SCIS 2002 The 2002 Symposium on Cryptography and Information Security
Shirahama, Japan, Jan. 29 - Feb.1, 2002
The Institute of Electronics, Information and Communication Engineers

改行位置の調整による自然言語テキストへの情報ハイディング
Hiding information into natural language text by adjustment of new-line positions

滝澤 修[*1] 牧野 京子[*2] 村瀬 一郎[*2] 松本 勉[*3] 中川 裕志[*4]

Osamu Takizawa[*1] Kyoko Makino[*2] Ichiro Murase[*2] Tsutomu Matsumoto[*3] Hiroshi Nakagawa[*4]

あらまし 情報ハイディングは従来、画像や音声などの信号を情報埋め込み媒体とするものがほとんどであり、文書を対象とするものであっても、行間や語間を調整するなどの画像レベルでの操作によるものが主であった。文字コードレベルでの操作による情報ハイディング手法としては、空白や制御コードなどの不可視な文字を挿入する方法や、単語レベルで類義語に置き換える方法などが提案されているものの、不自然なコードの挿入による露見の危険性や、単語の置き換えによる意味の微妙な変質の懸念が残り、決定的な手法は提案されていない。本稿では、日本語のような膠着言語は改行位置の自由度が高いことを利用し、文書作成に不可欠なコードである改行コードを文書内に挿入する際の位置を調整することによって、情報ハイディングを実現する手法を提案する。本手法は、情報埋め込み媒体である文書の意味の変質をきたすことがなく、また電子データだけでなく印字された文書にも埋め込み情報が残るという特長をもつ。

キーワード 情報ハイディング, 電子すかし, ステガノグラフィ, テキスト, 改行

1. はじめに

計算機ネットワークの利用拡大に伴い、ネットワーク上で情報を安全に伝送する情報セキュリティ技術が重要になってきている[1]。情報セキュリティ技術の一つである情報ハイディングは、情報伝送に際してのカムフラージュ手段、あるいは画像や音楽などの著作物に著作権情報や配布先情報を埋め込む手段としての応用が考えられる。情報ハイディングは、埋め込み媒体が持つ情報の冗長性を利用して別の情報を埋め込むものであり、画像や音響信号など冗長性の多い媒体に対しては、人間に識別できるような劣化をきたすことなく比較的実現しやすい。それに対して自然言語テキスト（キャラクタコード列）を埋め込み媒体とした場合、文字データには冗長性が全く無い[2]ため、情報を埋め込むことで１ビットでも改変されると正しい文字として再生されず、見た目の変質をきたすのみならず、秘匿情報の存在が容易に露見してしまう。そのためこれまでテキスト情報ハイディングとしては、行間や語間を調整するなどの画像レベルでの操作によるものが主であった。文字コードレベルでの操作による情報ハイディング手法としては、空白や制御コードなどの不可視な文字を挿入する方法[3]や、単語レベルで類義語に置き換える方法[4]などが提案されているものの、不自然なコードの挿入による露見の危険性や、単語の置き換えによる意味の微妙な変質の懸念が残り、決定的な手法は提案されていない。

ハイフネーションが必要な英語とは異なり、膠着言語である日本語の場合、改行位置は禁則処理などの例外を除けば比較的自由である。そこで本稿では、日本語を対象とし、文書作成に不可欠なコードである改行コードを文書内に挿入する際の位置を調整することによって、情報ハイディングを実現する手法を提案する。

*1 独立行政法人通信総合研究所
〒184-8795 東京都小金井市貫井北町 4-2-1
Communications Research Laboratory, 4-2-1, Nukuikita-machi, Koganei, Tokyo 184-8795, JAPAN.
*2 株式会社三菱総合研究所
〒100-8141 東京都千代田区大手町 2-3-6
Mitsubishi Research Institute, Inc., 3-6, Otemachi 2-chome, Chiyoda-ku, Tokyo 100-8141, JAPAN.
*3 横浜国立大学 大学院 環境情報研究院
〒240-8501 横浜市保土ヶ谷区常盤台 79-7
Yokohama National University, 79-7 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, JAPAN.
*4 東京大学 情報基盤センター
〒113-0033 東京都文京区本郷 7-3-1
University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033, JAPAN.

2. テキスト情報ハイディングの分類

テキストへの情報ハイディング手法としては、内外でいろいろ提案されているものの、調査報告書[5]などの僅かな例を除き、テキスト情報ハイディングの手法の分類について議論した例は見当たらない。そこで本節ではまず、その分類を試みることにする。

分類A　操作対象による分類

（分類A-1）レイアウトの操作による埋め込み

テキストを印字画像として扱い、Postscript 機能等を活用して、行間・語間[6]や、文字の大小・回転[7]などのレイアウトの操作によって情報を埋め込む手法がある。従来、テキスト情報ハイディングの多くはこの方法である。この方法では文字コードそのものへの操作は伴わず、印字された文字のパターンを処理対象とするものであるため、画像への情報ハイディングの一形態と見なすのが妥当と考えられる。

（分類A-2）文字コード上の操作による埋め込み

不可視な空白文字や制御文字を挿入したり、文字コード定義の冗長性を利用することなどで情報を埋め込む手法がある。また、FinPri.txt[8]のように、日本語文を埋め込み媒体とし、語彙の冗長性を利用することによって、意味的に近い単語で置き換え、文章の意図を大きく変えることなく情報を埋め込む手法もこの分類に属する。

分類 B　情報埋め込み媒体（カバーテキスト）の位置づけによる分類

（分類 B-1）カバーテキストを大きく変質させないことに主眼を置く場合

伝送もしくは頒布しようとするテキストが既に存在し、それに重畳して情報を埋め込む方法がある。著作権保護等を目的とした電子すかしがこれに該当する。

（分類 B-2）カバーテキスト自体を伝送することは主目的としない場合

伝送しようとする秘匿情報がまず存在し、それを秘匿伝送するためにカムフラージュする手段としてテキストを用いる方法がある。ステガノグラフィがこれに該当する。本分類のほうが実現のための制約条件が少なく、実用化しやすいため、先行研究が多く存在する。例えば uuencode ファイルやバイナリデータを自然に見える英文に変換するツールなどが提案されている。生成されるテキスト（ステゴテキスト）は、意味をなす文になっているかどうかまでは追求せず、機械的な検出攻撃に耐える程度の自然な文章になっていることが目標とされる。

分類C　出力結果の変化による分類

分類Cは、情報を埋め込む前（カバーテキスト）と埋め込んだ後（ステゴテキスト）の出力結果（印字結果あるいはCRT上の表示結果）の比較した結果の変化の有無による分類である。従って、カバーテキストを大きく変質させないことに主眼を置く方法である分類B-1を更に細分する分類と言える。但し出力結果については変化の有無は出力系（印字系、表示系）に依存することであるため、分類の境界は明確ではない。

（分類C-1）出力結果が変化する場合

情報を埋め込むことによって、出力結果が変化する場合がある。分類A-1の「レイアウトの操作による埋め込み」はすべてこの分類になると言える。出力結果が変化する方式は見破られる危険が大きい一方、埋め込み情報が出力結果（印字など）に残ることは応用上の利点ともなる。

（分類C-2）出力結果が変化しない場合

出力結果が全く変化しない場合がある。例として、英文を埋め込み媒体とし、複数の空白文字を行末に挿入することにより情報を埋め込むSNOW[3]が挙げられる。SNOWは、行末に0～7個の空白を挿入することによって1行当たり3ビットの情報を埋め込むものである。また、英文のLaTeXソースを埋め込み媒体とし、本文の各行の単語の個数を加減することにより、情報を埋め込む方式も提案されている[9]。この方式は、一般的なLaTeXの出力系において、コンパイル後の表示文書が埋め込む前後で全く変わらないので、この分類に属するといえる。

3. 提案する手法

本稿で提案する手法は、英語のようなハイフネーションが不要な膠着言語としての日本語の性質を生かし、単語を切断するような改行を積極的に行うことで、プレーンテキストの行末の僅かな凸凹に情報を埋め込むものである。

提案する手法は、2節で述べた分類に従うと、分類B-1およびC-1に属するものである(注)。分類Aに関しては、文字コードの挿入による埋め込みであるためA-2であるのと同時に、印字した際のレイアウトにも影響するので、A-1にも属するといえる。

提案する手法は以下の特徴をもつ。

（1）　空白文字のような無意味なコードを挿入せず、

(注)　ソーステキストの改行を表示に反映させないWebブラウザなどを表示系とした場合、提案手法は分類C-2に属することになる。

テキストにとって必須な改行コードのみを利用して埋め込む手法のため、不自然さや作為の発見されやすさが少なく、機械的な検出攻撃にも強いと思われる。

（2）　改行コードを無視しない出力系の場合、出力結果にも埋め込み情報が残るため、電子的な流通だけでなく印字出力としての流通にも有効である。

（3）　膠着言語の場合、単語を切断する改行が許されるため、一行文字数のぶれを小さく抑えることができ、不自然さを少なくすることができる。均等割付された印字出力としての使用を想定すれば一層不自然さを抑えることができる。

（4）　単語は全く置き換えないので、文書の変質劣化はきたさない。そのため文芸作品や契約文などにも適用でき、著作物の権利主張手段としても適している。

（5）　意図的・非意図的な文書の整形によって改行位置が付け替えられることによる無効化攻撃には弱い。また結託攻撃にも弱い。

4. 提案する手法の具体的方法

4.1 概要

提案する手法は、ワープロ文書のように、段落（パラグラフ）の末尾にのみ改行コードが入ったベタテキストを埋め込み媒体(カバーテキスト)とすることを想定し、適当な長さ毎に改行コードを入れることによって秘匿情報（エンベデッドデータ）を埋め込んだ結果、改行が多数挿入された文書（ステゴテキスト）が生成されるというものである。

改行位置と秘匿情報との対応づけについては、単語中の改行位置による方法（方法1）と、一行文字数による方法（方法2）の2つを検討した。以下ではそれぞれについて述べる。

4.2 単語中の改行位置による情報埋め込み(方法1)

方法1では、形態素解析辞書の見出し単語を対象に、各単語（形態素）中の改行位置と、埋め込み情報のビットとの対応関係に基づき情報を埋め込む。例えば図1に例示するように、「する」を「す｜る」と改行したら”1”などと予め定めておく（"|"は改行位置）。埋め込み処理時に指定する基準一行文字数に従い、行末の近傍に来た単語を埋め込み対象とする。図1に示すように、「プログラミング」や「コミュニケーション」などの長い単語は、複数の改行位置を0,1に対応させておき、どれを選んでもいいようにしておく。こうすることで基準一行文字数から大きくかけ離れない文字数で改行できる。

0	1	
する|	す|る	(動詞-自立 サ変・スル)
プログラミング|	プログラミン|グ	(名詞-サ変接続)
プロ|グラミング	プログ|ラミング	(名詞-サ変接続)
言|語	言語|	(名詞-一般)
獲得|	獲|得	(名詞-サ変接続)
コミュニケーション|	コミュニケーショ|ン	(名詞-一般)
コミュニケ|ーション	コミュニケー|ション	(名詞-一般)
コミュ|ニケーション	コ|ミュニケーション	(名詞-一般)
役立つ|	役|立つ	(動詞-自立 五段・タ行)
と|して	として|	(助詞-格助詞-連語)
同時に|	同時|に	(副詞-一般)
こと|	こ|と	(名詞-非自立-一般)
考|え	考え|	(動詞-自立 一段)
言語|	言|語	(名詞-一般)
そこで|	そこ|で	(接続詞)
研|究	研究|	(名詞-サ変接続)

図1　形態素毎のビット対応表の例
（形態素は参考文献[10]の付属辞書に基づく）

翻訳などの実用的な自然言語処理にとっては、性能向上を阻害する	0
困った性質といえます。なぜ人類はこれまでの進化において、プログ	1
ラミング言語のような、もっと曖昧性の少ない効率的な自然言語を獲	1
得してこなかったのでしょうか。それは、曖昧性がコミュニケー	1
ションにとって必要だからではないかと思われます。曖昧性が役	1
立つ例として、大量の意味を少ない言葉に含めたり、複数の意味を同時	1
に伝えたりできることや、特定の相手にだけ真意を伝えられること	0
、状況の変化に応じて新たな意味を容易に定義できること、などが考え	1
られます。無限の状況を有限の言葉によって表現できるのも、自然言語	0
が曖昧性を持っているがゆえに可能なのではないでしょうか。そこ	1
で、自然言語が持つ曖昧性に積極的に着目し、工学的に扱うための研究	1
は、大変重要なものです。	

….

図2　方法1により情報を埋め込んだ例
（右端の数字は埋め込まれた情報(実際は非表示)）

図1の対応表を用いて情報を埋め込んだ例を図2に示す。情報を埋め込んだ単語（形態素）を下線で示している（下線は実際には非表示)。図2は均等割付をしたものであるが、一行文字数のバラツキはほとんど気づかれない程度であることがわかる。図2の例では、“01111101011…”が埋め込まれた情報となる。

方法1は、以下の特長を持っている。

（1）　字種(ひらがな/カタカナ/漢字)による切り分けを行えば、形態素解析を使わず軽い処理が可能。

（2）　単語単位で埋め込み方を定義できるため、後述する方法2と比較して、埋め込み情報のビットと改行との対応関係の法則性を見破ることが困難であり、従って抽出攻撃に強い。

（3）　単語毎に改行位置を定義できるため、不自然な位置での改行を回避することが可能。

一方、課題としては、形態素解析処理の誤りへの対処、一文字形態素への対処などがある。

4.3 一行文字数による情報埋め込み（方法2）

方法２では、埋め込み処理時に基準一行文字数を指定し、各行の文字数と埋め込み情報のビットとを対応させる。例えば文字数が偶数なら 0、奇数なら１などとする。本方法を用いて情報を埋め込んだ例を図３に示す。

```
翻訳などの実用的な自然言語処理にとっては、性能向上を阻害する　0
困った性質といえます。なぜ人類はこれまでの進化において、プ　1
ログラミング言語のような、もっと曖昧性の少ない効率的な自然　1
言語を獲得してこなかったのでしょうか。それは、曖昧性がコミ　1
ュニケーションにとって必要だからではないかと思われます。曖　1
昧性が役立つ例として、大量の意味を少ない言葉に含めたり、複　1
数の意味を同時に伝えたりできることや、特定の相手にだけ真意を　0
伝えられること、状況の変化に応じて新たな意味を容易に定義で　1
きること、などが考えられます。無限の状況を有限の言葉によって　0
表現できるのも、自然言語が曖昧性を持っているがゆえに可能な　1
のではないでしょうか。そこで、自然言語が持つ曖昧性に積極的　1
に着目し、工学的に扱うための研究は大変重要なものです。
....
```

図３　方法２により情報を埋め込んだ例
（右端の数字は埋め込まれた情報(実際は非表示)）

図３の例は、基準一行文字数を全角30文字とし、1行目を30文字、2行目を29文字、、とした結果、図２と同じく、“01111101011…”が埋め込まれた情報となる。

本手法は方法１のようなビット対応表との照合を必要としないため、処理が速く誤処理が少ないと考えられる。反面、埋め込み方の法則性が平易なので、抽出攻撃の危険性が高い問題がある。

課題としては、不自然な位置での改行の回避方法や、半角文字と全角文字とが混在した場合の文字数の計数方法などがある。

5. 実装

本稿で提案した手法について、プロトタイプシステムの実装を現在進めている。現在はまず、4.3 項の方法２（一行文字数による情報埋め込み）について実装を進めている。

実装に際しては、以下に示す要件を満たすこととしている。

（1）多様な文字コードさらには多言語対応を考慮

4.3 項で指摘した、半角文字と全角文字とが混在した場合の文字数の計数方法の問題を回避するため、また、出力結果（ステゴテキスト）の見た目の自然さ（各行の文字密度の均一さ）を保つため、実装に当たっては、基準一行文字数の代わりに「基準一行出力文字幅」を基準とする方法を採用する。

文字幅は、例えばシフト JIS コードの場合、制御コード、ASCII コード、および半角カタカナを 1、かな漢字を 2 と定義する。

文字コードについては、現在はシフト JIS のみを実装対象としているが、文字コードと出力文字幅との関係を定義した対応表を取り替えることで、他の文字コードや他の言語にも拡張可能な設計とする。またフォントの幅も加味した対応表を用いることにより、等幅フォントとプロポーショナルフォントの違いなどにもきめ細かい対応が可能となる。

（2）エンベデッドデータの埋め込み方法

エンベデッドデータの埋込方法については、以下の 2 種類の方法について実装を行なう。

［1］開始/終了を示すフラグシーケンスをエンベデッドデータに付加し、カバーテキスト中の埋込開始位置をランダムにする。即ち開始のフラグシーケンスが現れるまでの改行はダミー改行とする。さらに、エンベデッドデータを分割してカバーテキスト中の複数箇所に散らして埋め込むことも可能な設計とする。

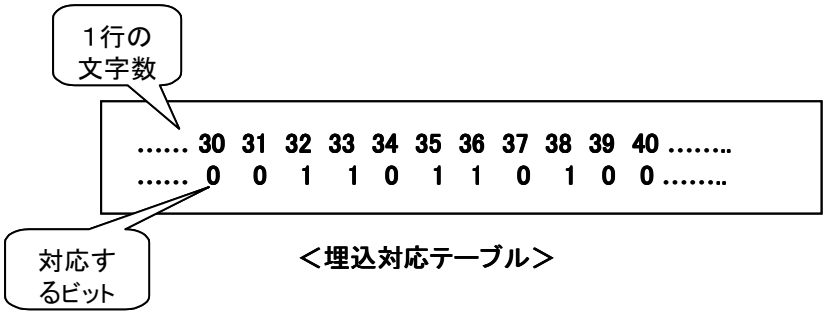
［2］全ての改行にエンベデッドデータを繰り返し埋め込み、情報に冗長性を持たせることで無効化攻撃への耐性を少しでも強化する。但し繰り返し埋め込むことによって抽出攻撃が容易にならないような工夫を講じる。

（3）エンベデッドデータの暗号化

無効化・検出・抽出の３攻撃のうち、本手法は３節で述べた通り、無効化攻撃には弱い。しかし少なくとも抽出攻撃（解読や、なりすまし）は回避する手立てを講じる必要がある。そこでエンベデッドデータを暗号化して埋め込む機能を実装する。但し脆弱性の評価を行うため、暗号化しない埋め込みも可能とする。

（4）情報埋め込みと抽出

実装に際しては、4.3 項で例に挙げたような、偶数を０、奇数を１とする単純な方法ではなく、１行文字数とそれに対応する埋込ビットの対応テーブル（埋込対応テーブル）を埋め込みと抽出に使用する方式を採用する。埋込対応テーブルを参照し、埋込情報の各ビットに対応する文字数のうち、基準１行出力文字幅に最も近い文字数の文字の後に、改行コードを挿入する。



＜埋込対応テーブル＞

ｼﾌﾄJISなどのように1ﾊﾞｲﾄｺｰﾄﾞと2ﾊﾞｲﾄｺｰﾄﾞが混在する場合、1ﾊﾞｲﾄｺｰﾄﾞを「半角」、2ﾊﾞｲﾄｺｰﾄﾞを「全角」とする実装が広く行なわれている。文字幅もそれに対応しているように見える場合が多いため、「半角」、「全角」ということばが文字幅の意味として混同されて使用されている場合があるが、文字ｺｰﾄﾞのﾊﾞｲﾄ数と文字幅の間にはなにも取り決めがない。したがって、文字ｺｰﾄﾞと文字幅については各文字ｺｰﾄﾞ体系毎に文字ｺｰﾄﾞと、（ﾊﾞｲﾄ数、）文字幅の関係を定義することが必要となる。	0(40) 1(33) 0(30) 0(30) 1(32) 0(31) 1(33)

＜出力結果＞

図４　埋込対応テーブルの例と、そのテーブルを用いた出力結果の例
（太数字は埋め込まれたビット、括弧数字は各行文字数）

図４に、埋込対応テーブルの例と、そのテーブルを用いた出力結果の例を示す。

（5）開発言語

開発言語は、開発環境、今後の拡張性、暗号化アルゴリズムの利用を考慮し、JAVA 言語を利用することとする。

6. 議論

4 節で提案した 2 つの方法は共に、情報を埋め込んでも本文の意味内容に全く影響しない特長がある。また、1 行につき必ず１ビットは情報を埋め込む方式なので、埋め込める情報量を保証できる特長がある。

両手法に共通する検討課題として、より不自然でない改行位置の制御がある。即ち、禁則処理や章題、箇条書き、固有名詞等の扱いを定義する必要がある。

Maxemchuk は、電子すかし技術が満たすべき条件として、以下の 3 つを提示している[11]。

（1）　埋め込まれた情報の除去が困難であること。
（2）　埋め込まれた情報を除去した場合、除去された事実がわかるようになっていること。
（3）　埋め込まれた情報を除去した場合、埋め込み媒体の質の低下をきたすようになっていること。

分類 A-2 の場合、埋め込み媒体であるキャラクタコード列自体には１ビットの冗長性も無いため、テキストの見た目の変質をきたすことなくこれらの条件を満たすことは原理的に困難である。そのため、情報が埋め込まれていること自体を隠すことに第一義を置くものとし、いかに自然性を損なうことなく埋め込まれているかを評価の尺度とすべきである。その点で本稿の提案手法は自然なものであり、上記 3 条件を満たさなくても支障の少ないアプリケーションに限定すれば、利用価値があるものと考えられる。

本手法は、改行位置を付け替える整形には無力である。但し一行文字数を揃える機械的な整形はメーラなどで行われるものの、多くの場合、行末を折り返すだけであり、当初の改行コードの位置は保存される場合が多い。本手法において脅威となるのは、改行コードを削除したり改行位置をつけかえたりするような整形であるが、これは、本文と章題や箇条書きとの区別の困難さや、段落途中と段落末との区別の困難さなどから、プレーンテキストを対象としたアプリケーションにおいてはあまり行われていないと考えられる。そのため、非意図的な整形による無効化については、懸念する必要は少ないものと思われる。

提案した手法は、分類A-1とA-2の両方に位置づけられ、両者の特長を併せ持っていると言える。前者は、印字出力としての使用が前提なので、画像ステガノグラフィの手法を適用でき、多くの情報を違和感なく埋め込める可能性がある。後者は、文書をコードとして処理するデジタル的な流通への適用には不可欠な手法であるが、埋め込める情報量が少ない難点がある。それぞれの特長を生かした利用法を考える必要がある。

7. まとめ

本稿では、日本語文を対象として、改行位置を調整することにより、文章の意味を全く変えることなく情報を埋め込むテキスト情報ハイディングとして、２つの手法を提案した。またそのうちの一つの方法について、実装の方針を示した。

現時点では実装作業の途上であり、実システムによる実証および問題点の抽出までは至らなかったが、今後、今回提案した方法の他にも各種方法を検討し、実装した上で不自然さの評価や気づかれにくさの評価を行う予定である。

【謝辞】

本研究は、(株)三菱総合研究所、東京大学中川研究室、横浜国立大学松本勉研究室、および通信総合研究所による定期的な意見交換により得られた成果である。有益な助言をいただいた横浜国立大学松本勉研究室の井上大介氏、赤井健一郎氏、吉岡克成氏、(株)三菱総合研究所の井上信吾氏、通信総合研究所の山村明弘氏、大野浩之氏の各位に感謝する。

【参考文献】

[1] 辻井重男,「暗号と情報社会」,文藝春秋,1999.

[2] 松井甲子雄,「電子透かしの基礎」,p.197,森北出版,1998.

[3] “The SNOW Home Page”, http://www.darkside.com.au/snow/, Feb. 2001.

[4] 中川,木村,三瓶,松本,「辞書変換法に基づく日本語テキストへの情報ハイディング」,情処論, Vol.41, No.8, pp.2272- 2279, 2000.

[5] 情報処理振興事業協会,「インフォメーションハイディングの技術調査」報告書, http://www.ipa.go.jp/security/fy10/contents/crypto/report/Information-Hiding.htm, 平成10年2月.

[6] J.T.Brassil, S.Low, N.F.Maxemchuk, L.O'Gorman, "Electronic Marking and Identification Techniques to Discourage Document Copying", Proc. IEEE INFOCOM '94, Vol.3, pp.1278-1287, 1994.

[7] 中村,松井,「著作権保護のための和文印刷文書への署名情報の埋め込み」, 情報処理学会第50回全国大会, 4N-11, 1995.

[8] 松本, 中川, 村瀬, 「ネットワーク向けインフォメーションハイディング技術開発 テキスト用フィンガープリンティング方式FinPri.txt の開発」, 情報処理振興事業協会 次世代デジタル応用基盤技術開発事業 先端的情報化推進基盤整備事業 論文集, pp.97-104, 2000年6月.

[9] 松本, 糸山, 「Lawful Accessの無効化を狙う暗号通信の検出は容易か？」, 信学技報ISEC96-79, pp.159-164, 1997年3月.

[10] “日本語形態素解析システム茶筌 version 2.0 for Windows”, 奈良先端科学技術大学院大学情報科学研究科 自然言語処理学講座(松本研究室), 1999.

[11] N.F.Maxemchuk, “Electronic Document Distribution”, AT&T Tech.J., Vol.73, No.5, pp.73-80, 1994.