研究報告

誤り訂正符号の構成法に関する研究

学習院コンピュータシステム支援組織　助教　山　口　健　二

１．はじめに

現代の情報化社会では、CD や DVD、HDD、USB メモリなどのデジタル記憶装置が幅広く利用されている。これらのデジタル記憶装置では、文字や画像のデータを 0 と 1 の数値列に変換して保存している。このように、文字や画像などの情報源を、数値などの符号に変換することを符号化と呼ぶ。

この 0 と 1 の数値列の一部は、伝送の際やデジタル記憶装置の経年劣化により、数値が 0 から 1 に変化したり、0 でも 1 でもない数値に変化したりすることがある。そうなると、文字や画像データが元とは違う情報になってしまう。このため、文字や画像のデータを数値列にする際に、数値列に冗長性を設け、数値列の一部が誤っていた場合でも可能な限り訂正を行えるようにしている。このような誤り訂正能力を持つ符号を誤り訂正符号と呼ぶ。世界中で利用されているインターネット上でのデータ伝送や、長距離のため符号化された情報が変化しやすい衛星通信においては、誤り訂正符号の技術が必要不可欠である。

また、誤り訂正符号は暗号技術とも深いつながりがある。例えば、共通鍵暗号に分類されるストリーム暗号は平文系列（暗号化したい情報を数値列にしたもの）と鍵系列（乱数列）を排他的論理和することで暗号文系列を生成する。この鍵系列を生成する際に、コンバイナ型乱数生成器を使うことがある。コンバイナ型乱数生成器は、複数の LFSR（線形フィードバックレジスタ）の出力を非線形関数の入力とし、非線形関数の出力を鍵系列として用いる。このストリーム暗号を攻撃する際に、出力された鍵系列と各 LFSR の構造から、各 LFSR の初期状態を推定するために、誤り訂正符号の技術を利用することがある。

本研究では、誤り訂正符号の構成法とその応用に関する調査を行った。

２．LDPC 符号について

ここでは、代表的な符号の一つとして、LDPC 符号を説明する［1］。

LDPC（Low Density Parity check Codes）符号は、Robert G. Gallager が 1960 年代初期に発案した誤り訂正符号である。C を LDPC 符号の符号語、IF を有限体、H を LDPC 符号の検査行列とすると、以下のようになる。

$$C = \{c \in \mathbb{F}_q^n : cH^T = 0\}$$

LDPC 符号は線形符号のひとつであり、Shannon 限界に近い性能がある。また計算機への並列実

装に適している。LDPC 符号にはレギュラー LDPC 符号とイレギュラー LDPC 符号がある。レギュラー LDPC 符号は M 行 N 列の 2 元検査行列 H の各列のハミング重み（1 の個数）が同じであり、各行のハミング重み（1 の個数）が同じであり、かつ各列のハミング重みが列数 N より非常に小さいときに検査行列 H により定義される LDPC 符号であり、H は非常に疎な行列（要素として 0 が非常に多い）である。また、イレギュラー LDPC 符号は M 行 N 列の 2 元検査行列 H の各列のハミング重み（1 の個数）が「すべて同じ」ではなく、各行のハミング重み（1 の個数）も「すべて同じ」ではなく、かつ各列のハミング重みが列数 N より非常に小さいときに検査行列 H により定義される LDPC 符号である。復号法は様々なものがあるが sum-product 復号法およびそれを応用したものによって行われることが多い。

３．誤り訂正符号を用いた暗号解読について

ここでは、誤り訂正符号の復号法の一つである確率伝播法（Belief propagation, BP）による、ストリーム暗号のコンバイナ型生成器に対する攻撃法について具体例をあげて説明する［2, 3］。

ストリーム暗号は、共通鍵暗号の一つで、平文系列と鍵系列を排他的論理和して暗号文系列を得る。よって、既知平文攻撃（既知暗号文攻撃）や選択平文攻撃（選択暗号文攻撃）の場合、ストリーム暗号への攻撃は、鍵系列への攻撃に帰着される。コンバイナ型生成器を使ったストリーム暗号の場合、鍵系列の元となる種を複数の LFSR の入力とし、複数の LFSR の出力を非線形関数の入力とし、非線形関数の出力を鍵系列とする。今回の攻撃は、BP を利用して、非線形関数の出力、即ち鍵系列から LFSR の出力を推測するものである。そして、LFSR の出力から LFSR の初期状態を求めることで、鍵系列の元となる種を解読することができ、攻撃が成功する。

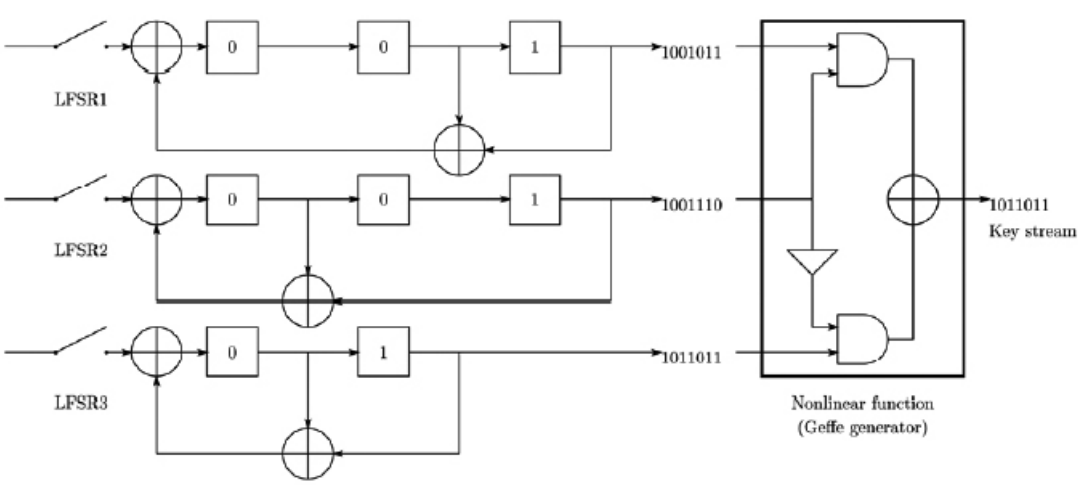

図 1　コンバイナ型生成器

例として、図 1 のようなコンバイナ型生成器を考える。これによって出力された鍵系列に対して、攻撃を行うこととする。

前提条件として、敵は既知平文攻撃によって、1011011 という鍵系列を得たとする。

また、非線形関数がジェフ型であることも既知であるとする。

さらに、任意の LFSR の結線構造も既知であるとする（レジスタの値は未知）。

そして、解読成功条件は、複数の LFSR のうち、どれかひとつの出力系列が判明すれば、攻撃成功とする（ある程度の長さ以上の出力系列が分かれば、Berlekamp-Massey algorithm から LFSR の初期状態が分かるため）。

まず、今回は LFSR1 の出力に着目する。

今回、非線形関数がジェフ型生成器なので、LFSR1 と Key stream の出力が確率 0.75 で一致する［4］。よって、この非線形関数を誤り率 p=0.25 の二元対称通信路（BSC）と考えることができる。

LFSR1 について、時刻 i での内部状態 S_i や、LFSR1 の内部状態の遷移行列 A について考えると、以下のようになる。

$$S_i = \begin{pmatrix} x_{i+3} \\ x_{i+2} \\ x_{i+1} \end{pmatrix}$$

$$S_i = AS_{i-1}$$

$$S_i = A^i S_0$$

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad A^2 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad A^3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad A^4 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

そして生成行列 G は、以下のようになる。$\left(\overleftarrow{A_j^i}\right)^T$ は A^i の j 行だけ抜き出し、成分の順番を逆転させた行列の転置行列をあらわす。

$$G = \begin{pmatrix} 100 \left(\overleftarrow{A_1^1}\right)^T \left(\overleftarrow{A_1^2}\right)^T \left(\overleftarrow{A_1^3}\right)^T \left(\overleftarrow{A_1^4}\right)^T \\ 010 \\ 001 \end{pmatrix} = \begin{pmatrix} 1001011 \\ 0101110 \\ 0010111 \end{pmatrix}$$

また、検査行列 H は、以下のようになる。

$$H = \begin{pmatrix} \overleftarrow{A_1^1} \\ \overleftarrow{A_1^2} 1000 \\ \overleftarrow{A_1^3} 0100 \\ \overleftarrow{A_1^4} 0010 \\ 0001 \end{pmatrix} = \begin{pmatrix} 1101000 \\ 0110100 \\ 1110010 \\ 1010001 \end{pmatrix}$$

実際、LFSR1 の出力を H の右側から掛けると、以下のように成立する。

$$H((1001011)^T) = \begin{pmatrix} 1101000 \\ 0110100 \\ 1110010 \\ 1010001 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = 0$$

そして、検査行列で、タナーグラフを作ると以下のようになる（図 2)。通常ならここから復号を開始してもよいが、枝が多く、計算量が多くなるため、パリティ検査式の定義に従って枝の数を減らすことにする。

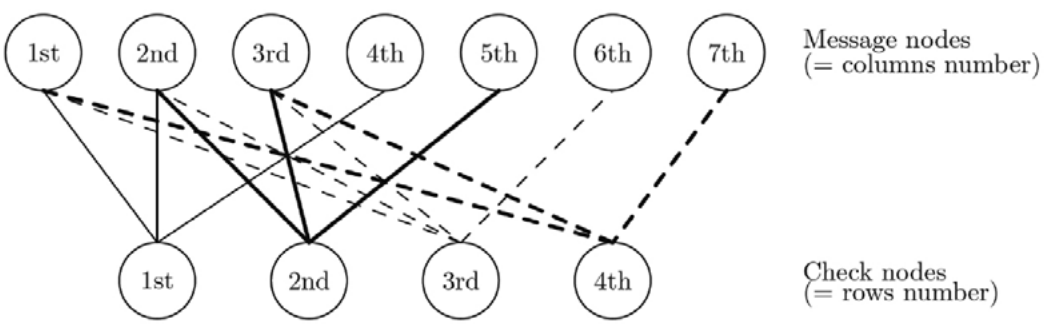

図２　タナーグラフ

[定義]
$n = L+1, L+2, \cdots, N$ と $1 \le w \le W$ と$B(< L)$ に対して、以下の作業を行って構成されたΩ_nを、ビットnに関係するパリティ検査式集合という。ただしW, Bは攻撃者の任意設定値である。
1. H の$n-L$ 行目と他の w 行の和を計算する。
2. これらの和の中から、ビット$i = B+1, B+2, \cdots, L$ がすべて 0 になるものをΩ_nとして記録する。

B=1 とすると、第 2 列と第 3 列が 0 になるように、検査行列 H の行の足し算を行い、第 2 列と第 3 列が 0 になったものを、パリティ検査式集合 Ωn の元とする。そして各元にインデックス付ける。

n=L+1=3+1=4 のときは、H の第 1 行と第 2 行と第 4 行を足すことで、第 2 列と第 3 列が 0 になる。また、第 1 行と第 3 行と第 4 行を足すことで、第 2 列と第 3 列が 0 になる。それ以外はない。よって、Ω_4 の元となるのは、「第 1 行と第 2 行と第 4 行の和（インデックス 1）と「第 1 行と第 3 行と第 4 行の和（インデックス 2)」である。よって、インデックスを用いて以下のように表記する（同じように、n=5, 6, 7 のときも同様に行う)。

$$\Omega_4 = \{1,2\} \quad (124, 134)$$

$$\Omega_5 = \{1,2\} \quad (23, 124)$$

$$\Omega_6 = \{1,2\} \quad (23, 134)$$

$$\Omega_7 = \{1,2\} \quad (124, 134)$$

また、Ω_n （n=4, 5, 6, 7）のパリティ検査式集合は以下のようになる。

$$\Omega_4 = \begin{Bmatrix}1\\2\end{Bmatrix}\begin{pmatrix}0001101\\1001011\end{pmatrix}$$

$$\Omega_5 = \begin{Bmatrix}1\\2\end{Bmatrix}\begin{pmatrix}1000110\\0001101\end{pmatrix}$$

$$\Omega_6 = \begin{Bmatrix}1\\2\end{Bmatrix}\begin{pmatrix}1000110\\1001011\end{pmatrix}$$

$$\Omega_7 = \begin{Bmatrix}1\\2\end{Bmatrix}\begin{pmatrix}0001101\\1001011\end{pmatrix}$$

ここから、復号処理に入る。今回、鍵系列が（$z_1, z_2, z_3, z_4, z_5, z_6, z_7$）=（1,0,1,1,0,1,1）であり、誤り率 p=0.25 を考慮すると、$f_n(0)$, $f_n(1)$ の初期値は次のようになる（表 1）。

表 1　f_n の初期値

n	1	2	3	4	5	6	7
$f_n(0)$	1/4	3/4	1/4	1/4	3/4	1/4	1/4
$f_n(1)$	3/4	1/4	3/4	3/4	1/4	3/4	3/4

（$x_1, x_2, x_3, x_4, x_5, x_6, x_7$）の最初の B ビット分は、全数探索（総当たり）で求める。今回は B=1 として復号処理を行う。まず 1 ビット目を 1 として仮定して進める。各変数は以下のように定義する。

m ($\in \Omega_n$)：ビット n に関係するパリティ検査式に付けたインデックス

q_{mn} (u)：m 以外のパリティ検査式によって得られた、ビット n の値が u となる確率

$\delta q_{mn} = q_{mn}(0) - q_{mn}(1)$

ω_n (m)：ビット n に関係するパリティ検査式 m

n' ($\in \omega_n(m)$)：ビット n に関係するパリティ検査式 m に含まれるビット

α：$q_{mn}(0) + q_{mn}(1) = 1$ となるようにする調整値

α'：$Q_n(0) + Q_n(1) = 1$ となるようにする調整値

まず、$q_{mn}(u)=f_n(u)$ として初期化する。

そして以下の復号処理を行う。

1．

$$\delta r_{mn} = \prod_{n' \in \omega_n(m)} \delta q_{mn'}$$

$$r_{mn}(u) = \frac{1}{2}(1 + (-1)^u \delta r_{mn})$$

2．

$$q_{mn}(u) = \alpha f_n(u) \prod_{m' \in \Omega_n \setminus m} r_{m'n}(u)$$

$$Q_n(u) = \alpha' f_n(u) \prod_{m \in \Omega_n} r_{mn}(u)$$

3．

$$Q_n(1) > 0.5 \Rightarrow \hat{x}_n = 1,\ \ Q_n(1) \leq 0.5 \Rightarrow \hat{x}_n = 0$$

として、現段階で推定したLFSR系列

$$\hat{X} = (\hat{x}_{L+1}, \hat{x}_{L+2}, \cdots, \hat{x}_N)$$

を生成する。$\hat{X}$ がすべてのパリティ検査式を満たせば、$\hat{X}$ を復号結果とする。そうでなければ、最初に戻る。

4．$\hat{X}$ から、$\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_L$ を生成し、これから得られる符号語 $\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_N$ で次の式を計算する。

$$S = \sum_{n=1}^{N} \tilde{x}_n \oplus z_n$$

5．T を閾値として、$S \leq T$ を満たすならば、$\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_L$ をLFSRの初期状態の推定値として出力する。

今回の例にこの復号処理を行うと、

$$Q_4(1) = \frac{49}{76} > 0.5,\ \ Q_5(1) = \frac{7}{52} < 0.5,\ \ Q_6(1) = \frac{7}{12} > 0.5,\ \ Q_7(1) = \frac{49}{76} > 0.5,$$

となり、推定系列として、（1 $\hat{x}_2$ $\hat{x}_3$ 1011）が生成された。これはすべてのパリティ検査式を満たす。そして、S=0+0+1+0+0+0+0=1 で $S \leq 1.75$ となったため、これを復号結果とする（誤り率が p = 0.25 より、0.25＊7 = 1.75 を閾値 T とした）。

5．まとめ

誤り訂正符号の中でもLDPC符号（低密度パリティ検査符号）は、1960年代にGallagerによって発表された符号であるが、計算量が膨大であったためあまり関心が集まらなかった。しかし、近

年になって MacKay らによって性能の良さを再発見された符号である。この符号の検査行列は非常に疎であり、sum-product 復号法などを用いて復号をおこなう。今回、誤り訂正符号を暗号解読に用いるなどの応用例について調査をおこなった。

参考文献

［1］ 萩原学、符号理論 －デジタルコミュニケーションにおける数学 －、日本評論社、2012.

［2］ 細渕智史、齋藤友彦、松嶋敏泰、“ストリーム暗号への攻撃法の改良に関する一考察 －多次元の相関を利用した攻撃 －”電子情報通信学会論文誌 A、Vol.J89-A、No.2、pp.121-128、2006.

［3］ M.J. Mihaljević, M.P.C. Fossorier, and H. Imai, “On decoding techniques for cryptanalysis of certain encryption algorithms,” IEICE Trans. Fundamentals, vol.E84-A, no.4, pp.919–930, 2001.

［4］（社）電子情報通信学会（編）、情報セキュリティハンドブック、オーム社、2004.