コンピュータセキュリティ 10－8
(2000. 7. 25)

FinPri.txtにおける改竄検出法

村瀬 一郎* 牧野 京子* 赤井 健一郎** 松本 勉**

*東京都千代田区大手町2-3-6
株式会社三菱総合研究所
情報技術開発部
{murase, makino}@mri.co.jp

**横浜市保土ヶ谷区79-5
横浜国立大学大学院工学研究科
人工環境システム学専攻
{akai, tsutomu}@mlab.jks.ynu.ac.jp

あらまし ディジタルデータをネットワーク等により多数のエンティティに配布する際に有効な著作権管理技術として，配布先毎にデータを個別化するフィンガープリンティングがある．我々は，日本語テキストデータへのフィンガープリンティング方式 FinPri.txt (フィンプリドットテキスト)を開発した．FinPri.txtは，日本語のテキストにおいて，置換しても意味が不変に保たれる部分と置換方法を多数抽出しておき，それらに対して，誤り訂正符号とメッセージ認証符号を付加した個別の埋込データを結合させてステゴテキストを生成する技術，ステゴテキストから識別情報を抽出する技術，および，ステゴテキストの改竄を見出す技術等からなる．そのうち，本稿では特にステゴテキストから識別情報を抽出する技術と改竄を見出す技術について述べる．

キーワード 電子透かし，フィンガープリンティング，改竄検出，誤り訂正符号，メッセージ認証符号

Alteration Detection Method in FinPri.txt

Ichiro Murase[1] Kyoko Makino[1] Ken-ichiro Akai[2] Tsutomu Matsumoto[2]

1)3-6, Otemachi 2-Chome,
Chiyoda-Ku, Tokyo
Mitsubishi Research Institute Inc.
Information Technology Development Dep.
{murase, makino}@mri.co.jp

2)79－5 Tokiwadai, Hodogaya,
Yokohama
Yokohama National University
Division of Artificial Environment and Systems
{akai, tsutomu}@mlab.jks.ynu.ac.jp

Abstract

Distributing digital data among many entities via the information network, there is the fingerprinting technology that enables to identify each entity. We have deployed the Japanese text fingerprinting method "FinPri.txt." FinPri.txt implements the word replace method without changing the contents of the text in the database. The system consists of the creating stego-text technology that uses the method ,the error correcting code and the message authentication code, the extracting technology, and the the alteration detection technology. In this paper, we discuss the extracting technology and the alteration detection technology.

key words fingerprinting, alteration detection, error correcting code, message authentication code

1 はじめに

我々は，日本語テキストデータへのフィンガープリンティング方式Finpri.txtを開発した[1]. Finpri.txtにおいては，日本語テキストへのデータの埋め込みによるステゴテキストの生成，誤り訂正符号とメッセージ認証子を付加した埋込データの作成，ステゴテキストからの埋込データの抽出，埋め込み時における改竄の有無の検証，をそれぞれ実装している.

本稿においては，Finpri.txt の概要を説明した後，埋込データの抽出方法および改竄の検出方法について述べる.

2 Finpri.txt の概要

テキストデータへのフィンガープリンティングの試みは少なく，有効な方式の模索が続けられている. 日本語テキストデータへのフィンガープリンティング方式FinPri.txtは，日本語の「カバーテキスト(＝テキストであるカバーデータ)」において，自然言語処理技術を用いることにより置換しても意味が不変に保たれる部分と置換方法を多数抽出し辞書として格納する技術，誤り訂正符号とメッセージ認証符号により冗長性を付加した個別の埋込データをカバーテキストと結合させてステゴテキスト(＝テキストであるステゴデータ)を生成する情報ハイディング技術と，その関連技術から成る.

具体的には，Finpri.txt は，テキストを変換するための規則を編集・定義する「文書変換辞書作成技術」，配布先毎のステゴテキストを生成する「配布先特定文書生成技術」，ステゴテキストに埋め込まれた配布先識別情報の抽出と改竄の検出を行う「不正利用検証技術」，ユーザの要求を統合的に受け付け各技術に受け渡す「融合技術」の4つの技術から構成される. 「文書変換辞書作成技術」は，置換により意味が不変に保たれる置換方法を格納した汎用辞書(汎用文書変換辞書)とカバーデータを入力とし，置換方法の選択を可能とするグラフィカルユーザインタフェースを提供することにより，カバーテキスト毎に変換可能箇所を特定するファイル(文書変換辞書)を出力する. 「配布先特定文書生成技術」は，文書変換辞書を参照し，ステゴテキストである配布文書を生成する. Finpri.txt においては，ステゴテキストはネットワークを介した配布要求が受けた際にステゴテキストを生成するのではなく，予め最大想定配布要求数分のステゴテキストを生成し，データベースに蓄積する方法を採った. 「不正利用検証技術」は，ステゴテキストから埋込データを抽出する機能と、ステゴテキストにおける改竄の有無を，BCH 誤り訂正符号とHMAC メッセージ認証子により、検証する機能を有する. 「融合技術」は，ステゴテキストを格納する「文書管理データベース」を中心として，「配布先特定文書生成技術」と「不正利用検証技術」の稼働を支援する機能を提供する.

これらの関係を図 1 に示す.

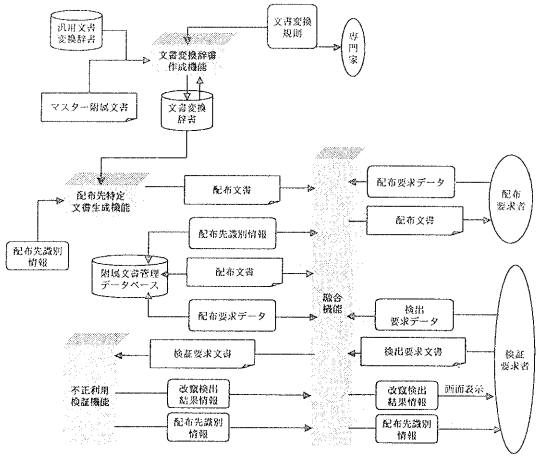

図 1. 開発した機能間の関係

3. 実装

実装したシステムの構成図を図2に示す.

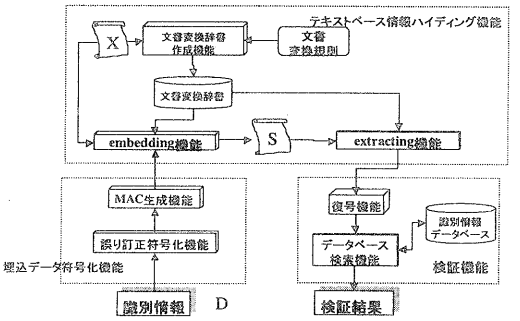

図2. システム構成

3.1 稼働環境

稼働環境を構成するハードウェアを以下に示す.

Server : DOS/V互換機 Pentium 133MHz 相当以上

Client : DOS/V互換機 Pentium 133MHz 相当以上

稼働環境を構成するソフトウェアを以下に示す.

Server

OS : Microsoft Windows NT 4.0以上

WWW Server : Apache1.3.2以上

形態素解析: 茶筅 for Windows V1.0

関係データベース: Oracle 7.0以上(Windows NT版)

ダイナミックリンクライブラリ: Inprise Jbuilder 2 以上

Client

WWW Browser : Netscape Navigator 3.0 以上

3.2 開発環境

開発環境として, OSはMicrosoft Windows NT 4.0以上またはMicrosoft Windows 95以上とした. 開発言語は, Java (JDK 1.1.2 以降に対応), C , C++を使用した. 開発ツールとして,Inprise Jbuilder 2, Microsoft Visual C++を利用した.

4. 日本語テキストへの情報ハイディング

4.1 基本アイディア

日本語テキストへの情報ハイディングは, 同一の内容であっても複数の表現が存在することを利用するものである. 例えば,「意味を変えない置き換え」という文に対して,「意味を変化させない置き換え」,「意味を変えない置換」,「意味を変化させない置換」といった, 同一の意味を持つ文が存在する. 仮に,「変えない」と「置き換え」がビット情報 0 を,「変化させない」と「置換」がビット情報 1 に対応しているとすれば, これら意味の同一な 4 つの文はそれぞれ, 00, 10, 01, 11 という情報を表していることになる.

4.2 表現の置換

4.2.1 一般的な置換

我々が普段使っている言語には, 同一の意味を表現するものであっても, 多種多様な表現が存在する. 同一の意味を示す文書における表現の置き換え方法として以下が存在する.

- ○ 並列句の順番の置き換え
- ○ 受動態から能動態へ, 能動態から受動態へ
- ○ 同義語, 類義語の利用
- ○ 送り仮名, 仮名⇔漢字等の表記揺れを利用する
- ○ 冗長である部分を付加, 削除する

これらのうち, はじめの 2 つは正確な構文解析または意味解析が必要である. これらの処理は現在のところ, 処理時間がかかる上にシステムが意味を正確に把握することにかなりの困難を伴うため, 有効な手法ではない. また, これら構文解析を用いる置き換えの手法は, 文節や文を単位とするので, 同義語等を用いた場合と比べ埋込できる情報量が少ない.

これらの理由により, Finpri.txt においては, 同義語・類義語の利用, 送り仮名・仮名⇔漢字等の表記揺れの利用, 冗長である部分を付加・削除、それぞれにより表現の置換を行うものとする.

4.2.2 置換条件の設定

表現の置換においては, 意味および文法性を損なわないことが求められる. そこで, 形態素解析システムを用いて文から形態素を分解し, 置換可能条件を設定する.

一般に, 置き換えられた表現に近い語ほど, 置き換えによって受ける影響が大きい. 我々はこのことを考慮し, 置換の対象となる表現の周囲の語によって置換可能かどうかの判定を行うような条件を設けた. 形態素解析によって得られる情報は, 品詞の種別やその活用形であるため, 条件を構成する要素として, 置換の対象となる語からの距離(形態素数)と, 語そのもの, 品詞名を選択した. 条件は「位置特定子」+「= または !=」+「品詞識別子または文字列」のように論理式風に記述され, 辞書に登録される.

位置特定子とは, 例えば,

			Pre			Self			Post			
表現	を	置換	する	こと	が	できる	場合	は	、	タグ	付け	を
	5	4	3	2	1		1	2	3	4	5	

のように, 置き換え可能かどうかを判定される語を Self とし, それ以前に出現する語に Self から近い順に Pre1, Pre2, …, Pre5 と名付け, Self 以降に出現する語に近い順に Post1, Post2, …, Post5 と名付けたものである. また, 品詞識別子とは, 名詞, 動詞, 形容詞等にそれぞれ, Noun, Verb, Adj というように別名を付けたものである.

また, 複数の条件を ‖ (または), && (かつ) でつなげることもできる. 例えば「直前が名詞でなく, 直後が動詞でないならば置き換え可能」という条件を記述すると次の通り.

(Pre1 != Noun) && (Post1 != Verb)

4.3 テキストへのデータ埋込方法

システム全体の処理の流れは図3のようになっている.

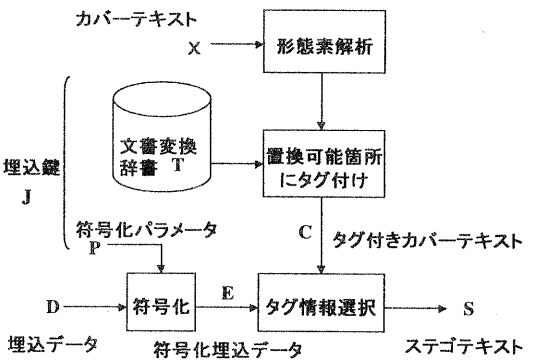

図3. テキストへのデータ埋込方法

ここではカバーテキストXとデータEからステゴテキストCを作る部分について述べる. 埋込データDからデータEを作る部分については後述する.

(1)カバーテキストXを形態素解析によって, 各形態素に

分解する.

(2)文書変換辞書Tの構造は，図4に示すように，置換対象となる語と，それに対する置換候補，置換の際の条件を保持する.

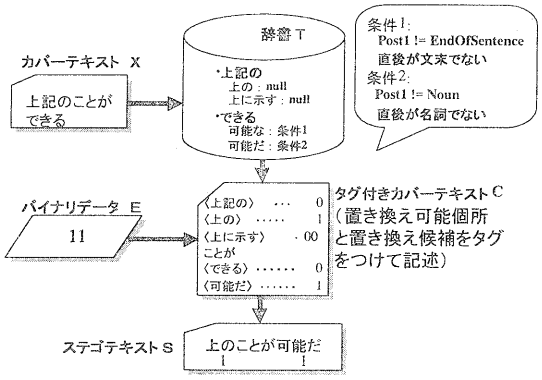

図4. 辞書の構造

(3)形態素解析されたカバーテキストと辞書Tの登録内容との比較を行い，T中に存在する表現がテキスト中にあれば，置換条件による判定を行う．置換が可能な場合，置換が可能である語と共にそれに対する候補もタグ付けされ，タグ付きカバーテキストCとして出力される.

(4)テキストCはHTML形式に変換する専用のフィルタをかけることで，どのように置換えられるかがWebブラウザで確認できるようになっており，C中に置換に不適切な部分があった場合，人手でそれを修正することができる.

(5)符号化埋込データEはバイナリデータであり，これに応じて，タグ付きカバーテキストCのタグ情報が選択され，表現の置換が行われ，ステゴテキストSが生成される.

カバーテキストとして，本稿1.1節の冒頭の3文をとり，これにデータを埋込んだステゴテキストの例を次に示す.

> ネットワークを[利用して]著作権保護の必要なコンテンツのデータを配布する際に[採用される]仕組みとして，次の[様な]ものが[ある]．コンテンツのデータXにコンテンツ管理情報D（著作権者名や配布先名[等]のデータ）を付加し，これに対して[電子署名]付き文書Aを生成して，Aを暗号化して利用者に[配布]する．復号が[行える]のは料金を払う利用者[のみ]としたいので，通常，耐タンパーモジュールというハードウェア[ないし]ソフトウェアを用いて，内部の不正[読出]や[改ざん]が[行えない]環境を[準備する].

5. 改竄検出方式

5.1 基本方式

提案方式は，以下の5フェーズからなる.

・埋込データ符号化フェーズ

・タグ情報選択フェーズ

・改竄検査フェーズ

・分析準備フェーズ

・改竄分析フェーズ

改竄検出における基本方針は，以下の通りである.

1) 「文書管理データベース」に格納されたステゴテキストと埋込データを最大限に活用する

2) 配布先を特定する情報に、パリティおよびメッセージ認証子を付加し、それらを活用することにより、改竄箇所を特定する

以上の基本方針を前提とした改竄検出の5つのフェーズについて、以下に述べる。

5.1.1 埋込データ符号化フェーズ

Step 1-1. ステゴテキストの想定配布数およびカバーテキストの埋め込み可能箇所数を参照し，16ビットからなる配布先識別情報（主データ D）を、想定配布数分生成する。

Step 1-2. ユーザを一意に識別するデータDを誤り訂正符号化し，符号 W を生成する．なお，この時パリティは P とする．通常，画像等の場合媒体の劣化を一定にする為，符号語の重みが小さな一定値であることが望ましいとされる．しかし，本方式の埋込手法では埋込符号語の重みに依らず，媒体の劣化が一定となる性質がある．よって，符号化・復号の容易である通常の線形符号とし，BCH 符号を用いた.

Step 1-3. Step 1-2で生成した符号語Wを入力とし，メッセージ認証符号の認証子生成関数により生成された値MACをWに連接したものを，符号化埋込データEとする．なお，この時メッセージ認証符号をMとする．ハッシュ関数SHA - 1採用の鍵付きハッシュ関数HMACを，認証子生成関数として採用した．図4を参照.

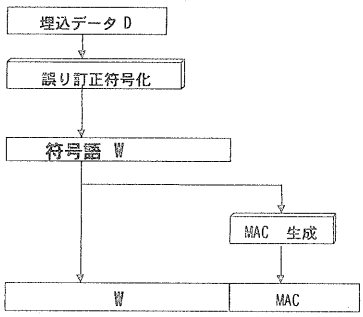

図5. 埋込データの符号化

5.1.2 タグ情報選択フェーズ

Step 2-1. 文書変換辞書Tとカバーテキスト Xから作られたタグ付きカバーテキストCに，2.の方法により符号化埋込データEを埋込み，結果をステゴテキストSとする．埋め込み可能箇所数が，符号化埋込データEより大きい場合，実際にEを埋め込む箇所は，乱数により選定され，いくつかの埋め込み可能箇所に埋め込みがなされない(図6参照)．データベース(文書管理データベース)を用意し，SとDと配布先名の組をこれに登録する．

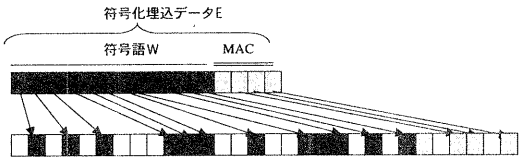

図6. 符号化埋込データEの埋め込み

5.1.3 改竄検査フェーズ

Step 3-1. 与えられたステゴテキストS'(検証を要求された文書)とそのカバーテキストを指定し，「文書管理データベース」に登録されている全てのステゴテキストSとの比較することにより，S'の改竄の有無を検査する．改竄の有無は，全てのステゴテキストSとS'とを全文マッチングを行い，S'と完全に一致するSが存在する場合，「改竄無し」と判断するという方法による．

5.1.4 分析準備フェーズ

Step 4-1. 文書変換辞書Tを参照し，S'から符号化埋込データE'を抽出する．

Step 4-2. 符号化埋込データE'を，主データD'，パリティP'，メッセージ認証子M'に分割する．

Step 4-3. 主データD'とパリティP'から構成される符号語W'に対し，HMACを施し、メッセージ認証子M0'を生成する．

Step 4-4. 主データD'とパリティP'から構成される符号語W'をBCH誤り訂正符号により復号し，主データD1'とパリティP1'を生成する．

Step 4-5. 主データD1'とパリティP1'に対し、HMACを施し、メッセージ認証子M1'を生成する．

Step4-1. ～Step4-5. を図7に示す．

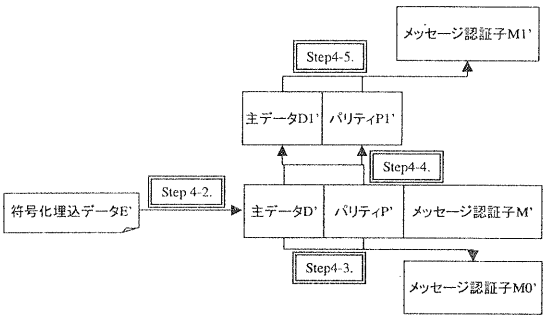

図7. 符号化埋込データの分析の準備

5.1.5 改竄分析フェーズ

Step3-1 において、「改竄有り」と判断された場合には，改竄が施されたステゴテキスト S を特定するために，分析を行う．

Step 5-1. 主データD'またはパリティP'に消失があるか否かを確認する．消失が無い場合Step5-2.に進み，消失がある場合Step5-9. に進む．

Step 5-2. メッセージ認証子M'と(D'とP'から生成された)M0'を比較し，M'とM0'が等しい場合Step5-3. に進み，異なる場合5-4. に進む．

Step 5-3.「文書管理データベース」において，D'を検索し，データベース中に無い場合パターン1とし，有る場合パターン2とする．

Step 5-4. メッセージ認証子M'と(D1'とP1'から生成された)M1'を比較し，M'とM1'が等しい場合Step5-5. に進み，異なる場合Step5-6. に進む．

Step 5-5.「文書管理データベース」において，D1'を検索し，データベース中に無い場合パターン3とし，有る場合パターン4とする．

Step 5-6. D'とD1'を比較し，等しい場合Step5-7. に進み，異なる場合Step5-8. に進む．

Step 5-7.「文書管理データベース」において，D'を検索し，データベース中に、無い場合パターン5とし、有る場合パターン6とする．

Step 5-8.「文書管理データベース」において，D'およびD1'を検索し，データベース中に，D'もD1'も無い場合パターン7とし，D'が無くD1'が有る場合パターン8とし，D'が有りD1'が無い場合パターン9とし，D'もD1'も有る場合パターン10とする．

Step 5-9. D'に消失が存在しない場合Step5-10. に進み，消失が存在する場合パターン13とする．

Step 5-10.「文書管理データベース」において，D'を検索し，無い場合パターン11とし，有る場合パターン12とする．

Step5-1. ～Step5-10. を図8に示す.

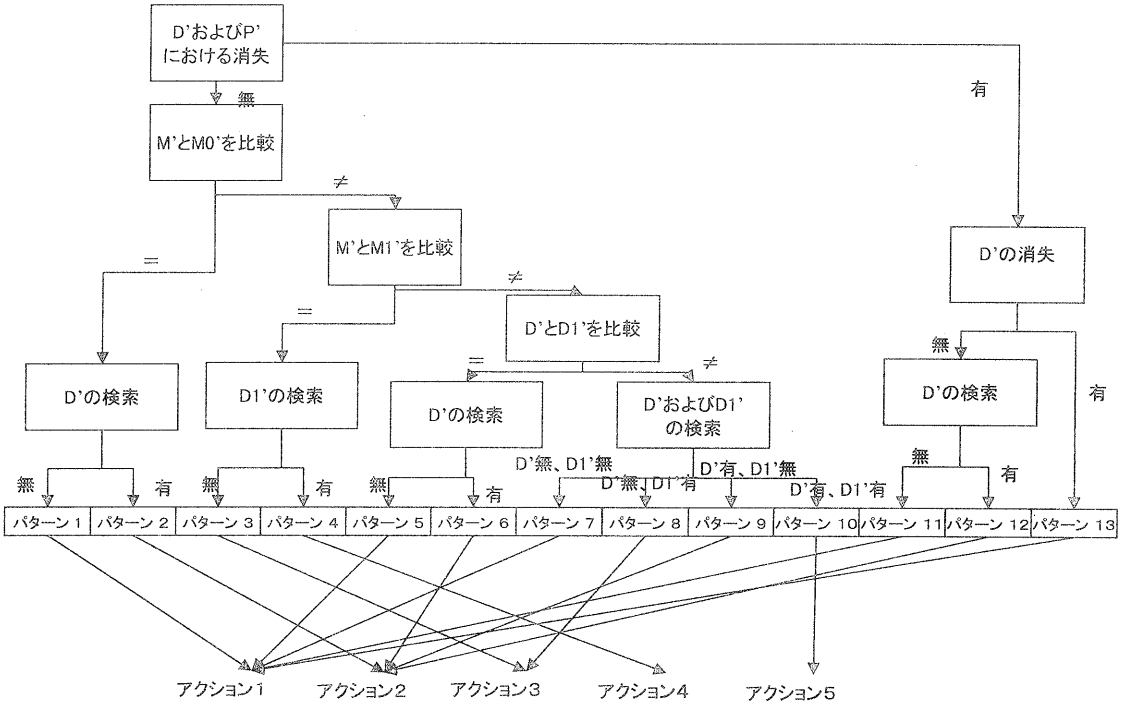

図8. Finpri.txtにおける改竄分析法

図 7 におけるパターンは、表1に示すような解釈が可能である.

表1改竄のパターンとその意味

パターン番号	意味
パターン1	D'(または P')および M'に改竄が加えられ, 偶然に M'=M0'となった
パターン2	埋め込み可能箇所に対する改竄はなされていない
パターン3	D'または P'に改竄が加えられ, 誤り訂正によっても補正不可能であった
パターン4	D'または P'に改竄が加えられたが, 誤り訂正により補正可能であった
パターン5	D'(または P')および M'に改竄が加えられ, 誤り訂正によっても補正不可能であった
パターン6	D'(または P')および M'に改竄が加えられ, 誤り訂正により補正可能であった
パターン7	D'(または P')および M'に改竄が加えられ, 誤り訂正によっても補正不可能であった
パターン8	D'(または P')および M'に改竄が加えられ, 誤り訂正により補正可能であった
パターン9	D'(または P')および M'に改竄が加えられ, 誤り訂正により補正可能であった
パターン10	D'(または P')および M'に改竄が加えられ, 誤り訂正により補正可能であったが, なりすましの可能性がある
パターン11	D'または P'が消失がしていたが, 補正不可能であった
パターン12	D'または P'が消失していたが、補正可能であった
パターン13	D'が消失していたが、補正不可能であった

また, 図7におけるアクションは, 表2に示すように, 主データの推定法を示すものである.

表2(改竄が有った場合の)主データの推定法

	主データ D の推定方法
アクション1	D'とハミング距離が近い候補をデータベースから抽出する
アクション2	D'が候補である
アクション3	D1'とハミング距離が近い候補をデータベースから抽出する

アクション4	D1'が候補である
アクション5	D'および D1'が候補である (D'≠D1'であり，一方に特定することは不可能である)

6. 実証実験

6.1 実験項目

FinPri.txt が達成するステゴテキストの品質(「意味同一性」および「文書正当性」)と耐性(「改竄耐性」)を，ソフトウェアに付属するマニュアル文書や利用許諾文書を対象とした実証実験により評価した．

ここで，品質とは，「同一のカバーテキストに対応するステゴテキスト間で意味に整合性が取れている(意味同一性が確保されている)程度，ならびに，ステゴテキストが，文法や技術，法律の観点から正当である(文書正当性が確保されている)程度」を表す．また，改竄耐性とは，「ステゴテキストに改竄がなされても，埋め込まれた識別情報が正しく抽出できる程度」を表す．本稿においては，特に改竄耐性を評価する改竄耐久性実験に関して述べる．

6.2 実験準備

6.2.1 カバーテキストの整形

カバーテキストは，実験参加者である NTT ソフトウェア及びジャストシステムから，ソフトウェアに添付する文書(マニュアル，使用許諾)の提供を受けた．これらから，固有名詞の削除，サイズの調整等を行って，実験に使用しても問題のない形態に整形した．用意したカバーテキストの構成は表1のとおり．

6.2.2 文書変換辞書の作成

文書変換辞書作成機能を使って，変換規則のデータベースである文書変換辞書を構築した．具体的には，汎用文書変換辞書をもとに，規則生成機能で自動的に生成された変換規則について，辞書作成者がチェックし，文書変換規則編集機能を操作して適当な規則へと編集した．このうち，マニュアル(小)及び使用許諾の専門家版文書変換辞書については，テクニカルライタがマニュアルを，法務担当者が使用許諾を担当した．

6.3 耐性検証実験の概要

本実験は，文書変換辞書作成機能，及び配布先特定文書生成機能を用いて生成されたステゴテキストに対する配布要求者による改竄への本方式の耐性を検証することを目的に実施した． 実験参加者は表3のとおりである．

表3 実験参加者の構成

実験参加機関	参加人数
三菱総合研究所	実験管理 2名 実験結果検証 2名
横浜国立大学	改竄作業管理 2名 被験者 76名
その他 (横浜国立大学 OB および関係者)	被験者 7名

改竄されたステゴテキストからの配布先識別情報の抽出可能性を測定・分析することにより行った．具体的には以下の実験を行った．

(1)結託攻撃に対する耐性検証実験

特定の利用者が，同一のカバーテキストから同一の文書変換辞書により派生した複数のステゴテキストを入手し，それらの比較を通じてステゴテキストを本来の意味を損なわない範囲で改竄した場合の配布先識別情報の抽出可能性を測定し，FinPri.txt が結託攻撃に対する耐性を有していることを検証した．

(2)文書変換規則に応じた改竄に対する耐性検証実験

利用者が汎用文書変換辞書ならびに FinPri.txt で採用されているデータ埋め込み方法，即ち文書変換規則知った上で，ステゴテキストに対し意味を損なわない範囲の改竄を行った場合の，その改竄されたステゴテキストに対する配布先識別情報の抽出可能性を測定し，FinPri.txt で用いられている文書変換規則に則った改竄に対する耐性を有していることを検証した．

(3)不特定改竄に対する耐性検証実験

利用者がステゴテキストに対して意味を損なわない範囲で自由に改竄を行った場合の，その改竄されたステゴテキストに対する配布先識別情報の抽出可能性を測定し，FinPri.txt が不特定の方法が採られる改竄に対しての耐性を有していることを検証した．

(4)埋め込み可能箇所数に応じた耐性検証実験

埋め込み可能箇所数の多い文書変換辞書と埋め込み可能箇所数の少ない文書変換辞書によりそれぞれ生成されたステゴテキストに対し，利用者が意味を損なわない範囲で改竄を行った場合の，その改竄されたステゴテキストに対する配布先識別情報の抽出可能性を測定し，ステゴテキストの埋め込み可能箇所数の違いが FinPri.txtの改竄耐性に与える影響を検証した．

6.4 実験結果

耐性検証実験の結果から，次の点が指摘できる

(1)総論

改竄ステゴテキストから配布先が特定できた割合は

- ○ 結託攻撃改竄: 15.6%
- ○ 文書変換規則に応じた改竄: 17.9%
- ○ 不特定改竄: 41.2%
- ○全体: 25.0%

であった．最も低い結果となった結託攻撃改竄に対する検証実験でも，全文書の約 15.6%について配布先

を特定できており，改竄者のリスクを考慮すれば，十分に実用に耐えうる改竄耐性があるといえる．

(2)改竄のパターン

それぞれの実験における改竄のパターンは、表 4 に示す通りである．

表 4 それぞれの実験における改竄パターン

実験名	配布総数	改竄パターンとその数(推定が正しかった数)
結託攻撃	32	パターン 5: 10(1) パターン 6: 1(1) パターン 7: 3(0) パターン 8: 2(2) パターン 11: 5(0) パターン 12: 1(1) パターン 13: 10(0) 推定率 5/32=15.6%
文書変換規則	39	パターン 5: 13(2) パターン 7: 9(1) パターン 8: 2(2) パターン 11: 8(1) パターン 12: 1(1) パターン 13: 6(0) 推定率 7/39=17.9%
埋め込み可能箇所数	34	パターン 2: 4(4) パターン 5: 4(0) パターン 6: 3(3) パターン 7: 5(1) パターン 8: 2(2) パターン 11: 4(0) パターン 12: 1(1) パターン 13: 11(3) 推定率 14/34=41.2%
全体	104	パターン 2: 4(4) パターン 5: 27(3) パターン 6: 4(4) パターン 7: 17(2) パターン 8: 6(6) パターン 11: 17(1) パターン 12: 3(3) パターン 13: 27(3) 推定率 26/104=25.0%

表 3 より，以下が考察できる．

(1)BCH 符号および HMAC により，元の配布先が1つに特定できた場合，その結果はほぼ正しい．

(2)ハミング距離により、配布先を特定した場合は，推定の確からしさは非常に低くなる．

本実験においては，文書サイズが小さいため，埋め込み可能箇所を十分に確保することが不可能であったことを考慮すると，埋め込み可能箇所を多くすることにより、推定率はさらに上昇することが予測される．したがって、以下の方法により埋め込み可能箇所を増やすことは有効であると推察される．

○適用対象とするカバーテキストのサイズを大きくする．

○データ埋込方法を同義語または語句への置換に限定せず多様化させる．

6.5 全体考察

実験から，FinPri.txt は，耐性のにおいて十分な実用性を有しており，テキストデータやそれをマニュアル等の付属文書とするソフトウェアやコンテンツの著作権管理のために有用な技術であることが確認できた．

結託攻撃，文書変換規則に応じた改竄，不特定改竄を実際に行い，(攻撃者にとって最も有利である)文書変換規則に応じた改竄に対しても，本方式は1割を超す割合で配布先を特定できる改竄耐性を有することを検証した．

7. 今後の展開

上に見たように，FinPri.txt は十分な実用性を有しているが，この技術をより優れたシステムとして実用化する際には，以下の諸点を考慮することが適当と考えられる．

(1)適用文書として，マニュアルが適当である．文書量を増やすことで，埋込可能箇所密度を高めることなく，埋込データ量の増加にも対応できる．

(2)改竄耐性を高める方策として，複数のデータ埋込方法の組合せなど利用者に判明しにくいデータ埋込方法の採用や，置換可能な同義語が想像しやすい箇所にデータ埋込みを行わないなどの対策が考えられる．

(3)また，埋め込みデータの冗長性を確保する等により，データ消失に対する対策を講じるか，データ埋込箇所の変換候補数を増やすことでも，改竄耐性の向上が期待できる．

謝辞

本技術開発は，情報処理振興事業協会(IPA)「次世代デジタル応用基盤技術開発事業」の一環として平成 11 年度に行われた．東京大学情報基盤センターの中川裕志先生，三菱総合研究所，横浜国立大学の関係諸氏は，システム開発から実証実験に渡る本プロジェクトの全般に渡って貢献された．また，実証実験においては，NTT ソフトウェアの関係各位およびジャストシステムの関係各位にはカバーテキストの提供と品質検証実験にご協力をいただいた．ここに謝辞を述べる．

参考文献

[1]松本勉,中川裕志,村瀬一郎:``テキスト用フィンガープリンティング方式の開発",情報処理振興事業協会次世代応用基盤技術開発事業論文集,pp.97-104,2000.