講　演

暗号技術の動向とセキュリティ†

土　居　範　久††

ただいまご紹介にあずかりました慶應の土居でございます．このような席でお話させていただくことに関しまして大変光栄に存じております．暗号を中心にセキュリティ全般にわたって話をせよということでございますが，なにぶん短時間ですので，端折り過ぎになるかもしれませんが，ひととおりお話をさせていただくつもりでおります．

情報セキュリティといいましたときに，私が普段使います定義は，自然災害，障害，ミス・エラー，あるいは意図的行為といいますようなもの，このようなものを脅威といいますが，このようなものから計算機システムを中心とした情報処理活動に係わる資産を守ろうとすることです．ほぼ同じような意味合いをもつ言葉として，コンピュータセキュリティとかデータセキュリティという言葉も使われています．

特に顕著なセキュリティの侵犯としては，情報の漏洩，情報の改ざん，および使用妨害といったものがあげられるかと思います．これに対して，セキュリティ対策を考えますと，外部セキュリティ制御，内部セキュリティ制御，およびインタフェース制御と呼ぶこともありますが，正当な使用者であることを認証する使用者の認証の三つに大きく分類できるかと思います．インタフェース制御は広義の意味でアクセス制御と呼ぶこともあります．

外部セキュリティ制御

それでは，これらを一つずつ簡単に説明させていただきます．まず，外部セキュリティ制御と呼ばれていますものにはいろいろあります．設備，運用，制度に係わる問題つまり計算機システムの外側の問題を扱い，まず，地震，火災，水害，停電，空気汚染，回線切断，あるいは鼠害といったようなものからどう守るかといった物理的対策があります．それから，管理運用上の対策があります．これは，人的災害，破壊活動，盗難，機密漏洩といったものに対する対策です．また，オーディット（決算）などを捕まえてシステムをチェックしようというシステム監査が最近特に賑わしているようです．それから，リスク分析だとか，評価基準だとか法的保護といったようなものがあるかと思います．これらのうちでも，特に，物理的対策とか管理運用上の対策といったようなことに関しては，各省庁が最近特に熱心でして，二三例をあげますと，まず，通産省が，電子計算機システム安全対策基準を1977年4月に制定し，それに合格している事業所かどうかを認定する制度を1981年7月に発足させています．基準にあった対策を講じるためには，かなり費用が掛かりますので，認定を受けた事業所の数は当初はごくわずかだったのですが，このところ徐々に増えているようです．また，郵政省は，今年1987年に情報通信ネットワーク安全信頼性基準を制定し，こちらは登録制度をとっておられます．また，自治省では地方公共団体コンピュータセキュリティ対策基準というものをこの7月に発表し，地方公共団体がデータの入力作業を外注するときにはどうせよといったようなことまでを含めた細かい基準を決めておられます．

しかし，このようなことをしましても，いろいろと問題が起こるわけですので，それをカバーするといったような意味の一つとして，俗にいうセキュリティ保険というものがあります．これはまだ完全に完備しておりませんが，既存のものと致しますと，これも俗称のようですが，情報化保険というものがありまして，コンピュータ総合保険と情報処理業者賠償責任保険というものが用意されております．もっとも，情報の価値は認められておりませんで，たとえば磁気テープあるいはディスクパックに入れておいたデータが部外者によって破壊されたとき何が保障されるかといいますと，復元するための諸費用あるいは新しい磁気テープないしはディスクパックの購入費用ということです．また，ソフトウェアハウスでものを作られたときに，

† 情報処理学会第35回全国大会特別講演（昭和62年9月28日）
場所　北海道大学
†† 慶應義塾大学

担保が半年だったかと思いますが，担保期間を過ぎてそのソフトウェアがもとで発注元に損害を与えたようなときに保障してくれるような保険などが情報処理業者賠償責任保険にあるようです．これらの他にも，つい先ごろできたようなのですが，金融機関包括補償保険というものがあり，その中にコンピュータ犯罪保険というものがあるようです．それから，世田谷の事故の後にできたようなのですが，店舗休業保険の中に通信途絶事故保担保というのがあるようです．といったような具合で，保険のほうは少しずつですが，整備されていきつつあるようです．

内部セキュリティ制御

さて，その次の内部セキュリティ制御は，通常，4つに大きく分けております．一つが狭い意味でのアクセス制御です．これは，システム内の対象への直接のアクセスがすべて正当と認められたものであることを保証しようということです．つまり，多くもなく少なくもなくちょうどよいアクセス権を与えようというわけです．代表的なモデルとしてアクセス行列があります．したがいまして，特にOSあるいはデータベースといったようなところで，この手のことを行うことがなされております．次がフロー制御ですが，これは正当なアクセス権をもつものによって権限をもたないものにデータが漏洩することを防ごうとするものです．アクセス制御とともにフロー制御を行う代表的なモデルとしてBell-LaPadulaモデルがありますが，このモデルでフロー制御をどのようにやるかといいますと，たとえば兵隊の位で申しまして，上官のファイルには書けるが部下のファイルには書けず，部下のファイルは読めるが上官のファイルは読めないというのが基本になっております．しかしそれだけでは，部下が上官のファイルにどんどんろくでもないことを書いてくる可能性がありますから，何が本当の情報か分からなくなるというようなこともありまして，D. Denningらが改良したりしております．それから3番目に推論制御があります．これは，いわゆるデータベースなどで統計データなどを一つ一つ見てゆきますときには個人情報いわゆるプライバシに関するようなことは分からないのが，あれやこれやうまい具合にやっていくと分かってしまうということがないようにしようというものです．そして，最後が暗号化制御です．ATMから銀行だとか銀行間の決済とかといったようなことで金銭に係わるデータが電線の上を飛んでおりますし，また各省庁管轄のもとでわれわれに関する情報が山と蓄えられたりしておりましていわば重大な金庫になっていたりするわけですから，そういうものをむやみやたらに簡単に引き出されると困ります．そこで一見したところで分からないようにしようというのが暗号化制御でして，これについてはあとで説明させていただきます．

使用者の認証

次は，使用者の認証ですが，これは通常，使用者の知っている何か，あるいは使用者がもっている何か，あるいは使用者自身の何かを使って行います．まず，もっともポピュラなものにパスワードがあります．これは使用者が知っている何かによって認証しようとするものです．いわゆるハッカーが侵入してくる場合の大半はこのパスワードが分かってしまうことにあるわけです．そこで，ただちに分かってしまうようなパスワードだとか生年月日だとかを使いますと困ることになりますので，最近はいろいろと仕掛けが設けられつつあります．もっとも，その昔TSSが始まりました当初に高橋秀俊先生が考案されましたような数式を使いましても，パスワードファイルが盗まれたら同じですので，最近は，それも暗号化するかそれに近いようなことをして守ろうとするようなこともされております．

それから，代表的なものとしますと行動による認証があります．これはサインだとか運筆を用いるものです．

また，生物学的情報による認証があります．これは，使用者自身の何かを用いた認証ですが，指紋，掌紋，手形，音声，あるいは「スタートレック」でも使われた網膜パターンといったものを使おうとするものです．網膜パターンによるものはすでに商品として売られていますが，この手のものは，心理的にあまり不快感を与えることはまかりならんというのが建前になっています．したがって，体液なども使えるようですが不適当かと思われます．

また，これからのものとしてカードがあります．これは，記憶媒体による認証で，現在，磁気ストライプカードが多用されていますが，ICカード，あるいは光カードが今後いろいろな場面で使われるのではないかと思われます．

一応，駆け足でしたが，こういうようなところが外部セキュリティ制御，内部セキュリティ制御および使用者の認証における研究開発の現状です．

暗号の必要条件

さて，それでは，暗号の話に移らせていただきたいと思います．暗号といいますものは，ご存じのように，古来からあります．特に，軍事・外交などで使われていたわけですが，先ほど申しましたようにコンピュータの時代にふさわしい暗号というものがないものだろうかというようなことでいろいろ研究がなされているわけです．

暗号といいますのは，通信文とか電文とも申しますがメッセージを正規の受信者以外の人から秘匿するためのもので，他人が見たところで分からないように変換するわけですが，その変換のときに重要な働きをしますものに手続きと鍵があります．手続き，これはアルゴリズムですが，これは広く知られたものであってもよろしいのですが，鍵は原則として秘密にしておくわけです．

それでは，暗号の必要条件はといいますと，まず，暗号化するのに時間がかかるのだと困りますので，とにかく（1）暗号化が容易である必要があります．特に，データ伝送で用いようとしますと，1秒間に何ビット変換できるかというようなことが死命を制することがあります．暗号化が容易だからといって解読が容易であったらなんのための暗号か分かりませんので，（2）解読は難しくなければいけないわけです．しかし，受け取ったほうがそれをもとへ戻すのがえらく難しいのだと，これまた使いものになりませんので，2番目の条件と相反するようなことになりますが，（3）もとへ戻すことつまり復号は容易でなければいけないわけです．

暗 号 系

そのための暗号系ですが，大きく分けて二つに分かれます．一つは公開鍵暗号系です．これは，非対称暗号系と呼んだりすることもあります．これに対して，実はそんなものが出てきてしまったものですから，古来から使っておりましたようなものを慣用の暗号系あるいは秘密鍵暗号系，または非対称に対して対称暗号系と呼んだりします．

それでは，この慣用の暗号系とはどんなものでコンピュータ関係ですとどうなっているかということ，および公開鍵暗号系というのはいったいどんなものかということを，次に話させていただきたいと思います．

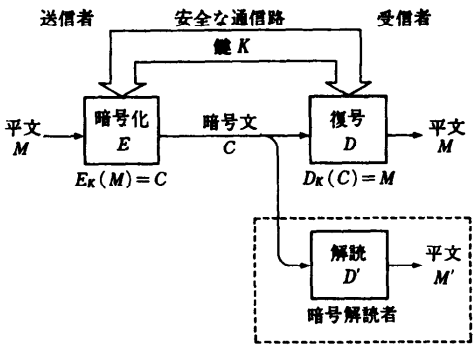

図-1 慣用の暗号系の構造

慣用の暗号系

まず，慣用の暗号系の構造ですが（図-1 参照），メッセージを送り出すほうは平文(ひらぶん)，要するにメッセージで暗号化されていないもの，を作ります．それを暗号化して送るわけですが，暗号化するためには先ほど申しましたように手続きと鍵が必要です．そこで鍵は安全な通信路を使って送り手と受け手の双方にあらかじめ渡しておくことになります．その鍵 K を使って平文 M を手続き E で暗号化して暗号文 C を作り，それを送ります．受け手のほうは秘密の鍵 K を使い復号の手続き D で暗号文 C をもとに戻します．このとき，悪意があろうとなかろうと正当な受信人以外の者がその暗号をもとへ戻そうとすることを解読といいます．このときに使う手続き D' は必ずしも復号の手続き D と同じであるとはいえませんし，出てきました平文 M' も M と全く同じであるかどうかも分かりません．こういう者がいるものですからいろいろと考えなければいけなくなるわけで，こういう者を暗号解読者とか暗号破りなどといいます．

このときの鍵 K の働きですが，これは，暗号化の手続きおよび復号の手続きのパラメータとなるわけで，これらの手続きの組の中から特定の組を一つ選ぶ働きをします．そのために，暗号方式をいちいち変えなくても鍵を変えさえすれば何度でも同じ手続きを使って秘密の通信をすることが可能になるわけです．

たとえば，鍵が 10 進数 2 桁だとすると 10^2 通りになりますし 30 桁だとすると 10^{30} 通りになるわけです．そこで，もしも鍵の候補の数が少ないときには，先ほど，手続きは公開しても構わないと申しましたが，周知のものですとしらみつぶしに調べさえすれば

ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC

DOI NORIHISA → GRL QRULKLVD
(1)

ABCDEFGHIJKLMNOPQRSTUVWXYZ
QWERTYUIOPASDFGHJKLZXCVBNM

DOI NORIHISA → RGO FGKOIOLQ
(2)

図-2 単文字換字式暗号（シーザー暗号）

暗号化
平文 1101000110110101101000001
鍵 0100110011010001011110100
暗号文 1001110101100100110101 01

復号
暗号文 1001110101100100110101 01
鍵 0100110011010001011110100
平文 1101000110110101101000001

図-3 バーナム暗号

解読できてしまいますので，これらの手続きも秘匿するのが普通です．

慣用の暗号系の基礎

次に，慣用の暗号系の基礎となる方式として，まず換字（かえじ）があります．これは，メッセージの中の個々の文字を系統的に別の文字で置き換えるもので，平文の文字と暗号文の文字との対応を付けたものが鍵になります．これに対して，転置があります．これは，メッセージの中の各文字の順番を一定の置換にしたがって入れ替えるもので，その入れ替え規則が鍵になります．そして，この二つともを使うような暗号を合成暗号と呼びます．

たとえば，換字式の暗号できわめてシンプルな形のものにシーザー暗号と呼ばれているものがあります．これはアルファベットを循環シフトさせているわけで，特にジュリアスシーザーは三つずらしたものを使ったようですが，その対応関係は図-2 の(1)のようになります．この場合，AはDに，BはEにというように置き換えるというのが約束で，この手続きを知らせておきますと鍵は3になります．たとえば，DOI NORIHISA は GRL QRULKLVD という暗号文になるわけです．これですと鍵は 26 通りしかありませんので，手続きが分かっておりますと簡単に解読できてしまいます．それに対して，図-2 の(2)のようにアルファベットを重複することなく拾いだして対応付けるようなことをしますと組合せの数は 26！ つまり 10^{26} 程度になるわけです．しかし，そうなったところで，通常の場合には英文ですと，たとえばEが出てくる頻度が高いといったようなことを使えば割と簡単に破られてしまうわけです．

それでは，何がなんでも破られるかといいますと，従来どおりの方法でやりましてもメッセージの長さと鍵の長さを同じにするような暗号にバーナム暗号と呼ばれる暗号がありますが，これは無条件に安全な暗号だということになっております（図-3 参照）．たとえば，平文をビットつまり2進数値に直したものを考え，平文と同じビット数の鍵を用意しまして，その鍵をワンタイムパッドといいますが，1回こっきりの使い捨てをすることを考えます．そして暗号化の手続きとして排他的論理和つまり2を法とする加算を行います．受け取ったほうでは，2を法として加算することで暗号化したのですから，2を法とする減算をすればよろしいわけですので，同じ鍵を使って再び排他的論理和をとりますと元に戻ります．簡単に申しますと，作り方から，暗号は0と1がランダムに並んだものになりますので，これは解読できません．ただし，メッセージと同じ長さの鍵を秘密の通信路を使って伝えなければいけないというところが難点でして，特別の場合以外に使うのはまず難しいわけです．

そうしますと，メッセージの長さよりも通常は極端に短いものを鍵として使うわけですが，メッセージの長さとは無関係な鍵の量に依存するような暗号は理論的には解読できてしまいます．それでは，そんなものを実際になぜ使っているのかということになりますが，そのわけは，実際に解読しようとしますと膨大な時間がかかる，あるいは膨大な費用がかかる，あるいはその両方がかかるというようなことが起こりまして，とても間尺に合わないからやれないということになり，それなら安全だろうというようなことで使われているわけです．だろう，といいましたのは，コンピュータがだんだん速くなりますから，いまは駄目でもしばらくすると解けてしまうかもしれないなどというようなことがありますものですから，だろうぐらいしかいえないわけです．

暗号方式としまして，このあと名前が出てきますのであげておきますと，逐次暗号とかブロック暗号というものがあります．逐次暗号というのは平文を端から順次暗号化していくもので，ブロック暗号というのはメッセージをブロックに区切り，そのブロック単位で

暗号化していくものです．それから，よく使われます言葉にコードとサイファがありますが，コードというのは語とか句とかいった意味のある項目を他の語とか数字群に置き換えるものであるに対し，サイファというのは語を構成している個々の記号を操作するものという使い分けがなされているようです．

DES

さて，この慣用の暗号系につきましては，コンピュータ周りですと，代表的なものとして DES (Data Encryption Standard) があります．これが，いま，世の中の，いわゆるこの手の暗号の範とされているものですが，これは 1977 年に米国商務省標準局が米国の規格として制定しまして，合衆国政府ではこれを使うが民間でもできるだけ使ったらどうかと持ち出してきたものです．FIPS Pub 46 として出版されております．

これは公募されたものですが，第 1 回目は，確か，うまくいかなかったのではなかったかと思いますが，何回目かに応募されたもので，IBM が 1975 年に発表した Lucifer という暗号系の縮小版のようなものになっていまして，ブロック暗号で換字と転置を使う合成暗号つまりブロック合成暗号です．

もちろん，手続きは規格として世の中にさらされております．いま時分になって，米国はさらしてしまったのはちょっとまずかったかなと反省しているようですが．データは 64 ビットごとのブロックに区切り，鍵は 56 ビットに 8 ビットのパリティを付けたものを用います．実際の鍵は 56 ビットですから，鍵の候補の数は 2^{56} つまり 7×10^{16} くらいになります．つまり，10^{16} より大きな数になりますので 1 マイクロ秒に一つずつしらみ潰しに調べてもスーパコンピュータで 1000 年はかかる，だから大丈夫だろうということです．規格そのものはソフトウェアで作ったものは認めないことになっていますが，数千ゲートでワンチップ

(1) 64ビットのデータに初期転置 IP を施し 32ビットずつ左半分 L_0 と右半分 R_0 に分ける

(2) $n=1,2,\cdots,16$ に対して
$L_n=R_{n-1}$
$R_n=L_{n-1}\oplus f(R_{n-1},K_n)$
を繰り返す

(3) R_{16} と L_{16} とをつなげた 64 ビットに初期転置の逆の転置 (IP^{-1}) を施す

図-4 DES の暗号化の手続き

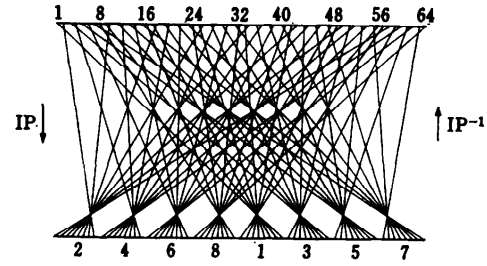

図-5 初期転置 IP とその逆 IP^{-1}

(1) 初期転置 IP を施し，結果を R_{16} と L_{16} に分ける

(2) $n=16,15,\cdots,1$ に対して
$R_{n-1}=L_n$
$L_{n-1}=R_n\oplus f(L_n,K_n)$
を繰り返す

(3) L_0R_0 に対して IP^{-1} を施す

図-6 DES の復号の手続き

化が可能でして，現実にそのチップが売られております．おおよそ 16 Mbit/秒くらいの速度のようです．マイクロプロセッサでソフト的に行ったとしますと数 Kbit/秒くらいの速度で行くような感じです．

それでは，DES はどうなっているかということですが，あまり細かいところまで絵ときをすることはできないかと思いますが，基本的な仕掛けは図-4 のようになっています．まず，64 ビットのデータに初期転置といわれています図-5 のような転置を施します．図 5 の下に書いてある数字は，頭から 8 ビットずつに切ったときの先頭からのビット位置です．結局，データを 8 ビットずつに切って，その 2 ビット目，4 ビット目，6 ビット目という順に揃え直すというような転置を行い，それを左の 32 ビットと右の 32 ビットに分けるわけです．分けてから 16 段にわたって図-4 の (2) のようなことを繰り返します．つまり，左半分として一つ前の右半分をそのまま使い，右半分としては一つ前の左半分と関数 f の値との排他的論理和を使うということを 16 回繰り返します．結果として出てきた右半分 R_{16} の次に左半分 L_{16} をつなげたものに初期転置の逆を施したものが最終的な暗号文になります．

復号はどのようにすればいいかといいますと，これは簡単で図-6 のようになります．実際には悪魔的なことをやってはいるのですが，要は，一つの暗号の箱のようなものを用意しておき，そこに上から平文を入れると暗号文が出てくるとしますと，できることなら，

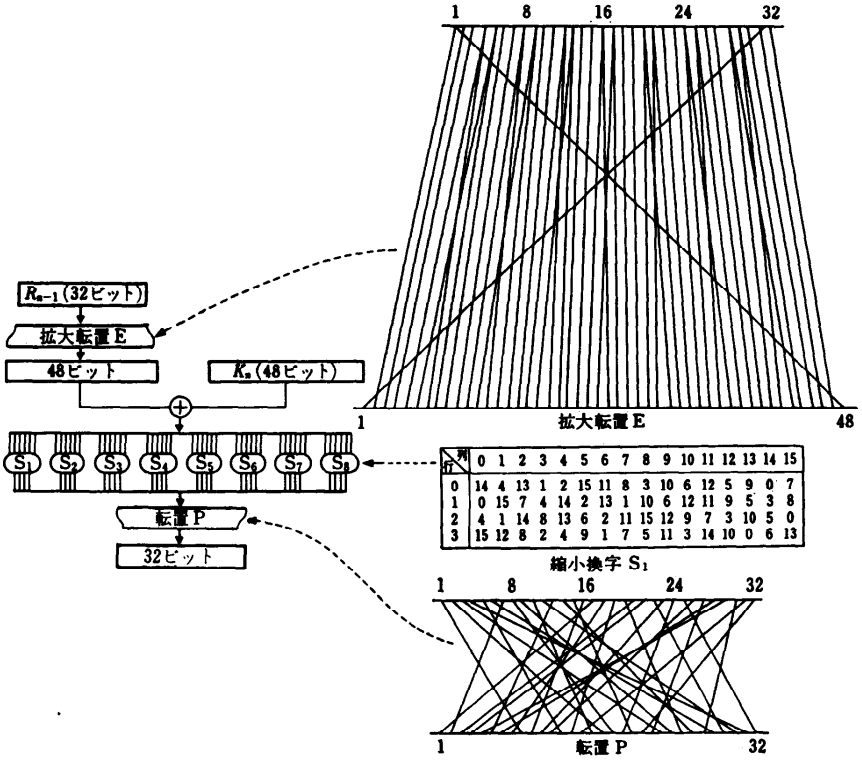

図-7 暗号関数 $f(R_{n-1}, K_n)$

その暗号文をもう1度同じ箱に上から入れたら元に戻ってくれるというのが一番単純なわけで，その手の仕組みをインボリューションなどといいますが，基本的にはそれをやっているわけです．

まず，初期転置を施します．その結果が R_{16} と L_{16} になります．次に，鍵は反対から使いながら，次の右半分として一つ前の左半分をそのまま使い，次の左半分 L_{n-1} は，R_n が L_{n-1} に $f(R_{n-1}, K_n)$ を2を法として加えたものなのですから，R_n からそれを2を法として引いてやればいいわけです．これを16回繰り返してやりますと元へ戻り，最後に結果として L_0 の次に R_0 がつながったものが出てきますので，それに初期転置の逆の転置を施してやると元に戻るというのが仕掛けです．

もっとも，この関数 f の値に相当するものを求める際にはいろいろなことをやります．一番核心部分ではSボックスと呼ばれるものを使って換字を行います．それが非線形になっているというのがミソです．この関数 f の仕組みは図-7 のようになります．まず，R に拡大転置を施します．これは，単に，重複を許して32ビットを48ビットに展開します．鍵のほうは，図-8 のようにして少しずつシフトさせては48ビットを作り出します．この48ビットどうしの排他的論理和を6ビットずつに切ってそれぞれを4ビットに換字します．この換字の仕組みは規格に書いてあるのですが，作り方が発表されていないので，当初からもめていたのですが，どうも線形ではないらしい．この換字をやったあとでごしゃごしゃと転置を施した32ビットが結果になるわけです．この32ビットがごしゃごしゃになっているので，解読がさらにやりにくくなっているわけです．

DES に関して分かってきたいろいろな性質などに関しては文献2) などをご覧になっていただきたいと思います．

DES の使い方

DES の使い方に関しても，同じように規格がありまして，4通りあります．1番目として ECB モード

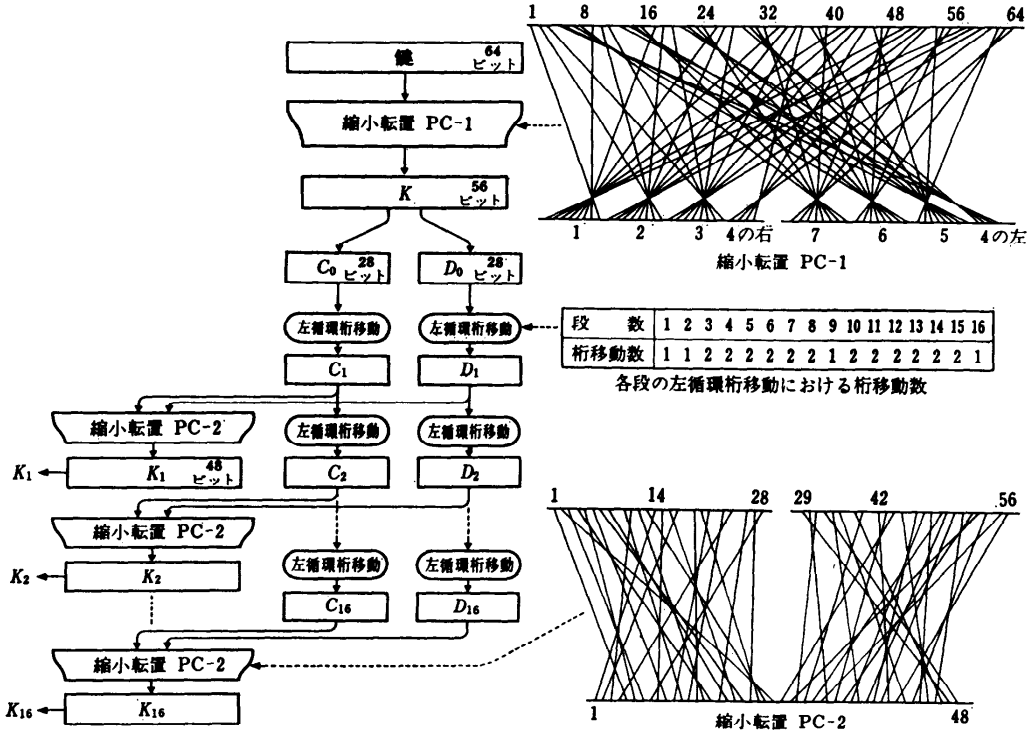

図-8 K_n の生成

(Electronic Code Book mode) つまり電子コードブックモードというのがあります．これが基本のモードで，何もしない，要するにそのまま使うというものです．2番目は CBC モード (Cipher Block Chaining mode) と呼ばれるもので，これはメッセージが途中で改ざんされていないことのチェック，それをメッセージの認証といいますが，そのために使うのがよいとされているものです．メッセージ全体をコンパクト化したものを認証子とするのですが，32ビットにするものにチェイスマンハッタン方式というのがあります．それから，CFB モード (Cipher Feed Back mode) と OFB モード (Output Feed Back mode) というのがありますが，これらはともに，DES を乱数発生器として使い，先ほどのバーナム暗号のような感じにしようとするものです．CFB モードのほうは自己同期方式の，OFB モードのほうは外部同期方式の逐次暗号になります．

慣用の暗号系の欠点

それでは，慣用の暗号系の欠点ですが，まず，何よ

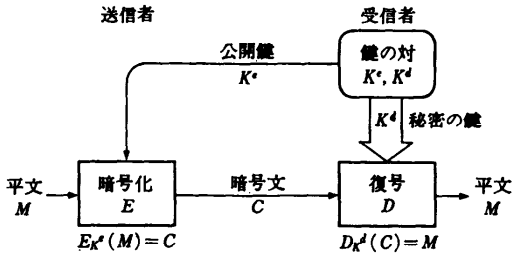

図-9 公開鍵暗号系の概念

りも，鍵を送るための安全な通信路が必要なわけです．それともう一つ，送受信する者どうしごとに異なった鍵が必要になります．つまり送受信する者の二乗のオーダで鍵が増えていくことになります．

公開鍵暗号系

これらの点を改善するといった意味で，1976年に W. Diffie と M. E. Hellman によって提案されたものが公開鍵暗号系です．これの基本は，復号の鍵と暗号化の鍵とが異なるということです．そして，復号の鍵を暗号化の鍵から類推することつまり計算することが

困難であるというような作り方をしたものです.

その概念図は図-9 のようになります. 受信者がまず暗号化の鍵と復号の鍵になる一対の鍵 K^e と K^d を作ります. そして, 暗号化の鍵 K^e を送信者に渡し, 復号の鍵 K^d は秘密にしておきます. 送信者がこの受信者に対してメッセージを送るときには平文 M を暗号化の鍵 K^e を使って手続き E で暗号化して送ると, 受け取ったほうでは秘密の鍵 K^d を使って復号の手続き D で元の平文に戻すというような仕掛けになっているわけです. このとき, K^e から K^d を類推するあるいは計算することが難しいという前提に立っておりますから, 鍵 K^e は電話帳のようなもので公開してしまうことすらできますので, K^e を公開鍵と呼んだりいたします.

このようにしますと, 慣用の暗号系の欠点でした秘密の通信路で送る必要がなくなります. 実はそれほど簡単ではないということがいろいろ分かってきてはいますが, 原則的にはなくなります. それから, この受信者に送るときにはだれもがこの暗号化の鍵 K^e を使えばいいわけですので, 1人に1個ずつ用意すればいいことになり鍵の数がきわめて少なくてすむことになります.

それでは, このような公開鍵方式が成り立つ条件ですが, まず, すべての M に対して $E(M)$, $D(E(M))$ が存在して $D(E(M))=M$ が成り立つ必要があります. 次に, E, D の手続きが比較的簡単でなければなりません. それから, E から D を割り出すことが, 実際上不可能である必要があるわけです.

そのためのアプローチですが, まず, 暗号化の必要条件の(1)と(2)を満たすために一方向性関数というものを利用します. これは, ある関数 f の定義域内の x に対して $y=f(x)$ を計算することは容易にできるが, $y=f(x)$ を満たす y が与えられたとき $x=f^{-1}(y)$ を計算することが実際上不可能な関数です. ところが, これを素直に利用しますと復号もままならなくなりますので, 暗号の必要条件(3)を満たすための仕掛けを設けます. この仕掛けを落し戸, この仕掛けを設けたものを一方向性落し戸関数といい, 仕掛けは鍵を作るときに仕組んでおきます.

RSA

このアイデアが発表されたときには絵に画いた餅だったのですが, 1978年になりまして R. Rivest, A. Shamir, L. Adleman という3人によって具体的な方法が考案されました. その方法はこの3人の名前の頭文字をとって RSA 法と呼ばれていますが, その方法がいまもってもっとも代表的な公開鍵暗号方式になっています. RSA 法では, メッセージを大きな整数に直して, これを大きな, およそ200桁程度の法の元でべき乗します. これは, 大きな整数の素因数分解は非常に手間がかかるという事実を解読のしにくさの根拠にしています. したがって, これも無条件に安全な暗号ではなく計算的に安全だろうというものです.

この RSA 法では公開鍵として二つ用意します. それを e と n とします. 実は, n は大きな二つの素数の積になっており, これがおよそ200桁程度の数になるのです. そして秘密の鍵を一つ用意します. これを d とします. 暗号化は, 平文 M をサイズ n のブロックに分割し, そのブロック単位に, n を法として e 乗します. 復号は, やはり n を法として d 乗するわけです. このとき, 両方を通じて平文 M が受ける変換は $D_d(E_e(M))=(M^e \bmod n)^d \bmod n=M^{ed} \bmod n$ ということになりますが, これが M になるような仕掛けがしてあるのです.

その e, d, n を決定する手順は, まず, 大きな素数 p と q を任意に選び $n=p\cdot q$ とします. 次に, $\phi=(p-1)\cdot(q-1)$ を求め, これと互いに素で, これよりも小さな任意の数 e を決めます. そして $d\cdot e\equiv 1 \pmod{\phi}$ を満たす d を求めればいいのです. このような e, d, n を用いると $M^{ed}\equiv M \pmod{n}$ となるのです. これの厳密な証明はオイラーの関数とフェルマーの小定理を使えばわりと簡単に行えるのですが, 時間の関係もありますので, ご興味のある方は文献6)をご覧になっていただければと思います.

これを解読するには秘密鍵 d がいります. d を知るためには二つの素数 p と q が必要です. $n=p\cdot q$ ですから n を p と q に素因数分解しなければなりません. ところが n が200桁ぐらいになると10億年くらいかかりそうだから, 大丈夫だろうというわけです.

ディジタル署名

さて, 他にもいろいろとお話したいことはたくさんありますが, 時間がだんだんなくなってきましたので, 公開鍵はこれくらいにして, 次の話題であるディジタル署名に移らせていただきたいと思います.

これは, ディジタル信号として送られるメッセージに, なんらかの形でサインをしようというもので, ディジタル情報に適用できる認証方式としては, このメッ

セージは確かにあの人から送られたものであると認証するユーザ認証つまり身元の確認と，このメッセージは送信途中で改ざんされていないということを確認するメッセージ認証の二つがあります．そのための手順ですが，まず，送信者は“秘密の鍵”でメッセージを暗号化します．この暗号化したものを署名文といいます．その署名文を受信者に送りますと，受信者は署名文を復号し，送信者の身元および改ざんの有無を確認するわけです．

ところで，ディジタル署名の3条件と呼ばれているものがあります．(1)署名文が第三者によって偽造できないこと，(2)署名文が受信者によって偽造できないこと，(3)送信者があとになって署名文を否定できないこと，の三つです．

慣用の暗号系の場合には，鍵が一つしかありませんから，3条件の(1)は満たされますが(2)は満たされないことになります．ましてや(3)は，いわゆる公証人役場みたいなものを作り，そこでタイムスタンプを押してもらうようなことをしなければなりません．

それでは，公開鍵でディジタル署名を可能にする条件というのはどういうことになるかといいますと，公開鍵は公知になっていますが，相棒の秘密の鍵はだれか特定の人ひとりが握っているわけですから，その秘密の鍵を使って暗号化したものが，もし公開鍵で復号できるものなら，その暗号文を作れるのは，その相棒の秘密の鍵を握っている人しかいないという理由に基づくわけです．したがって，すべての M に対して $D(M)$, $E(D(M))$ が存在して，$E(D(M))=M$ が成り立つことというのが条件になります．

このような認証付きの，つまりディジタル署名付きのメッセージを送るには，まず，送信者であるAさんがメッセージ M を平文で作り，自分の秘密の鍵 K_A^d を使って復号の手続き D で $D_{K_A^d}(M)$ と変換して M に“署名”します．これをそのまま送ってもよろしいのですが，秘密にしてBさんに送るものとしますと，Bさんの公開鍵 K_B^e を使って暗号化の手続き E で $E_{K_B^e}(D_{K_A^d}(M))$ と暗号化して送ります．これを C としますと，受け取ったBさんは自分の秘密の鍵 K_B^d

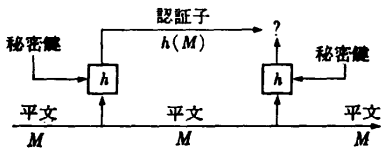

図-10 認証子照合法

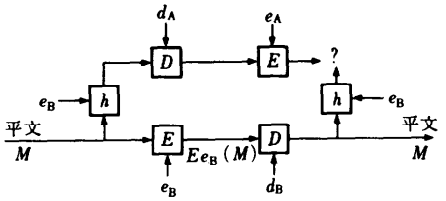

図-11 認証子照合法（公開鍵方式の場合）

を使って復号の手続き D で $D_{K_B^d}(C)$ と復号するとAさんの“署名文” $D_{K_A^d}(M)$ になるはずです．そこで，これをAさんの公開鍵 K_A^e を使って暗号化の手続き E で $E_{K_A^e}(D_{K_A^d}(M))$ と変換しますと元の平文 M がめでたく出てくるはずです．無事出てきたときにはAさんからのメッセージだというのが公開鍵の場合のからくりです．

これは RSA でできます．Aさんは平文 M を自分の秘密の鍵 d_A を使って復号の手続き D で $D_{d_A}(M)=M^{d_A} \bmod n=C$ と署名文を作ります．受信者であるBさんはAさんの公開鍵 e_A を使って暗号化の手続き E で $E_{e_A}(C)=E_{e_A}(M^{d_A} \bmod n)=M^{d_A e_A} \bmod n$ と復号し，元の平文 M を得ることによって認証できるわけです．

メッセージの認証はどうするかといいますと，たとえばメッセージと署名文とを別々に送り，受信者がメッセージから署名文を生成し，送られてきた署名文と突き合わせることによって行うようなことをします(図-10 参照)．このとき，この署名文を認証子といいます．これを公開鍵で，しかも認証子も平文も暗号化してやるには図-11 のようにすればいいわけです．このような認証子，つまり長いデータをビット数の少ない認証子に圧縮する変換を圧縮暗号化などともいいます．先ほどの DES ですと CBC モードはこの手に使おうというわけです．それ以外にも Rabin の方式とか Davies の方式などがあります．

鍵の配送

ところで，公開鍵の場合には一応公開できますから，鍵を特に配送する必要はないわけですが，慣用の暗号系の場合にはどのように配送すればよいかという問題が残っています．通常，軍事・外交の場合には密使（アタッシャ）を使ったり，ビジネスの場合には書留郵便とか電話が使われているようです．

DES と RSA では変換速度がずいぶん違いますので，DES で使う秘密鍵を RSA 暗号で送り，大量の

データはDESで暗号化しようという方法とか，DESの鍵をMarkelとHellmanが考案したPKDS (public key distribution system, 公開鍵配送法) という離散対数に根拠を置いた白日のもとで鍵を送る方法を用いて送り，大量のデータはDESで暗号化しようとする方法などが考えられております．

暗号技術の動向

まだ，お話しなければいけないことはたくさんあるのですが，ほとんど時間がなくなってまいりましたので，動向を最後にまとめさせていただきたいと思います．

まず，慣用の暗号系ですが，米国の規格であるDESが広く使われております．しかし，DESが発表されて10年になりますので標準局では安全性の保証を1988年で打ち切るといいだしました*．それにともなって，いろいろなところであおりを食らっております．まず，DESを国際規格にしようとISOの場でDEA 1 (Data Enciphering-Specification of Algorithm 1) という名のもとで審議されていたのですが，ISつまり国際規格にするための投票をしようという段階にまできて米国が規格にするのはやめにして暗号は登録制度にしようといいだしたのです．その結果，1986年10月にISとして出版することを中止にし，1987年末に向けて登録制度を審議しております．利用モードはISとして出版されています．JISは，私が主査を仰せつかり，1985年にDEA 1のDISを元に規格案を作り始めたのですが，こういう事態になったものですから，1987年に規格にするはずだったのを中止して棚上げにしてあります．

ところで，米国では現在，NSAつまり国家安全保障局でCCEP (Commercial Communications Security Endorsemant Program) つまり商用通信安全保障計画という計画のもとで秘密裏に暗号系作りが行われているようです．しかも，今度は一切公開されることはないようです．

日本のほうでは，NTTの宮口さんたちが作られたFEALという名の暗号系があります．16ビットのパソコンでプログラム規模がDESの6分の1の390バイト，暗号化の速度がDESの75倍，そして解読のしにくさはDES以上といわれております．これは，DESで16段繰り返したような仕組みを4段使い，その前後にインボリューションをかまし，核心部はアミダ構造といったような構造になっているものです．

公開鍵暗号系のほうもいろいろ発表されております．秘密通信と認証の両方ができるものとしては，RSA以外にもRabinの方法，Williamsの方法，横浜国大の松本・今井さんたちの方法，ElGamalの方法などがあります．秘密通信のみですと，代表的なものにMerkle-Hellmanの方法があります．しかし，これは超増加数列を埋め込んだナップザック問題を使ったものですが，1984年にShamirに破られております．このほか，Chor-Rivestの方法とか東工大の辻井先生の方法などがあります．また，認証通信のみですとGoldwasser-Micali-Rivestの方法などがあります．

ところで，RSAもDEA 2として国際規格にしようと審議されていたのですが，こちらのほうも登録制にしようということになったようです．

また，OSIにもセキュリティメカニズムを入れようと，OSIの各レイヤごとで行う安全保障も検討されておりまして，DISの投票の締め切りが今年の12月18日になっております．

どうも駆け足で十分な説明ができなかったかと思いますが，この分野にご興味のある方は参考文献としてあげておきましたようなものをご覧になるとよろしいかと思います．

それでは，時間がまいりましたので，この辺で終わらせていただきたいと思います．どうも，ご清聴ありがとうございました．

* 金融業界など民間からの要求であと5年延長された．

参考文献

1) 土居範久，小山謙二編：コンピュータ・セキュリティ，共立出版 (1986).
2) 池野信一，小山謙二：現代暗号理論，電子情報通信学会 (1986).
3) 暗号と情報セキュリティの基礎と応用，専門講習会講演論文集，電子情報通信学会関西支部 (1986).
4) 情報処理，Vol. 25, No. 6 (1984).
5) 土居範久，広瀬 健，西村恕彦：公衆暗号系，情報処理，Vol. 22, No. 1, pp. 38-46 (1981).
6) 土居範久，広瀬 健，一松 信，西村和夫：公衆暗号系の実現可能性と問題点，情報処理，Vol. 22, No. 1, pp. 38-46 (1981).