共通鍵ブロック暗号の変遷

4 班　植竹聡　岡山麻裕子　田中永

アドバイザ教員　片岸一起　繆瑩

2005 年 10 月 25 日

概 要

高度情報化社会の進展に伴い，いたるところで個人情報が扱われており，私たち自身が，自分の情報がどのように扱われ，どのようなセキュリティ対策が行われているかについての知識を持つ必要が出てきている．そこで我々のグループでは共通鍵暗号の概観と現在広く使用されている DES と AES について説明する．

序論

1　背景と目的

近年，フィッシング詐欺と呼ばれる行為やスパイウェア等による個人情報の盗取など情報セキュリティに関する話題が連日のように報告されている．また，偽造キャッシュカードによる被害や，不正アクセス等による企業の機密情報の漏洩など，高度なハイテク犯罪による被害が頻繁に発生・拡大し，IT の問題が社会に損害を及ぼす危険性が飛躍的に大きくなっている [1], [2].

これらの行為は非常に巧妙に行われるため，企業や組織としての対応だけでなく私たち利用者が主体的に対応しなければ，被害の予防や拡大防止はできない．

このような背景からも，情報セキュリティを巡るリスクが増大する現代社会において，専門家でない私たちも，情報セキュリティに関する基礎的な知識を持つことは重要であると考えられる．

現代暗号の概観と共通鍵暗号の基礎

2　情報セキュリティとは

「情報セキュリティ」とは，機密情報の漏洩，情報の偽造，不正利用などを防止し，情報の安全性・信頼性を確保することである [3]．具体的には，機密性・完全性・可用性の 3 つの観点から，情報を保護することを意味する．

このような情報セキュリティを確保するための技術の基本となるものが「暗号技術」である．

3　暗号とは何か

暗号とは，当事者以外には意味がわからないように，当事者間でのみ理解できるように取り決めた，特殊な記号や文字，またはその手順や方式のことである．送信者が送りたい元の文章を「平文」，暗号文に変換する作業のことを「暗号化」，受信者が暗号文を元の文章に戻すことを「復号」と言い，暗号化や復号化を行うための手順を「アルゴリズム」，暗号化に用いる数十〜数百桁の数値列からなるパラメータのことを「鍵」と呼ぶ．

暗号において利用される鍵の長さ（鍵長）はアルゴリ

ズムにより様々である．この鍵長が短いと，可能性のあるすべての鍵を試してみるという攻撃方法が可能となる．このため，暗号解析技術の進歩に対抗して暗号の強度を維持するために，今後はより長い鍵を利用する暗号アルゴリズムが使われるようになると考えられる．

3.1 共通鍵暗号と公開鍵暗号

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．公開鍵暗号は，暗号化と復号に異なる鍵を用いる暗号であり，復号に用いる鍵を秘密にする一方，暗号化に用いる鍵を公開することができる．

共通鍵暗号と公開鍵暗号の特徴を比較したものは表 1 のとおりである．

これらの特徴により，膨大なデータの処理が要求されるデータの暗号化には共通鍵暗号が利用され，デジタル署名や共通鍵暗号に用いられる鍵の配送には公開鍵暗号が利用されるケースが多い [4].

3.2 暗号アルゴリズムの分類

暗号アルゴリズムを利用目的と鍵の配置の仕方により分類し，代表例を整理したものが次の表 2 である．

今回の調査では共通鍵暗号方式に着目し，現時点で最新の標準暗号 AES 及び従来の標準暗号 DES から Triple DES の変遷などに関してグループでの検討を行った．本章では，共通鍵暗号に関する基本的事項について調査を行う．

3.3 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．

1. ストリーム暗号
   ストリーム暗号では，鍵から生成された擬似乱数系列（鍵系列）と平文とのビット毎の排他的論理和をとることで暗号化が行われる．復号側では暗号文と鍵系列の同期の取り方により，同期式 (synchronous) ストリーム暗号と自己同期式 (self-synchronizing) ストリーム暗号に分類できる．軍事用に利用されているものが多いため，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．

2. ブロック暗号
   ブロック暗号では，一定の長さ（64 ビット，128 ビット等）のブロック単位で暗号化が行われる．この処理単位をブロック長という．ブロック長が n ビットのブロック暗号を n ビットブロック暗号と呼ぶ．n ビットブロック暗号では，n ビットの平文ブロックから n ビットの暗号文ブロックへ変換される．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．このことから，以下ブロック暗号について説明する．

3.4 ブロック暗号の構造

ブロック暗号は，データスクランブル部と鍵スケジュール部の 2 つを主な要素として構成される．

3.4.1 鍵スケジュール部

鍵スケジュール部は，入力された秘密鍵からデータスクランブル部での変換に必要な拡大鍵を生成する部分である．

3.4.2 データスクランブル部

データスクランブル部は，鍵スケジュール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．

ほとんどのものが複数の関数（繰り返し関数）を重ねた繰り返し構造を採用している．繰り返す関数を F 関数やラウンド関数と呼び，繰り返す回数をラウンド数という．ラウンド関数がすべて同じ関数の暗号もあれば，異なる関数を重ねる暗号もある．鍵スケジュール部で生成された拡大鍵は，各ラウンド関数に入力される．

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．これらの構造以外にも，様々なタイプのラウンド関数が考案されている．

DESの解読方法と現在の動向

4　DES暗号とは

DES（Data Encryption Standard）はアメリカ合衆国商務省標準局（National Bureau of Standard, NIST）が1977年に連邦政府関係の一般コンピュータデータ用標準暗号として制定した事に始まる．一般商用にも使うことを推奨されていたため，多くの組織がDESを組織内標準に定めており，最も良く使われている暗号アルゴリズムである．しかしながら，コンピュータの発達により解読が可能になってきたため，NISTは次期暗号標準AES（Advanced Encryption Standard）をRijndael暗号に定めた．AESが普及するまでの期間のためにDESを3回用いるT-DESも定めている．

5　DESが破られた経緯

1990年代からDES解読の研究がさかんに行われ，数多く発表されるようになってきた．そのなかで有力な方法として差分解読と線形解読がある．差分解読はBihamとShamirが1989年に発表した選択平文アタックである．特殊な平文（差分が0となるもの）の入力とSボックスの入出力差分の偏りを利用し，候補となる鍵を絞り込む方法である．しかし，事前に対策がとられていたためDES自体は破られていない．線形解読は松井充が1993年に発表した既知平文アタックで，線形解読はDESの解読に成功した最初の例である．どちらの方法もインボルーション型やハッシュ関数に有効であるとされる．ただし，これらの方法での解読には膨大な量の平文—暗号文の対が必要となる．　そのほかの解読例としては鍵の全数探索（総当り攻撃）が挙げられる．可能性のある鍵を全て試していく方法であり，最も単純で時間がかかるとされる．なお，ここでの「解読された」とは鍵がすべて判明することを表している [2], [5].

5.1　線形解読法

線形解読法の原理はまず与えられた暗号関数を部分的により簡単な関数に近似（線形近似）し，もとの暗号関数を解読するかわりに近似された関数を解読することで，より少ない計算量で未知の鍵を求めようというものである．解読対象が近似された関数であるため，結果は正しくない可能性もあるが，得られる情報量（平文—暗号文対）を増やすことで成功確率をあげることが出来る [6], [7].

5.1.1　原理と成功確率

暗号化関数を F，暗号化鍵を K とおくとき任意の平文 P と暗号文 C に対して次の式（平文から暗号文を求める式）が成立している．

$$C = F(P, K) \tag{1}$$

この式においての特定の1ビットに注目すると

$$C[i] = F(P, K)[i] \tag{2}$$

F を近似式 F' に置き換える．

$$C[i] = F'(P, K) \tag{3}$$

するとこの式は必ずしも成り立たなくなる．この式を確率的関係式とみなすと，もしこの F' がでたらめならば，この式が成り立つ確率は $1/2$ になるだろう．しかしこの式が $1/2$ ではない確率で成り立ったとすれば，左辺と右辺に何らかの相関があり，P と C から K の情報を得られたことになる．

5.2　最良表現

すべての S ボックスにおいて，最も線形に近い表現（最良表現）は S_5 の入力の右から5番目のビット x_4 と出力の全ビット y_o, y_1, y_2, y_3, y_4 に対して

$$x_4 = y_0 \oplus y_1 \oplus y_2 \oplus y_3 \tag{4}$$

であり，この確率は $\frac{12}{64} = 0.19$ である．S ボックスの線形近似式は一意的に F 関数の線形近似に拡張することが可能となり，等価になるので

$$R_{n-1}[15] \oplus K_n[22] = f(R_{n-1}, K_n)[7, 18, 24, 29] \tag{5}$$

が成り立つ．ここで，$X[7, 18, 24, 29]$ は $X[7] \oplus X[18] \oplus X[24] \oplus X[29]$ を意味する．f 関数への入力データ R は，まず拡大転置されてから鍵 K_1 と XOR されるので，R の添え字15と K_1 の添え字22がずれる．

5.2.1 3段 DES

このことを初期転置 IP とその逆転値 IP^{-1} を除いた3段 DES に当てはめてみる．データ X に対して，右を下位ビットとし，最右ビットから $X[0], X[1], X[2], \ldots$ と表す．3段 DES の第1段の S_5 の式（5）を適用すると確率 $\frac{12}{64}$ で

$$\begin{aligned} K_1[22] &= R_0[15] \oplus f(R_0, K_1)[7, 18, 24, 29] \\ &= R_0[15] \oplus L_0[7, 18, 24, 29] \oplus R_1[7, 18, 24, 29] \end{aligned} \tag{6}$$

が成立する．同様に第3段 DES の S_5 に式（5）を適用すると，確率 $\frac{12}{64}$ で

$$\begin{aligned} K_3[22] &= R_2[15] \oplus f(R_2, K_3)[7, 18, 24, 29] \\ &= L_3[15] \oplus R_3[7, 18, 24, 29] \oplus L_2[7, 18, 24, 29] \\ &= L_3[15] \oplus R_3[7, 18, 24, 29] \oplus R_1[7, 18, 24, 29] \end{aligned} \tag{7}$$

式（6）と式（7）から $R_1[7, 18, 24, 29]$ を消去すると

$$\begin{aligned} &K_1[22] \oplus K_3[22] \\ &\quad = R_0[15] \oplus L_3[15] \\ &\quad \oplus L_0[7, 18, 24, 29] \oplus R_3[7, 18, 24, 29] \end{aligned} \tag{8}$$

が得られる．式（8）が成立する確率は式（6）と式（7）が両方成立する場合と両方とも成立しない場合の和であるから，$\frac{12}{64}^2 + (1 - \frac{12}{64})^2 = 0.70$ となる．式（8）は3段 DES の最良表現である．

この式は左辺が既知であるが，右辺は未知である．この式が成立する場合は0.5より高いので多くの平文ー暗号文対で式（8）を計算し，0か1の頻度が高いほうであると推定できる．したがって $K_1[22] \oplus K_3[22]$ も推定でき，鍵の情報が1ビット得られたことになる．

5.2.2 １６段 DES への適用

f 関数の線形近似を１６段全体へと拡張する際に問題となるのは，各段における f 関数のよい線形近似が必ずしも全体のよい線形近似にならないということである．なぜなら，各段の線形近似が全体として矛盾なく連結されなければならないからである．このような条件のもとで各段の f 関数を確率 P_i で近似すると，全体の近似確率は

$$\frac{1}{2} + 2^{n-1} \prod_i (P_i - \frac{1}{2}) \tag{9}$$

となる．したがって全体の最良の線形近似確率を求めることは上記の条件のもとで (9) の確率と 1/2 との差の絶対値の最大値を求める問題に帰着される．全体の近似式を求めた後は他と同様に多数決を用いることによって決定する．全体の式の成立確率は 1/2 からわずかしかずれていないので，2^{45} 個程度の平文ー暗号文対が必要となる．

5.3 鍵の全数探索

鍵の全数探索は可能性のある鍵 $2^{56} = 72,057,594,037,927,936$ を全て試していく方法であり，最も単純だが時間がかかる方法である．近年のコンピュータ技術の向上により，DES の 56 ビットの鍵長では，この鍵の全数探索による解読の危険にさらされている．

5.3.1 DES Challenge

鍵の全数探索による DES の危険性を示したのが 1997〜1999 年に開催された RSA 社 [8] による解読コンテストである．コンテストの問題は DES により暗号化された暗号文であり，もとの平文テキストは誰にも知られていないテキストでメッセージ・ヘッダは固定長かつ既知のもの（The secret message is : ）で構成されている．また秘密鍵はコンテスト用の生成ソフトウェアがランダムに作成し，破壊したものを使用してコンテスト管理者も知らされない．以下に開催時期とその結果について表3に示す [9]〜 [11]．表中の PC の数は，インターネットでつながったパソコンの CPU のアイドル時間を利用したものである．**このコンテストにより *1999* 年には１日以内で *DES* が解読可能なことが示され，もはや *DES* は安全ではないということが証明された．また，専用解読マシンは詳細が公開されており，購入・作成すれば誰もが短時間で解読可能である．**

6　DESの動向

近年，コンピュータ技術の向上と低価格化により，全数探索法により正しい鍵を見つけ出すための時間と所要費用が低下してきており，DES の安全性低下が指摘されてきた．そこで T-DES が考案された．T- DES は，DES を 3 回繰り返すことにより，鍵長を 56bit から 112bit または 168bit に伸長し，暗号化段数を 16 段から 48 段に増やすことによってアルゴリズムの統計的偏りを減らし，暗号強度を高めている．DES を前提として構築されている既存コンピュータシステムにおいて，T-DES に移行することは比較的容易であるという利点があるが，DES の処理を 3 回繰り返すことから，基本的には暗号化および復号に要する時間はそれだけ長くなるという欠点もある．さらに安全で T-DES よりも暗号化が短時間で行えるアルゴリズムの必要性から，2001 年に次期標準暗号が AES に制定された．AES が一般に広く普及するまでには，第三者機関による安全性の評価などにより時間がかかる．T-DES は AES の橋渡しの役割と考えられており，現在は AES と T-DES の両方が広く使われている．

AESによる暗号化

AES:Advanced Encryption Standard

7　DESからAESへ

近年，計算機の高速化・広帯域化にともない，指数関数的な計算速度向上が見られており，加速度的に解読が行われることも予想される．鍵長 56 ビットの DES に総当り攻撃を行った場合必要となる，2 の 56 乗回の計算程度では現在のコンピュータで十分に処理可能と考えられ，解読の危険が高まっている．そこで，DES を三重にかけることで強度を高めた，Triple DES と呼ばれる暗号方式を提案した．しかし，Triple DES を使えば安全ではあっても処理が重く効率が悪い．そのため，DES に代わる新しい米国政府標準暗号が必要となり，1997 年から NIST（National Institute of Standards and Technology）により AES の選定計画が開始され，2001 年に FIPS197 に制定された [4].

8　AESとは

AES とは，不定長の文章データを先頭から順に 128bit の長さのブロックに区切って，各ブロックに対して共通鍵により暗号化・復号を行う，共通鍵ブロック暗号である．暗号化の過程では，DES と同様に同一の暗号化処理を何回も繰り返して行う．その繰り返す回数を『ラウンド数』と言い，単位は段で表す.

8.1　AESの特徴

主な特徴として，以下の 3 つが挙げられる．また，これらの特徴以外の性質を次節以降で示す [12], [13].

- 共通鍵ブロック暗号である
- ブロック長は 128bit が利用可能
- 鍵長は 128bit,192bit,256bit が選択可能

8.2　ブロック長，鍵長，ラウンド数

DES と AES のブロック長と鍵長，ラウンド数をまとめた表 4 で示すように，AES のラウンド数は鍵長に依存している．AES はブロック長と鍵長を DES より長くすることによって，全数探索法や暗号文一致攻撃，辞書攻撃に対して十分な安全性が確保できるように配慮されている [14].

8.3　基本構造

AES は DES の持つ Feistel 構造とは異なる，SPN（Substitution Permutation Network）構造を持つ（図 3 参照）．SPN 構造とは，入力されたブロックデータを複数のサブブロックに分割した後，それぞれ換字処理（substitution）を行う．換字処理終了後に，全体をまとめて転置処理（permutation）を行う．構造上 Feistel 構造よりも 1 ラウンドの実行処理が早いという特徴がある反面，関数の選択によっては強度が保証できないという欠点がある．また，暗号化変換と復号化変換が同じアルゴリズムにならない [12].

9　暗号化処理の流れ

この章では，まず暗号化全体の流れを追う．その後，暗号化処理 1 ラウンドの流れを追うと共に，各暗号化

関数について説明する．

9.1 暗号化処理全体の流れ

AES は暗号化に用いる鍵生成部と暗号化処理を行う部分の 2 つから成る．暗号化処理部では，最初に鍵生成部で共通鍵から生成されたサブ鍵との排他的論理和をとる．その後，同一の暗号化処理を『ラウンド数-1』回繰り返す．最終ラウンドではラウンド内で行われる処理の 1 つ（MixColumns）を省く（図 4 参照）．

9.2 暗号化処理 1 ラウンドの流れ

各ラウンドでは，入力に対して『換字変換 (SubBytes)』,『転置変換 (ShiftRows)』,『演算による線形変換 (MixColumns)』,『サブ鍵との排他的論理和 (AddRoundKey)』の変換を順に行い出力を得る．各関数の説明は次章以降で行う（図 5 参照）．

10 暗号化関数

10.1 SubBytes

SubBytes 処理は，128bit の入力データを 8bit ごとに区切り，各ブロックごとに換字変換を行う関数である．

本来は，8bit のデータを有限体 GF(2^8) 上の元とみなし，GF(2^8) 上の既約多項式：$m(x) = x^8+x^4+x^3+x+1$ を用いて乗法に関する逆元をとる．その後，アフィン変換を行う [13].

しかし，入力された 8bit 長のデータに対して出力を一意に決定できることから，現実的には先に置換表（S-box）を計算しておき，その置換表に従って変換を行っている（表 5 参照）．

10.2 ShiftRows

変換を行うための準備として，入力データを 1byte ずつに区切り 4 行 4 列のマスに配置する（図 6 参照）．ShiftRows ではこの 4 行 4 列の各行に対して行う線形の転置変換で，byte 単位の巡回シフトを行う（図 7 参照）．シフトする量は表 6 に示すように，行ごとに決まっている．

10.3 MixColumns

MixColumns は，入力データを 1byte ずつに区切り，4 つの要素（4byte）ごとに演算による線形変換を行う関数である（図 8 参照）[13]．以下に c_5 から c_8 を入力とし，d_5 から d_8 を出力として得る処理を例に挙げて手順を説明する．

step 1 図 9 に示すように，4 つの要素を各項の係数とする，4 項からなる有限体 GF(2^8) 上の 3 次多項式とみなす．

step 2 以下に示す多項式 $A(t)$ と $C(t)$ の乗算を行う．ただし，多項式としての積 は 4 次を越えるので，多項式 $M(t) = t^4 + 1$ を用いて余り（剰余）を求める．また，係数についても a_j と c_j の積が 1 つの要素（4byte）で示すことができる $0 \leqq a_j \times c_j < 256$ の範囲外になる場合もあるので，既約多項式 $m(t) = x^8 + x^4 + x^3 + x + 1$ を用いて剰余をとる．

$A(t) = (03)_{16}t^3 + (01)_{16}t^2 + (01)_{16}t + (02)_{16}$

$C(t) = c_5t^3 + c_6t^2 + c_7t + c_8$

$$D(t) = A(t) \otimes C(t) (mod\ t^4 + 1)$$

この演算を行い，$D(t) = d_5t^3 + d_6t^2 + d_7t + d_8$ として整理すると，

$d_5 = (02)_{16}c_5 \oplus (01)_{16}c_6 \oplus (01)_{16}c_7 \oplus (03)_{16}c_8$

$d_6 = (03)_{16}c_5 \oplus (02)_{16}c_6 \oplus (01)_{16}c_7 \oplus (01)_{16}c_8$

$d_7 = (01)_{16}c_5 \oplus (03)_{16}c_6 \oplus (02)_{16}c_7 \oplus (01)_{16}c_8$

$d_8 = (01)_{16}c_5 \oplus (01)_{16}c_6 \oplus (03)_{16}c_7 \oplus (02)_{16}c_8$

となる．

上記の手順により，出力 d_j を得ることができる．また，多項式 $A(t)$ を固定された多項式とすると，上で述べた乗算は式 (10) のような行列の積で表すこともできる．

$$\begin{bmatrix} d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} = \begin{bmatrix} 02 & 01 & 01 & 03 \\ 03 & 02 & 01 & 01 \\ 01 & 03 & 02 & 01 \\ 01 & 01 & 03 & 02 \end{bmatrix} \begin{bmatrix} c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix} \tag{10}$$

10.4 AddRoundKey

入力 d_j と鍵生成部で生成したサブ鍵 k_j との byte 毎の排他的論理和をとる（図 10 参照）．

この AddRoundKey 処理の出力が暗号化処理 1 ラウンドの出力となる．

11　鍵生成部

9.1 節で述べたように，AES を構成する鍵生成部について説明する．サブ鍵は各ラウンドで異なる鍵を用いる．つまり，異なるサブ鍵が「ラウンド数＋ 1」個，表 7 に示す個数だけ必要となる．よって，この鍵生成部では，入力された 1 つの暗号化鍵（128bit, 192bit, 256bit）から，ブロック長 128bit と同じ長さのサブ鍵を，表 7 に示す個数だけ生成することが目的である．

11.1　サブ鍵生成の手順

- 暗号化鍵を拡張し，長さ「128 ×サブ鍵の数」bit の拡張鍵を作る
- 拡張鍵から表 7 に示す個数のサブ鍵（128bit）を得る

拡張鍵の生成方法とサブ鍵を得る方法の説明は次節以降を参照.

11.2　拡張鍵の生成手順

step 1 入力された暗号化鍵を 1byte(8bit) ずつに区切りる．そして，順番に 4 要素ずつとり，W[0]…W[3] とおく（図 11 参照）

step 2 拡張鍵から表 7 に示す個数のサブ鍵（128bit）を得る暗号化鍵から $W[3]$ までは作ることができる．しかし，長さ「128×11」bit の拡張鍵を得るためには，$W[4]$ 以降も必要となる．$W[4]$ 以降の $W[i]$ は次の式 (11) で求める．

$$W[i] = W[i-4] \oplus W[i-1] \qquad (11)$$

$4 \leqq i \leqq 43$ について，$W[i]$ を求めることで，暗号鍵を拡張した，長さ「128×11」bit の拡張鍵を生成できる．

※ ただし，i が 4 の 4 の倍数であるとき，次に述べる処理を先に行い，式 (11) の $W[i-1]$ に当てはめる値を求める．

$$W[i-1] = SubWord(RotWord(W[i-1])) \oplus Recon[i/4] \qquad (12)$$

ここで，式 (12) の関数は以下のように定義する [13].

- $SubWord()$：byte ごとに換字変換を行う
- $Rotword()$：1byte 左シフトする
- $Rcon[i/4]$：$i/4$ の値によって決められた値をとる（表 8 参照）

11.3　サブ鍵の選択

長さ「128×11」bit の拡張鍵を生成したので，図 12 のように表すことができる．この拡張鍵からサブ鍵を得るためには，図 12 に示すように，$W[i]$ を順番に 4 つずつとり，サブ鍵 j とする．つまり，サブ鍵 j は $W[4 \times j]$ から $W[4 \times j + 3]$ のビット列をつないだものとなる．

12　強度

鍵長の長さ以外に暗号の強度を判断する材料として，どれほど「クチャクチャこねまわされているか」ということが挙げられる．たくさんこねまわせば，それだけ強い暗号であると言えるが，計算量が膨大になり暗号化に時間的なコストが掛かるといった問題をクリアする必要がある．

AES は種類の違う非線形変換（SubBytes, ShiftRows）と線形変換（MixColumns）による変換によりグチャグチャにこねまわされている．それでいて，Triple DES よりも効率が良い．

ただし，「代数処理」であることから，その特徴をついた攻撃法が既に学会レベルでは提唱されている．とはいえ，その攻撃法の実行可能な計算時間を考慮した有効性については未詳である．

まとめ

情報漏洩，不正アクセス，ウィルスなど様々な問題が起こっている中，住民基本台帳ネットワークシステムや電子申請といった，個人情報を含めた情報の電子化が進んでいる．また，電子政府，電子自治体関連だけでなく，企業でも大量の情報を扱っている．個人情報保護法が制定され，個人情報の取り扱いが注目されている現在，各自が情報の取り扱いやセキュリティについて，ある程度の知識を持つ必要が出てきている．

多くの専門書が発売されているが，初心者には理解

が困難なものが多い．初心者にもわかり易くという点に焦点をおいた我々の説明が，暗号やセキュリティについての理解のきっかけとなることと思う．

参考文献

[1] 辻井重男, 岡本栄司, ”暗号のすべて－ユビキタス社会の暗号技術 ”電波新聞社,2002

[2] 岡本龍明, 山本博資, “ 現代暗号 ”, 産業図書, 1997

[3] 辻井重男, 笠原正雄, ”情報セキュリティ ”, 昭晃堂 2003

[4] 宇根正志, 太田和夫, ”共通鍵暗号を取り巻く現状と課題－ DES から AES へ－ ”金融研究,1999

[5] 岡本栄司, ”暗号理論入門 第２版 ” 共立出版

[6] Matsui, M, ”Linear Cryptanalysis Method for DES Cipher”, Advances in Cryptology Eurocrypt '93, Lecture Notes in Computer Science 765, Springer-Verlag, pp.386-pp.397(1994)

[7] 松井充, ”暗号の攻撃・解読方法：線形解読 ”

[8] RSASecurity, http://www.rsasecurity.com/rsalabs/

[9] RockeVercer, http://home.earthlink.net/ rcv007/deschall.htm

[10] disutributed.net, http://distributed.net/

[11] EFF, http://www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/

[12] NIST（National Institute of Standards and Technology）, http://www.nist.gov/

[13] Joan Daemen, Vincent Rijmen, ”The Design of Rijndael”

[14] 独立行政法人 情報通信研究機構, ”暗号技術評価報告書 CRYPTREC Report 2002”, http://www2.nict.go.jp/

付録 A：図表

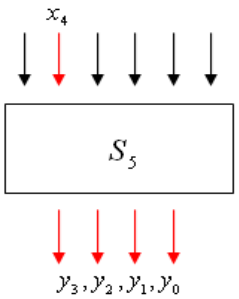


図 1: S_5 ボックスの入出力

表 1: 共通鍵暗号と公開鍵暗号の特徴

	共通鍵暗号方式	公開鍵暗号方式
暗号鍵と復号鍵	同一（共通）	異なる
秘密にすべき鍵	多数（相手ごとに必要）	自分の秘密鍵のみ
鍵の受渡し	秘匿が必須	公開可
処理速度	高速	低速
適用例	長文 膨大なデータ処理が求められるデータなど	短文 デジタル署名や共通鍵暗号の鍵配送など

表 2: 暗号アルゴリズムの分類と代表的な例

	利用者への鍵の配置の仕方	
	共通鍵暗号方式	公開鍵暗号方式
守秘	共通鍵暗号方式 DES，Triple DES，IDEA，MULTI，MISTY，CAST，RC2，RC4，RC5，AES 等	公開鍵暗号方式 RSA，ElGamal，楕円曲線 ElGamal 等
認証	共通鍵認証方式 同上（共通鍵暗号アルゴリズムを利用して MAC 等を生成）	公開鍵署名方式 RSA 署名，ElGamal 署名，DSA，楕円曲線 ElGamal 署名，楕円曲線 DSA 等

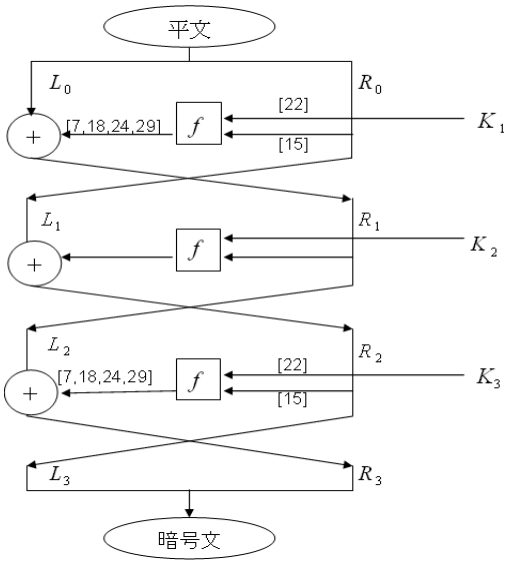


図 2: 3 段 DES

表 4: ラウンド数

暗号方式	ブロック長	鍵長	ラウンド数
DES	64bit	56bit	16段
AES	128bit	128bit	10段
		192bit	12段
		256bit	14段

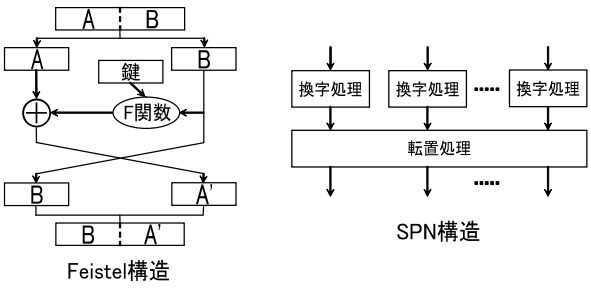


図 3: Feistel 構造と SPN 構造

表 3: DES Challenge

時期・勝者	解読時間	
１９９７年１月 DES Challenge Rocke Vercer らのグループ	９６日	約７万台の PC ７ billion keys per second “ Strong cryptography makes the world a safer place ”
１９９８年２月 DES Challenge　-1 Distributeted.net	４１日	約５万台の PC 34billion keys per second “ Many hands make light work ”
１９９８年７月 DES Challenge　-2 EFF	５６時間	$ 25,000 で作成した解読専用マシン 88billion keys per second “ It ’s time for those 128-,192-,and 256-bit keys ”
１９９９年１月 DES Challenge Distributeted.net	２２時間	DES 専用解読マシン＋ 約１０万台の PC 250billion keys per second “ See you in Rome (second AES Conference, March 22-23,1999) ”

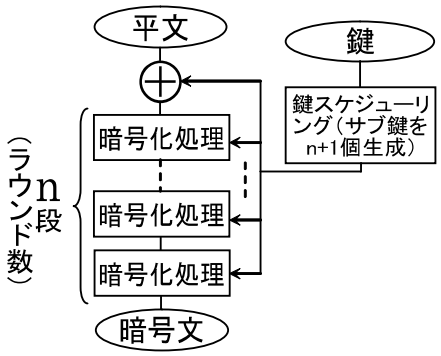


図 4: 暗号化処理全体の流れ

表 5: SubBytes 関数の置換表

		下位4bit														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
上位4bit	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	BD
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

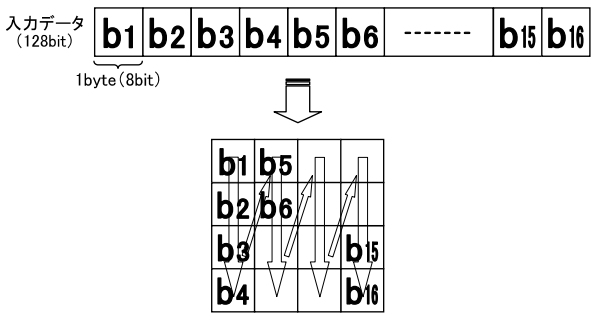


図 6: ShiftRows 処理前の準備

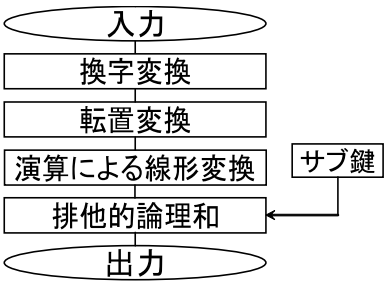


図 5: 暗号化処理 1 ラウンドの流れ

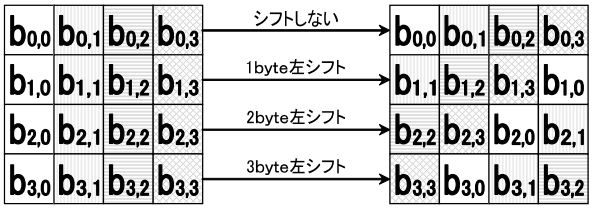


図 7: ShiftRows 関数による転置変換

表 6: シフトの量

	0行目	1行目	2行目	3行目
シフトの量	シフトしない	1byteシフト	2byteシフト	3byteシフト

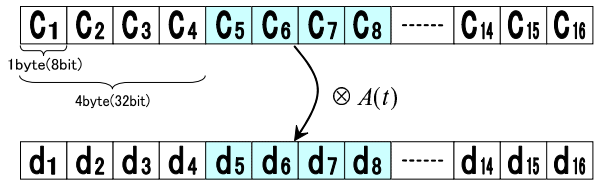
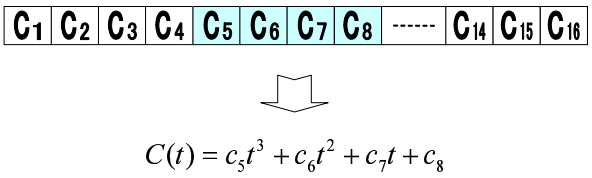


図 8: MixColumns 処理前の準備



$$C(t) = c_5 t^3 + c_6 t^2 + c_7 t + c_8$$

図 9: MixColumns

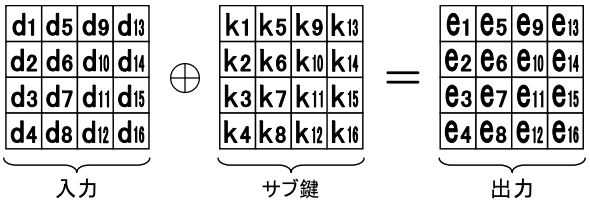


図 10: AddRoundKey 関数による排他的論理和

表 7: 必要なサブ鍵の個数

鍵長	ラウンド数	サブ鍵の個数
128bit	10段	11個
192bit	12段	13個
256bit	14段	15個

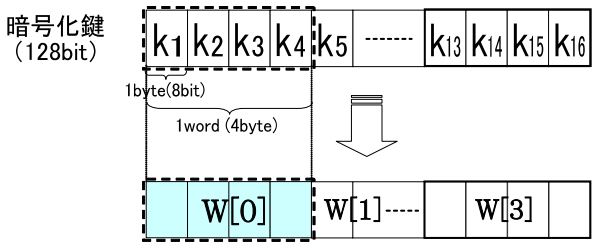


図 11: サブ鍵生成の準備

表 8: $Rcon[i/4]$ の値

Rcon[$i/4$]	値
Rcon[1]	01000000
Rcon[2]	02000000
Rcon[3]	04000000
Rcon[4]	08000000
Rcon[5]	10000000
Rcon[6]	20000000
Rcon[7]	40000000
Rcon[8]	80000000
Rcon[9]	1b000000
Rcon[10]	36000000

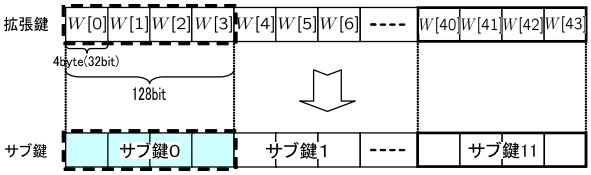


図 12: サブ鍵の選択