マルチメディア通信と分散処理　90－4
コンピュータセキュリティ　2－4
（1998．9．17）

暗号アルゴリズムのコード化（動的暗号アルゴリズム）

五十嵐　育弘　（ E-mail : igarashi@iwangc.eec.toshiba.co.jp ）

平文、鍵、暗号文の間には、暗号アルゴリズムによる関数関係がある。この関数関係により、暗号文には平文や鍵の痕跡が残る。この痕跡を不鮮明にするために、暗号アルゴリズムを頻繁に変える方法を提案する。

Metacipher algorithm

Ikuhiro Igarashi

提案する暗号手法の特徴

- 暗号アルゴリズムが確定していない。
- 与えられる任意のデータの内容に依存して暗号関数が確定し、暗号アルゴリズムが決定する。
- 同一平文、同一鍵で暗号化を施しても、生成される暗号文の内容が、その都度異なるように構成できる。
- 暗号文の中に、復号化のための情報が混在している。

用語の定義

基本暗号アルゴリズム

データに対して、小規模な換字や転置等の基本的な操作を施す方法で、暗号アルゴリズムの部品という観点でとらえうる基本的な暗号アルゴリズム。

基本暗号アルゴリズムコード

基本暗号アルゴリズムをある方法で数値化した時、その数値を「基本暗号アルゴリズムコード」とする。

静的暗号アルゴリズム

基本暗号アルゴリズムを静的に組み合わせたものが、「静的暗号アルゴリズム」である。
静的暗号アルゴリズムは固定的であり、パラメータに依存してアルゴリズム自体が変化する事は無い。

メタ暗号アルゴリズム

暗号アルゴリズムを生成するアルゴリズムとして、「メタ暗号アルゴリズム」の概念を導入する。
メタ暗号アルゴリズムは、「平文をいかに暗号化するか」という事を問題にせず、「暗号アルゴリズムをいかに構成するか」という事を問題にする。基本暗号アルゴリズムをあるパラメータに依存して動的に組み合わせるアルゴリズムが、メタ暗号アルゴリズムである。

動的暗号アルゴリズム

メタ暗号アルゴリズムを用いて生成された暗号アルゴリズムが、「動的暗号アルゴリズム」である。

暗号アルゴリズムのモデル

静的暗号アルゴリズムとメタ暗号アルゴリズム、動的暗号アルゴリズムのそれぞれのモデルの構造を、見通しをよくするために、論理記号を使って表現してみる。

静的暗号アルゴリズム

$\exists f\ \forall M\ \forall K\ \exists E\ (\ E = f\ (\ M,\ K\ )\ )$

静的暗号関数fが確定しており、その関数で、任意の平文Mと鍵Kに対して暗号文Eが決定する。

メタ暗号アルゴリズム

$\exists g\ \forall R\ \exists h\ (\ h = g\ (\ R\ )\ )$

メタ暗号関数gが確定しており、その関数で、任意の乱数Rに対して動的暗号関数hが決定する。

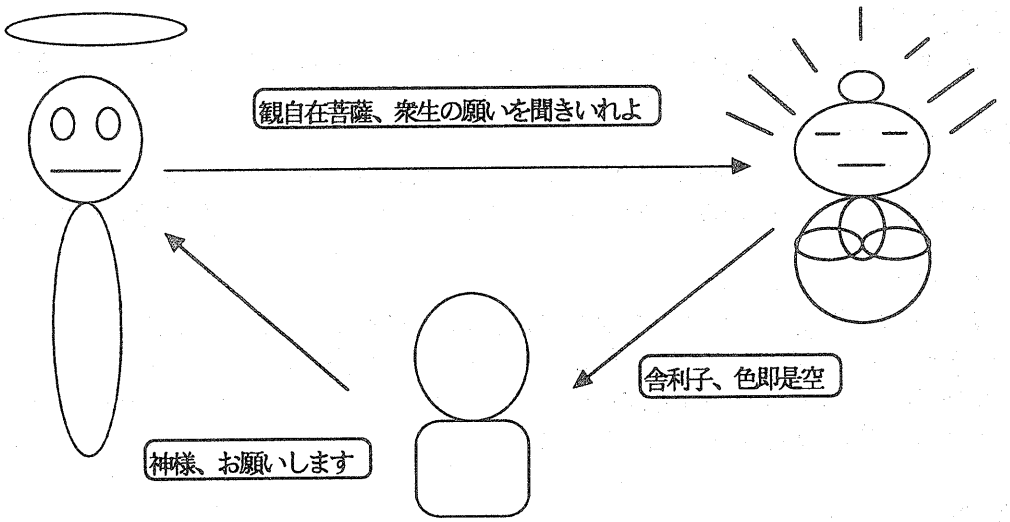
動的暗号アルゴリズム

$\exists g\ \forall R\ \exists h\ \forall M\ \forall K\ \exists E\ (\ h = g\ (\ R\ ),\ \ E = h\ (\ M,\ K,\ R\ )\ )$

メタ暗号関数gが確定しており、任意の乱数Rに対して、ある動的暗号関数hが決定し、動的暗号関数hで、任意の平文Mと鍵K、乱数Rに対して暗号文Eが決定する。

f：静的暗号関数　g：メタ暗号関数　h：動的暗号関数　M：平文　K：鍵　R：乱数　E：暗号文

静的暗号アルゴリズムは、「静的暗号関数fを神様に例えるなら、衆生が神様にお願いをし、神様が直接衆生の願いを聞きいれる」ような構造をしている。これに対し、動的暗号アルゴリズムは、「メタ暗号関数gが神様となり、衆生が神様にお願いをし、神様はその願いに対応する動的暗号関数hという仏様を選択、遣わし、仏様が衆生の願いを聞きいれる」ような構造になっている。



基本暗号アルゴリズム

メタ暗号関数は、任意のデータを動的暗号関数に写像する。動的暗号関数は、基本暗号アルゴリズムの組み合わせからなる。基本暗号アルゴリズムは、「小規模な換字や転置等の操作をデータに施す方法」を述べたものだが、その例を以下に示す。

- 任意の方法で1バイトの乱数を必要個数分選択し、対象文の奇数バイト目に挿入する。
- 対象文の偶数バイト目のバイトデータの3ビット目のビット値を反転する。
- 鍵を7以下の数値に写像し、その数値分、対象文の各々のバイトデータに左ローテイトシフトを施す。
- 対象文の奇数バイト目のバイトデータに否定演算を施す。
- 対象文の各々のバイトデータの2ビット目と7ビット目のビット値を互いに交換する。
- 対象文の偶数バイト目と奇数バイト目のデータを各々排他的論理和をとり、その結果を奇数バイト目に反映させる。
- 対象文の各バイトデータの上位ニブルと下位ニブルを交換する。

動的暗号関数空間は、基本暗号アルゴリズムの数をm、メタ暗号関数に与える任意の乱数のバイト数をnとした場合、

$$ {}_{m}P_{n} $$

の広さを持つ。

任意の乱数から基本暗号アルゴリズムコードへの写像

基本暗号アルゴリズムを256種類分集め、1バイトデータにコード化をすると、与えられた乱数の対象バイトデータをそのまま基本暗号アルゴリズムコードとみなす事ができる。乱数に対する写像であるため、この写像で必要十分であり、これ以上の凝ったものは必要がない。

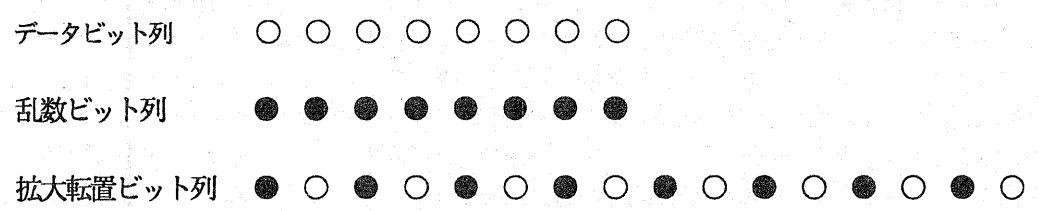
しかし、単一写像では、以下のような都合の悪い状況が発生する事がある。

- 基本暗号アルゴリズムの中に、互いに逆関数の関係にあるものがリストアップされている場合、これらを連続して作用させると、元のデータに戻ってしまう。
- 基本暗号アルゴリズムの中に、固定的なビットの反転や同一値での排他的論理和演算等がリストアップされている場合、このようなものを連続して作用させると、元のデータに戻ってしまう。

以上のような写像にならないように、例外処理を設けておく必要がある。

例外処理は、静的暗号関数で十分であり、拡大転置アルゴリズムを作用させるのが良いと思う。

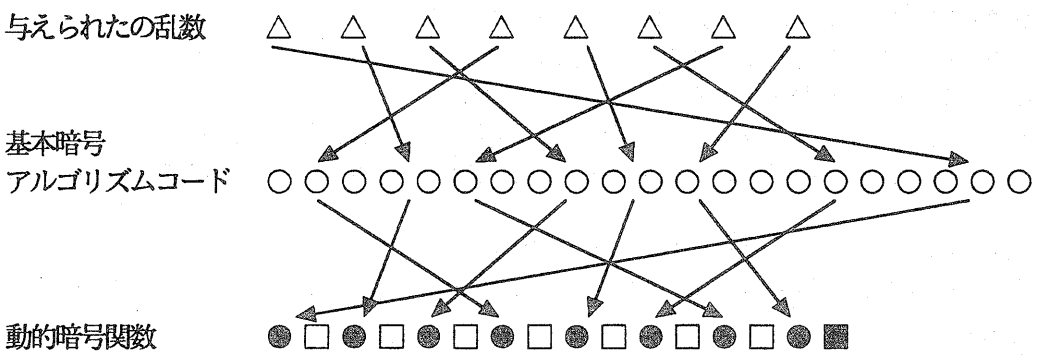
拡大転置アルゴリズムとしては、任意の方法で乱数を選択し、その乱数のビット列を、対象となるデータ（平文や暗号化途中のデータ）の偶数ビット目と奇数ビット目の間に挿入するような方法で十分と考える。



メタ暗号関数と動的暗号関数の概念

メタ暗号関数の作用を以下に示す。

1．与えられた任意の乱数の各々のバイトデータを、基本暗号アルゴリズムコードに写像する。
2．写像された基本暗号アルゴリズムコードの組み合わせから、動的暗号関数を決定する。



●部は、与えられた乱数に対応した基本暗号アルゴリズム。
□部は、拡大転置アルゴリズムである静的暗号関数（必要な場合にのみ作用させる）。
■部は、復号化のための情報を暗号文の中に混入させる静的暗号関数であり、以下の作用をする。
- 与えられた乱数（上図△部）を、鍵と、ある確定した静的暗号関数を用いて暗号化する。
- ●部と□部の作用により生成された暗号文に、暗号化を施した乱数を埋め込む。
- 鍵をパラメータとしたシャッフル関数（下記にて定義）を用いて、暗号文の順序を入れ替える。

暗号文に復号情報を埋め込む（木を隠すなら森の中）

動的暗号アルゴリズムにおける鍵の役割は、復号情報（与えられた乱数を暗号化したもの）を暗号文の中に埋め込む事が主である。暗号文に復号情報を埋め込む方法の例を示す。今、暗号文のバイト数をm、復号情報のバイト数をnとした場合、m／（n＋1）を基数として、復号情報の1バイト目を、その基数の1倍、2バイト目は、その2倍、nバイト目は、そのn倍を、暗号文のバイト位置に対応させ、各々の復号情報のバイトデータを暗号文に埋め込む。

シャッフル関数（森の木の植えなおし）

トランプをシャッフルするように、個々の性質は変えず、その順序のみを変える関数をシャッフル関数と定義する。復号情報が埋め込まれた暗号文に、鍵をパラメータとするシャッフル関数を作用させて、最終暗号文を生成する。シャッフル関数の例を示す。鍵の内容に依存して発生する乱数系列を利用する。最終暗号文の1バイト目のバイトデータは、シャッフル前の暗号文のバイト数をmとした場合、鍵の内容に依存して生成される乱数のmでの剰余を求め、その値のバイト位置にあるシャッフル前の暗号文のバイトデータを、最終暗号文の1バイト目のデータとする。そして、シャッフル前の暗号文の中からそのデータを除外する。最終暗号文の2バイト目のデータは、同様に乱数を発生させ、（m－1）での剰余を求め、その値のバイト位置にある、更新されたシャッフル前の暗号文のバイトデータを、2バイト目のデータとする。以下、同様。

動的暗号関数の構成と実施例

与えられた任意の乱数を、そのまま基本暗号アルゴリズムコードとみなすような写像で構成された動的暗号関数hの例を示す。以下、基本暗号アルゴリズムコードnの基本動的暗号関数をhnと表記する。
動的暗号関数を生成するために、メタ暗号関数に与える乱数を“CSEC”とした場合、そのASCIIコードは、“43 53 45 43”である。これがそのまま基本暗号アルゴリズムコードとなる。それぞれの基本動的暗号関数は、h 43, h 53, h 45, h 43であり、その合成関数である動的暗号関数の一部は、

h 43(h 45(h 53(h 43(M, K), K), K), K)　　平文:M,　鍵:K

となる。これに、基本動的暗号関数の連鎖間に必要に応じて拡大転置アルゴリズムを施し、最後に、生成された暗号文に、メタ暗号関数に与える乱数を暗号化して混入させ、シャッフルさせる関数をもって、動的暗号関数の完全版が確定する。基本暗号アルゴリズムコード“43”と“53”が互いに逆関数の関係があるとし、それ以外に上記で用いられているコード間の連鎖には、不都合が生じないと仮定する。拡大転置アルゴリズムを示す関数をs0、メタ暗号関数に与える乱数を暗号化して混入させ、シャッフルさせるアルゴリズムを示す関数をs1と表記した場合の動的暗号関数の完全版は、

s 1(h 43(h 45(h 53(s 0(h 43(M, K), K), K), K), K), K, R)　乱数:R

となる。この動的暗号関数と、平文を“Computer△Security△Group”（ASCIIコード“43 6F 6D 70 75 74 65 72 20 53 65 63 75 72 69 74 79 20 47 72 6F 75 70”）、鍵を“KAGI”（ASCIIコード“4B 41 47 49”）とし、基本暗号アルゴリズムを随時、定義しながら暗号化の流れを追ってみる。

h 43（）：対象文の各々のバイトデータに1ビット左ローテイトシフトを施す。
E 43 ＝ h 43（“Computer△Security△Group“, ”KAGI”）の暗号文のASCIIコードによる16進ダンプデータは、

86 DE DA E0 EA E8 CA E4 40 A6 CA C6 EA E4 D2 E8 F2 40 8E E4 DE EA E0

s 0 （）：拡大転置アルゴリズムを表す静的暗号関数。前記の拡大転置アルゴリズムを用いる。
便宜上、この関数に与えられた乱数値を オール0 として計算してみる。

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

E s0 ＝ s 0（ E 43 , ”KAGI” ）の暗号文の16進ダンプデータは、

40 14 51 54 51 44 54 00 54 44 54 40 50 44 54 10 10 00 44 14 50 44 50
14 54 44 54 10 51 04 54 40 55 04 10 00 40 54 54 10 51 54 54 44 54 00

h 53（）：対象文の各々のバイトデータに1ビット右ローテイトシフトを施す。
E 53 ＝ h 53（ E s0 , ”KAGI” ）の暗号文の16進ダンプデータは、

20 0A A8 2A A8 22 2A 00 2A 22 2A 20 28 22 2A 08 08 00 22 0A 28 22 28
0A 2A 22 2A 08 A8 02 2A 20 AA 02 08 00 20 2A 2A 08 A8 2A 2A 22 2A 00

h 45 ( ) : 鍵の各々のバイトデータの上位ニブルと下位ニブルを交換する。
E 45 = h 45 (E 53 , "KAGI") の暗号文の１６進ダンプデータは、
02 A0 8A A2 8A 22 A2 00 A2 22 A2 02 82 22 A2 80 80 00 22 A0 82 22 82
A0 A2 22 A2 80 8A 20 A2 02 AA 20 80 00 02 A2 A2 80 8A A2 A2 22 A2 00

h 43 ( ) : 対象文の各々のバイトデータに１ビット左ローテイトシフトを施す。
E 43 = h 43 (E 45 , "KAGI") の暗号文の１６進ダンプデータは、
04 41 15 45 15 44 45 00 45 44 45 04 05 44 45 01 01 00 44 41 05 44 05
41 45 44 45 01 15 40 45 04 55 40 01 00 04 45 45 01 15 45 45 44 45 00

s 1 ( ) : メタ暗号関数に与える乱数を暗号化し、暗号文に混入させ、シャッフルさせる静的暗号関数。便宜上、与られた乱数を鍵を用いて暗号化する方法は、「与えられた乱数と鍵とで排他的論理和演算を行い、その結果を与えられた乱数の暗号文にする」とする。復号情報を暗号文に混入させる方法は、前記「暗号文に復号情報を埋め込む」方法を用いる。（紙面の都合と簡易な例を示す目的のため、シャッフル関数は割愛した）。

メタ暗号関数に与える乱数を鍵で暗号化する

与えられた乱数 "CSEC" ("43 53 45 43") と鍵 "KAGI" ("4B 41 47 49") の各々のバイトデータの排他的論理和をとった値 "08 12 02 0A" が、与えられた乱数に暗号化を施した結果である。

暗号文に復号情報を埋め込む位置の決定

暗号文は４６バイト、鍵 "KAGI" のバイト数は4バイト。４６／（４＋１）＝ ９ が基数となる。復号情報 ("08 12 02 0A") の各々のバイトデータに対応する暗号文への挿入位置は、それぞれ暗号文の ９、１８、２７、３６バイト目になる。

E＝ s 1 (E 43 , "KAGI", "CSEC") の暗号文の１６進ダンプデータは、

04 41 15 45 15 44 45 00 **08** 45 44 45 04 05 44 45 01 01 **12** 00
44 41 05 44 05 41 45 44 **02** 45 01 15 40 45 04 55 40 01 **0A** 00
04 45 45 01 15 45 45 44 45 00

これがシャッフル前の暗号文である。鍵のバイト数を減らし、単純な静的暗号関数を例に取ってみた。
暗号文のバイトデータ値に偏りがあるのは、拡大転置アルゴリズムで使用した乱数値が主な要因となっている。英語や日本語をコード化したものはある値に偏るが、この偏りを解消すべく、平分に対して最初に乱数を選択し、拡大転置を施してから動的暗号アルゴリズムを施すのがよいと考える。
動的暗号アルゴリズムを用いて生成した暗号文に対し、さらに、別の乱数を用いて動的暗号アルゴリズムを作用させ、多重度をあげていくと、いくらでも複雑な暗号文が生成できる。復号に際しては、多重度を考慮し、鍵をもとに暗号文から「暗号化されたメタ暗号関数に与える乱数」を抽出、復号し、暗号化と同様にメタ暗号関数を用いて、基本暗号アルゴリズムの逆関数を構成し、対象文に施せば良い。

参考文献

土居範久・小山謙二 編、コンピュータ・セキュリティ、共立出版株式会社、１９８６