電子透かしを用いた手軽な不正複製,改ざん抑制の提案

2003MT039 河路 将宏　　2003MT088 大澤 健仁
指導教員　後藤 邦夫

1　はじめに

近年のインターネットの急速な普及・デジタル化に伴い，デジタルコンテンツの製作者に無許可での不正な複製や改ざんが容易になり，製作者への著作権侵害が問題となっている．そこで，コンテンツに電子透かしを埋め込むことが提案されている．電子透かしとは電子文書や画像，音楽などのデジタルコンテンツの中に，著作者の固有の識別符号を密かに埋め込み，著作者が著作権の所在を主張しようというものであり，著作権侵害に対する検証の道具として利用することができるものである [1][2].

本研究では高校や大学などの教育機関において配布される授業資料（コンテンツ）の不正な複製や改ざん，また持ち出し抑制を目的とした手軽な電子透かし埋め込み機能を組み込んだリアルタイムでのコンテンツ配信サーバを提案する．この手軽とは，透かし埋め込みに時間のかかる複雑な通信システムや難解な透かし埋め込みアルゴリズムを避け，プログラム処理時間の短いリアルタイムでのコンテンツ配信を行うことである．

コンテンツ配信サーバには，ユーザ認証機能を組み込み，ユーザ毎にユーザ固有の異なる透かし（ユーザ名，ユーザの IP アドレスなど）を動的に埋め込む．これにより，不正に複製されたコンテンツから透かし情報を検出して不正複製者を特定することができる．

そして，配信するコンテンツファイル形式は，教育機関で配布される授業資料（コンテンツ）として考えられる HTML プレインテキスト，静止画，音声データの 3 種類とした．なお，河路は主にプログラム作成を担当し，大澤は主に実験環境の構築を担当した．

2　システムの概要

この節では，電子透かしとシステムの概要について説明する．

2.1　電子透かしについて

電子透かしは，人間の知覚特性を利用して，デジタルコンテンツに対して，コンテンツとは別の情報を人間に知覚できないように埋め込む技術である．コンテンツ自体に変更を加えるが，品質を劣化させずに情報を埋め込むことが重要である．

電子透かしの用途は，不正複製者の特定と改ざんである．不正複製者の特定は，コンテンツ配信のさいにコンテンツにクライアント情報や著作権情報を電子透かしとして埋め込めば，インターネット上に配布された不正複製コンテンツから，透かし情報を検出し不正複製者を特定できる．改ざん検知は，図 1 に示すように，コンテンツに特定の情報を埋め込んでおき，それが破壊されたことで不正改ざんを検知できる [3].

本研究でのコンテンツに埋め込む情報は，不正コピーされたさいに流出元を特定し，不正複製者に対して著作権の所在を明示的に証明するために必要と考えられる次の 9 つとする．

- 著作物名
- 著作物の種類
- コンテンツ配布日時
- ユーザ名
- クライアントの IP アドレス
- 著作者の名前
- 著作物の完成年月日
- サーバ名
- クライアントのブラウザ

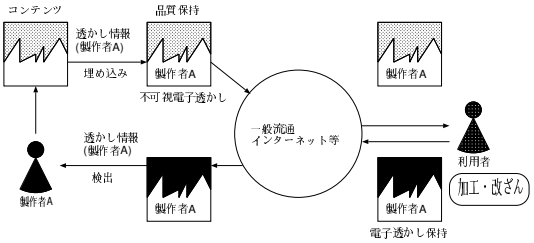

図 1　電子透かしの流れ

2.2　提案するシステム

本研究では高校や大学などの教育機関において配布される授業資料（コンテンツ）の不正な複製や改ざん，また持ち出し抑制を目的とした手軽な電子透かし埋め込み機能を組み込んだリアルタイムでのコンテンツ配信サーバを提案する．

コンテンツ配信環境は，学内ネットワークにおいて 1 台のサーバ・複数のユーザ（生徒）のサーバ・クライアント方式を想定している．サーバはクライアントからの要求に応じて，リアルタイムで動的にユーザ固有の異なる透かし（ユーザ名，ユーザの IP アドレスなど）を生成し，ユーザに透かし入りコンテンツの配信を行う．また，配信者があらかじめ人数分の電子透かし入りコンテンツを生成しておき，配信するという手法も考えられるが，人数分の透かし入りコンテンツを先に生成しても全員がアクセスするか分からないこと，人数が多いと透かし入りコンテンツを先に生成するのに手間がかかり，大変であるという理由から本研究は先に述べた手法をとる．

そして，コンテンツ配信のさいに，コンテンツ自体に電子透かしが埋め込まれていることをユーザに告知する．コンテンツに電子透かしが埋め込まれているということを告知することで，ユーザの不正な複製や二次配信を抑制することができると考えられる．

2.3 コンテンツ配信の流れ

ここではコンテンツ配信の流れを説明する（図 2 参照）.

1. コンテンツ配信希望者は，あらかじめユーザ名とパスワードの登録をする.
2. ユーザからユーザ認証の要求がある度に，入力されたユーザ名とパスワードが登録されたユーザ名とパスワードと一致するか確認をする.
3. ユーザ名とパスワード共に一致したユーザからダウンロードファイルのリクエストを受け取り，そのファイルタイプの判定をする.
4. 判定を行ったファイルタイプに応じた透かしをコンテンツに埋め込む.
5. 透かしを埋め込んだコンテンツをユーザに配信する.

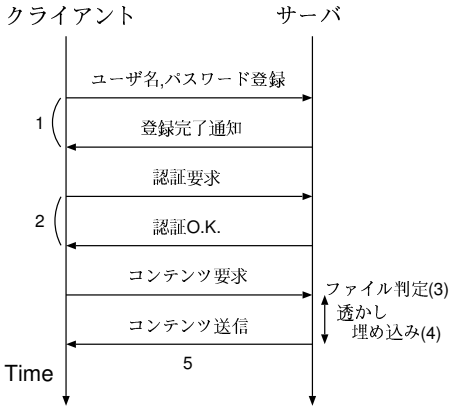

図 2　コンテンツ配信の流れ

3　電子透かしの実現

この節では，電子透かしの埋め込み手法，またその実現について説明する.

3.1 テキストデータへの電子透かし

テキストデータへの電子透かし埋め込み手法には，ワード置換法とワードスペーシング法，文字間隔を制御する方法，文字の回転や縮小倍率を制御する方法があるが，本研究ではワードスペーシング法を用いた [4]．その理由は，ワード置換法では単語自体を変えるため厳密な文書で不都合が生じるためである．またワードスペーシング法は，単語間のスペースを透かしビットによりわずかに変化させて情報を埋め込む手法で，他の方法に比べて埋め込み処理における計算時間が短く済むためである.

第 2 節で挙げた埋め込む情報をワードスペーシング法を用いてビット単位で埋め込む．なお本研究では，ビット 0 の時は「 word1　word2（single space, double space)」とし，ビット 1 の時は「 word1 word2（double space, single space)」 とした.

透かし情報の検出はビット単位でそのまま埋め込んであるので容易にできる．テキストファイルから「 word1 word2（single space, double space)」であればビット 0,「 word1 word2（double space, single space)」であれば，ビット 1 としてビットデータを取り出し，ビットデータを順に並べることで署名情報を検出する.

また，この手法では日本語テキストの場合単語間スペースがないため埋め込むことができない．そこで日本語テキストの場合は別の手法を用いた．日本語テキストの透かし埋め込みは，透かし情報がビット 1 の場合，ある特定の文字または記号があった時，その後にスペースコード'0x20'+ バックスペースコード'0x08' の文字コードを入れる．スペースを文章の途中に入れても，バックスペースが次にあるので見た目は何も変わらない.

3.2 静止画データへの電子透かし

静止画への透かし埋め込み方法には，画素置換法と周波数を利用した方法，画素空間利用法があるが，本研究では画素置換法を用いた．その理由は，周波数利用の方法と画素空間利用の方法に比べ，画素置換法は計算処理が少ないので速いからである.

静止画の明るさの変化（輝度）は最大 256 段階あり，人には± 1 の変化は気付きにくいことを利用して電子透かしを埋め込んだ．画像は PPM ファイル P6（カラー，バイナリ）形式に変換して扱う．理由はピクセルの持っている輝度情報を単純に数値化して配列したフォーマットであり，ビット単位で容易に埋め込むことが出来るからである.

埋め込み方法は，電子透かしの情有効な対策であると考えられる．報を 8 ビットで埋め込むので，電子透かし情報がビット 0 の時，輝度値を偶数に変更する．輝度値が奇数であったものは 1 を加え，元から偶数の場合は変更しない．輝度値が 255 の場合は，輝度値 256 はないため 1 減らす．電子透かし情報がビット 1 の時，輝度値を奇数に変更する．輝度値が奇数の場合は変更せず，偶数の場合 1 を加える．例えば「A」という文字を埋め込むとする．「A」の文字コードを 8 ビットで表すと 01000001 となる．これを輝度値が 225，135，124，225，…と続く画像に埋め込むと表 1 のようになる.

表 1　画素置換法による電子透かし埋め込みの例

輝度番号	輝度値	透かし情報	変更後の値
1（R）	225	0	226
2（G）	135	1	135
3（B）	124	0	124
4（R）	225	0	226
5（G）	135	0	136
6（B）	124	0	124
7（R）	226	0	226
8（G）	136	1	137

静止画に埋め込んだ電子透かし情報は，輝度値を先頭から参照することで検出が可能となる．輝度値が偶数値の時は 0 を，輝度値が奇数値の時は 1 を取り出す．0 と 1 を並べることで埋め込んだ透かし情報が読める．コンテンツ配信のさいには，PNG 形式に変換して配信する．

圧縮は，JPEG 圧縮では通常圧縮時にデータを切り捨てて，画像の劣化の代わりに高い圧縮率を得ているため透かし情報は壊れてしまう．透かしを壊さずに圧縮するには，lossless 圧縮を用いる必要がある，lossless 圧縮は，通常ファイル全般を圧縮する時に使用し，1 ビットも欠けないようにする圧縮方法である．

3.3 音声データへの電子透かし

音声データへの透かし埋め込みは，PCM サンプルデータ（WAV）を使用した．また，他の形式は WAV に変換して用いた．WAV の構造はデータの説明（ヘッダ）とデータが順に記憶されているので，データ部分を変更し，透かし埋め込みには，データ説明部分は使わないため，Linux sox コマンドでデータ部分のみの音声ファイル，RAW 形式にする．RAW は，一般的に 8 ビットサンプルまたは 16 ビットサンプルであるため 8 ビットサンプルのデータは 16 ビットサンプルへ変換して用い，16 ビットサンプルのデータはそのまま透かしを埋め込む．透かしは，3.2 節の静止画データへの電子透かしと同じ要領で埋め込む．透かし埋め込み後はヘッダを付け直し，WAV 形式として配信する．

本研究に使用している CPU では，2 バイト以上のデータ量を持つ数値データを記録するときに 1 バイトごとに分割し，下位のバイトから順番に記録する Little Endian であるため，透かしは奇数バイトの最下位ビットへ埋め込む．これは上位バイトの値を少しでも変更してしまうと，大きく音が変わってしまうためである．音声ファイルは 16 ビットサンプル形式でできているため，2 バイトごとに埋め込む．

検出は静止画データへの電子透かしと同様に，奇数バイト目の最下位ビットが奇数か偶数かで判断する．

4 Web サーバにおける透かし処理の実現

この節では，Web サーバにおける透かし処理の実現について詳しく説明する．

4.1 CGI プログラム

CGI プログラムの処理の流れは，次のようになる（図 3 参照）．

1. ユーザ認証を行う．ユーザ名，パスワード共に一致しない場合は再度ユーザ認証をする．
2. ファイルタイプに応じて透かし埋め込み手法が異なるのでユーザからリクエストのあったファイルがテキスト，静止画，音声データのどのタイプであるか判定をする．
3. ユーザ固有の情報（ユーザ名，ユーザの IP アドレスなど）とリクエストされたファイルに関する情報（著作者の氏名，著作物名など）を記述した署名ファイルを動的に生成する．
4. 生成された署名ファイルの内容を透かしとしてコンテンツに埋め込む．
5. 透かし入りのコンテンツをユーザに配信する．

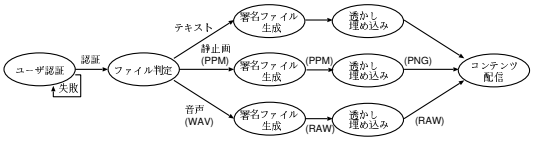

図 3　CGI プログラムの処理の流れ

4.2 ユーザ認証機能

ユーザ認証の流れについて詳しく説明すると次のようになる．

1. コンテンツ配信希望者は，あらかじめユーザ名とパスワードの登録をする．
2. 登録されたユーザ名はそのまま，パスワードは DES（Data Encryption Standard）によってパスワードハッシュ値に変換して認証用パスワード専用ファイルに格納する．
3. ユーザからユーザ認証の要求がある度に，入力されたユーザ名とパスワードが格納されているそれらと一致するか確認する．

4.3 ファイル判定機能

本研究では，2.3 節のコンテンツ配信の流れで説明した通り，ファイルタイプ判定機能を Perl の「File::MMagic」モジュールを使用して組み込んだ．その利点は以下の 2 点である．

- 今後配信するコンテンツが増加したさいに，そのコンテンツのタイプ判定機能が組み込まれていれば，CGI プログラムを大きく変更する必要がなく，プログラミング労力を軽減することができる．
- 配信するコンテンツの拡張子に誤りがあった場合にも正確なファイルタイプに応じた透かし埋め込みを実現することができる．

4.4 署名ファイル生成

サーバはユーザ固有の情報（ユーザの IP アドレス，コンテンツダウンロード日時，ユーザのブラウザ名）が書かれた署名ファイルを動的に生成しなければならない．そこで，本研究では Web サーバとクライアント（Web ブラウザ）がデータを送受信するさいに送信される環境変数を利用してユーザの IP アドレス，コンテンツダウンロード日時，ユーザのブラウザ名といった情報を得て，署名ファイルの生成を行った．また，ユーザ名に関しては，ユーザ認証のさいに得ることができる．

5 評価

この節では，システムとテキストデータ，静止画，音声の電子透かし埋め込み実験の評価について説明する．

5.1 システムの評価

第 3 節で説明した透かし埋め込み方法を用いて本研究で提案したシステムでファイルダウンロードの実験をした．CGI プログラムにより，ユーザはユーザ認証，ファイルリクエストともにキーボード入力とクリックといった簡単な操作で，ファイルをダウンロードすることができた．操作は簡単であり，誰にでもできるものと考えられ，本システムの利便性は高いと考えられる．

また，透かし埋め込み処理時間は，実験に使ったテキストファイル 59.8KB と静止画ファイル 192KB においては時間を待つことなく配信することができた．しかし，音声ファイル 2.4MB に透かしを埋め込むさい，数秒から数十秒程度かかり，実用するには音声ファイルに対しての埋め込み時間の短縮が必要であると考えられる．

5.2 テキストデータへの電子透かし埋め込み評価

今回のこの埋め込み手法は，埋め込む情報データ 1 文字につき単語間スペースを 16 個使用していることから，他人が署名情報を引き出すことは難しいと考えられ，実用性があると判断できる．テキストデータへの電子透かしプログラムでは，埋め込むデータの量にもよるが，大抵の場合長い文章を必要とする．実験では単語数が 8021 である RFC2462 を用いたため署名情報が多量でも埋め込むことができた．しかし，埋め込み先の文章自体が短い場合，この手法では埋め込みきれないことがある．

また，日本語テキストへの電子透かしについては表面上は全く変化が見られない．テキスト自体には変化はないが，データを見ると透かしを入れた部分にスペース＋バックスペースが見られた．

5.3 静止画への電子透かし埋め込み評価

本研究で利用した原画像（lenna）[*1]（図 4）を次に示す．図 5，図 6 はそれぞれの画像の最も左上の点（0, 0）から（1, 1）の 4 ピクセルの拡大画像である． 静止画へ

図 4　透かし埋め込み前の原画像

の情報埋め込みは，最大 256 段階ある輝度値を 1 だけずらすことにより実現している．この変化は図 5，図 6 から分かるように見かけは全く同じであり，人の目では

(0, 0) R:225 G:135 B:124	(0, 1) R:225 G:135 B:124
(1, 0) R:225 G:126 B:122	(1, 1) R:225 G:136 B:122

図 5　電子透かし埋め込み前の拡大画像

(0, 0) R:224 G:136 B:125	(0, 1) R:224 G:135 B:124
(1, 0) R:225 G:126 B:122	(1, 1) R:225 G:136 B:122

図 6　電子透かし埋め込み後の拡大画像

気付くことがまずできない．さらにこの手法では，3 × 256 × 256 バイトある画像ピクセルのうち，透かし埋め込みに使用している部分は先頭の 200 バイト程度なので画像に全く損傷を与えていない．

5.4 音声への電子透かし埋め込み評価

音声への透かし埋め込みは，透かし埋め込み前と埋め込み後の音楽を聞き比べたところ音程やリズムの違いを全く認識することができなかった．静止画と同様にデータの値を 1 だけ変更した程度では，人間には認識できないレベルの違いである．よって音声データへの透かし埋め込みは，音声データ自体にはほとんど損傷を与えず，透かし埋め込みに成功したと言える．

6 おわりに

本研究で提案してきた手軽な電子透かし埋め込み機能を組み込んだコンテンツ配信サーバの提案は，コンテンツ配信のさいに，コンテンツ自体に電子透かしが埋め込まれていることをユーザに告知することで，ユーザの不正な複製や配信を抑制することができると考えられる．また，前節の評価から分かるようにシステムは利便性が高く，電子透かしは，コンテンツのどこにどのような情報が埋め込まれているか解読することがを難しいと考えられるため実用的であると考えられる．

本研究では，デジタルコンテンツの不正複製の抑制という面では効果が期待できると考えられるが，今後の課題としては，JPEG など lossy 圧縮すると透かしが消えてしまうので，消えない埋め込み方法の考案や透かし埋め込み処理時間の短縮が挙げられる．

参考文献

[1] 松井甲子雄：電子透かしの基礎，森北出版株式会社 (1998).

[2] 小松尚久，田中賢一：電子透かし技術，東京電機大学出版局 (2004).

[3] 小野　束：電子透かしとコンテンツ保護，pp. 169–173, オーム社 (2001).

[4] 特許庁総務部技術部技術調査課技術動向班：電子透かし. http://www.jpo.go.jp/shiryou/s_sonota/hyoujun_gijutsu/denshi_sukashi/index.html.

[*1] lenna は画像圧縮アルゴリズムの評価に広く使用されている標準イメージの 1 つ．