

ストリーム暗号 の最新動向

－ ユビキタス社会を実現するセキュリティ技術 －



田中俊昭* 清本晋作* 櫻井幸一** *(株)KDDI研究所 ** 九州大学大学院システム情報科学府

暗号アルゴリズムの研究は日進月歩である。共通鍵暗号の一種であるストリーム暗号については、軽量・高速に動作し、かつ、ハードウェア化が容易なため、従来、無線、メディアの暗号化などさまざまな用途で用いられてきた。また、ユビキタス社会においては、ICチップやセンサなどスモールデバイスを利用するユースケースも想定され、このようなスモールデバイスに対するセキュリティ確保を目的とした組み込み系アルゴリズムへの応用も期待されている。しかしながら、従来、その安全性については、客観的な評価手法が確立していないため、方式が非公開とされる場合が多かった。このためストリーム暗号の解説書として、すでにいくつかあるものの^{1), 2)}、これらの最新動向について解説したものは数少ない。一方、ストリーム暗号に関する解析技術が進展し、最新の研究成果を応用した新たなストリーム暗号が種々提案されている。したがって、本稿では、ストリーム暗号の概要・特徴（ブロック暗号との違いなど）に加えて、その研究開発の動向、その安全性に関する議論や、標準化の最新動向について分かりやすく解説する。さらに、どのような分野への応用が期待されるのかなど適用例についても詳しく述べる。

ストリーム暗号とは

暗号方式には、大別して共通鍵暗号方式と公開鍵暗号方式がある。ストリーム暗号は、暗号化と復号の鍵が共通の共通鍵暗号方式である。また、共通鍵暗号方式には、ブロック暗号、ストリーム暗号およびバーナム暗号がある。以下、本稿では、暗号化する対象のデータを平文、平文に対応する暗号化されたデータを暗号文と呼ぶ。現代の暗号においては、鍵が第三者に知られないことが安全性の担保となっている。

ブロック暗号は、図-1のように、暗号器と復号器より構成される。暗号処理は、ある鍵が与えられたとき、平文をある一定の長さのブロックに分割し、分割された

平文を暗号器に入力し、ブロック単位に暗号化する。その出力である暗号化されたブロックを入力ブロックの順番に結合し暗号文を形成する。通常ブロックは、64ビット、128ビットなどの単位で処理されることが一般的である。復号処理は、暗号化と同一の鍵を復号器に与え、暗号文をブロック単位に分割、復号器に入力し、分割された平文を出力する。この分割された平文を暗号化のブロックと同一の順序で結合することにより、最終的に平文を得る。代表的な方式としては、米国の標準であるDES (Data Encryption Standard) やAES (Advanced Encryption Standard) などがある。

バーナム暗号は、平文と同じ長さの乱数列（鍵系列）を用いて、平文と乱数列をビットごとに排他的論理和（ビットごとのmod2の足し算）により暗号文を作成する。復号時には、復号処理側も何らかの方法で同じ乱数列を事前に共有し、暗号文とその乱数列の排他的論理和を行うことで、元の平文が生成される。ここで、乱数列が真性乱数（半導体の熱雑音などの物理現象を利用して生成するまったく予測不可能な0,1のビットパターン）の場合には、平文がどんな情報（たとえば、統計的に偏った情報）であっても、情報理論的に安全であることが知られている。このような鍵系列が真性乱数で、毎回使い捨てする利用方式をワンタイムパッドと呼ぶ。実際に利用する場合には、平文と同じ長さの乱数列を暗号化処理と復号処理の双方で事前に共有しなければならない。このような鍵の共有手順や鍵の保管が現実的ではない。このため、バーナム暗号は非常に高いセキュリティが要求されるユースケースに限定されており、一般的にはあまり用いられない。

一方、ストリーム暗号は、図-2のように暗号化の際には、ある鍵と初期値が与えられ、平文と同じ長さの鍵系列を生成し、平文とその鍵系列を排他的論理和により暗号文を作成する。復号の際には、同一の鍵と初期値を与え、決定的に生成される乱数列と暗号文の排他的論理和を計算することで元の平文が得られる。ストリーム暗

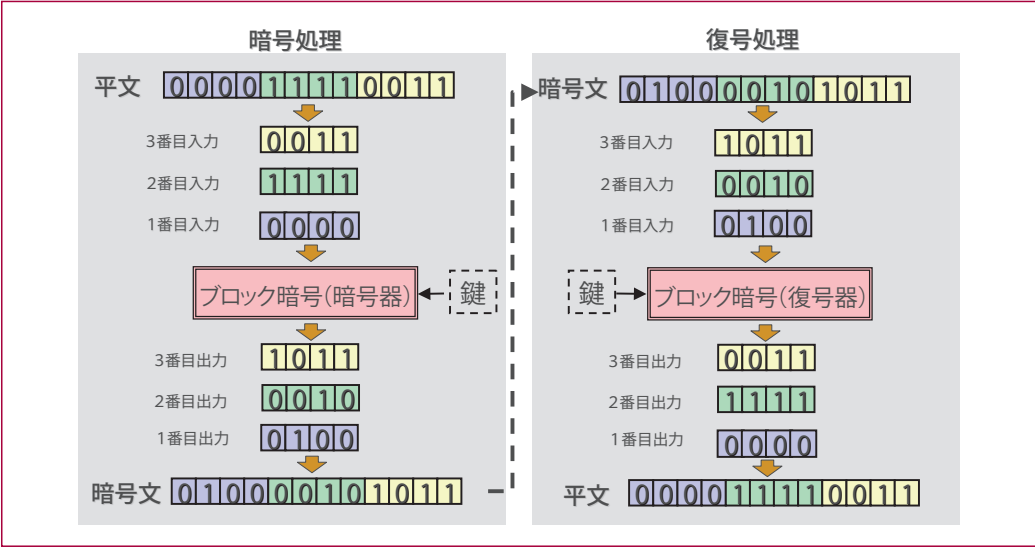


図-1 ブロック暗号の構成

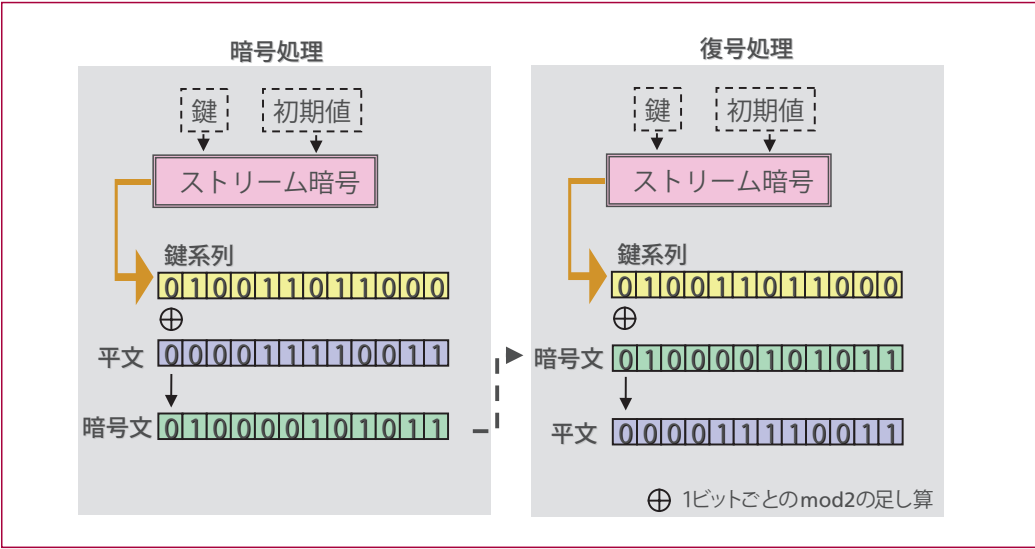


図-2 ストリーム暗号の構成

号では、1回の処理ごとに生成される鍵系列の長さ（以下、鍵系列の生成単位と呼ぶ）が、通例、1ビットや1バイト程度と短く、この処理を繰り返し行うことにより鍵系列が生成される。一見、バーナム暗号と似ているが、鍵と初期値が決まれば、生成される鍵系列が決定的である点が、真性乱数と異なる。しかしながら、鍵を知らない第三者にとっては、乱数のように見えるため、このような鍵系列は擬似乱数と呼ばれる。

ストリーム暗号の歴史は、ブロック暗号よりも古く、16世紀頃考案されたヴィジユネル暗号は、鍵として周期的な文字列を用いており、広い意味でストリーム暗号といえる。かつて暗号が主として軍事目的で用いられていたころは、ハードウェア化が容易であるストリーム暗号が用いられていたようである。特にストリーム暗号の研究は、効率的な擬似乱数生成の研究とも密接に関連しており、後述のLFSR（線形フィードバックレジスタ）やNLFSR（非線形フィードバックレジスタ）の研究成果が

	ブロック暗号	ストリーム暗号
暗号文の長さ	ブロック単位	平文と同一
内部状態	保有せず	保有
同期の必要性	無	有（同期ずれに対する対策要）
誤り伝播	ブロック単位	鍵系列の生成単位

表-1 ブロック暗号とストリーム暗号の特徴の比較

現在も活用されている。

ここで、ストリーム暗号の性質を整理するためブロック暗号との比較を行う（表-1）。ブロック暗号は、ブロック長を単位（たとえば、64bitなど）として暗号化を行うため、任意の長さの平文を暗号化するためには、ブロック長の整数倍の長さに調整する必要がある。最後のブロックもブロック長となるようパディング処理を行う。これに対して、ストリーム暗号は、平文と同一の長さの鍵系列を生成することにより、任意の長さの平文に対し

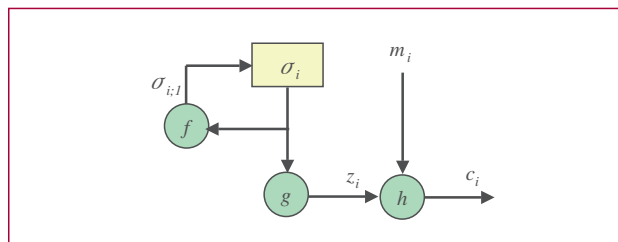


図-3 同期型ストリーム暗号の一般化モデル

て、同一の長さの暗号文が生成できる。このように、ストリーム暗号は、余分なデータを処理する必要がないことから、帯域が限られた無線通信などで有利である。

次に、ブロック暗号は、内部状態を持たない暗号方式であり、鍵が固定であれば、入力される平文に対して、出力される暗号文は一意に決定される。一方、ストリーム暗号は、擬似乱数を生成するために内部に状態を持つ。内部状態とは、暗号・復号の処理過程の中で生成される中間的なデータ群を表し、通常、メモリ内に保管される。この内部状態は、鍵と初期値によって一意に決定されるが、ストリーム暗号の処理が進むにつれて内部状態が時々刻々変化（遷移）していく。つまり、内部状態が異なれば、同じ平文を暗号化しても生成される暗号文が異なる。すなわち、内部状態が時間とともに変化するために、暗号処理、復号処理間で同期が必要となる。このためストリーム暗号は別名、State Cipherとも呼ばれる。

暗号文を送信する際は、通信によるデータの誤りが発生する可能性がある。たとえば、暗号文のなかである1bitが誤って送信された場合、ブロック暗号では、該当するブロック全体に誤りが伝播されるのに対して、ストリーム暗号では、通常、鍵系列の生成単位の誤りに限定される。

ストリーム暗号の構造

ストリーム暗号の研究は、現在、盛んに行われているが、その設計においては、現状、ブロック暗号のような確立された手法が存在しない。したがって、さまざまな構造のストリーム暗号が考案されている。本章では、代表的な方式について解説する。

【同期型】

前章の述べた暗号処理と復号処理で同期が必要なストリーム暗号を同期型ストリーム暗号と呼ぶ（図-3）。本方式では、同期ずれが発生した場合に回復ができないため、実用においては、定期的に再同期を行う。同期型ストリーム暗号を一般化したモデルを以下に示す。

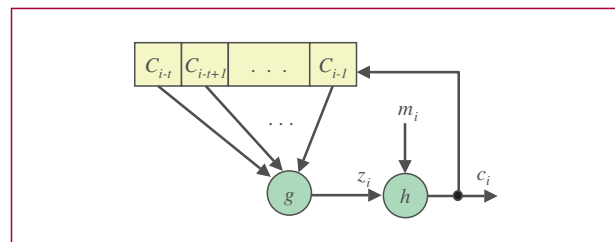


図-4 自己同期型ストリーム暗号の一般化モデル

$$\begin{aligned}\sigma_{i+1} &= f(\sigma_i), \\ z_i &= g(\sigma_i), \\ c_i &= h(z_i, m_i)\end{aligned}$$

ここで、 m_i, c_i, z_i, σ_i はそれぞれ、平文、暗号文、鍵系列、状態を、また、 f, g, h はそれぞれ、状態遷移関数、鍵系列 z_i を生成する関数、出力関数を表す。 σ_0 は初期状態であり、鍵 k と初期値 IV より生成される。一般的に出力関数 h は排他的論理和が用いられる。

σ_i は内部状態で、1回の処理を実行するごとに、内部状態のカウント値 i が1ずつ増加していく。すなわち、図-3では、 i 回目の処理において、内部状態 σ_i をそれぞれ、関数 f, g の入力とする。 g は直ちに処理を実行し z_i を出力する。出力関数 h は z_i と平文 m_i を入力として、暗号文 c_i を出力する。関数 f は、 $i+1$ 回目の処理における内部状態 σ_{i+1} を出力する遅延回路である。すなわち、 i 回目の処理が終了した時点で内部状態 σ_i は σ_{i+1} に遷移する。この処理を $i=0$ 、すなわち初期状態から繰り返し行い、最終的に暗号文 $C_0, C_1, \dots, C_i$ を出力する。

【自己同期型】

同期ずれに対して自動回復の機能を持つストリーム暗号を自己同期型ストリーム暗号と呼ぶ。自己同期型では、ある一定時刻（ t 時刻）までさかのぼった暗号文列のみから鍵系列が生成される（図-4）。自己同期型ストリーム暗号を一般化したモデルを以下に示す。

$$\begin{aligned}\sigma_i &= (c_{i-t}, c_{i-t+1}, \dots, c_{i-1}) \\ z_i &= g(\sigma_i), \\ c_i &= h(z_i, m_i)\end{aligned}$$

以下、ストリーム暗号の設計において核となる擬似乱数生成のための構造について述べる。

【LFSRに基づく方式】

前述のようにLFSRは、良好な乱数性を有する、回路設計などハードウェア化が容易である、周期性などの振舞いが解析しやすいという点で、従来からストリーム暗

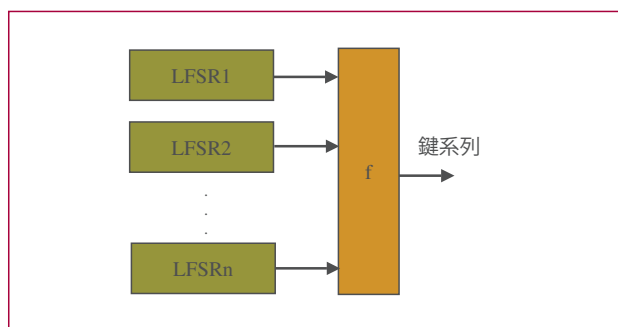


図-5 非線形コンバイナ型

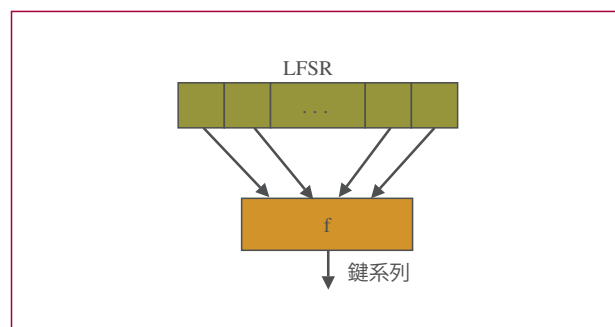


図-6 非線形フィルタ型

号の擬似乱数生成機能として用いられてきた。すなわち、 n 個のレジスタから構成され、フィードバック関数が

$$p(x) = 1 + c_1x + c_2x^2 + \dots + c_nx^n \text{ over } GF(q)$$

で構成されるLFSRは、 $p(x)$ が原始多項式のとき最大周期 $q^n - 1$ となることはよく知られた定理である。LFSRは擬似乱数生成器としては有効であるが、暗号方式として利用するには各種既存の攻撃に対する耐性を持たないため、このまま利用できない。このため、LFSRに非線形要素を加える工夫がなされている。

• 非線形コンバイナ型

最大周期を持つ複数のLFSRの出力を非線形ブール関数に入力し、乱数系列を得る方式である。本方式は線形複雑度を高めることができる点で優れている(図-5)。

• 非線形フィルタ型

最大周期を持つ1つのLFSRを用いて、LFSRの各レジスタの値を入力とする非線形関数 f から鍵系列が生成される方式である。この f はフィルタ関数と呼ばれる(図-6)。

【独自構造を有する方式】

LFSRを用いたストリーム暗号が多く提案されているが、LFSRを用いない独自構造を用いたストリーム暗号も種々提案されている。その代表例としては、RC4がある。RC4は、SSLや無線LANの規格であるWEP(Wired Equivalent Privacy)で用いられており、ソフトウェアでの高速性を確保した方式である。本方式は、インデックスのあるテーブルを用いてテーブルの個々の値に依存してテーブルの値をシャッフルし、一部のテーブルの値を鍵系列として出力する方式である。RC4は、公式には非公開アルゴリズムである。

【ブロック暗号を利用した方式】

ブロック暗号を利用したストリーム暗号も考案されている。暗号利用モードと呼ばれ、ISOの国際規格として標準化されている³⁾。各種利用モードの中でも、同期型ストリーム暗号に分類されるOFB(Output FeedBack)

モード、CTR(Counter)モードおよび自己同期型ストリーム暗号に分類されるCFB(Cipher FeedBack)モードがある。安全性がブロック暗号の安全性に帰着する点で評価がしやすいという利点がある。ここでは、紙面の都合上、OFBモードについてのみふれる。OFBモードは、ブロック暗号を用いて固定の鍵 k で、初期値から生成されるデータを暗号化し、その出力を次の暗号処理の入力とする。これを繰り返して生成されるデータを鍵系列として平文と排他的論理和をとる手法である。ここで、 Enc_k は鍵 k によるブロック暗号の暗号処理を表す。

$$H_i = Enc_k(H_{i-1})$$

$$C_i = M_i \text{ XOR } H_i$$

ストリーム暗号の安全性について

暗号の安全性評価は、暗号性能の客観性を確保するために必須の指標である。ここで、ストリーム暗号の安全性評価においては、攻撃者として既知平文攻撃(攻撃者が選択不可能な既知の平文に対応する暗号文を得られる条件で、暗号文から平文を求める攻撃)が可能な攻撃者を想定するため、攻撃者は、ストリーム暗号が出力する鍵系列を入手可能であるとの前提で議論を行う。なぜなら、既知である平文と暗号文の排他的論理和をとることによって、鍵系列が入手可能だからである。したがって、安全性の評価は、鍵系列に対して行う。ストリーム暗号の安全性評価には、大別して、乱数性を評価する統計的検定と、さまざまな攻撃に対する安全性評価がある。

【統計的検定】

統計的検定は、一定の大きさの鍵系列を複数使用し、その鍵系列群を用いてさまざまな検定を実施する。たとえば、代表的な検定に線形複雑度検定がある。線形複雑度検定は、鍵系列をある長さのブロックに分割し、各ブロックの線形複雑度をBerlekamp-Masseyアルゴリズムを用いて導出し、それらの値を用いて鍵系列の線形複雑度

を評価するものである。線形複雑度とは、ある系列が与えられた場合に、その系列を生成することができる最小のLFSRの段数であり、ストリーム暗号の安全性として、線形複雑度が十分大きいことが求められる。

統計的検定では、通常複数の検定手法を組み合わせ、総合的に安全性が評価される。代表的な検定セット（検定手法の組合せ）として、NIST SP 800-22があり、NIST SP-800-22では16種類の検定手法が用いられている。また、CRYPTRECでは、擬似乱数生成系の検定方法に関する調査報告書において、ミニマムセットとして、12種類の検定を行うことを推奨している。

次にストリーム暗号に対する代表的な攻撃手法について述べる。ストリーム暗号では、初期鍵を拡大してある内部状態を生成し、内部状態を遷移させながら、一部分を鍵系列として出力する。無論、初期鍵が導出されれば、そのストリーム暗号は破られたとみなされるが、内部状態のすべてのビットが導出された場合も、攻撃成功と判断される。なぜなら、一度内部状態を導出してしまえば、それ以降の鍵系列が生成可能で、攻撃者が以降の暗号文を復号可能となるためである。

【Time-Memory-Dataトレードオフ攻撃】

本攻撃では、事前計算によって、あらかじめ鍵系列と内部状態の関係を導出しておき、実際の鍵系列から内部状態を求める計算を効率化する手法である。この攻撃手法は、Alex BiryukovらによってA5に適用されて大きな効果を上げており、個々のストリーム暗号アルゴリズムに対して最適化することで、効率を上げる手法も研究されている。また、Hongらによって一般的な安全性評価指標も提案されている。

【関連攻撃】

本攻撃は、鍵系列と内部状態との相関を利用して内部状態の推定を行う攻撃法であり、非線形コンバイナを連結したストリーム暗号に対する代表的な攻撃手法となっている。また、分割統治攻撃は、内部状態のうち、ある一部分に絞って攻撃を行い、その後確定した一部分の内部状態を有効に活用しながら他の内部状態を推定する攻撃手法である。代表的な攻撃例として、ChenらのAlpha-1への分割統治攻撃がある。

【推測決定攻撃 (Guess-and-Determine Attack)】

本攻撃は、ワードオリエンテッドなストリーム暗号アルゴリズムに対する攻撃法として知られている。推測決定攻撃は、攻撃者が内部状態のいくつかのビットを予測し、その予測した値と鍵系列を用いて内部状態の別のビットを決定する攻撃手法である。この攻撃手法におい

て、攻撃者は、鍵系列、予測した一部の内部状態、未知の内部状態からなる式を用いて、未知の内部状態を決定するが、ワードオリエンテッドなストリーム暗号では、この評価式の各項がそれぞれワード長の値で与えられるため、より多くの内部状態を決定できる。Howkesらは、いくつかの効果的な評価式を用い、評価式に含まれる非線形変換部分のある確率で成立する仮定を導入することで、SNOW1.0への推測決定攻撃が可能であることを示した。

【代数的攻撃】

ブール関数に代表されるような非線形変換では、出力ビットは、入力ビットの代数式で与えられる。代数的攻撃は、多くの鍵系列から得られる代数式の2次以上の項を別の1次項で置き換えることにより線形化し、線形化した式を用いて連立一次方程式を解くことにより、非線形変換の出力ビットから入力ビットを求める攻撃手法である。多くのストリーム暗号アルゴリズムでは、内部状態のいくつかのビットがそのまま非線形関数に入力されるため、入力ビットを求めることは内部状態を求めることと等しくなる。Courtoisらは、代数的攻撃手法によって、Toyocrypt, LILI-128への攻撃を成功させている。また、一般的な評価指標についても検討を行っている。

【識別攻撃】

識別攻撃は、他の攻撃手法とは異なる攻撃である。他の攻撃は、内部状態を導出することが目的であったのに対し、識別攻撃は、真の乱数列に対して、ストリーム暗号が出力した擬似乱数列である鍵系列を識別することを目的とする。識別攻撃の成功が直ちに、内部状態の導出に繋がるものではないが、攻撃の仮定で導出した線形式を利用することにより、内部状態を導出できる場合もあるため、識別攻撃に対して安全であることが望ましいとされている。識別攻撃では、まずDistinguisherとして、鍵系列のビットのみからなる線形式を導出する。本攻撃手法は、ブロック暗号における線形攻撃に対応しており、ブロック暗号で用いられていた手法が応用できる。本攻撃手法は非常に強力で、多くの暗号方式が識別攻撃によって破られている。

最近の動向

ストリーム暗号は、内部状態を遷移させる関数と、内部状態を出力する際に非線形変換を行うフィルタ関数から構成されることが多い。好例は、非線形コンバイナ型ストリーム暗号である。従来、安全なストリーム暗号を構成するための設計指針の研究は、きわめてシンプルな

構造のアルゴリズムに対して、あるいはLFSR、ブール関数といった個々の機能部品に対してなされており、前章で挙げたような攻撃に対する安全性を保証する設計指針や、評価手法の確立には至っていない。また、攻撃手法はアルゴリズムに依存することも多く、さまざまなアルゴリズムが存在するストリーム暗号においては、統一した評価基準を検討することが困難となっている。しかし、前章で挙げたHongやCourtoisらのような一般化した評価手法を検討する試みもなされている。また、欧州では、eSTREAMというストリーム暗号アルゴリズム評価プロジェクトを通して、統一的な安全性評価基準の策定を進めている。アルゴリズムの形式分類と、評価手法の検討が進めば、近い将来、アルゴリズムの設計指針も確立されるのではないかと考えられる。

一方、設計したアルゴリズムの効率性を評価する指標として、ZennerはInner State Efficiency (ISE) という評価尺度を提案しており⁴⁾、アルゴリズムの性能評価尺度の1つとして、処理性能とともに重要視されている。これは、 $(\log_2(\text{最良の攻撃の計算量}) / (\text{内部状態のビット長}))$ で計算される指標であり、1に近づくほどより優れた設計であるといえる。なぜなら、ストリーム暗号の安全性は、内部状態のビット長に大きく依存しており（一般に、内部状態のサイズが大きいくほど、安全なストリーム暗号を実現できる）、より短いビット長で、より高い安全性を達成したほうが、効率的な設計手法であるといえるからである。なお、安全なストリーム暗号であれば、（一般的には、内部状態のビット長は、初期鍵のビット長よりも長いので）最良の攻撃は初期鍵の全数探索になるため、ISEは、 $(\text{対応する鍵のビット長}) / (\text{内部状態のビット長})$ となる。代表的なストリーム暗号のISEを見ると、SNOW2.0では0.44、Rabbitでは0.25である。

従来ストリーム暗号は、ハードウェア実装を想定して設計されていたため、ビット単位で処理を行うアルゴリズムが多かった。しかし、近年、ソフトウェア実装において高速処理を実現するために、CPUのワード長単位での処理を基本とするアルゴリズムが考案されている。たとえば、SNOW2.0では、LFSRの各レジスタのサイズを32bitsとし、テーブル参照を用いたフィードバック関数を実装することで、ソフトウェア実装における高速処理を実現している。これは、LFSRは32ビットのレジスタで構成し、レジスタ単位でシフトするが、フィードバックにおいて、レジスタ内のデータ8ビット×4に分割したデータとして扱い、8ビットシフトさせる処理を行うというものである。はみ出た8ビットの処理は、事前計算により作成しておいたテーブルを参照することで処理を高速化する。LFSRを定義する多項式は、複数の多項式で逐次的に構成されるが、ビット単位のLFSRと見た

場合に、原始多項式となるように構成されており、その周期性が保証されている。現在、後述のeSTREAMで評価されている最新のストリーム暗号アルゴリズムの多くは、ソフトウェア実装において、5～8 cycle/byteの鍵系列生成速度を実現しており、AESと比較すると4～5倍の処理性能を達成している。ワード単位で高速化を実現している国産暗号の例としては、KCipher-2がある。携帯電話上のアプリ実装においてAESの約10倍の処理性能を達成している⁵⁾。

ストリーム暗号の標準化動向

暗号アルゴリズムを利用して安心・安全なサービスやITシステムを構築するには、ユーザやベンダが仕様として共通に認識できる暗号アルゴリズムを規定する必要がある。このため、各種標準機関において、暗号アルゴリズムの標準化が行われている。標準化については、暗号アルゴリズムのユースケースを特定しない汎用的な標準化と、特定のアプリケーションやサービスを想定した標準化がある。後者については、インターネット、無線通信、コンテンツ保護を目的とした著作権管理などを目的とし、さまざまな機関で標準化が行われている。各組織で標準化されているストリーム暗号の代表的な例を表-2に示す。ここで、無線関係の暗号であるA5やE0は、非公開アルゴリズムであるが、脆弱性が指摘されている。

利用目的が限定されない標準に関しては、ITセキュリティの標準化が所掌である国際標準組織ISO/IEC JTC 1 SC 27において、2005年7月にストリーム暗号の標準ISO/IEC 18033-4が制定された⁴⁾。本標準では、ストリーム暗号の一般モデルとして同期型、自己同期型および、出力関数としてバイナリ加算型、MULTI-S01型を規定している。MULTI-S01は、国産のメッセージ認証付きのストリーム暗号である。さらに、本標準では具体的な構造として、ブロック暗号を利用したOFBモード、CFBモードおよびCTRモードを規定し、さらに、専用の乱数生成器として、MUGIおよびSNOW2.0を採用している。

また、国内においては、政府が企画したCRYPTREC(暗号技術監視委員会)において、電子政府の構築を目的とした政府調達のための政府推奨暗号リストを作成する作業が、2000年から2003年にかけて行われた。公募のあった暗号アルゴリズムに対して、暗号の専門家らが評価を行い、政府推奨リストを作成している。現在、ストリーム暗号については、RC4、MUGI、MULTI-S01がリストに掲載されている。

また、欧州の暗号研究者による暗号評価プロジェクトとして ECRYPT (European Network of Excellence for

	公開／非公開	同期／自己同期	構造	用途	鍵長	標準化団体
RC4	非公開	同期型	独自構造（テーブル参照型）	SSL/TLS (option), SSH (option), WEP/WPA 等	8-2048	IETF
MUGI	公開	同期型	擬似乱数生成機 PANAMA と類似の構造を採用	汎用	128	ISO 18033-4 (乱数生成器)
MULTI-S01	公開	同期型	擬似乱数生成機 PANAMA を採用	汎用	256	ISO 18033-4 (利用モード) 認証付
SNOW2.0	公開	同期型	LFSR (Clock 制御)	汎用	128	ISO 18033-4 (乱数生成器)
A5/1	非公開	同期型	LFSR (Clock 制御)	GSM の携帯電話，基地局間の秘匿方式（CEPT 会員のみが利用可能な strong version）	64	ETSI
A5/2	非公開	同期型	LFSR (Clock 制御)	GSM の携帯電話，基地局間の秘匿方式（輸出規制対象外の weak version）	64	ETSI
E0	非公開	同期型	LFSR	Bluetooth の無線区間秘匿方式	128	Bluetooth
HDCP Cipher	公開	同期型	LFSR, Block module, Output function の組合せ利用	HDMI 規格のコンテンツ保護, 24bit ごとの鍵系列を生成	56+84	DCP LLC

表-2 標準化されたストリーム暗号

Cryptology) が，2004年に発足し，その活動の一環としてストリーム暗号の評価を行う eSTREAM プロジェクトが進められている．本プロジェクトは，欧州の暗号標準を策定する NESSIE プロジェクト（2000～03年）で候補となったストリーム暗号の候補すべてに問題が顕在化して標準化が成し得なかった経緯を踏まえ，AES チャレンジ（米国の標準機関である NIST がブロック暗号の標準化を目的としアルゴリズムを公募し，選定を行ったコンテスト）と同様にストリーム暗号を広く公募し，評価を経て2008年に利用可能なストリーム暗号の候補を絞り込む予定となっている．本プロジェクトにおいては，新たなストリーム暗号が多数提案されており，ストリーム暗号研究の活性化に貢献している．評価作業は2つのフェーズに分けて行われており，2006年3月よりフェーズ2に移行している．構造としては，従来からの LFSR を用いたアルゴリズムや，RC4 のような状態置換型のアルゴリズムに加え，LFSR のフィードバック関数に非線形要素を追加した変形型 LFSR を用いたアルゴリズムや，T-Function と呼ばれる周期関数を用いた方式も提案されている．設計方針としては，（1）理論的な周期性の保証を目指しつつ安全性を向上させる，（2）理論的な周期性を考慮せず，安全性向上のみに努める，という2つの方針に分かれる．従来多くのアルゴリズムで用いられていた LFSR は，周期性を保証する上では便利な部品であるが，自明な線形式がアルゴリズム内に残ってしまう可能性があり，新規アルゴリズムの設計においては LFSR を用いない手法も種々提案されている．eSTREAM プロジェクトの進展によって，今後設計手法や評価手法が確立されていくものと考えられる．

応用例

本章では，ストリーム暗号が用いられている具体的な利用例にふれる．先述のようにストリーム暗号は，計算資源が乏しく高速な暗号が必要となる環境，低コストでシステムを実現する必要がある環境，通信帯域が限定された環境などに向いているといえる．このような要件が必要となる環境としては，携帯電話や無線 LAN などの無線通信，大容量データを扱う AV 機器などが想定される．本章では，製品や実証システムでの事例をもとに，ストリーム暗号の具体的な利用手法を紹介する．

先述のように暗号化データを送受信する際に，伝送エラーや同期ずれなどを考慮する必要がある．同期型ストリーム暗号の場合は，一度同期がずれると回復できなくなるため，一定間隔でストリーム暗号を初期化し，再同期させる対策を実施している．具体的な初期化の手法としては，通信帯域，通信機器の計算能力，通信のパケットフォーマットなどのシステム要件を考慮して，いくつかの方法が提案されている．

【初期値 (Initial Vector: IV) を明示的にデータに入力する手法】

代表的な応用例としては，無線 LAN の規格である IEEE 802.11 のセキュリティ技術である WEP における端末と基地局間の暗号化手法である．WEP のアルゴリズムとしては，RC4 が用いられる．WEP では，図-7 に示すようにパケットごとに IV を挿入し，ストリーム暗号を初期化する方法を用いている．ここで，IV は 24bit である．なお，WEP は固定鍵であることや，IV の長さが

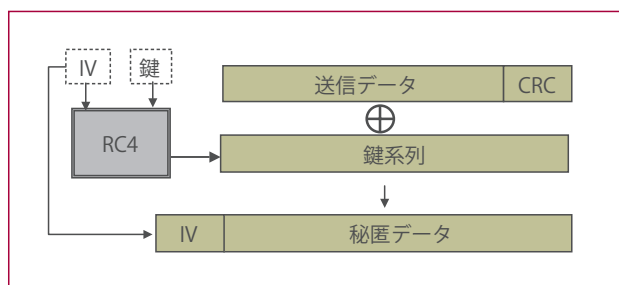


図-7 WEP での暗号化方式

不十分であることが指摘されており、IV を48bitに拡張し、通信時に鍵を定期更新する、メッセージ改ざん機能を追加する等の対策を施したWEPの改良版のTKIPが提案されている。

【IVを一定間隔で挿入する方法】

同期ずれがあまり生じない回線品質での利用、アプリケーションレベルでエラー耐性機能を有する、あるいは、計算資源に制約がある等の条件下では、IVを一定間隔で挿入することにより、初期化の回数を削減し、通信帯域および計算資源を削減する方法がある。実験的な試みとしては、本年4月より放送が始まった携帯電話向けのデジタル放送(1セグ)の放送波を暗号化して放送コンテンツを保護するシステムが提案されている。1セグは処理負荷の高いH.264/AVCの符号化を用いているため、携帯電話でリアルタイムの復号、再生を行うには暗号化された放送波の復号処理が可能な限り軽量であることが望ましい。本システムでは、ストリーム暗号を用い、1セグの放送用データフォーマット仕様にIV専用のパケットを追加して、IVを一定間隔(たとえば、0.5秒ごと)に挿入する方法を用いている。

むすび

本稿では、共通鍵暗号の一方式であるストリーム暗号の最新動向について概説した。ストリーム暗号は、インターネットにおいて実用的な暗号化方式として広く使われているのみならず、回路設計が容易であることから、携帯電話やデジタル機器などの組み込み系アルゴリズム

ムとして実用化されてきた。また、最近では、構造の多様性と安全性評価技術の確立という観点で学際的にも注目を集めている。来るべきユビキタス社会においては、さまざまな通信機器が環境に組み込まれるため軽量、高速なアルゴリズムに関して高いニーズがあると思われる。ストリーム暗号はそのニーズにこたえる暗号方式として、今後ますます注目を浴びるであろう。

参考文献

- 1) Menezes, A., Oorschot, P. van and Vanstone, S. : Handbook of Applied Cryptography, CRC Press (1996).
- 2) Rueppel, R. A. : Analysis and Design of Stream Ciphers, Springer-Verlag (1986).
- 3) ISO/IEC 18033-4 : Information Technology - Security Techniques - Encryption Algorithms Part 4 Stream Ciphers.
- 4) Zenner, E. : On the Role of the Inner State Size in Stream Ciphers, available at <http://th.informatik.uni-mannheim.de/pub/zenner04a.pdf#search=%22Inner%20state%20efficiency%22>, Reihe Informatik 01-2004 (2004).
- 5) Kiyomoto, S., Tanaka, T. and Sakurai, K. : Design of a Stream Cipher using Efficient Clock Control, IEICE Technical Report (2006).

(平成 18 年 11 月 6 日受付)

田中俊昭 (正会員)
toshi@kddilabs.jp

1986 年 KDD (株) 入社。現在、(株) KDDI 研究所情報セキュリティグループリーダー。暗号理論、暗号プロトコル、著作権保護、モバイルセキュリティの研究に従事。

清本晋作 (正会員)
kiyomoto@kddilabs.jp

2000 年 KDD (株) 入社。現在、(株) KDDI 研究所情報セキュリティグループ研究主査。ストリーム暗号、暗号プロトコル、モバイルセキュリティの研究に従事。博士 (工学)。

櫻井幸一 (正会員)
sakurai@csce.kyushu-u.ac.jp

九州大学大学院システム情報科学府情報工学部門教授。2004 年より九州システム情報技術研究所第二研究室長併任。暗号理論・情報セキュリティ・社会情報工学の研究に従事。博士 (工学)。