ディジタル署名とパトロールを用いた電子情報改ざん検知方式とWWWへの応用

猪股俊光[†] 板垣晋[††,☆]
曽我正和[†] 西垣正勝[†††]

ディジタル署名とパトロールを基にした電子情報改ざん検知方式を提案する．本方式はシステムの情報管理方法の改善もさりながら，情報そのものに，情報発生の時点から，ある種の改ざん耐性を与えることを狙っている．本方式では，ディジタル署名は電子情報の作成者自身が行い，作成された電子情報とともにディジタル署名はファイルサーバに保存され，周期的なパトロールによって改ざんの有無が検知される．そのため，クラッカーが正規ユーザになりすまし，電子情報を改ざんしたとき，パトロールによって改ざんを検知できるとともに，改ざんありが検知されたあとの処理を迅速かつ柔軟に行うことができる．また，パトロールは，ファイルサーバとは独立に稼働するパトロールサーバによって周期的に行われるため，分散処理によりファイルサーバの負荷が軽減されるとともに，クラッカーの侵入に対する安全性が高い．さらに，この方式に基づきながらディジタル著作物の一種であるホームページの改ざんを検知するための監視システムを設計した．監視システムは，FTP サーバと Web 用サーバを独立させる構成をとっており，これにより，偽のホームページが公開されることを防止している．この監視システムのプロトタイプの実装を通じて所期の動作が得られることを確認した．

A Method of Tamper-proof Using Digital Signature and Patrol, and Its Application to the WWW

Toshimitsu Inomata,[†] Susumu Itagaki,[††,☆] Masakazu Soga[†]
and Masakatsu Nishigaki[†††]

This paper proposes a method for the tamper-proof of the digital contents based on digital signature and patrol. This method aims not only to improve contents-management, but also to have a tamper-proof contents at the time of contents-creation. In this method, the writer of the digital contents makes digital signature for digital contents made in the self. Then, the digital contents and the digital signature are sent to the file server. Afterwards, the existence of the tamper-proof of the digital contents is verified by the periodic patrol by the patrol server. Therefore, it can quickly and flexibly deal with it, after the fact of the tampering is detected. The load of the file server is reduced, because the patrol job is carried out periodically by patrol server who independently works with the file server, and the patrol server is more safe against the intrusion from the cracker. In addition, we developed an actual prototype system for the tamper-proof of Web pages which are a kind of the digital contents. In this system, the Web files are separated FTP server and Web server because the false Web pages never opened to public. The total system was operated properly through an actual implementation and evaluation.

1. はじめに

ディジタル著作物（プログラム，ディジタル作品，ホームページ等）は電子情報として存在するため，インターネットにより瞬時に広範囲に配布可能であり，21 世紀の重要な社会基盤を構成する要素である．すなわち，電子政府と称される政府や自治体の住民への電子サービス機能や電子申請機能，病院や学校のサービスや共同作業の基盤となる電子カルテや電子成績書，あらゆる企業の広告・流通部門を革命的に一新する電

† 岩手県立大学ソフトウェア情報学部
Faculty of Software and Information Science, Iwate Prefectural University
†† 岩手県立大学大学院ソフトウェア情報学研究科
Graduate School of Software and Information Science, Iwate Prefectural University
††† 静岡大学情報学部
Faculty of Information, Shizuoka University
☆ 現在，日本電営株式会社
Presently with Nihon Denei Corporation

子商取引等々，これらすべての分野において電子情報がその基盤となっている．しかし，昨今は電子情報であるがための弱点（コピー容易性，改ざん容易性等）を突く不正行為，嫌がらせ行為，サイバーテロ等の脅威にさらされている．これらの脅威からディジタル著作物を有効に守る技術的な手段を開発することは，安全な情報化社会を築くうえで急を要する課題である．

そこで，本研究では，電子情報の改ざんの有無を検知するための方式を提案するとともに，その方式に基づきながらディジタル著作物の 1 つであるホームページの改ざんを検知する監視システムを設計し，そのプロトタイプの実装を試みた．提案する改ざん検知の方式は，ディジタル署名[1] とパトロールを基にしている[2)~4)]．ディジタル署名は，電子情報が真正であることの証として電子情報作成者が，電子情報のダイジェスト値（一方向性ハッシュ関数[5] により求められたハッシュ値）に対して電子情報作成者の秘密鍵で暗号化することにより行われる．一方，パトロールは，電子情報が保存されているサーバ（たとえば，ファイルサーバ等）上の電子情報が改ざんされたかどうかを周期的にチェックする処理をいい，その処理はサーバに保存されている電子情報のダイジェスト値（チェック時における値）と，ディジタル署名を電子情報作成者の公開鍵により復号して得られるダイジェスト値（作成時における値）が一致するかどうかを調べることにより行われる．このような基本的なアイデアに，安全性や効率性，運用性を考慮にいれた電子情報改ざん検知方式を考案した．

従来，電子情報の改ざん検知については，ハッシュ関数を用いた Tripewire[6] やディジタル署名と電子透かしを用いたインターネット・マーク[7] がある．Tripewire では，サーバが改ざんの有無を検知するが，ディジタル署名を使っておらず，真正な電子情報がサーバに保存されているという保証はない．インターネット・マークは，ホームページの改ざん検知のためのシステムであり，改ざんの有無の検知はホームページの閲覧者が閲覧時に限って行うため，改ざんを検知した後のサーバ側の対応が遅れることになる．本稿で提案する方式では，改ざん検知の対象となるすべての電子情報に作成者自身のディジタル署名を付加したうえでサーバ上に保存することにより，なりすましによる改ざんを防いでいる．さらに，サーバに格納されている電子情報の真正性がパトロールによってチェックされるため，改ざん後の対応が迅速に行える等という特徴を持つ．

さらに，考案した方式を WWW 環境のもとでのホームページ改ざん検知に適用することを通じて，実際の監視システムの開発に必要なシステム要件を明らかにするとともに，そのプロトタイプを実装し，考案した方式が有効であることを確認した．

以下，2 章において提案する電子情報改ざん方式について述べ，3 章では，その方式に基づいてホームページの改ざんを検知する監視システムについて述べる．そして，4 章で本方式の特徴ならびに他の研究との比較を行い，5 章でまとめを述べる．

2. 電子情報改ざん検知方式

2.1 用語とネットワーク環境

電子情報 記憶装置にファイルとして保存可能であって，ディジタル署名可能なデータ．たとえば，プログラム，画像，文書等．具体例として Web ページをとりあげた．本稿では通称に従って Web ページを一括してホームページと称している．

正規ユーザ ファイルサーバに登録済のユーザ．正規ユーザは秘密鍵と公開鍵を持ち，このうち公開鍵はパトロール用のサーバに登録済みであるものとする．

オリジナルデータ 正規ユーザよって作成された電子情報．

改ざん 電子情報の作成者以外のユーザによる電子情報の修正，削除，差し替え操作．

ディジタル署名 電子情報をハッシュ関数により圧縮して得られるハッシュ値を，電子情報の作成者の秘密鍵で暗号化する操作．ディジタル署名によって得られたデータをディジタル署名データという．

本稿では**図 1** に示すような装置からなるネットワーク環境を対象とする．

- ファイルサーバ

  改ざん検知の対象となる電子情報とそれに付随するディジタル署名データが組になって保存されているコンピュータであり，電子情報等の更新履歴情報（ファイルの更新日時やユーザごとのアクセスログ）を記録する．

- パトロールサーバ

  2.4 節で述べる操作手順に従いながらファイルサーバ中の電子情報の改ざんの有無を周期的にチェックするプログラムが実行されているコンピュータであり，正規ユーザの公開鍵が保存されている．また，パトロール中は実行履歴情報（パトロール時刻，判定結果等のログ）を記録する．

なお，本方式に関する過去の発表[3),4)] では，ファイルサーバ内の 1 つのアプリケーションとしてパトロールの機能を実装していたが，ここでは耐攻撃性

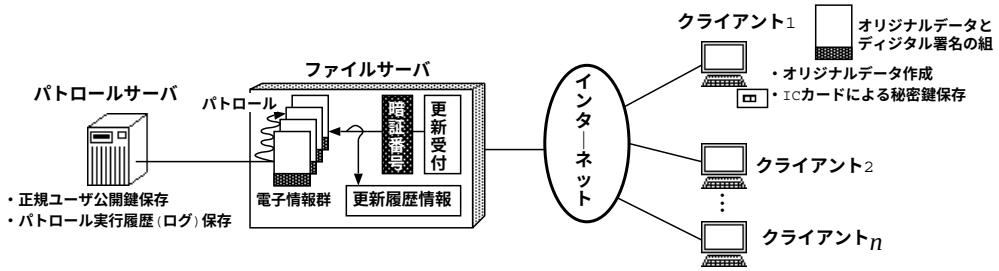

図 1 対象とするネットワークモデル
Fig. 1 Network model.

を向上させる目的で，ファイルサーバとは独立なものとし，インターネットとは別経路でファイルサーバと接続するものとした．

- クライアント

正規ユーザやクラッカーが，ファイルサーバにリモートログインし，2.4 節で述べる操作手順に従いながら電子情報とディジタル署名データをアップロードするために用いられるコンピュータ．

2.2 前提条件

(a) パトロールサーバと正規ユーザは同じハッシュ関数を利用する．

(b) 正規ユーザの秘密鍵は，正規ユーザが持つ非接触型 IC カード内に保管されており，クライアント上でのディジタル署名のときに限って読み取り装置を通じて参照されるものとする．そのため，クライアントがネットワークに常時接続されていたとしても正規ユーザの秘密鍵が盗まれる確率は非常に低い．

(c) 正規ユーザの公開鍵ならびにパトロールの実行履歴情報は，パトロールサーバの特権領域（特定のユーザだけがアクセス可能な領域）に保持される．

(d) 電子情報とディジタル署名データの更新履歴情報は，ファイルサーバの特権領域（特定のユーザだけがアクセス可能な領域）に保持される．なお，この更新履歴情報をパトロールサーバは読み取ることができる．

(e) 正規ユーザが作成したオリジナルデータとディジタル署名データは組として，ファイルサーバ中の正規ユーザの占有領域（いわゆる，ホームディレクトリ）に保存される．この占有領域をパトロールサーバは読み取ることができるが，なりすまされた場合を除いて，その他のユーザはこの占有領域にアクセスできない．

(f) ファイルサーバへのディジタル署名データおよび電子情報のアップロードは，中断されることなく完了する．

2.3 対象とする攻撃法

クラッカーが改ざんする手段としては，ファイルサーバのスーパーユーザの権限を不当に取得する方法と，ファイルサーバの正規ユーザの権限を不当に取得する方法がある☆．通常，危機管理意識が低いユーザのパスワードが露呈し，ユーザの権限が奪われることが多い．そこで，本稿では，クラッカーが正規ユーザになりすまし，ファイルサーバにリモートログインしたのち，オリジナルデータを改ざんあるいは不正コピーをするという攻撃を対象とする．

2.4 提案する方式の基本操作手順

(1) 正規ユーザの操作手順

正規ユーザはクライアント上で次の操作手順に従いながらオリジナルデータをアップロードする．

Step-U01 オリジナルデータを作成する．

Step-U02 ハッシュ関数を用いて，オリジナルデータのハッシュ値を求める．

Step-U03 ハッシュ値を正規ユーザの秘密鍵でディジタル署名し，ディジタル署名データを作る．

Step-U04 オリジナルデータとディジタル署名データの組をファイルサーバにアップロードする．

(2) パトロールサーバの基本的操作手順

パトロールサーバは，ファイルサーバに保存されているパトロール対象の電子情報とディジタル署名データの組（オリジナルデータもしくは改ざんデータのいずれかが考えられる）ごとに基本的には次の操作を周期的に繰り返す．

Step-P01 ハッシュ関数を用いて，パトロール対象の電子情報のハッシュ値 H^S を求める．ただし，パトロール対象の電子情報が存在しない場合には，“改ざん（削除）あり”と判定する．

Step-P02 パトロール対象のディジタル署名データを作成者の公開鍵で復号し，ハッシュ値 H^U を

☆ OS 等のセキュリティホールをつく攻撃は対象からはずす．

求める．ただし，パトロール対象のディジタル署名データが存在しない場合には，“改ざん（削除）あり” と判定する．

Step-P03 2 つのハッシュ値 H^S と H^U を比較し，一致していれば “改ざんなし”，不一致の場合は “改ざん（修正）あり” と判定する．

これら一連の操作が基本となるが，次節以降では性能面や運用面からの改良法について述べる．

（3） 改ざんの検知

本稿では，2.1 節で述べたように，電子情報の作成者以外のユーザによる電子情報の修正，削除，差し替え操作を改ざんと見なしている．もしも，クラッカーがファイルサーバ上の電子情報の一部分を修正したり，他の電子情報と差し替えたりした場合には，改ざん後の電子情報から求められるハッシュ値 H^S と，正規ユーザが作成したディジタル署名データから求められるハッシュ値 H^U が一致しなくなるため，Step-P03 において改ざんが検知される．また，クラッカーがファイルサーバ上の電子情報やディジタル署名データ，あるいはその両方を組として削除した場合には，Step-P01 や Step-P02 においてそのことが検知される．

ただし，クラッカーがファイルサーバ上の最新の電子情報とディジタル署名データを，あらかじめ不正コピーしておいた古いバージョンの電子情報とディジタル署名データに差し替えた場合には，それぞれから求められるハッシュ値が一致するために “改ざんなし” と判定されてしまう．このような差し替えに対する対策は，2.5.3 項で述べる．

2.5 基本操作手順の改良

2.5.1 パトロールの高速化

2.4 節 (2) の操作手順では，パトロールごとに，ディジタル署名データの復号を行うが，正規の更新や改ざんが行われない限りは，復号した結果は毎回同じになる．そこで，Step-P02 によって求められるハッシュ値をパトロールサーバの特権領域に保存しておく．通常は，Step-P02 を省略し，すでに保存してあるハッシュ値を用いて Step-P03 で照合を行う．その結果が不一致であるときに限り，Step-P02 を行ってその時点でのハッシュ値を求めることにする．これにより，Step-P02 の実行回数を必要最小限に減少することができ，パトロールの高速化が期待される．

たとえば，**図 2** において時刻 t_1 のパトロール時に改ざんなしと判定された場合，時刻 t_2 のパトロール時には Step-P02 をやらずに Step-P03 を行っても，その間に改ざんがなされなければ，ハッシュ値は一致する．しかし，時刻 t_3 からの電子情報のアップロードが完了した後のパトロール時刻 t_5 には，Step-P02 をやらずに Step-P03 を行うと，ハッシュ値は一致しない．その場合には，Step-P02 を行って，その時点でのハッシュ値を求めたのち，Step-P03 を行えばよい．

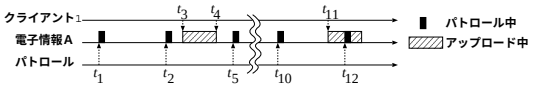

図 2 電子情報の更新とパトロール周期
Fig. 2 An example of patrol.

この改良に対応する操作手順は 2.5.4 項で示す．

2.5.2 パトロールと更新作業の排他制御

ファイルサーバには，パトロールサーバと複数台のクライアントがアクセスする．そのため，たとえば，図 2 の時刻 t_{12} のようにあるユーザが電子情報をアップロードしている最中に，そのユーザの電子情報がパトロール対象になる場合がありうる．このユーザが正規ユーザであったとしても，アップロードが完了しない間は，ハッシュ値が一致しないためにパトロールサーバは “改ざんあり” と判定してしまう．そこで，ディジタル署名データにオリジナルデータがアップロード作業中であるかどうかを判定するためのフラグの役割を持たせることにする．たとえば，図 2 において時刻 t_3 でオリジナルデータをアップロードするときに，特別な内容のディジタル署名データ（これをキーワード署名データとよぶ）を最初にファイルサーバに送っておく．これは，特定のキーワード（パトロールサーバとクライアントの間の合言葉）に対してディジタル署名をすることにより得られるディジタル署名データであり，キーワード署名データとよぶことでディジタル署名データと区別するが，ファイルサーバに保存する際には同名ファイルとする．その後，オリジナルデータのアップロードが完了した時刻 t_4 で，オリジナルデータのハッシュ値から作られたディジタル署名データをファイルサーバに送り，キーワード署名データと置き換える．パトロールサーバはディジタル署名データを復号したときに当該のキーワードが現れたときにはアップロード中であると判定すればよい．さらに，オリジナルデータのアップロードに時間がかかる場合を考慮し，パトロールの判定結果を，“改ざんあり” と “改ざんなし” に “更新中” を加えた 3 種類とし，“更新中” はアップロード中の可能性がある状況をあてはめて，次回以降のパトロール時に改ざんの有無を判定することにする．なお，アップロード中に障害が発生したり，クラッカーによる攻撃が起きたりした場合も考慮し，所定の回数あるいは時間が経過しても “更新

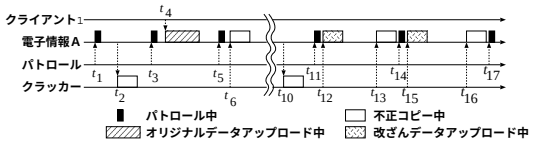

図 3 書き戻し攻撃とサブリミナル攻撃
Fig. 3 An example of attacks.

中” が継続している場合には，“改ざんあり” と判定することとする．

オリジナルデータのアップロードに限らずに，ディジタル署名データのアップロードとパトロールが競合する場合がある．キーワード署名データがアップロード中の場合には，2.5.1 項で述べたように前回求めておいたハッシュ値と，まだ更新されずに保存されている現在のオリジナルデータのハッシュ値との照合を行うので，改ざんの有無を検知できる．一方，新しいディジタル署名データのアップロード中にパトロールが行われた場合には，すでに，オリジナルデータの更新は終わっているために，前回求めておいたハッシュ値と照合しても一致はしない．そして，ディジタル署名データの復号を試みても，アップロードが完了していないために，ハッシュ値を求めることができない．そこで，この場合にも “更新中” と見なし，先ほどと同様に “更新中” の継続回数や時間によって “改ざんあり” と判定することとする．

この改良に対応する操作手順は 2.5.4 項で示す．

2.5.3 提案する方式に固有な攻撃方法

(1) 書き戻し攻撃

図 3 を例にとり説明する．たとえば，クラッカーがある正規ユーザへのなりすましに成功し，時刻 t_2 にそのオリジナルデータとディジタル署名データの不正コピーを始めたとする．そして，時刻 t_4 に当該正規ユーザがオリジナルデータの更新を始めたのちの時刻 t_6 にクラッカーが不正コピーしておいたオリジナルデータとディジタル署名データ（これらは古いバージョン）をともに書き戻せば，その後のパトロール時にはオリジナルデータとディジタル署名データ，それぞれから求められるハッシュ値が一致するために “改ざんなし” と判定されてしまう．このように新しいバージョンを古いバーションに差し替えてしまうという攻撃を書き戻し攻撃とよぶことにする．

(2) サブリミナル攻撃

パトロール方式では，パトロール時刻のときに保存されている電子情報の真正をチェックしており，パトロールとパトロールの期間の真正はチェックしていない．そのため，たとえば，クラッカーが，図 3 の時刻 t_{10} にオリジナルデータを不正コピーしたのち，パトロール直後の時刻 t_{12} にクラッカーが作成した改ざんデータに差し替えて，パトロールの直前の時刻 t_{13} にオリジナルデータに書き戻すことを繰り返せば，パトロール時には “改ざんなし” と判定されているにもかかわらず，一定間隔で改ざんデータがファイルサーバに保存されることになる．このようにパトロール時刻を避けるような攻撃法をサブリミナル攻撃とよぶことにする．

(3) 固有な攻撃方法の対策

書き戻し攻撃では，最新のバージョンの電子情報とディジタル署名データがそれぞれ古いバージョンのものに差し替えられる．そのため，ファイルサーバ上の電子情報とディジタル署名データそれぞれから求められるハッシュ値が一致したときに，ファイルサーバ上の電子情報が古いバージョンであるかどうかを調べれば書き戻し攻撃による改ざんを検知することができる．そこで，正規ユーザがファイルサーバにアップロードするディジタル署名データに，オリジナルデータのハッシュ値に加えてオリジナルデータを作成した時刻（ユーザタイムスタンプとよぶ）も含めることとする．そして，パトロールサーバは “改ざんなし” と判定した際にディジタル署名ファイルを復号して得られるユーザタイムスタンプを記録しておき，パトロール時にファイルサーバ上のディジタル署名データから得られるユーザタイムスタンプと比較し，ファイルサーバ上のタイムスタンプの方が古い場合には，書き戻し攻撃があったと判定する．

また，サブリミナル攻撃では，最新のバージョンの電子情報がパトロールの直前にクラッカーによって改ざんデータと差し替えられるため，正規ユーザが電子情報の更新をしていないにもかかわらず，ファイルサーバ上では電子情報の更新が行われている．そのため，ハッシュ値が一致していたときに，前回のパトロール時以降にファイルサーバ上の電子情報の更新が行われたかどうかを調べれば，サブリミナル攻撃による改ざんを検知することができる．そこで，ファイルサーバ上で電子情報が更新された時刻（サーバタイムスタンプとよぶ）をパトロールサーバが更新履歴として保存しておき，パトロールのたびに電子情報が更新されているかどうかを判定ればよい．

このような対策に対応する操作手順を 2.5.4 項で示す．

2.5.4 改良後の操作手順

2.5.1〜2.5.3 項で述べた改良点を取り入れた操作手順を以下に示す．ここで，ユーザタイムスタンプは，

正規ユーザがオリジナルデータを作成したときのクライアントの内部クロックの時刻とする．

(1) 正規ユーザの操作手順（改良版）

次の操作手順に従いながらオリジナルデータをアップロードする．

Step-U1 オリジナルデータを作成する．

Step-U2 ハッシュ関数を用いて，オリジナルデータのハッシュ値を求める．

Step-U3 ハッシュ値とユーザタイムスタンプを，正規ユーザの秘密鍵でディジタル署名し，1 つのディジタル署名データを作る．

Step-U4 特定のキーワードとユーザタイムスタンプからなるキーワード署名データを作成し，それをファイルサーバにアップロードする．

Step-U5 オリジナルデータをファイルサーバにアップロードする．

Step-U6 ディジタル署名データをファイルサーバにアップロードし，キーワード署名データと差し替える．

(2) パトロールサーバの基本的操作手順（改良版）

パトロール対象の電子情報ごとに次の操作を周期的に繰り返す．以下では，パトロールサーバの実行履歴情報（電子情報ごとに，パトロール時刻，判定結果，ハッシュ値，ユーザタイムスタンプ，サーバタイムスタンプが記録されている）を単にログとよぶ．また，次のような表記を用いる．下付の添字 i はパトロール回数を表し，たとえば，今回が i 回目であるとき，$i-1$ は 1 回前のパトロールを表す．

H_i^U ディジタル署名データに含まれているハッシュ値

T_i^U ディジタル署名データに含まれているユーザタイムスタンプ（クライアント上での作成時刻）

T_i^S ファイルサーバの更新履歴情報から取得されたサーバタイムスタンプ（ファイルサーバ上にアップロードされた時刻）

H_i^S ファイルサーバ上に保存された電子情報から計算されたハッシュ値

R_i 判定結果

H_{i-1}^U ログから取得された前回パトロール時刻でのハッシュ値（ディジタル署名データから得られた値）

T_{i-1}^U ログから取得された前回パトロール時刻でのユーザタイムスタンプ

T_{i-1}^S ログから取得された前回パトロール時刻でのサーバタイムスタンプ

R_{i-1} ログから取得された前回パトロール時刻での判定結果

第 i 回目のパトロール手順：

Step-P1 ハッシュ関数を用いて，パトロール対象とする電子情報のハッシュ値 H_i^S を求める．ただし，パトロール対象の電子情報が存在しない場合には，“改ざん（削除）あり” と判定する．

Step-P2 ログ中に前回パトロール時刻でのハッシュ値等が保存されているのであれば，ハッシュ値 H_{i-1}^U，ユーザタイムスタンプ T_{i-1}^U，サーバタイムスタンプ T_{i-1}^S，判定結果 R_{i-1} を取得し，Step-P3 へ．それ以外の場合は，Step-P3-2 へ．

Step-P3 2 つのハッシュ値 H_i^S と H_{i-1}^U を比較し，一致していた場合は Step-P3-1 へ．不一致の場合は Step-P3-2 へ．

Step-P3-1 ファイルサーバの更新履歴情報より当該電子情報の最終更新時刻 T_i^S を求め，当該電子情報の T_{i-1}^S と比較し，一致していれば判定結果 R_i を “改ざんなし” とし，そうでなければ判定結果 R_i を “改ざん（サブリミナル攻撃）あり” とする．いずれの場合も，H_{i-1}^U，T_{i-1}^U，T_i^S，R_i をログに書き込む☆．

Step-P3-2 ディジタル署名データを作成者の公開鍵で復号し，ハッシュ値 H_i^U ならびにユーザタイムスタンプ T_i^U を求める．ただし，対象とするディジタル署名データが存在しない場合（削除されている）には，Step-P3-5 へ．ハッシュ値 H_i^S と H_i^U が一致した場合は，Step-P3-3 へ．復号した際に特定のキーワードが現れた場合（キーワード署名データである）またはハッシュ値が求められない場合（ディジタル署名データがアップロード中である）には，Step-P3-4 へ．一致しない場合（修正されている）には，Step-P3-5 へ．

Step-P3-3 ユーザタイムスタンプ T_i^U とログ中のユーザタイムスタンプ T_{i-1}^U を比較する．T_i^U の方が過去（書き戻し攻撃）であれば，Step-P3-5 へ．そうでなければ，Step-P3-6 へ．

Step-P3-4 アップロード中かどうかを考慮し，前回の判定結果 R_{i-1} が “更新中” 以外であった場合には，判定結果 R_i を

☆ 復号を省略したため，ディジタル署名データに含まれているハッシュ値とユーザタイムスタンプは前回と同じ値を書き込む．

“更新中” とし，Step-P3-7 へ．“更新中” だった場合には，ログより，更新中の判定が出てからの経過時間を算出し，あらかじめ決めていた期間を超過していたら Step-P3-5 へ．そうでなければ，Step-P3-7 へ．

Step-P3-5 判定結果 R_i を “改ざんあり” とし，Step-P3-7 へ．

Step-P3-6 判定結果 R_i を “改ざんなし” とし，Step-P3-7 へ．

Step-P3-7 ファイルサーバの更新履歴情報より，T_i^S を求めたのち，H_i^U，T_i^U，T_i^S，R_i をログに書き込む．

(3) 固有な攻撃法による改ざんの検知

2.2 節の前提条件 (c) と (d) で述べたように，H_{i-1}^U や T_{i-1}^U 等の実行履歴情報はパトロールサーバの特権領域に保持され，電子情報等の更新履歴情報はファイルサーバの特権領域に保持される．これにより，これらの情報はクラッカーにより改ざんされないという前提のもとで，固有な攻撃法による改ざんの検知は次のようにして行われる．

書き戻し攻撃が行われた場合，前回のパトロール以降に，クラッカーによって古いバージョンの電子情報とディジタル署名データがファイルサーバ上にアップロードされているため，Step-P3 における H_i^S と H_{i-1}^U の比較は不一致になるものの Step-P3-2 における H_i^S と H_i^U の比較では一致する．Step-P3-3 におけるユーザタイムスタンプ T_i^U と T_{i-1}^U の比較では，現在のファイルサーバ上の電子情報は古いバージョンであるために，T_i^U の方が過去であり，書き戻し攻撃が行われたことが検知される．

一方，サブリミナル攻撃が行われた場合には，パトロールの直前にクラッカーによって前回のパトロールと同じ最新のバージョンの電子情報がファイルサーバ上にアップロードされるため，Step-P3 における H_i^S と H_{i-1}^U の比較は一致するものの，次の Step-P3-1 において T_i^S と T_{i-1}^S は一致しない．このことからサブリミナル攻撃が行われたことが検知される．

3. ホームページ改ざん監視への応用

3.1 対象とする WWW

(1) 対象とする電子情報の種類

あらかじめディジタル署名された HTML ファイルや画像ファイルを対象とする（これを HP ファイルとよぶ）．ただし，たとえば，CGI の利用等により，作成者以外によって書き換えられる電子情報を閲覧時に動的に引用して作られるようなホームページ内の引用部分は対象外とする．

(2) HP ファイルの更新方法

HP ファイルはホームページの作成者がイントラネットあるいはインターネットを通じてファイルサーバにアップロードするものとする．たとえば，自前で Web サーバを保持している企業等のような組織において，ホームページ作成者がイントラネットを通じてアップロードする場合や，一般のプロバイダやホームページ作成を外注しているような企業等において，ホームページ作成者がインターネットを通じてアップロードする場合がこれにあたる．

(3) 想定される攻撃

ファイルサーバ経由のなりすましによる HP ファイルの改ざんを対象とし，クライアントのブラウザ経由の攻撃は対象としない．

3.2 改ざん監視システム

3.2.1 監視システムの要件

電子情報が，HP ファイルであることから，新たな前提条件が必要となるとともに，2 章で述べた各構成要素の機能や操作手順を拡張する必要がある．それらを以下に示す．

(a) ディレクトリ単位で改ざんを検知する

1 つのホームページは複数個の HP ファイルから構成されている．そこで，あるホームページに必要とされるすべての HP ファイルは，HP ファイル作成者の所有するディレクトリに格納されているものとする．このディレクトリをホームディレクトリとよぶ．

(b) 複数人によって 1 つのホームページが作成される

たとえば，会社等の組織のホームページはグループで作成されることがある．そのため，1 つのホームディレクトリに対して複数人のユーザがアクセス可能であるものとする．ただし，同一グループの複数人のユーザは同時にログインすることはできないものとする．

(c) 改ざんページが公開されることを防ぐ

2.5.3 項で述べたように，一定間隔ごとに改ざん有無を検知するため，その時刻がくるまでは改ざんされた電子情報（この場合には偽ホームページ等）が公開されることになる．たとえ短時間であっても偽のホームページ等が公開されることは作成者にとって避けたい事態であると考えられる．

(d) “改ざんあり” 検知後の対応処理を行う

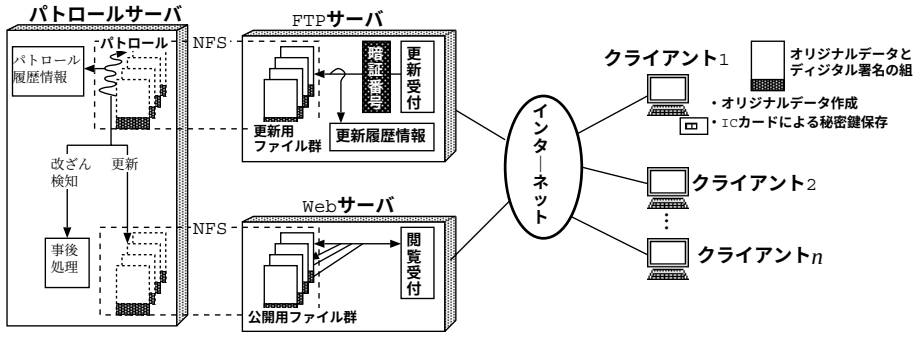

図 4 ホームページ改ざん検知システムの構成
Fig. 4 Prototype system.

“改ざんあり” と判定された後で行うべき処理を事後処理とよぶ．具体的には，HP ファイルの作成者への通知，改ざんデータの記録等である．

3.2.2 システムの構成要素

3.2.1 項の要件 (c) に対応するために，HP ファイルの更新窓口となって，最新の HP ファイルを保存するためのサーバと，公開用の HP ファイルを保存するサーバを別々にすることとした．また，3.2.1 項の要件 (d) の事後処理の機能はパトロールサーバに持たせることにした．**図 4** にこれらに対応したシステム構成を示す．なお，図中のインターネットにはイントラネットも含む．

- FTP サーバ
  2.1 節で述べたファイルサーバの役割を果たす．監視対象となるディレクトリ内に存在する HP ファイルやディジタル署名データの集まりを更新用ファイル群とよぶ．
- パトロールサーバ
  2.1 節で述べたパトロールサーバの役割に加えて，“改ざんなし” と判定された HP ファイルを Web サーバに転送する機能を持つ．これにより，つねに “改ざんなし” と判定された HP ファイルだけが Web サーバ上に保存されて，公開の対象となるため，改ざんデータが公開されることはない．また，“改ざんあり” と判定されたときには，事後処理を行う．このようなパトロールのための操作手順は 3.2.3 項で述べる．なお，パトロールサーバは，FTP サーバ上のファイル群を読み取れるような NFS の設定で FTP サーバとファイルシステムを共有するとともに，Web サーバ上に公開用の HP ファイル等を書き込めるような NFS の設定で Web サーバとファイルシステムを共有する．パトロールサーバは，実行履歴情報（ログ）として，監視対象のディレクトリごとに，パトロール時刻，判定結果，ハッシュ値，ユーザタイムスタンプ，サーバタイムスタンプを記録する．
- Web サーバ
  公開用のオリジナルデータを保存するコンピュータであって，保存されている公開用の HP ファイルの集まりを公開用ファイル群とよぶ．また，HTTP のみが利用可能であって，FTP，Telnet 等を利用してのログインはできない設定とする．
- クライアント
  2.1 節で述べたクライアントの役割に加えて，ホームページ閲覧のためにも用いられる．HP ファイルのアップロードの操作手順は 3.2.3 項で述べる．

3.2.3 ホームページ改ざん監視システムの操作手順

(1) HP データ作成者の操作手順

3.2.1 項の要件 (a) に対応するために，2.5.4 項 (1) の Step-U2 では，HP ファイルごとではなくて，ホームディレクトリに含まれている HP ファイル群を 1 つのオリジナルデータと見なし，ハッシュ関数の対象とするとともに，ホームディレクトリに含まれている HP ファイル群をアップロードする．そのために，次のように修正する．

- Step-U2 を次の Step-U2-1 と Step-U2-2 に分けて，順に実行する．
  Step-U2-1 対象とするホームディレクトリ内に含まれているすべての HP ファイルの属性（ファイル名，ホームディレクトリからの相対パス名）からなるファイルリストを作成する．
  Step-U2-2 ファイルリスト中の各 HP ファイルに対して，i) ハッシュ値の計算，ii) ハッシュ値と相対パス名の組の生成，を繰り返し，こうして得られたすべての組からなる

データを 1 つのオリジナルデータと見なし，ハッシュ値を計算する．このハッシュ値をエッセンシャルハッシュとよぶ．

- Step-U3 では，エッセンシャルハッシュとユーザタイムスタンプを，正規ユーザの秘密鍵でディジタル署名し，1 つのディジタル署名データを作る．
- Step-U5 のオリジナルデータをホームディレクトリに含まれている HP ファイル群とする．

(2) HP パトロールサーバの操作手順

3.2.1 項の要件 (b) に対応するために，全正規ユーザをグループに分割し，グループに対して 1 つのホームディレクトリを割り当てることにする．そのため，パトロール対象は各グループのホームディレクトリとなる．また，3.2.1 項の要件 (a)〜(d) に対応するために，2.5.4 項 (2) の操作手順を次のように修正する．

- 各 Step における電子情報を，ホームディレクトリ内の全ファイルとし，電子情報のハッシュ値をエッセンシャルハッシュとする．
- 復号のときには，更新履歴情報から当該ディレクトリをアップロードしたばかりのユーザ名を調べ，その公開鍵を用いる．
- Step-P3-5 で “改ざんあり” と判定した後は，事後処理を行う．
- 正規ユーザによる更新が行われた直後のパトロールによって，“改ざんなし” と判定されたときには，Web サーバに当該ホームディレクトリをコピーする必要がある．そのために，Step-P3-3 における,「そうでなければ，Step-P3-6 へ」を,「そうでなければ，Web サーバへ当該ホームディレクトリの HP ファイルをコピーしたのち，Step-P3-6 へ」に変更する．

3.3 実 装 例

3.2.3 項で述べた各操作手順をクライアントとパトロールサーバ，それぞれで実行できるように以下のアプリケーションを作成した．

- クライアントアプリケーション
  - 対象 OS は Windows 2000/XP.
  - ハッシュ関数は MD5.
  - X509 規格のフォーマットで IC カードに記録されている秘密鍵を使用.
  - 3.2.3 項 (1) の操作手順，ならびにそれに対応する GUI の実装.
- パトロールサーバ用アプリケーション
  - 対象 OS は Linux.
  - ハッシュ関数は MD5.
  - 3.2.3 項 (2) の操作手順の実装.

表 1 パトロールサーバにおける平均パトロール時間
Table 1 The average time of the processing at the patrol server.

ディレクトリ構成（ファイル個数一定）	平均時間	ディレクトリ構成（ファイル容量一定）	平均時間
64 KB × 64 個	0.3 秒	64 KB × 64 個	0.3 秒
256 KB × 64 個	1.3 秒	64 KB × 256 個	1.4 秒
512 KB × 64 個	2.1 秒	64 KB × 512 個	2.4 秒
1,024 KB × 64 個	4.0 秒	64 KB × 1,024 個	5.1 秒
2,048 KB × 64 個	7.4 秒	64 KB × 2,048 個	10.6 秒

このうち，次の機能は別プロセスとして実行するようにした☆.

- ∗ 事後処理機能（Step-P3-5 に対応）
  “改ざんあり” の場合，そのことを HP ファイル作成者と Web サーバ管理者へ通知する．改ざん HP ファイルを証拠として保存する．
- ∗ 更新された HP ファイルの Web サーバへの転送機能（Step-P3-6 に対応）

3.4 性 能 評 価

(1) パトロールサーバアプリケーションの改ざん検知時間

ともに 100BASE-TX のイーサネットを持つパトロールサーバ（CPU 1 GHz，128 MB）と FTP サーバ（CPU 1 GHz，128 MB）が HUB によって結合され，パトロールサーバが FTP サーバ上のホームディレクトリを NFS を利用しながら読み込むという環境のもとで，10 種類のホームディレクトリ（ディレクトリ中のファイル個数や 1 個あたりのファイル容量を変化させた）について，1 回のパトロール時間を計測した結果（10 回計測した平均時間）を**表 1** に示す．なお，1 回のパトロール時間には，ディジタル署名データ（サイズは 1,024 ビット）の復号にかかる時間（約 1.8 ミリ秒）も含まれている．

この結果からは，ホームディレクトリの総容量が同じであっても，ファイル個数が多くなるほど時間がかかることが分かる．

(2) クライアントアプリケーションのディジタル署名データ作成時間

ノート型 PC（CPU 1 GHz，128 MB）をクライアントとし，内蔵ハードディスク中のホームディレクトリ（ファイル構成は表 1 と同様の 10 種類）について，ディジタル署名データの作成時間を計測した結果，表 1 の実行時間とほとんど差がなかった．これは，ハッシュ値を求める処理が両アプリケーショ

☆ アプリケーションを親プロセスとし，2 つの機能を fork() による子プロセスとして実現した．

ンの計算時間の大半を占めるからだと思われる．また，高性能 PC（CPU 2.2 GHz，256 MB）を用いて，同様の計測を行ったところ，CPU の性能よりもハードディスクのアクセス速度の差の影響が大きいことが分かった．

(3) パトロール間隔の決定

表 1 の結果をもとに，パトロール対象のユーザ数や各ユーザのディレクトリサイズに応じたパトロール間隔を算出することができる．たとえば，パトロール対象のユーザ数が 30 であるプロバイダにおいて，1 ユーザのディレクトリサイズを最大約 4 MB と最大約 16 MB にした場合，それぞれ 9 秒（=0.3 秒× 30）と 42 秒（=1.4 秒× 30）間隔でパトロールが可能であることが分かる．また，2.5.1 項で述べたようにホームページの更新や改ざんが行われない間は，ディジタル署名データの復号を省略できるため，たとえば，ユーザ数 30 に対するパトロール間隔が 9 秒であるとき，1 時間では最大 12,000 回の復号を省略することが可能であり，パトロールサーバの処理時間は約 22 秒少なくなる．

パトロール間隔は，正規ユーザが新しい HP ファイルを FTP サーバにアップロードしてから公開されるまでの時間，あるいは，クラッカーが改ざんを行ってからそれが検知されるまでの時間に相当しており，ディレクトリサイズが最大 16 MB の場合であってもパトロール間隔を 1 分以内とすることが可能であることから，本稿で提案する方式は有効であると思われる．

4. 考　　察

本稿で提案した方式の特徴を述べながら他の方式との比較を述べる．

(a) ディジタル署名による真正性の検証

本稿では，作成者自身によってディジタル署名されたオリジナルデータをファイルサーバにアップロードしたのち，そのファイルサーバ上のオリジナルデータが改ざんされたかどうかをチェックする方式を提案した．

電子情報の 1 つであるホームページの真正性をディジタル署名を用いて検証する方式にインターネット・マーク[7] がある．それは次のような方式である：① ホームページ作成者がインターネット・マーク発行申請をする，② インターネット・マーク発行者が，発行申請のあったホームページから特徴となるデータを取り出し，それらのデータにディジタル署名をしたのち，ディジタル署名や電子証明書等を素材画像に透かし込んでインターネット・マークを作る，③ それをホームページ作成者がホームページの隅の方に張り付けて公開する，④ 閲覧者が閲覧時に，ホームページの真正性を検証する．

この方式と本方式とは，誰がディジタル署名をするのか，誰が改ざんを検知するのか，という点で異なる．本方式では，電子情報の作成者自身がディジタル署名をするが，インターネット・マークでは，インターネット・マーク発行者が行う．今後ますます PKI が整備され，IC カード等が普及されることを考慮すれば，作成者が持つ秘密鍵によってディジタル署名することの方が安全であると考えられる．さらに，正規ユーザは新しいホームページを作成しさえすれば，特定のマークの発行といった手順を必要とせずに，ファイルサーバにただちにアップロードすることができる．また，インターネット・マークでは，改ざんを検知するのは閲覧者であって，改ざんの発見が閲覧者任せであるため，改ざんを検知するのが遅れるばかりでなく，改ざんが検知されてからの処理も遅れてしまう．これに対して，本方式では電子情報（ホームページ）を保存しているサーバ側で改ざんの有無を検知しており，改ざんを検知した後の対応処理をただちに行える．

(b) パトロールによる電子情報の定期的なチェック

改ざんチェックのタイミングには，電子情報へのアクセス要求があったときと，あらかじめ決められた一定間隔ごとの 2 種類があるが，本方式は後者である．前者の方式の 1 つに Tripwire for Web Pages[6] がある．

Tripwire は，対象とするすべての HP ファイルのハッシュ値を求め，それらをデータベースとして保管する．そして，閲覧要求があれば，その HP ファイルのハッシュ値がデータベース上のハッシュ値と一致するかどうかにより改ざんの有無を検知する．このように，Tripwire では，サーバ自身が改ざんの有無の検知を閲覧要求のたびに行っており，改ざんされた HP ページが公開されることを防ぐことができるが，閲覧に要するオーバヘッドが問題となる．また，ディジタル署名技術を使っていないので，正規ユーザになりすまされた場合，サーバもだまされてそれとは知らずに偽ホームページのハッシュ値を生成してしまう．これに対して，本方式では，ディジタル署名と照合するので，なりすまし侵入による改ざんは検知可能であり，別マシンであるパトロールサーバがある時間間隔をおきながらチェックを繰り返すので，ファイルサーバにかかる負荷は少なくてすみ，ファイルサーバ本来の処理に与える影響がほとんどない．さらに，閲覧要求がなくても周期的にチェックするために，閲覧要求がく

る前に改ざんを検知して対処可能である．また，以下の (d) でも述べるように更新用サーバと公開用サーバによる分散処理により，改ざんデータが公開されることを防いでいる．

(c) ファイルサーバとパトロールサーバの分散処理

本稿で提案するパトロール処理を，文献 6) のようにパトロールアプリケーションとしてファイルサーバ上で稼働させることもできる．しかし，その場合には，クラッカーによるパトロールアプリケーションへの攻撃を考慮する必要がある．そのため，本稿ではパトロールアプリケーションをファイルサーバとは別のマシン上で稼働することにした．さらに，ファイル更新処理とパトロール処理の分散処理が行われるため性能向上が実現されている．

もしも，クラッカーがファイルサーバに侵入し，電子情報やディジタル署名データを改ざんしたとしても，秘密鍵を持たないクラッカーは改ざんした電子情報に対応する正しいディジタル署名データを作成することができず，改ざんが発見される．また，パトロールサーバとファイルサーバの間はインターネットとは別経由でネットワーク接続されるため，パトロールサーバへの直接的な侵入を防ぐことが可能である．ただし，ファイルサーバに侵入後に，更新履歴情報が改ざんされて，パトロール対象の電子情報が更新されたのかどうかが判定できなくなった場合は，サブリミナル攻撃と特別な書き戻し攻撃☆による改ざんの有無が検知できなくなる．そのための対策としては，更新履歴情報にディジタル署名をかける等して改ざんの防止をすることが考えられる．

Web サーバに対する攻撃法として DNS エントリを書き換える手法[11] がある．これについて，文献 12) では，監視システムもまたクラッカーと同様のインターネットを利用して Web サーバへリモート接続することによって，この攻撃を検知する方法が提案されている．本方式においては，パトロールサーバの安全性の面から別のネットワークによる接続とする監視システムを設計したが，インターネット経由のリモート接続によるパトロールは，特別な修正を必要とせずに可能である．また，リモート監視のためのパトロールサーバを追加で導入することも可能である．

(d) 分散処理によるホームページ監視システムの設計

従来の方式[7),12)] では，一時的にせよ改ざんされたホームページが閲覧されることになる．たとえ，短時間であったにせよ偽のホームページが公開されることは当該サイトの信用を失うことになる．そこで，3 章では，更新処理と公開処理をそれぞれ FTP サーバと Web サーバにまかせて分散処理させることにした．これにより，FTP サーバ上の更新用ファイル群の中でも “改ざんなし” と判定された HP ファイル群だけが Web サーバ上の公開用ファイル群となるため改ざんされたホームページが公開されることが防止される．

この構成では，Web サーバは公開処理，パトロールサーバは改ざん検知と Web サーバへの公開ファイル群のコピーをそれぞれ行っている．そのため，指定された日時に新しいホームページを公開するというサービスを行う場合には，次のようにすればよい．FTP サーバへの公開用ファイル群のアップロードの際に，Web サーバ上での公開予定日時を指定することにしておく．そして，パトロールサーバが “改ざんなし” と判定し，かつ公開日時になったときに，当該 HP ファイル群を Web サーバにコピーする．

(e) プロトタイプの利便性と移植性

3.3 節で述べたように，本稿では，3.2.3 項 (1) の操作手順に基づいたクライアントアプリケーションと，同じく 3.2.3 項 (2) の操作手順に基づいたパトロールサーバ用アプリケーションを実装した．クライアントアプリケーションを利用すれば，正規ユーザは秘密鍵の提示とアップロード対象の HP ファイル群を指定するだけで，一連の操作が自動的に行われるので，複雑な操作は必要ない．クライアントアプリケーションは，普及度を考慮し，現在のところ Windows 上のアプリケーションとして実装したが，他の OS 上に移植可能である．

また，図 4 のように，既存の FTP サーバや Web サーバからなるシステム構成に，パトロールサーバを追加する方式であり，既存のサーバ上に新たなアプリケーションをインストールしたり，OS やアプリケーションを修正したりするといった作業が必要ない．

5. おわりに

本稿では，ディジタル署名とパトロールを基にした電子情報改ざん検知方式を提案した．対象とする攻撃法は，クラッカーが正規ユーザになりすまし，ファイルサーバにリモートログインしたのち，オリジナルデータを改ざんする方法である．本方式では，ディジタル署名は電子情報の作成者自身が行い，作成された電子情報とともにディジタル署名はファイルサーバに保存され，周期的なパトロールによって改ざんの有無

☆ 図 3 において，t_4 の正規ユーザのアップロードが完了してから次回のパトロール開始時刻 t_5 までの間に書き戻しが行われるような場合．

が検知される．そのため，“改ざんあり” が検知されたあとの処理を迅速かつ柔軟に行うことができる．また，パトロールは，ファイルサーバとは独立に稼働するパトロールサーバによって周期的に行われるため，分散処理によりファイルサーバの負荷が軽減されるとともに，クラッカーの侵入に対する安全性が高い．

さらに，この方式に基づきながらディジタル著作物の一種であるホームページの改ざんを検知するための監視システムを設計した．監視システムは，FTP サーバと Web 用サーバを独立させる構成をとっており，これにより，偽のホームページが公開されることを防止している．この監視システムのプロトタイプの実装を通じて所期の動作が得られることを確認した．本稿では，OS 等のセキュリティホールをつく攻撃等により，FTP サーバや Web サーバのシステム管理者の権限が奪われるケースを想定しておらず，仮に，DNS エントリが書き換えられた場合には，閲覧者が閲覧するホームページの真正性を保証しきれない．しかし，提案する方式は，一定の時間間隔をおきながらではあるが，FTP サーバ内と Web サーバ内のホームページの真正性をパトロールによって保証する．一般に，この種のセキュリティをトータルに保証するためには，いくつかの異る角度からの防護方式をうまく組み合わせて使う必要があると考えられ，本方式はその 1 つの材料となるものと思われる．たとえば，閲覧要求があったときに，Web サーバが HP データとともに作成者のディジタル署名を閲覧者のブラウザに送るように仕様を拡張すれば，文献 7) で行われているように閲覧者が閲覧時に真正性をチェックすることが可能となる．

なお，本稿では，WWW への応用について述べたが，提案した方式は，各種データベースの改ざんチェックにも応用できると期待される．

最後に，今後の課題として以下のことがあげられる．

- 短い周期でパトロールを行うほど，改ざん検知の時間的分解能は向上するが，その分，ファイルサーバやパトロールサーバに負荷がかかることになる．そのために最適なパトロール間隔の導出法の考案．
- CGI の利用等により，電子情報が直接書き換えられる場合がある．このように動的に変更するような電子情報を含むホームページへの対応．
- HP ファイル作成者がファイルの一部分を修正した場合であっても，ディレクトリ全体のエッセンシャルハッシュを求め直す必要があることへの対応．
- DNS エントリの書き換え等のような Web サーバへの攻撃の対応策．

参 考 文 献

1) Goldwasser, S., Micali, S. and Rivest, R.: A Digital Signature Scheme against Adaptive Chosen Message Attack, *SIAM Journal on Computing*, Vol.17, No.2, pp.281–308 (1998).
2) 特願 2000-13211 および特開 2001-202288「電子情報パトロール装置及び電子情報改竄パトロール方法」.
3) 可部，曽我，西垣，田窪：ホームページ改竄パトロール方式，情報処理学会研究報告，2000-CSEC-8-30, pp.173–178 (2000).
4) 板垣，曽我，西垣，田窪：強化型ホームページ改竄パトロール方式，コンピュータセキュリティシンポジウム 2001 (CSS2001)，pp.403–408 (2001).
5) Zhao, J. and Koch, E.: Embedding robust labels into images for copyright protection, *Proc. ICIPR* (1995).
6) トリップワイヤ・ジャパン株式会社：Tripwire for Web Pages. http://www.tripwire.co.jp/
7) 洲崎誠一ほか：Web サイトの真正性を確認可能とするインターネット・マークの提案，情報処理学会論文誌，Vol.41, No.8, pp.2198–2207 (2000).
8) Menezes, A.J., van Oorschot, P. and Vanstone, S.A.: *Handbook of Applied Cryptography*, CRC Press (1996).
9) Rivest, R., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. ACM*, Vol.21, No.2, pp.120–126 (1978).
10) Rivest, R.: The MD5 Message-Digest Algorithm (1992). ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt
11) IPA/ISEC：小規模サイト管理者向けセキュリティ対策マニュアル (2001). http://www.ipa.go.jp/security/fy12/contents/crack/soho/soho/chap1/webcrack.html
12) 竹森敬祐ほか：Web サーバリモート監視システムの実装および評価，情報処理学会論文誌，Vol.43, No.8, pp.2542–2551 (2002).

(平成 14 年 12 月 4 日受付)
(平成 15 年 6 月 3 日採録)

猪股　俊光（正会員）

昭和 59 年豊橋技術科学大学工学部卒業．平成元年同大学大学院博士後期課程修了．同年同大学工学部助手．平成 4 年静岡理工科大学理工学部講師．現在，岩手県立大学ソフトウェア情報学部助教授．工学博士．ディジタル著作物の保護，分散システムの高信頼化，並列システムの仕様記述とその実現に関する研究に従事．電子情報通信学会，ソフトウェア科学会各会員．

板垣　　晋

平成 13 年静岡大学情報学部情報科学科卒業．平成 15 年岩手県立大学大学院博士前期課程修了．同年，日本電営株式会社入社．現在，三菱電機情報技術総合研究所に出向中．主な研究テーマはホームページの改竄検知システム．現在，Web コンテンツの不正利用防止技術に関する研究・開発に従事．

曽我　正和（正会員）

昭和 33 年京都大学工学部電子工学科卒業．昭和 35 年同大学大学院修士課程修了．昭和 35 年〜平成 8 年三菱電機，計算機製作所副所長，情報電子研究所所長を経て平成 8 年静岡大学情報学部教授，平成 11 年岩手県立大学ソフトウェア情報学部教授，現在に至る．博士（工学）（東京大学）．汎用計算機，制御用計算機，制御用システムの開発．フォールトトレラントシステム，セキュリティシステムに関する研究に従事．IEEE，電子情報通信学会各会員．

西垣　正勝（正会員）

平成 2 年静岡大学工学部光電機械工学科卒業．平成 4 年同大学大学院修士課程修了．平成 7 年同大学院博士課程修了．日本学術振興会特別研究員（PD）を経て，平成 8 年静岡大学情報学部助手．平成 11 年同講師，平成 13 年同助教授．博士（工学）．情報セキュリティ，ニューラルネットワーク，回路シミュレーション等に関する研究に従事．