Vol.2010-CSEC-51 No.1
2010/12/10

表関数を用いた電子透かしの検出器ソフトウェアの難読化

大関和夫[†]　　魏　遠玉[††]

難読化で表関数(ROM 方式)によるものは，状態数が増加するが，解読が不可能に近いと考えられる．ROM 方式では，演算に使用する関数を予め計算し，固定の配列に記録し，参照することにより，どのような関数を用いたかのアルゴリズムを隠蔽する．Barak の論旨と何点かで異なる方式を提案している．この難読化方式を電子透かしの埋込み・検出ソフトに適用し，実験を行った．

Obfuscation of Software for Watermark Detector Using Table Function

Kazuo Ohzeki[†]　and Engyoku Gi(YuanYu Wei)[††]

Obfuscation with a table function (ROM function) can be difficult to understand. The number of states will increase as the size of table increases. Obfuscation with the table function hides all algorithms which show what kind of functions are used. It calculates all results in advance and save them in a memory or software consatnt arrays. The obfuscation differs in several points from the method Barak et al proved.. We build the proposed obfuscation method in a watermarking embedding and detecting software. Experimental results are shown.

1. はじめに

ソフトウェアの難読化は，Collberg らの３方式[1]と多数の実施例や，Barak[2-3]らの不可能性を証明する考えなど多岐にわたっている．Collberg の多数の実施例は，解読を難しくすることはできるが，解読を課題として取り組めば，比較的少ない手順で解読可能なものである．一方，Barak らの難読化不可能論により，不可能が前提となってきた．しかし，Barak らの証明は難解で把握が難しいが，その理論は，連続関数に限定された場合のものと見做され，離散的な現実の実関数は難読化可能なものがあると考えられる．ここでは，このような分析を進め，難読化技術の詳細化を行い，表関数による究極的な難読化を構成することを試みる．

Barak らの不可能論により，消極的な研究が進められてきたが，今回，Barak の論旨と何点かで異なる方式を提案する．提案する方式は，予め全ての計算結果を求め，ROM（Read Only Memory : 読み出し専用メモリー）乃至は，ソフトウェアプログラムの固定配列に格納しておくもので，入力と，出力の関係以外は，何も示されていない．これは神託機械（Oracle 機械）と同じ形状となるが，Barak の説明のような Virtual な Black-Box ではなく，実在する White-Box になっている．

このような考えに類似の研究は，Chow らの White-Box 難読化[4]にも見られる．Chow 等の方式は，DES 暗号化処理を難読化するもので，処理中のアフィン変換を表形式に変形している．一方，課題として，表にする処理は部分的で，多くの計算部が残っているため，計算量が多く，遅いことが述べられており，問題となっている．筆者らの提案方式は，メモリサイズは増加するが，処理速度は表の値を参照するステップが基本で，配列構造を多段にした場合は，その段数分だけのステップ数になるが，いずれにしても処理速度は，ほとんど０に近く，Chow 等の方式と異なると考えられ，実用性が高いといえる [5-6]．筆者らは，検出器公開型の電子透かしにおいて，公開する検出器を難読化する方式について提案してきた．難読化には ROM 関数によるもの[5]は，状態数が増加するが，解読が不可能に近いと考えられる．

2. 従来難読化方式と問題点

難読化を筆者らの観点で分類すると，まず対象としてデータとプログラムに 2 分さ

[†] 芝浦工業大学工学部情報工学科
ISE, College of Eng., Shibaura-Institute of Technology
[††] 芝浦工業大学 大学院 機能制御システム専攻
Functional control systems, Graduate School of Eng., Shibaura-Institute of Technology

Vol.2010-CSEC-51 No.1
2010/12/10

れる．データとは，暗号鍵，復号鍵，パスワード，個人情報，秘密定数などで，プログラムとは，計算機で実行するソフトウェアプログラムのことである．次にプログラムの難読化は全てのプログラムに適用できる汎用型と，特別な機能を有すプログラムに固有の難読化に分けられる．汎用型にはあらゆる関数類が含まれるが，その中には，ごく単純な関数，例えば

$$f(x) = 3x - 2 \quad (1)$$

も含まれる．この関数プログラムを難読化した結果に何個かの入力を入れ，出力を下の表１のように得れば，これが x の一次関数で，係数が３，定数部が-2 であることは，容易に判明する．従って，もとの関数乃至は演算アルゴリズムがあるレベル以下の簡易なものであるとき，どのような難読化を行っても，入出力の関係の観察により，アルゴリズムを推定することは容易にできる．従って，このように容易な計算に難読化を施すのは効果がなく，また，可能性または不可能性を調べても実用性が無い．

一方，筆者らは，電子透かしの検出器を公開する際に難読化する応用をあげているが，この検出器は変形フーリエ変換を含んでいるため，未知数に関する多次元方程式を解く必要があると考えられる．また，この変形フーリエ変換は，非線形に拡張でき，5 次方程式を組み込めば，一般解法が存在しないものが組み込み可能となる．本論文では，後半で，その構成を述べる予定である．

表１　難読化後の解析試行例

試行	a	b	c	d	e
入力	0	1	2	3	4
出力	-2	1	4	7	10

Barak の難読化不能な関数の例には，上記(1)式のようなものがあるが，これは，上で述べたように，元来難読化前からアルゴリズムが無い単純な関係でしかない．それを母数を無限の所に一般化と称して持って行って，確率的に解析を不能にしただけで，証明として正しくても，難読化という意味的な例としてふさわしくない．つまり，連続領域では，母数が無限で，いくら試行して，x=α というものにたどり着く確率は，限りなく 0 ではあるが，実用領域での離散変数領域では出力がβになるまで探せば良いので，探索問題としては，有限になり簡易なものである．一次関数の係数を探す問題を基準とすれば，それ以下の 0 次の定数探索といえる．もし，全てのプログラムが，難読化不可能なものと，難読化可能なものに分かれているなら，不可能なものを取り上げて，不可能であるという例示は可能だが，可能なものについての検討はなされていないと考えられる．

もう一点異なる所は，難読化後のプログラムに対し，依然として入力から計算して出力を得るという表現をとっているが，筆者らの ROM 関数では，計算ではなく表のデータを検索するだけで，アルゴリズムが観測される元になるような計算は行っていない．また，Barak の例では，連続関数についての証明であり，従って入力の変数も連続で，無限個ある．しかし，実際の例では，離散的な関数になり，入力の種類は，有限個の例も多い．

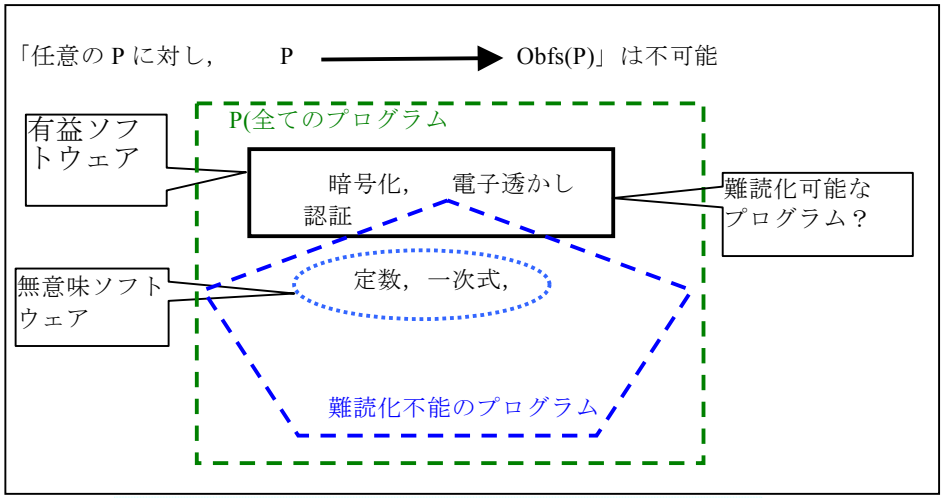

図１　難読化対象の分類．

Chow らの White-Box 難読化は，Black-Box での処理は，各個別の PC 等での演算は秘密にしても，公開されてしまう場合も有るため，初めから公開領域で行うことを前提に機密性を持たせることを提案している．これは Black-Box より，難読化変換への課題が増加した分だけ，難易度が高くなったと考えられる．DES を用いた暗号化演算処理を機密化することを目指している．DES 演算のうち，固定の鍵を使った演算部を表にして，鍵のデータを直接表示しないようにしている．演算はアフィン変換という 1 次変換である．小さいサイズのモジュールに分割している．問題点として，演算速度が遅いことと，プログラムサイズが大きくなることをあげている．

3. 表関数を用いた難読化方式

3.1 関数の複雑さ

筆者らは電子透かしの検出器の難読化の手法を多数検討しているが，その中にＲＯＭ 方式とその変形がある．ＲＯＭ 方式は，演算に使用する関数を予め計算し，固定の配列に記録し，参照することにより，どのような関数を用いたかのアルゴリズムが隠蔽する．関数の形が 1 次式等の場合は，入力と出力を照合することにより，関数の形式が推定可能になる．そこで，関数型を複雑化すること，また微小な誤差を入れておくこと，使用しない入力に対応した出力は，間違った値を設定しておき，解析が攪乱されるようにするなどの工夫が可能である[6].

従来方式の難読化では,全てのソフトウェア一般に関するもの,暗号化や認証など，秘密鍵を演算に使用するため，その秘密鍵の秘匿化を行うものが多い．全てのソフトウェア一般に関するものは，上で述べたように対象自体が単純で無いことが必要である．また，暗号化や認証などは，演算自体は，アルゴリズムが定型に近くむしろ鍵などの演算中に現れるデータの隠蔽に焦点が絞られている．一般的な難読化は難易度が高いので，応用分野を設定した上で，それに適した難読化を検討する方が実用的である．

筆者らは，電子透かしの検出器を難読化し，公開領域に提示して行く手法を提案している．電子透かしの埋込み手法として，大関らは，変形フーリエ変換を用いる例を検討している[5]．これは，フーリエ変換のように周波数領域に変換し，低中域の一部にある幅の量子化を行い，透かしとしてのマークとするものである．変換は，周波数的なものに変換するもので，逆変換が存在するものであれば，どのようなものでも良く，フーリエ変換の多数の係数を摂動させたものが無限に近く多数存在する．また，量子化する場所と個数と量子化幅は，誤差の総量を考慮すれば，その範囲で多数の値が採用可能である．電子透かしの埋込みの種類は，これらのパラメータの組み合わせで決定できる．フーリエ変換は線形変換であるが，上記のような変形フーリエ変換は非線形のものや，高次式の一部を使用したり，特殊な関数を使用することも可能である．これらも一括して ROM 化すれば，入力値と出力値の対応関係を多数求めても，大規模な多変数の解析となりうる．

難読化する対象の写像（関数）の入出力の関係と，未知アルゴリズムの解読の難易度の関係について述べる．

計算アルゴリズムの主体が１入力１出力の一次式の場合は,（１）式の例で検討したように，解読は異なる入出力の対応の事例を 2 個取得すれば，簡単にできる．この解読は,アルゴリズムが一次式であるという仮定の下に連立一次方程式を作りなされる．表１に主な計算アルゴリズムとその未知係数部分の解読に関する計算量の元になる未知数の個数との対応を示す．未知係数は，写像（関数）の形式が既知とした時，その係数の個数だけあると想定できる．写像（関数）の形式が未知の場合は，まず，その形式を推定する必要がある．通常，一次式や２次式が多く使用されるので，未知の場合でも，１次式や２次式は，比較的容易に推定でき，未知係数の探索も個数が少なければ数値計算可能になる．一方，写像（関数）の次数が高次であるものや，非線形な関数や特殊関数にまで拡張したものである場合は，未知係数の推定が難しくなり，連立方程式をたてることができなくなる．

単独の写像（関数）の場合でも形式が非線形で入力の変数の個数が増加すれば，写像（関数）の形式を推定することが困難になり，また，形式を仮定した場合でも連立方程式の個数が増大し，数値計算での近似解の精度が悪化して解析が難しくなる．実際の難読化の対象となるプログラムは,電子透かしの埋込みや検出のソフトウェアで,単なる写像（関数）に加え量子化や逆変換などが組み合わされた，複雑な非線形の合成写像（関数）であり，この計算過程を表示すること無く，計算結果のみを表によって表示し出力とすることで，難読化が達成できると考えられる．

図２にある電子透かしの埋込み処理[5]を簡略化したものを示す．計算を表にするため,変形離散フーリエ変換の次数を４に制限してある．変形離散フーリエ変換（MDFT）は変換係数を変化させることで埋込み手法を隠蔽するものである．入力画像のモノクロ成分４画素を１ブロックとし，フーリエ変換のような周波数的な領域へ変換する MDFT で変換する．中間周波数に相当する２個目３個目の成分を所定の特性で量子化器 Quant で量子化し，逆変換部 IMDFT で逆変換され，画像に戻す．

3.2 難読化の構成

図２にある電子透かしの埋込み処理[5]を簡略化したものを示す．計算を表にするため,変形離散フーリエ変換の次数を４に制限してある．変形離散フーリエ変換（MDFT）は変換係数を変化させることで埋込み手法を隠蔽するものである．入力画像のモノクロ成分４画素を１ブロックとし，フーリエ変換のような周波数的な領域へ変換する MDFT で変換する．中間周波数に相当する２個目又は３個目の成分を所定の特性で量子化器 Quant で量子化し，逆変換部 IMDFT で逆変換され，画像に戻す．

図３に４画素のデータの変換を 16 ビットから 24 ビット入力で２バイト（16 ビット）出力の３個の表により構成する例を示す．これを４個の出力に対して４種用意する．また，各表の 16 ビットの出力は出力前に値の並べ替えがなされ，この途中結果の値を観測しても関数関係は推定できない．量子化後の逆変換も同様の構成で実行できる．表関数の容量の見積もりを表２に示す. ROM1 は 128KB で, ROM2,3 が 16MB になる．

Vol.2010-CSEC-51 No.1
2010/12/10

量子化器 Quant は最終段の ROM3 に含めることができる．また，ROM1 は ROM2 と統合できるので，変換と量子化までで，1 出力につき，32MB となる．逆変換は，各出力を 8 ビットとして，図３と同じ構成となり，やはり，ROM1 と ROM2 を統合して，１出力に対して，32MB となる．以上より，埋込み器の総容量は (32MB+32MB)×４出力=256MB となる．検出器の方は，逆変換が不要となるので，32MB×４＝128MB で構成できる．

表１． 写像（関数）の例と未知係数の個数

	写像（関数）の形式	例	未知係数の個数
1	2 入力, 1 出力, 1 次式	$ax+by+c$	3
2	n 入力,1 出力, 1 次式	$a_1x_1+\cdots+a_nx_n+b$	n+1
3	2 入力, 1 出力,2 次式	$ax^2+by^2+cxy+dx+ey+f$	6
4	2n 入力, 1 出力,2 次式	$a_1x_1^2+b_1y_1^2+c_1x_1y_1+d_1x_1+e_1y_1$ $+c_2x_1y_2+\bullet\bullet\bullet+c_nx_1y_n$ $\bullet\bullet\bullet$ $+c_{n(n-1)+1}x_ny_1+\bullet\bullet\bullet+c_{n2}x_ny_{n-1}$ $+a_nx_{n2}+b_ny_{n2}+c_{n2}x_ny_n+d_nx_n+e_ny_n$ $+f$	n^2+4n+1
5	2 入力, 1 出力,3 次式	$ax^3+bx^2y+cxy^2+dy^3+ex^2+fxy+gy^2$ $+hx+iy+j$	10
6	2 入力, 1 出力,特殊関数	$J_{\alpha 1}(x)+J_{\alpha 2}(y)$ （ベッセル関数）	2

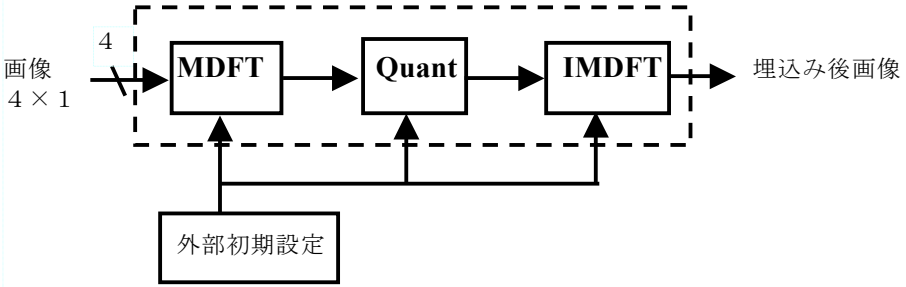

図２ ４次電子透かしの埋込み処理

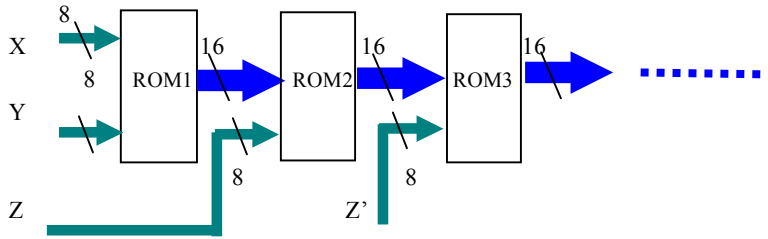

図３ ４次変換のデータフロー（検出器の場合，埋込みの場合の前半部に相当）

Vol.2010-CSEC-51 No.1
2010/12/10

表２　データ容量の見積もり

	入力	ROM1	ROM2	ROM3
入力 アドレス	８×４ （３２）	16	24	24
出力	1	16	16	16
容量 (bits)	2^{16} $=65536$	$2^{16}\times16$ $=1048576$	$2^{24}\times16$ $=16777216$	$2^{24}\times16$ $=16777216$
概数	64KB	128KB	16MB	16MB

3.3 実験

以上のような電子透かしを埋込むプログラムを作成し，難読化を行う．4 次の MDFT として，(2)式のようなアダマール変換を変形した実数の変換を行う例を構成する．(3)式に変形後の変換を，(4)式にその逆変換を示す．逆変換は小数点以下 3 桁まで表示してある．

$$H_4=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\\ 1 & -1 & 1 & -1\end{bmatrix} \tag{2}$$

$$MDFT_4=\begin{bmatrix}1.07 & 0.92 & 1.05 & 0.95\\ 0.92 & 1.03 & -1.1 & -1.1\\ 1.05 & -1.1 & -1 & 1.05\\ 0.95 & -1.1 & 1.05 & -0.9\end{bmatrix} \tag{3}$$

$$IMDFT_4=\frac{1}{4}\begin{bmatrix}1.069 & 1.057 & 0.945 & 0.939\\ 1.057 & 0.871 & -0.918 & -1.019\\ 0.945 & -0.918 & -1.01 & 0.941\\ 0.939 & -1.019 & 0.941 & -1.11\end{bmatrix} \tag{4}$$

例として，図４の画像の(200,100)から横に並ぶ４画素について，式(2)の変換を用いる場合を示す．

図４　実験画像例（720×480 画素，モノクロ）

表３に埋込み処理結果と検出処理の一部を示す．埋込みは４カ所しか無いため，交流成分の平均絶対値最大の第二位置に行う．他に DC に入れる等も可能である．変換後の第二位置に量子化を施すもので，０は IQ=8 にして埋込みを示している．埋込み後のデータに対し，JPEG 圧縮を行って劣化した結果を c,d に示す．検出は，再度変換のみ行い，第二成分の値が，８の４倍を中心として±16 の範囲にあれば，検出されたと判定する．この場合は，e,f の第二成分が 32 であり，検出ができたことになる．

表３　埋込みと検出データ(a-d は画素値，e-f は変換後の値を示す)

	配列位置	1	2	3	4
a	原画像	244	244	244	244
b	埋込み後	252	252	236	236
c	JPEG 1/5 圧縮	252	252	236	236
d	JPEG 1/10 圧縮	246	247	241	239
e	検出 b	976	32	0	0
f	検出 c	976	32	0	0
g	検出 d	976	13	-3	1

3.4 考察

表関数を用いて，計算の手法を隠蔽した電子透かしの埋込み器と検出器を構成した．試行として４次の変換により，例を構成したが，埋込み位置を画像の内側に設定できるため，画像の周辺部を切り取る様な攻撃にも耐性がある．埋込み位置は，従来の周波数の中域成分への埋込みにならって，第二成分にしたが，今後は，変換で，中域の成分が現れる様なものに変更することにより，より，効率が上がり，耐性も向上すると考えられる．一般のプログラムに比べて，容量は増加するが，現在の構成で，検出器は 128MB で有るため，さらに埋込み画素のブロックサイズ４を拡大することが可能である．難読化の種類の数は，形式的には，入力のアドレス数である 24 ビットになっている．また，演算処理の種類は，今回は，提案の中で，４次の線形変換という枠組みが既知となっているが，実際の運用では，未知のパラメータを多数含む非線形の関数とすれば，数値計算による推定の解析演算量は膨大になる．次数が４に限定されていると，関数の形式が明確化しなくても，近似の数値を求めることが容易になると考えられるので，今後は，この次数も増加させる方が良い．本応用で取り上げた電子透かしの埋込みと検出のソフトウェアの難読化では，埋込み方式である関数の形式乃至は埋込み処理の形式まで判明しないと，埋込みを除去することができない．

4. おわりに

Barak らの難読化不可能論は，連続関数に対するもので，また，入力として無限の数のものがあることを仮定している．提案方式の表関数を用いた難読化は離散的な入力に対して行なわれる．難読化は，Barak 等の例のデルタ関数の様な１点しか無いものは簡易で，離散の場合でも定数や１次式などの推定が容易であり，難読化する甲斐の無い関数である．難読化する甲斐のある関数は，もともと直感的に当てたり，簡単な最小二乗法などで推定できるものは，どのように途中を隠蔽しても求めることができる．一方，高次の式や未知係数の多い関数は，数値計算の負荷が増加し，これに対応して難読化を評価することが可能である．

表関数による難読化の例は Chow 等の DES 暗号化処理に見られるが，表にするのは，鍵の演算の部分で，鍵の情報を特に隠蔽したいためと考えられる．従って，全体では，演算時間が多くかかることが問題点として述べられている．本提案では，入力から出力まで，全てを表で扱うことを目指している．実際に入力の画素数が多数になるとき，入力アドレスビット数が増加するので，分割し，多段の構成にしていく必要がある．これにより，結果を得るまでの速度が少し遅くなるが，計算は行っていないので，全体の速度は，大幅に早くなり，難読化によって遅延が増大するような問題はない．今後は，変換式の選定や量子化手法の選定，事前の計算の負荷などの検討を行う．

参考文献

1)　Christian Collberg, Clark Thomborson, Douglas Low, "Manufacturing cheap, resilient, and stealthy opaque constructs", Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, pp.184 - 196, 1998.
2) Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan and Ke Yang, "On the (Im)possibility of Obfuscating Programs", Advances in Cryptology — CRYPTO 2001, Lecture Notes in Computer Science, 2001, Volume 2139/2001, pp.1-18,
3) http://www.cs.princeton.edu/~boaz/Papers/obf_informal.html
4)　Stanley Chow, Philip Eisen, Harold Johnson and Paul C. Van Oorschot, "White-Box Cryptography and an AES Implementation", Lecture Notes in Computer Science, 2003, Volume 2595/2003, pp.250-270.
5)　大関和夫，叢力，「計算量的難読化を仮定した，不特定第三者の認証に依存する電子透かし方式」情報処理学会，研究報告，コンピュータセキュリティ研究会，CSEC-32，pp.61-66,2006 年 3 月.
6)　Yuanyu Wei and Kazuo Ohzeki, “Obfuscation Methods with Controlled Calculation Amounts and Table Function”, Proc. 3rd International Symposium on　Multimedia – Applications and Processing (MMAP), pp. 775 – 780 Oct. 2010.