

O_001

ソフトウェア実装を想定した自己同期型ストリーム暗号の検討 Study of a Self-Synchronous Stream Cipher for Software Implementation

清本 晋作† 田中 俊昭† 櫻井 幸一‡
Shinsaku Kiyomoto, Toshiaki Tanaka, Kouichi Sakurai

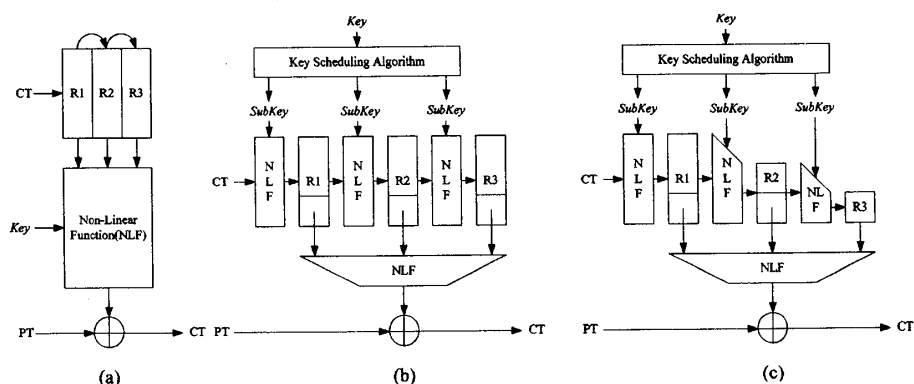


図1 自己同期型ストリーム暗号の構成例

1. まえがき

映画など大容量データの伝送を保護するため、ブロック暗号よりも高速な暗号方式が注目されている。ストリーム暗号は、ブロック暗号よりも高速な処理が期待できる暗号方式であるが、暗号処理と復号処理の同期を取らなければならないという問題点があった。そこで、一定サイズ以上の誤りなく受信した暗号文を利用して、同期を自動的に回復する自己同期型のストリーム暗号が研究されている。自己同期型ストリーム暗号は、VOIP など帯域幅が制限され、同期用データを伝送できないようなサービスに特に有効な方式である。

自己同期型ストリーム暗号は、受信した暗号文と共通鍵を入力として、暗号化のための鍵系列を生成するアルゴリズムである。1991年に、Maurer は、一般的な自己同期型ストリーム暗号の構成手法を提案した[1]。また、Deamenらは、2005年に、MOSQUITO と呼ばれる、Maurer の構成手法を利用したハードウェア実装向け自己同期型ストリーム暗号を提案している[2]。MOSQUITO は、ビット単位での処理を基本としており、ソフトウェア実装においては、必ずしも最適な設計ではない。また、Joux らによって選択暗号文攻撃が提案されている[3]。

自己同期型ストリーム暗号は、ブロック暗号の CFB と類似の機能を提供するが、ステートフルなアルゴリズムとすることで、非線形関数などの演算を軽量化しても、同等の安全性が確保できる。従って、同期回復に時間を要するが、より高速に処理できることが期待され、TS ストリームなどのパケット単位での大容量データ伝送における暗号化/復号処理に向いている。

本稿では、ソフトウェア実装において高速な処理を目指した自己同期型ストリーム暗号の構成手法を提案し、安全性を確保するための各パラメータの設計手法について考察

を行なう。提案方式では、ワード長を処理単位とすること、出力鍵系列長を大きく取り内部状態のレジスタを多段とすることで、ソフトウェア実装における高速処理を実現する。

2. 自己同期型ストリーム暗号の構成

2.1 構成方法の比較

ハードウェア実装などを想定したビット単位で処理を行う自己同期型ストリーム暗号は、ある L ビットの内部状態に対して、1 ビットの鍵系列を出力する処理を行う。また、次の鍵系列を出力する前に 1 ビットの暗号文が入力され、 L ビットの内部状態の内、もっとも古い 1 ビットが廃棄され、 $L-1$ ビットの内部状態に入力された 1 ビットの暗号文を加えたものが、新たな内部状態となる。ソフトウェア実装を考慮した場合、ワード単位での処理が望ましい。従って、暗号化/復号処理においては、内部状態 L ビットに対して、 mw ビット(m は自然数)の鍵系列を出力し、 mw ビットの暗号文を入力して内部状態を更新する必要がある。このような自己同期型ストリーム暗号の構成方法として、図1に示す(a),(b),(c)の3種類の方法がある。(a)は、従来のビット単位の自己同期型ストリーム暗号をそのままワード単位の自己同期型ストリーム暗号に拡張した方式である。この方式では、すべての内部状態が非線形関数(NLF)に入力され、鍵系列が計算される。一方、(b)は内部状態の一部を NLF へ入力するように変更した方式である。鍵系列を出力する NLF への入力ビット数を小さくして軽量化する一方で、レジスタ間のデータ移動の処理にも、非線形関数を導入して安全性の強化を図る。最後に、方式(c)は方式(b)を軽量化した方式である。具体的には、内部状態を保持するレジスタ数を徐々に絞っていくことで、レジスタ間のデータの移動の際の処理を軽量化する。方式(a)は全レジスタのデータを NLF に入力するため、鍵系列を生成する NLF の処理負荷が大きくなる可能性があると考えた。また、方式(b)と方式(c)では、方式(c)のほうが処理負荷は小さいと考えられるため、本稿では、方式(c)について検討を行なうこととした。

† (株) KDDI 研究所, KDDI R&D Laboratories Inc.

‡ 九州大学, Kyushu University

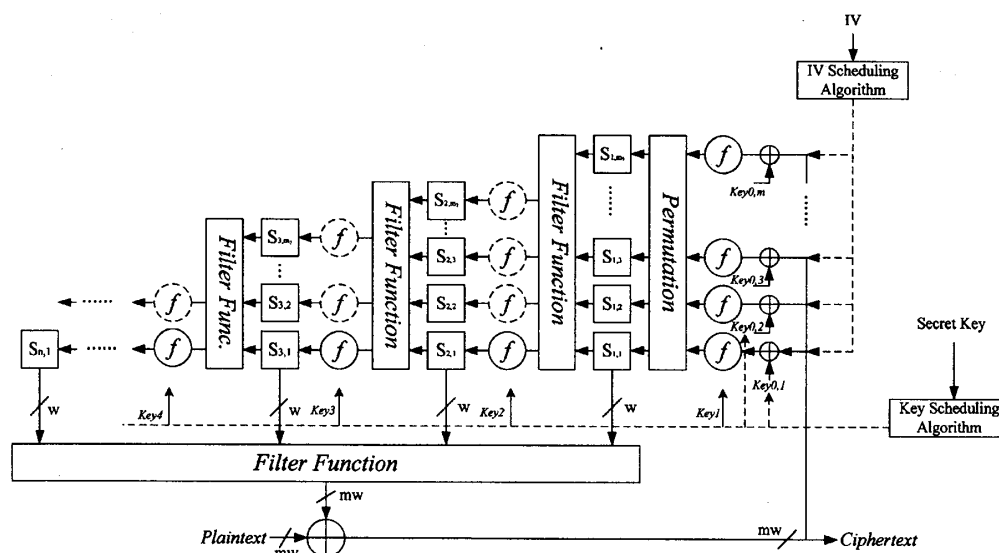


図2 方式(c)に従った自己同期型ストリーム暗号の構成例

なお、アルゴリズムの構造は安全性を考慮して設計しなければならないため、(a), (b), (c)の比較で(a)が一概に処理負荷が大きいいとはいえない。例えば、(a)においても NLF を十分な安全性を満たす範囲において軽量化することで、結果として処理負荷が小さくなる可能性がある。本稿では、(a), (b), (c)の内部状態の大きさは同じ大きさとする、小さいデータを処理するほうが処理負荷を小さくできるという前提のもとで、(c)について検討を行なうこととしている。

2.2 構成手法の検討

方式(c)の構成について述べる。図 1 に構成を示す。S_{i,j} をレジスタと呼び、そのサイズを使用する環境のワード w とする。提案方式は、一般的な鍵スケジューアルgorithmを利用して拡大した初期値(IV)を各レジスタに入力して、初期化を行なう。そして、 mw ビットの平文入力に対し、 mw ビットの鍵系列を生成して排他的論理和演算をすることにより、 mw ビットの暗号文を出力する。鍵系列は最下段のレジスタ値を入力として、フィルタ関数 $g: mw \rightarrow mw$ により計算される。暗号化処理後、各レジスタの状態は、非線形関数、およびフィルタ関数によって更新される。

フィードバックする暗号文に対する排他的論理処理及び各非線形関数には k ビットの共通鍵から拡大した拡大鍵が入力される。復号側では、随時、受信した暗号文 mw を最右行のレジスタに入力していく。暗号文は、拡大鍵 $Key_{0,1} - Key_{0,m}$ でマスクされた後、非線形変換処理、攪拌処理を経て、各レジスタに入力される。本アルゴリズムでは、TS ストリームのように、一定のデータサイズごとに暗号文を受信するアプリケーションを想定している。すなわち、上記データサイズ内において、ビット誤りが生じた場合は、同データサイズ分の暗号文が破棄されると仮定している。従って、 $n \cdot mw$ ビットの暗号文を正しく受信すれば、同期を回復することができる。 n は安全性を考慮して設計される。非線形関数 f は、入出力長が一致した全単射関数を想定する。例えば、AES のラウンド関数のようにブロック暗号で用いられている S-Box などの、代数次数、非線形性などが最良となる置換関数とバイト単位での MDS 行列を組合せて構成する。 f にはそれぞれ拡大鍵が入力される。

2. 3 内部状態のサイズについて

鍵系列において、同様のパターンが繰り返し出現した場合、暗号文同士を排他的論理和することで鍵系列を削除することが可能となる。従って、1つの共通鍵を使用する場合に、同様のパターンが出現しないように、内部状態を大きく取る必要がある。本節では、内部状態のビット数についての設計指針を考察する。Maurerは、論文[1]で、安全性を確保するために必要な最小の内部状態のビット数について、以下のように定式化を行なっている。今、内部状態のビット数を M 、1つの共通鍵で暗号化する平文のビット数を N 、暗号文パターンが繰り返す確率を p とすると、 $M > \log_2(N^2/p)$ を満たす M を用いれば、安全性が確保できる。提案した構造において、内部状態を更新するフィルタ関数を入力 kw に対して出力 $kw/2$ である関数と仮定すると、最下段のレジスタ数 n は、 $mnw > \log_2(2^{256} * 2^{mw})$ となる。例えば、ワード長 w を 32bits、 $m=n/2$ とした場合、 $n>16$ で条件を満たす。従って、パラメータ設定によっては、方式(b)のほうが効率的となるか旺盛がある。ただし、以下の式では、出力する鍵系列を完全ランダムと仮定している。また、1つの秘密鍵を使用して暗号化する平文のビット数を、 2^{128} としている。

3. あとがき

本稿では、ソフトウェア実装において高速処理を達成する自己同期型ストリーム暗号の構成方法を検討した。今後は、実際にアルゴリズムを設計し、安全性、処理性能などの詳細評価を行なっていきたいと考えている。

参考文献

- [1] U. M. Maurer, "New Approaches to the Design of Self-Synchronizing Stream Ciphers," Proc. of Eurocrypt'91, LNCS, No.547, pp.458-471, 1991
- [2] J. Daemen, and P. Kitsos, "The self-synchronizing stream cipher MOSQUITO", eSTREAM Doc., 2005, available at <http://www.ecrypt.eu.org/stream/mosquito.html>
- [3] A. Joux, and F. Muller, "Chosen Ciphertext Attacks Against MOSQUITO," Proc. of FSE2006, LNCS, to appear, 2006.