〔非公開〕

TR－C－0018

暗号研究の現状

手塚　集
SYU TEZUKA

1988. 10. 11.

ＡＴＲ通信システム研究所

暗号研究の現状

目次

O．はじめに

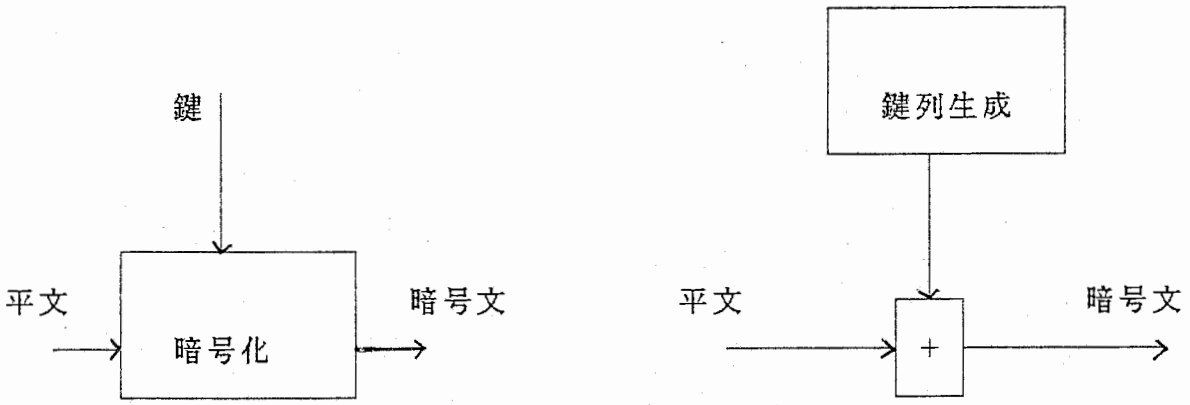
この報告書は、昭和６１年１１月から昭和６３年１０月迄の２年間、筆者がＡＴＲ通信システム研究所のセキュリティ・グループで行った暗号研究の際に調べた事柄を整理したものである。

全体は、３部にわかれている。第１章では、暗号方式についてまとめた。従来からの分類に従って、ブロック暗号とストリーム暗号の２方式に分けて研究の現状を整理した。ブロック暗号では、秘密鍵方式としてＤＥＳ、公開鍵方式としては素因数分解の難しさに基づくＲＳＡ方式および離散対数計算の難しさに基づくＥｌＧａｍａｌ方式を採り上げた。素因数分解と離散対数計算はこの分野ではよく利用される代表的問題である。ストリーム暗号に関しては、ヨーロッパで伝統的に研究されている非線形シフトレジスタ方式を取り上げた。また、最近話題になっている暗号的に強い乱数についても簡単にまとめておいた。第２章では、暗号プロトコルという題名の下に、鍵配送・管理及び認証方式についてまとめた。鍵配送・管理は、大規模なネットワークでは、重要なテーマである。鍵配送・管理センターを設ける場合の方式として、現在研究の盛んなＩＤ－ｂａｓｅｄ方式のうち、岡本（栄）鍵配送方式を採り上げて説明した。また認証方式としては、メッセージ認証・個人識別・デジタル署名についてＺｅｒｏ－Ｋｎｏｗｌｅｄｇｅ－Ｉｎｔｅｒａｃｔｉｖｅ Ｐｒｏｏｆも含めてまとめた。認証は、最近普及し始めたＥＦＴ（電子資金移動）技術などで重要な役割を担っているので、将来不可欠な技術である。しかし、暗号技術も含めてこの分野は、いま盛んにいろいろなアイデアが提案されている状況なので、数年で消えてしまうようなものも含めたかもしれない。また、こういう種類の技術は実際に世の中で使われなければ意味がないという点では、使いやすさや標準化にもっていくための諸々の要因のほうが、理論的なエレガントさなどよりはるかに重要であろうから、今後の理論的発展がどの程度実用となる技術に貢献するのか判断しにくい所がある。最後に、第３章では、インプリメンテーションということで、ＲＳＡのＶＬＳＩ化アルゴリズムとＧＦ（2^n）上の高速演算アルゴリズムについてまとめた。

Ｉ．暗号方式

通信のセキュリティが今後重要な研究テーマになるといろいろなところで聞かれるようになってきている。このセキュリティという言葉の含む内容は非常に広く、法律のような制度的な面やリスク分析のような経営的な側面もある。技術という面では、通信ネットワークの管理技術やデータ伝送技術にいかにセキュリティを導入するかが重要な研究テーマになる。信頼性という意味でのセキュリティは、誤り訂正技術やフォールト・トレラント技術のように、もうすでに研究も行われ、実用に供されているものも数多くあるが、それに比べ、もうひとつのテーマである人為的な災害に対するセキュリティ、このような犯罪ともよべる災害にたいする防御技術は、未だ十分確立されたとは言い難い。しかし、将来のインフラストラクチャアとも言われている大規模通信ネットワークにおいては、極く僅かの潜在的犯罪者のためにネットワーク全体の安全が脅かされることも起こりうる為、そのような災害に対する防御技術は不可欠となっている。このような防御技術のひとつに暗号がある。この技術はかつては戦争のような非常時か、スパイや漁船のような限られた環境でしか使われていなかったが、上のような背景から、一般社会での実用を目指した研究が最近急速に活発化してきている。

伝統的に、暗号方式は、ブロック暗号とストリーム暗号の２方式にわけて議論されることが多い。２方式は、次の図に示される。



ブロック暗号　　ストリーム暗号

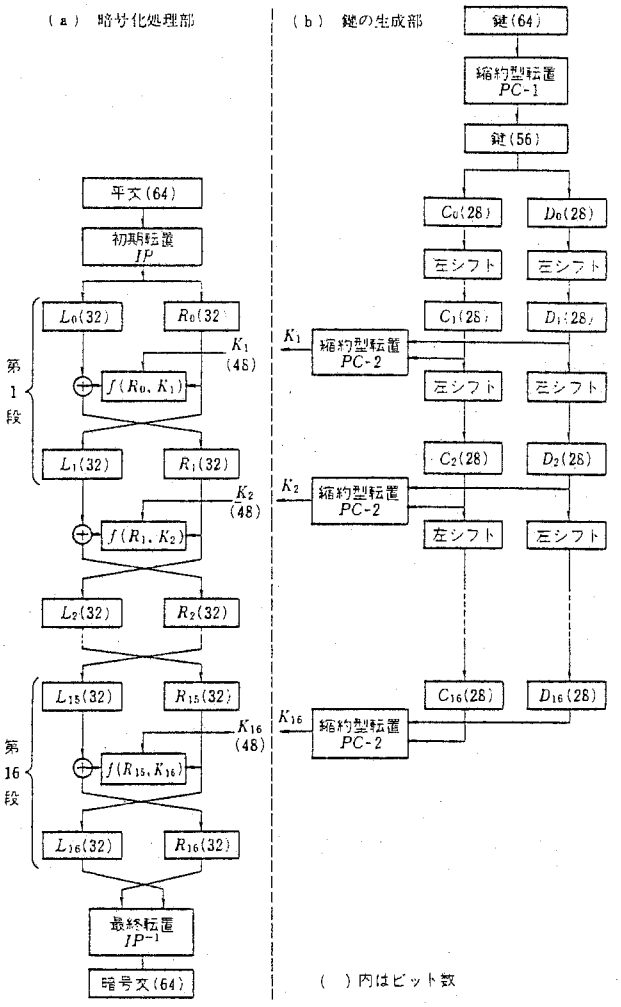
大きな違いは、ブロック暗号では、同一の平文ブロックが同一の暗号文ブロックに変換されるのに対して、ストリーム暗号では、そうはならない点である。実際には、ブロック暗号でも、チェインをかけて（ＣＢＣモード）前のブロックに依存して暗号文が生成されるような方式が採られたりしているので区別は難しい。

１．ブロック暗号

ブロック暗号には、秘密鍵方式と公開鍵方式がある。秘密鍵方式は、暗号化鍵と復号化鍵が同一となるもので、そのため鍵を安全に配送するという問題を生じる。公開鍵方式は、暗号化鍵と復号化鍵が異なり、暗号化鍵を公開するので鍵の配送という手間がなくなる。

１．１．秘密鍵方式

１９７６年に公開鍵方式が提案されるまでは、秘密鍵方式しかなかった。ＤＥＳ（Ｄａｔａ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）はそれまでの秘密鍵方式に関する理論の積み重ねと、当時最新のエレクトロニクス技術のうえに作り上げられた方式で、これがＵＳＡの標準になったのが１９７７年である。注意すべきことは、標準として決定される前に行われたＤＥＳの安全性に関するワークショップで、本方式の安全性の保証期限が１０年後の１９８７年までと結論された事実である。実際、現在（１９８８年）、ＮＳＡは新しい暗号化方式（ＣＣＥＰ）の開発に入っている。また、ＮＢＳはＤＥＳの保証期限を５年延長したが、その後どうなるのかはわかっていない。一方、日本にはＮＳＡに相当する機関もなく、標準化どころか安全な暗号化方式の本格的研究すら殆ど行われていないのが現状である。以下、ＤＥＳについて簡単にまとめておく。



１．１．１．ＤＥＳ方式

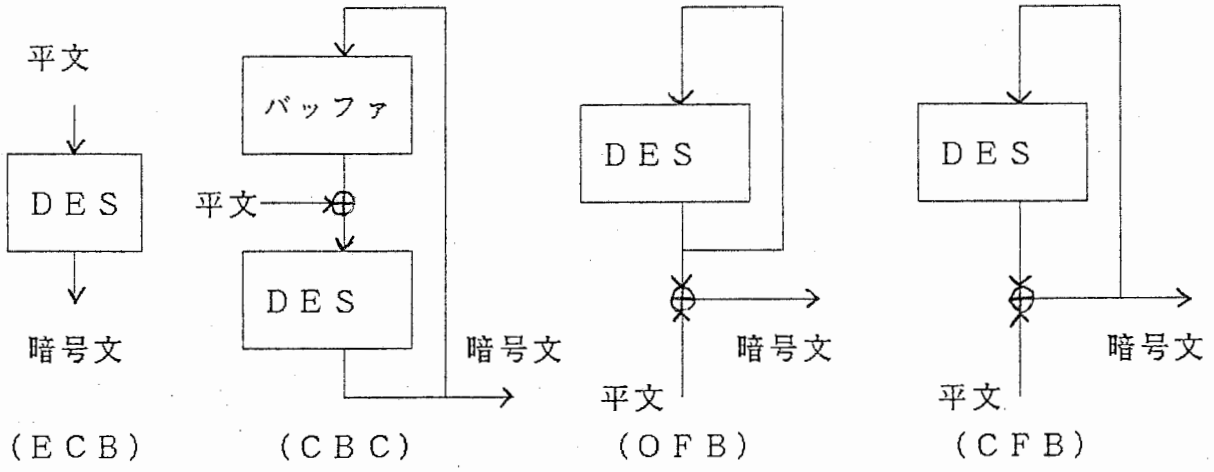
ＤＥＳのブロック図を次に示す。図からもわかるように、基本的には、換字と転置からなるブロック（ｆ（・），ＩＰ，ＰＣ－２）が繰り返されているだけである。繰り返しの段数は、実験的に１６段で十分と決められた。換字の部分は、Ｓ－ｂｏｘと呼ばれ、６ビット入力対４ビット出力の非線形変換部でこの部分が暗号強度を高めていると言われている。

（１）ＤＥＳの利用モード

ＮＢＳは、次のような４つのＤＥＳの利用モードを提案している。

１．ＥＣＢモード（基本モード）
２．ＣＢＣモード（暗号文連鎖モード）
３．ＯＦＢモード（出力フィードバックモード）
４．ＣＦＢモード（暗号文フィードバックモード）

このうち３と４は、ストリーム暗号に分類できる。通常、ＥＣＢモードは短いデータ（６４ビット以下）例えば、パスワード等に用いられる。ＣＢＣモードの考え方は、ＤＥＳ以外のブロック暗号にも適用できる。エラーの伝播という問題はあるがよく用いられている。このモードはｃｈｅｃｋｓｕｍが自動的に生成できるので認証子の生成にも使える。ただ６４ビットのブロックが単位になるため、ｔｒａｎｓｐａｒｅｎｃｙにかける。ＯＦＢモードは、６４ビットモード以外では、周期の点で問題があるため、安全性に欠ける。ただ原理的には、ｏｎｅ－ｔｉｍｅ－ｐａｄなのでよく用いられている。ＣＦＢモードは、ｃｈｅｃｋｓｕｍを生じるので認証にも使えるし、また１ビットモードは、データ通信に広く使われている。しかし、理論的解析は非常にむずかしい。



（２）ＤＥＳの安全性

ＤＥＳの安全性に関する議論の中で、よく採り上げられるのが、Ｓ－ｂｏｘの設計基準である。ＤＥＳに非線形性と多対１変換を導入して解読を難しくする為に考えられたのがこのＳ－ｂｏｘであるとされているが、パラメーターの決定がどのようにしてなされたのかが判っていないため、落とし戸が仕組んであるのかもしれないと疑っている人がいる。また、５６ビットという鍵の長さにも問題があるといわれている。これは、そもそもＩＢＭが提案した最初のバージョンが１２８ビットの鍵長を持っていたのに何故短くしたのかが不明であることからきている。

１．２．公開鍵方式

公開鍵方式のｉｄｅａは１９７６年、ＤｉｆｆｉｅとＨｅｌｌｍａｎが提案したもので、それを具体的に実現したのがＲＳＡ方式である。基本的な考え方は、暗号化鍵と復号化鍵を異なったものにし、暗号化鍵は、公開してしまうというものである。このようにすることで鍵を秘密に配送する必要がなくなる。この考えを実現するために必要となるのが一方向性関数とよばれるものである。一方向性関数とは、順方向（暗号化）の計算は容易であるが、逆方向（復号化）の計算は難しいような関数である。

ＲＳＡ方式は、大きな２つの素数の積の素因数分解の難しさを利用したものでこの２つの素数の値を知らなければ逆方向の計算は難しいような関数を利用して作られた。厳密には、この場合の一方向性関数は、素因数分解の難しさという落とし戸のついた関数とよばれる。次節で説明するＥｌＧａｍａｌ方式は、離散対数計算の難しさを落とし戸とする一方向性関数による例である。

１．２．１．ＲＳＡ方式

ＲＳＡの原理は次の通りである。公開鍵は（ｅ，Ｎ）の組である。ここでＮは２つの異なる大きな素数ｐ，ｑの積とする。平文Ｍ，暗号文Ｃとすると、暗号化は、

$$C = M^{e} \pmod{N}$$

となり、復号化は、

$$M = C^{d} \pmod{N}$$

となる。ここで、ｐ，ｑ，ｄの値が復号化のための秘密鍵になる。Ｍの値によらないｄの値をもとめるためには、

$$e \cdot d = 1 \pmod{L}$$

を解くことが必要になるが、Ｌ＝ＬＣＭ（ｐ－１，ｑ－１）の値を計算する為には、ｐ，ｑの値を知らなければならない。ところが、Ｎの値からｐ，ｑを計算するのは、Ｎが大きい時は難しいので、結局、ｐ，ｑを知らずにｄを求めることはできないことになる。つまり、ｐ，ｑがこの方式の落とし戸になっている。ここで、$M^{L} = 1 \pmod{N}$となるＭは、ｇｃｄ（Ｍ，Ｎ）＝１の時に限るが、$M^{e \cdot d} = M \pmod{N}$は（Ｎが２つの異なる素数ｐ，ｑの積ならば）すべてのＭについて成り立つ事に注意したい。したがって、平文Ｍは任意にとれる。また、暗号化鍵ｅは復号を一意に行えるようにするため、Ｌと互いに素となっている。

ｅ＝２とした場合は、Ｒａｂｉｎの方式と呼ばれるもので、この方式は、解読の難しさが素因数分解の難しさと同等であることが証明されている。

（１）ＲＳＡ方式の鍵の選び方

この方式の強度は、大きな２つの素数の積の素因数分解の難しさに依存しているのでｐ，ｑの選び方は、重要である。しかし、素因数分解を用いなくともｐ，ｑの選び方によっては、平文をもとめることができることをシモンズとノリスは示した。

そこで、そのような欠点を避け、かつ素因数分解も難しいようなｐ，ｑの選び方がリベストらによって以下のように提案されている。

（ｐ，ｑの選び方）

１．｜ｐ－ｑ｜が大きい。

２．ｇｃｄ（ｐ，ｑ）が小さい。

３．ｐ＝ａ’ｐ’＋１，　ｑ＝ｂ’ｑ’＋１，
ｐ’＝ａ’’ｐ’’＋１．　ｑ’＝ｂ’’ｑ’’＋１
ここで、ｐ’，ｑ’，ｐ’’，ｑ’’は、それぞれ10^{90}程度の素数。

（２）共通のＮを用いる場合

複数の利用者が共通のＮ、異なるｅを用いる場合、２人以上が同一の平文を暗号化するとｐ，ｑが計算できることがシモンズによって指摘されている。

（３）特殊な暗号文

（ａ）暗号文がｇｃｄ（Ｃ，Ｎ）≠１かつｇｃｄ（Ｃ，Ｎ）≠０（ｍｏｄ　Ｎ）となる時は、ユークリッドの互除法によりｐ，ｑが計算できる。

（ｂ）特殊な暗号文は平文と同一となる。つまり、$M^{e}=M$　（ｍｏｄ　Ｎ）となる場合がＭ＝０，１以外に存在する。

（ｃ）特殊な暗号文は、秘密鍵ｄの値より、はるかに小さい値で巾乗してもとに戻る場合がある。

（４）最下位ビットの安全性

ＲＳＡ暗号文から平文の最下位ビットを求めることがどの程度難しいかを検討したものである。もし、平文の最下位ビットを求めることが易しければ、平文全部を求めることも同様に易しいという結論であった。逆の言い方をすると平文の最下位ビットを求めることは、ＲＳＡを解くのと同じ位難しいことになる。ここで、「易しい」とは、確率的多項式時間で解けるという意味である。この議論は、２．２節の暗号的に強い乱数とも関連する概念なので重要である。

この議論は、２つの点に分けて考えられる。

（ａ）ＲＳＡ解読と平文の最下位ビットを求めることの同等性
これは、ＲＳＡ暗号文が与えられた時平文の最下位ビットを出力するサブルーチンを考えることで証明できる。

（ｂ）ＲＳＡ解読の難しさとＲＳＡ暗号文のランダム性
１／２＋ｅの確率で正しく平文の最下位ビットを出力するサブルーチンを考える。１／ｅが平文の長さに関する多項式なら、つまり確率１／２（ランダム）との違いが多項式なら、ＲＳＡ解読を確率的多項式時間でできるという結果が知られている。逆の言い方をすると、ＲＳＡ解読は難しいことから、ＲＳＡ暗号文を与えて平文の最下位ビットを確率１／２＋ｅで正しく出力するできるようなサブルーチンはないことになる。つまり、確率的多項式時間のサブルーチンを使う限り、ＲＳＡ暗号文は、ランダムと区別ができないということになる。

（５）素因数分解

アルゴリズムとしては、複数多項式２次ふるい法、楕円曲線法、連分数法等のようなヒューリスティックな方式が有効な方式であるといわれている。どちらもＮのけた数をｎとすると、素因数分解の計算量は平均的に、

$$O(\exp((n\log n)^{1/2}))$$

となるといわれている。このオーダーは、ｓｕｂｅｘｐｏｎｅｎｔｉａｌといわれほぼ、指数オーダーの計算量である。また楕円曲線法では、Ｎがおおきな２つの素数の積の時が最悪の場合で、この時、上の計算量に一致するといわれている。

ＲＳＡの解読のような場合の素因数分解には、複数多項式２次ふるい法が有効であると考えられる。この方式は、連分数法と同じアプローチをとっており、簡単に説明すると次のようになる。

$$X^2 = Y^2 \pmod N$$

なるＸ，Ｙを求め、もしｇｃｄ（Ｘ－Ｙ，Ｎ）が１でもＮでもなければ、それが素因数となる。Ｘ，Ｙを探すのに、連分数法では連分数展開を用いるが、この方式では、複数の２次多項式を用いて効率をよくしている。

１９８８年１０月１１日現在では、４００台のコンピューターを２６日間走らせて１００桁（４１桁と６０桁）の数が素因数分解できたという報告がある。２００桁ぐらいの数では、10^{11}ドルあれば一年でできるという人もいる。

（６）素数判定

秘密鍵ｐ，ｑを求めるのに、９０桁ぐらいの素数を見つけなければならない。そのために必要になるのが素数判定である。よく用いられるのが次に示すラビン法である。

（　ラビン法　）

ｍが奇数で、$N = 2^{s} m + 1$で表されているとする。

１．〔２，Ｎ－１〕からランダムに整数ａを選ぶ。

$$y = a^{2^{i} m} \quad (\mathrm{mod}\ N)$$

と定義する。

２．ｉ＝０とおき、ｙを求める。

３．もし、ｙ＝Ｎ－１なら（５）へ行く。
もし、ｉ＝０かつｙ＝１なら（５）へ行く。
もし、ｉ＞０かつｙ＝１なら（６）へ行く。

４．ｉ＝ｉ＋１として、

もし、ｉ＜ｓなら、$y = y^{2} \mathrm{mod}\ N$として（３）へ行く。
もし、ｉ＝ｓなら、（６）へ行く。

５．「Ｎは素数又はａを底とする強擬素数である。」

６．「Ｎは合成数である。」

この方法は、ｎをＮのけた数とすると、確率的多項式時間（n^{3}）の合成数判定ラスベガスアルゴリズムである。ここでラスベガスアルゴリズムとはもしその判定でｙｅｓとでた場合は、確率１で正しいことが保証されたアルゴリズムである。この方法の計算時間（１回の判定にかかる時間）は、９０桁ぐらいの数で数秒である。

また最近、楕円曲線を用いた確率的多項式時間（n^{2}）の素数判定ラスベガスアルゴリズムが提案されて話題になっている。決定的なアルゴリズムとしてはコーエン・レンストラ法がある。この計算量は、

$$O(n^{c \log(\log n)})$$

といわれている。ｌｏｇ（ｌｏｇ　ｎ）は、殆ど一定とみなせるのでこの計算量はほぼ多項式時間と考えられる。

１．２．２．ＥｌＧａｍａｌ方式

この方式は、次のようなものである。受け手の公開鍵は（ｐ，Ｂ，Ｙ）の組で、秘密鍵はｋとする。ここで、Ｙ（＝B^{k}），ｐは大きい素数とする。

送り手は、受け手の公開鍵Ｙと自分で生成した乱数ｒを用いて、暗号文Ｃを

$$C = Y^{r} \cdot M \pmod{p}$$

として作成し、受け手に暗号文として（B^{r}，Ｃ）の組を送る。

受け手は、

$$M = C / (B^{rk}) \pmod{p}$$

を計算して、平文を再生する。ここで、ｋの値をＢ，Ｙから求める問題は、離散対数計算と呼ばれ、一般に難しいことが知られている。この場合は、ｋの値そのものが、逆変換の落とし戸になっている。

（１）特徴

暗号文のサイズが平文のサイズの２倍になる。同一のｒを用いると既知平文攻撃が可能になる等の短所がある。基本的には、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの公開鍵配送方式の変形と見做せる。

（２）離散対数問題

有限体ＧＦ（ｑ）において原始根をｇとすると、与えられたａに対して

$$a = g^{x}$$

となるｘを求める問題を離散対数問題という。

この問題の解法には、大きく２通りある。１つは、ポーリック・ヘルマンの方式で、ｑ－１が小さい因数のみからなる時に有効である。ＧＦ（ｑ）でｑのけた数をｎとすると、その計算量は、

$$O(n^{2})$$

となる。

$q - 1 = s \cdot 2^{t}$，ｓは奇数，と書く時、この方式を用いるとｘの最下位ｔビットを求めることができる。特に、ｑが大きい素数ではｔは１以上なので少なくとも最下位１ビットは確実に求まる。

もう１つがアドルマン－ウォータールーによる方式である。このアルゴリズムは、２段階になっており、まず第１段階で、必要なデータベース（小さい素数のみを因数とする数の表）の作成が行われる。第２段階では、乱数ｒを発生して、

$$b = g^{x} g^{r} = g^{x+r}$$

を計算し、データベースに存在するようなｂが現れるまで繰り返す。そのようなｂが見つかればｘ＋ｒの値がわかるのでｘが求まることになる。

処理の手間は、第１段階が多くかかり、ＧＦ（ｑ）でｑのけた数をｎとすると、その計算量は、

$$O(\exp((n \log n)^{1/2}))$$

となることが知られている。この手間は、桁数ｎの素因数分解と同じ計算量である。第２段階の手間は、上の量の平方根ぐらいになると言われている。

また、ＧＦ（2^{n}）では、コッパスミスによる改良の結果、計算量は、

$$O(\exp(cn^{1/3} \cdot (\log n)^{2/3}))$$

となった。

具体的には、ＧＦ（ｐ）ではｐは十進２００桁程度の素数、ＧＦ（2^{n}）ではｎは１０００程度であれば現状では安全である。一般に、素体のほうが拡大体より難しいといわれている。

また、ａと原始根ｇが与えられた時、ｐが素数（即ち、素体）ならば、

$$a = g^{s} \pmod{p},\ 0 \leqq 2s < q-1,$$

となるかどうかの判定ができれば、平方根の計算を繰り返し行うことにより、離散対数問題は多項式時間で解けることが知られている。（拡大体でも平方根の計算が容易であればこの話は成り立つ。）

筆者が、かつて提案したデジタル動画像暗号化方式は、このＥｌＧａｍａｌ方式を拡張したものである。この場合、平文がデータ系列　$M_1\ M_2\ M_3 \ldots$，となっており、暗号文　$C_1\ C_2\ C_3 \ldots$ がそれぞれ、

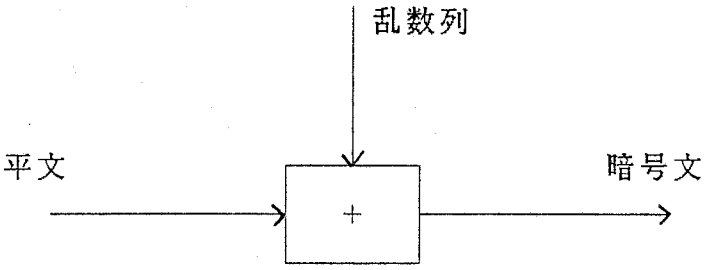
$$C_i = Y^{r_i} \cdot M_i \pmod{p}$$

として作成される。受け手には、この暗号文系列とB^{r}が送られる。この方式は、データ毎に鍵がかわっていくので、「公開鍵方式のストリーム暗号」とみなすことができる。

2．ストリーム暗号

暗号文に対して鍵のしらみつぶしをした時任意の有意な平文が等確率で生成されるならば、その暗号方式は、ｐｅｒｆｅｃｔ　ｓｅｃｒｅｃｙ（原理的に安全）であるといわれる。また、暗号文がある長さｎを越えると鍵のしらみつぶしをした時一意に平文が求まるなら、ｎをその暗号方式の判別距離とよぶ。ｎは、暗号方式と言語の冗長性に基づいて決まる量であるので、暗号強度の尺度の１つであるが、鍵のしらみつぶしを仮定している、つまり計算量的限界を無視しているのですべての暗号方式に有効に働くわけではない。

原理的に安全な暗号化方式として有名なのがＳｈａｎｎｏｎのｏｎｅ－ｔｉｍｅｐａｄである。この方式は、真にランダムな数列を仮定している点で実用性にかける。下図に方式を示す。



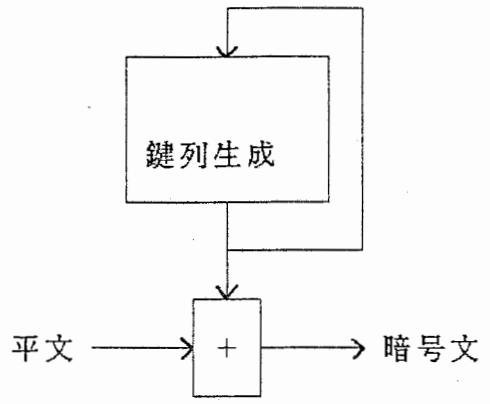
Ｓｈａｎｎｏｎのｏｎｅ－ｔｉｍｅ　ｐａｄ

真にランダムな数列を擬似乱数に置き換えて、系列を０―１の２値系列に絞った方式が、バーナム暗号と呼ばれる最も実用的な方式である。しかし、擬似乱数を用いた事によって原理的な安全性は失われることになる。ただし、鍵の変更を頻繁に行えば、十分大量の乱数を「１回限り」で用いることになり、「真にランダムな数列」に近づく。それでも擬似乱数が「決定論的」に作られているという点が問題として残る。そこで、擬似乱数生成のメカニズムを鍵として、この鍵を知らないものには真にランダムな数列と区別がつかないような擬似乱数生成方法が研究のテーマとなる。ここで「区別」とは、何かを考えなければならない。まず、考えられるのが従来からある有限列のランダム性のための統計的テストである。これらは全て、真にランダムな数列であることを仮説としてたてた棄却テストであるのでどのテストも真にランダムであるための必要条件にすぎない。更に、これらのテストは生成された系列そのものを見ており、どのようなメカニズムで系列が生成されたかは直接考慮されていない。暗号では、特に生成メカニズムが重要なのでこのメカニズムの複雑さ（ランダムさ）を大きくする必要がある。以下でのべる「線形複雑度」は、そのようなメカニズムの複雑さを測る尺度の１つである。また真にランダムな数列との「区別」を行うのに必要となる手間の量に着目したのが２．２でのべる暗号的に強い乱数である。

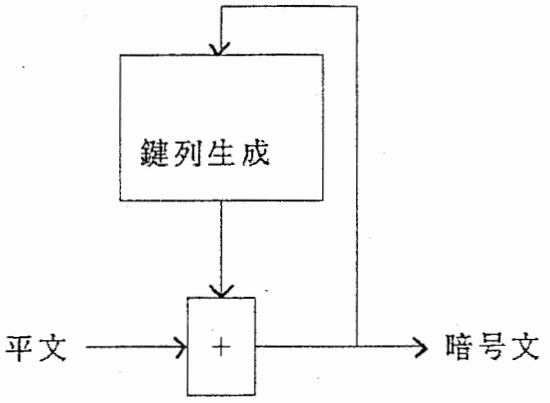
2．1．シフトレジスタ方式

2．1．1．方式の概略

一般に、ストリーム暗号には、次のような2通りの方式がある。1つは、同期型（ＯＦＢ）でもう1つが自己同期型（ＣＦＢ）である。



ＯＦＢモード



ＣＦＢモード

自己同期型は、暗号文がフィードバックするため、暗号強度の解析は、極めて難しい。一方、同期型では、鍵列が平文に独立しているので解析しやすい。また、上で述べたｏｎｅ－ｔｉｍｅ－ｐａｄまたはバーナム暗号と形式的に一致しているので安全性解析に向いている。

実際には、ストリーム暗号の鍵列生成には、シフトレジスタを用いる方式が広く使われている。この方式は、シフトレジスタ部と非線形変換部から構成されており、非線形変換部の出力がシフトレジスタ部にフィードバックするかどうかでフィードバック方式とフィードフォワード方式に分けられる。フィードバック方式は、自己同期型のストリーム暗号に用いられることが多い。ここでは、最近特に研究の盛んなフィードフォワード方式について考える。この場合、シフトレジスタ部は、通常ｎ個の線形フィードバック・シフトレジスタ（ＬＦＳＲ）からなっている。非線形変換部は、インプリメンテーションの方法でみるといろいろあるが、一般的に表せば、2値のｎ変数関数は、ＧＦ（2）上で

$$F(x_1, \ldots, x_n) = a_0 + \sum a_i x_i + \cdots + a_{1\cdots n} x_1 \cdots x_n$$

として定義される。この時、非線形変換部への入力パターンの周期が出力列の周期の最大実現可能な値である。もし、入力パターンがその周期中に同一のものを含まなければ、ｎ変数関数をうまく選ぶことでその周期と同一長のすべての系列を生成できる、という事実は重要である。

２．１．２．強度解析及び設計方法

有限の２値系列は、すべてある長さの線形フィードバック・シフトレジスタで生成できる。この事実から与えられた系列を生成する最小段数の線形フィードバック・シフトレジスタの長さを解読の難しさの目安にしようという考え方が起こった。この考えは、実際の解読法とも一致していたので広く用いられるようになった。この尺度は、今日では、「線形複雑度」と呼ばれている。更に、この考え方を発展させたのが「線形複雑度プロファイル」というもので、これは従来の「線形複雑度」が２値系列全体のみを注目していたのに対し、部分の「線形複雑度」も含めて考えるというものである。この「線形複雑度プロファイル」に基づくランダム性のテストが、最近では研究されてきている。ともかくこの考え方では、「線形性」を特別視して、「非線形」を「線形」で置き換えてその「非線形性」を「線形複雑度」として捉えようと云うものである。注意すべき性質としては、「周期」は常に「線形複雑度」より大きいかまたは等しいという事実である。従って、「線形複雑度」を十分大きくとれば、「周期」も自動的に十分大きくなる。

その他、０・１バランス、相関、等の暗号強度の目安が提案されている。ここで相関とは非線形変換部のｎ入力と出力の間の相互相関のことである。ｎ入力を、i_1，．．．，i_n、出力をｏで表すと、ｋ次無相関は、$1 \leqq j_1 < \ldots < j_k \leqq n$で

$$Pr(o ; i_{j1}, i_{j2}, \ldots, i_{jk}) = 1/2$$

を充たすことである。特に、ｋ＝０の時が０・１バランスに対応している点は注意を要する。Ｓｉｅｇｅｎｔｈａｌｅｒの結果から、ｋ次相関と線形複雑度の間には、トレードオフがあることが知られている。具体的には、次の関係式で表せる。

$$k + d \leqq n, \qquad 0 \leqq k \leqq n-1,$$

また、出力が０・１バランスを満たす場合は、

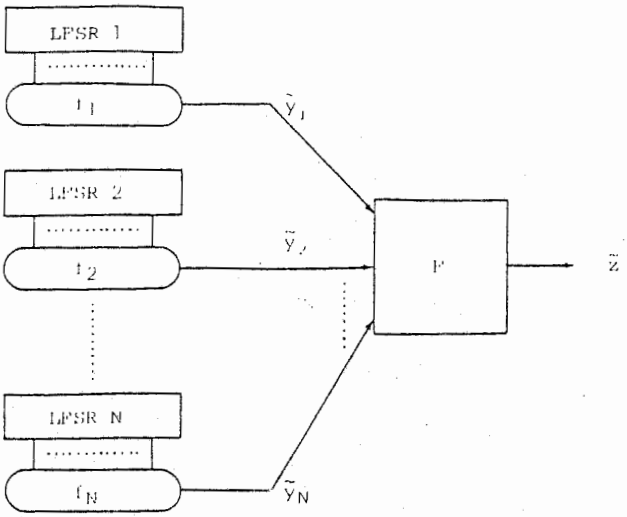
$$k + d \leqq n-1, \qquad 1 \leqq k \leqq n-2,$$

となる。ここで、ｋは無相関の次数、ｄは非線形の次数である。

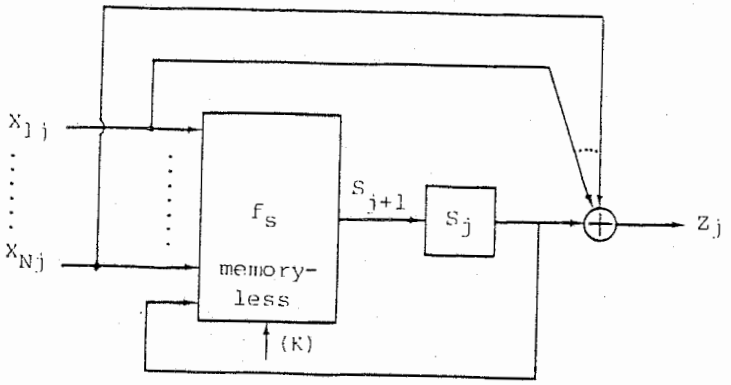
最近、メモリのついた非線形変換部を考えることにより、このトレードオフは解消できる事がＲｕｅｐｐｅｌ及びＳｉｅｇｅｎｔｈａｌｅｒによって示された。つまり、ｎ－１次無相関で非線形次数がｎとなるような系列の生成が可能となる。

これら以外に、ランダム性の統計テストとして知られる連検定、系列相関、ｋ－均等分布などのテストも当然暗号強度の目安として必要となるが、その詳細は、他に譲ることにする。

以上のような種々の暗号強度の基準を満足するように非線形変換部を設計することがこの方式におけるメインの研究テーマである。Ｒｕｅｐｐｅｌは、非線形変換部がメモリを持つ場合と持たない場合に分けて、次のような構成を提案している。

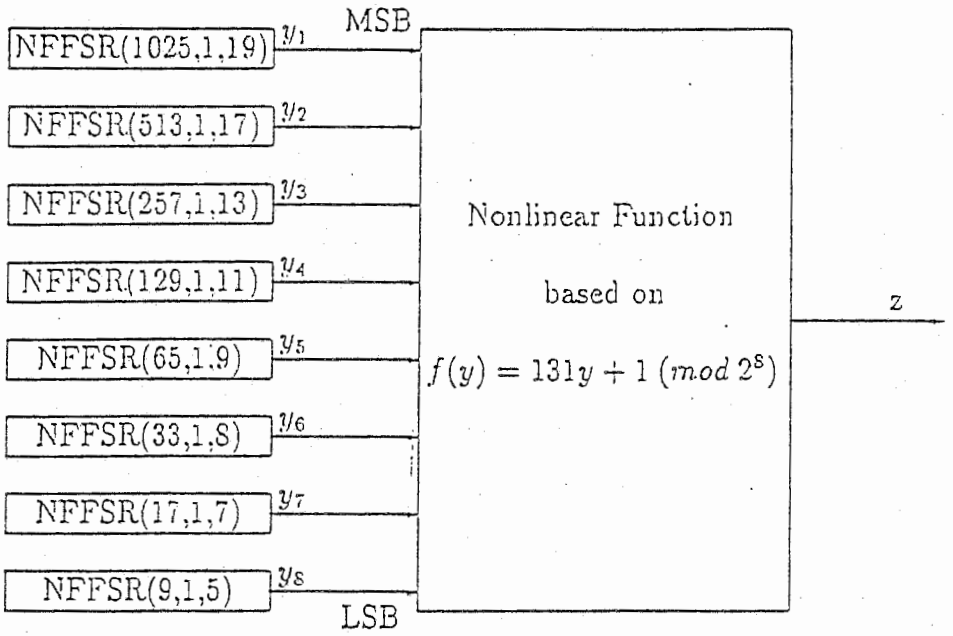


メモリを持たない場合



メモリを持つ場合

ここで問題になるのは、非線形変換部をどのように合成するかである。当然、暗号強度も強くなければ成らないが、多くの異なる内部構造の非線形変換部をなるべく容易に合成できるような方法が必要となる。筆者が提案した方式は、この問題に対する１つの解答である。この方法は、加算や乗算のような整数演算を組合せて、非線形変換部を合成するもので、整数演算を用いるため暗号強度の解析も比較的しやすいという特徴がある。次の例は、この方式に従って作成したものである。ここで非線形変換部は入力ビット・パターンを整数ｘとみなしａｘ＋ｃ（ｍｏｄ　2^n）の最上位ビットを出力としている。



この方式では、加算と乗算を各１回しか用いないので高速な鍵列生成が可能となる。

２．２．暗号的に強い乱数

暗号的に強い乱数という考え方は、Ｓｈａｍｉｒ，Ｙａｏらによって提案されたものでつぎのようなものである。

（　暗号的に強い乱数　）

ビット列$b_1\ b_2\ \ldots\ b_m$において、任意のｉに対して、b_iを$b_1\ b_2\ \ldots\ b_{i-1}$から、確率的多項式時間で計算できなければ、ビット列は暗号的に強いと言う。

上のような性質は、ビット列$b_1\ b_2\ \ldots\ b_m$の「予測不可能性」とよばれる。そして、Ｙａｏは「多項式時間統計的テスト」の概念を定義し、次の定理を得た。（ただし、証明は未だに発表されていない。）

（　定義　）

ビット列$b_1\ b_2\ \ldots\ b_m$が多項式時間統計的テストをすべてパスするとは、ビット列が真にランダムなビット列といかなる確率的多項式時間アルゴリズムを用いても区別できないことである。

（　Ｙａｏの定理　）

ビット列$b_1\ b_2\ \ldots\ b_m$が「予測不可能性」をもつことと、ビット列が多項式時間統計的テストをすべてパスすることとは同値である。

ここでは、次の平方剰余判定問題が重要な役割を果たすので先に述べておく。

（　平方剰余判定問題　）

$N=pq$としてｐ，ｑは異なる奇素数とする。Z^*_NをZ_Nの乗法群とすると、Z^*_Nの要素のうち丁度半分がＪａｃｏｂｉ記号が＋１で残りが－１になる。前者を$Z^*_N(+1)$，後者を$Z^*_N(-1)$で表す時、$Z^*_N(+1)$の丁度半分だけが平方根を持ち、それぞれに対し平方根は４つ存在する。また、Z^*_Nの任意の要素に対して、Ｊａｃｏｂｉ記号の値は、Ｎの素因数がわからなくても多項式時間で計算可能である。（有名なガウスの相反定理を繰り返し用いる。）

平方剰余判定問題とは、$Z^*_N(+1)$の要素が与えられたときそれが平方剰余かどうかを判定する問題である。この問題は、素因数分解と同程度に難しいことが知られている。特に、$p=q=3\ (\mathrm{mod}\ 4)$の場合、ＮはＢｌｕｍ整数と呼ばれており、このとき各平方剰余はその４つの平方根のなかに、丁度１つ平方剰余を持つことが知られている。

これに関連して、平方根そのものを求める問題がある。Ｎの素因数がわかっていれば各因数毎に解くことになるが、任意の素数ｐ，平方剰余ｃに対して、

$$x^2 = c \quad (\mathrm{mod}\ p)$$

を解く問題は確率的（ラスベガス）ではあるが高速な解法が知られている。

２．２．１．確率的暗号化法

暗号的に強い乱数という考え方とは異なるが、多項式時間内では解読不能という点では共通する方式が、ＧｏｌｄｗａｓｓｅｒとＭｉｃａｌｉにより提案された。彼らは、ＲＳＡ（一般には、公開鍵方式全般）の問題点として、

（１）どんな平文に対しても安全とはなっていない。

（２）平文の部分情報まで安全であるわけではない。

の２点を挙げ、その解決として確率的暗号化法なるものを考えた。

この方法は符号化率が増えるが、そのかわり平文の情報を１ビットも洩らさず、どのような平文に対しても、安全性が保証できるような暗号化方式であると彼らは主張している。しかし、その符号化率増大の為に実用性は全くなく、単に理論的興味をそそるにすぎない。

具体的な例として、次のような方式が提案された。

（準備）

ｋビットの素数ｐ，ｑを選び、Ｎ＝ｐｑとする。Z^*_N（＋１）の要素である平方非剰余ｙを選ぶ。（Ｎ，ｙ）を公開し（ｐ，ｑ）を秘密にする。

（暗号化法）

ＢがＣにメッセージ$b_1\ b_2\ \ldots\ b_m$を送りたいとする。そのとき、

各々のb_iに対し、Ｂはランダムにｘを選び、もし$b_i = 1$なら、

$e_i = y x^2 \pmod N$、さもなければ$e_i = x^2 \pmod N$として、

Ｂは$e_1\ e_2\ \ldots\ e_m$を送る。

この場合、メッセージの１ビットがＮの長さに拡大されることになる。

（復号法）

Ｃは、Ｎの素因数を知っているので、e_iが平方剰余かどうか容易に計算できる。

この方式の安全性は、Ｎが十分大きい時の平方剰余判定の困難さに基づいている。基本的に平文の冗長なｎビットをランダムなｎｋビットに拡大するという暗号化法は、この方法以外にもＷｅｙｎｅｒの方式等でも使われているが、逆に、データ圧縮では、冗長な平文をいかにより短いランダムなビットに変換するかがテーマになっており、またＳｈａｎｎｏｎのｉｄｅａｌ　ｓｅｃｒｅｃｙの基本でもあることから、このようなデータ拡大による暗号化は、実用的にも理論的研究の方向としてもあまり関心しない。但し、上の方式は、有限の鍵長で任意の長さの平文に対して多項式時間内安全性が保証できる点は面白い。これに対して、次に述べる暗号的に強い乱数では、用いるビット列の長さが入力長（鍵長）の多項式サイズという制限がついているので有限長の平文にしか使えない。

２．２．２．Ｂｌｕｍらの方式

この考え方は上の確率的暗号化法とＳｈａｎｎｏｎのｏｎｅ－ｔｉｍｅ－ｐａｄを組み合わせたようなもので、次のような方式である。

（準備）

ｋビットの素数ｐ，ｑを選び、Ｎ＝ｐｑとする。ｎを公開し、（ｐ，ｑ）を秘密にする。ＮはＢｌｕｍ整数とする。

（暗号化法）

ＢがＣにメッセージ$b_1\ b_2\ \ldots\ b_m$を送りたいとする。そのとき、Ｂはまずx_1を任意に選び、$x_{n+1} = x_n^2 \pmod N$に従って、$x_1, x_2\ \ldots\ x_m\ x_{m+1}$を計算する。$k_i =$ ｐａｒｉｔｙ（x_i）を求め、各々のb_iに対し、$c_i = b_i + k_i \pmod 2$を計算して、Ｃに、$c_1\ c_2\ \ldots\ c_m$とx_{m+1}を送る。ここでｍは入力の多項式サイズとする。

（復号法）

Ｃは、Ｎの素因数を知っているので、x_{m+1}から$x_m\ \ldots\ x_1$を容易に計算できるので、メッセージ$b_1\ b_2\ \ldots\ b_m$が復号される。

この方式は、ｎの素因数を知らないと（１）x_{i+1}からx_iが計算困難、及び（２）$N - x_i$とx_iがx_{i+1}の平方根であることがわかってもどちらが平方剰余か決めるのは困難、つまり、ｐａｒｉｔｙ（$N - x_i$）とｐａｒｉｔｙ（x_i）は異なるのでk_iを決めることはできない、という２つの事実に基づいている。

本方式の実用性は、かなり高い。現状のハードウェアで実現すれば５０ｋｂｐｓぐらいは達成できる。また見方によっては、この方式は公開鍵方式によるストリーム暗号とみなせる。ただ「公開鍵方式によるストリーム暗号」は既存の公開鍵方式を鍵配送に用いた既存のストリーム暗号でも実現できるので大して重要な性質ではない。むしろ、周期性、線形複雑度、ランダム性といった性質の方が重要である。彼らは、この方式は多項式時間統計的テストはすべてパスするという意味でランダムであると主張しているが、「多項式時間統計的テスト」が具体的に何を意味するのかよくわからない。通常の統計的テストとどう違うのか、それらを含むのか含まれるのか、よくわからない。ビット列の長さｍも、入力の長さの多項式という条件がついているが指数オーダーだとどうなのか、よくわからない。「通常の統計的テスト」を殆どパスするには、「並列的多項式時間統計的テスト」を全てパスしなければならないと主張している人もいるので、このあたりをきちんと整理する必要があると思う。

ＩＩ．暗号プロトコル

１．鍵管理・配送

暗号のセキュリティは、鍵の管理にかかっている。秘密鍵方式では、鍵配送という問題も生じる。公開鍵方式においても大規模なネットワークになると、公開鍵ファイルの安全な管理は重要なテーマである。いずれの場合にも、ユーザーからは独立した中立的立場の鍵管理・配送センターの存在が必要になる。したがって、多数のユーザーと鍵管理・配送センターという図式のなかで鍵の問題を扱うことになる。

１．１．ＤＥＳを用いた鍵管理・配送

秘密鍵方式であるＤＥＳを用いた大規模ネットワークにおける鍵管理・配送の問題を考える。秘密鍵方式では一般に、ｎユーザーがそれぞれのペアに対し、異なる鍵を用いるとした時、全体でｎ（ｎ－１）／２個の鍵が必要になり、各ユーザーは、ｎ－１個の鍵を管理しなければならない。これでは、ｎが大きくなると非常にわずらわしい。そこで考えられたのが「セッション鍵」を用いる方式である。この方式では、Ａ，Ｂは、それぞれ「端末鍵」とよばれる「セッション鍵」より一段上位の鍵を予め所有している。そしてＡ，Ｂ間に通信が行われる時、それに先立って「セッション鍵」が鍵管理・配送センターから送られる。この時、鍵配送に使われるのが「端末鍵」である。「端末鍵」は「セッション鍵」の暗号化のみに用いられ、実際の通信の暗号化には、この送られてきた「セッション鍵」が使われることになる。したがって、各ユーザーは、自分の「端末鍵」のみを管理すればよいことになる。また、「セッション鍵」は、１つのセッションのみで使い棄てるため安全性も高まる。　一方、鍵管理・配送センターは、全ユーザーの「端末鍵」を知っていなければならないので、ある程度負荷を負うことになる。

（１）ＴＲＭの使用

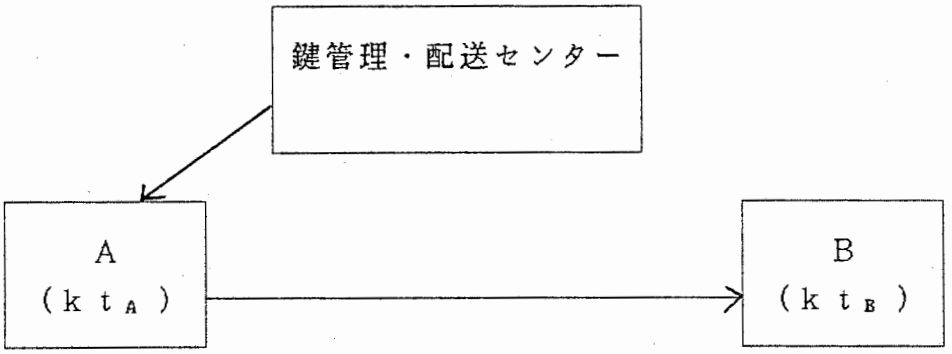
ＴＲＭ（Ｔａｍｐｅｒ　Ｒｅｓｉｓｔａｎｔ　Ｍｏｄｕｌｅ）は、その名の通り鍵を物理的に保護するための堅固なモジュールである。ＤＥＳを用いたシステムでは、重要な鍵はすべてこのＴＲＭに記憶されており、また一度記憶されると二度と呼び出せないようになっている。さらに暗号化操作そのものもこのＴＲＭの内部で行われるように設計されている。

（２）「端末鍵」の管理・配送

秘密鍵方式では、最低一回はネットワークを通して、安全に鍵配送を行わなければならない。この場合「端末鍵」の配送がそれにあたる。この配送は、物理的におこなわざるを得ず、盗聴を防ぐため光学的手段も用いられている。また、鍵のロードも手作業になるので安全性に対する配慮が必要になる。鍵管理・配送センターは、「端末鍵」・「セッション鍵」を「マスター鍵」で暗号化して保管し、「マスター鍵」そのものは、ＴＲＭの中に保管する。各ユーザーも、自分の「端末鍵」を保管するためにＴＲＭを使用する。

（３）「セッション鍵」の配送

ＡがＢに対して通信を行おうとする場合を考える。よく用いられるのは次のような方式である。まず、Ａは鍵管理・配送センターから「セッション鍵」を受けとり、その後、ＡはＢに「セッション鍵」を配送する。以上を図で示すと次のようになる。



鍵管理・配送センターは、ＡにE_{ktA}（ｋｓ，E_{ktB}（ｋｓ））を送る。Ａは、自分の端末鍵ｋｔ$_A$で復号しセッション鍵ｋｓを受け取り、残りのE_{ktB}（ｋｓ）をＢに送る。

（４）鍵管理・配送センター及び端末間の認証

上記の「セッション鍵」の配送を行うにあたっての問題の１つは、第３者が鍵管理・配送センターになりすますことである。これは、以前に行われた通信文を再使用する等の方法により可能である。また、第３者が端末Ａになりすまして「セッション鍵」を不正に入手する場合も起こりうる。しかし、後者の場合は第３者がＡの「端末鍵」を知らなければ原則的に不正はできない。従ってここでは、１方向の認証で十分である。一方、端末Ａ，Ｂ間での「セッション鍵」の受け渡しでは、互いに相手を確認しまた、以前の通信文との区別をするため双方向の認証が必要である。

いずれにしても、相手認証の技術がここでは、必要になってくる。暗号プロトコルとしての認証技術については、次節にまとめる。

１．２．Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの方式

公開鍵配送方式の考え方と具体例は、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの有名な論文の中で、公開鍵暗号方式の考え方と共に提案されたものである。基本的な考えは安全でない通信路を介して共有の秘密鍵を生成しようというものである。彼らの具体例は、ＧＦ（ｐ）上の離散対数問題に基づくもので以下のような方式である。

Ａ，Ｂ間で秘密鍵方式の暗号通信を行うものとして通信に先立って、共有鍵を生成する場合を考える。予め、素数ｐとＧＦ（ｐ）上の原始根ｇを公開しておき、Ａは、整数ａをランダムに選び、

$$k_a = g^a \quad (\mathrm{mod}\ p)$$

を計算し、Ｂにk_aを送りａを秘密にする。Ｂは、

$$k_b = g^b \quad (\mathrm{mod}\ p)$$

を計算し、Ａに送りｂを秘密にする。その後、Ａ，Ｂ両者は、共有鍵として

$$k_{ab} = g^{ab} \quad (\mathrm{mod}\ p)$$

をそれぞれで生成する。

第３者がk_a，k_bを盗聴したとしても、この値から、k_{ab}を算出する手間は離散対数問題を解くことに一致するので、ｐが十分大きな素数である時、１．２．２．で述べたように計算量が大きくなり解読は困難である。但し、第３者がＡＢ間で、k_a，k_bの受け渡しの際、能動的にこれらの値をかいざんすると、異なった共有鍵が生成されることになって、安全な通信は行なえなくなる。例えば、整数ｃを用いてＡ，Ｂにk_a，k_bの代わりに、

$$k_c = g^c \quad (\mathrm{mod}\ p)$$

を送るとＡ，Ｂとそれぞれk_{ac}，k_{bc}を共有することになり盗聴が可能となる。この方式はセンターを仮定せず、当事者のみで行う鍵配送プロトコルである。暗号方式は秘密鍵方式になっているため、大規模なネットワークでは共有鍵の数がふえるので管理が難しくなる。

因みに、１．２．２．で述べたＥｌＧａｍａｌ方式は、上記の公開鍵配送方式を基にして考えられた公開鍵暗号方式である。

１．３．ＩＤ－Ｂａｓｅｄ鍵配送方式

大規模なネットワークでは、鍵配送・管理センターが必要になるが、センターにどのくらいの権限をもたせるかは、問題である。１つの考え方にＩＤ－Ｂａｓｅｄ方式と呼ばれるものがある。

基本的にＩＤ－Ｂａｓｅｄ方式とは、ユーザーがそのＩＤ情報（各ユーザーの名前や住所、等）に対してセンターの署名（お墨付き，ｃｅｒｔｉｆｉｃａｔｅ）をもらい、それを用いて、暗号・鍵配送・認証・署名などの処理を行おうというもので、Ｓｈａｍｉｒによって１９８４年に提案された。暗号・鍵配送・認証・署名のそれぞれに対して、ＩＤ－Ｂａｓｅｄ方式が提案されているが、暗号に対する「ＩＤ－Ｂａｓｅｄ方式」はあり得ないと主張している人もいる。

ここでは、鍵配送に対する「ＩＤ－Ｂａｓｅｄ方式」として、岡本方式を採り上げ説明する。本方式は２段階で構成されている。第１段階は、ユーザー加入で、管理センターが各ユーザーに個別秘密情報s_iと全ユーザー共通情報のＮ，ａ，ｅを渡す。ここでＮは、２つの大きな素数の積である。ａはＧＦ（ｐ），ＧＦ（ｑ）での原始根、ｅは（ｐ－１）（ｑ－１）と互いに素な数とする。ユーザーｉの個別秘密情報s_iは、

$$s_i = ID_i^{-d} \pmod N$$

で与えられる。ここで、$ed = 1 \pmod{(p-1)(q-1)}$である。管理センターは、ｄを秘密にしておく必要がある。

第２段階では、ユーザーｉ，ｊは、各々乱数r_i，r_jを生成し、

$$x_i = s_i \cdot a^{r_i} \pmod N$$

$$x_j = s_j \cdot a^{r_j} \pmod N$$

を相手に送る。各ユーザーは相手のＩＤを用いて、

$$K_i = (x_j^{e} \cdot ID_j)^{r_i} \pmod N$$

$$K_j = (x_i^{e} \cdot ID_i)^{r_j} \pmod N$$

で共通鍵を生成する。ここで、$K_i = K_j$となるところがこの方式のみそである。この方式の長所は、ユーザーは秘密鍵r_iを管理センターとは自由にとれる点である。また、ユーザーの結託によってセンターの秘密情報がもれない点も大きな長所になっている。基本的な構成は、先に述べたＤｉｆｆｉｅ－Ｈｅｌｌｍａｎのものと非常によく似ている。

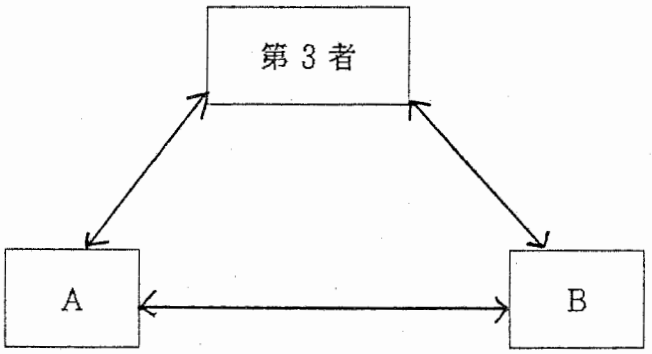
2．認証、識別、署名

ここでは、Ｆｉａｔ－Ｓｈａｍｉｒの定義にしたがって、次の３項目で考える。

1．Ａｕｔｈｅｎｔｉｃａｔｉｏｎ：Ａは、Ｂに自分がＡであることを証明できる。しかし、第３者がＢに自分がＡであることは証明できない。

2．Ｉｄｅｎｔｉｆｉｃａｔｉｏｎ：Ａは、Ｂに自分がＡであることを証明できる。しかし、Ｂが第３者に自分がＡであることは証明できない。

3．Ｓｉｇｎａｔｕｒｅ：Ａは、Ｂに自分がＡであることを証明できる。しかし、Ｂが自分自身に自分がＡであることは証明できない。



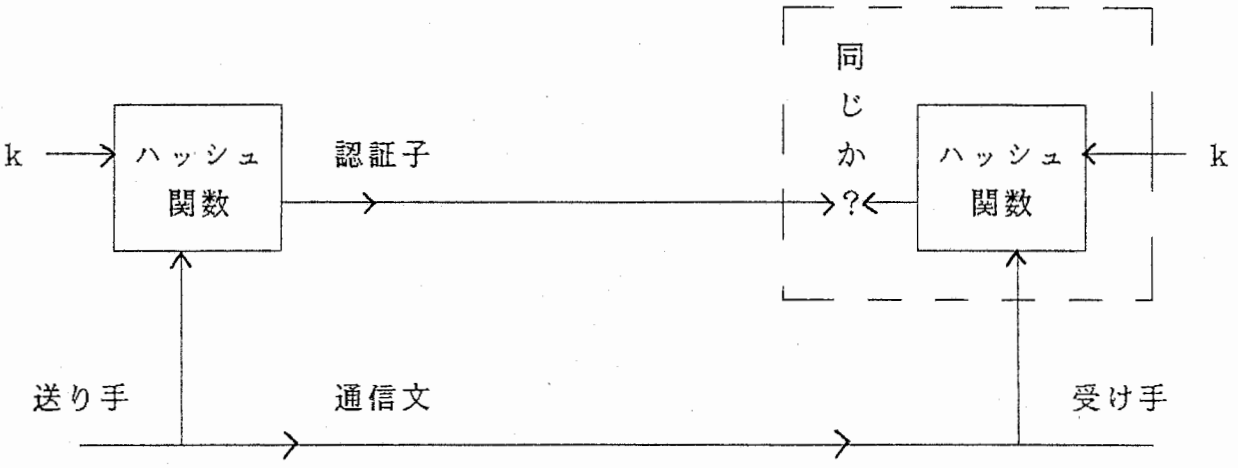
Ａｕｔｈｅｎｔｉｃａｔｉｏｎ（認証）は、Ａ・Ｂ間に信頼関係が存在する時にのみ有効な手続きである。Ａ・Ｂ間に紛争が生じると２及び３のＩｄｅｎｔｉｆｉｃａｔｉｏｎ（個人識別）とＳｉｇｎａｔｕｒｅ（デジタル署名）が必要になってくる。個人識別はインタラクティブな手続きである。違いは些細で後の調停に提出する証拠の偽造可能性の有無である。個人識別では、そのような偽造の可能性があるが、デジタル署名では無い。但し、実際の応用では、リアルタイムに偽装を見破り、応答やサービスを中止することを目的とすることが多いため、２でも３でもどちらでも使える。

2．1．認証

データのＩｎｔｅｇｒｉｔｙ（完全性）は、データ通信の重要なテーマである。誤り訂正技術は、そのために開発された技術であるが、これもメッセージ認証の１つといえる。メッセージ認証は、送信者の送ったメッセージが途中改ざんされたり、ノイズで内容が乱されたりすることなく受信者に届くようにするための技術である。ここでは、狭い意味で改ざんを防ぐ技術ととらえて整理する。誤り訂正技術との違いは、送信者・受信者間に秘密のパラメーターがあるかどうかである。誤り訂正技術では、方式は公開されるので秘密のパラメーターは存在しない。認証技術にはもう１つの機能として送信者認証があるが、送信者・受信者間に秘密が共有されれば、この働きも果たしていることになる。

通信文の秘密鍵による暗号化は、１つの認証方法である。共通の秘密鍵で復号して意味の或るメッセージが生じれば、途中改ざんが無かった可能性が高い。また送信者認証の働きも同時にしていることになる。しかし、通信文全体の暗号化は、実際上大きな手間がかかるため、通常は認証子による方式が広く用いられている。

次の図は、メッセージ認証の典型的な例である。通信文全体を縮約する関数がＨで通常ハッシュ関数とよばれる。ハッシュ関数Ｈは、送信者・受信者に共通な関数で、送信者・受信者のみが知る秘密鍵がついている。通信文全体とは別にハッシュ関数の出力が認証子として受信者へ送られる。受信者は、受け取った認証子が自分で計算したものと一致するかどうかを調べて途中改ざんが無かったかどうかを調べる。



メッセージ認証の原理

また、これは秘密鍵を共有しているので送信者認証にもなっている。

実際に用いられているハッシュ関数としては、ＤＥＳや公開鍵方式で議論されている１方向性関数などがある。一般にそのみたすべき性質としては、次のようなものが考えられる。

（１）Ｈは、ブロック毎に独立に計算するのではなく、通信文すべてのビットに依存した方法で計算する。

（２）任意に与えられた平文Ｍ，Ｘに対して、Ｈ（Ｘ）＝Ｈ（Ｍ）となる確率は非常に小さい。

（３）Ｈは、Ｈ（Ｘ・Ｙ）≠Ｈ（Ｘ）・Ｈ（Ｙ）を充たす。（・は演算）

（４）Ｈ（Ｍ）は高速に計算できる。

特に、（２）の条件は圧縮率と密接に関連している。また、誕生日攻撃法と呼ばれる偽造法の為、たとえ（２）の条件がある程度充たされていても認証子のビット数が短い（３２ビットぐらい）と安全でなくなる。

２．２．個人識別

上で述べた認証では、すでに鍵が共有された後の手続きを考えていた。ここでは、それ以前の段階及び送信者・受信者間に信頼関係のない場合を想定する。以下、手続きはすべてインタラクティブに行われるとする。

２．２．１．Zero-Knowledge Interactive Proof

ある秘密の知識をその内容は一切知らせずにその知識を自分がもっているという事実のみを相手につたえるためのインタラクティブな手続きをＺｅｒｏ－ｋｎｏｗｌｅｄｇｅ　Ｉｎｔｅｒａｃｔｉｖｅ　Ｐｒｏｏｆ（ＺＫＩＰ）という。基本的には、ＢがＡに任意に選んだ問題を提出し、Ａはそれに答えるという形をとる。Ｂが任意に問題を作るのでＡは予め答えを準備することはできない。また、Ｂが意図的に秘密を探れるような問題を提出してもＡが答えしか述べなければ、Ｂにとって新しい情報は得られないようにしなければならない。

その具体例を次にしめす。この例は、ＲＳＡ暗号において、自分は秘密鍵の所有者であることを秘密鍵の値を相手に知られずに示そうというものである。方式は、平方剰余判定と素因数分解が同程度に難しいという事実に基づいている。Ｎは、２つの大きな奇素数の積とし、ＡはＮの素因数を知っており、その事実をＢに対して示す場合を考える。

（　例　１　）

１．Ａは、Z^*_N（＋１）にある平方非剰余ｘをＢに送る。

２．Ｂは、ｍ個の乱数$r_1, \ldots, r_m$を選ぶ。また、ランダムなｍビットパターン$b_1, \ldots, b_m$も同時に選ぶ。そして、

$$t_i = \begin{cases} r_i^2 \pmod N, & \text{if } b_i = 0, \\ r_i^2 \cdot x \pmod N, & \text{otherwise;} \end{cases}$$

に基づいて$t_1, \ldots, t_m$を計算し、Ａに送る。

３．Ａは、$t_1, \ldots, t_m$から$b_1, \ldots, b_m$を求めて、それをＢに送る。

４．ＢはＡの答えを検査し、答えが正しければ、Ａは平方剰余判定が出来ると結論する。

上の方法そのものは、２．２．１．の確率的暗号化法と全く同じｉｄｅａを使っている。ｍの値が大きい程、高い確率で信頼できる結果が得られる。

次の例は、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵配送方式やＥｌＧａｍａｌ暗号方式のような離散対数問題に基づく方式において秘密鍵の値を相手に知られずに鍵の所有者であることを示そうというものである。ここでは、答が鍵になっている。ｐを大きな素数、ｇをｍｏｄ　ｐにおける原始根とする。Ａは、秘密鍵ｘを持ち、

$$a = g^{x} \quad (\mathrm{mod}\ p)$$

として、Ｂにａ，ｐ，ｇを知らせておくものとする。

（　例　２　）

１．Ａは、乱数ｒを選び、

$$R = g^{r} \quad (\mathrm{mod}\ p)$$

を計算して、ＲをＢに送る。

２．Ｂは、乱数ビットｅをＡに送る。

３．Ａは、もしｅ＝０ならｒを、そうでなければｔ＝ｒ＋ｘをＢに送る。

４．Ｂは、ｅの値に応じて、

$$g^{r} = R \quad (\mathrm{mod}\ p) \quad \mathrm{if}\ e=0,$$

または、

$$g^{t} = aR \quad (\mathrm{mod}\ p) \quad \mathrm{otherwise},$$

を検査する。

検査式が成立しなければ、Ａでないことになり、成立した時は、上の１．から４．をさらに繰り返す。

この方式では、繰り返し回数を十分大きくとれば殆ど間違いなく正しい検査ができることになる。もし、Ａが乱数Ｒをｍ個まとめてＢに送り、Ｂがｅの値をまとめて１度にＡに与えてしまうと、例１のようなパラレル形になる。（一般に、パラレル形は厳密な意味でＺＫＩＰではないという人もいる。）

どちらの例も、Ｂの判定結果の正しさは”確率的”にしか与えられない。この特徴は、いままで提案されているＺＫＩＰに共通のものである。

２．２．２．ＺＫＩＰ　ＩＤ－ｂａｓｅｄ識別

ここでは、Ｆｉａｔ－Ｓｈａｍｉｒによって提案された、ＺＫＩＰとＩＤ－ｂａｓｅｄを組み合わせた個人識別（自己証明）方式について説明する。

まずセンターは、大きな２つの素数の積Ｎを選ぶ。また〔０，Ｎ）上のランダム関数ｆを選ぶ。ランダム関数とは決定的多項式時間内では真にランダムな関数と区別できないような関数とする。Ｎとｆは公開し、この条件のもとでセンターは次の手続きを行う。

１．小さいｊに対して$v_j = f(ID, j)$を計算する。

２．ｍｏｄ　Ｎで平方剰余になるようなｋ個のv_jを選び、v_j^{-1}の最小の平方根s_jを計算する。

３．ＩＤとｋ個のs_jの値とそれらのインデックスｊの値を記憶したスマートカードを発行する。

このように準備した後に、ＡがＢに対して自身の証明を次のように行う。

１．ＡはＢにＩＤを送る。

２．Ｂは、$v_j = f(ID, j)$，$j = 1, \ldots, k$を生成する。

以下のステップ３から６はｉ＝１，．．．ｔまで繰り返す。

３．Ａは乱数r_jを生成し$x_i = r_i^2 \pmod N$をＢに伝送する。

４．Ｂは２進乱数ベクトル$(e_{i1}, e_{i2}, \ldots, e_{ik})$を送る。

５．Ａは次式を計算しＢにy_iを送る。

$$y_i = r_i \prod_{e_{ij}=1} s_j \pmod N$$

６．Ｂは、

$$x_i = y_i^2 \prod_{e_{ij}=1} v_j \pmod N$$

が成立するかどうかを調べ、成立しなければＡは偽者となり成立すればステップ３にもどってｉ＝ｔまで繰り返す。ｔ回すべて検査にとおればＡは本人となる。

この方法では、第３者がＡに偽装しても成功する確率は、ｋとｔが大きければ非常に小さい。またＡがＢに渡す情報は乱数なので平方剰余判定の難しさを仮定すればＺＫＩＰにもなっている。

２．３．デジタル署名

２．３．１．センターがない場合の署名

デジタル署名の必要性は、先にも述べたように送信者・受信者間に紛争が起こるような状況のときに生じる。そのような可能性がなければ認証のみで十分である。秘密鍵方式によるデジタル署名もないわけではないがどれも実用性にかける。それに対して、公開鍵暗号方式はその非対称性がデジタル署名の機能を実現するのにおおいに役立つ。これは、送信者のみが知っている情報（秘密鍵）を用いることができるためで、そのおかげで第３者による署名の偽造ができないばかりか受信者が署名文を偽造することもできない。但し、公開鍵暗号はすべてデジタル署名の機能をもつわけではない。復号化関数の定義域が暗号化関数の定義域より広い場合、あるいは復号化関数が１対多になるような場合は、全平文空間に対してデジタル署名を行うのは不可能である。

センターを用いない最も単純なデジタル署名は次のようになる。Ａを送信者、Ｂを受信者とし、Ｅ（　）をＡの公開鍵による暗号化関数、Ｄ（　）をＡの秘密鍵による復号化関数とすると、

１．Ａは、通信文Ｍを自分の秘密鍵を用いて、署名文Ｓを次のように作り、

$$S = D(M)$$

Ｂに送る。

２．Ｂは、署名文ＳをＡの公開鍵を用いて、

$$M = E(S)$$

により、通信文Ｍをうる。

Ｍが意味のあるものであれば、Ａから送られたものであると判断する。

この方法にかぎらず、署名や暗号では復元した通信文が意味のある文章であるかどうかに基づいて送信者が正しいかどうかを判定することが多い。署名では、判別距離以上の署名文でないと一意に判定できないことになるが、一方暗号では、判別距離以下の暗号文でないと安全な暗号（Ｓｈａｎｎｏｎの　ｐｅｒｆｅｃｔ　ｓｅｃｒｅｃｙ）にならない。この関係は署名と暗号の双対性といわれている。

上の例では、通信文全体を秘密鍵で暗号化しそれを署名としているが、実際上通信文全体を処理するのは手間がかかるため、先にのべた認証における認証子のみを秘密鍵で暗号化して署名文とするほうが実用的である。ＺＫＩＰでは、自身の秘密を相手に示さないことに重点が置かれていたが、デジタル署名ではその秘密を積極的に用いることになるのでＺＫＩＰはデジタル署名と「逆の関係」にあるという人もいる。

２．３．２．ＩＤ－ｂａｓｅｄ署名

次に、センターのある場合としてＳｈａｍｉｒの提案したＩＤ－ｂａｓｅｄ署名を述べる。

まず、センターはＲＳＡ暗号と１方向性関数ｈも公開する。即ち、

１．２つの素数ｐ，ｑおよびｄを秘密にする。

２．Ｎ＝ｐｑおよびｅおよびｈを公開する。

そして、各ユーザーｊはセンター加入時に

１．センターにＩＤ$_j$を登録し、

２．センターから

$$s_j = ID_j^d \pmod N$$

をもらう。

ＡがＢに平文ｍを送る場合を考える。

１．Ａは乱数ｒを選び、

$$t = r^e \pmod N$$

$$u = s_a \cdot r^{h(t,m)} \pmod N$$

を計算する。

２．Ａは（ｔ，ｕ）を署名文としてＢに

（ｍ，（ｔ，ｕ））

を送る。

３．Ｂは

$$u^e = ID_a \cdot t^{h(t,m)}$$

が成り立つかどうかを検査する。

Ａは秘密の乱数ｒを用いたので、Ｂが署名を偽造することはできない。また、s_aを用いるのでＡのＩＤ情報が含まれることになる。

以上、暗号プロトコルとして、認証・個人識別・テジタル署名のうち現在提案されているもののなかでも主要なものについてまとめてみたが、ＤＥＳを用いたもの以外は実用化されているものはないし、近く使われるという話も聞いたことがない。すべて研究者が案として発表しているにすぎず、これから安全性、実用性についてさらに検討され、改良されていくと思われる。

ＩＩＩ．高速演算アルゴリズム

１．ＲＳＡの高速化

ＲＳＡ暗号のハードウェアによる高速化の方法がいくつか提案され試作されている。現時点では、約５０ｋｂｐｓ程度の速度が達成されており、デジタル署名や鍵配送への応用は実現可能になりつつある。しかし、数１０Ｍｂｐｓといった高速データ通信へ応用できるようになるためには、まだかなりの技術イノベーションが必要だと考えられている。

ＲＳＡ暗号では、２００桁程度の巾乗剰余演算をおこなわなければならない。つまり、Ｎ，ｍ，ｅを２００桁程度の数として、

$$m^e \quad (\mathrm{mod}\ N)$$

を計算する。

巾乗演算の高速化には、次のアルゴリズムが知られている。まず、ｅを２進表現（$e_1, \ldots, e_n$）で表し、

```
c＝1
do  i＝0  to  n－1
    if  e_{n-i} ＝1  then  c＝c・m  (mod N)
    m＝m・m  ( mod N )
end
```

で計算するのである。ｃの最終的な値が求める値である。

こうすると剰余乗算が高々２・ｌｏｇ　Ｎ回で済むことになり、従って基本となる演算は、

$$X \cdot Y \quad (\mathrm{mod}\ N)$$

なる剰余乗算であることがわかる。

この演算の高速化の方式には２通りある。１つは逐次方式で、もう１つが並行方式である。

（１）逐次方式

この方式は、まず積Ｘ・Ｙを求め、そのあとｍｏｄ　Ｎ計算をする方式である。後半の剰余計算を高速化するために、メモリに予め計算しておいた剰余をいれてテーブルを作っておき、剰余計算をテーブル参照でおきかえる方法がよく使われる。積Ｘ・ＹはＮより桁数が倍になるのでＮより大きい部分に対するｍｏｄ　Ｎのテーブルを効率よく作る（あるいは持つ）ことが課題になる。

（２）並行方式

基本的な考え方は、乗数Ｙを部分積で表し、各部分乗算毎に剰余演算も並行して行うというものである。つまり、Ｙを2^k進表現（y_1，．．．．，y_n）で表し、

```
S = 0
do  i = 0  to  n − 1
    S = S + X · y_{n-i}  (mod N)
    X = X · 2^k          (mod N)
end
```

で計算する。Ｓの最終的な値が求める値になる。ここでも剰余計算をテーブル参照でおきかえることによって高速化できる。

どちらの場合も剰余テーブルを用いて高速化を図っているが、Ｎの値が固定でない場合は剰余テーブルそのものを毎回作成しなければならないので使いづらいかも知れない。また、（２）において剰余テーブルを用いる代わりに近似計算に基づく方式も提案されている。

２．ＧＦ（2^n）上の乗算

この演算は、誤り訂正符号や離散対数問題を用いた暗号等で必要となってくる。体の元は基底の取り方で表し方が異なり、またそれに伴って、演算の手間が違ってくることから、演算を行うのに効率のよい基底が幾つか提案されている。一般に、ＧＦ（2^n）の基底とは、ＧＦ（２）上線形独立なｎ個の元の組のことを指しているが、そのなかで代表的なものについてまとめると次のようになる。

（１）多項式基底

最もポピュラーなもので通常、基底といえばこれをさすことが多い。ＧＦ（2^n）の多項式基底は、ＧＦ（２）上線形独立なｎ個の元のなかで次のような形をした組（１，ａ，．．．，a^{n-1}）で定義される。元をこの基底で表現した時の演算は通常の多項式剰余演算になる。

（２）正規基底

この基底は、（ａ，a^2，．．．．，$a^{2^{n-1}}$）の形をしたＧＦ（２）上線形独立な元の組である。正規基底を用いると二乗が巡回シフトで実現できるので、乗算も簡略になる。例えば、

$x=(x_1,\ldots,x_n)$, $y=(y_1,\ldots,y_n)$ で、$z=xy$ を求めるとした場合、$z=(z_1,\ldots,z_n)$ は同一の関数 $f(\cdot)$ を用いて次のようになる。

$$z_{n-1}=f(x_1,\ldots,x_{n-1},x_n,y_1,\ldots,y_{n-1},y_n)$$
$$z_{n-2}=f(x_n,x_1,\ldots,x_{n-1},y_n,y_1,\ldots,y_{n-1})$$
$$\vdots$$
$$z_1=f(x_2,x_3,\ldots,x_1,y_2,y_3,\ldots,y_1)$$

つまり、変数を順に巡回シフトするだけで1つの関数 $f(\cdot)$ のみから z が得られるのである。この御蔭で回路規模がかなり削減できる。例として、$n=4$ のときの関数 $f(\cdot)$ を次に示す。

$$f(x_1,x_2,x_3,x_4,y_1,y_2,y_3,y_4)$$
$$=x_3y_3+x_4y_3+x_3y_4+x_4y_2+x_2y_4+x_4y_1+x_2y_1+x_1y_2\ .$$

（3）双対基底

ある基底 $(a_1,\ldots,a_n)$ に対する双対基底 $(b_1,\ldots,b_n)$ とは、

$$Tr(a_i\,b_j)=\begin{cases}1 & \text{if } i=j;\\ 0 & \text{otherwise;}\end{cases}$$

で定義される。$(a_1,\ldots,a_n)$ と $(b_1,\ldots,b_n)$ が一致した時、それは自己双対基底であると言う。多項式基底には自己双対基底が存在しないこと及び、自己双対正規基底は n が奇数の時存在することが証明されている。

積 $z=xy$ を求める場合、$x=(x_1,\ldots,x_n)$ をある基底で表し、$y=(y_1,\ldots,y_n)$ をその双対基底で表しておくと、双対基底で表した積 $z=(z_1,\ldots,z_n)$ の各要素は、一般に、$z_i=Tr(a_i\cdot z)$ と書ける。この性質を用いて、多項式基底とその双対基底により、ビットシリアル型の効率のよい乗算アルゴリズムが構成できることが知られている。

また、多項式基底の双対基底は、一般に $(c\cdot r_1,\ldots,c\cdot r_n)$ と表せることがわかっている。ここで、$f(x)$ を a の最小多項式とすると

$$c=1/f'(a)$$

で、r_i は

$$f(x)=(x-a)(r_1+r_2x+\cdots+r_{n-1}x^{n-1})$$

により決まる元である。

以上見たようにそれぞれの基底は特徴を持っている。しかし、実際にはシステムそのものは多項式基底で実現されることが多いので正規基底や双対基底を用いる場には、基底の変換を行わなければならずその分の手間も考えにいれなければ正確な効率の評価はできない。

ＩＶ．おわりに

ここで整理したトピックは、暗号で話題になるもののうちの一部にすぎず、これら以外にもとりわけ重要なものとしては次のものがあげられる。

１．ナップザック暗号

特に、ＬＬＬアルゴリズムによる解読に関連した話題はおもしろい。

２．Ｓｉｍｍｏｎｓの認証理論

Ｓｈａｎｎｏｎの暗号理論にも匹敵する（？）といわれている話題である。

３．Ｃｈａｕｍのプロトコル

擬似ＩＤ（ｐｓｅｕｄｏｎｙｍ）を用いたプライバシー保護を目的とする電子取引プロトコルの話。

４．電子的ポーカー・ジャンケンのプロトコルの研究。

勉強不足の為、これらについて整理することができなかったのは大変残念である。

参考文献

—— 洋書 ——

1. Proceeding of Eurocrypt '84, Lecture Notes in Computer Science, vol.209, Springer-Verlag, 1985.

2. Proceeding of Eurocrypt '85, Lecture Notes in Computer Science, vol.219, Springer-Verlag, 1986.

3. Proceeding of Eurocrypt '87, Lecture Notes in Computer Science, vol.304, Springer-Verlag, 1988.

4. Proceeding of Crypto '84, Lecture Notes in Computer Science, vol.196, Springer-Verlag, 1985.

5. Proceeding of Crypto '85, Lecture Notes in Computer Science, vol.218, Springer-Verlag, 1986.

6. Proceeding of Crypto '86, Lecture Notes in Computer Science, vol.263, Springer-Verlag, 1987.

7. Proceeding of Crypto '87, Lecture Notes in Computer Science, vol.293, Springer-Verlag, 1988.

8. SIAM Journal of Computing, vol.17, no.2, (1988). (Special issue on cryptography).

9. Proceeding of IEEE, vol. 76, no.5, (1988). (Special section on cryptology).

10. IEEE Network, vol.1, no.2, (1987). (Special issue on Network Security).

11. R.A. Rueppel, Analysis and Design of Stream Ciphers, Springer-Verlag, 1986.

12. R.Lidl and H.Niederreiter, Introduction to Finite Fields and Their Applications, Cambridge University Press, 1986.

13. B-Z. Chor, Two issues in Public Key Cryptography, The MIT Press, 1985.

14. H.C.A. van Tilborg, An Introduction to Cryptology, Kluwer Academic Publishers, 1988.

15. M.R.Gary and D.S.Johnson, Computers and Intractability, W.H.Freeman and Company, 1979.

16. D.E.R.Denning, Cryptography and Data Security, Addison-Wesley, 1983.

17. H.Beker and F. Piper, Cipher Systems: The Protection of Communiocations, Wiley-Interscience, 1982.

18. W.Diffie and M.E.Hellman, New directions in cryptography, IEEE Trans. Informat. Theory, vol.IT-22 (1976), 644-654.

19. R.L.Rivest, A.Shamir, and L.Adleman, A method for obtaining digital signatures and public key cryptosystems, CACM, vol.21 no.2 (1978), 120-126.

20. L.Blum,M.Blum, and M.Shub, A simple unpredictable pseudorandom number generator, SIAM J. Computing, vol.15 no.2 (1986), 364-383.

21. S.Goldwasser and S.Micali, Probabilistic Encryption, J. Comput. System Sci., 28 (1984), 270-299.

22. S.Goldwasser,S.Micali, and C.Rackoff, The Knowledge complexity of interactive proofsystems, in 17th Symp. on the Theory of Computing, 291-304, 1985.

２３．S.Goldwasser and J.Killian, All primes can be quickly certified, in 18th Symp. on the Theory of Computing, 316-329, 1986.

２４．I.S.Hsu,T.K.Truong,L.J.Deutsch, and I.S.Reed, A Comparison of VLSI Architecture of Finite Field Multipliers Using Dual, Normal, or Standard Bases, IEEE Trans. on Computers, vol.37 no.6 (1988), 735-739.

２５．A.Yao, Theory and applications of trapdoor functions, Proc. 23rd IEEE Symp. on Foundations of Computer Science, 80-91, 1982.

—— 和書 ——

２６．別冊「数理科学」（暗号）、（１９８２—５）、サイエンス社.

２７．数理科学、ｎｏ．８，（１９８６）、（特集：新しい暗号－高速処理の技術）サイエンス社.

２８．Computer Today, vol.4, no.1, (1987).（特集：情報化社会の暗号システム）サイエンス社.

２９．マイヤー・マチス（細貝他訳）：暗号、自然社（１９８６）.

３０．デービス・プライス（上園他訳）：ネットワーク・セキュリティ、日経マグロウヒル　（１９８５）.

３１．池野信一、小山謙二：現代暗号理論、電子通信学会（１９８６）.

３２．和田秀男：コンピュータと素因子分解、遊星社（１９８７）.