社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＣＳＥＣ－38（33）
2007／7／20

セキュリティパラメータを変更可能なブロック暗号の構成法

平井 康雅† 松尾真一郎† 尾形わかは††

† NTT データ 〒135-8671 東京都江東区豊洲 3-3-9
†† 東京工業大学 〒152-8550 東京都目黒区大岡山 2-12-1
E-mail: †{hiraiys,matsuosn}@nttdata.co.jp, ††wakaha@mot.titech.ac.jp

あらまし 一般的に，暗号技術が危殆化した場合には，それらを利用するシステムでは，求められる安全性要件を満たすために，より安全な新しい暗号技術への移行が求められる．しかしながら，コスト等を考慮した場合，システム更改時期以外での暗号技術の移行は困難である．本稿では，既存のブロック暗号が危殆化した際に，容易に安全性を高めることが可能となるよう，ブロック暗号で用いる鍵長およびブロックサイズをパラメータとし，変更可能な方式の構成例を示す．

キーワード システムセキュリティ，組合わせ暗号，ブロック暗号，危殆化

On Construction of Block Cipher with Variable Security Parameter

Yasumasa HIRAI†, Shin'ichiro MATSUO†, and Wakaha OGATA††

† NTT DATA Corporation., 3-9, Toyosu 3-chome, Koto-ku, Tokyo 135-8671, JAPAN
†† Tokyo Institute of Technology., 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, JAPAN
E-mail: †{hiraiys,matsuosn}@nttdata.co.jp, ††wakaha@mot.titech.ac.jp

Abstract Currently, cryptographical technology such as an encryption algorithm is used in many IT systems. When an encryption scheme becomes insecure in computational sense, to fill the security requirement, we need to replace all IT systems which use the algorithm. Considering the cost, however, the replacement of IT systems outside the schedule is very difficult. In this research, we introduce the variable key length and the variable block size as "security parameters" of the block cipher, and show some examples of block ciphers with security parameters. By using such block ciphers, we can easily enhance the security of IT systems without exchanging the algorithm, even if the block cipher becomes computationally insecure.

Key words system security, n times encryption, block cipher, algorithm compromise

1. はじめに

多くの情報システムでは，システムに求められる安全性要件を満たすための手段として暗号技術が広く利用されている．そのため，情報システムの安全性は，利用されている暗号技術の安全性に依存していることも少なくない．しかしながら，暗号技術は，いつまでも安全であるというわけではなく，共通鍵暗号 DES [1] の解読の成功やハッシュ関数 SHA-1 への攻撃の成功等のように，計算機能力の向上や新たな攻撃方法の発見等により危殆化することが考えられる．情報システムでは，利用されている暗号技術が危殆化した際に，システムの安全性要件を満たすために，より安全な暗号技術に速やかに移行する必要がある．しかしながら，実社会における情報システムでは，暗号技術が危殆化したからといっても直ちに新しい暗号技術への移行を実施することは困難である．これは，スマートカードのように，新たなアルゴリズムの実装に時間を要するハードウェアが用いられているシステムについてはもちろんのこと，ソフトウェアで実現されているシステムについても同様に困難である．情報システムでは，代替の暗号アルゴリズムが存在する場合であっても，それらのアルゴリズムに利用している製品が対応していないために，対策を実施できない場合がある．また，開発者により独自に暗号アルゴリズムを実装し，システムのバージョンアップが実施可能であるとしても，予め設定されたシステムライフサイクルのシステム更改時期以外での更改は，システム開発コストの問題により実施することが非常に困難である．

RSA 暗号のような公開鍵暗号では，計算機能力の向上により暗号技術が危殆化した際にも，暗号アルゴリズムそのものについては変更することなく，セキュリティパラメータとなる鍵長を変更することにより，安全性を高めることができる．そのため，公開鍵暗号を利用した多くの情報システムでは，より安

全な鍵長への移行を考慮し，鍵長を可変のパラメータとして容易に変更可能となるよう設計されている場合が多い．一方，ブロック暗号やハッシュ関数では，任意のセキュリティパラメータに変更することはできないため，危殆化した際には，システムを再構築する必要が生じる．そのため，以前に DES(Data Encryption Standard) が危殆化した際には，これに対応する手法として，DES を 3 度繰り返し処理を行う組み合わせ暗号である Triple DES (T-DES) が提案され，現在も既存の情報システムでは広く利用されている．現在，標準となっている，AES (Advanced Encryption Standard) では，3 種類の鍵長を選択可能となっているが，危殆化による新たな暗号アルゴリズムへの移行については考慮しておかなければならない．

そこで，本稿では，システム更改時期や開発コスト等，システムライフサイクルを考慮した際に，直ちに新たな暗号アルゴリズムへの移行が困難な状況の場合においても暗号技術の危殆化への対応を可能とする，セキュリティパラメータを変更可能なブロック暗号に関する考え方および構成例について示す．

2. ブロック暗号の安全性

ブロック暗号の安全性は，攻撃を実行するために必要なコスト（時間，使用メモリ，計算量など）によって評価できる．ブロック暗号に対する攻撃は，ブロック暗号の内部設計によらず不可避である Brute force method と，設計に大きく依存する Short cut method に大別できる．

2.1 Brute Force Method

a） 鍵全数探索

入手した平文，暗号文の組に対して，候補となる全ての鍵で暗号化することにより，秘密鍵を探索する攻撃法である．鍵全数探索攻撃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向上する．鍵長が n bit の場合，候補となる鍵の個数は 2^n 個となる．そのため，平均して 2^{n-1} 回程度の暗号化処理によって秘密鍵を発見可能となる．

b） タイムメモリトレードオフ法

送信されると予想できる平文に対する暗号文および鍵を示した索引表を作成し，暗号文を入手した後に短時間で鍵の検索を可能とする攻撃法である．予め作成しておく索引表が大きいほど鍵探索の時間を短縮することが可能となる．また，一度，索引表を作成すれば、鍵が変更されたとしても予め想定した平文に対する暗号文を入手可能である限り適用できる．一般的に，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向上する．

c） 暗号文一致攻撃

「n bit のサイズのデータをランダムに $2^{\frac{n}{2}}$ 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およそ $\frac{1}{2}$ になる」という性質を利用した攻撃法である．例えば，ブロック長が 64 bit のブロック暗号であれば，2^{32} 個程度の暗号文を集めると，およそ $\frac{1}{2}$ の確率で同じ暗号文を見つけることができる．一般的に，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向上する．

d） 辞書攻撃

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，記録しておき，それを辞書のように利用することにより，盗聴等によって入手した暗号文に対応する平文を得る攻撃法である．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，暗号文ペアを記録可能な容量をもつ記憶媒体が必要となる．一般的に，ブロック長が長いほど，辞書攻撃に対する安全性は向上する．

2.2 Short Cut Method

a） 差分解読法

Biham, Shamir によって，1991 年に発表された攻撃法である [3]．ある特定の差分を有するような平文ペアに対し，特定の差分を有する暗号文のペアが生じる確率が高い場合，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ことが可能となる．

b） 線形攻撃法

松井によって，1992 年に発表された攻撃法である [9]．平文と暗号文の bit 値の間に線形関係が発生する確率が $\frac{1}{2}$ から乖離している場合、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ことが可能となる．

c） 中間一致攻撃

T-DES 等の組み合わせ暗号に対して有効な攻撃法である [10]．例えば，ある平文に対し，3 つの異なる鍵 Key_1, Key_2, Key_3 のそれぞれを順に用いて暗号処理を行い，出力された暗号文を所持しているとする．この場合，平文に対して，Key_1，Key_2 の全ての候補を用いて暗号化を行い，中間値 a を計算する．また，暗号文に対して Key_3 の全ての候補を用いて復号処理を行い，中間値 b を計算し，$a = b$ となる Key_1, Key_2, Key_3 を見つけ出す攻撃法である．

また，高階差分攻撃，補間攻撃，分割攻撃，関連鍵攻撃といった Short cut 攻撃に対する安全性も必要とされている．

2.3 鍵長と安全性

Brute force method の攻撃コストは鍵長やブロック長に依存しており，設計を工夫しても避けることができない．例えば，最も典型的な Brute force method である鍵の総当り攻撃においては，鍵長を n とすると最悪でも 2^n の計算コストで鍵を見つけることができる．一方，Short cut method に必要なコストは設計に大きく左右される．このため，ブロック暗号の設計においては，全ての既知の Short cut method に対する攻撃コストが，鍵全数探索攻撃に必要なコスト 2^n を下回らないことが重要視されている．たとえば，128 bit 鍵を用いるブロック暗号においては，全ての Short cut method の攻撃コストが 2^{128} 以上となるときに「安全である」と言われる．

3. 関連研究

本章では，いくつかのブロック暗号の構成について紹介する．伝統的なブロック暗号として DES の構成を，鍵長を選択できるものとして AES の構成を示す．また，既存のブロック暗号を用いる方式として，T-DES と DEAL について示す．

3.1 DES

ブロック暗号の最初の規格であり，1977年に米国の政府調達基準に示された標準暗号となった[1]．SSLをはじめとして，広く情報システムで利用されていたが，鍵長が56 bitのため現在では安全ではない．現在では主に下記で示すT-DESとして利用されている．ブロック長は64 bitである．

このブロック暗号はFeistel型（図1）と呼ばれる繰り返し構造を持つ．各ラウンドのF関数が用いるラウンド鍵 k_i は，鍵スケジュール部によって生成される．

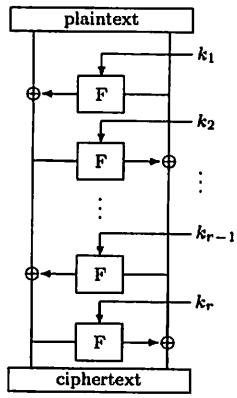

図1 r roundのFeistel構造

3.2 AES

DESの後継となった暗号であり，2000年にRijndael方式[5]が選定された．128 bit，192 bit，256 bitの3つの鍵長を選べる．ブロック長は128 bitである．

構造は，複数のS-boxからなる換字処理部と，転置処理部の繰り返しによるSPN構造（図2）をしており，鍵長によって，繰り返し回数も異なる．

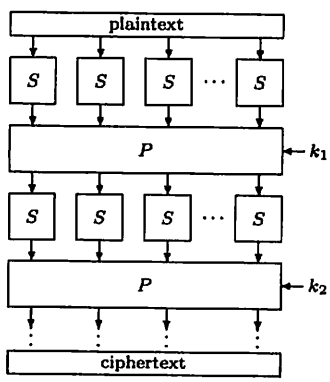

図2 SPN構造

3.3 T-DES (triple-DES)

T-DESは，1979年にTuchmanによって提案された，既存のブロック暗号DESを，暗号処理，復号処理，暗号処理の順に，繰り返し行う方式である[11]．T-DESには，2つの鍵 (k_1, k_2) を利用する2key T-DESと，3つの鍵 (k_1, k_2, k_3) を利用する3key T-DESがある．ただし k_i はそれぞれ独立ランダムに選ばれたDESの鍵（56 bit）である．（図3を参照．）

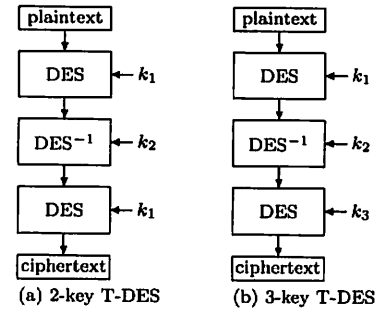

図3 T-DESの構成

T-DESは，広く利用されていたDESアルゴリズムを利用しつつ安全性を向上させることができるため，多くの情報システムで使用されている．しかしながら，いくつかの欠点が指摘されているため，AESへの移行が求められている．

まず，T-DESのような組み合わせ暗号では，中間一致攻撃が可能となる．そのため，3-key T-DESの鍵長は168 bitであるにもかかわらず，2^{56} の選択平文と 2^{108} 程度の計算で鍵の探索が可能であることが知られている[7]．

また，鍵の総当り攻撃による攻撃コストはDESより大きくなっているが，ブロック長はDESと同じ64 bitであり，辞書攻撃等に対する安全性は向上していない．これに対して，ブロック長をのばすためのいくつかの手法が提案されているが，安全でないことが示されている[2]．

効率の面では，T-DESは1ブロック(64 bit)当たり，DESを3回実行する必要がある(鍵スケジューリング含む)ため，暗号化・復号に時間がかかるという欠点を持つ．

3.4 DEAL

AES選定プロジェクトにおいて，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．

3.4.1 DEALの構成

DEALは，Feistel構造を有しており，F関数としてDESのアルゴリズムを使用する[6]（図4）．また，各ラウンドにおけるF関数（DES）の鍵 RK_i を生成するための鍵スケジューリング部においても，DESのアルゴリズムが使用される．そのため，T-DESと同様に，既にDESが実装されている環境からの移行を容易にでき，T-DESと同程度の処理速度での実装を可能としている．

鍵長が128 bit，192 bitの場合は6段以上，256 bitの場合は8段以上が推奨とされている．

3.4.2 DEALの安全性

DEALは，提案者により，差分解読法に対して，ラウンド関数が6段の場合に 2^{70} 個以上の選択平文と 2^{121} 回のDESの暗号化処理により解読可能なことが示されている．また，Lucksは[8]にて，鍵長192 bit，ラウンド関数6段のDEALが，2^{33} 個の選択平文を入手可能であれば，2^{145} 回のDESの暗号化処

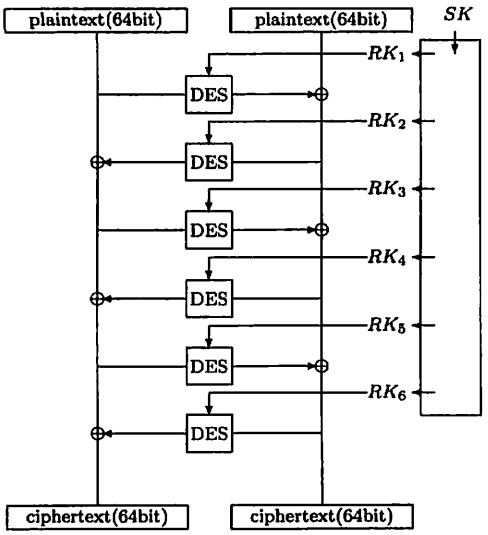

図 4　128 ビット，192 ビット DEAL の構成

理によって解読可能であることが示されている．これは，鍵全数探索に必要な計算量 2^{192} に比べて大幅に少ない．

4. ブロック暗号の安全性の考え方

実社会における情報システムにはシステム更改時期が予め設定され，システムはライフサイクルに従って運用されている．更改時期以外でのシステム更新には想定外のコストが必要となり，容易に更新が行えないことが多い．そのため，情報システムへ組み込むアルゴリズムの選択にあたっては，システムライフサイクルと将来の危殆化を十分考慮した上で行う必要があるが，一方で，新たな攻撃方法の発見等，予測できない技術の向上により，システム更改時期よりも前に暗号技術が危殆化する可能性もある．その場合は新規アルゴリズムへの移行のための多大なコストを負担するか，危殆化したアルゴリズムを更新時期まで使用し続けるかの二者択一となる．

先に示した T-DES は，このような現状を解決する 1 つの方法であった．すなわち，DES というアルゴリズムは変更することなく，比較的小さいシステム変更により安全性の向上に成功している．

本稿では，T-DES の考え方を広く一般化し，ブロック暗号に対して，変更可能なセキュリティパラメータを導入することを提案する．具体的には，鍵長とブロック長をブロック暗号に対する 2 つのセキュリティパラメータと考え，鍵長およびブロック長が可変であるブロック暗号の構成を検討する．

変更可能なセキュリティパラメータをブロック暗号に導入するにあたり，ブロック暗号の安全性および鍵長に関して以下の観点を考慮する．現在のブロック暗号は，鍵長と安全性が一致するように設計されている．すなわち，鍵長が n bit の場合，既知の攻撃手法に必要な計算コストが 2^n を上回るように設計されている．しかし，求められる安全性に応じた適切な 2 つのセキュリティパラメータを設定可能であれば，鍵長と安全性は必ずしも等しくなくてもよいと考える．このように考える理由は，主に以下の 2 つである．

- 情報システムでは，システムが満たすべき安全性を 2^s とした場合，最も効率の良い攻撃に必要となる計算コストが 2^s を上回る暗号技術を使用する必要があり，鍵長等のセキュリティパラメータは，2^s の安全性を満たすよう設定される．現在のブロック暗号では，最も効率の良い攻撃が鍵全数探索となるよう設計されているため，安全性と鍵長が一致している．しかしながら，最も効率の良い攻撃は，必ずしも鍵全数探索とする必要はなく，既存の攻撃のいずれかで，最も効率的な攻撃に必要となる計算コストが 2^s を上回るように設計され，これを満たす適切なセキュリティパラメータが設定されていればよい．
- 一般に，鍵長の増加により処理効率の低下といった問題が生じるが，処理効率の低下がサービス実行上，許容できる場合，鍵管理コストの増加のみが問題となる．ここで問題となる鍵管理コストは，鍵長に応じて増大する鍵を記録する媒体の容量である．記録媒体の大容量化が進んでいる現在，このことが情報システムに与える影響は，無視できるほど小さいと言える．例えば公開鍵暗号では，実社会で利用するために安全な鍵長は，現在 2048 bit とされている等，かなり長い鍵長が必要とされているが，スマートカード実装など特殊な場合を除き，特に問題視されていない．例えば，ブロック暗号において，現在用いられている鍵スケジューリング等による各ラウンド数に割り当てられる拡大鍵の生成を行わず，独立した拡大鍵からより長い鍵長を設定したとしても，実システムを実現する上で問題は生じないことを意味している．

5. 構 成 例

本章では，鍵長およびブロック長を変更可能なセキュリティパラメータとするブロック暗号の構成例を示す．まず，構成法の 1 つとして，例えば T-DES のように，既存のブロック暗号を組み合わせることにより安全性を高めるモードを用いる手法が考えられる．また，既存のブロック暗号を組み合わせて安全性を高めるのではなく，暗号アルゴリズムを設計する段階で，セキュリティパラメータを有するよう設計する手法が考えられる．

以下で，前者についての構成例を示す．また，後者については，既存のブロック暗号の構成を基に，鍵長およびブロック長を可変とする方式を実現するために必要となる構成の変更について考察を行う．

5.1 モ ー ド

本節では，T-DES，DEAL のように，既存のブロック暗号を複数回組み合わせ，異なる鍵で処理することにより，総当り攻撃に対する安全性を向上させる手法の構成例について示す．ここでは，ブロック暗号が鍵全数探索に対してのみ危殆化した場合を仮定しており，既存のシステムで利用されているブロック暗号を再利用することで，容易に危殆化対策を実施できるようにすることを目的としている．また，一般 Feistel 構造を用いることにより，T-DES，DEAL では実現されていないブロック長の変更を可能とした．

本構成法では，ブロック長が m ビットであるブロック暗号

を元に，ブロック長が $M(=l\times m)$ ビットとなる組み合わせ暗号を構成する．暗号化処理において，平文ブロックは l 個の m ビットのサブブロックに分割され，ブロック暗号の処理を r 回施される（図5）．ラウンド数 r はサブブロック数 l の整数倍であり，$r=l\times x$ $(x=2,3,\ldots)$ となる．ブロック長を変更しない $M=m$ の場合，T-DES と同様の繰り返しの構成となる．

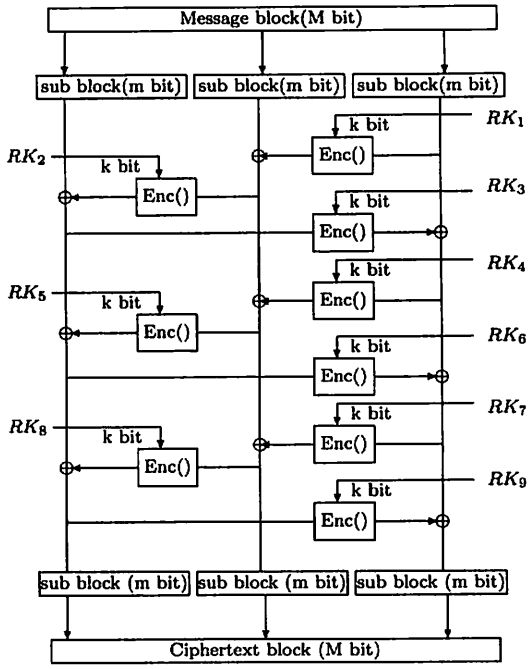

図5　既存ブロック暗号組み合わせによる構成例

r 回の暗号化処理で用いられる鍵（RK_i）は，F 関数として用いるブロック暗号の鍵長と等しい k bit であり，全て独立に r 個生成されるものとする．そのため，本構成法の共通鍵の鍵長 K は，$K=k\times r$ となる．

利用者は，まず，サービス要件より，満たすべき安全性（s bit 安全）を決定する．満たすべき安全性は，現在の計算機能力および今後の計算機能力の向上の予想と，サービスの保証期間を考慮し，最も有効な攻撃に必要となる攻撃コストより決定されるものとする．次に，決定した安全性より，実現するために必要な鍵長 K およびブロック長 $M(=m\times l)$ を選択し，対応したラウンド数 r が決定する．

本構成法で，n bit の安全性を有するブロック暗号を用いた場合，中間一致攻撃の計算コストは $2^{\frac{n}{2}r}$ 程度であると考えられる．これは，中間までの暗号化処理に鍵の半分のビットしか用いないためである．そこで，F 関数として用いる r 個のブロック暗号の鍵を全て独立にするのではなく，1 つの K bit の共通鍵を元に鍵拡張関数を用いて r 個の鍵を生成することによって，中間一致攻撃に対する耐性を高めることができると考えられる．

5.2　ブロック暗号の拡張に関する考察

本節では，ブロック暗号の構造を汎用化し，鍵長およびブロック長を可変とする手法について検討を行う．現在，AES として採用されている Rijndael では，AES の仕様では求められていないために利用されていないが，3 種類の鍵長を扱えることに加え，128 bit から 256 bit までの可変のブロック長を扱うことができる．また，AES 選定プロジェクトに提案された，RC6 [13] は鍵長，ブロック長共にが可変となるよう設計されている(注1)．RC6 は，安全性に関する問題点は特に指摘されておらず，最終候補にも選定されており，可変のパラメータが導入されている安全なアルゴリズムであると言えるが，スマートカードや携帯端末などの小型デバイスでは低速である．また，RC6 の構造そのものに問題が生じた際にも対応できるよう，Feistel 構造，SPN 構造といった構造についても，鍵長およびブロック長が可変となるような構成法を，検討しておくことが望ましい．

そこで我々は，SPN 構造および Feistel 構造を有するブロック暗号について，それら鍵長およびブロック長を可変とするための拡張手法についての考察を行う．

5.2.1　SPN 構造に関する考察

本節では，利用者が，サービス要件に適した安全性を選択可能とするために，SPN 構造を用いたアルゴリズムを汎用化し，任意の鍵長およびブロック長の設定可能とするために考慮すべき点について考察を行う．

a）ラウンド数

求められる安全性を実現するための適切なラウンド数を明らかにする必要がある．また，検討にあたっては，鍵長およびブロック長の両方のパラメータとの関係について明らかにする必要がある．

b）ブロック長

SPN 構造において，任意のブロック長を実現するためには，換字処理部 (S) および転置処理部 (P) を拡張する必要がある．

- 換字処理

換字処理の拡張には，2 つの手法が考えられる．まず，1 つ目として，処理するブロック長が増加した場合，S-box には変更を加えず，1 つの換字処理部での S-box の数を増加させる手法が考えられる．この場合，S-box そのものを拡張する必要がないため，拡張は比較的容易であると考える．

2 つ目の手法として，1 つの換字処理部での S-box の数を変更せずに，S-box を，ビット長が可変の入力を処理できるように拡張する．この場合，S-box の汎用化が必要となるため，前者に比べ，実現が困難であると考える．

- 転置処理

ブロック長の増加に伴い，転置処理部についても，可変の入出力に対応した汎用的なものを検討する必要がある．

c）鍵　　長

鍵長は，鍵全数探索に対する安全性を考慮し，求められる安全性を上回るよう設定しなければならない．鍵拡張を使用する場合と使用しない場合について，考慮すべき点を以下に示す．

- 鍵拡張を使用しない場合

(注1)：AES プロジェクトでは，鍵長，ブロック長が可変のアルゴリズムとして HPC，FROG が提案されているが，安全性に関する問題点が指摘されている．

鍵拡張を使用しない場合，鍵長は，ラウンド数と1つのラウンド鍵のビット長から決まる．そのため，与えられた安全性を満たすためのラウンド数の導出のみが必要となる．

- 鍵拡張を使用する場合

求められる安全性を保障するラウンド数に必要な拡大鍵を生成できるように鍵拡張を汎用化する必要がある．一般的に，ブロック暗号で用いられる鍵拡張では，繰り返しにより拡大鍵が生成されている．そのため，汎用化は比較的容易であると考えるが，汎用化にあたり，以下を考慮する必要がある．

まず，鍵拡張を使用しない場合と同様に，求められる安全性を満たすために必要なラウンド数を導出する必要がある．次に，ラウンド数およびブロック長の変更に対応するために，(必要なラウンド数) ×（ビット長）を満たす任意のビット長の出力を可能となるように拡張部を構成する必要がある（図6）．

また，拡張部において，必要なビット長の拡大鍵を安全に出力するために，出力と必要となる入力（共通鍵）のビット長との関係を導出する必要がある．これにより，共通鍵のビット長を決定され，共通鍵のビット長と満たされる安全性の対応関係を明確にすることができる．

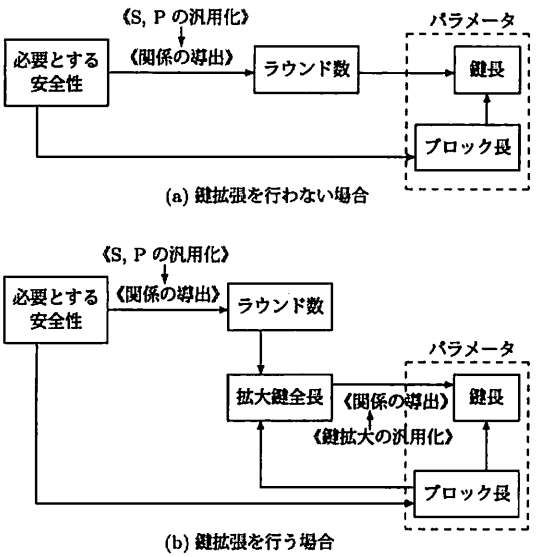

図6 求められる安全性とパラメータの関係

5.2.2 Feistel 構造に関する考察

Feistel 構造を用いたアルゴリズムに対してセキュリティパラメータを導入する場合も，ブロック長に以外については，SPN 構造の場合と同様に考えることができる．

a) ブロック長

Feistel 構造において，任意のブロック長を実現するためには，Feistel 構造あるいは F 関数を拡張する必要がある．

- 一般 Feistel 構造の適用

5.1 節で示したモードと同様に，一般 Feistel 構造を用いることにより，F 関数の入出力のビット長を変更せず，サブブロック数を増やしていくことにより，任意のブロック長を扱う手法が考えられる．

- F 関数の汎用化

Feistel 構造において，任意のブロック長を扱うためには，ラウンド関数として利用される F 関数の構成を汎用化し，サブブロック数を変更せず，入出力のビット長を可変にすることが考えられる．

6. ま と め

本稿では，計算機能力の向上によるブロック暗号の危殆化に対し，システムライフサイクルへの影響を考慮し，情報システムの安全性を向上させるために，鍵長およびブロック長を可変とする構成法について考察を行った．まず，システムで利用されている既存のブロック暗号を用いて安全性を向上させる手法について，その構成例を示した．次に，新規にブロック暗号を設計する際に，鍵長およびブロック長が可変となるよう構成する上で考察すべき点について考察を行った．今後は，構成例の安全性および効率に関する評価を行う．また，既存のブロック暗号を基に拡張方式についての検討を行う．

文 献

[1] American National Standards Institute, "X3.92 — 1981, Data Encryption Algorithm," 1981.

[2] E. Biham, "Cryptanalysis of Multiple Modes of Operation," Journal of Cryptology, Vol. 11, No. 1,1998, pp. 45-58.

[3] E. Biham and A. Shamir, "Differential Cryptanalysis of the Data Encryption Standard," Springer-Verlag, 1993. (The extended abstruct appeared at CRYPTO'90 and Journal of Cryptology, Vol.4, No.1, 1991)

[4] D. Coppersmith, D. B. Johnson, and S. M. Matyas, "A proposed Mode for Triple-DES Encryption," IBM Journal of Research and Development, Vol. 40, No. 2, 1996, pp. 253-262.

[5] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," June 11, 1998.

[6] L. R. Knudsen, "DEAL - a 128-bit Block Cipher," May 15, 1998.

[7] S. Lucks, "Attacking Triple Encryption," Proceedings of Fast Software Encryption '98, Lecture Notes in Computer Science, Vol. 1372, 1998, pp.239-253.

[8] S. Lucks, "On the Security of the 128-Bit Block Cipher DEAL," August 20, 1998.

[9] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," Advances in Cryptology - Proceedings of EUROCRYPT '93, Lecture Notes in Computer Science, Vol. 765, Springer-Verlag, 1994, pp. 386-397.

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

[11] NIST, "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher", SP 800-67

[12] 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, Springer-Verlag, 1990, pp. 318-325.

[13] RSA Laboratories, "RC6 Statements," 1998.

[14] A. Shamir, "Differential Cryptanalysis of the Full 16-Round DES, " Advances in Cryptology - Proceedings of CRYPTO'92, Lecture Notes in Computer Science, Vol. 740, Springer-Verlag, 1993, pp. 487-496.