データハイディングにおける埋め込み・抽出の順序指定鍵

5 L − 1

小出 昭夫
日本アイ・ビー・エム株式会社 東京基礎研究所

1. はじめに

データハイディングとは画像やオーディオなどのデジタルコンテンツに人間の目や耳で検知できないメッセージを埋め込み検出する技術の総称である[1]。その実現方法の一つとして、整数の配列であるデジタルコンテンツをブロックに分割し、特定の順でブロックを指定し、各ブロックでわずかな数値の操作で1ビットづつを埋め込む、または、取り出す手法[2]がある。この手法では、埋め込まれた情報の機密性、及び、改ざん耐性はブロック指定の順序を生成する手法の機密性に依存する。

ブロック指定の順序を生成するということは、順序番号nが与えられたとき、ブロック識別番号f(n)を算出することである。すなわち、識別番号も順序番号も 非負整数の集合 Z[n]={ 0, 1, 2, ..., N-1 }の要素とすると、Z[n]からZ[n]への関数 f を作成することである。

本報告では、与えられた非負の整数（鍵 k）に基づき一様性の呈する一対一関数 f_k を生成し、かつ、異なる鍵は異なる一対一関数を生むようにする手法を提示する。すなわち、次の問題を解決するものである。

- 同じブロックが複数回指定されない。もし指定されると、先に埋め込んだビットが後のビット埋め込みで消されてしまう。
- 順序列 f(n) の長さを短く制限しても画像の狭い範囲に限定されることがない。もし限定されると、画像の切り取りで埋め込み領域を避けることができる。
- 異なる鍵が同じ順序列 f を生むことがない。もし異なる鍵が同じ順序を指定すると、偶然、別の鍵で埋め込み情報が読めるかもしれない。
- 識別番号集合の数Nが小さくても異なる鍵が充分多数ある。さもないと、辞書式攻撃で簡単に鍵が破られてしまう。

2. 一様性を呈する一対一関数の生成

Nが2のべき乗であるとき、一様性を呈する一対一関数は次のようにして与えることができる。$N=2^M$とし、識別番号f(n)をMビットの語

$$b_0(n)\, b_1(n)\, b_2(n) \ldots b_{M-1}(n) \qquad (1)$$

としたとき、すなわち、

$$f(n) = \Sigma_j\, b_j(n)\, 2^{M-1-j} \qquad (2)$$

としたとき、各 b_m が 2^{m+1} を周期とする周期関数でビット $b_m(n+2^m)$ が ビット$b_m(n)$ の反転であるとする。すなわち、

$$b_m(n+2^{m+1}) \equiv b_m(n) + 1 \pmod 2 \qquad (3)$$

このとき、関数 f は $Z[2^M]$から$Z[2^M]$への一対一関数で、次の意味で一様性を示す。長さ2^mの順序番号の連続部分列、{ f(n), f(n+1), ..., $f(n+2^m-1)$ }は、識別番号の範囲を2^m等分した各領域に1個づつはいるようにマップされる。

上記の一対一関数を本報告では ビット倍周期関数と呼ぶことにする。

画像に適用する場合、二次元なので、Mビットの語を次ぎのように分解し、

$$b_0(n)\, b_2(n)\, b_4(n) \ldots b_{M-2}(n) \qquad (4)$$

$$b_1(n)\, b_3(n)\, b_5(n) \ldots b_{M-1}(n) \qquad (5)$$

し、それぞれを、f(n)の座標成分x(n)とy(n)とすれば良い。

Data Hiding Keys for Ordering Blocks at Embedding and Detection,

Akio Koide, Tokyo Research Laboratory, IBM Japan

このビット倍周期関数は次のいずれの方法でも生成でき、2の(2^{M-1})乗個の独立な鍵を使用できる。

- 独立ビット充填による生成手法
- 多項式とビット列逆転による生成手法
- 漸化式とビット列逆転による生成手法

3．独立ビット充填による生成手法

独立ビット充填による生成手法とは前述のビット倍周期関数を鍵(非負整数)から下記のようにして生成する手法である。

鍵の各ビットの値を、ビット倍周期関数b_mの独立なビットに代入し、鍵のビット長が独立なビット数より小さい場合には、鍵k_0を初期値とし、次々と非負整数k_{n-1}からk_nを一方向関数などによって生成し、そのビット値を残りの独立なビットに代入することによってZ[2^M]からZ[2^M]への一対一関数を作成する。各関数b_mの独立な値は$b_m(0),b_m(1)...b_m(2^m-1)$なので、独立なビットの数は、$1+2+\ldots+2^{M-1}=2^M-1$となり、すなわち、$2^M-1$ビット長の鍵が可能となる。

4．多項式とビット列逆転による生成手法

この手法は、鍵より非負整数a_jを求め、多項式

$$g(n)\equiv \Sigma_j\ a_j\ [(n+j)!/n!\ ]/\ 2^{mj}\pmod{2^M} \qquad (6)$$

またはそれと同値な式によって求まる非負整数g(n)をビットで表したときのビット列を逆転することによって非負整数f(n)を算出し、Z[2M]からZ[2M]への一対一関数を作成する手法である。ここで、和はj=0から2^M-1までとるものとし、(n+j)!/n!は多項式 n (n+1) ... (n+j) のことである。また、m_jは次で与えられる整数とする。

$$m_0=m_1=m_2=m_3=m_4=m_5=0 \qquad (7)$$

で、j>5については

$$m_j=\phi(j)-\psi(j)-1 \qquad (8)$$

とする。ここで、$\phi(j)$はj!を素因数分解したときの2のべき乗の指数で、$\psi(j)$はjを越えない最大の2のべき乗の指数とする。また、各a_jはM-$\psi(j)$個のビットが独立な非負整数で、a_0は任意のMビットの非整数、a_1は非負の奇数、a_2、a_3は$a_2<2^{M-1}$、$a_3<2^{M-1}$を満たす偶数で、j>3については$a_j<2^{M-\psi(j)}$なる非負整数である。係数a_jのM-$\psi(j)$個のビットが独立なことより、2^{M-1}個のビットの値が自由に選択できる。

5．漸化式とビット列逆転による生成手法

この手法は、鍵より非負整数a_jと初期値x_0を求め、（6）の多項式によって定まる関数gを用いて、初期値x_0より、漸化式

$$x_n=g(x_{n-1}) \qquad (9)$$

で次々と非負整数x_nを算出し、それをビットで表したときのビット列を逆転することによって非負整数f(n)を算出することによってZ[2^M]からZ[2^M]への一対一関数を作成する手法である。初期値x_0は任意のMビットの非負整数、a_jはM-$\psi(j+1)$個のビットが独立な非負整数で、（a_0、a_1は非負の奇数）、2^M-1個のビットの値が自由に選択できる。

6．終わりに

ブロックの指定順序で埋め込まれた情報の機密性、及び、改ざん耐性を守るデータハイディングの手法に関して、非負整数の鍵から一様性を呈する順序列を生成する方法として、独立ビット充填による生成手法、多項式とビット列逆転による生成手法、漸化式とビット列逆転による生成手法を記述した。いずれの手法でも、2章で述べたビット倍周期関数を生成し、2^M-1のビット長の鍵が可能となる。

謝辞

本研究の一部は創造的ソフトウェア育成事業の一環として行われた。

参考文献

(1) W. Bender, D. Gruhl, N. Morimoto, and A. Lu: "Techniques for data hiding," IBM Systems Journal, Vol. 35, pp. 313-336 (1996).

(2) 森本典繁、清水周一、小出昭夫:「データ・ハイディングの開発」、創造的ソフトウェア育成事業およびエレクトロニック・コマース推進事業中間成果発表会論文集、創造的ソフトウェア育成事業編、pp. 467-474、情報処理振興事業協会、(1997).