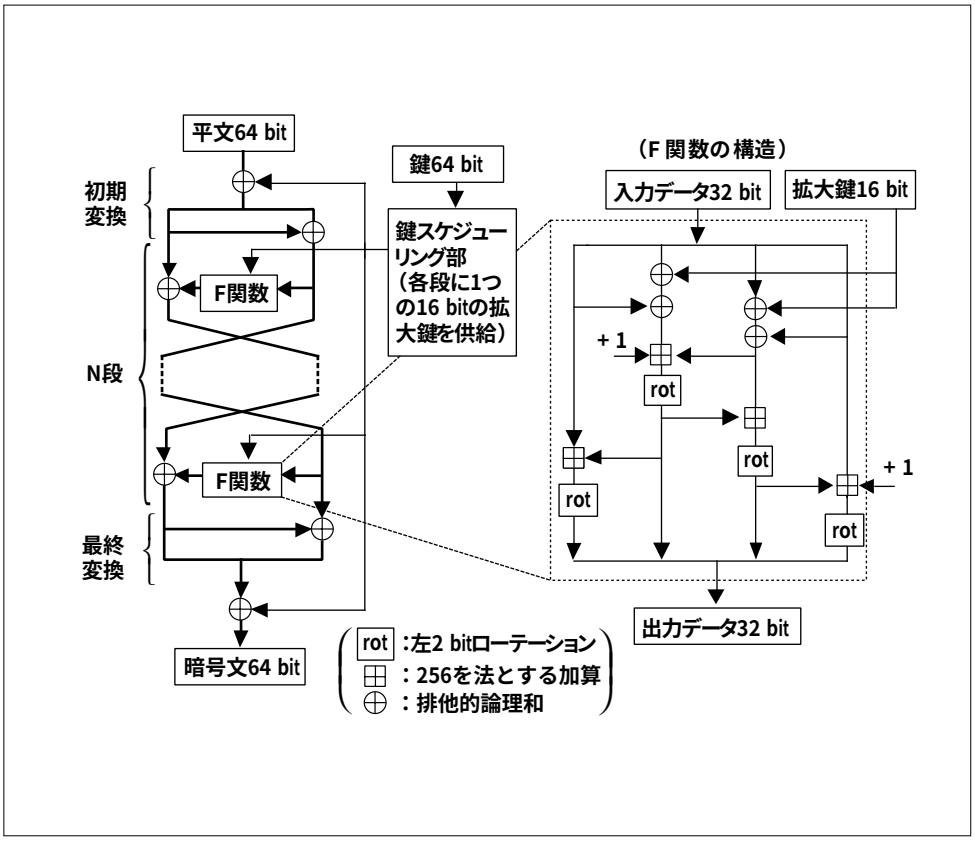
FEALのラウンド関数には、DES同様、Feistel構造が採用されており、段数は４、８、16、32段が利用可能となっているが、安全性の観点から32段のFEAL-32Xが推奨されている。FEAL-Nの暗号化手順は次の通り。

（ステップ１）最初に、64 bitの平文ブロックを、４つの16 bit拡大鍵との排他的論理和によって変換し、２つの32 bitサブブロックに分割する。
（ステップ２）右32 bitサブブロックを、左32 bitサブブロックとの排他的論理和によって変換する（ステップ１と２が初期変換）。
（ステップ３）ひとつの16 bit拡大鍵を変数とするF関数を含むラウンド関数による変換をN回繰り返す。
（ステップ４）右32 bitサブブロックを、左32 bitサブブロックとの排他的論理和によって変換する。
（ステップ５）２つの32 bitサブブロックを結合して64 bitブロックとし、４つの16 bit拡大鍵との排他的論理和によって変換する（ステップ４と５が最終変換）。これが暗号文ブロックとなる。

10 CAFIS（Credit And Financial Information System）は、クレジットカード会社、金融機関、クレジットカード加盟店等を結び、クレジットカードの決済情報や信用照会情報、銀行POSの資金決済情報等を扱うネットワークシステムである。NTTデータが管理・運営しており、1998年5月現在で、クレジットカード会社や金融機関等約1500社が加盟しているほか、CAT端末等の設置台数は約50万台となっている。

11 ANSER（Automatic Answer Network System for Electronic Request）は、NTTデータが運営・管理している金融サービス専用ネットワークシステムであり、顧客端末と金融機関とを結び、各端末から金融機関への口座残高照会、振込・振替請求、取引照会等の金融ネットワークサービスを提供している。ANSER-WEBは、ANSERを用いてインターネットバンキングを実現するサービスである。

図11　FEAL-Nのデータランダム化部とF関数の構造

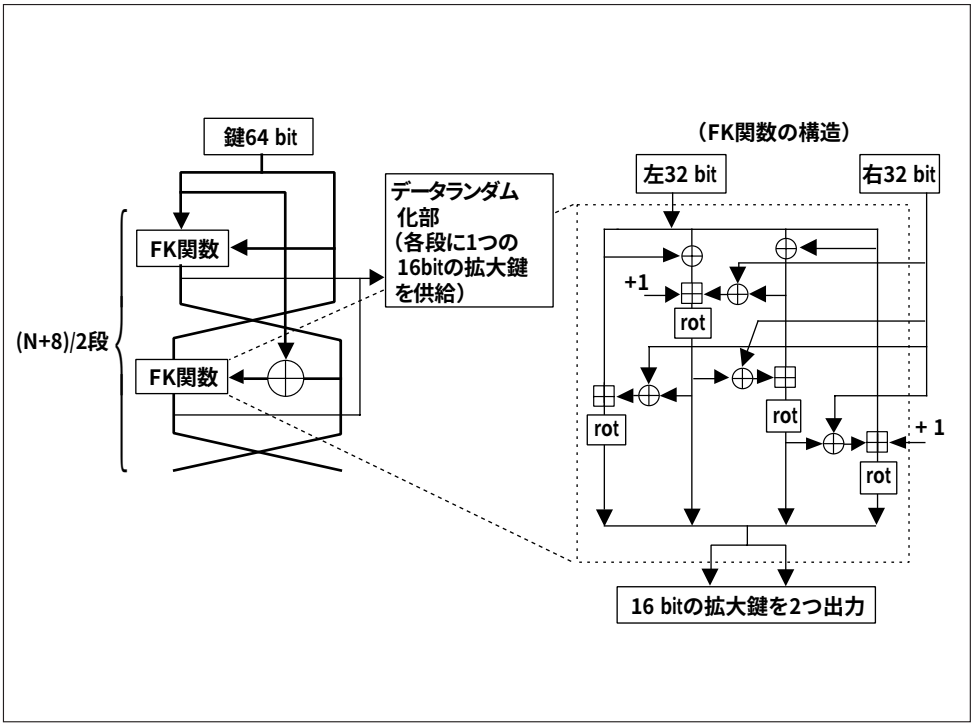


DESではラウンド関数での変換の前後に転置変換が行われているが、FEALの場合、拡大鍵との排他的論理和による変換が採用されている。また、FEALのF関数内部の変換には、算術演算（左2 bitローテーション、256を法とする加算）が利用されており、DESのF関数で利用されているテーブル参照型のS-boxに比べて少ないメモリー量での実装が可能となっている。F関数では、まず32 bitのサブブロックが４つの8 bitデータに分割され、それぞれの8 bitデータは２種類の算術演算および排他的論理和によって変換された後、再び32 bitサブブロックに結合されるという手順で変換される。16 bitの拡大鍵はひとつのF関数にひとつ入力され、２つの8 bitデータに分割された後、排他的論理和変換に利用される。

②鍵スケジューリング部

鍵スケジューリング部は、入力される鍵64 bitを基にして16 bitの拡大鍵を(N+8)個生成する。構造はデータランダム化部と類似しており、FK関数を含むラウンド関数が(N+8)/2回繰り返される（図12参照)。FK関数も、データランダム化部のF関数と類似の構造を有しており、左2 bitローテーションと法256の加算という２種類の算術演算を変換手段として利用している。

図12　FEAL-Nの鍵スケジューリング部



（2）安全性に関する評価

FEAL-Nは、鍵長とブロック長が64 bitであり、全数探索法によって真の鍵を見つけるためには平均2^{63}回の暗号化処理が必要となる。

差分解読法に対する安全性については、差分特性確率を利用した近似的な評価により、FEAL-16の解読には2^{29}個の選択平文が必要であるほか、FEAL-32の解読には2^{66}個の選択平文が必要となることが示されている（Biham and Shamir［1993］）。

線形解読法については、FEAL-8は、2^{24}個の既知平文によって解読可能であることが示されている（Biham［1995］）。しかし、FEAL-16は、近似的な評価からDESとほぼ同等の安全性を有していることが示されており（Ohta and Aoki［1994］）、FEAL-32は近似的な評価から線形解読法に対して十分な安全性を有しているとみられている。

3. LOKI91

（1）アルゴリズムの概要と構造

LOKI91は、1991年にBrownらによって提案された鍵長とブロック長が64 bitのブロック暗号（Brown, Kwan and Pieprzyk［1993］）である。LOKI91は1990年に提案されたLOKI89と呼ばれる暗号方式（Brown, Pieprzyk and Seberry［1990］）を原形としている。14段以下のLOKI89が差分解読法に対して十分な安全性を有しているとは

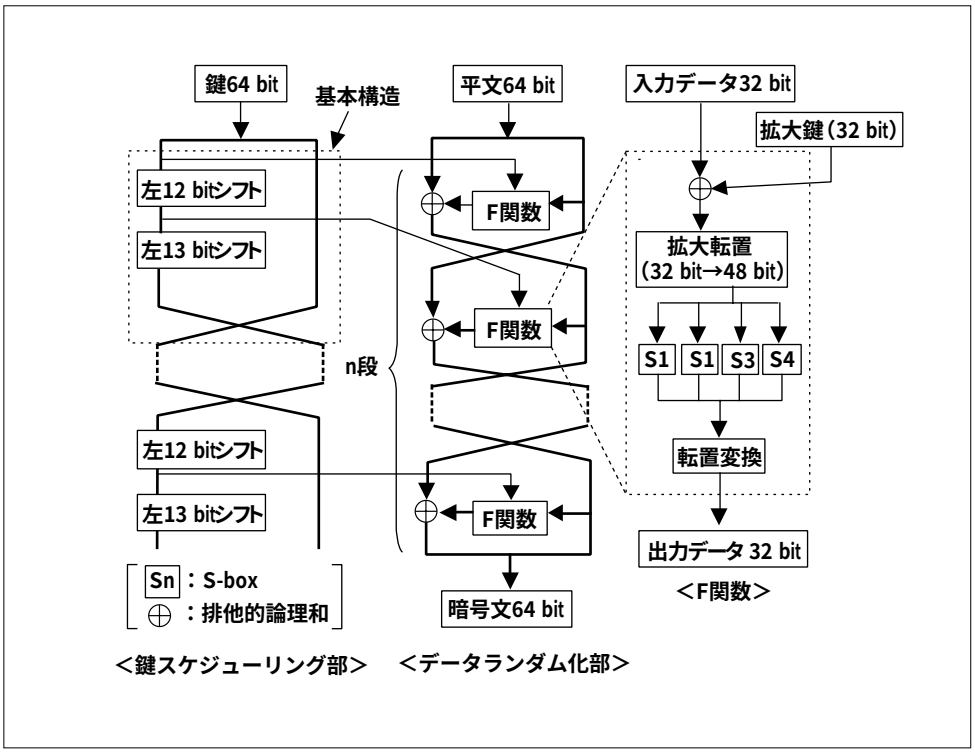
いえないことが示された（Biham and Shamir［1992］）ため、アルゴリズムの改良が行われ、その結果開発されたのがLOKI91である。LOKI91では、差分解読法に対する安全性を高めるために、(i) データランダム化部の最初と最後の排他的論理和変換の削除、(ii) 換字変換の一部変更、(iii) 鍵スケジューリング部の bitシフト量の変更、が行われた。

①データランダム化部

LOKI91のデータランダム化部はFeistel構造を有しており、64 bitの平文ブロックは２つの32 bitサブブロックに分割された後、ラウンド関数に入力される。ラウンド関数とF関数の構成はDESと類似している（図13を参照）。F関数（32 bit入出力）はSPN構造を有しており、以下の手順で変換される。

（ステップ１）32 bitの拡大鍵との排他的論理和によって変換された後、拡大転置変換によって32 bitから48 bitに拡大される。
（ステップ２）48 bitサブブロックは、４つの12 bitデータに分割された後、S-box（12 bit入力、8 bit出力）によって換字変換される。
（ステップ３）４つの8 bitデータは結合されて32 bitサブブロックとなり、転置変換を経た後出力される。

図13　LOKI91の暗号化アルゴリズムの構造



LOKI91で採用されているS-boxは12 bit入力8 bit出力となっており、DESに比べて大きいほか、４つとも同一となっている。利用されている演算は、加算、乗算、排他的論理和、べき乗演算である。

②鍵スケジューリング部

LOKI91の鍵スケジューリング部は、64 bit鍵から16 個の32 bit拡大鍵を生成する（図13参照）。64 bit鍵は２つの32 bitサブブロックに分割され、左32 bitのサブブロックが２種類の bitシフト（左12 bitシフト、左13 bitシフト）によって変換される過程で２つの拡大鍵が生成された後、２つのサブブロックの位置が入れ替えられる。これが鍵スケジューリング部の基本構造となっており、この基本構造が６回繰り返される。

（2）安全性に関する評価

全数探索法に対しては、LOKI91の鍵長が64 bitであることから、解読には2^{64}回の暗号化処理が必要となる。

差分解読法や線形解読法に対しては、14段以上のLOKI91は近似的な評価から十分な安全性を有しているとみられている（Tokita, Sorimachi and Matsui［1995］）。13段のLOKI91を差分解読法によって解読するためには2^{64}個以上の選択平文が必要であることが示されているほか、14段のLOKI91を線形解読法によって解読するためには2^{86}個以上の既知平文が必要であることが示されている。一方、関連鍵攻撃によって、全数探索法よりも少ない計算量（1.375×2^{61}回の暗号化処理）で解読が可能となることが示されているが（Biham［1994］）、関連鍵攻撃が適用できるケースは非常に限定されていることから実際の脅威にはならないとみられている。

4. IDEA

（1）アルゴリズムの概要と構造

IDEA（International Data Encryption Algorithm）は、1991年にLaiとMasseyがスイスのAscom社と共同開発したブロック暗号（Lai, Massey and Murphy［1991］）であり、ブロック長64 bit、鍵長128 bit、段数８段のブロック暗号である。

①データランダム化部

IDEAのデータランダム化部は、DESやFEAL等のFeistel構造とは異なる構造となっており、64 bit平文ブロックを４つの16 bitサブブロックに分割した上で、８段のラウンド関数と最終変換によって暗号化を行う仕組みとなっている（図14参照）。ラウンド関数は、16 bitのサブブロック単位で変換を実行し、変換手段には2^{16}を法とする加算、$(2^{16}+1)$を法とする乗算、排他的論理和が利用されている。最終変換には、ラウンド関数の一部分が利用されている。

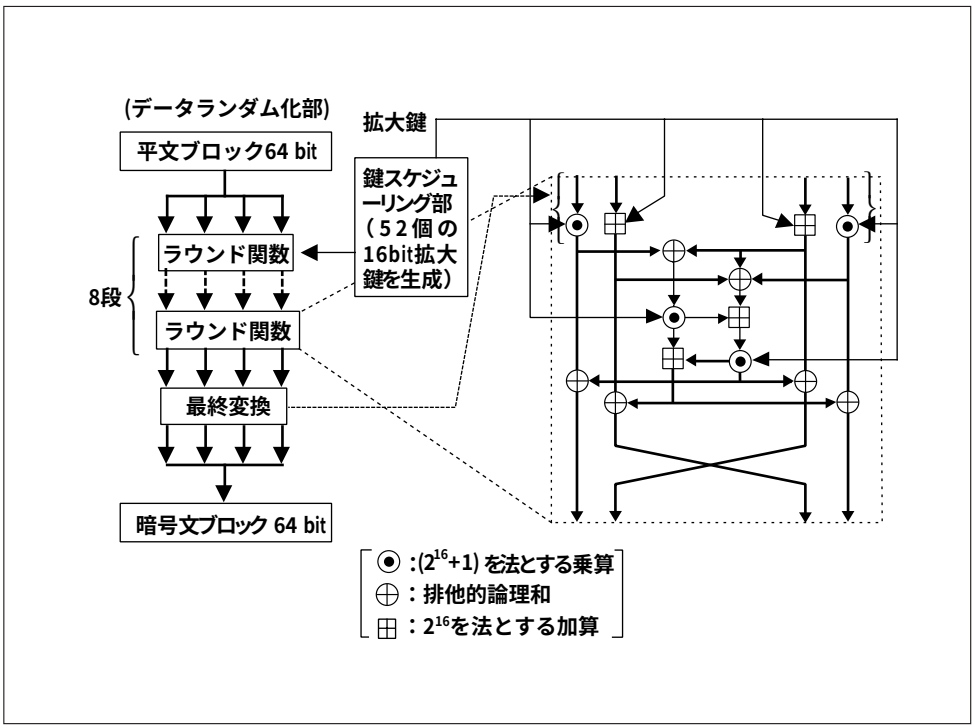
②鍵スケジューリング部

鍵スケジューリング部では、ひとつのラウンド関数に16 bit拡大鍵６個が入力されるほか、最終変換には４個の拡大鍵が利用されることから、128 bit鍵から16 bitの拡大鍵が52個（＝６個×８段＋４個）生成される。拡大鍵は、次の手順で生成される。

（ステップ１）128 bit鍵を16 bit毎に分割し、まず８個の拡大鍵を生成する。

（ステップ２）128 bit鍵を左に25 bitだけ巡回シフトし、変換後のブロックを16 bit毎に分割して８個の拡大鍵を生成する。この手続きを６回繰り返して、44個の拡大鍵を生成する。

図14　IDEAのラウンド関数の構造



なお、復号の際には、データランダム化部には変更は不要だが、鍵スケジューリング部に若干の変更が必要となる。

（2）安全性に関する評価

IDEAは鍵長が128 bitであることから、全数探索法で真の鍵を見つけるためには2^{128}回の暗号化処理が必要となる。

IDEAは差分解読法に対する安全性を考慮して設計されており、ラウンド関数１段の差分特性確率は高々2^{-18}程度となるほか、３段の差分特性確率は2^{-64}を下回ることから、厳密な評価ではないものの、差分解読法に対して十分な安全性を有している

とみられている（Lai, Massey and Murphy［1991］）。最近では、3.5段のIDEAに対して、2^{56}個の特殊な選択平文と2^{67}回の暗号化処理によって80%以上の確率で解読可能であるほか、３段のIDEAに対しては、2^{29}個の選択平文と2^{44}回の暗号化処理によって解読可能であるとの研究成果も示されている（Borst, Knudsen and Rijmen［1997］）。もっとも、８段のIDEAに対して適用可能かどうかは示されていない。一方、線形解読法に対しては、近似的な評価から十分な安全性を有するとみられている（Harpes, Cramer and Massey［1995］）。

また、IDEAには、鍵の特定の69 bitがすべて0となる2^{59}個の弱鍵が存在し、弱鍵の推定には、2^{39}個の選択平文と2^{45}回程度の暗号化処理量が必要であることが示されている（三上・金子［1997］）。

5. SAFER

(1) アルゴリズムの概要と構造

SAFER（Secure And Fast Encryption Routine）は、1994年にMasseyと米Cylinkによって共同開発されたブロック暗号であり、鍵長とブロック長が64 bitのSAFER K-64（Massey［1994］）、128 bitのSAFER K-128（Massey［1995］）のほか、これらの暗号方式の鍵スケジューリング部に改良が加えられたSAFER SK-64とSAFER SK-128が提案されている。以下では、SAFER K-64を例に説明する。

①データランダム化部

SAFERのデータランダム化部は、複数回繰り返されるラウンド関数と最終変換によって構成されており、段数は６段、８段、10段、12段のいずれかが利用可能となっているが、８段以上での利用が推奨されている（図15参照）。64 bit平文ブロックは、最初に８個の8 bitサブブロックに分割された後にラウンド関数に入力され、ラウンド関数内部では以下の６つのステップによってサブブロック単位で変換される。

（ステップ１）64 bitの拡大鍵は８つの8 bitデータに分割され、排他的論理和と256を法とする加算による変換が行われる。

（ステップ２）２種類の換字変換（$S(x) = 45^x \bmod 257$と$S'(x) = \log_{45} X \bmod 257$）が行われる。変換には換字表が利用されている。

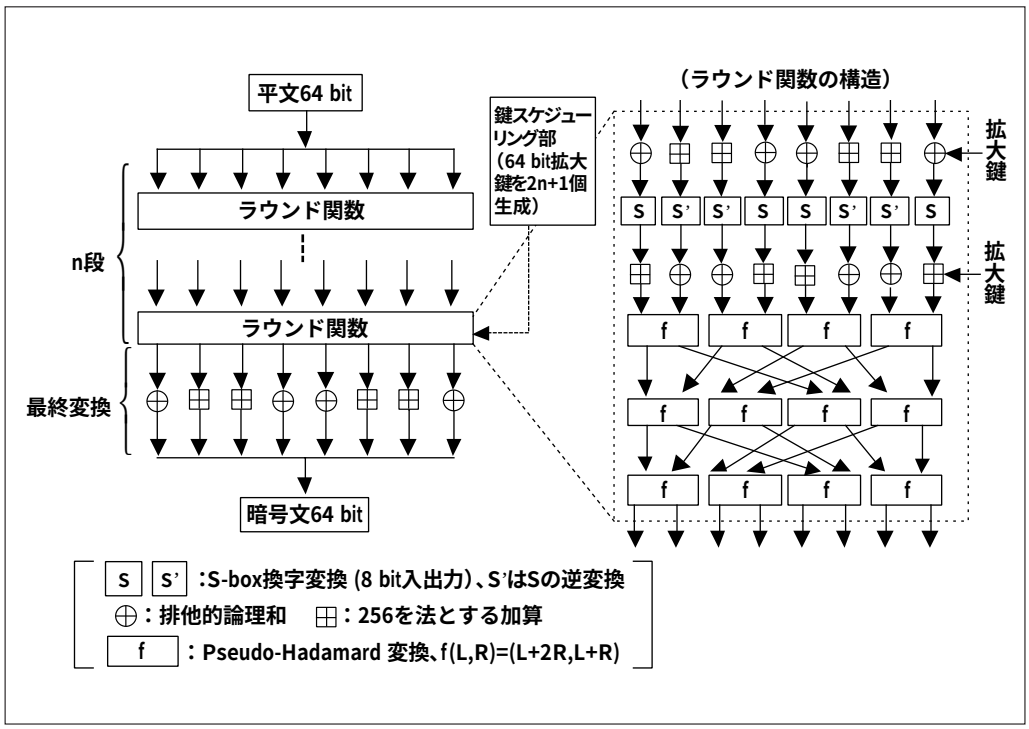
（ステップ３）ステップ１と同様の変換であり、排他的論理和と加算の位置が入れ替えられている。

（ステップ４～６）関数fによる変換と4 bitサブブロック単位での転置変換が、それぞれ３回行われる。関数fは、入力される左サブブロックをL、右サブブロックをRとすれば、f(L,R)=(2L+R, L+R)と表される（Pseudo-Hadamard変換と呼ばれている）。

所定の回数のラウンド関数による変換終了後、最終変換が行われるが、これはラ

ウンド関数のステップ１の部分に対応しており、排他的論理和と加算を組み合わせた構造となっている。

図15　SAFER K-64のデータランダム化部とラウンド関数の構造



②鍵スケジューリング部

鍵スケジューリング部は、段数をnとすると、64 bitの鍵から（2n+1）個の64 bit拡大鍵を生成する。拡大鍵の生成手順は次の通り。

（ステップ１）最初に64 bit鍵を第一の64 bit拡大鍵とする。
（ステップ２）64 bit鍵を８つの8 bitデータに分割する。
（ステップ３）8 bitデータをそれぞれ左3 bitシフトした後に、予め用意された定数との法256の加算によって変換し、変換結果を結合して拡大鍵とする。
（ステップ４）上記手続きを2n回繰り返し、2n+1個の拡大鍵を生成する。

SAFER SK-64は、上記のSAFER K-64の鍵スケジューリング部を改良した方式である。改良の内容は、①ステップ２と３の間に、８個の8 bitデータの排他的論理和を計算して９番目の8 bitデータを生成するステップを挿入する、②ステップ３において定数との加算を行う前に、変換に利用する各8 bitデータの位置を入れ替えるという手続きを加える、という２点である。

(2) 安全性に関する評価

SAFERには、ブロック長・鍵長が64 bitおよび128 bitのSAFER K-64とSAFER K-128のほか、それぞれの鍵スケジューリング部に改良が加えられたSAFER SK-64とSAFER SK-128の合計４種類がある。SAFER K-64とSK-64を全数探索法によって解読するためには2^{64}回の暗号化処理が必要となるほか、SAFER K-128とSK-128を全数探索法によって解読するためには2^{128}回の暗号化処理が必要となる。

差分解読法に対しては、５段のSAFER K-64が、選択平文2^{46}個と2^{35}回の暗号化処理によって解読可能であることが示されている（Knudsen and Berson［1996］）。しかし、８段以上のSAFER K-64やK-128は、差分解読法に対して近似的な評価から十分な安全性を有しているとみられているほか、線形解読法に対しても近似的な評価から十分な安全性を有しているとみられている（Harpes, Cramer and Massey［1995］）。

また、SAFER K-64に対して、ある特別の関係を有する１組の鍵ペアと2^{47}個の選択平文を利用した関連鍵攻撃によって、鍵の8 bit分のデータを入手できるという分析結果が示されている（Knudsen［1996］）ほか、2^{8}個の鍵ペアと2^{29}個の選択平文によって鍵の28 bitのデータを入手可能であるとの分析結果も示されている（Kelsey, Schneier and Wagner［1996］）。これらの分析結果を基に鍵スケジューリング部に改良が加えられたSAFER SK-64に対して、同様の攻撃が成立するかどうかに関する研究成果はこれまでのところ発表されていない。

6. RC5

(1) アルゴリズムの概要と構造

RC5は、1994年にRivestによって提案されたブロック暗号（Rivest［1995］）であり、ブロック長、鍵長、段数とも可変となっている。ブロック長は32、64、128 bitが利用可能であるほか、鍵長は8 bit単位で上限2,048 bitまでが利用可能となっている。段数は上限255段までが利用可能であり、ブロック長64 bit、鍵長128 bit、段数12段の組み合わせ、もしくはブロック長128 bit、鍵長128 bit、16段の組み合わせが一般的とされている。以下では、ブロック長64 bit、鍵長128 bit、段数12段のRC5を例に説明する。なお、RC5では、ブロック長2w bit、鍵長b byte、段数rの場合、「RC5-w/r/b」との標記が一般的に使われており、以下で説明するRC5はRC5-32/12/16と表される。

①データランダム化部

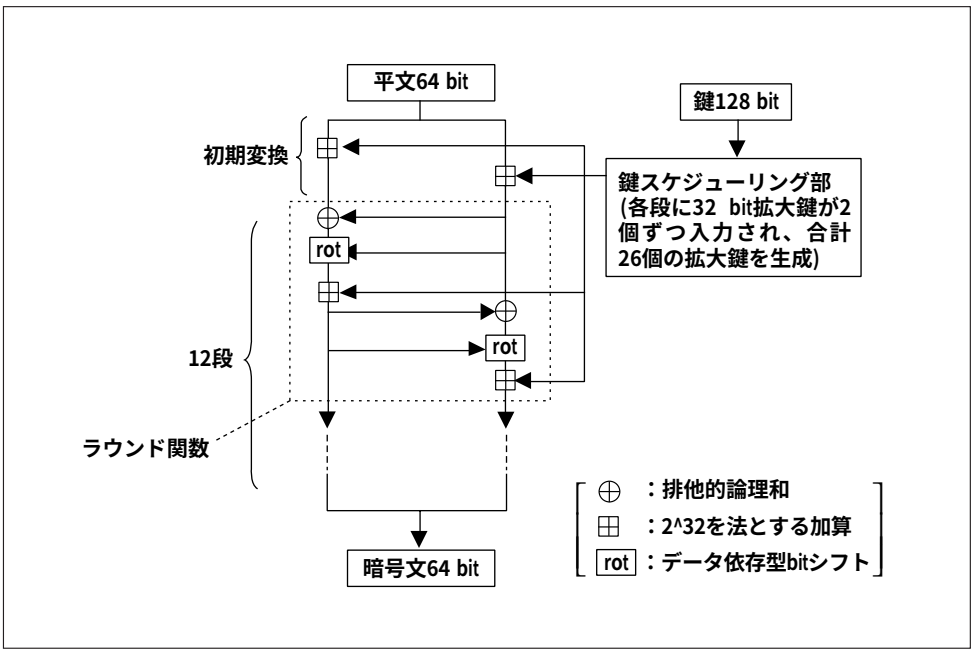
データランダム化部においては、平文ブロックを２個の32 bitサブブロックに分割し、初期変換を行った後、ラウンド関数を12回繰り返すことによって暗号化を実施する（図16参照）。初期変換では、２つのサブブロックは、２個の32 bitの拡大鍵との排他的論理和によって変換される。ラウンド関数は、Feistel構造でもSPN構造でもなく、排他的論理和、2^{32}を法とする加算、データ依存型 bitシフトによって構成されており、加算とデータ依存型 bitシフトによって、ラウンド関数の出力データの非線形性を高めている。RC5で利用されているデータ依存型 bitシフトは、「入力

データの下位5 bitの値だけ左 bitシフトさせる」というものである。

②鍵スケジューリング部

鍵スケジューリング部では、128 bitの鍵から、２つの定数との加算、bitシフト等の演算によって32 bit拡大鍵が合計26個生成される。特に、ある段の拡大鍵が次の段の拡大鍵の生成に利用される点が特徴である。拡大鍵は初期変換部に２個入力されるほか、各ラウンド関数に２個ずつ入力される。

図16　RC5-32/12/16のデータランダム化部の構造



（2）安全性に関する評価

RC5の差分解読法に対する安全性については、近似的な評価において、RC5-32/12/16を解読するために必要な選択平文数が2^{44}個であると見積もられており、これだけの数の選択平文を利用すればワークステーション１台を使って１分以内で解読可能であることが示されている（Biryukov and Kushilevitz［1998］）。また、RC5-32/12/16に対して、条件付きの差分確率の最大値が$2^{-55.15}$であることが示されている（反町・松井［1997］）。

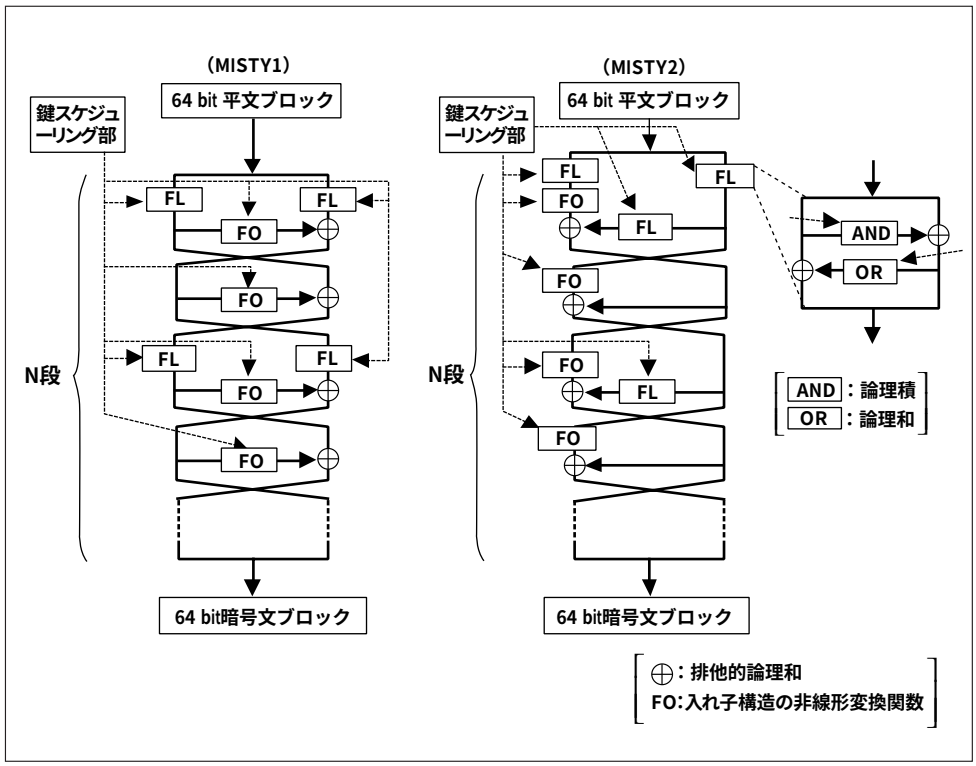
線形解読法に対しては、RC5-32/6/16における条件付きの線形確率の最大値が$2^{-49.15}$となるほか、線形確率が2^{-64}以下となる最大の段数が7.5となることが示されており、７段までであれば、全数探索法よりも線形解読法の方が有効となる可能性があることが示されている（反町・松井［1997］）。また、RC5-32/6/16に対して線形解読法を適用するためには、2^{57}個の既知平文が必要との研究成果も示されている（Biryukov and Kushilevitz［1998］）。

7. MISTY

(1) アルゴリズムの概要と構造

MISTYは、1995年に三菱電機の松井らが発表したブロック暗号（松井［1996］）であり、ブロック長64 bit、鍵長128 bit、段数は可変（4の倍数）となっている。MISTYの特徴は、差分解読法と線形解読法に対する安全性が数値的に保証されるように設計されている点である。

図17　MYSTYのデータランダム化部とラウンド関数



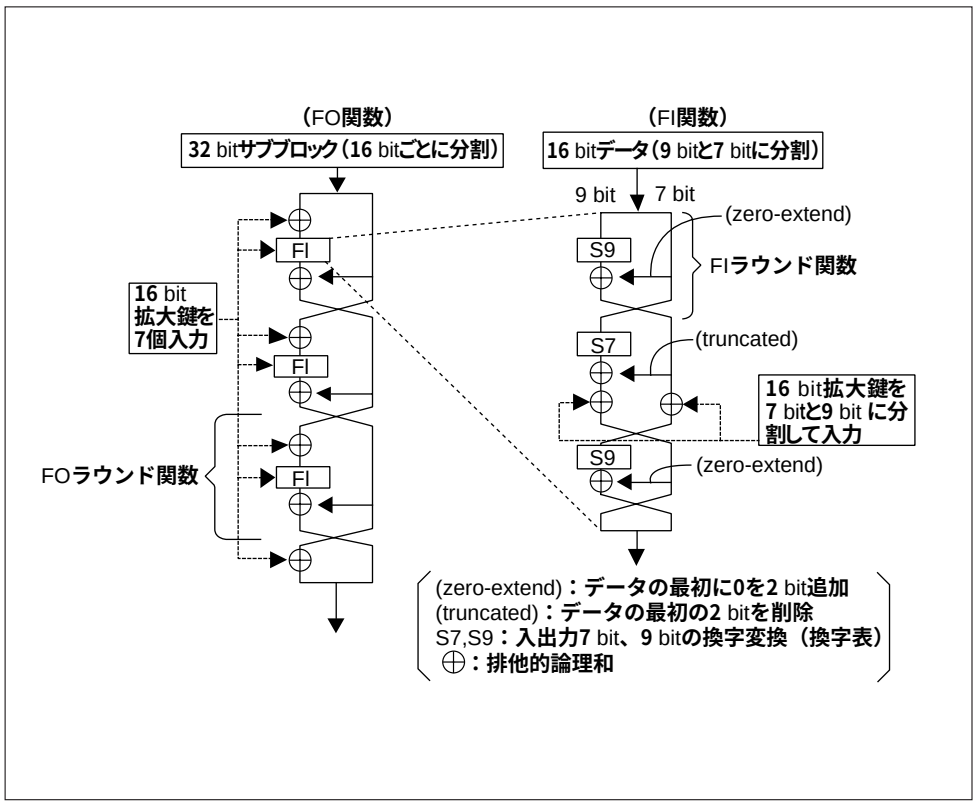
①データランダム化部

データランダム化部では、64 bitの平文ブロックが32 bitのサブブロック２個に分割されて変換される。MISTYには、ラウンド関数の異なる２種類の暗号方式MISTY1とMISTY2が提案されており、MISTY1のラウンド関数には、FO関数（F関数に相当）とFL関数を含むFeistel構造が採用されている。MISTY2のラウンド関数は、FO関数の並列処理が可能なように設計されており、Feistel構造とは異なっている（図17参照）。MISTY1は８段、MISTY2は12段で利用することが推奨されている。

FL関数では、32 bitのサブブロックがさらに16 bitの2つのデータに分割され、16 bit拡大鍵との bit単位での論理和および論理積によって変換された後、再び32 bitのサブブロックに結合されて出力される。

また、FO関数は、差分解読法および線形解読法に対して十分な安全性を確保できるように、MISTY2のラウンド関数に類似したFOラウンド関数を３段有する構造となっており、FOラウンド関数内のFI関数もFIラウンド関数を３段有する構造となっている（図18参照）。このように、構造が類似している３段のラウンド関数が再帰的に利用されており、この点についてはMISTY1とMISTY2で共通となっている。FI関数では、安全性に配慮するために16 bitの入力データを7 bitと9 bitのデータに分割し、それぞれを7 bit入出力と9 bit入出力の換字変換S-boxによって変換する仕組みになっている。換字変換にはべき乗演算が採用されており、換字表が利用されている。

図18　MYSTYのFO関数とFI関数



②鍵スケジューリング部

鍵スケジューリング部は、実用的な観点からできるだけ簡素な構造となるように設計されており、FI関数の構造をそのまま利用する形となっている。拡大鍵は、128 bitの鍵から段数に応じた数だけ生成される。

（2）安全性に関する評価

MISTYの鍵長は、MISTY1、MISTY2ともに128 bitであり、全数探索法を使って解読するためには2^{128}回の暗号化処理が必要となる。

また、MISTYは差分解読法と線形解読法に対する安全性を考慮して設計されており、平均差分確率および平均線形確率が2^{-56}以下となることが示されている（松井［1996］）。

なお、MISTYそのものではないものの、MISTYのアルゴリズムの一部に対する攻撃方法が提案されている。まず、FL関数のない４段のMISTY1に対して、36個の選択平文を利用した高階差分攻撃によって解読が可能になることが示されている（久松・金子［1998］）ほか、５段のMISTYに対しては、1,708個の選択平文を用いた高階差分攻撃によって2^{17}の計算量で解読可能であるとの研究成果が発表されている（田中・久松・金子［1998］）。また、FO関数３段で構成されるMISTYのアルゴリズムを解読するためには、2^{28}個の選択平文が必要であることが証明されているが、FO関数の入出力データを統計的に分析することによって、選択平文222個で解読に成功したとの実験結果も示されている（角尾・山田［1998］）。

表2　既存の攻撃法に対する安全性評価結果

暗号方式	鍵長（bit）	ブロック長（bit）	全数探索法（必要計算量）	差分解読法（解読に必要な①選択平文数と②計算量、もしくは評価指標）	線形解読法（解読に必要な①既知平文数と②計算量、もしくは評価指標）	高階差分攻撃	分割攻撃	弱鍵、関連鍵攻撃等その他の攻撃法
DES	56	64	2^56	・16段：①2^47 ②2^37	・16段：①2^43 ②2^42	—	13段 ①1.34×2^40	4個の弱鍵、6組の準弱鍵が見つかっている
FEAL-N（N:段数）	64	64	2^64	・32段：①2^66 ②2^64以上 ・16段：①2^29	・16段：①2^39	—	—	—
LOKI91	64	64	2^64	・13段：①2^64以上	・14段：①2^86	—	—	関連鍵攻撃 ①選択平文数：—— ②計算量：1.375×2^61
IDEA	64	64	2^128	・3.5段：①2^56 ②2^67 ・3段：①2^29 ②2^44 ・1段の差分特性確率：2^(-18)程度	—	—	—	弱鍵：2^59個存在 ①選択平文数：2^39 ②計算量：2^45
SAFER	64 or 128	64	K,SK-64:2^64 K,SK-128:2^128	・K64・5段： ①2^46、②2^35	—	—	—	K-64への関連鍵攻撃 ①選択平文数：2^47
RC5-w/r/b[3]	可変 8〜2,084	32, 64, or 128	2^64	・12段：①2^44 ・12段の最大差分確率[1]：2^(-55.15)	・6段：①2^57 ・6段の最大線形確率[1]：2^(-49.15)	—	—	—
MISTY1,2	128	64	2^128	・平均差分確率：2^(-56)以下	・平均差分確率：2^(-56)以下	4段[2]：選択平文36	—	—

（注）1. 探索したF関数の差分値をある一定の範囲に限定して計測。
2. FL関数のない４段のMISTY１に対する解読研究。
3. RC5の安全性評価は、RC5-32/r/16（鍵長128 bit、ブロック長64 bitのバージョン）を想定したもの。

Ⅴ．DESの安全性を高めるための方策

1. DESの安全性低下を実証する試み

DESが標準化された当初、DESの安全性の高さは米国政府によって強く印象づけられた。標準化を担当したNBS[12]は、「１秒間に2万5000個の鍵を検証可能なチップを開発できたとしても、並列処理に利用することができるのは高々1000個程度であり、この計算機を利用しても全数探索法によって真の鍵を見つけるためには約91年かかる」との試算を発表した。また、DESの認定に際して安全性評価を担当したNSA[13]は、「DESのアルゴリズムについては、現時点（1976年）で知り得る限り、いかなる統計的・解析的弱点も見つからなかった」との分析結果を発表した。こうした安全性評価によって、DESは米国政府標準暗号に認定され、金融分野を中心として幅広い分野において利用されるようになった。

一方、1977年に、DiffieとHellmanは、NBSやNSAとは異なる分析結果を発表した（Diffie and Hellman［1977］）。その内容は、「１秒間に100万個の鍵を検証可能なチップを設計し、このチップを100万個用いて並列処理を行えば１日で候補となるすべての鍵を探索することができる。こうした解読装置を構築するための費用は、現時点で約2000万ドルとなるが、今後の技術進歩によって、近い将来現実的な費用で解読装置を製造できるようになると考えられる」というものである。解読装置の製造コスト逓減は、近年のコンピューターのコスト・パフォーマンス向上に伴って現実のものとなり、Brute Force Methodに対するDESの安全性低下が多くの暗号研究者によって指摘されてきた。たとえば、1993年には、約100万ドルの予算によって平均3.5時間で解読できる専用装置を開発可能であることがWienerによって示されている（Wiener［1993］）ほか、1996年には、Blazeらによって、約30万ドルの予算によって平均３時間で解読に成功する専用装置を開発できることが示されている（Blaze et al.［1996］）。さらに、KusudaとMatsumotoは、タイムメモリートレードオフ法を利用して、１時間以内に50%の確率で真の鍵を発見することが可能な解読装置を、現実的な費用で製作することが可能であることを示している（Kusuda and Matsumoto［1996］）。

こうした中、RSA社は、全数探索法に対するDESの安全性を定量的に検証することを目的に、1997年から1999年にかけて４回の懸賞金付きDES解読コンテストを実施している（表３参照）[14]。これらの解読コンテストの内容は、CBCモードのDESによって変換された暗号文と初期値がRSA社によって公表された後、暗号化に利用

12 NBS（National Bureau of Standard）：米国内における科学技術全般の標準規格の策定を担当する米商務省の下部組織であったが、1988年に名称変更され、現在のNIST（National Institute of Standards and Technology）となっている。NBSは、米国政府内で利用する情報通信技術に関する標準規格FIPS（Federal Information Processing Standard）の認定も担当しており、DES等の暗号アルゴリズムの認定を行っていた。

13 NSA（National Security Agency）：米国防総省の下部組織であり、国家安全保障の観点から国内外で通信情報の諜報活動を行っている。暗号技術についても高度な技術力を有しているとされており、FIPS等の標準策定の際にはNISTに対して助言を行っている。

14 RSA社の関連サイトのURLは、http://www.rsa.com/rsalabs/des3/。

表3 全数探索法によるDESの解読時間と費用（試算と実証）

	DES認定当時の安全性評価		RSA社・DES解読コンテスト			
	NBSの主張（試算）	Diffie-Hellmanの主張（試算）	第1回（解読結果）	第2回（解読結果）	第3回（解読結果）	第4回（解読結果）
試算発表時期（コンテスト開始日）	1976年	1977年1月	1997年1月28日	1998年1月13日	1998年7月13日	1999年1月18日
解読方法	毎秒2万5000個の鍵を検証可能な専用チップを1000個用いて並列処理できる装置を想定	毎秒100万個の鍵を検証可能な専用チップを100万個用いて並列処理できる装置を想定	インターネットを利用して平均約1万台のパソコンを動員	インターネットを利用して平均約4万台[16]のパソコンを動員	毎秒約5900万個の鍵を検証可能な専用チップを約1500個用いて並列処理できる専用装置を利用	インターネットを利用して平均約10万台のパソコンを動員したほか、EFFの解読装置を利用
解読時間（鍵全体に対する鍵検証個数の割合）	約91年（100%）	約20時間（100%）	約140日（約25%）	約40日（約88%）	約56時間（約25%）	約22時間（約22%）
1秒当たりの平均鍵検証個数	約2500万個（1.5×2^{24}）	約1兆個（1.8×2^{39}）	約17億個（1.6×2^{30}）	約184億個（1.1×2^{34}）	約888億個（1.3×2^{36}）	約2000億個（2.9×2^{37}）
解読に必要な費用	—	約2000万ドル	—	—	約25万ドル	—

された鍵をどれだけ早く見つけることができるかを競うというものである。候補となる鍵を用いて暗号文を復号し、与えられた平文が得られた場合、その鍵が真の鍵であることがわかる[15]。賞金は、第１回のコンテストでは１万ドルとなっていたほか、残りのコンテストでは、それぞれ①前回の解読時間の1/4未満で解読できた場合には１万ドル、②前回の解読時間の1/4以上1/2未満の場合には5000ドル、③前回の解読時間の1/2以上3/4未満の場合には1000ドル、となっていた。

第１回および第２回の解読コンテストでは、インターネットによってボランティアを募り、大量のコンピューターを並列的に利用するという方法によって、それぞれ約140日、約40日で解読に成功している。さらに、第３回の解読コンテストにおいては、３日足らずの間にDESを解読可能な専用装置を約25万ドルで構築できることが実証された（EFF［1998］）。この装置を研究・開発したのは、コンピューター・ネットワークにおける情報セキュリティやプライバシー確保等を目的とする米国の非営利団体EFF（Electronic Frontier Foundation）の研究チームであり、有力な暗号研

15 RSA社は平文の一部を予め公表している。平文は、“The unknown message is:” (24 byte) というメッセージで始まり、RSA社が考案した秘密のメッセージがこれに続く形となっている。したがって、平文の最初の３ブロック（１ブロックは8 byte）が明らかとなっていることから、候補となる鍵のひとつで暗号文を復号したときに対応する平文が得られた場合、その鍵が真の鍵である可能性が極めて高い。

16 これは、利用されたすべてのパソコンのCPUがIntel Pentium 166 MHzと仮定したときの台数である。

究者や技術者約10名が中心となっている。EFFは、「DESによって通信データの安全性を確保することはもはや不可能であることを実証するために、DES Crackerの研究・開発を進めてきた」と発表しており、EFFのホームページには、“EFF Builds DES Cracker that proves that Data Encryption Standard is insecure.”と大きく記載されている[17]。なお、EFFは、この解読装置に関する技術的詳細を本として刊行しており、こうした技術情報は容易に入手可能となってきている（EFF［1998］）。さらに、1999年1月に実施された第４回のコンテストでは、分散環境における情報処理技術を研究する米国非営利団体Distributed.netが、EFFが開発した解読装置のほかに、インターネットを使って約10万台のパソコンを動員し、約22時間で解読に成功している。

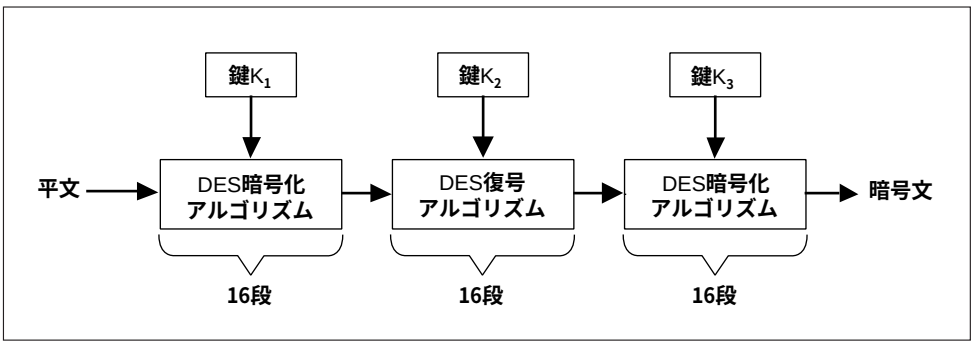
このように、一連のコンテスト実施によって、全数探索法に対するDESの安全性が急速に低下していることが実証されるとともに、DESに代わる暗号方式の必要性がより一層印象づけられた。

2. Triple DESの提案

（1）アルゴリズムの概要と構造

RSA社のDES解読コンテストの結果に象徴されるDESの安全性低下の原因は、DESの鍵長が56 bitと比較的短いことである。一方、DESのアルゴリズム自体は、これまでの研究成果によって、致命的な欠陥がなく優れた構造を有しているとの見方が一般的となっている。このため、DESのアルゴリズムを利用しながら全数探索法に対する安全性を高める方法として、1979年にIBMのTuchmanによってDESの組み合わせ暗号Triple DESが提案された（Tuchman［1979］）[18]。

図19　Triple DESのアルゴリズム



17　EFFの関連サイトのURLは、http://www.eff.org/descracker.html。

18　組み合わせ暗号は、ある特定のアルゴリズムを複数回繰り返して暗号化する方式であり、繰り返しの回数だけ異なる鍵を利用することができるため、鍵長を長くして全数探索法に対する安全性を高めることができるとみられている。

Triple DESは、異なる２つないしは３つの56 bitの鍵を使って、DESのアルゴリズムを３回繰り返して暗号化・復号する方式である。Triple DESの変換手順は、「DESの暗号化アルゴリズム→復号アルゴリズム→暗号化アルゴリズム」という３段階の変換によって構成されるケースが一般的であり[19]、それぞれ異なる３つの鍵を利用する場合には3-key Triple DES、異なる２つの鍵を利用する場合（K_1=K_3の場合）には2-key Triple DESと呼ばれている（図19参照）。Triple DESは、３つの鍵を同一にすると通常のDESと同じ暗号化・復号処理となることから、通常のDESとの互換性を確保できるという特徴を有している。

Triple DESは、DESと比較すると、段数が３倍（48段）となるほか、鍵長は2-key Triple DESの場合には２倍（112 bit）となり、3-key Triple DESの場合は３倍（168 bit）に拡張される。ブロック長は、DESと同じく64 bitである。

（2）Triple DESの標準化

現在、DESに代わる暗号アルゴリズムとして、金融分野を中心にTriple DESの標準化が進められている。米国金融業界における標準規格を策定しているANSI X9は、1998年10月、金融業務に利用される暗号アルゴリズムとしてTriple DESの標準化を完了した（ANSI X9.52）。また、NISTは、1999年1月、Triple DESをDESの後継の米国政府標準暗号（FIPS 46-3）として認定することを発表した[20]。NISTは、現在DESを利用している米国政府機関に対し、DESからTriple DESへの移行に関して次のような対応を要請している。

①現在DESを利用しているシステムにおいては、リスクに見合ったセキュリティ水準を確保できるように、Triple DESへの移行計画を慎重に策定すること。
②今後新しく暗号を利用するシステムを構築する場合、機密ではないが取り扱いに注意を要する情報（sensitive, unclassified data）を暗号化する手段として、Triple DESを採用すること。

これにより、現在DESを採用している米国政府機関はTriple DESへの移行に本格的に取り掛かることになるが、米国政府機関だけでなく、一般のDESユーザーの間でもTriple DESに移行する動きがさらに拡大するものとみられる。

また、ISO/TC68においては、現在DESの使用を前提としている金融分野における国際標準を、今後は「アルゴリズムを特定しない標準」に改訂する方針としており、現在改訂作業が進められている。

19「暗号化→復号→暗号化」という組み合わせ暗号のアルゴリズム構成は、銀行業務における鍵管理（ホールセール業務）の方法に関する標準規格ISO 8732（Banking - Key management（wholesale））において、管理対象の鍵や初期値の暗号化・復号方法のひとつとして記載されており、そのAnnex Cにおいては、例としてTriple DESを利用する方法について記載されている。

20 Federal Register（米国官報）1999年1月15日第64巻第10号pp. 2625-2628

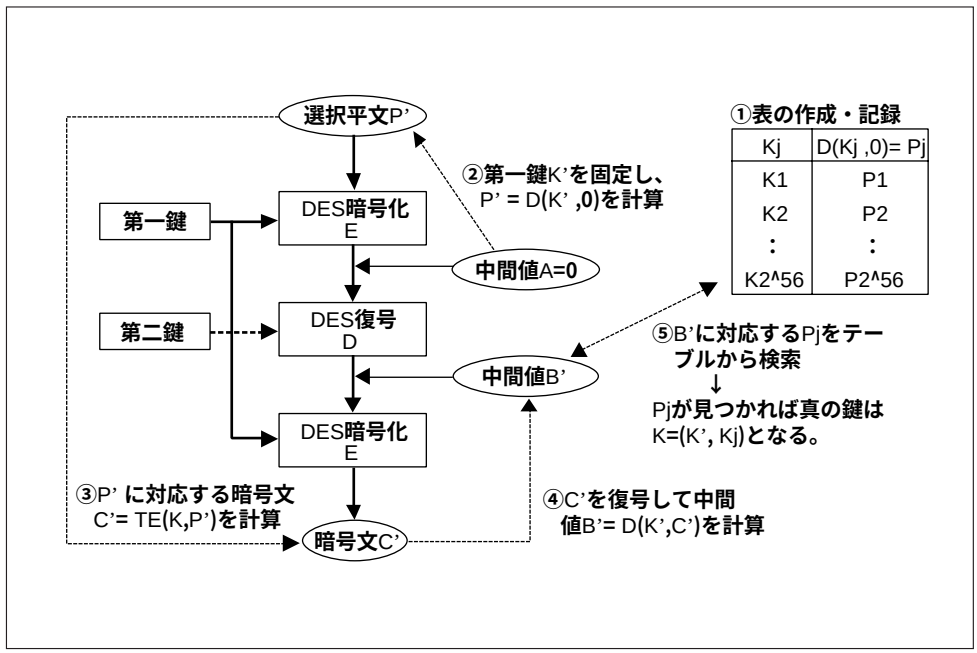
（3）Brute Force Methodに対する安全性

①Merkle-Hellman選択平文攻撃

Triple DESは、DESの鍵長を拡張する効果を有している。しかし、Triple DESは組み合わせ暗号であるため、選択平文攻撃によって実質的な安全性は鍵長が拡大するほど向上しないことが、MerkleとHellmanによって示されている（この解読法はMerkle-Hellman選択平文攻撃と呼ばれている。Merkle and Hellman［1981］）。

Merkle-Hellman選択平文攻撃を2-key Triple DESに適用する場合、以下のような手順で攻撃が行われる（図20参照）。

図20　Merkle-Hellman選択平文攻撃の概要



（i）最初のDES暗号化ステップと次のDES復号ステップとの間の値を中間値Aとし、まずAを0とおく。

（ii）最初と最後のDES暗号化に利用する第一鍵Kj（$j = 1,...,2^{56}$）を用いて、DESの復号アルゴリズムD（暗号化アルゴリズムはE）によって0を変換し、その計算結果$Pj = D(Kj ,0)$（ただし$j = 1,...,2^{56}$）を表として外部記憶媒体に記録する。したがって、必要となる計算量は2^{56}回のDES復号に等しくなる。

（iii）第一鍵をある値K'に設定し、この第一鍵に対して中間値Aが0となるような平文P'を得るために、$P' = D(K' ,0)$を計算する。

（iv）設定した選択平文P'に対する暗号文$C' = TE(K, P')$を入手する（Triple DESの暗号化アルゴリズムをTE、真の鍵を$K = (K_1,K_2)$とする）。

（v）予め設定している第一鍵K'を用いて暗号文C'をDESのアルゴリズムで復号し、中間値$B' = D(K', C')$を計算する。

（vi）ここで（ii）で作成した表から、Pj = B'を満足するPjが存在するか否かをチェックする。このようなPjが存在する場合、真の鍵はK=(K' ,Kj)であることがわかる。一方、Pjが存在しない場合、第一鍵K'を別の鍵に取り替えて、(iii)～(vi)の手続きを繰り返す。

この攻撃では、まず2^{56}回のDESの復号処理によって表を作成し、外部記憶媒体に記録した上で、2×2^{56}回のDESの復号処理（鍵の候補は2^{56}個存在し、(iii)と(v)においてDESの復号が必要となる）によって真の鍵を見つけることができる。したがって、必要な外部記憶媒体を用意することができれば、必要な計算量は3×2^{56}となり、全数探索法(2^{112}回の暗号化処理)に比べて大幅に計算量を削減することができる。

3-key Triple DESについても、同様の攻撃法により、表作成に必要な2^{56}回のDES復号処理に加えて、2^{112}回（上記例の(iii)と(v)の復号において、それぞれ独立に2^{56}回の復号処理が必要となる）のDESの復号処理によって解読が可能となる。したがって、必要な記憶媒体を用意できれば$2^{112}+2^{56}$の計算量によって解読することができ、全数探索法の場合（2^{168}回の暗号化処理）に比べて大幅に計算量を削減できる。なお、Lucksは、Merkle-Hellman選択平文攻撃における処理回数を削減する方法を提案しており、2^{112}回の復号処理を最大で$2^{108.2}$回まで削減できるとしている(Lucks［1998a］)。

Merkle-Hellman選択平文攻撃は、Triple DESに限らず、あらゆる組み合わせ暗号に対して有効であることから、組み合わせ暗号によって鍵長を長くしてBrute Force Methodに対する安全性を向上させる試みには、一定の限界が存在するといえる。このことから、より高い安全性を確保するためには、組み合わせ暗号ではなく、より長い鍵長を有する単体の暗号アルゴリズムの利用が望まれる。

また、OorschotとWienerは、Merkle-Hellman選択平文攻撃をベースに2-key Triple DESに対する既知平文攻撃を提案しており（Oorschot and Wiener［1990］)、N個の既知平文・暗号文ペアと$2^{120-\log_2 N}$回のDESによる暗号化処理によって、2-key Triple DESを解読することが可能であることを示している。

②暗号文一致攻撃と辞書攻撃

Triple DESのブロック長は、DESと同様に64 bitと比較的短い。したがって、暗号文一致攻撃や辞書攻撃に対する安全性は、DESと比較して全く向上していない。

暗号文一致攻撃は、バースデー・パラドックスを利用して一致する暗号文を検索し、平文に関する情報を入手するという方法である。Triple DESのブロック長は64 bitであるから、2^{32}個の暗号文を入手できれば約0.5の確率で同一の暗号文を見つけることが可能となり、それらの暗号文に対応する平文に関する情報を入手することが可能となる。また、辞書攻撃は、ある固定された鍵によって暗号化された暗号文と平文のペアを集めて「辞書」を作っておき、その後入手した暗号文に対応する平文をこの辞書を利用して見つけるという攻撃法である。したがって、ブロック長が短いほど辞書攻撃は有効となる。

こうした攻撃に対して安全性を向上させるためには、ブロック長を長くする必要がある。DESやTriple DESに限らず、FEAL、LOKI91、IDEA、SAFER、MISTY等主要なブロック暗号のブロック長は64 bitとなっており、今後より長いブロック長を有する方式の利用が望ましい。

③Triple DESの利用モードに対する攻撃

ブロック長を変更しないで、Triple DESの暗号文一致攻撃や辞書攻撃に対する安全性を高める方法として、前述の４つの利用モードのほかに、これまで様々な利用モードが提案されている[21]。特殊な利用モードを適用することによって、変換手順をより複雑にして暗号文と平文の単純な関係を絶ち切り、たとえ一致する暗号文ブロックが見つかったとしても、容易に平文に関する情報が漏れないようにすることが可能となる。Triple DESの利用モードとして提案されている方式は、(i) 組み合わせ暗号の構造を生かして、変換の途中で発生する中間値を次のブロックの変換にフィードバックして利用する方式（CBCMモード等）と、(ii) Triple DESをひとつのアルゴリズムとみなし、中間値をデータの攪拌に一切利用しない方式（TCBC-IモードやTCFB-Pモード等）に大別することができる。

しかし、(i) の中間値を変換に利用するモードでは、その中間値を操作することによって、通常のTriple DESよりも大幅に少ない計算量によって解読が可能となるとする分析結果が発表されている。たとえば、3-key Triple DESをCBCMモードにおいて利用する場合、2^{65}個の選択平文を入手できれば、2^{58}回のDESの暗号化処理によって解読可能であることが示されている[22]（Biham and Knudsen［1998］）。これらの分析結果から、Triple DESを利用する場合、その中間値を変換に利用するモードは、安全性の観点から使用しない方がよいとの研究成果が発表されている（Biham［1998］、谷口・太田・大久保［1999］）。

(4) Short Cut Methodに対する安全性

Triple DESは、代表的なShort Cut Methodである差分解読法や線形解読法に対して十分な安全性を有していることが示されている。Triple DESを48段のDESとみなした場合、現在知られている差分特性確率と線形特性確率から計算される攻撃に必要な平文数が膨大となるため、ブロック長64 bitの下で理論的に作成可能な2^{64}個すべての平文・暗号文ペアを利用したとしても、効率的に鍵の候補を絞り込むことができないと考えられている。

21 Triple DESの利用モードを規定しているANSI X9.52には、Triple DESのECB、CBC、OFB、CFBの４つの利用モード（それぞれTECB、TCBC、TOFB、TCFBモードと呼ばれている）に加えて、TCBC-I、TCFB-P、TOFB-Iという３つの利用モードが記載されている。これらの利用モードは、いずれもTriple DESをひとつのアルゴリズムとして利用するものであり、変換途中の中間値を次のブロックの変換に利用したりフィードバックしたりする方式ではない。

22 ANSI X9.52には、CBCMモードは記載されていない。

また、関連鍵攻撃によって、3-key Triple DESがMerkle-Hellman選択平文攻撃よりも少ない計算量によって解読可能であるとの研究成果が発表されている（Kelsey, Schneier and Wagner［1996］）。具体的には、１組の選択平文・暗号文ペアとある特定の関係を有する１組の鍵ペアを利用することによって、2^{56}～2^{72}回程度のDESの暗号化処理で解読できることが示されている。もっとも、この攻撃法は、2-key Triple DESには適用できないほか、攻撃可能な環境は極めて限定されているため、実際の脅威となるとの見方は少ない。

以上のTriple DESの安全性評価に関する分析結果をまとめると、以下の表４の通り。

表４　Triple DESの主要な安全性評価結果（解読に必要な計算量）

	2-key Triple DES	3-key Triple DES
全数探索法	2^112	2^168
Merkle-Hellman 選択平文攻撃	2^57 （選択平文数 2^56）	2^112 （選択平文数 2^56）
Lucksによる攻撃	—	2^108.2
Oorshot-Wiener 既知平文攻撃	$2^{(120-\log_2 N)}$ （既知平文数 N）	—
CBCMモードにおける 選択平文攻撃*	—	2^58 （選択平文数 2^65）
差分解読法	（選択平文数 2^174）	
線形解読法	（既知平文数 2^118）	
関連鍵攻撃*	—	2^56～2^72 （選択平文 1） （選択鍵ペア 1）

＊ CBCMモードにおける選択平文攻撃や関連鍵攻撃は、攻撃が成立する条件が非常に限定されていることから、実際の脅威にはならないとみられている。

Ⅵ．次世代の米国政府標準暗号AES

AES（Advanced Encryption Standard）は、現在米国政府が策定を進めている次世代の米国政府標準暗号の名称である。NISTは、これまで標準暗号として認定してきたDESの安全性低下が深刻化していることから、1997年1月にAESのアルゴリズムを公募によって選定する方針を発表し、同年9月にはアルゴリズムの要件や評価基準等を公表している[23]。

AESのアルゴリズム選定プロセスの特徴は、候補となるアルゴリズムの設計基準や仕様のほか、NISTのアルゴリズムの評価尺度等関連情報をすべて公開するとともに、外部の暗号学者や研究者の分析・評価を参考にしてアルゴリズムの絞り込みを行うことによって、選定プロセスの透明性を高めている点である。NISTは、インターネット上にAES関連のサイトを開設しており[24]、各アルゴリズムの詳細な仕様に関する情報や、暗号研究者による各アルゴリズムの分析結果に関する情報を入手することが可能となっている。

1. AESの候補アルゴリズムの要件、評価基準、標準化スケジュール

（1）アルゴリズムの要件

AESの候補アルゴリズムの要件[25]は、①共通鍵ブロック暗号であること、②鍵長は128 bit、192 bit、256 bitのいずれも利用可能であること、③ブロック長は128 bitが利用可能であること、④ロイヤリティ・フリーで使用できること、の４点である。

このように、鍵長とブロック長をより長くすることによって、全数探索法、暗号文一致攻撃、辞書攻撃に対して十分な安全性を確保できるように配慮されている。

（2）アルゴリズムの評価基準

アルゴリズムの評価基準は、①安全性（解読の困難性）、②コスト、③その他のアルゴリズムの特徴、の３点であり、この順序で優先順位が付けられている。NISTは、これらの基準に関する評価を行うにあたり、外部の暗号研究者や技術者による独自の分析結果の発表を参考にするとしている。

①安全性

安全性は最も重要な評価基準とされており、主に以下の３点について分析される。ただし、最終的な各アルゴリズムの安全性に関する評価は、NIST自身による各アルゴリズムの分析結果だけではなく、外部の暗号研究者の分析結果も十分参考にして決定されることになっている。

23 NISTの発表内容については、宇根［1997］、NIST［1997］を参照。
24 NISTのAES関連サイトのURLは、http://www.nist.gov/aes。
25 アルゴリズムの要件、評価基準、標準化スケジュールは、NISTの公表資料（NIST［1997］, Smid and Dworkin［1998］）から引用している。

（i） 暗号文のランダム性[26]
（ii）安全性に関する理論的根拠
（iii）評価プロセスにおいて指摘された安全性に関する問題点

②コスト

コストについては、主に以下の２点について測定・評価される。

（i） 鍵のセットアップ、暗号化、復号等の処理速度
（ii）実装に必要となるメモリー容量

コストに関する評価は、２回の技術的評価ラウンドに分けて行われる。技術的評価第１ラウンドでは、NISTは、ソフトウェアによる実装（鍵長・ブロック長ともに128 bitに設定）での処理速度等を測定・評価する[27]。技術的評価第２ラウンドでは、NISTは、ソフトウェアによる実装（上記以外の鍵長・ブロック長の組み合わせ）での処理速度等を測定・評価するほか、8 bit プロセッサーやハードウェアによる実装についても、各アルゴリズムの提案者および外部の技術者から寄せられた測定結果を参考に評価を行う。

③その他のアルゴリズムの特徴

その他の評価基準として、（i）様々なアプリケーションへの利用可能性、（ii）ハードウェアやソフトウェアへの適用性、（ii）構造の単純性等が挙げられている。様々なアプリケーションへの利用可能性については、たとえば、メッセージ認証コード（MAC）、疑似乱数生成装置やハッシュ関数等に利用可能かどうか等について評価される。また、ハードウェアやソフトウェアへの適用性に関しては、そのアルゴリズムがハードウェアとソフトウェアのどちらで実装するのに向いているか等が評価されることとなっている。

（3）標準化スケジュール

アルゴリズムの標準化スケジュールは、以下の表５の通り。現在、NISTや暗号研究者によって各アルゴリズムの分析・評価が進められており、早ければ2000年内にも、AESのアルゴリズムが決定・発表される見込みである。

26 具体的には、ある平文データに対応する暗号文データと、その平文データの bitをランダムに転置変換して得られるデータを生成し、両者の間の類似性を分析することにより、そのアルゴリズムによって生成される暗号文にどの程度偏りが生じているかを評価するもの。

27 NISTが採用するソフトウェアの実装環境は、CPU：Pentium Pro 200 MHz、メモリー：64 MB RAM、OS：Windows95、PC機種：IBM PC互換機、プログラム言語：Borland C++ 5.0およびJAVA JDKとなっている。

表５　アルゴリズムの標準化スケジュール

フェーズ	日程	作業内容等
アルゴリズムの募集	1998/6/15 最終締切	NISTは、提出資料に不備がないかどうかをチェックし、候補となるアルゴリズムを決定する。
技術的評価第1ラウンド	1998/8/20 ～1999/4/15	NISTは、暗号研究者から寄せられたアルゴリズムの分析結果を参考にして、安全性や処理速度等について分析・評価を行う。
第1回 AES候補コンファレンス	1998/8/20 ～ 8/22	各アルゴリズム提案者が自分のアルゴリズムの説明を行い、参加者からコメントを得る。NISTは、各アルゴリズムの詳細に関する情報を各参加者に提供し、アルゴリズムの分析・評価を依頼する。
第2回 AES候補コンファレンス	1999/3/22 ～ 3/23	技術的評価第1ラウンドにおけるアルゴリズムの分析結果について議論するとともに、アルゴリズムの絞り込みを行う際の留意点等について議論する。
第2ラウンド候補の発表	1999年秋頃	NISTは、候補のアルゴリズムを5つ程度に絞り込み、発表する。
技術的評価第2ラウンド	1999年から2000年にかけて6～9か月間	NISTは、絞り込んだアルゴリズムについて、暗号研究者から寄せられた分析結果を参考に、より詳細に分析・評価を行う。
第3回 AES候補コンファレンス	1999年末 ～2000年初	技術的評価第2ラウンドにおける分析結果について議論するとともに、アルゴリズムの一層の絞り込みの方法等について検討する。
アルゴリズムの最終選定	発表時期不明	NISTは、第3回コンファレンスの結果を参考にして候補アルゴリズムをひとつに絞り込み、発表する。

2. 候補アルゴリズムの概要

NISTは、1998年8月に開催された第１回AES候補コンファレンスにおいて、「全体で21のアルゴリズムが提案されたものの、提出されたアルゴリズムのプログラムが正常に稼動するか否か、提出書類に不備がないか否か等について審査した結果、15のアルゴリズムが事前審査にパスした」旨を発表し、各アルゴリズムの仕様や関連論文等を公表した（表６参照）。15のアルゴリズムのうち、５件が米国から提案されているほか、カナダ（２件）、韓国、フランス、日本、コスタリカ、オーストラリア、ドイツ、ベルギー、イギリスといった国々から提案されている。

―― 各アルゴリズムの詳細については、巻末の５つの表を参照。各表の内容は以下の通り。本節および以下の表は、第１回AES候補コンファレンスのプロシーディングス（NIST［1998b］）や各提案者が発表した資料[28]に基づいて作成されている。

28 各提案者が発表した資料については、NISTのホームページ（http://www.nist.gov/aes）からダウンロードすることが可能となっている。

表７：各アルゴリズムの設計方針
表８：各アルゴリズムの構造上の特徴
表９：アルゴリズムの構造に関する概念整理（Feistel構造、SPN構造、2-round SPN構造等、ラウンド関数やF関数の構造について整理）
表10：各アルゴリズムの安全性に関する分析結果
表11：各アルゴリズムの処理速度に関する分析結果

表６　AESの候補として提案されているアルゴリズム

名称	国・代表提案者（会社名等）
CAST-256（キャスト256）	カナダ・Entrust Technologies社
CRYPTON（クリプトン）	韓国・Future Systems社
DEAL（ディール）	カナダ・Outerbridge
DFC（ディ・エフ・シー）	フランス・Centre National pour la Recherche Scientifique
E2（イーツー）	日本・NTT社
FROG（フロッグ）	コスタリカ・TecApro Internacional社
HPC（エイチ・ピー・シー）	米国・Schroeppel（アリゾナ大学）
LOKI97（ロキ97）	オーストラリア・Brown（Australian Defense Force Academy）
MAGENTA（マジェンタ）	ドイツ・Deutsche Telekom社
MARS（マーズ）	米国・IBM社
RC6（アール・シー6）	米国・RSA Laboratories社
RIJNDAEL（ラインデール）	ベルギー・Daemen（Banksys社）
SAFER+（セイファー・プラス）	米国・Cylink社
SERPENT（サーペント）	イギリス、イスラエル、ノルウェー・Anderson（ケンブリッジ大学）
TWOFISH（トゥーフィッシュ）	米国・Schneier（Counterpane Systems社）

（1）CAST-256

CAST-256はカナダのEntrust Technologies社によって提案されたアルゴリズムであり、ブロック長は128 bit、鍵長は128、192、256 bitが利用可能である（Adams［1998］）。

①設計方針

CAST-256の主な設計方針は、同社によって既に発表されているCAST-128（ブロック長64 bit、鍵長128 bit）のアルゴリズムをベースにして、安全性と処理速度を向上させる、というものである。

②アルゴリズムの構造

128 bitのデータブロックは４つの32 bitサブブロックに分割された後、48段のラウンド関数によって変換される。ラウンド関数は一般形Feistel構造と呼ばれる構造を有しており、３種類のF関数によって構成されている。各F関数はCAST-128で利用

されているものが採用されている。F関数は、bitシフト、排他的論理和、2^{32}を法とする加算、引算のほか、４種類のS-box（8 bit入力、32 bit出力）によって構成されている。

③安全性と処理速度

安全性に関しては、提案者は、「40段の最大差分特性確率が2^{-140}以下であるほか、48段の最大線形特性確率が2^{-122}であることから、差分・線形解読法に対して十分な安全性を有している。また、高階差分攻撃や関連鍵攻撃等、他の攻撃法に対しても安全であるとみられる」との分析結果を発表している。

暗号化の処理速度については、NIST指定のC言語によるソフトウェア実装において約14 Mbps、アセンブリ言語での実装において約47 Mbpsと試算されている。

（2）CRYPTON

CRYPTONは、韓国のFuture Systems社によって提案されたアルゴリズムであり、ブロック長は128 bit、鍵長は可変（64～256 bitまで32 bit毎に利用可能）である（Lim［1998］）。

①設計方針

CRYPTONの主な設計方針は、(i) 差分解読法や線形解読法等、既存の攻撃法に対して十分な安全性を確保する、(ii) 非線形変換を並列処理できる構造を採用し、ハードウェア・ソフトウェア両方において高速処理を可能にする、(iii) 安全性評価を容易にするためにシンプルな構造とする、の３点である。

②アルゴリズムの特徴

CRYPTONのデータランダム化部は、SPN構造を有するラウンド関数12段で構成されており、128 bitデータブロックは４つの32 bitサブブロックに分割され、各々が並列に処理されるように構成されている。ラウンド関数では、換字変換、 bit単位およびbyte単位での転置変換、32 bit拡大鍵との排他的論理和が採用されている。鍵スケジューリング部では、鍵から32 bit拡大鍵が52個生成される。

③安全性と処理速度

安全性に関する提案者の評価では、８段のラウンド関数における最大差分・線形特性確率はそれぞれ2^{-160}、2^{-128}程度となることから、差分・線形解読法に対して十分な安全性を確保しているとされている。また、４段のラウンド関数の出力をブール多項式で表示すると次数が128以上となることから、高階差分攻撃に対して十分な安全性を有していると分析されている。

暗号化の処理速度については、Pentium Pro 200 MHz、32 MB RAM、Windows95、Microsoft Visual C 5.0での実装において約51 Mbps、アセンブリ言語での実装において約62 Mbpsとの測定結果が発表されている。

（3）DEAL

DEAL（Data Encryption Algorithm with Larger Blocks）は、カナダのOuterbridgeとノルウェー・ベルゲン大学のKnudsenによって提案されたアルゴリズムであり、ブロック長128 bit、鍵長128、192、256 bitが利用可能である（Knudsen［1998］）。

①設計方針

DEALの主な設計方針は、（i）DESのアルゴリズムをF関数に利用することで、これまでに提案されている解読法に対して安全性を確保する、（ii）Triple DESと同程度の処理速度で実装可能にする、（iii）既にDESが実装されている環境において、DESからの移行を容易にする、である。

②アルゴリズムの構造

データランダム化部では、128 bitデータブロックは２つの64 bitサブブロックに分割された後、Feistel構造を有しているラウンド関数によって変換される。段数は６段以上が推奨されている。ラウンド関数内のF関数にはDESのアルゴリズムが利用されている点が特徴である。鍵スケジューリング部においてもDESのアルゴリズムが利用されており、鍵は64 bit単位に分割された後、段数に等しい数の64 bit拡大鍵が生成される。

③安全性と処理速度

提案者によって差分解読法に対する安全性について分析がなされており、「ラウンド関数６段で暗号化されたデータを解読するためには、2^{70}個以上の選択平文と2^{121}回のDESの暗号化処理が必要であり、差分解読法に対して安全性を確保している」と発表している。

暗号化・復号の処理速度については、C言語によるソフトウェア実装において、暗号化・復号ともにDESのスピードの約６分の１と試算されている。

（4）DFC

DFC（Decorrelated Fast Cipher）は、フランス・CNRSによって提案されたアルゴリズムであり、ブロック長128 bit、鍵長可変（上限は256 bit）である（Gilbert et al.［1998］, Vaudenay［1998］）。

①設計方針

DFCの主な設計方針は、（i）decorrelation theory[29]を利用することによって、Triple DESよりも高い安全性を確保する、（ii）DESよりも高速での実装を可能とする、（iii）様々なプラットフォームにおいて実装可能にする、である。

29 decorrelation theoryは、差分解読法や線形解読法に対する安全性の証明が可能な暗号アルゴリズムを構築するための新しい技術であり、Vaudenayらによって発表されている（Vaudenay［1998］）。

②アルゴリズムの特徴

DFCのデータランダム化部はFeistel構造を有するラウンド関数８段で構成されており、データブロックは２つの64 bitサブブロックに分割される。F関数には、2^{64}を法とする加算や乗算のほか、非線形変換（6 bit入力、32 bit出力）が利用されており、128 bit拡大鍵がパラメーターとして入力される。鍵スケジューリング部では、128 bit拡大鍵が８個生成される。

③安全性と処理速度

DFCの安全性に関しては、decorrelation theoryによって差分・線形解読法に必要な計算量が見積もられており、差分解読法に対しては2^{110}個以上の選択平文が必要であるほか、線形解読法に対しては2^{92}個以上の既知平文が必要になるとの分析結果が示されている。

暗号化の処理速度については、Pentium Pro 200 MHz、C言語による実装において、約34 Mbpsとの測定結果が発表されている。

（5）E2

E2（Efficient Encryption Algorithm）は、日本のNTT社によって提案されたアルゴリズムであり、ブロック長128 bit、鍵長128、192、256 bitが利用可能である（神田他［1998］、盛合他［1998］、NTT［1998］）。

①設計方針

E2の主な設計方針は、(i) 既存の攻撃法に対する安全性を客観的尺度によって保証する、(ii) 簡素なラウンド関数を利用することで、高速処理を実現する、(iii) あらゆるプラットフォーム上で柔軟な実装を可能にする、である。

②アルゴリズムの構造

E2のデータランダム化部では、拡大鍵による排他的論理和と乗算、byte単位の転置変換によって構成される初期変換部、Feistel構造を有するラウンド関数12段、初期変換部の逆変換となる最終変換部によって構成される。初期変換部による変換後、128 bitデータブロックは２つのサブブロックに分割され、ラウンド関数に入力される。F関数は、「S-boxによる換字変換→線形変換（8 bit単位の排他的論理和を利用）→S-boxによる換字変換」という構造を有している。S-box（8 bit入出力）は差分・線形解読法等いくつかの既存の攻撃法に対して安全性が確保されるように設計されている。鍵スケジューリング部では、鍵から128 bit拡大鍵が16個生成される。

③安全性と処理速度

E2のF関数に含まれるS-boxは、最大差分特性確率、最大線形特性確率等10項目の安全性評価基準を満足するように選択されており、ラウンド関数９段で既存の主な攻撃法に対して十分な安全性が確保されていると分析されている。たとえば、９

段の最大差分特性確率および最大線形特性確率がそれぞれ2^{-140}、2^{-131}程度であることが示されている。また、鍵スケジューリング部では、データランダム化部のF関数の一部や8 bit単位の転置変換が利用されており、鍵スケジューリング部への攻撃が容易に適用できない構造となっていると分析されている。

暗号化・復号の処理速度については、NIST指定のC言語によるソフトウェア実装において、暗号化・復号ともに約36 Mbps、アセンブリ言語での実装においては、暗号化・復号ともに約62 Mbpsとの測定結果が発表されている。

(6) FROG

FROGは、コスタリカのTecApro Internacional社によって提案されたアルゴリズムであり、ブロック長（64～1024 bit)、鍵長（40～1000 bit）とも可変である（Georgoudis, Leroux and Chaves［1998］)。

①設計方針

FROGの主な設計方針は、(i) 簡素な変換を利用することで、様々なアプリケーションにおいて高速処理を実現する、(ii) 毎回異なる変換手段を利用することによって安全性を確保する、(iii) 様々なブロック長と鍵長の組み合わせを利用可能にする、(iv) 様々なプラットフォームでの実装を可能とする、である。

②アルゴリズムの構造

FROGでは、まず鍵スケジューリング部において、鍵から３種類の変換表（大きさはブロック長に依存）が８組生成される。データランダム化部はラウンド関数８段によって構成されており、３種類の変換表が１組ずつ各ラウンド関数の変換に利用される。ラウンド関数におけるデータブロックの変換はbyte単位での排他的論理和と換字変換のみとなっており、非常に単純な構造となっているほか、各ラウンド関数における変換方法がすべて異なっている点が特徴である。

③安全性と処理速度

安全性に関して、提案者は、「全数探索法よりも効率的な解読法はないとみられる」と評価している。

暗号化の処理速度については、Pentium 200 MHz、64 MB RAM、Windows95、C言語での実装において、約10 Mbpsとの測定結果が発表されている。

(7) HPC

HPC（Hasty Pudding Cipher）は、米国・アリゾナ大学のSchroeppelによって提案されたアルゴリズムであり、ブロック長・鍵長ともに可変である（Schroeppel［1998a］［1998b］)。

①設計方針

HPCの主な設計方針は、(i) 既存の解読法に対して十分な安全性を確保する、(ii) 64 bitの汎用CPUを用いたソフトウェアでの実装で最高速度での暗号化処理が可能となる構造にする、(iii) 任意のブロック長や鍵長で利用可能とし、様々なアプリケーションに柔軟に対応できるようにする、である。

②アルゴリズムの構造

HPCの特徴は、暗号化・復号の際に、鍵のほかにSPICEと呼ばれる256 bitのデータ（毎回変更される）を利用する点である。SPICEのデータが変更されると暗号文も変化することから、SPICEも鍵の一部といえる。データブロックは64 bit単位のサブブロックに分割され、段数８のラウンド関数では、拡大鍵やSPICEを変数とする排他的論理和、2^{64}を法とする加算や引算、bitシフトによって変換される。鍵スケジューリング部では、鍵から256個の64 bit拡大鍵を含む表が生成され、ラウンド関数に入力される拡大鍵は段数等に依存して決定される。

③安全性と処理速度

安全性について、提案者は、「差分・線形解読法などの既存の解読法は適用困難である」との分析結果を発表している。

暗号化・復号の処理速度については、Pentium 200 MHz、C言語での実装において、暗号化・復号ともに約7 Mbpsと試算されている。

(8) LOKI97

LOKI97は、オーストラリアのAustralian Defense Force AcademyのBrown、ウォロンゴング大学のPieprzykとSeberryによって提案されたアルゴリズムであり、ブロック長128 bit、鍵長128、192、256 bitが利用可能である（Brown and Pieprzyk［1998］）。

①設計方針

LOKI97の主な設計方針は、(i) LOKI89やLOKI91のアルゴリズムをベースにする、(ii) 差分解読法、線形解読法、関連鍵攻撃等既存の攻撃法に対して十分な安全性を確保する、である。

②アルゴリズムの構造

LOKI97のデータランダム化部は16段のラウンド関数によって構成されており、各ラウンド関数にはFeistel構造が採用されている。F関数には2-round SPN構造が利用されており、拡大鍵依存型転置変換や２種類のS-boxによって構成される換字変換等が利用されている。各ラウンド関数の変換には３個の64 bit拡大鍵が必要となるため、鍵スケジューリング部では、データランダム化部のF関数を利用したアルゴリズムによって48個の拡大鍵が生成される。

③安全性と処理速度

LOKI97の安全性について、提案者は「差分特性確率の算出は困難であるものの、ラウンド関数14段の最大線形特性確率は2^{-141}程度と見込まれることから、線形攻撃法に対して十分な安全性を有している」との分析結果を発表している。

暗号化・復号の処理速度については、Pentium II 233 MHz、64 MB RAM、Windows95、Microsoft Visual C++による実装において、暗号化・復号ともに約6 Mbpsとの測定結果が発表されている。

(9) MAGENTA

MAGENTA（Multifunctional Algorithm for General-Purpose Encryption and Network Telecommunication）は、ドイツテレコム社によって提案されたアルゴリズムであり、ブロック長128 bit、鍵長128、192、256 bitが利用可能である（Jacobson and Huber［1998］）。

①設計方針

MAGENTAの主な設計方針は、(i)アバランシュ性（入力データの1 bitが変化した場合、その影響が出力データのすべての bitに及ぶ性質）等、安全性の観点から望ましい性質を有する関数を利用して簡素な構造とする、(ii)ハードウェア・ソフトウェアの両方で高速処理を実現する、である。

②アルゴリズムの構造

MAGENTAのデータランダム化部にはFeistel構造が採用されており、128 bitのデータブロックは２つの64 bitサブブロックに分割された後に変換される。ラウンド関数の段数は、鍵長が128 bitおよび192 bitの場合には６、鍵長が256 bitの場合には８と設定される。F関数（MAGENTAではE関数と呼ばれている）には、線形変換と256を法とするべき乗剰余演算等の非線形変換が利用されている。ラウンド関数１段当たり64 bit拡大鍵が１個必要となるが、鍵スケジューリング部では、鍵を64 bit単位のデータに分割し、それらのデータを拡大鍵としてそのままラウンド関数に入力する仕組みとなっている。

③安全性と処理速度

安全性に関しては、提案者は、「ラウンド関数１段の差分および線形特性確率はそれぞれ2^{-40}、2^{-29}以下になると見込まれることから、差分解読法や線形解読法に対して十分な安全性を有している」と分析している。

暗号化・復号の処理速度については、Pentium Pro 200 MHz、C言語による実装において、暗号化・復号ともに約1 Mbpsとの測定結果が発表されている。

（10）MARS

MARSは、米国のIBM社によって提案されたアルゴリズムであり、ブロック長128 bit、鍵長可変（128～1248 bit）である（Burwick et al.［1998］）。

①設計方針

MARSの主な設計方針は、（i）32 bit汎用CPUを搭載したコンピューターでのソフトウェア実装において最も高速での処理を可能とする、（ii）複数種類のラウンド関数を利用することによって高い安全性を実現する、（iii）安全性に関する分析が容易となるような変換手段を利用してアルゴリズムを構築する、である。

②アルゴリズムの構造

MARSのデータランダム化部は、32段のラウンド関数によって構成されており、128 bitデータブロックは４つの32 bitサブブロックに分割されて変換される。type-3 Feistel構造と呼ばれるラウンド関数は4種類存在し、それぞれ８段ずつ組み込まれている。変換手段としては、差分特性や線形特性を考慮して選択されたS-boxのほか、排他的論理和、加算、乗算、データ依存型および非依存型 bitシフトが利用されている。鍵スケジューリング部には、データランダム化部のラウンド関数の構造が採用されているほか、生成した拡大鍵の bitの値をチェックすることにより、その拡大鍵が弱鍵でないことを検証する仕組みが取り入れられている。40個の32 bit拡大鍵が生成される。

③安全性と処理速度

MARSの安全性について、提案者は、「差分解読法を利用するためには2^{280}個以上の選択平文が必要であるほか、線形解読法を利用するためには2^{128}個以上の既知平文が必要である。また、弱鍵も見つかっていない」との分析結果を発表している。

暗号化・復号速度については、NIST指定のC言語によるソフトウェア実装において、暗号化・復号ともに約28 Mbps、JAVAによる実装において、暗号化が約15 Mbps、復号が約17 Mbpsとの測定結果が発表されている。

（11）RC6

RC6は、米国のRSA Laboratories社によって開発されたアルゴリズムであり、ブロック長、鍵長（上限2040 bit）ともに可変である（Rivest et al.［1998］, RSA［1998］）。

①設計方針

RC６の主な設計方針は、（i）32 bitデータの乗算等32 bit CPUによるソフトウェア実装において高速処理が可能となる演算を組み合わせる、（ii）既存の攻撃法に対して安全性を確保する、（iii）安全性に関する分析が容易になるようにアルゴリズムの構造をシンプルにする、である。

②アルゴリズムの構造

RC6のデータランダム化部では、データブロックは長さの等しい４つのサブブロックに分割された後、ラウンド関数に入力される。ラウンド関数の段数は可変であるが、ブロック長128 bitの場合には20段が推奨されている。ラウンド関数は、データ依存型 bitシフトのほか、2^w（wはサブブロックの長さ）を法とする加算、引算、乗算といった算術演算によって構成されている。鍵スケジューリング部では、ラウンド関数をr段とすると、ブロック長と同じ長さの拡大鍵が（2r+4）個生成される。

③安全性と処理速度

RC6の安全性について提案者は、「18段での最大差分特性確率が2^{-264}程度であり差分解読法に対して安全であるほか、線形解読法についても、少なくとも2^{182}個の既知平文が必要となるため安全である。高階差分攻撃等その他の解読法も適用困難であるとみられる」との分析結果を発表している。

段数20の場合の暗号化・復号速度は、Pentium II 266 MHz、32 MB RAM、Windows95、C言語での実装において、暗号化が約42 Mbps、復号が45 Mbpsとの測定結果が発表されているほか、アセンブリ言語での実装においては、暗号化・復号ともに約101 Mbpsとの測定結果が発表されている。

（12）RIJNDAEL

RIJNDAELは、ベルギー・Banksys社のDaemenとルーベン・カトリック大学のRijmenによって提案されたアルゴリズムであり、ブロック長・鍵長ともに128、192、256 bitが利用可能である（Daemen and Rijmen［1998］）。

①設計方針

RIJNDAELの主な設計方針は、(i) 既存の攻撃法に対して十分な安全性を確保する、(ii) 様々なプラットフォームにおいて実装可能とする、(iii) 安全性に関する分析が容易になるようにアルゴリズムの構造をシンプルにする、である。

②アルゴリズムの構造

RIJNDAELのラウンド関数はSPN構造を有しており、データブロックはラウンド関数内で8 bit単位で変換される。ラウンド関数の段数はブロック長と鍵長に依存し、10、12、14段のいずれかとなる。ラウンド関数は３種類の変換部によって構成されており、線形変換層（bitシフト等）、非線形変換層（換字変換）、拡大鍵変換層（拡大鍵との排他的論理和）という順番で変換が行われる。鍵スケジューリング部では、ブロック長と同じ長さの拡大鍵が（r+1）個（rは段数）生成される。鍵スケジューリング部の変換には、データランダム化部のbitシフトと換字変換が利用される。

③安全性と処理速度

RIJNDAELの安全性に関しては、「８段の最大差分・線形特性確率が、それぞれ2^{-350}、2^{-300}程度となることから、差分・線形解読法に対して安全である。また、補間攻撃や関連鍵攻撃等の攻撃に対しても安全であるとみられる」との分析結果が提案者によって発表されている。

暗号化・復号速度については、Pentium 200 MHz、Linux、C言語での実装において、暗号化・復号ともに約27 Mbps、アセンブリ言語での実装においては、暗号化・復号ともに約80 Mbpsとの測定結果が発表されている。

（13）SAFER+

SAFER+（Secure And Fast Encryption Routine +）は、米国のCylink社によって提案されたアルゴリズムであり、ブロック長は128 bit、鍵長は128、192、256 bitが利用可能である（Massey, Khachatrian and Kuregian［1998］）。

①設計方針

SAFER+の主な設計方針は、(i) SAFER-KやSAFER-SKをベースにして、既存の攻撃法に対して安全性を確保する、(ii) シンプルな構造を採用し、様々な実装環境において高速処理を可能にする、である。

②アルゴリズムの構造

SAFER+のラウンド関数は４種類の変換層によって構成されており、第一および第三層では256を法とする加算と排他的論理和、第二層では指数関数45^X mod 257と対数関数$\log_{45}X$、第四層では法256の乗算が利用されている。データブロックは16個の8 bitサブブロックに分割される。段数は鍵長に依存し、鍵長128、192、256 bitに対してそれぞれ８、12、16段となっている。鍵スケジューリング部では、弱鍵の発生を防止するために乱数による演算が利用されており、128 bitの拡大鍵が(2r+1)個生成される（rは段数）。

③安全性と処理速度

安全性については、提案者から「５段の差分特性確率が高々2^{-128}以下であり、６段で差分解読法に対して十分な安全性を有しているほか、３段で線形解読法に対して十分な安全性を有している。また、弱鍵や関連鍵も見つかっていない」との分析結果が発表されている。

暗号化・復号の処理速度については、Pentium 200 MHz、64 MB RAM、Windows95、C言語での実装において、暗号化・復号ともに約12 Mbpsとの測定結果が発表されている。

（14）SERPENT

SERPENTは、イギリス・ケンブリッジ大学のAnderson、イスラエル・テクニオンのBiham、ノルウェー・ベルゲン大学のKnudsenが提案したアルゴリズムであり、ブロック長が128 bit、鍵長が可変（上限256 bit）である（Anderson, Biham and Knudsen［1998a］［1998b］）。

①設計方針

SERPENTの主な設計方針は、(i) これまで十分な安全性評価が行われている構造を利用することによって、安全性評価を容易にする、(ii) Triple DESよりも高い安全性を確保する、(iii) bitslice implementation[30]によって高速実装を可能にする、である。

②アルゴリズムの構造

SERPENTは、初期・最終転置と、SPN構造を有する32段のラウンド関数によって構成されている。ラウンド関数では、入力されたデータブロックは32個の4 bitサブブロックに分解され、32個の換字変換S-box（4 bit入出力）によって変換された後、排他的論理和やbitシフトによって構成される線形変換が行われる。なお、bitslice implementationにおいては、データブロックは４つの32 bitサブブロックに分割された後、32 bit入出力のS-boxや線形変換によって変換される。鍵スケジューリング部では、33個の128 bit拡大鍵が生成される。

③安全性と処理速度

安全性に関する提案者の分析結果では、「24段の最大差分・線形特性確率が、それぞれ2^{-232}、2^{-109}以下になることから、差分・線形解読法に対して安全であるほか、５段の出力データのブール多項式次数が243程度になることから、高階差分攻撃に対しても安全である」と発表されている。

暗号化・復号の処理速度については、NIST指定のC言語によるソフトウェア実装において、暗号化・復号ともに約15 Mbps、JAVAによる実装において、暗号化・復号ともに約583 Kbpsと試算されている。

（15）TWOFISH

TWOFISHは米国・Counterpane Systems社のSchneier、Kelsey、Hall、Ferguson、Hi/fn社のWhiting、カリフォルニア大学バークレー校のWagnerによって提案された

30 bitslice implementationは、通常平文ブロックあるいはサブブロック単位で暗号化を行うアルゴリズムを、1bit単位の論理変換を使って実装する方法。たとえば、32 bitプロセッサーを利用する場合、通常のソフトウェアによる実装では一度にひとつのデータブロックしか暗号化できないが、bitslice implementationを利用すると、32個のデータブロックを並列的に処理することが可能となる（ただし、各データブロックの処理速度は低下することから、全体的に処理速度が向上するか否かはアルゴリズムの構造等に依存する）。

アルゴリズムであり、ブロック長は128 bit、鍵長は128、192、256 bitが利用可能である（Schneier et al.［1998］）。

①設計方針

TWOFISHの主な設計方針は、(i) これまでに安全性に関する分析が十分行われている変換や構造を利用する、(ii) 様々なプラットフォームにおいて高速処理を可能にする、(iii) 安全性の分析が容易になるようにラウンド関数の構造をシンプルにする、である。

②アルゴリズムの構造

TWOFISHのデータランダム化部は、「拡大鍵との排他的論理和→ラウンド関数16段→拡大鍵との排他的論理和」という構成となっており、128 bitデータブロックは４つの32 bitサブブロックに分割された後にラウンド関数に入力される。ラウンド関数にはFeistel構造が採用されており、F関数は bitシフト、８つのS-box（8 bit入出力）による換字変換、線形変換、2^{32}を法とする拡大鍵との加算によって構成されている。拡大鍵は32 bitであり、鍵スケジューリング部において40個生成される。

③安全性と処理速度

安全性に関しては、提案者は、「ラウンド関数７段に対して差分解読法を適用するためには少なくとも2^{131}個の選択平文が必要であるほか、12段に対して線形解読法を適用するためには少なくとも2^{121}個の既知平文が必要となるため、16段のTWOFISHは差分・線形解読法に対して十分な安全性を有している。また、高階差分攻撃、補間攻撃等その他の既存の攻撃法に対しても安全であるとみられる」と分析している。

暗号化・復号の処理速度については、NIST指定のC言語によるソフトウェア実装において約40 Mbps、アセンブリ言語による実装において約90 Mbpsとの測定結果が発表されている。

3. これまでに発表されているアルゴリズムの分析結果

第１回AES候補コンファレンスにおいて発表された15の候補アルゴリズムのうち、CRYPTON、DEAL、DFC、E2、FROG、LOKI97、MAGENTA、MARS、RIJNDAEL、SAFER+、SERPENTの安全性に関する分析結果が寄せられているほか、NIST指定の実装環境における処理速度の測定がNISTによって実施されており、その結果が公表されている。また、新しい共通鍵ブロック暗号の解読法のアイデアが発表されている。

（1）安全性に関する分析結果

①CRYPTON

６段のCRYPTONは、鍵長128 bitの場合、線形和攻撃[31]によって全数探索法よりも効率的に鍵を検索可能である（青木［1999］）。また、鍵長が256 bitの場合、2^{-224}の確率で弱鍵（解読の手掛りとなるような暗号文を生成する鍵）が存在する[32]。

②DEAL

鍵長192 bit、ラウンド関数６段のDEALは、2^{33}個の選択平文を入手できれば、2^{145}回の暗号化処理によって解読可能。2^{33}個の選択平文を入手するための計算量は非現実的とはいえないほか、2^{145}回の暗号化処理に必要な計算量は、全数探索に必要な計算量2^{192}に比べて大幅に少ない（Lucks［1998b］）。

③DFC

ラウンド関数の出力データが入力データに依存しなくなるような拡大鍵が、2^{-64}の確率で発生。このような場合、DFCの段数（８段）は実質的に１段減少することとなり、その安全性も低下すると考えられる。また、ラウンド関数の入力データと出力データが一致するような拡大鍵が2^{-128}の確率で発生する[33]。

④E2

５段のE2は、鍵長128 bitの場合、線形和攻撃によって全数探索法よりも効率的に鍵を検索可能である（青木［1999］）。また、初期変換と最終変換のない８段のE2においては、byte単位の差分を利用したbytewise differential cryptanalysisによって、2^{97}個の選択平文を用いると８段目の拡大鍵を効率的に検索することが可能となる（時田・松井［1999］）

⑤FROG

ブロック長・鍵長ともに128 bitのFROGは、ある一定の条件を満足する鍵（約2^{95}個存在する、鍵空間2^{128}の2^{33}分の1）によって暗号化された選択平文・暗号文ペアを2^{58}個入手できれば、差分解読法によって全数探索法よりも効率的に解読可能である。また、ある一定の条件を満足する鍵（約2^{96}個存在する、鍵空間2^{128}の2^{32}分の

31 線形和攻撃：補間攻撃を改良した攻撃法のひとつ。補間攻撃では、（n+1）組の平文・暗号文ペアを利用して、平文を変数とするn次多項式の暗号化関数を構成し、導出した暗号化関数を利用して真の鍵の候補を絞り込む。一方、線形和攻撃では、暗号化関数をひとつの多項式でなく複数の多項式の和として構成することで、補間攻撃に比べて必要となる平文・暗号文数を削減できるため、より効率的に鍵の候補を絞り込むことが可能となる。

32 本分析は、CNRSのVaudenayによるもの。NISTのAES関連サイト（http:// www.nist.gov/aes）の電子掲示板“an electronic forum for the AES”のCRYPTONの欄に掲載されている。

33 本分析は、IBM社のCoppersmithによるもの。NISTのAES関連サイト（http:// www.nist.gov/aes）の電子掲示板“an electronic forum for the AES”のMARSの欄に掲載されている。

1）によって暗号化された既知平文・暗号文ペアを2^{56}個入手できれば、線形解読法によって全数探索法よりも効率的に解読することができる（Wagner, Ferguson and Schneier［1998］）。

⑥LOKI97

ブロック長・鍵長128 bitのLOKI97は、2^{56}個の選択平文・暗号文ペアを用いた差分解読法によって解読可能であるほか、一部のバイアスを有する拡大鍵（拡大鍵全体の約25%）によって暗号化された2^{56}個の既知平文・暗号文ペアによって解読可能である（Rijmen and Knudsen［1998］）。

⑦MAGENTA

中間一致攻撃により、2^{64}個の選択平文を入手できれば、2^{64}回の暗号化処理によって解読可能。また、2^{33}個の既知平文を入手できれば、2^{97}回の暗号化処理によって解読可能である（Biham et al.［1998］）。

⑧MARS

MARSには等価鍵（同一の拡大鍵を生成する鍵のペア）が存在し、比較的容易に発見することができる。鍵長を160 bitに設定した場合には鍵検索アルゴリズムを2^{16}回繰り返すことによって等価鍵を見つけることができるほか、1248 bitに設定した場合には2^{32}回で等価鍵を見つけることができる。これは、MARSにおいて利用可能な鍵長が128～1248 bitと範囲が広いため、鍵長が長くなるにつれて鍵空間が拡大し、その結果等価鍵の発生確率が高くなることが原因とみられる（Saarinen［1998］）。

⑨RIJNDAEL

６段のRIJNDAELは、鍵長が128 bitの場合、線形和攻撃によって全数探索法よりも効率的に鍵を探索することができる（青木［1999］）。

⑩SAFER+

鍵長が256 bitの場合、２つの既知平文と2^{37} byteのメモリーを利用できれば、中間一致攻撃によって、2^{241}回の暗号化処理で鍵を探索可能である。また、関連鍵攻撃によって、256個の選択平文と、その平文を２つの関連鍵によってそれぞれ暗号化した暗号文を利用できれば、2^{216}回の暗号化処理によって効率的に鍵を探索することができる[34]。

34 本分析は、Counterpane Systems 社のKelseyによるもの。NISTのAES関連サイト（http:// www.nist.gov/aes）の電子掲示板“an electronic forum for the AES”のSAFER+の欄に掲載されている。

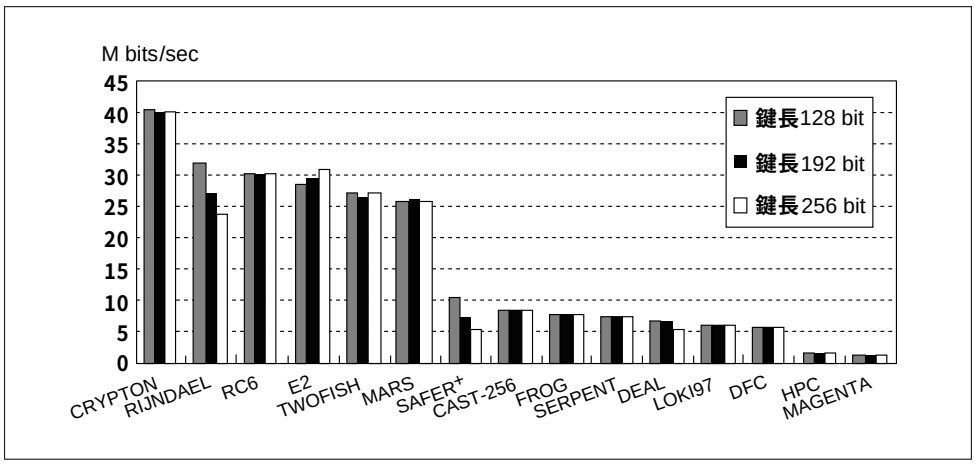
⑪SERPENT

SERPENTのS-box（4 bit入出力）の出力データ1 bitを入力データ4 bitのブール多項式で表すと、その次数が２次となるものが存在する。SERPENTの提案者は、S-boxのブール多項式の次数がすべて３次以上となることを前提に高階差分攻撃に対する安全性を確認しているが、この前提が正しいとは言えず、高階差分攻撃に対する安全性を再検討する必要がある（矢嶋・下山・辻井［1998］）

（2）処理速度に関する分析結果

現在、主にソフトウェアで実装した場合の処理速度の測定・評価が実施されている。NISTは、各候補アルゴリズムの提案者が提出したC言語のソースコードを基に、NISTが指定した実装環境（注27を参照）における処理速度の測定結果を発表している。これによると、NIST指定の実装環境における暗号化処理については、CRYPTON、RIJNDAEL、RC6、E2、TWOFISH、MARSの順番で高速となっている（図21参照[35]）。

図21　NISTによる暗号化速度の測定結果



35　この図は、NISTが1998年12月8日付で公表した資料から引用したものである。本資料には、暗号化処理のほか、復号処理や鍵セットアップの処理速度測定結果が含まれている。なお、本資料についてもAES専用サイトから入手可能。

また、Schneier et al.［1999］やGladman［1998］等、民間の研究チームも、8 bit CPU等の様々な実装環境における処理速度の試算・測定を実施している。NISTは、独自の測定結果だけではなく、このように発表されている様々な測定結果を参考にして候補アルゴリズムを評価するとしている。

ただし、処理速度を評価する場合、①各候補アルゴリズムの中には、特定の実装環境において高速処理が可能となるように設計されているものもあり、そうしたアルゴリズムをどのように評価するのか、また②NISTはAESを幅広い実装環境において利用可能なアルゴリズムとして位置づけているが、評価の際に各実装形態をどのようにウェイトづけするのか、といった問題点が残されている。これらの点については、今後開催されるAES候補コンファレンスや、NISTのAES関連サイトの電子掲示板等の場において議論されることとなろう。

（3）新しい解読法の発表

テクニオンのBiham、Biryukov、ワイツマン研究所のShamirは、第１回AES候補コンファレンス直後に開催された暗号の国際学会CRYPTO'98において、共通鍵ブロック暗号の新しい解読法 "Impossible Differential Cryptanalysis" を考案した旨を発表し、今後これを利用して各候補アルゴリズムの解読を試みるとの考えを示している。

Impossible Differential Cryptanalysisは差分解読法の一種であり、ある特定の差分を有する平文ペアに対し、絶対生成されることがない差分を有する暗号文ペアを利用して鍵の候補を絞り込む方法である。まず、真の鍵の下で、ある特定の差分ΔPを有する平文ペアに対して、どのような暗号文ペアの差分がどの程度の確率で生成されるかを調べ、生成確率がゼロとなる暗号文ペアの差分ΔCを見つける。次に、ΔPの差分を有する平文ペアを任意の鍵Kで暗号化して、その結果生成される暗号文ペアの差分ΔDとΔCを比較する。ΔD=ΔCが成立する場合、鍵Kは真の鍵ではないと判断することにより、候補となる鍵の絞り込みが可能となる。

Bihamらは、この攻撃法を利用して31段のSkipjack[36]を解読するために必要な選択平文・暗号文数と計算量を見積もっており、第１段から第31段のSkipjackに対しては、2^{41}個の選択平文・暗号文ペアと2^{78}回の暗号化処理が必要になるとしている（Biham, Biryukov and Shamir［1998］）。

36 Skipjack：1993年にNSAによって開発されたブロック長64 bit、鍵長80 bit、段数32のブロック暗号方式。Skipjackのアルゴリズムは、発表当初から非公開とされていたが、1998年6月に米国政府によって公開された（アルゴリズムの詳細については、NIST［1998a］を参照）。Skipjackは、1993年4月に米国政府が発表したキーエスクロー政策（暗号の利用者に対して、暗号化鍵に関する情報を２つの信頼できる機関に寄託することを義務づけ、法律執行上の必要が生じた場合には、捜査当局がこれらの情報を基に暗号鍵を復元することを可能にする仕組み）において利用される専用装置Clipper Chipに組み込まれる暗号アルゴリズムに指定されていた。また、Skipjackのアルゴリズムは、ISO/IEC 9979（Information technology - Security techniques － Procedures for the registration of cryptographic algorithms）に基づくISO暗号アルゴリズム登録制度に登録されている。

Ⅶ．おわりに　―DESからAESへ―

DESは、これまで共通鍵暗号の事実上の標準として利用されてきたが、鍵長が56bitと比較的短いことから、Brute Force Methodに対する安全性低下の懸念が多くの暗号学者によって指摘されてきた。RSA社が開催してきた４回のDES解読コンテストの結果は、こうした指摘の正当性を実証するものである。こうした背景から、米国政府は、DESに代わってTriple DESを米国政府標準暗号に認定することを決定している。

一方、米国政府は、Brute Force MethodやShort Cut Methodに対して十分な安全性を有する単体の共通鍵ブロック暗号として、AESの標準化を進めている。NISTは、AESを米国政府標準暗号として20～30年の間利用する方針を発表しており、Triple DESの次の主要な暗号方式となる可能性が高い。標準化スケジュールによると、AESの標準化は早ければ2000年にも完了する見通しであるが、AESがその安全性について高い信頼を得て、AESを利用した暗号製品が広く普及するまでには、標準化完了後さらに数年程度はかかるものとみられている。AESが実際に利用可能になるまでの間、Triple DESがDESの後継暗号として金融分野を中心に幅広い分野において利用されるようになり、今後主要な共通鍵ブロック暗号は「DES→Triple DES→AES」と移行していくと考えられる。

現在、米国政府は、AESの候補アルゴリズムの安全性・処理速度等に関する分析・評価を進めているほか、各国の暗号学者・研究者も候補アルゴリズムの安全性に関する分析を積極的に進めている。今後、こうしたAESの標準化の進展とともに、共通鍵暗号に関する研究がより一層活発になることが望まれる。共通鍵暗号の研究動向について、引き続き注視していく必要があろう。

表７　AESの候補アルゴリズムの設計方針（1/2）

上段：名称 中段：代表提案者 下段：代表者の国名・所属名	各アルゴリズムの論文に明記されている設計方針
①CAST-256 Adams カナダ・Entrust Technologies社	・CAST-128を改良：CAST-128のアルゴリズムをベースに、安全性および処理速度を向上させる。
②CRYPTON Lim 韓国・Future Systems社	・安全性：差分攻撃法や線形攻撃法等、既存の攻撃法に対して、十分な安全性を確保する。 ・処理速度：非線形変換を並列処理することが可能な構造を採用し、ハードウェア・ソフトウェア両方において高速処理を可能にする。 ・簡素な構造：安全性の分析を容易にするために、シンプルな構造にする。
③DEAL Outerbrigde カナダ	・安全性：DESのアルゴリズムをF関数に利用することにより、既存の解読法に対して安全性を確保する。 ・処理速度：Triple DESと同程度の処理速度で実装可能にする。 ・実装：既にDESが実装されている環境において、DESからの移行を容易にする。
④DFC Vaudenay フランス・CNRS	・安全性：decorrelation theoryを利用することで、Triple DESよりも高い安全性を確保する。 ・処理速度：DESよりも高速で実装可能にする。 ・実装：パソコンやICカード等、様々なプラットフォームにおいて実装可能にする。
⑤E2 神田雅透 日本・NTT社	・安全性：差分解読法・線形解読法のほか既存の攻撃法に対して十分な安全性を有することを客観的尺度によって示す。 ・処理速度：簡素なラウンド関数を利用することで、高速処理を実現する。 ・実装：あらゆるプラットフォーム上で柔軟な実装が可能となる構造とする。
⑥FROG Georgoudis コスタリカ・TecApro Internacional社	・安全性：鍵データによって毎回異なる変換表を作成し、変換処理自体を変更することによって安全性を確保する。 ・処理速度：簡素な変換処理を採用することによって、高速処理を可能にする。 ・鍵長とブロック長：鍵長とブロック長の様々な組み合わせによる実装を可能にする。 ・実装：ICカード、ATM、HDTV、B-ISDN等、様々なプラットフォームでの実装を可能にする。
⑦HPC Schroeppel 米国・アリゾナ大学	・安全性：既存の攻撃法に対して十分な安全性を確保する。 ・処理速度：64 bitの汎用CPUを利用したソフトウェア実装において最高速度での暗号化・復号処理を可能にする。 ・実装面での柔軟性：任意の鍵長やブロック長で利用可能とし、様々なアプリケーションに柔軟に対応できるようにする。
⑧LOKI97 Brown オーストラリア・Australian Defense Force Academy	・LOKI89とLOKI91を改良：LOKI89やLOKI91のアルゴリズムをベースとする。 ・安全性：差分解読法、線形解読法、関連鍵攻撃等既存の攻撃法に対して、十分な安全性を確保する。

表７　AESの候補アルゴリズムの設計方針（2/2）

上段：名称 中段：代表提案者 下段：代表者の国名・所属名	各アルゴリズムの論文に明記されている設計方針
⑨MAGENTA Huber ドイツ・Deutsche Telekom社	・アルゴリズムの構造：アバランシュ性等、安全性の観点から望ましい性質を有する関数を利用して、簡素な構造とする。 ・実装：ハードウェアとソフトウェアの両方で高速処理を可能にする。
⑩MARS Zunic 米国・IBM社	・安全性：複数種類のラウンド関数を利用することで高い安全性を実現する。 ・処理速度：32 bit汎用CPUを搭載したコンピューターにおけるソフトウェア実装において最も高速での処理を可能とする。 ・分析が容易な構造：安全性に関する分析が容易となるような変換手段を利用してアルゴリズムを構築する。
⑪RC6 Robshaw 米国・RSA Laboratories社	・安全性：既存の攻撃法に対して安全性を確保する。 ・処理速度：32 bitデータの乗算等32 bit CPUによるソフトウェア実装において高速処理が可能な演算を組み合わせる。 ・簡素な構造：安全性に関する分析が容易となるように、アルゴリズムの構造をシンプルにする。
⑫RIJNDAEL Daemen ベルギー・Banksys社	・安全性：既存の攻撃法に対して十分な安全性を確保できる。 ・実装：様々なプラットホームにおいて実装可能にする。 ・簡素な構造：安全性に関する分析が容易となるように、アルゴリズムの構造をシンプルにする。
⑬SAFER+ Chen 米国・Cylink社	・安全性：SAFER-KおよびSAFER-SKのアルゴリズムをベースに、既存の攻撃法に対して安全性を確保する。 ・処理速度：シンプルなアルゴリズム構造により、様々な実装環境において高速処理を実現する。
⑭SERPENT Anderson イギリス・イスラエル・ノルウェー	・安全性：DESのS-box等これまで研究の蓄積のある変換処理や構造を採用することで安全性評価を容易にし、Triple DESよりも高い安全性を確保する。 ・処理速度：bitslice implementationによって高速処理が可能な構造とする。
⑮TWOFISH Schneier 米国・Counterpane Systems社	・安全性：安全性に対する信頼性を高めるために、これまで安全性に関する研究が十分行われている変換処理や構造を利用する。 ・処理速度：様々なプラットフォームにおいて高速処理が可能となるような構造を選択する。 ・簡素な構造：安全性に関する分析が容易となるように、ラウンド関数の構造をできるだけシンプルにする。

表８　AESの候補アルゴリズムの構造上の特徴（1/3）

名称	全体構造	鍵長（上段）、 ブロック長（下段）	データランダム化部 （拡大鍵でデータブロックを変換）	鍵スケジューリング部 （拡大鍵を生成）
①CAST-256	・一般形Feistel構造を利用 ・データブロックを4つの32 bit サブブロックに分割して変換	・128, 192, 256 bit ・128 bit	3種類のF関数を利用 ・段数は48 ・S-box（8 bit入力、32 bit出力）は4種類	データ変換部のアルゴリズムを利用
②CRYPTON	・SPN構造を利用 ・並列処理が可能な構造 ・データブロックを4つの32 bit サブブロックに分割して変換	・可変（32 bit毎に64 bitから256 bitまで利用可能） ・128 bit	ラウンド関数（段数は12）には、換字変換（S-box）、bit単位の転置変換、byte単位の転置変換、拡大鍵との排他的論理和（XOR）を連続的に利用 ・4つのS-box（32 bit入出力）を利用	鍵（256 bit未満の場合には不足bitをpadding）から52個の32 bitの拡大鍵を生成
③DEAL	・Feistel構造を利用 ・データブロックを2つの64 bit サブブロックに分割	・128, 192, 256 bit ・128 bit	ラウンド関数には、F関数とXORを利用 ・F関数には、DESのアルゴリズムを利用 ・段数は6段以上	鍵を64 bit単位に分割後、DESのアルゴリズムを利用し、段数に等しい数の64 bitの拡大鍵を生成
④DFC	・Feistel構造を利用 ・並列処理が可能な構造 ・データブロックを2つの64 bit サブブロックに分割して変換	・可変 （上限は256 bit） ・128 bit	ラウンド関数（2種類）には、2^64を法とする加算と乗算、XOR、非線形変換を利用 ・段数は8段 ・非線形変換は変換表（6 bit入力、32 bit出力）を利用	鍵をpaddingによって256 bitとし、4段のFeistel構造を有する変換部（XOR等を利用）によって128 bit拡大鍵を8個生成
⑤E2	・Feistel構造を利用 ・データブロックを2つの64 bit のサブブロックに分割して変換 ・データランダム化部の最初と最後に、算術演算、byte単位の転置変換等を実施	・128, 192, 256 bit ・128 bit	ラウンド関数には、F関数とXORを利用。ラウンド関数の処理を簡素にする一方、安全性の観点から十分な段数を確保 ・段数は12 ・F関数には、2-round SPN構造を採用し、16のS-boxを利用 （10項目の安全性評価基準を基にS-boxを選択）	拡大鍵生成過程でbyte単位での転置変換を行い、拡大鍵の一部から鍵の発見を困難にしている。128 bitの拡大鍵を16個生成
⑥FROG	・鍵から3種類の変換表を8組作成。8組の変換表は互いに異なっており、ラウンド関数1段の変換に1組ずつを利用 ・各ラウンド関数の変換はすべて異なる内容	・可変 （40 bit～1000 bit） ・可変 （64 bit～1024 bit）	ラウンド関数には3つの変換表を利用。データブロックをbyte単位で、4段階に分けて変換 ・変換手順は「XOR→換字変換→XOR→XOR」であり、データブロックのbyte数だけ繰り返し ・段数は8 ・3つの変換表はそれぞれ8組準備	3つの変換表が拡大鍵の役割を担っている。変換表の大きさはブロック長に依存。鍵からXOR、乱数による変換等を利用して、変換表を生成

表８　AESの候補アルゴリズムの構造上の特徴（2/3）

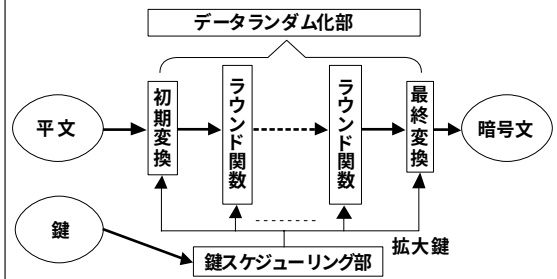
名称	全体構造	鍵長（上段）、 ブロック長（下段）	データランダム化部 （拡大鍵でデータブロックを変換）	鍵スケジューリング部 （拡大鍵を生成）
⑦HPC	・暗号鍵のほかに、変換に利用する変数としてSPICEと呼ばれる256 bitデータを利用 ・データブロックを64 bit単位のwordに分割して変換 ・8段のラウンド関数の最初と最後に拡大鍵とのXORを実行	・可変 ・可変	ラウンド関数にはXOR、法2^64を法とする加算、引算、bitシフトが多用 ・段数は8段	任意の長さの鍵から、256個の64 bit拡大鍵を生成。乱数による初期値を基に256個の64 bitデータを生成した後、各データと64 bit毎に分割した鍵とのXORを計算して拡大鍵を生成
⑧LOKI97	・Feistel構造を利用 ・データブロックを2つの64 bitサブブロックに分割して変換 ・各段のラウンド関数の変換を複雑にする一方、段数を少なくして処理速度の低下を抑制	・128, 192, 256 bit ・128 bit	ラウンド関数には、F関数と2^64を法とする加算を利用 ・F関数（入出力64 bit）には2-round SPN構造を利用 F関数では、「拡大鍵を変数とする転置変換→拡大変換（64→96 bit）→換字変換（2種類8個のS-box＜13→8 bitおよび11→8 bit＞）→転置変換→換字変換（2種類8個のS-box）」という手順で変換 ・S-boxは、非線形次数等4つの基準を基に生成 ・段数は16	鍵を64 bit単位に分割後、F関数を用いて48個の64 bitの拡大鍵を生成
⑨MAGENTA	・Feistel構造を利用 ・128 bitデータブロックを2つの64 bitサブブロックに分割して変換	・128, 192, 256 bit ・128 bit	ラウンド関数には、E関数とXORを利用 ・E関数（入出力64 bit、通常のラウンド関数のF関数に相当）では、Fast Hadamard Transform（FHT）と呼ばれる線形変換に、法256によるべき乗演算（f関数と呼ばれる）を中心とする非線形変換によって改良を加えた関数（T関数と呼ばれる）を採用。変換は8 bit単位で実行 ・段数は、鍵長128、192 bitの場合は6、256 bitの場合は8	特別な変換はなく、64 bit単位に分割した鍵をそのままデータランダム化部に入力
⑩MARS	・type-3 Feistel構造を採用 ・データブロックを4つの32 bitサブブロックに分割して変換 ・データランダム化部の最初に拡大鍵との加算、最後に拡大鍵との引算を実行	・可変 （128～1248 bit） ・128 bit	4種類のラウンド関数（各8段）を利用。換字表（S-box）、XOR、加算、乗算、データ依存型・非依存型bitシフトを採用 ・最初と最後の8段はS-box（入力8 bit、出力32 bit）とデータ非依存型bitシフトを中心に構成され、中間の16段は、S-box（入力9 bit、出力32 bit）やデータ依存型bitシフトを含むE関数（通常のラウンド関数のF関数に相当）によって構成されている ・S-boxは、SHA-1のアルゴリズムを基に生成し、差分特性や線形特性を考慮して選択 ・段数は32	段数7のFeistel構造を利用。転置変換やXORによる変換に加え、生成した拡大鍵に弱鍵がないことを検証するアルゴリズムを採用。32 bit拡大鍵を40個生成

表８　AESの候補アルゴリズムの構造上の特徴（3/3）

名称	全体構造	鍵長（上段）、 ブロック長（下段）	データランダム化部 （拡大鍵でデータブロックを変換）	鍵スケジューリング部 （拡大鍵を生成）
⑪RC6	・高速処理が可能な算術演算を中心とする構造 ・データブロックを４つのサブブロックに分割して変換	・可変 （2040 bit以下） ・可変	ラウンド関数には、データ依存型bitシフトのほか、2^w（w：サブブロック長）を法とする加算、引算、乗算、XORを利用 ・段数は可変（20段以上を推奨）	鍵から、2r+4個のブロック長と同じ長さの拡大鍵を生成（r：段数）
⑫RIJNDAEL	・SPN構造を採用 ・データはbyte単位で変換 ・データランダム化部の最初には鍵ブロックとの加算を実行し、最終段では、換字変換、bitシフト、XORを実行	・128,192,256 bit ・128,192,256 bit	ラウンド関数には３種類の変換部が存在し、線形変換層（bitシフト等）、非線形変換層（S-boxによる換字変換）、拡大鍵変換層（拡大鍵とのXOR）を採用 ・S-boxには、2^8を法とするべき乗剰余演算を利用 ・段数はブロック長と鍵長に依存（10,12,14段のいずれか）	鍵から、bitシフト変換と換字変換によって拡大鍵を生成。拡大鍵の長さはブロック長に等しく、(r+1）個生成される（rは段数）
⑬SAFER+	・４つの変換層から構成されるラウンド関数を繰り返して暗号化・復号する構造を採用 ・サブブロックは8 bit ・ラウンド関数の最終段が終了後、byte単位での法256の乗算とXORによる変換を配置	・128, 192, 256 bit ・128 bit	ラウンド関数は4種類の変換層によって構成。第１および第３層では256を法とする加算とXOR、第２層では指数関数45^Xmod 257と対数関数$\log_{45}$X、第４層では法256の乗算を採用 ・段数は鍵長に依存（鍵長と段数の対応：（128,192, 256 bit）→（8, 12, 16段）） ・変換処理はbyte単位で実行 ・16×16の行列を利用している点が従来のSAFERとの差異	鍵から、2r+1個の128 bitの拡大鍵を生成（r：段数）。128 bitの乱数（biasと呼ばれている）を利用して変換 ・SAFER-SKで利用されている鍵スケジューリング部の構造を利用
⑭SERPENT	・SPN構造を採用 ・Bitslice implementation が可能な構造を利用 ・データランダム化部の最初と最後には転置変換を配置	・可変 （上限256 bit） ・128 bit	ラウンド関数は、拡大鍵とのXOR、32個のS-box（入出力4 bit）、線形変換によって構成 ・S-boxは8種類存在し、DESのS-boxを利用して生成。bitslice modeでは、4種類の32 bit入出力のS-boxを利用 ・段数は32	鍵をpaddingして256 bitに拡張した後、bit シフト、S-boxを用いて33個の128 bit拡大鍵を生成
⑮TWOFISH	・Feistel構造を採用 ・データブロックを４つの32 bitサブブロックに分割して変換 ・データランダム化部の最初と最後に拡大鍵とのXORを配置	・128,192,256 bit ・128 bit	ラウンド関数はXORとF関数から構成されており、F関数はbitシフト、g関数（８つのS-boxによる換字変換と線形変換によって構成）、2^32を法とする拡大鍵との加算によって構成 ・S-box（8 bit入出力）は4種類存在し、各々2個ずつが利用 ・線形変換には、32×32 bitの正方行列による積を利用 ・段数は16	ラウンド関数で利用した変換（g関数）を利用して、40個の32 bit拡大鍵を生成

（注）「XOR」は、排他的論理和を表す。

表9　アルゴリズムの構造に関する概念整理

<table>
<tr>
<td>アルゴリズムの構造</td>
<td>データランダム化部:
鍵スケジューリング部から入力された拡大鍵を利用して、平文ブロックの暗号化や暗号文ブロックの復号を行う部分。一般的には、初期変換、最終変換、ラウンド関数といった変換手段によって構成される。</td>
<td>ラウンド関数:
データの暗号化や復号の基本変換となる部分。排他的論理和や加算などの複数の変換手段が含まれる。通常、データランダム化部には複数のラウンド関数が含まれる。</td>
<td>鍵スケジューリング部:
鍵ブロックから、鍵スケジューリング部に利用するための拡大鍵を生成する部分。</td>
<td colspan="2"> 

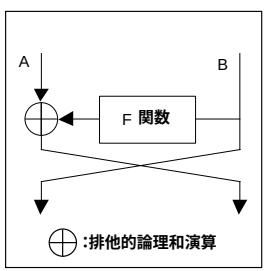
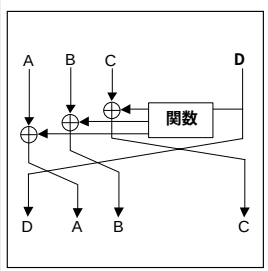
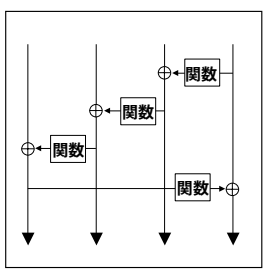
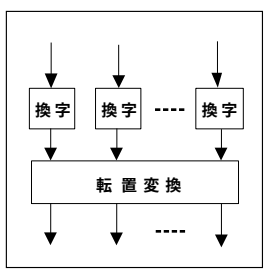
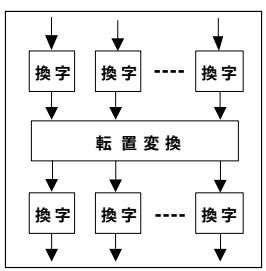
＜一般的な共通鍵ブロック暗号の構造＞</td>
</tr>
<tr>
<td>ラウンド関数やF関数の構造</td>
<td>Feistel（type-1 Feistel）構造:
2つのデータブロックA，Bに対して、一方のデータブロックBをF関数と呼ばれる非線形変換手段によって変換し、他方のデータブロックAを変換後のデータブロックBとの排他的論理和によって変換する。変換後、2つのデータブロックの位置を入れ替える。
たとえば、DESやFEALなどのラウンド関数に採用されている。
 
</td>
<td>type-3 Feistel構造:
4つのデータブロック（A，B，C，D）のうち、ひとつのデータブロックDをある関数によって変換し、その変換結果を用いて他の3つのデータブロックを排他的論理和によって変換する。変換後は、各データブロックの転置変換を行い、（A，B，C，D）→（D，A，B，C）とする。
たとえば、MARSのラウンド関数において利用されている。
 
</td>
<td>一般形Feistel構造:
4つのデータブロックを4つの関数でそれぞれ変換し、変換後の各データブロックを用いて他のデータブロックを排他的論理和で変換する。
たとえば、CAST-256のラウンド関数に利用されている。
 
</td>
<td>SPN構造:
複数のデータブロックを換字変換（substitution）によって変換した後、転置変換（permutation）によって変換する。たとえば、CRYPTON等のラウンド関数に利用されている。
なお、転置変換の代わりに、より一般的な線形変換が利用される場合も、SPN構造と呼ばれることがある。このような例として、SERPENTのラウンド関数が挙げられる。
 
</td>
<td>2-round SPN構造:
各データブロックを「換字変換→転置変換→換字変換」の順番で変換する。たとえば、LOKI97のF関数に利用されている。
なお、転置変換の代わりにより一般的な線形変換が利用されている場合も、2-round SPN構造と呼ばれることがある。このような例としては、E2のF関数が挙げられる。
 
</td>
</tr>
</table>

表10　AESの候補アルゴリズムの安全性に関する分析結果（1/3）

名称	提案者による分析結果			提案者以外による解読法に関する研究成果
	差分解読法	線形解読法	高階差分攻撃等その他	
①CAST-256	「40段の最大差分特性確率は2^(140)以下であり十分な安全性を有している」と評価 ・1段の最大差分特性確率は2^(-14)以下	「48段の最大線形特性確率は2^(-122)であり、十分な安全性を有している」との評価	「高階差分攻撃や関連鍵攻撃等既存の攻撃法は適用困難」との評価	—
②CRYPTON	「8段の最大差分特性確率は2^(-160)であり、差分解読法に対して十分な安全性を有している」との評価 ・S-boxの最大差分特性確率は、2^(-5)	「8段の最大線形特性確率は2^(-128)であり、線形解読法に対して十分な安全性を有している」との評価 ・S-boxの最大線形特性確率は、2^(-4)	「高階差分攻撃に対しては、S-boxの非線形次数が5であり、4段の非線形次数が5^4(>128)となることから、十分な安全性を有している」との評価	青木［1999］:「線形和攻撃により、6段で128 bit鍵長の場合、全数探索法よりも効率的に鍵を探索可能」 Vaudenay:「鍵長256 bitの場合には2^(-224)の確率で弱鍵が存在」
③DEAL	「6段の場合の選択平文攻撃には、2^70個以上の選択平文と2^121回のDESの暗号化処理が必要となることから、適用困難。8段の場合、全数探索法よりも効率的な解読法は存在しない」との評価	—	「弱鍵が発見されているものの、極めて少数であり、実際の脅威とはならない」との評価	Lucks［1998b］:「6段で鍵長192 bitのDEALは、2^33個の選択平文と2^145の計算量によって解読可能」
④DFC	「2^110個以上の選択平文が必要であり、適用困難」との評価	「2^92個以上の既知平文が必要であり、適用困難」との評価	「同一の鍵を2^48回以上の暗号化に利用しないことが安全性を確保する上で必要」との評価	Coppersmith:「ラウンド関数の出力データが入力データに依存しない拡大鍵が2^(-64)の確率で発生する」
⑤E2	「9段の最大差分特性確率は2^(-140.34)以下であり、高い確率の差分特性は存在せず、適用困難」との評価 ・F関数の最大差分特性確率は2^(-23.39)以下 ・S-boxの最大差分特性確率は2^(-4.67)	「9段の最大線形特性確率は2^(-131.52)以下であり、効率的な線形特性は存在せず、適用困難」との評価 ・F関数の最大線形特性確率は2^(-21.92)以下 ・S-boxの最大線形特性確率は2^(-4.38)	「3段以上で高階差分攻撃は適用困難なほか、補間攻撃も適用困難」との評価 ・S-boxのブール多項式の最大次数は7(F関数では40以上) ・S-boxの多項式表示における項数:254以上 ・S-boxの差分線形確率:2^(-2.59)	青木［1999］:「線形和攻撃により、5段で128 bit鍵長の場合、全数探索法よりも効率的に鍵を探索可能」 時田・松井［1999］:「bytewise differential cryptanalysisにより、8段の場合、2^97個の選択平文によって8段目の拡大鍵を効率的に探索可能」
⑥FROG	「全数探索法よりも効率的な解読法はない」との評価	—	—	Wagner et al.［1998］:「弱鍵(2^95個存在)によって暗号化された2^58個の選択平文・暗号文ペアによって解読が可能」

表10　AESの候補アルゴリズムの安全性に関する分析結果（2/3）

名称	提案者による分析結果			提案者以外による解読法に関する研究成果
	差分解読法	線形解読法	高階差分攻撃等その他	
⑦HPC	「通常の差分解読法は適用困難」と評価する一方、「攻撃者にとって都合のよいSPICEデータを利用することによって解読を試みる『選択SPICE攻撃』に対する安全性については未知数」としている	「線形解読法は適用困難」と評価	「暗号化の際に利用される中間データや拡大鍵の量を基に推定すると、解読に成功するためには少なくとも2^400の暗号化処理が必要」と評価	Coppersmith：「SPICEを安全に配送する方法について何らの提案もなく、SPICEを利用した攻撃法が成立する可能性がある」
⑧LOKI97	「ラウンド関数に加算が用いられているため、通常の差分特性確率を数学的に算出することは困難」との評価	「最大線形特性確率は14段で2^(-141)であり、適用困難」との評価 ・1段の最大線形特性確率は2^(-11)	—	Rijmen and Knudsen［1998］：「2^56個の選択平文あるいは既知平文によって解読可能」
⑨MAGENTA	「E関数（通常のラウンド関数のF関数に相当）の差分特性確率は、少なくとも2^(-40)であり、ラウンド関数の差分特性確率はDESよりも小さくなっているとみられることから適用困難」との評価 ・f関数の差分特性確率は2^(-5)以下 ・T関数の差分特性確率は2^(-20)以下	「E関数の線形特性確率は約2^(-29)以下であることから、適用困難」との評価 ・f関数の線形特性確率は約0.6以下	—	Biham et al.［1998］：「2^64個の選択平文を入手すれば2^64回の暗号化処理で解読できるほか、2^33の既知平文があれば2^97回の暗号化処理で解読可能」
⑩MARS	「2^280　個以上の選択平文が必要であり、適用困難」との評価 ・E関数(通常のラウンド関数のF関数に相当　の差分特性確率は概ね2^(-9) ・16段の差分特性確率は概ね2^(-240)	「2^128個以上の既知平文が必要であり、適用困難」との評価 ・4段の線形特性確率は概ね2^(-69)	「弱鍵は見つかっていないほか、光学的暗号攻撃や故障利用暗号攻撃の適用は困難」との評価	Saarinen［1998］：「鍵長160 bitでは2^16の計算量で等価鍵が見つかるほか、鍵長1248 bitでは2^32の計算量で見つかる」
⑪RC6	「18段の最大差分特性確率は概ね2^(-264)であり、適用不可能」との評価	「18段のRC6に線形解読法を適用するためには、少なくとも2^182個の既知平文が必要であり、適用不可能」との評価	「高階差分攻撃や弱鍵の発見に必要なデータを入手することは困難」と評価	—
⑫RIJNDAEL	「8段の最大差分特性確率は2^(-350)であり、適用不可能」との評価 ・S-boxの最大差分特性確率は2^(-7)	「8段の最大線形特性確率は2^(-300)であり、適用不可能」との評価 ・S-boxの最大線形特性確率は2^(-6)	「補間攻撃や関連鍵攻撃に対して安全であるとみられる」との評価	青木［1999］：「線形和攻撃により、6段で128 bit鍵長の場合、全数探索法よりも効率的に鍵を探索可能」

表10　AESの候補アルゴリズムの安全性に関する分析結果（3/3）

名称	提案者による分析結果			提案者以外による解読法に関する研究成果
	差分解読法	線形解読法	高階差分攻撃等その他	
⑬SAFER+	「６段以上のSAFER+に対して適用困難」との評価 ・5段の差分特性をすべて調べた結果、差分特性確率は2^(-128)以下	「3段以上のSAFER+に対して適用困難」との評価 ・2.5段において、線形バイアスを有する入出力ペアが見つかっていない	「拡大鍵生成に乱数を利用していることから、弱鍵や関連鍵が発生する可能性は低い」との評価	Kelsey：「256 bit鍵長の場合、中間一致攻撃によって2つの既知平文と2^38 byteのメモリーを用いて2^241回の暗号化処理で効率的に鍵を探索可能。また、関連鍵攻撃により、256個の選択平文と、2つの関連鍵により、2^216回の暗号化処理で解読可能」
⑭SERPENT	「６段の最大差分特性確率が2^(-58)以下であり、24段の場合には2^(-232)以下となることから、適用困難」との評価	「24段の最大線形特性確率は2^(-109)以下であることから、適用困難」との評価	「高階差分攻撃は適用困難」との評価 ・S-boxのブール多項式次数が3であることからr段後の出力データの次数は3^r(5段で243)	矢嶋他［1998］：「S-boxの中で、出力データのブール多項式が2次になるものが存在」
⑮TWOFISH	「7段のTWOFISHに対して、2^131個の選択平文が必要であることから、適用困難。全数探索法よりも効率的な解読法は知られていない」との評価	「12段のTWOFISHに対して、2^121個の既知平文が必要であることから、適用困難」との評価	「S-boxは高次の非線形性を有していること等から、高階差分攻撃、補間攻撃、関連鍵攻撃に対して高い安全性を有している」との評価	—

表11　AESの候補アルゴリズムの処理速度に関する分析結果（1/2）

		C言語			アセンブリ言語	JAVA		実装環境
		Borland C++ 5.0	MSV C 5.0	その他		JDK	その他	
①CAST-256	暗号化	14 M			47 M			•Borland: NIST指定
	復号	14 M			47 M			•アセンブリ:
	鍵セットアップ	〈9090〉			〈4130〉			Pentium Ⅱ300MHz, Windows NT 4.0
②CRYPTON	暗号化		51 M		62 M			•MSV 5.0:
	復号		51 M		62 M			Pentium Pro 200MHz, 32 MB RAM, Windows95
	鍵セットアップ		〈325〉		—			•アセンブリ:詳細不明
③DEAL	暗号化	DES暗号化の1/6						
	復号	DES復号の1/6						
	鍵セットアップ	DES鍵セットアップの1/7						
④DFC	暗号化			34 M				•C言語その他:Pentium Pro 200MHz
	復号			—				
	鍵セットアップ			—				
⑤E2	暗号化	36 M			62 M	11 M*		•Borland:NIST指定
	復号	36 M			62 M	11 M*		•アセンブリおよびJAVA JDK:
	鍵セットアップ	〈2076〉			—	—		Pentium Pro 200MHz, 64MB RAM, Windows95
⑥FROG	暗号化			10 M				•C言語その他:
	復号			13 M				Pentium 200MHz, 64MB RAM, Windows95
	鍵セットアップ			〈1960000〉				
⑦HPC	暗号化			7.3 M				•C言語その他:Pentium 200MHz
	復号			7.3 M				
	鍵セットアップ			〈140000〉				
⑧LOKI97	暗号化		6.1 M					•MSV C:
	復号		6.3 M					Pentium Ⅱ 233MHz, 64 MB RAM, Windows95
	鍵セットアップ		〈4870〉					

表11　AESの候補アルゴリズムの処理速度に関する分析結果（2/2）

		C言語			アセンブリ言語	JAVA		実装環境
		Borland C++ 5.0	MSV C 5.0	その他		JDK	その他	
⑨MAGENTA	暗号化			1.1 M				•C言語その他:Pentium Pro 200MHz
	復号			1.1 M				
	鍵セットアップ			〈7100〉				
⑩MARS	暗号化	28 M		85 M		15 M		•Borland:NIST指定
	復号	28 M		85 M		17 M		•C言語その他:PowerPC 604e, C Set ++3.1.1
	鍵セットアップ	〈9200〉		〈2050〉		〈1760〉		•JAVA JDK:NIST指定
⑪RC6（20段）	暗号化			42 M	101 M	1.6 M		•C言語その他およびアセンブリ言語:
	復号			45 M	101 M	1.6 M		PentiumⅡ266MHz, 32MB RAM, Windows95
	鍵セットアップ			〈4710〉	—	〈107000〉		•JAVA JDK:Pentium Pro 180MHz, 64MB RAM, WindowsNT 4.0
⑫RIJNDAEL	暗号化			27 M	80 M		1.1 M	•C言語その他:Pentium 200MHz, Linux
	復号			27 M	80 M		1.1 M	•アセンブリ言語:Pentium 200MHz, Linux
	鍵セットアップ			〈2100〉	—		—	•JAVAその他:Pentium 200MHz, Linux
⑬SAFER+	暗号化			12 M				•C言語その他:
	復号			12 M				Pentium 200MHz, 64MB RAM, Windows95
	鍵セットアップ			〈15342〉				
⑭SERPENT	暗号化	15 M				583 K		•Borland:NIST指定
	復号	15 M				583 K		•JAVA JDK:NIST指定
	鍵セットアップ	—				—		
⑮TWOFISH	暗号化	40 M	43 M		90 M			•Borland:NIST指定
	復号	40 M	43 M		90 M			•MSV Cおよびアセンブリ:
	鍵セットアップ	〈10300〉	〈8000〉		〈12700〉			Pentium Pro 200MHz, 64MB RAM, Windows95

（注）1. NISTが指定した実装環境は、CPU：Pentium Pro 200MHz、メモリー：64MB RAM、OS：Windows95を搭載したIBM PC互換機。プログラム言語はANSI C（Borland C++ 5.0）とJAVA（JDK 1.1）。

2. 表中の数値の単位はbps。鍵セットアップの数値（〈〉内の数値）の単位はclock cycle。

3. 暗号化および復号は、鍵長128 bit、ブロック長128 bitのケースを想定。

4.「*」は、JIT compilerを利用した場合の計数。

5. 各計数は、第１回AES候補コンファレンス（1998年8月20～22日）に発表された各資料に掲載されているもの。CAST-256、DEAL、HPC、SERPENTの計数は試算値であり、これら以外のアルゴリズムに関する計数は実測値である。

参考文献

青木和麻呂、「線形和攻撃」、1999年暗号と情報セキュリティシンポジウム予稿集、pp. 691-694、1999年1月

岩下直行・谷田部充子、「金融分野における情報セキュリティ技術の国際標準化動向」、『金融研究』第18巻第２号、日本銀行金融研究所、1999年4月

宇根正志、「AES（Advanced Encryption Standard）について」、日本銀行金融研究所ディスカッションペーパーシリーズ、No. 97-J-16、1997年

――――、「最近のAESを巡る動向について」、日本銀行金融研究所ディスカッションペーパーシリーズ、No. 98-J-21、1998年

神田雅透、「AES暗号について」、1999年暗号と情報セキュリティシンポジウム予稿集、pp. 1-8、1999年1月

――――・盛合志帆・青木和麻呂・植田広樹・大久保美也子・高嶋洋一・太田和夫・松本勉、「128ビットブロック暗号E２の提案」、電子情報通信学会技術報告ISEC98-12、1998年7月

下山武司・盛合志帆・金子敏信、「高階差分攻撃の改良とKN暗号の解読」、電子情報通信学会技術報告ISEC97-29、 1997年

反町 亨・松井充、「RC5の強度評価に関する一考察（その３）」、1997年暗号と情報セキュリティシンポジウム予稿集SCIS'97-18A, 1997年

田中秀磨・久松和之・金子敏信、「FL関数のない場合のMISTY1に対する高階差分攻撃」、電子情報通信学会技術報告ISEC98-5、1998年

谷口文一・太田和夫・大久保美也子、「トリプルDESを巡る最近の標準化動向について」、1999年暗号と情報セキュリティシンポジウム予稿集、pp. 875-880、1999年1月

角尾幸保・山田真紀、「選択差分を用いた証明可能安全な暗号に対する攻撃に関する一考察」、1998年暗号と情報セキュリティシンポジウム予稿集SCIS'98-7.2.E、1998年

時田俊夫・松井 充、「Byte-oriented暗号の強度評価に関する考察」、1999年暗号と情報セキュリティシンポジウム予稿集、pp. 93-98、1999年1月

浜出 猛・横山尚史・島田徹・金子敏信、「DES暗号に対するpartitioning解析に関する一考察」、1998年暗号と情報セキュリティシンポジウム予稿集SCIS'98-2.2.A、1998年

久松和之・金子敏信、「FL関数のない場合のMISTY1に対する高階差分攻撃による強度評価の一考察」、1998年暗号と情報セキュリティシンポジウム予稿集SCIS'98-1.2.C, 1998年

松井 充、「ブロック暗号アルゴリズムMISTY」、電子情報通信学会技術報告ISEC96-11、1996年

松本 勉・岩下直行、「金融分野における情報セキュリティ技術の現状と課題」、『金融研究』第18巻第２号、日本銀行金融研究所、1999年4月

三上正・金子敏信、「二段消去攻撃におけるIDEA暗号の弱鍵」、電子情報通信学会技術報告ISEC97-110、1997年

盛合志帆・青木和麻呂・神田雅透・高嶋洋一・太田和夫、「既知のブロック暗号攻撃に対する安全性を考慮したS-boxの構成法」、電子情報通信学会技術報告ISEC98-13、1998年7月（http://info.isl.ntt.co.jp/~shiho/E2sbox.ps.gz）

――――・下山武司・金子敏信、「SNAKE暗号の補間攻撃」、1998年暗号と情報セキュリティシンポジウム予稿集SCIS'98-7.2.C、1998年

矢嶋純・下山武司・辻井重男、「共通鍵ブロック暗号SERPENT（AES候補）のラウンド関数の高階差分について」、電子情報通信学会技術報告ISEC98-39、1998年11月

C. Adams, "The CAST-256 Encryption Algorithm," 1998. (http://www.entrust.com/resources/pdf/cast-256.pdf)

R. Anderson, E. Biham and L. R. Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard," 1998a.（http://www.cl.cam.ac.uk/ftp/users/rja14/serpent.pdf）

――――, ――――― and ――――, "Serpent: A Flexible Block Cipher With Maximum Assurance," 1998b.（http://www.cl.cam.ac.uk/ftp/users/rja14/ventura.ps.gz）

American National Standards Institute, "X3.92 ― 1981, Data Encryption Algorithm," 1981.

――――, "X9.52 ― 1998, Triple Data Encryption Algorithm Mode of Operation," 1998.

E. Biham, "New types of cryptanalytic attacks using related key," Advances in Cryptology ― Proceedings of EUROCRYPT'93, Lecture Notes in Computer Science, Vol. 765, pp. 398-409, Springer-Verlag, 1994.

――――― , "On Matsui's Linear Cryptanalysis," Advances in Cryptology ― Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science, Vol. 950, pp. 341-355, Springer-Verlag, 1995.

――――, "Cryptanalysis of Multiple Modes of Operation," Journal of Cryptology, Vol. 11, No. 1, pp. 45- 58, 1998.

――――, A. Biryukov, N. Ferguson, L. R. Knudsen, B. Schneier, and A. Shamir, "Cryptanalysis of Magenta," August 20, 1998.

――――――, ――――― and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds using Impossible Differentials," Technion ― Computer Science Department ― Technical Report CS0947, October 1998.

――――― and L. R. Knudsen, "Cryptanalysis of the ANSI X9.52 CBCM Mode," Advances in Cryptology ― Proceedings of EUROCRYPT'98, Lecture Notes in Computer Science, Vol. 1403, pp. 100-111, Springer-Verlag, 1998.

――――― and A. Shamir, "Differential cryptanalysis of DES-like Cryptosystems," Journal of Cryptology, Vol. 4, No. 1, pp. 3-72, 1991.

――――― and ――――――, "Differential cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer," Advances in Cryptology ― Proceedings of CRYPTO'91, Lecture Notes in Computer Science, Vol. 576, pp. 156-171, Springer-Verlag, 1992.

――――― and ――――――, "Differential cryptanalysis of the Full 16-Round DES," Advances in Cryptology ― Proceedings of CRYPTO'92, Lecture Notes in Computer Science, Vol. 740, pp. 487-496, Springer-Verlag, 1993.

A. Biryukov and E. Kushilevitz, "Improved Cryptanalysis of RC5," Advanced in Cryptology ― Proceedings of EUROCRYPT'98, Lecture Notes in Computer Science, Vol. 1403, pp. 275-286, Springer-Verlag, 1998.

M. Blaze, W. Diffie, R. L. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Wiener, "Minimum Key Lengths For Symmetric Ciphers To Provide Adequate Commercial Security," A Report By An Ad Hoc Group Of Cryptographers And Computer Scientists, January 1996. (http://www.bsa.org/policy/encryption/cryptographers _c.html)

J. Borst, L. R. Knudsen, and V. Rijmen, "Two Attacks on Reduced IDEA," Advances in Cryptology — Proceedings of EUROCRYPT'97, Lecture Notes in Computer Science, Vol. 1233, pp. 1-13, Springer-Verlag, 1997.

L. Brown, M. Kwan and J. Pieprzyk, "Improving resistance to differential cryptanalysis and the redesign of LOKI," Advances in Cryptology — Proceedings of ASIACRYPT'91, Lecture Notes in Computer Science, Vol. 739, pp. 36-50, Springer-Verlag, 1993.

——— and J. Pieprzyk, "Introducing the new LOKI97 Block Cipher," June 12, 1998. (http://www.adfz.oz.au/~lpb/research/loki97/loki97spec.ps.gz)

———, ———, and J. Seberry, "LOKI — A cryptographic primitive for authentication and secrecy applications," Advances in Cryptology — Proceedings of AUSCRYPT'90, Lecture Notes in Computer Science, Vol. 453, pp. 229-236, Springer-Verlag, 1990.

C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, M. Matyas Jr., L. O'Connor, M. Peyravian, D. Safford, and N. Zunic, "MARS - a candidate cipher for AES," July 10, 1998. (http://www.research.ibm.com/security/mars.html)

D. Coppersmith, C. Holloway, S. M. Matyas, and N. Zunic, "The Data Encryption Standard," Information Security Technical Report, Vol. 2, No.2, pp. 22-24, ZERGO, 1997.

J. Daemen and V. Rijmen, "AES Proposal: Rijndael," June 11, 1998. (http://www.esat.kuleuven.ac.be/rijmen/rijndael/Rijndaeldoc.pdf)

Department of Commerce, National Institute of Standards and Technology, "Announcing Draft Federal Information Processing Standard (FIPS) 46-3, Data Encryption Standard (DES), and Request for Comments," Federal Register, Vol. 64, No. 10, pp. 2625-2628, January 15, 1999.

W. Diffie and M. E. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," Computer, Vol. 10, No. 6, pp. 74-84, 1977.

Electronic Frontier Foundation, Cracking DES, O'Reilly & Associates, 1998.

D. Georgoudis, D. Leroux, and B. S. Chaves, "The 'FROG' Encryption Algorithm," June 15, 1998. (http://www.tecapro.com/aesfrog.htm)

H. Gilbert, M. Girault, P. Hoogvorst, F. Noilhan, T. Pornin, G. Poupard, J. Stern, and S. Vaudenay, "Decorrelated Fast Cipher: an AES Candidate," July 12, 1998.(ftp://ftp.ens.fr/pub/dmi/users/vaudenay/GG+98b.ps)

B. Gladman, "AES Algorithm Efficiency," 1998.(http://www.seven77.demon.co.uk/aes.htm)

C. Harpes, G. G. Cramer, and J. L. Massey, "A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma," Advances in Cryptology — Proceedings of EUROCRYPT'95, Lecture Notes in Computer Science, Vol. 921, pp. 24-38, Springer-Verlag, 1995.

——— and J. L. Massey, "Partitioning Cryptanalysis," Fast Software Encryption'97, Lecture Notes in Computer Science, Vol. 1267, pp. 13-27, Springer-Verlag, 1997.

International Organization for Standardization, “ISO 8731-1 Banking ― Approved algorithms for message authentication ― Part 1: DEA,” 1987.

――――, “ISO 8732 Banking ― Key management (wholesale),” 1988.

――――, “ISO 9564-2 Banking ― Personal Identification Number management and security ― Part 2: Approved algorithm(s) for PIN encipherment,” 1991.

――――― and International Electrotechnical Commission, “ISO/IEC 9979 Information technology― Security techniques ― Procedures for the registration of cryptographic algorithms,” 1991.

――――― and ――――, “ISO/IEC 10118-2 Information technology ― Security techniques ― Hash-functions Part 2: Hash functions using an n-bit block cipher algorithm,” 1994.

――――― and――――, “ISO/IEC 10116 Information technology ― Security techniques ― Modes of operation for an n-bit block cipher,” 1997.

M. J. Jacobson Jr. and K. Huber, “The MAGENTA Block Cipher Algorithm,” June 8, 1998.

T. Jakobsen and L. R. Knudsen, “The Interpolation Attack on Block Cipher,” Fast Software Encryption’97, Lecture Notes in Computer Science, Vol. 1267, pp. 28-40, Springer-Verlag, 1997.

J. Kelsey, B. Schneier, and D. Wagner, “Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple DES,” Advances in Cryptology ― CRYPTO’96, Lecture Notes in Computer Science, Vol. 1109, pp. 237-251, Springer-Verlag, 1996.

L. R. Knudsen, “A Key-schedule Weakness in SAFER K-64,” Advances in Cryptology ― Proceedings of CRYPTO’95, Lecture Notes in Computer Science, Vol. 963, pp. 274-786, Springer-Verlag, 1996.

――――, “DEAL - a 128- bit Block Cipher,” May 15, 1998. (http://www.ii.uib.no/~larsr /aes.html)

―――――― and T. Berson, “Truncated differential of SAFER,” Proceedings of Fast Software Encryption, Third International Workshop, Lecture Notes in Computer Science, Vol. 1039, pp. 15-26, Springer-Verlag, 1996.

K. Kusuda and T. Matsumoto, “Optimization of Time-Memory Trade-Off Cryptanalysis and Its Application to DES, FEAL-32, and Skipjack,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E79-A, No. 1, January 1996.

――――― and ――――, “A Strengh Evaluation of the Data Encryption Standard,” Institute for Monetary and Economic Studies, Bank of Japan, DPS No. 97-E-5, 1997.

X. Lai, “Higher Order Derivatives and Differential Cryptanalysis,” Communications and Cryptography, pp. 227-233, Kluwer Academic Publishers, 1994.

――――, J. L. Massey, and S. Murphy, “Markov ciphers and differential cryptanalysis,” Advances in Cryptology ― Proceedings of EUROCRYPT ’91, Lecture Notes in Computer Science, Vol. 547, pp. 17-38, Springer-Verlag, 1991.

C. Lee and Y. Cha, “The Block Cipher: SNAKE with Provable Resistance against DC and LC attacks,” Proceedings of JW-ISC’97, pp.3-17, 1997.

C. H. Lim, “CRYPTON: A New 128-bit Block Cipher -Specification and Analysis-,” 1998. (http://crypt.future.co.kr/~chlim/pub/cryptonv05.ps)

S. Lucks, “Attacking Triple Encryption,” Proceedings of Fast Software Encryption ’98, Lecture Notes in Computer Science, Vol. 1372, pp.239-253, 1998a.

————, “On the Security of the 128- bit Block Cipher DEAL,” August 20, 1998b.（http://th.informatik.uni-mannheim.de/m/lucks/papers/deal.ps.gz）

S. M. Matyas, C. H. Meyer, and J. Oseas, “Generating strong one-way function with cryptographic algorithm,” IBM Technical Disclosure Bulletin, Vol. 27, pp. 5658-5659, 1985.

J. L. Massey, “SAFER K-64: A byte-oriented block-ciphering algorithm,” Proceedings of Fast Software Encrypiton, Cambridge Security Workshop, Lecture Notes in Computer Science, pp. 1-17, Springer-Verlag, 1994.

————, “SAFER K-64: One year later,” Proceedings of Fast Software Encrypiton, Cambridge Security Workshop, Lecture Notes in Computer Science, Vol. 1008, pp. 212-241, Springer-Verlag, 1995.

————, G. H.Khachatrian, and M. K. Kuregian, “Nomination of SAFER+ as Candidate Algorithm for the Advanced Encryption Standard（AES),” June 12, 1998.（http://www.cylink.com/internet/objects.nsf/reference/safer/$file/safer.pdf）

M. Matsui, “Linear Cryptanalysis Method for DES Cipher,” Advances in Cryptology — Proceedings of EUROCRYPT '93, Lecture Notes in Computer Science, Vol. 765, pp. 386-397, Springer-Verlag, 1994a.

————, “The first experimental cryptanalysis of the Data Encryption Standard,” Advances in Cryptology — Proceedings of CRYPTO ’94, Lecture Notes in Computer Science, Vol. 839, pp. 1-11, Springer-Verlag, 1994b.

R. C. Merkle and M. Hellman, “On the Security of Multiple Encryption,” Communications of the ACM, Vol. 24, No. 7, pp. 465-467, 1981.

S. Miyaguchi, A. Shiraishi, and A. Shimizu, “Fast data encipherment algorithm FEAL-8,” Review of Electrical Communication Laboratories, Vol. 36, No. 4, pp.321-327, NTT, 1988.

National Institute of Standards and Technology, “Data Encryption Standard（DES）,” Federal Information Processing Standards Publicatio（iFIPS PUB）46-2, December 13, 1993.

————, “Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard（AES）,” September 12, 1997.（http://csrc.nist.gov/encryption/aes/aes_9709.htm）

————, “SKIPJACK and KEA Algorithm Specifications,” May 29, 1998a.（http://csrc.nist.gov/encryption/skipjack-1.pdf）

————, “AES — The First Advanced Encryption Standard Candidate Conference,” The Proceedings of The First Advanced Encryption Standard Candidate Conference, August 20, 1998b.

Nippon Telegraph and Telephone Corporation, “Specification of E2 - a 128-bit Block Cipher, ” June 14, 1998.（http://info.isl.ntt.co.jp/e2/E2spec.pdf）

K. Nyberg, “Linear Approximation of Block Ciphers,” Advances in Cryptology — Proceedings of EUROCRYPT’94, Lecture Notes in Computer Science, Vol. 950, pp. 439-444, Springer-Verlag, 1995.

K. Ohta and K. Aoki, “Linear cryptanalysis of the Fast Data Encipherment Algorithm,” Advances in Cryptology ― Proceedings of CRYPTO’94, Lecture Notes in Computer Science, Vol. 839, pp. 12-16, Springer-Verlag, 1994.

P. C. van Oorschot and M. J. Wiener, “A known plaintext attack on two-key triple encryption,”Advances in Cryptology ― Proceedings of EUROCRYPT ’90, Lecture Notes in Computer Science, Vol. 473, pp. 318-325, Springer-Verlag, 1990.

V. Rijmen and L. R. Knudsen, “Weaknesses in LOKI97,” June 15, 1998.（ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen/loki97.ps.gz）

R. L. Rivest, “The RC5 encryption algorithm,” Fast Software Encryption, Second Internationl Workshop, Lecture Notes in Computer Science, Vol. 1008, pp. 86-96, Springer-Verlag, 1995.

――――, M. J. B. Robshaw, R. Sidney, and Y. L. Yin, “The RC6 Block Cipher,” 1998.（http://www.theory.lcs.mit.edu/~rivest/rc6.ps）

RSA Laboratories, “RC6 Statements,” 1998.

M. J. Saarinen, “Equivalent keys in MARS,” August 18, 1998.（http://www.math.jyu.fi/~mjos/mars-eqk.ps）

B. Schneier, J. Kelsey, D. Whiting, D. Wagner, and C. Hall, “Performance Comparison of the AES Submissions,” Version 1.4a, January 5, 1999.（http://www.counterpane.com/aes-performance.pdf）

――――, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Fergusen, “Twofish: A 128-bit Block Cipher,” June 15, 1998.（http://www.counterpane.com/twofish.ps.zip）

R. Schroeppel, “The Hasty Pudding Cipher Specification,” June 1998a.（http://www.cs.arizona.edu/~rcs/hpc/hpc-spec）

――――, “The Hasty Pudding Cipher: Specific NIST Requirements,” June 1998b.（http://www.cs.arizona.edu/~rcs/hpc/hpc-nist-doc）

M. Smid and M. Dworkin, “Special Report on the First AES Conference,” August 1998.（http://csrc.nist.gov/encryption/aes/round1/crypto98.pdf）

T. Tokita, T. Sorimachi, and M. Matsui, “Linear cryptanalysis of LOKI and s^2DES,” Advances in Cryptology ― Proceedings of ASIACRYPT ’94, Lecture Notes in Computer Science, Vol. 917, pp. 293-303, Springer-Verlag, 1995.

W. Tuchman, “Hellman presents no shortcut method to the DES,” IEEE Spectrum, Vol. 6, No. 7, pp. 40-41, Springer-Verlag, July 1979.

S. Vaudenay, “Provable Security for Block Ciphers by Decorrelation,” Lectures Notes in Computer Science, Vol. 1373, pp. 249-275, Springer-Verlag, 1998.

D. Wagner, N. Ferguson, and B. Schneier, “Cryptanalysis of FROG,” August 15, 1998.（http://www.counterpane.com/frog.pdf.zip）

M. Wiener, “Efficient DES Key Search,” A Rump Session Talk at CRYPTO ’93, 1993.