秘密分散法を用いたソフトウェア電子透かしの実験的検討

2004MT039 加藤 紘基　　2004MT072 野田 英則
指導教員　真野 芳久

1　はじめに

近年の著作権侵害の現状から、いかにソフトウェアの著作権を保護するかが課題となっている。これを目的としたソフトウェアプロテクション技術に、「電子透かし」と呼ばれる技術がある。また、電子透かし技術と技術的には同様なものとして、「ステガノグラフィ」技術がある。

「電子透かし」は、価値のあるものに情報を埋め込むことを目的として使われる。著作者があらかじめソフトウェアに著作権情報を埋め込んでおき、盗用が発覚した場合に、埋め込んだ情報を取り出すことで著作権を主張できる。

「ステガノグラフィ」は価値のある情報を埋め込む目的で秘密通信として使用される。

次に、情報漏洩を防ぐセキュリティ技術に「秘密分散法」がある [1]。これは、秘密情報を個々には意味のない複数の情報に分散化することによって、元情報の安全性を確保する技術である。分散後の情報 (以下、これをシェアと呼ぶ) からは元情報を推測できないという利点を持ち、シェアが規定個数以上集まれば、元情報を復元することができる。

本研究では、ソフトウェアの秘密情報を保護する方法として、秘密分散法をソフトウェア電子透かしに適用した。そして、秘密分散法のひとつである「視覚型秘密分散法」[2] を用いて、プログラム中の分散情報から bit 列を取り出し、視覚的に透かしを取り出す方法を提案する。分散情報 n 個の中から k 個で復元することを (k,n) 分散法と言い、私達は 2 つの (2,3) 分散法を提案する。この手法による応用の可能性としては、電子透かしやステガノグラフィが考えられるが、詳しい応用方法については言及しない。

2　秘密分散法と視覚型秘密分散法

ここで述べる秘密分散法は「(k,n) しきい値分散法」を指す。定義は、1) 分散情報 n 個のうち、任意の k 個以上で復元可能である、2) 分散情報 $k-1$ 個以下では何の情報も得られない、である (**図 1**)。Adi Shamir[1] は、多項式による実現を示している。

秘密分散法のひとつである視覚型秘密分散法は、計算機を使わず人間の視認によって秘密情報を復号する。滝澤ら [2] は、テキストに応用した手法を提案している。

私達は、視覚型秘密分散法を用いてプログラム中の分散情報から bit 列を取り出し、視覚的に透かしを取り出す方法を提案する。

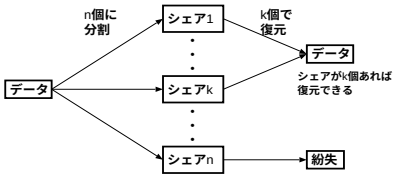

図 1　(k,n) しきい値分散法

3　プログラムに対する視覚型秘密分散法

プログラムに対する視覚型秘密分散法を述べる (**図 2**)。これは、透かし入りプログラムの中のモジュールから bit 列を取り出し、それらを一定の法則に従って改行してできる分散画像を重ね合わせることによって秘密情報 (例えば文字の形状) が浮かびあがるというものである。プログラム (P) と透かし (w) を用意し、エンベッダ (E) による変換を行ない透かし入りプログラム (P_w) を作成する。透かし入りプログラム中のモジュールからレコグナイザ (R) による透かしの認識を行い bit 列を取り出す。bit 列は一定の法則に従い改行し、bit「1」と bit「0」を羅列した画像を作成する。複数の分散画像のうち、規定個数の分散画像を重ね合わせると秘密情報が現われる。bit の取り出しには XOR(排他的論理和) を用いる。

秘密情報復元の際に持たせたい性質は、分散画像からは何の情報も得られないが、復元した画像を目で見たときにその画像が明らかに故意であると確認できることである。これを目的として具体的手法を検討する。観点としては、秘密画像を復元したところ、ノイズ無しの完全な秘密画像を復元できれば故意であると判断できる。また、視覚型秘密分散法の利点である視覚の認識力を利用し、ノイズは含むが偶然性は低いと思われる秘密情報を視認できれば故意であると判断できる。以上の観点から、私達は 2 種類の (2,3) 分散法を提案する。

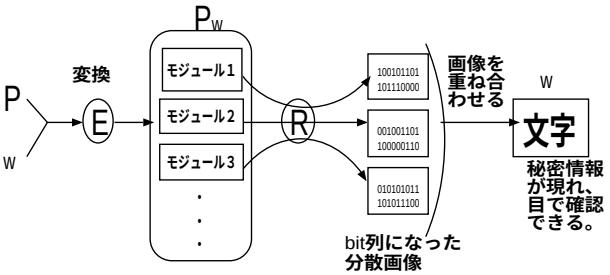

図 2　視覚型秘密分散法での透かしの取り出し方

4 分散画像の 2 bit から秘密画像の 1 bit を復号する (2,3) 分散法

4.1 秘密画像の復元

分散画像の 2 bit から秘密画像の 1 bit を復号する (2,3) 分散法を述べる (**図 3**)。bit 列で表された分散画像が 3 枚あり、このうち任意の 2 枚を用いて秘密画像を復元する。まず、分散画像を 2 枚選び、それらの XOR を取り合成画像を作成する。次に、合成画像の bit 列を変換規則に従って 2 bit ずつ区切って変換していくと秘密画像が現われる。bit は「0」と「1」の 2 値を用いる。

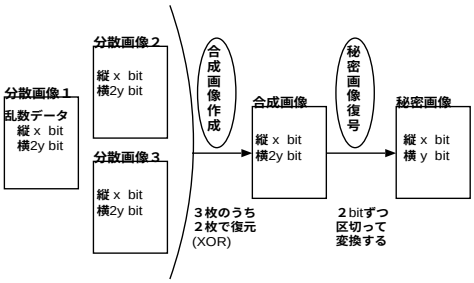

図 3　秘密画像復元方法

4.2 変換規則の構成について

秘密画像復号の際に、合成画像の 2 bit を秘密画像の 1 bit に変換する。また、分散画像作成の際には、秘密画像の 1 bit を分散画像の 2 bit に変換する。このように相互に bit が置き換わる規則 (変換規則) を次のように定める。

変換規則

1. 合成画像の 2 bit が「00」のとき、秘密画像の 1 bit を「0」とする。また、合成画像の 2 bit が「00」のとき、3 枚の分散画像の 2 bit はすべて同じ値とする。
2. 合成画像の 2bit が「01」「10」「11」のとき、秘密画像の 1bit を「1」とする。また、合成画像の 2bit が「01」「10」「11」のとき、3 枚の分散画像の 2bit は、画像順不同で (00,01,10), (00,01,11), (00,10,11), (01,10,11) のいずれかとする。

これを図式化すると**図 4** になる。

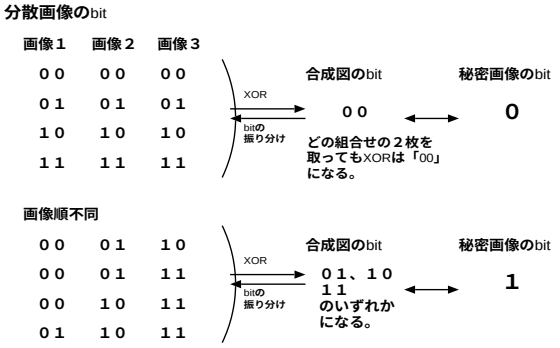

図 4　2 bit ずつ区切って変換するときの規則

図 5　3 枚の分散画像

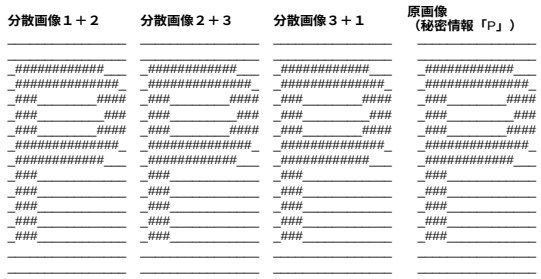

図 6　記号化した規則変換後の合成画像と原画像

4.3 実験

実験は、分散画像を縦 16 bit、横 32 bit にして行なった。分散画像は 3 枚あり、分散画像 1 は乱数データである。分散画像 2 と分散画像 3 は、分散画像 1 と秘密画像を基にして作成した画像である。**図 5** に分散画像 3 枚を示す。bit「0」を「_」で表わし、bit「1」を「#」で表わした。分散画像 3 枚のうち 2 枚を用いて XOR を取った後、合成画像から変換規則に従い変換したものが**図 6** である。分散画像 1 と分散画像 2 を用いて秘密画像を復元する様子を述べる。まず、分散画像 1 と分散画像 2 の XOR を取り、合成画像 (分散画像 1 ＋分散画像 2) を作る。次に、合成画像の bit 列を 2 bit ずつ区切っていき、変換規則に従って合成画像の 2 bit を秘密画像の 1 bit にしていく。3 行目は「00/10/10/11/01/01/11/01/10/01/11/01/10/00/00/00」であり、変換規則に従うと、「0111111111111000」となる。全ての行を変換すると秘密画像ができあがる。

4.4 評価

実験から、秘密分散法の特徴である分散画像を規定個集めることによって秘密画像を復元することができた。また、当初の目的である分散情報が規定個未満では秘密画像に関する情報を一切得ることはできないという特徴も満たすことができた。分散画像から秘密画像を復元すると、ノイズなしで秘密情報を完全に復元できることも達成できた。利点は、分散画像 1 は乱数データなので、実質他の 2 枚の分散画像を作るだけで良く、分散画像を 3 枚作る場合に比べて時間的に短縮できる。欠点は、秘密情報 1 bit を取り出すのに 2 bit 要るので、秘密画像から分散画像を生成する際に、データサイズが 2 倍になってしまうことである。また、実験の原画像のように明確な背景がある場合には、秘密画像復号の手順を踏まなくても、合成画像作成の時点で秘密情報を推測できることも起こりうる。

5 分散画像の 1 bit から秘密画像の 1 bit を復号する (2,3) 分散法

4 章の手法は、秘密画像を完全に復号できる利点がある一方で分散画像から 2 bit 取り出すために、情報量が多くなってしまう欠点があった。5 章では分散画像 1 bit から秘密情報の 1 bit を復号することで、情報量の減少を実現し、完全に復号するのではなく、視覚型秘密分散法の特徴である視覚機能を利用することで、多少のノイズが入っても秘密情報だと判別できる (2,3) 分散法を述べる。

5.1 分散画像の作成概要と規則定義

乱数 bit 列をランダムに分散画像 1、2、3 に分配する。このとき分散画像は互いに重複しないところに分配されることとする。分散画像の中で値が分配されずに空白の場所となっているところには、ある規則に従って値を入れていくことで、分散画像 1、2、3 を生成する (**図 7**)。今回は縦 16 個、横 16 個の 256(16×16) 個で考える。

ある規則について述べる。(2,3) 分散法において bit 振り分け規則を考えると、合成画像に入るノイズの個数は最小で 2 つある。ノイズ個数が 2 個で合成画像を作成するときの bit の振り分け方は 9 通りある。具体的例として、分散画像に割り振られた値が「0」、秘密情報が「0」の場合、割り振られなかった分散画像 2 つ共に「0」とする。また、秘密情報が「1」の場合、割り振られなかった分散画像 2 つ共に「1」とする。分散画像に割り振られた値が「1」、秘密情報が「0」の場合、割り振られなかった分散画像 2 つ共に「1」とする。また、秘密情報が「1」の場合、割り振られなかった分散画像 2 つ共に「0」とする。以上のように条件を定義することで、分散画像ひとつあたり 256 個のビットが入っている分散画像を生成することができる。**表 1** にこの場合の規則をまとめる。

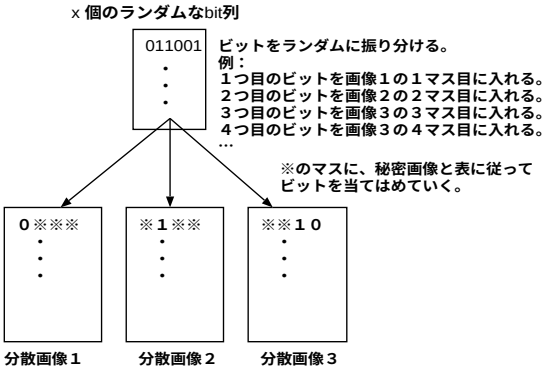

図 7　分散画像作成方法

表 1　bit 振り分け規則 (9 通りのうちの 1 つ)

秘密情報の bit	割り振られた bit	その他の bit1 つ目	その他の bit2 つ目
0	0	0	0
1	0	1	1
0	1	1	1
1	1	0	0

5.2 実験結果

bit を振り分けるプログラムを作成し実験を行なったところ、**図 8** の結果が得られた。ノイズが全体で 2 つしか入らない考え方は 9 通りあるが、9 通り全部の合成画像から秘密情報を読み取ることができた。今回の例では、秘密情報の「1」の部分を分散画像ごとに値を変えて実験をした。逆に、秘密情報の「1」を固定して、秘密情報の「0」を変化させた場合、ノイズが 2 つの場合でも、合成画像から秘密画像を読み取ることができなかったため、秘密情報の「0」を固定した。

合成画像と秘密情報の全体の一致率はノイズが全体で 2 つしか入らない 9 通りのすべてが 90% と高かった。合成画像ひとつひとつを見ても、89% から 92% と約 90% 一致している。一方、合成画像と秘密情報の文字の一致率は、全体で 67%、合成画像をひとつひとつを見てみると 40% 近い破損率でも、秘密情報を認識することができた。

図 8　復元した秘密画像

5.3 評価

実験から秘密分散法の特徴である分散画像を規定個集めることによって、秘密画像を復元することができた。また、規定個未満のとき視覚的にも秘密情報を推測できないという特徴も満たすことができた。

欠点として、(k,n) 分散法に拡張する場合は、単純な拡張では難しいので新たな手法が必要になる。

利点は、合成画像に多少のノイズが入っても、人の視覚機能を利用することで、4 章の手法と比べて復元過程においても生成する過程においても情報量を少なくすることができる点である。また、3 つの分散画像を XOR で合成した場合は、原画像と同じ、ノイズなしの秘密画像が完全に復元できる点である。

6 プログラムへの透かし挿入

ここでは名前への透かし挿入について述べ、ワードスペーシング法については省略する。

名前への透かし挿入では、あらかじめ文字に対応する bit を定めておく。分散画像の bit 列を定めた規則に従い文字情報に変換する。文字情報を順序は変えずにプログラム中の名前に挿入する。長さは自由で、文字情報の順が変わらなければよい。実験では、アルファベット 26 文字 (小文字) ＋ 6 文字 (大文字) の 32 文字を使うことにする。分散画像は、縦 15 bit、横 15 bit の画像を用意した (**図 9**(上))。bit から文字への変換では、1 文字で 5 bit 使うことにする。**図 9**(下) は、bit 列対応規則 (a:00000 b:00001 c:00010 ・・・ z:11001 A:11010 B:11011 C:11100 ・・・ F:11111) に従い、bit 列を文字情報に変換した図である。分散画像の 1 行がアルファベット 3 字に置き換わる。分散画像 1 の文字情報をサンプルプログラムに埋め込んだものが**図 10** である。挿入は、元プログラムの変数を文字情報に置き換える。文字情報の順序は変えないで、長さは自由で挿入する。透かしの取り出し方は、「プログラムを始めから見ていき、初めて出てきた変数を取り出して bit に変換」、「以下で出てきた同じ変数は無視」とする。

この方法の利点は、1 文字に対する bit の長さを自由に設定できる点である。実験では、1 文字を 5 bit で表わした。また、元のプログラムから変数だけを変えていて、冗長的記述が増えているわけではないので、プログラムの動作時間は変わらない。さらに、復元の際にはノイズが入らないので、ノイズ無しの完全な秘密画像を復元することができる。

問題点は、このような変数名では不自然なので、攻撃者が変数名を変えてしまう可能性がある。変数名が変われば秘密画像は復元できなくなる。耐性の面から言えば、この手法は耐性が弱いが、ワードスペーシング法と比べると、字句レベルになっているのでワードスペーシング法よりは耐性はある。しかしまだまだ改善しなければならない。ひとつの改善点としては、変数は重複して取り出さないので重複しても bit を取り出せるようにしたい。

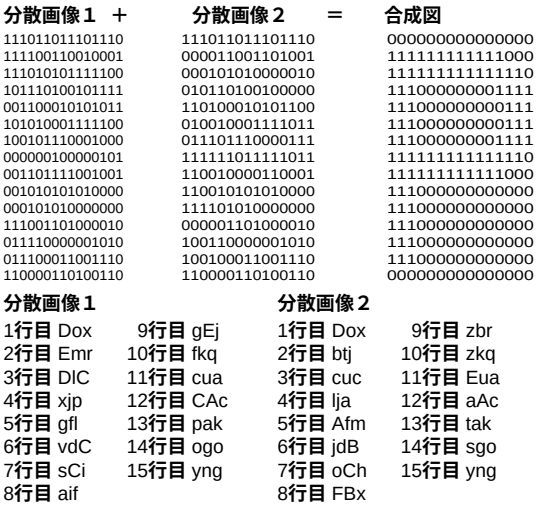

図 9　bit 列分散画像と秘密情報 (上)、bit 列対応から文字に変換した分散画像 (下)

```
public Clock2(){
    setTitle("時計");
    setSize(400,450);
    setLocation(250,50);
    DoxEmrDlCxjpgfl=getContentPane(); /*透かしの一部*/
    CPanel cp=new CPanel();
    DoxEmrDlCxjpgfl.add(cp,"Center") ;
    clockStart();
    setDefaultCloseOperation(JFrame .EXIT_ON_CLOSE);
}
```

図 10　分散画像 1 の情報を埋め込んだ (一部抜粋)

7 おわりに

提案した 2 つの (2,3) 分散法では、実験の結果、秘密情報を視覚により確認することができた。プログラムへの透かし挿入では、耐性については深く検討する必要がある。これからの課題としては、(k,n) 分散法への拡張、取り出した bit 列を XOR を用いてひとつの画像にするプログラムの作成、電子透かし・ステガノグラフィの用途に合わせた視覚型秘密分散法の検討、透かし入りプログラムの透かしを破壊して秘密情報を復元できるかの検討、復元できなければ、どの程度の攻撃までなら耐えられるかの検討などがある。

参考文献

[1] Adi Shamir:“How to share a secret”, C.ACM, Vol.22, No.11, pp.612-613 (Nov.1979).

[2] 滝澤修, 山村明弘:“自然言語テキストを用いた秘密分散法”, 情報処理学会論文誌, Vol.45, No.1, pp.320-323 (Jan.2004).

[3] 小野束：“電子透かしとコンテンツ保護”, オーム社 2001.