マルチメディア通信と分散処理 37－8
（１９８８．５．２０）

高速秘密鍵暗号方式のＬＡＮ適用に関する考察

清水明宏
ＮＴＴ通信網総合研究所

本論文では、秘密鍵暗号方式の高速マルチメディアＬＡＮ環境への適用について論じる。まず、メディア種別、レイヤを考慮した適用法について、ＯＳＩの動向を踏まえて考察し、レイヤ２の機密保護サービスとして、高速暗号ＬＳＩによりＭＡＣ副層でマルチメディア情報を共通に暗号化する方式を示す。さらに、秘密鍵暗号方式をモデル化し、高精細動画にも適用できる高速秘密鍵暗号方式実現の要素技術となるｆ関数の高速化について論じる。ｆ関数は秘密鍵暗号方式においてデータ拡散の中核処理となるものである。新しいｆ関数の性能は、ハードウェア化した際の速度性能ならびにデータ乱数化能力のふたつの観点から評価した。

F-functions Suitable for Hardware implementation

Akihiro Shimizu

NTT Telecommunication Networks Laboratories,
1-2356, Take, Yokosuka-shi, Kanagawa-ken, 238-03 Japan.

This paper first considers the secret-key cryptomethod application in high-speed and multi-media LAN. In the consideration, common application of secret-key cryptomethod in MAC sublayer is proposed, which achieves the multi-media data confidentiality on the second layer in OSI reference model.

Secondly, a model of secret-key cryptography is proposed. Then, three types of f-function are designed to achieve hi-speed data confidentiality service for multimedia including high division image data. The hardware speed performance and data randomization ability are evaluated for each f-function.

1．はじめに

情報ネットワーク構築手段であるＬＡＮは、データメディアのみを扱うものを中心に普及期にあり、次世代ＬＡＮ開発の焦点は、高速・大容量化、マルチメディア化へと移行しつつある。

ＬＡＮセキュリティ固有の問題として、次のものがあげられる。

①職位や利害関係の異なるグループの同一ＬＡＮ内での共存。

②ＬＡＮ内での権限と実際のオフィスの職位との差。

②はすなわち、スーパユーザの存在である。ＬＡＮシステム内でオールマイティな権限を持つ彼らは、システム内のあらゆる情報を手にいれることができる。担当技術者である彼らが、経営、人事などに関わる機密情報にアクセスすることができる。オフィスにＬＡＮシステムを導入する場合、この問題の解決が重要である。

さらに、今後、ＬＡＮと公衆網との接続がますます盛んになってくる状況では、かなりの部分一般のネットワークセキュリティ技術を適用することができる。公衆網との違いとして、

③高速マルチメディア環境

について考えることが重要である。

最大のポイントは伝送速度である。いままでのネットワークセキュリティでは論じられなかった数百Mbpsに達する暗号化速度を実現できる暗号方式が必要となってくる。このような高速化に対応するには、公開鍵暗号方式より秘密鍵暗号方式の方が現実的である。

本論文では、高速秘密鍵暗号方式に関して議論する。まず、ＯＳＩでのセキュリティアーキテクチャを参考に、メディアとレイヤを考慮した秘密鍵暗号方式の適用形態について述べる。続いて、秘密鍵暗号アルゴリズムをモデル化する。さらに、秘密鍵暗号方式を高速化するためのひとつの手段として、データ乱数化の中核処理として用いられるｆ関数の構造の高速化について論じる。ｆ関数の性能は、ハードウェア化した際の速度性能ならびにデータ乱数化能力の二つの観点から評価した。

2．ＯＳＩのセキュリティアーキテクチャ

ＯＳＩにおけるセキュリティアーキテクチャ[1]から、特に、秘密鍵暗号方式を用いて実現できるセキュリティサービスとレイヤの関係を示す。

表１に網掛けで示したセキュリティサービスは秘密鍵暗号方式を用いて実現できる。セキュリティサービスのうち、送受信者の認証を行ったり送受信の事実の否認を防止したりするサービス、および、アクセス制御サービスを除くものを、秘密鍵暗号方式により実現できることがわかる。信頼性の問題もあるが、低レイヤにおいて高速性が必要な場合、相互エンティティ認証（Peer Entity Authentication）、機密保護（Confidentiality）、完全性（Integrity）の各サービスについては、公開鍵暗号方式よりも秘密鍵暗号方式の方が適していると考えている。

上記セキュリティサービスを含めて、ＬＡＮのセキュリティを保証するための要素技術として、

①パスワード管理方式

②アクセス制御方式

③秘密鍵暗号方式

④鍵管理・配送方式

⑤公開鍵暗号方式

の五つを考えている。これらの技術により、ＯＳＩでいうセキュリティメカニズムが構成され、さらに、そのメカニズムによりセキュリティサービスが構築される。

上記の観点から、以下では③の秘密鍵暗号方式について、高速マルチメディアＬＡＮ環境への適用方式ならびに高速化のための設計法について議論を進める。

表１　ＯＳＩセキュリティサービスとレイヤの関係

セキュリティサービス	レイヤ 1 2 3 4 5 6 7
相互エンティティ認証	・・ＹＹ・・Ｙ
データ発信元認証	・・ＹＹ・・Ｙ
アクセス制御	・・ＹＹ・・Ｙ
コネクションデータ機密保護	ＹＹＹＹ・ＹＹ
コネクションレスデータ機密保護	・ＹＹＹ・ＹＹ
選択フィールドデータ機密保護	・・・・・ＹＹ
トラフィックフローデータ機密保護	Ｙ・Ｙ・・・Ｙ
コネクションデータ完全性保証 リカバリ有	・・・Ｙ・・Ｙ
コネクションデータ完全性保証 リカバリ無	・・ＹＹ・・Ｙ
選択フィールド コネクションデータ完全性保証	・・・・・・Ｙ
コネクションレスデータ完全性保証	・・ＹＹ・・Ｙ
選択フィールド コネクションレスデータ完全性保証	・・・・・・Ｙ
発信否認不可保証	・・・・・・Ｙ
受信否認不可保証	・・・・・・Ｙ

Ｙ：対象レイヤにおいてオプションとして提供される。
・：提供されない。
レイヤ7については、アプリケーションによって提供される場合もある。

3．高速マルチメディアＬＡＮ環境への適用

3．1　高速マルチメディアＬＡＮ環境の特徴

高速マルチメディアＬＡＮ環境では、データ、音声、静止画、動画などのメディアが複合した状態で種々の通信形態が共存する。各メディアの伝送速度の概値を表２に示す。各メディアへの要求条件は、情報の精度と遅延時間の二つの観点から規定できる。各メディアについて、これを評価して表３に示す。

表２　メディアと伝送速度

メディア	伝送速度
データ	～ 10Mbps
音声	～ 1Mbps
静止画	～ 数10Mbps
動画	数10Mbps～数100Mbps

表３　メディアへの要求条件

メディア	情報の精度	遅延時間小
データ	○	×
音声	×	△
静止画	×	×
動画	×	○

○，△，×はこの順に要求の強さを示す。

表３に示す通り、音声、動画については、情報の正確さよりも遅延時間の縮小が要求される。また、データに対しては、遅延の問題よりも情報の精度が要求される。したがって、音声、動画については、プロトコルを簡略化するなどして高速性を確保する（データ圧縮技術によるところもある）ことが必要である。ただし、音声については、表２に示す通り伝送速度がそれほど大きくないため、高速マルチメディアＬＡＮ環境での遅延時間の影響は、動画に比べて小さいと考えられる。

ＬＡＮトポロジーとしては、高速性や効率を考慮してループ型が採用される場合が多い。また、通信方式としては、回線交換方式およびパケット交換方式が考えられる。回線交換方式は、物理回線を占有して使用するため実時間特性には優れているが、送出停止期間のあるような情報を送る場合には効率が低下する。一方、パケット交換方式では、マルチメディアを統一的に扱うことができる反面、遅延や消失のために実時間通信を保証できない場合がある。このように、両通信方式とも長短所がある。そこで、両者の長所を活かした方式として、動画には回線交換方式、その他のメディアにはパケット交換方式を用いる混合方式が提案されている[2]。さらに、マルチメディアに対して、スロット多重による高速パケット交換方式を用いるＯｒｗｅｌｌ　Ｒｉｎｇ[3]などが研究されている。

３．２　マルチメディア統合型秘密鍵暗号適用方式

図１に、ＬＡＮ環境でマルチメディアを扱うノードのモデルを示す。図１において、ＰＨＹはＯＳＩ参照モデルでいう物理層、ＭＡＣ（Media Access Control）は、媒体アクセス制御副層を示している。

マルチメディア統合型のセキュリティサービスを実現するためには、低レイヤにおいて動画も含めた情報の暗号化を行う必要がある。この基本構成要素として、高速な暗号ＬＳＩが必要になる。動画の品質や符号化方法によっても異なるが、数十Mbpsから数百Mbpsの暗号化処理速度性能を有するＬＳＩが必要となってくる。

ＬＡＮの方式によっても異なるが、ＬＬＣ（Logical Link Control）副層以上のレイヤはメディアによって異なる場合が多い。そこで、各メディア共通に暗号機能を適用する方法として、ＭＡＣ副層での適用を考えている。

表１に示したＯＳＩのセキュリティサービスと対応させてみると、ＭＡＣ副層での秘密鍵暗号アルゴリズム適用は、レイヤ２における機密保護サービスにあたる。

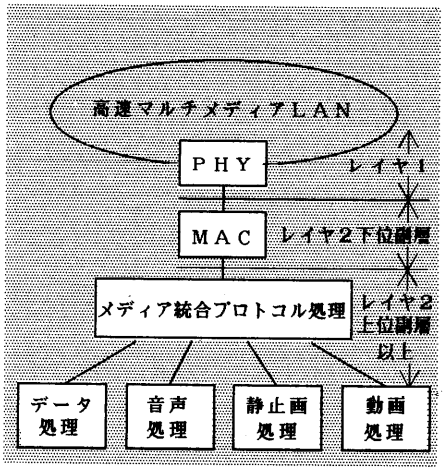

図１　マルチメディアノードのモデル

４．秘密鍵暗号方式の高速化について

高速な暗号ＬＳＩを開発するためには、
①高度なＬＳＩ技術を用いる。
②暗号アルゴリズムを高速化する。
の二つが考えられる。①については、動作クロックの高速化、素子の高速化などが考えられる。一方、②については、
③一度に暗号化するデータ長を増加する。

④処理段数を減らす。
⑤データ乱数化の中核処理であるｆ関数を高速化する。
などが考えられる。

③のデータ長については、６４ビットを１ブロックとして、ブロック単位に処理する方式ＤＥＳ[4]が知られている。このタイプの暗号方式において、ブロック長をあまり大きくすると、効率のいいデータ融合処理の実現、あるいは、データ長の変化への小刻みな対応などに問題が生じてくる。６４ビットの倍の１２８ビットが、ブロック長として限界ではないかと考えている。

④の処理段数について述べる。ＤＥＳは１６段の処理を採用している。ＤＥＳタイプの構造を踏襲するならば、４段処理で簡単には破れない暗号を作ることができると考えられるが、安全性に余裕をもった設計とするために８段処理としたい。

以上述べた二つのポイントに比べて、⑤にあげたｆ関数の高速化は重要である。後に示すが、ｆ関数は各処理段数毎に用いられるため、これを２倍高速にすれば、それがそのままアルゴリズムの全体性能を２倍に向上させることになる。

以下では、８段処理の６４ビットブロック暗号アルゴリズムをモデル化して示す。さらに、ソフトウェアで高速に処理できるｆ関数ＴＹＰＥ－１、ハードウェアに適したｆ関数ＴＹＰＥ－２、ＴＹＰＥ－３の構造および性能について順に示す。

５．記法

本論文において用いる記法について、以下にまとめて示す。

(1) Aなど、英大文字は、複数バイトからなるブロックデータを表す。
(2) A_iなど、右下の添字ｉは、処理手順に対応したブロックデータを表す。
(3) A^iなど、右上の添字ｉは、そのブロックデータ内のｉ番目（ｉ＝0,1,･･･）の１バイトを表す。
(4)（Ａ，Ｂ，・・・）は、この順序でのデータの連結を表す。
(5)⊕はビット対応の排他的論理和を表す。
(6)等号は右辺から左辺への代入を表す。

６．暗号強度評価指標

ｆ関数の設計にあたり、データランダム化性能の測定に用いた暗号強度評価指標[5]についてその概要を説明する。

この指標は、平文、鍵それぞれの入力変化に対する暗号文の変化分布が二項分布Ｂ（ｎ,1/2）（ｎ：暗号文のビット長）に近似する度合を示しており、Ｍ，Ｍσの２種類存在する。

Ｍは、平文あるいは鍵を１～ｎビット変化させたときの暗号文の変化分布の、Ｂ（ｎ,1/2）への近似率の平均を示しており、Ｍσはその標準偏差を示している。Ｍが１００％、Ｍσが０に近いほど、暗号文中に、入力平文あるいは秘密鍵を逆算する手がかりをのこしていないアルゴリズムであるということができる。ＭおよびＭσは、平文変化と鍵変化に対して区別するため、平文変化に対してM_P，$M_P\sigma$、また鍵変化に対してM_K，$M_K\sigma$と定義している。

指標算出のためには、大量の平文および鍵データを用いることが必要であるが、一般的に、実験的に取り扱うことのできるデータの数は、母集団全体に比較して希少である。そこで、統計的計算により、データ数に応じた指標の理論値を算出する。例えば、平文１６個、鍵１６個、平文あるいは鍵の変化データ１６個を組み合わせた計４０９６個のデータに対する理論値は、Ｍ＝９６．５％，Ｍσ＝２．６％と求められる。指標の測定値がこの理論値に近いほど、そのアルゴリズムはデータランダム化の性能が優れているといえる。

７．秘密鍵暗号アルゴリズムのモデル化

図２に秘密鍵暗号アルゴリズムをモデル化して示す。このモデルは、６４ビットの秘密鍵から拡大鍵を生成する鍵スケジュール部と、その拡大鍵を用いて６４ビットの平文から６４ビットの暗号文を生成するデータ処理部からなる秘密鍵ブロック暗号アルゴリズムである。ＤＥＳ[4]、ＦＥＡＬ[7]がほぼこのタイプの暗号としてあげられる。

一般に、鍵スケジュール部は通信のはじめに一度実行しておけば良いので高速性は要求されない。データ処理部の高速化が重要である。

図２に示すように、このモデルは、拡大鍵K_iを用いて初期処理ＩＰを６４ビットの平文データＰに施したのち、図３(a)、(b)に示すインボリューション[6]の組み合わせ処理を８段行い、最後に拡大鍵K_fを用いて最終処理ＦＰを施し、６４ビットの暗号文Ｃを生成する。８段の各処理においては、拡大鍵をパラメータとして用いている。ｆは３２ビットのデータ拡散処理関数を示している。

インボリューションとは、簡単にいえば二度続けて適用すると何も適用しないことに等しいという性質を有する関数のことである。したがって、図２に示したアルゴリズムが、鍵スケジュールを除く同一プログラムで暗号化／復号の両処理を実現するためには、ＩＰとＦＰを同一のインボリューション処理あるいは上下対称のインボリューション組み合わせ処理とする必要がある。

暗号化の手順について説明する。復号の手順は拡大鍵の使用順序を除いて暗号化手順と同じである。

まず、入力された６４ビットの平文Ｐに初期処理ＩＰが施される。

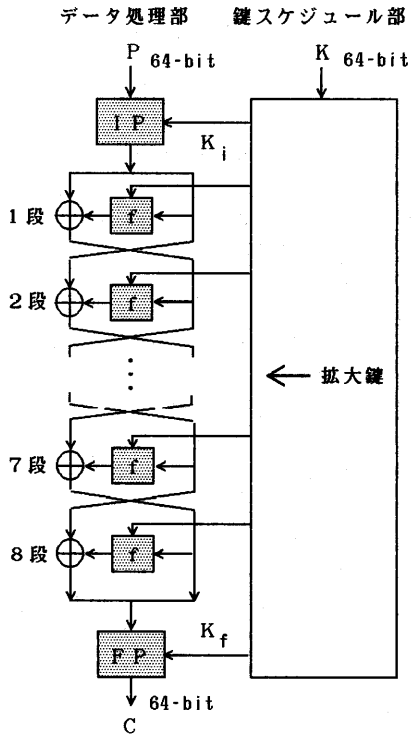

図２　秘密鍵暗号アルゴリズムのモデル

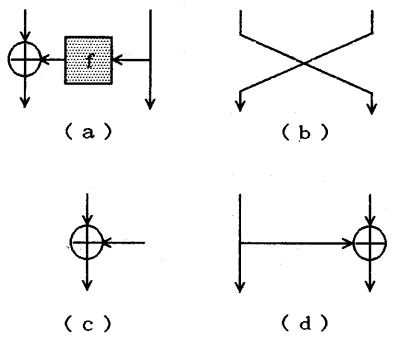

図３　インボリューション

次に、ｒ＝１～８について、L_rとR_rを以下の手順により逐次計算する。

$$R_r = L_{r-1} \oplus f(R_{r-1}, K_{r-1})$$
$$L_r = R_{r-1}$$

最後に、最終処理ＦＰを施して暗号文Ｃ（R_8, L_8）を得る。

ＩＰの例としては、Ｐを３２ビットずつの左右のデータL_0, R_0に分け、以下の処理を施すものが考えられる。ここで、K_iは６４ビットの拡大鍵である。

$$(L_0, R_0) = (L_0, R_0) \oplus K_i$$
$$R_0 = R_0 \oplus L_0$$

ＦＰの例としては、先に示したＩＰと対称な例として、R_8, L_8に対して次の処理を施す。K_fは６４ビットの拡大鍵である。

$$L_8 = L_8 \oplus R_8$$
$$(R_8, L_8) = (R_8, L_8) \oplus K_f$$

ＩＰ、ＦＰは図３(c)、(d)のインボリューションの組み合わせとして構成している。

８．高速ｆ関数の設計

８．１　ＴＹＰＥ－１

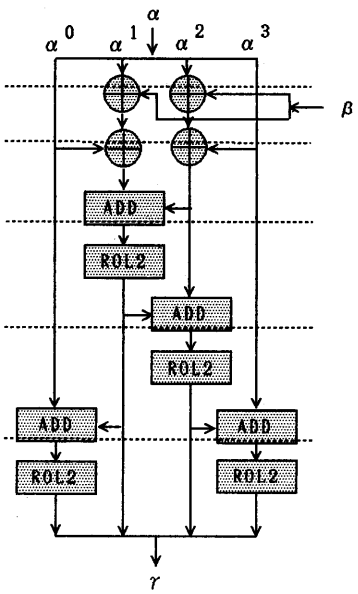

図４　ＴＹＰＥ－１の構造

図４に示すＴＹＰＥ－１のｆ関数は、ソフトウェアで高速に処理できる構造である。あみだ状の構造により、

入力のビット変化を効率よく出力のビットに影響させることができる。

図４において、αは３２ビットの入力データ、βは１６ビットの拡大鍵、γは３２ビットの出力データ、ＡＤＤは加算（最上位ビットからの桁上がりは無視）、ROL2は左方向への２ビット回転処理である。

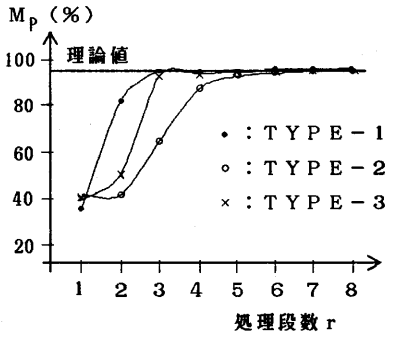

図５　処理段数とM_Pの関係

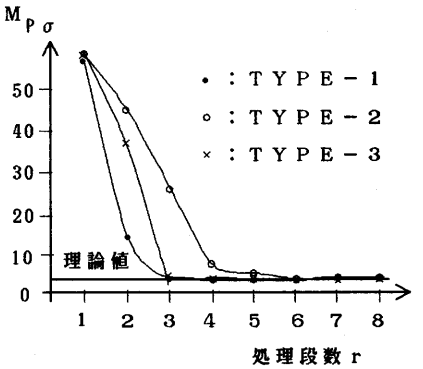

図６　処理段数と$M_{P\sigma}$関係

ＴＹＰＥ－１のｆ関数について動作を説明する。まず、入力された３２ビットのデータαはα^0～α^3の各８ビットの４データに分けられる。ｆ関数の出力γは、以下の手順で得られる。

$\gamma^1 = \alpha^1 \oplus \beta^0 \oplus \alpha^0$
$\gamma^2 = \alpha^2 \oplus \beta^1 \oplus \alpha^3$
$\gamma^1 = ROL2(\gamma^1 + \gamma^2)$
$\gamma^2 = ROL2(\gamma^2 + \gamma^1)$
$\gamma^0 = ROL2(\alpha^0 + \gamma^1)$
$\gamma^3 = ROL2(\alpha^3 + \gamma^2)$

ＴＹＰＥ－１のｆ関数は、二つの１バイトデータを加算して１バイトデータを作成し、さらにそれを左方向に２ビット回転させる処理と、それらを互いに融合させるあみだ構造により、優れたデータ乱数化性能と速度性能を実現できる[7]。

ＴＹＰＥ－１のｆ関数を前章に示した秘密鍵暗号アルゴリズムに適用した場合について、先に述べた暗号強度評価指標のうち、平文変化指標の測定結果を図５，図６に示す。図５には、処理段数ｒに対するM_Pの値および指標の理論値を示す。また、図６には、処理段数ｒに対する$M_P\sigma$の値および指標の理論値を示す。

図５、図６に示した測定結果からわかるように、データ処理部の３段目の処理を終えた段階で、データはほぼ完全に乱数化されている。

次に、このｆ関数のソフトウェアでの性能について考えてみる。図４から、１バイトデータの排他的論理和４回、１バイトデータの加算４回、１バイトデータの１ビット回転処理８回で実現できることがわかる。

さらに、ＬＳＩなどのハードウェアで実現した場合を考える。このｆ関数は、データの拡散効率を向上させるため、４分割したデータのうちの一つデータの拡散処理の処理結果を新しく他のデータの拡散処理に用いている。このため、並列処理が可能な箇所が限られ、これがハードウェアでの処理速度の向上を妨げている。図４に点線で示すように、このｆ関数をハードウェアで構成した場合、２段の排他的論理和処理と、３段の加算とビット回転処理が必要となる。

８．２　ＴＹＰＥ－２

図７に示すＴＹＰＥ－２のｆ関数は、ＴＹＰＥ－１のｆ関数に比べて、データの乱数化処理を並列化できるように構造を工夫したものである。

図７において、αは３２ビットの入力データ、γは３２ビットの出力データ、α^0～α^3はαを４分割した各８ビットのデータ、βは１６ビットの拡大鍵、ＡＤＤは加算（最上位ビットからの桁上がりは無視）、ROLは左方向への１ビット回転処理、RORは右方向への１ビット回転処理を示している。

ＴＹＰＥ－２のｆ関数の動作について説明する。まず、入力された３２ビットのデータはα^0～α^3の各８ビットの４データに分けられる。出力γは、以下の手順で得られる。

$\gamma^0 = \alpha^0 \oplus \beta^0$
$\gamma^3 = \alpha^3 \oplus \beta^1$
$\alpha^1 = \alpha^1 \oplus \gamma^0$
$\alpha^2 = \alpha^2 \oplus \gamma^3$
$\gamma^0 = ROL(\gamma^0) + ROR(\alpha^2)$
$\gamma^1 = ROL(\gamma^1) + ROR(\alpha^2)$
$\gamma^2 = ROL(\gamma^2) + ROR(\alpha^1)$

$\gamma^3 = ROL(\gamma^3) + ROR(\alpha^1)$

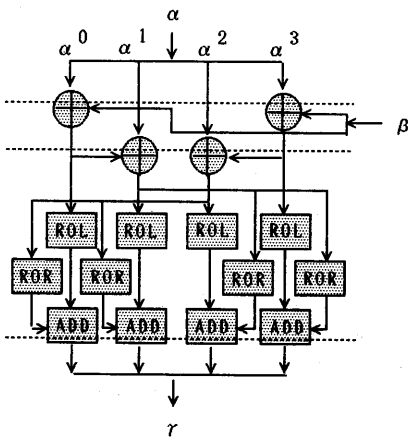

図７　ＴＹＰＥ－２の構造

ＴＹＰＥ－２のｆ関数について性能を示す。まず、ソフトウェアで実現した場合、１バイトデータの排他的論理和４回、１バイトデータの加算４回、１バイトデータの１ビット回転８回の処理で実現できる。これは、ＴＹＰＥ－１のｆ関数と同じである。従って、ＴＹＰＥ－２のｆ関数のソフトウェアでの実行速度は、ＴＹＰＥ－１のｆ関数とほぼ同一であると考えられる。

このｆ関数をハードウェアで実現した場合、図７中に点線で示した処理を並列化することができる。従って、２段の排他的論理和処理と、１段の加算／ビット回転処理の構成となる。加算処理は排他的論理和のほぼ３倍の処理時間がかかるものと考えられる。したがって、２段の排他的論理和処理と３段の加算／ビット回転処理で構成されているＴＹＰＥ－１のｆ関数と比較して、２倍を超える速度性能を実現できる。

ＴＹＰＥ－２のｆ関数を秘密鍵暗号アルゴリズムに適用した場合についても、先に述べた暗号強度評価指標のうち、平文変化指標の測定結果について図５，図６に示している。

測定結果から、データ処理部の６段目の処理を終えた段階で、データが完全に乱数化される。これは、３段処理で飽和したＴＹＰＥ－１のｆ関数に比較して劣るが、８段処理のアルゴリズムを用いる場合には、暗号強度の観点から同一効果である。

８．３　ＴＹＰＥ－３

図８に示すＴＹＰＥ－３のｆ関数は、ＴＹＰＥ－２のｆ関数と同様に、データの乱数化処理を並列化できるように構造を工夫したものである。さらに、データに応じてビットの回転数をかえるようにしており、安全性、データ乱数化効率ともに向上している。

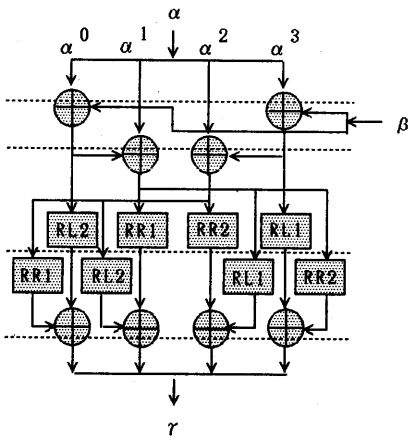

図８　ＴＹＰＥ－３の構造

図８において、αは３２ビットの入力データ、γは３２ビットの出力データ、α^0～α^3はαを４分割した各８ビットのデータ、βは１６ビットの拡大鍵、RL1、RL2は左方向へのビット回転処理、RR1、RR2は右方向へのビット回転処理を示している。

図９にRL1の構造を示す。RL1は、１バイトの入力データの下位２ビットの表す０～３の値のビット数だけそのデータを左方向へ回転する処理を行う。図１０にRL2の構造を示す。RL2は、１バイトの入力データの下位から４ビット目と３ビット目の２ビットの表す０～３の値のビット数だけ、そのデータを左方向へ回転する処理を行う。RR1、RR2については、それぞれRL1とRL2のビット回転方向を右にした処理を行う。

ＴＹＰＥ－３のｆ関数の動作について説明する。まず、入力された３２ビットのデータはα^0～α^3の各８ビットの４データに分けられる。出力γは、以下の手順で得られる。

$\gamma^0 = \alpha^0 \oplus \beta^0$
$\gamma^3 = \alpha^3 \oplus \beta^1$
$\alpha^1 = \alpha^1 \oplus \gamma^0$
$\alpha^2 = \alpha^2 \oplus \gamma^3$
$\gamma^0 = RL2(\gamma^0) \oplus RR1(\alpha^2)$
$\gamma^1 = RR1(\gamma^1) \oplus RL2(\alpha^2)$
$\gamma^2 = RR2(\gamma^2) \oplus RL1(\alpha^1)$
$\gamma^3 = RL1(\gamma^3) \oplus RR2(\alpha^1)$

ＴＹＰＥ－３のｆ関数について性能を示す。まず、ソフトウェアで実現した場合はデータによっても異なるが、ＴＹＰＥ－１、ＴＹＰＥ－２より１０％～２０％ほど処

理速度が低下する見通しである。

このｆ関数をハードウェアで実現した場合、図８中に点線で示した処理を並列化することができる。従って、３段の排他的論理和処理と、１段のビット回転数算出処理の構成となる。ビット回転数の算出処理は、排他的論理和とほぼ同じかそれ以下の処理時間で実現できるため、ＴＹＰＥ－２より、さらに早い構造であるといえる。

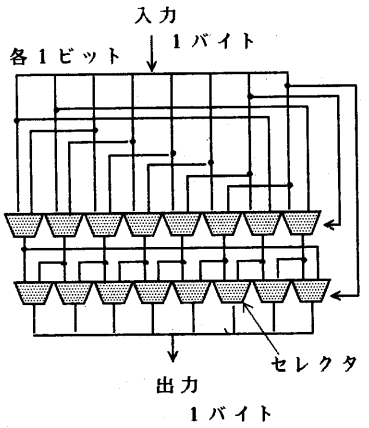

図９　ＲＬ１の構造

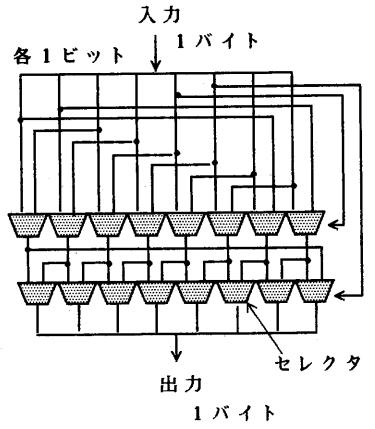

図１０　ＲＬ２の構造

ＴＹＰＥ－３のｆ関数を秘密鍵暗号アルゴリズムに適用した場合についても、暗号強度評価指標のうち、平文変化指標の測定結果について図５，図６に示している。

測定結果からわかるように、データ処理部の３段目の処理を終えた段階で、データはほぼ完全に乱数化されている。これは、ＴＹＰＥ－１のｆ関数の性能とほぼ同等である。

さらに、データによってビット回転数をかえる処理を導入したことで暗号強度を向上させている。したがって、データ処理部４段の暗号アルゴリズムを実現できる可能性がある。

９．おわりに

本論文では、高速マルチメディアＬＡＮ環境への秘密鍵暗号方式の適用について論じた。

動画を含むマルチメディアに対して、ＯＳＩセキュリティアーキテクチャにおけるレイヤ２の機密保護サービスを実現するために、高速暗号ＬＳＩによりＭＡＣ副層で共通に暗号化する方式を示した。さらに、高速暗号ＬＳＩを実現するための要素技術として、データ乱数化処理部であるｆ関数の構造に着目して３タイプの構造ならびに性能を示した。性能については、ハードウェアとしての速度性能、データ乱数化性能に２点について考察した。

本論文で示したｆ関数により、数十Ｍbps～数百Ｍbpsの秘密鍵暗号ＬＳＩを実現することができると考えている。

今後の課題として、暗号アルゴリズム全体の具現化と安全性の検証、さらに、ＬＳＩ技術と性能の詳細な検討などがあげられる。

謝辞

本報告をまとめるにあたりご指導いただいた、ＮＴＴ通信網総合研究所個別通信網研究部柏村卓男部長、木下研作主幹研究員、また、ハードウェアの性能についてコメントいただいたＬＳＩ研究所カスタム化技術研究部平田道広研究主任に感謝いたします。

文献

[1] ISO/DIS 7498-2

[2] H. Goto, F. Akashi, B. Hirosaki, H. Shimizu: "A 1.2Gbps Optical Loop LAN for Wideband Office Communications", GLOBCOM'85, 462.

[3] R. M. Falconer, J L Adams: "Orwell: a protocol for an integrated services local network", Br Telecom Technol J, Vol 3 No 4, (1985).

[4] NBS : "Data Encryption Standard", FIPS-PUB-45, (1977).

[5] 宮口, 平野: " 暗号／認証アルゴリズム強度評価指標", 信学論(A), J69-A, 10, pp. 1252-1259, (1986).

[6] A. G. Kohnheim, Cryptgraphy: "A primer, clause 6.6 INVOLUTIONS, pp 236-240", A Wiley Interscience Publication, John Wiley & Sons, New York (1986).

[7] 清水, 宮口: " 高速データ暗号アルゴリズムＦＥＡＬ", 信学論(D), J70-D, 7, pp. 1413-1423, (1987).