L-009

改ざん箇所の検出が可能な電子文書のデータ構造

Data structure of electronic document that can detect falsified parts

中村 勇介† 汐崎 陽† 岩田 基† 荻原 昭夫†
Yusuke Nakamura Akira Shiozaki Motoi Iwata Akio Ogihara

1. はじめに

電子文書はコンピュータの文書作成ソフトを用いて簡単に作成できる. 電子文書は, 複製, 編集, 配布, 保管が容易であるため, 一般に広く普及している. その一方で, 電子文書の一部が改ざんされたとしても, その痕跡が残らないため, 電子文書の原本性 (改ざんされていないこと, 正真性) の証明, 保証が必要な分野において, 電子文書が改ざんされているか否かの判断を可能とする技術が求められている.

原本性を証明する手段としては電子署名による方法が知られている. 署名者は秘密鍵で署名の作成をして電子文書に添付し, 鑑定者は秘密鍵に対応する公開鍵で署名を検証することにより改ざんの有無を確認できる. しかし, 電子署名では電子文書の改ざんの有無は検出できるが, どの文字が改ざんされたかは特定することができない. そこで本稿では改ざん箇所およびその改ざんの種類が特定できる電子文書のデータ構造を提案する.

2. 電子文書のデータ構造

本手法ではテキスト形式の電子文書を対象とする. また, 文字コードとして JIS コードを対象とする. 電子文書がそれ以外の文字コードである場合は, あらかじめ JIS コードに変更しておく. JIS コードでは英数字は 7 ビット, 漢字は 14 ビットで表現される. 英数字, 漢字それぞれバイト単位で見るといずれも各バイトの最上位ビットが 0 であり, 文字識別に用いられていない冗長ビットとなっている. 本手法ではこの冗長ビットに, 秘密情報を埋め込む. 以下では電子文書のビットの並びをビットデータと呼ぶ.

3. 秘密情報の埋め込み・抽出と改ざんの検出

ここでは, ハッシュ関数「SHA-1」を用いて秘密情報を生成し, それを埋め込み, 抽出した上で改ざんを検出する手順について説明する. 「SHA-1」とは 2 の 64 乗ビット以下のビット列から 160 ビットのハッシュ値を生成するハッシュ関数である [1]. 秘密情報が埋め込まれる前の電子文書の n バイト目を C_n, 秘密情報が埋め込まれた後の電子文書の n バイト目を C'_n, n バイト目に埋め込む 1 ビットの秘密情報ビットを H_n とする. また, $0 \leq n < L$ とする.

3.1 埋め込み手順

m バイトのビットデータを参照し, 秘密情報を算出し埋め込む手順について説明する.

STEP1 $n = 0$ とする.

STEP2 $0 \leq i < m-1$ の範囲について, C'_i=C_i とする.

STEP3 $q_r = n + r \pmod{L} (r = 0, 1, 2 \ldots, m-1)$ とする. $C'_{q_0}, C'_{q_1}, \ldots, C'_{q_{m-2}}$ と冗長ビットを 0 とした $C_{q_{m-1}}$ の計 m バイト分のビットデータに, 秘密鍵である 256 ビットのデータを加えた $8m + 256$ ビットのビットデータから SHA-1 を用いてハッシュ値を算出する.

STEP4 算出したハッシュ値のビット系列に含まれる "1" の個数の偶奇によって $C_{q_{m-1}}$ の冗長ビットに埋め込む値 $H_{q_{m-1}}$ を決定する.

STEP5 $H_{q_{m-1}}$ を $C_{q_{m-1}}$ の冗長ビットと置き換えることにより, $C'_{q_{m-1}}$ を得る.

STEP6 $n \leftarrow n+1$ とし, $n = L$ ならば処理を終え, それ以外であれば STEP3 に戻る.

3.2 抽出手順と改ざん検出手順

鑑定対象の電子文書から秘密情報を抽出し, 抽出された秘密情報と鑑定対象の電子文書から算出された秘密情報を比較することによって, 改ざんの有無を検証する.

STEP1 C'_n の冗長ビットを抽出し, $H'_n (0 \leq n < L)$ とする.

STEP2 鑑定対象の電子文書の m バイト分のビットデータと 256 ビットの秘密鍵を参照し, 3.1 節 STEP1, STEP2, STEP3, STEP4, STEP6 の手順でハッシュ値を算出する. こうして得られる系列を H''_n とする. ただし, $n < m-1$ のとき, $C'_0, \ldots, C'_{m-2}$ の最上位ビットは 0 として計算する.

鑑定対象の電子文書が改ざんされていなければ, H'_n と H''_n は一致する. よって, H'_n と H''_n が一致しなければいずれかのビットデータに改ざんがあったと判定できる. さらに H'_n と H''_n が一致しないビットの位置により改ざんの箇所を特定することができる.

4. 改ざんの種類

電子文書に対する改ざんの種類として, 文字変更, 文字追加, 文字削除があげられる. ここではそれぞれの改ざんが影響を及ぼす秘密情報の範囲, 改ざんを見過ごす確率について述べ, 改ざんの箇所, 種類を特定する方法について説明する.

† 大阪府立大学大学院工学研究科

4.1 文字変更, 文字追加

図 1(a) は, 連続する複数バイトの文字が変更（あるいは追加）された場合を表しており, 図中の網かけされた部分は改ざんされたビットデータを表す. 改ざんがビットデータの n バイト目 C'_n から $n+k-1$ バイト目 C'_{n+k-1} までの k 個に対してなされた場合, その影響は図 1(a) に示す n バイト目 C'_n の秘密情報 H''_n から, 図 1(b) に示す $n+k+m-2$ バイト目 $C'_{n+k+m-2}$ の秘密情報 $H''_{n+k+m-2}$ までの秘密情報全てに影響を及ぼす.

ここで, 改ざんを見過ごす確率について考える. 冗長ビットに埋め込まれている秘密情報 H'_n や改ざん検出のために算出される秘密情報 H''_n はいずれも 1 ビットのデータである. 改ざんの影響を受ける部分の秘密情報 H''_n は 0 または 1 のいずれかとなり, 図 1(c) に示すように, H'_n と H''_n が一致する場合 (○) と一致しない場合 (×) の確率は共に 1/2 である. 改ざんの影響が及ぶ $(m+k-1)$ 個の全てについて秘密情報を比較するため, 改ざんを見過ごす確率は $\frac{1}{2^{m+k-1}}$ になる.

4.2 文字削除

1 バイト分のビットデータが削除されたとする. この改ざんが秘密情報に及ぼす影響の範囲は, 文字変更や文字追加と同様に考えると, 参照する m バイトのビットデータが文字削除箇所をまたぐ $m-1$ 個の秘密情報となる. つまり, 改ざんを見過ごす確率は $\frac{1}{2^{m-1}}$ となる. k 個の連続する文字が削除された場合も同様である.

4.3 改ざん箇所と種類

文字変更や文字追加の場合 $m+k-1$ 個, 文字削除の場合 $m-1$ 個の秘密情報に影響が及ぶ. ここで, H'_n と H''_n の不一致が e バイト分連続していたとする. このとき, $e \geq m$ の場合は文字変更か文字追加, $e < m$ であれば文字削除による改ざんであると判断することができる. ただし, e は, H'_n と H''_n が異なっているバイト同士の間隔が m 個以上離れていない集まりを指す.

改ざん箇所は, 文字変更や文字追加の場合は H'_n と H''_n が異なっている先頭ビットから $e-(m-1)$ バイト分であると特定でき, 文字削除の場合は H'_n と H''_n が異なっている先頭のビットの直前の文字であると特定できる.

4.4 文字変更と文字追加の判定

文字変更と文字追加は次のようにして区別できる. 文字が改ざんされたバイトを特定し, そのバイトを除去した上で再び周辺の秘密情報を算出する. 冗長ビットから抽出された秘密情報と再び算出された秘密情報が一致する場合, 文字追加がなされたと判断できる. 電子文書に p バイト分の文字追加による改ざんがなされ, 改ざん箇所を特定した結果, q バイト分の改ざんと検出したとする. H'_n と H''_n が偶然一致する場合があるため, p と q の関係は $p \geq q$ となる. $p \neq q$ のとき, 改ざん箇所を正確に除去できないため文字改変と判断される. これを解決するため, 特定した q バイトから前後を r バイト分まで余分に除去し, $(r+1)^2$ 通りバイトを除去し, 秘密情報を比較する. これにより, 文字追加を見逃しにくくできる.

5. 実験と考察

$m=9$ として改ざん検出実験を行った結果を図 2 に示す. ■, △, □はそれぞれ文字削除, 文字追加, 文字改変

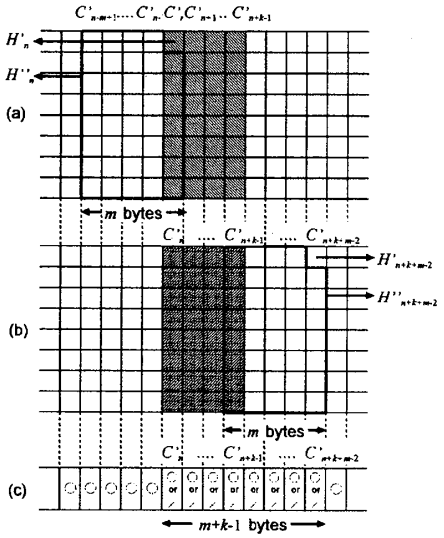

図 1 文字変更と文字追加

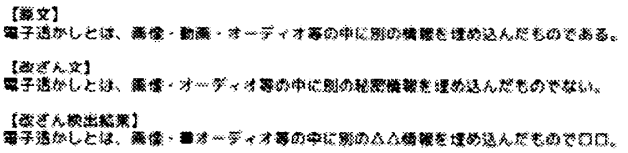

図 2 実験結果

として検出されたことを示す. 図 2 より, 改ざんを見過ごさずに検出できていることがわかる. しかし実験の結果, 改ざん箇所とその種類を正確には特定できない場合があった. これは, 埋め込まれた秘密情報と再び算出した秘密情報が 1/2 の確率で偶然一致するためである. m の値を大きくすると改ざんの有無を検出できる確率が高くなるが, その一方, 複数箇所に改ざんがある場合, それらを一ヶ所の改ざんと誤る確率が高くなる.

6. おわりに

本稿では, 改ざん箇所と改ざんの種類を特定できる電子文書のデータ構造について提案した. 文字の改ざん箇所, 改ざんの種類を正確に特定できない場合はあるものの, 改ざんを見過ごす確率を低く抑えた改ざん検出が可能であることを確認した.

参考文献

[1] "SHA-1(US Secure Hash Algorithm 1(SHA1)," http://www.ipa.go.jp/security/rfc/RFC3174JA.html