

セキュリティデバイスとヒステリシス署名を用いたデジタルフォレンジックシステムの提案と評価

芦 野 佑 樹^{†1} 佐々木 良一^{†1}

インターネットの普及にともない、流通している情報の大部分はデジタル化されている。今後は、企業に対して、法令を遵守していること（コンプライアンス）に加えて、説明責任を果たすこと（アカウントビリティ）が求められていることから、デジタルデータの証拠性を確保し、訴訟に備えるための技術や社会的仕組みであるデジタルフォレンジック（Digital Forensic）が最近注目されている。したがって、メールログや財務会計情報などのデジタルデータの証拠性を確保し、訴訟に持ち込まれてもよいようにしていくことが大切となると考えた。そこで、筆者らはこのようなことを可能にするために PC を操作した記録が、(1) 暗号処理機能と耐タンパ領域を備えたセキュリティデバイスと、(2) ヒステリシス署名技術を用い、(a) PC を操作すれば必ず記録が残るようにするとともに、(b) 改ざんや消去をしていないことを証明できる方式を提案する。あわせて、本提案方式のプロトタイプを開発し、基本的な有効性を検討したので報告する。

Proposal and Evaluation of Digital Forensic System Using Security Device and Hysteresis Signature

YUKI ASHINO^{†1} and RYOICHI SASAKI^{†1}

With development of Internet society, Digital Forensic which is the technology and the social structure to prepare digital evidence for lawsuit attracts against illegal attack has been paid attention recently. In order to achieve the accountability of a company from now on, we thought that the evidence that mail logs and financial accounting information including digital data must be not altered and that proof must be secured become important in order to cope with (a) lawsuit. We developed the system which use (1) USB device and (2) hysteresis signature technology, and can prove (a) that logging is obtained only when user operate the computer, (b) that log is not changed and not erased. In addition, we report the proposed system and results evaluated by using the prototype program.

1. はじめに

近年、インターネットの普及にともない、流通している情報の大部分はデジタル化されている。

一方、企業の不祥事などによる訴訟が数多く起きていることから、企業に対して法令遵守（コンプライアンス）や説明責任（アカウントビリティ）が求められている。このため、デジタルデータの証拠性を確保し、不正を行っていないことを証明できるようにしておくことが重要となる。一般に、コンピュータ上で作業を行うと、その痕跡がデジタルデータとして残る可能性が高い。しかしながら、デジタルデータは、加工が容

易であるためその痕跡を消してしまうことも容易である。

そのため、(1) コンピュータによるデジタルデータの取扱中に生じる作業の痕跡や作業の履歴をデジタルデータとして確保し、(2) 訴訟を起こすだけでなく訴訟された際でも説明責任が果たせるようにするための手順や技術であるデジタルフォレンジック（以下、DF）の整備が重要になってきている¹⁾。

それらを実現するデジタルフォレンジックシステム（以下、DF システム）は、管理されたネットワーク環境下で利用するネットワーク型と、それ以外の場所でも利用するスタンドアロン型に分類できる。ネットワーク型はコンピュータを操作する従業員が属している組織によって管理されるネットワーク（以下、管理されたネットワーク）を介してコンピュータを一元管理しているためリアルタイムで動作を監視することができる。ネットワーク型の DF システムは企業にも導入さ

^{†1} 東京電機大学大学院先端科学技術研究科情報通信メディア工学専攻

Graduate School of Advanced Science and Technology Information, Communication and Media Design Engineering, Tokyo Denki University

れはじめてきており、関連研究もいろいろ行われている²⁾。しかしながら、従業員のコンピュータによる作業はつねに管理されたネットワーク環境下であるとは限らない。管理されたネットワークに接続されていないコンピュータによる従業員の作業を記録するためには、スタンドアロン型の DF システムが必要である。このようなスタンドアロン環境下での DF システムの研究は筆者らが知る限りでは従来行われてこなかった。

そこで、筆者らは、プログラムが不正に変更されないという前提の下ではあるが、(1) スタンドアロン環境下で、(2) コンピュータの操作を行えば必ず記録が残るようにするとともに、(3) 操作した人を特定し、(4) 記録したデータが改ざんされていないことを後から第三者が検証できるスタンドアロン型 DF システムの検討と評価を行ってきた³⁾。

本稿では、2章で提案する DF システムについて述べ、3章ではログ保全方式の提案を行う。4章ではプロトタイプを実際に開発し評価を行った結果を記述し、5章で今後の課題を述べる。

2. 提案するスタンドアロン型 DF システム

2.1 DF システムの概要

企業はあらゆるビジネスシーンで様々なトラブルに遭遇する可能性がある。たとえば、法人や個人に対して訴訟を行う必要が出てきたり、また逆に、法人や個人から訴訟をされるといったトラブルが考えられる。このようなシーンでは、従業員がコンピュータによって取り扱った Word による文書や、PowerPoint などのプレゼンテーション資料のほか、電子メールといったデジタルデータが証拠資料として注目されはじめてきている。今後は、証拠資料としてこれらのデジタルデータのほかに、従業員がコンピュータを取り扱うことにより生じた作業履歴も重要視されてくると考えられている。したがって、だれが何をどのように作業したのかを、第三者によって作業履歴をもとに検証できる手段を設ける必要があると考えられる。

そこで筆者らは次にあげる 3 つの要件を満足する DF システムが必要であると考え、開発を目的として検討を開始した。

(要件 1) DF システムは、作業履歴をログデータとして確実にログファイルに追記する。

(要件 2) 第三者によって、だれの操作によるログデータであるかを特定することができる。

(要件 3) 第三者によって、ログファイルが改ざんされていないことを判定できる。

筆者らはこの 3 つの要件を満たす DF システムはロ

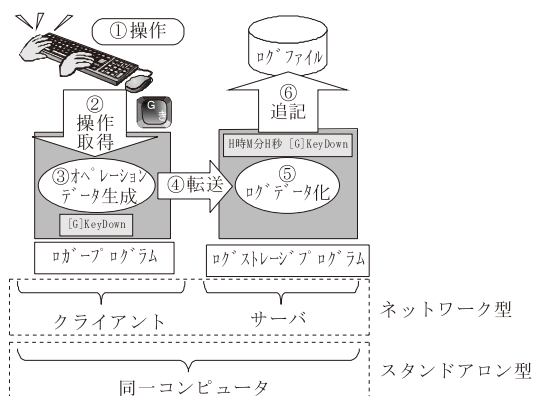


図 1 DF システム概要

Fig. 1 Overview of the DF system.

ガープログラムとログストレージプログラムの 2 つのプログラムから構成する必要があると考えた (図 1)。DF システムが、人によるコンピュータの操作を行った内容をログファイルに追記するフローを以下に示す。

まず、ロガープログラムはコンピュータの操作 (図 1 ①) を検知しその内容を取得する (図 1 ②)。その内容はオペレーションデータとして変換され (図 1 ③)、ログストレージプログラムに転送される (図 1 ④)。ログストレージプログラムでは、転送されたオペレーションデータに時刻情報などの必要な情報を付加したデータ (以下、ログデータ) に変換して (図 1 ⑤)、ログファイルに追記する (図 1 ⑥)。

2.2 DF システムの形態

DF システムは、ログストレージプログラムの設置形態によって分類できると考えられる。そこで筆者らは下記のネットワーク型とスタンドアロン型の 2 つに分類した。

(a) ネットワーク型

ネットワーク型の DF システムは、コンピュータを操作する者 (以下、操作者) が管理されたネットワーク環境下で操作することを前提としている。コンピュータおよびログストレージプログラムは管理されたネットワークに接続された形で設置される。ロガープログラムは、操作者がコンピュータを操作した内容をログストレージプログラムに逐次転送することから、コンピュータの操作内容をリアルタイムに監視することができる。したがって (要件 1) を十分に満たすことができる。また、アカウントを一元管理することから (要件 2) を満たすことも容易であると考えられる。ログストレージプログラムが遠隔地または別室のサーバ室に設置されていれば、外部および内部の攻撃者がログファイルを変更することは非常に困

難である．したがって（要件 3）も満たすことが容易であると考えられる．

(b) スタンドアロン型

スタンドアロン型の DF システムは，管理されたネットワーク環境下以外で利用されることを前提としたシステムである．たとえば，営業担当者がコンピュータを外に持ち出している場合，訪問先から管理されたネットワークに接続することができるとは限らない．また，建築士がコンピュータで行う構造設計や，公認会計士が行う会計データの書類作成といったシーンでも，組織に管理されたネットワークにコンピュータが接続されているとは限らない．このように管理されたネットワークに接続しなくても利用することのできる DF システムをスタンドアロン型と定義する．スタンドアロン型は，コンピュータが管理されたネットワークに接続できないため，アカウントを一元管理することが困難である．また，操作者が攻撃者となった場合，ログファイルはコンピュータ上に存在しているため，改ざんの危険性が高いと考えられる．

2.3 提案システムの構成

筆者らは（要件 1～3）を満たすスタンドアロン型 DF システムを提案する．

提案システムでは，2.1 節で説明したプログラムのほかに暗号演算機能と暗号演算機能からしか内容を書き換えることができないストレージ領域（以下，耐タンバ領域）を持ったセキュリティデバイスを新たに設ける．提案システムを構成する各要素の役割と機能については 2.4 節で詳細に述べる．また，提案システムに関与する者を 2.5 節で詳細に述べる．

提案システムでは，以下の流れで操作内容をログファイルに追記する．

ロガープログラムは，操作者が操作した内容を取得し（図 2 ①）オペレーションデータに変換を行いログストレージプログラムに転送する（図 2 ②）．ログストレージプログラムは，オペレーションデータに時刻情報などを付加しフォーマットデータへと加工を行った後（図 2 ③），それらのデータをまとめてハッシュ値の計算を行った結果をセキュリティデバイスに転送する（図 2 ④）．セキュリティデバイスでは，受け取ったハッシュ値に対して 3 章で詳しく述べるログ保全方式を施す（図 2 ⑤）．保全方式を施した結果は署名データとして耐タンバ領域に上書き更新される（図 2 ⑥），その後ログストレージプログラムに転送される（図 2 ⑦）．ログストレージプログラムは，ログ保全方式が施された署名データとフォーマット化されたデータをあわせたものをログデータとしてログ

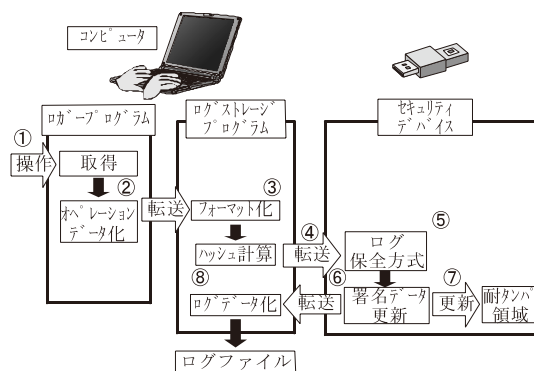


図 2 提案システムの構成

Fig. 2 Overview of the proposal DF system.

ファイルに追記する（図 2 ⑧）．

2.4 提案システムを構成する要素の機能と役割

(a) コンピュータ

管理者によって提案システムの各プログラムがインストールされた PC などのコンピュータである．

(b) ロガープログラム

ロガープログラムは，コンピュータ内に存在し以下の機能を有する．

- (I) セキュリティデバイスの装着の状態を監視している．
- (II) セキュリティデバイスが装着されていないときはコンピュータの操作をロックする．
- (III) 装着を確認したらロックを解除し，コンピュータへの入力操作をオペレーションデータに変換しログストレージプログラムに転送する．

これにより，セキュリティデバイスが装着された場合のみ入力が可能となり，入力していないときに記録が残ることはないので（要件 1）を満たすことができる．また，2.6 節に述べる（前提条件 4）により，セキュリティデバイスは他人に渡ることにはないので（要件 2）である入力操作を行った人を特定することができる．

(c) オペレーションデータ

オペレーションデータは，ロガープログラムが生成した操作内容をデジタルデータ化したものである．操作内容としては，キーボードの操作内容や，マウスの位置，操作しているアプリケーション名のほか，必要に応じて Word や Excel といったドキュメントデータや電子メールなどが考えられる．

(d) ログストレージプログラム

ログストレージプログラムは，コンピュータ内に存在する．ロガープログラムから転送されたオペレーションデータに時刻情報などを付加したフォーマットデータを，セキュリティデバイスと連携して 3 章で述

べるログ保全方式を施し、ログデータ化した内容をログファイルに記録する。

(e) フォーマットデータ

オペレーションデータに、時刻情報などを付加したデータである。

(f) セキュリティデバイス

セキュリティデバイス(たとえば、IC カード機能付きの USB デバイス)は、以下の機能を有する。

(I) ログストレージプログラムから送られてきたデータに対して 3 章で述べるログ保全方式による処理を施す。

(II)(I)の結果をセキュリティデバイス内の耐タンパ領域に記録する。

(III)(II)の結果をログストレージプログラムに転送する。

(g) ログデータ

ログデータは、フォーマットデータに対して(要件3)を満たすために 3 章で述べるログ保全方式を施したデータとフォーマットデータのセットである。

(h) ログファイル

ログファイルは、ログストレージプログラムがログデータを追記していったファイルである。

2.5 関与者

関与者として以下のような人たちがいるものとする。

(a) 管理者

管理者は以下を行う。

(I) コンピュータヘログガープログラムとログストレージプログラムをインストールする。

(II) セキュリティデバイスに設定する署名鍵および公開鍵の作成を行う。

(III) セキュリティデバイス内へ初期連鎖用データを設定する。

(IV) 検証者に公開鍵と初期連鎖用データを送る。

(b) 操作者

管理者から渡されたコンピュータとセキュリティデバイスを持っている。

操作者は、セキュリティデバイスをコンピュータに装着することにより操作ロック機能を解除して操作を行う。離席の際は、セキュリティデバイスを外しコンピュータのロックを行う。操作者は、すべての作業が終了した後、OS の終了とコンピュータの電源を切ったうえで、コンピュータとセキュリティデバイスを検証者に手渡す。

(c) 検証者

操作者からセキュリティデバイスとコンピュータを受け取り、ログファイルの検証を行う。管理者と検証

者が同じであってもよい。

2.6 前提条件および機能目標

提案システムは、下記の前提条件の下で検討を行った。

(前提条件 1) 各プログラムは不正に変更されることはない。この条件については、実現するうえで問題となる場合が多いと考えられる。したがって 5 章に示すように対策案を開発中であるが、今回はこの前提条件の下で検討した。

(前提条件 2) 管理者が使用するプログラムは、個人認証が正しく行われていて、あらかじめ指定している人間のみがアクセスできる。この条件については、各種認証機能によって実現が可能であると考えられる。また、検証プログラムはログファイルの検証機能のみを有しており、ログファイルおよびセキュリティデバイス内のデータは変更しない。

(前提条件 3) 操作者は不正を行う可能性はあるが、管理者は不正を行わない。この条件については、一般に採用されている前提であると考えられる。

(前提条件 4) 操作者はセキュリティデバイスを他人に渡さない。この条件については、操作者がセキュリティデバイスを、自宅の鍵やクレジットカードのほか、社員証と同様に取り扱うことを前提にしている。

提案システムでは、上記の前提条件に加えてセキュリティデバイスを用いていることから下記の機能目標を設けることにする。

(機能目標) セキュリティデバイス内の耐タンパ領域の容量は必要最小限にする。この目標は、耐タンパ領域が一般に高コストであることから、できる限り容量を最小にする必要があると考えた。

3. スタンドアロン型に適したログ保全方式の提案

3.1 ログファイルに対する攻撃手順

操作者はコンピュータの各種デバイスに直接触ることができることから、ハードディスク上のログファイルに記録されているログデータの削除や変更といった改変を行うことができる。ログデータの改変を試みる操作者(以下、攻撃者)は、(前提条件 1)により提案システムの各プログラムを不正に変更することができないため、ログデータに対する操作内容がログデータとしてログファイルに記録されてしまう。したがって、攻撃者は、コンピュータの電源を切ったうえで、以下の手順でログデータの改変を試みると考えられる。

Step1 コンピュータからログファイルの入ったハードディスク(以下、ハードディスク)を取り外し、ログ

ファイルの書き換えを行うコンピュータ（以下、書き換え用コンピュータ）にハードディスクを接続させる。
Step2 書き換え用コンピュータでハードディスク上のログファイルを書き換える。

Step3 書き換え用コンピュータからハードディスクを取り外し元のコンピュータに装着する。

Step4 コンピュータの電源を投入し提案システムを稼働させる。

スタンドアロン型 DF システムは、上記の書き換えそのものを防ぐことは大変難しい。したがって、攻撃者がログファイルを改変したとしても、後から第三者によって改変が検出できるようにしておく必要がある。

3.2 ログ保全方式

ログ保全方式は、1 つ以上のログデータに対して何らかの改変が行われたとしても、(要件3)である第三者によって改変が検出できることを目的とする。

(1) MAC (Message Authentication Code) を用いる方式

本方式は、セキュリティデバイスを用いなくても実施は可能であるが、議論を行ううえで重要な方式であると考え検討に加えることにする。

Schneier らによって考案された安全なログデータ保存システム⁴⁾は、ログデータの MAC を生成する際に用いる署名鍵を処理ごとに更新と破棄を行うことを特徴としている。このため、仮に署名鍵が盗まれたとしても、過去の署名鍵はすでに破棄されているので、過去に遡って検証に成功するログデータを偽造することは不可能である。

しかしながら、検証者は最初の署名鍵を持っている必要があるため、任意のログデータを変更したり削除したりすることができてしまう。したがって、このシステムでは検証者が署名鍵を所有していることから改ざんしてもそれを検出できないログファイルの作成が可能になってしまう。これは検証者に最大限の信頼をしなければならないことから（要件２）および（要件３）を満たすことができない。

よって、今回の目的には適さないと考えられる。

(2) セキュリティデバイスとデジタル署名を用いる方式

デジタル署名は MAC と違い、第三者によってデータの改変が行われていないことを検証する技術である。したがって、ログ保全方式としてデジタル署名を行うことで、ログファイルを第三者によって検証することができるようになる。フォーマットデータがログストレージプログラムに転送されてくるたびにログファイル全体に対して署名を行いその結果をセキュリティデ

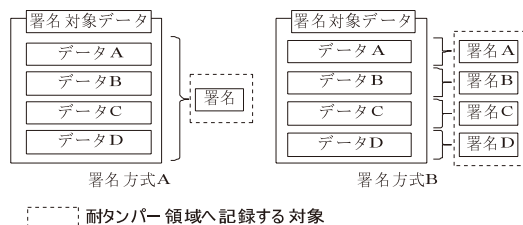


図 3 デジタル署名の署名対象

Fig.3 Method of using a digital signature.

バイスの耐タンパ領域に記録する方法（以下、デジタル署名方式 A）、個別のフォーマットデータに対して署名を行い過去に署名したすべての署名を耐タンパ領域に記録する方法（以下、デジタル署名方式 B）の 2 つに分類する（図 3）。

署名方式 A はすべてのログデータに対して署名処理を行うため、どのログデータに対する改変も検出が可能であるが、ログデータが生成されるたびにログファイル上の全ログデータのハッシュ値を計算する必要がある。ログファイルの容量は時間に比例することが予想される。そのため、容量が膨大になってしまったときはハッシュ値の計算に時間がかかってしまう。

署名方式 B は、フォーマットデータに対して署名処理を行うためハッシュ値を計算する時間はログファイルの容量には依存しないため、署名方式 A と比べるとハッシュ計算については処理効率が良い。しかしながらフォーマットデータの署名結果をログファイルに記録するだけでは、各署名データ間の相関関係が存在しないため、ログデータ全体を削除してしまうと、後からログデータの存在を検出することができなくなる。そこで、過去にわたるすべての署名データを耐タンパ領域に記録することで、ログデータ全体を削除してもその存在を耐タンパ領域から検出することができる。しかしながら、この方式では、耐タンパ領域に膨大な容量を必要とすることから（機能目標）を満たすことができない。

これらを総合すると、セキュリティデバイスとデジタル署名はログ保全方式としては不向きであるといえる。

(3) セキュリティデバイスとヒステリシス署名を用いる方式

ヒステリシス署名とは、デジタル署名を拡張した概念であり、署名を行う際に過去に署名したデータ（以下、連鎖用データ）を署名対象データに含めた状態で署名処理を行うのが特徴である。ヒステリシス署名は、暗号が危殆化しても署名付きの文書の安全性を確保するために開発されたものである。連鎖用データによ

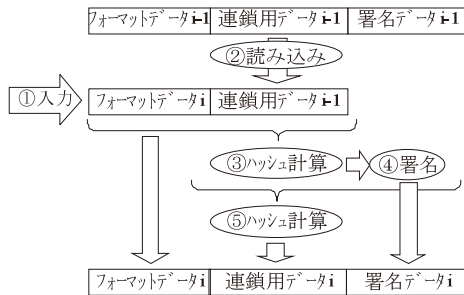


図4 ヒステリシス署名の基本的仕組み

Fig. 4 Overview of the hysteresis signature.

て過去にわたる署名の連鎖構造を構築することによって、過去から現在に至るまで署名を施したのと同じ効果を得ることができる^{5),6)}。ヒステリシス署名を本提案システムに適用した際の基本的仕組みを図4に示す。

最新の署名に1つ前のデータが反映されることにより署名の連鎖構造が形成される(図4⑤)。これを繰り返すことで、さらに前のデータが改ざんされた場合であっても、最新の署名さえ持っていればそれを検知できるという特徴がある。連鎖用データを生成するフローを以下に説明する。

ログファイル上の最新(*last* 番目)のログデータ *last* には、フォーマットデータ *last*、連鎖用データ *last*、署名データ *last* の各データがある。連鎖用データ *last* + 1 を生成する際は、ログファイル上のログデータ *last* に記録された各データからハッシュ値を計算する。このようにすることで、*last* から *last* + 1 の間に連鎖構造を形成していくことができる。これは、従来のヒステリシス署名を適用することによって実現できる。

しかし、このままでは、攻撃者が下記の手順により、ログファイルの末尾方向(*last* 番目)から任意の個数分のログデータを削除すると、それを検出できないログファイルを作成することが可能になる。

例として、攻撃者が3.1節の手順によりログファイルに記録されているログデータ *last* を削除した後に、再びDFシステムを稼働したとする。このとき、DFシステムは *last* - 1 番目のログデータが最後のログデータであると誤認識し、連鎖用データ *last* + 1 をログデータ *last* - 1 から生成してしまう。連鎖用データ *last* + 1 は *last* - 1 番目からの連鎖構造になることから、ログデータ *last* が消えてしまったことを検出することができない。攻撃者はこの方法を応用して、*last* 番目よりも前のログデータに対して攻撃を実施できる。

攻撃者によるこうした攻撃を防ぐため、ログファイル上のログデータから連鎖用データを生成しないよう

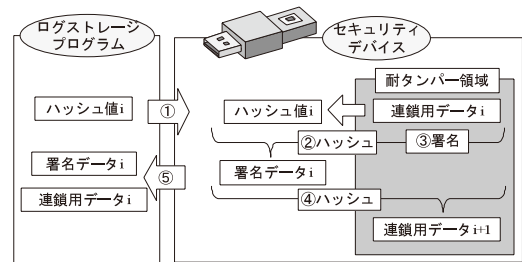


図5 セキュリティデバイスを用いたログ保全方式

Fig. 5 Hysteresis signature with security device.

にする必要がある。そこで、セキュリティデバイス内で、下記の手順によって連鎖用データの生成を行うことにした。

前提として直前に生成した連鎖用データ *i* はセキュリティデバイスの耐タンパ領域に記録されているとする。

フォーマットデータ *i* に対してヒステリシス署名を行う場合、ログストレージプログラムはフォーマットデータ *i* のハッシュ値 *i* をセキュリティデバイスに転送する(図5①)。セキュリティデバイスは転送されてきたハッシュ値 *i* と耐タンパ領域に記録されている連鎖用データ *i* とのハッシュ値を計算し(図5②)、デジタル署名を行いその結果を署名データ *i* とする(図5③)。署名データ *i* と連鎖用データ *i* とハッシュ値 *i* のハッシュ値を計算して連鎖用データ *i* + 1 として、耐タンパ領域に上書きする(図5④)。ログストレージプログラムには、連鎖用データ *i* と署名データ *i* を転送する(図5⑤)。

なお、演算のフローなどの詳細は、4.1節で詳しく述べる。

検証の際は、セキュリティデバイス内に記録されている連鎖用データ *last* + 1 と、ログファイル上のログデータ内にあるハッシュ値 *last*、署名データ *last*、連鎖用データ *last* のハッシュ値が同一であるかを判定する。

もし、一致しなければ、正しく記録されていないと判断し検証を終了する。*last* 番目を *i* 番目として、(1)ハッシュ値 *i* と連鎖用データ *i* のハッシュ値を公開鍵で復号した値を署名データ *i* と比較する。一致しなければ署名データは信頼できないと判断して検証を終了する。(2)オペレーションデータ *i* のハッシュ値とハッシュ値 *i* を比較する。一致しなければログデータ *i* が改変されたと判断して検証を終了する。(3)ログデータ *i* - 1、連鎖用データ *i* - 1 および署名データ *i* - 1 のハッシュ値と連鎖用データ *i* とを比較する。一致しなければ、*i* - 1 と *i* のログデータ間では相関関係がないと判断し検証を終了する。一致していた場

合は、 $i = i - 1$ として (1)~(3) の処理を i が 0 になるまで繰り返す。このように、セキュリティデバイスで最後のデータが一致を確認する工程を除けば、ヒステリシス署名はデジタル署名と同様に公開鍵によって第三者が検証できる (要件 3)。これに加えて耐タンパ領域に格納しなければならないデータは、1 つの連鎖用データで済むため (機能目標) を満たす。また、デジタル署名方式 A と違いハッシュ計算の時間を大幅に減らすことができる。

筆者らは、ログ保全方式としてヒステリシス署名とセキュリティデバイスを用いる上記のような方式を採用した。

4. 提案システムの運用とプロトタイプの開発

筆者らが提案するスタンドアロン型 DF システムに、3 章で検討したヒステリシス署名とセキュリティデバイスを用いたログ保全方式を採用し、運用方法について検討を行った。あわせて、プロトタイププログラムを開発し運用方法の妥当性を検討した。

4.1 提案システムの運用フロー

運用フローは、以下の 3 つのフェーズに分けることにした。導入フェーズでは、管理者により初期設定を行う。蓄積フェーズでは、操作者がコンピュータを操作する。検証フェーズでは、コンピュータ内に記録されたログデータの検証を行う。

(1) 導入フェーズ

導入フェーズでは、管理者が使用するコンピュータおよびセキュリティデバイスに必要なプログラムのインストールおよび初期データの設定を行う。管理者は、以下の Step でセキュリティデバイスとコンピュータのセットアップを行う。なお、管理者が使用するプログラムは (前提条件 2) により正しく管理者を識別しているものとする。

Step1 秘密鍵 S と公開鍵 P を生成する。

Step2 セキュリティデバイスに秘密鍵 S をセットする。

Step3 また、連鎖用データ R_1 を任意に選びセキュリティデバイスにセットする。

Step4 検証者に公開鍵 P 、連鎖用データ R_1 を送付する。

Step5 システム管理者はコンピュータにロガープログラムとログストレージプログラムをインストールする。

Step6 セキュリティデバイスとコンピュータを操作者に渡し、フェーズを終了する。

(2) 蓄積フェーズ

蓄積フェーズでは、コンピュータの電源を入れ OS

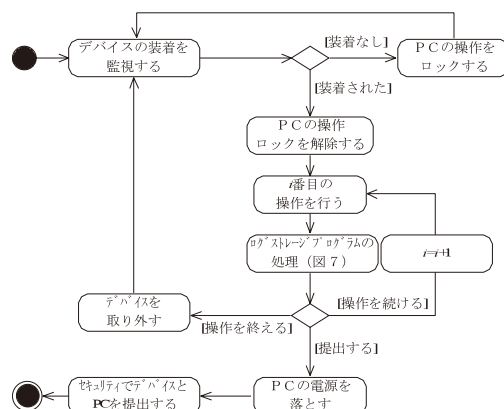


図 6 蓄積フェーズ
Fig. 6 Storage phase.

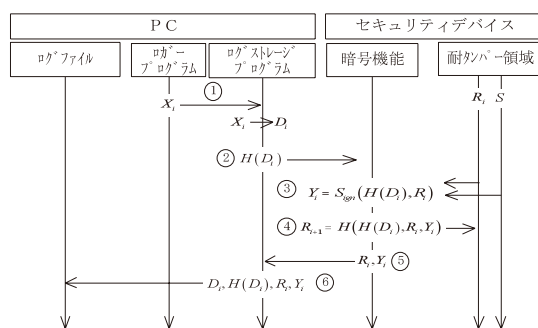


図 7 ログ保全方式のデータフロー
Fig. 7 Data flow of the secure logging.

を起動した後、以下の Step により操作者およびコンピュータ内部の各プログラムとセキュリティデバイスが通信を行いログファイルにログデータを追記していく。また、操作者はすべての作業が終了した後、OSの終了とコンピュータの電源を切ったうえでフェーズを終了の手続きを行うものとする (図 6)。

Step1 ロガープログラムは、セキュリティデバイスの装着を監視しており、セキュリティデバイスが装着されていない間はコンピュータの操作をロックする。

Step2 操作者はコンピュータを操作するためにセキュリティデバイスをコンピュータに装着する。

Step3 ロガープログラムは、セキュリティデバイスの装着を確認したら、コンピュータが操作できるようにロック機能を解除する。

Step4 ロガープログラムは、操作者が i 番目に操作した内容をオペレーションデータ X_i としてログストレージプログラムに転送する (図 7 ①)。

Step5 ログストレージプログラムは、 X_i に時刻情報などを付加したフォーマットデータ D_i のハッシュ値 $H(D_i)$ をセキュリティデバイスに転送する (図 7 ②)。

Step6 セキュリティデバイスは $H(D_i)$ と耐タンパ領域に記録されている連鎖用データ R_i に対し、同じ領域にある秘密鍵 S を用いてデジタル署名 $S_{\text{ign}}(H(D_i), R_i)$ より Y_i を計算する (図 7 ③)。

Step7 $H(D_i)$, R_i , Y_i のハッシュ値を連鎖用データ R_{i+1} として耐タンパ領域に記録する (図 7 ④)。

Step8 セキュリティデバイスは、 R_i , Y_i をログストレージプログラムに送る (図 7 ⑤)。

Step9 ログストレージプログラムは、 D_i , $H(D_i)$, R_i , Y_i をログデータ i としてログファイルに書き込む (図 7 ⑥)。

Step10 操作者が操作を継続する場合は、 $i = i + 1$ として Step4 へ。操作者が操作を終了しセキュリティデバイスを取り外したら、Step1 へ。操作者が自らの操作を検証者に報告する場合は、Step11 へ。

Step11 操作者は OS を終了させ、コンピュータの電源を切った後、セキュリティデバイスとコンピュータを検証者に渡しフェーズを終了する。

(3) 検証フェーズ

検証者は、操作者からコンピュータとセキュリティデバイスを受け取り、検証プログラムを用いて以下の手順でログファイルの検証を行う。なお、検証プログラムは (前提条件 2) により検証者を正しく識別するものとする。

Step1 検証者は、ログファイルに記録されている $last$ 番目のログデータ内にある $H(D_{last})$, R_{last} , Y_{last} のハッシュ値を計算し、セキュリティデバイスに記録されている連鎖用データ R_{last+1} のデータと比較する。一致した場合は、 $last$ を i として次の Step に進む。不一致だった場合はログファイルに記録されている $last$ 番目のログデータに異常があったと判断して、警告を出してフェーズを終了する。

Step2 $i = 0$ なら Step5 へ。そうでなければ、ログファイル上にある D_i のハッシュ値を計算し (図 8 ①)、同じくログファイル上の $H(D_i)$ を比較する (図 8 ②)。一致していたら次の Step3 に進む。一致していない場合は D_i が変更されたと判断して警告を出してフェーズを終了する。

Step3 ログデータ内の Y_i を公開鍵 P で復号した結果 (図 8 ③) と $H(D_i)$, R_i のハッシュ値を比較する (図 8 ④)。一致していたら次の Step4 に進む。一致しなかった場合は署名の検証が失敗したと判断して警告を出してフェーズを終了する。

Step4 ログデータ内の $H(D_{i-1})$, R_{i-1} , Y_{i-1} からハッシュ値を計算し (図 8 ⑤), R_i と比較する (図 8 ⑥)。一致していたら $i = i - 1$ として Step2 に

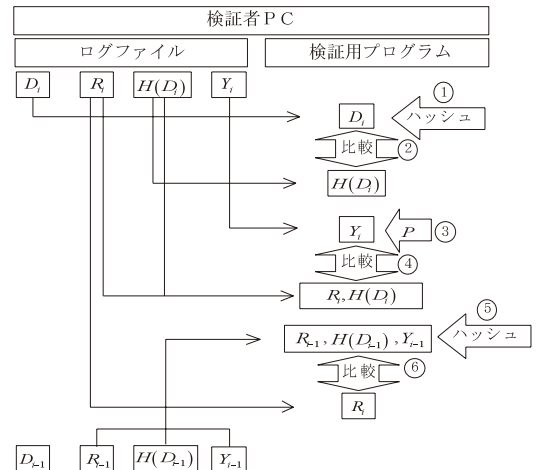


図 8 検証フロー
Fig. 8 Verifying flow.

進む。一致しなかった場合は i 番目と $i - 1$ 番目の間に相関関係が成立しなかったと判断して警告を出してフェーズを終了する。

Step5 ログファイル上の R_1 と管理者から送られてきた R_1 とを比較し、一致したら全部のログデータが 1 つも欠損がなく、かつ、変更されていないことを証明できたと判断してフェーズを終了する。一致していない場合は、検証が失敗したと判断し警告を出してフェーズを終了する。

4.2 プロトタイプの開発

本提案システムの有効性を検証するために、プロトタイプシステムを開発し、実運用上の可能性と処理性能についての検証を行った。検証環境は Windows XP Professional を使った。また、以下のような装置やシステムを用いた。

(1) セキュリティデバイス

アラジン社の e-Token Pro (以下、eToken) を用いた⁷⁾。eToken 内部には耐タンパ領域に秘密鍵と公開鍵をセットすることができるうえ、トリプル DES による共通鍵暗号機能のほか、RSA の公開鍵暗号機能が利用できる。eToken はコンピュータとの接続には USB インタフェースを利用する。USB はほとんどすべてのコンピュータに装備されており、そのコンピュータに eToken ランタイムライブラリをインストールするだけで eToken を利用することができる。

(2) ロガープログラム

フリーウェアの logol2⁸⁾ を用いた。logol2 では、アクティブウィンドウのタイトルのほか、ユーザが入力したキーボードの情報を logol2 が生成するログファイル (以下、テンポラリーログファイル) に記録する。

(5) に述べる制限により、コンピュータのロック機能はログストレージプログラム側で行う。

(3) ログストレージプログラム

開発環境は VisualStudio.NET 2003 で言語は C++ を用いた。ログストレージプログラムでは、logol2 が生成したテンポラリーログファイルの変更差分を 1 秒ごとに監視を行いフォーマットデータとして取り扱い、そのデータと eToken に記録されている連鎖用データを読み込みハッシュ値を計算することで新たな連鎖用データを生成する。

(4) 検証プログラム

プロトタイプが生成したログファイルを検証する目的で作成した。(3) 同様に VisualStudio.NET 2003 で言語は C++ を用いて開発を行った。

(5) プロトタイプの制限

プロトタイプには、各種機能および運用を簡素化するために以下の制約を設けている。

(a) 連鎖用データはログストレージプログラム側で行う

eToken の暗号機能は、直接 eToken 内の耐タンパ領域のデータにアクセスすることができない。そのため、ログストレージプログラム側で eToken の耐タンパ領域のデータからデータをロードして、ログストレージプログラム側で連鎖用データを作成し、その結果を eToken の耐タンパ領域に書き戻す。

(b) コンピュータのロック機能はログストレージプログラム側で行う

logol2 に新しい機能を追加することはできないので、コンピュータのロック機能はログストレージプログラム側で行う。

4.3 プロトタイプでの運用

(1) 導入フロー

管理者として PC に対して以下の作業を行った。

- (I) ロガープログラムのインストール。
- (II) ログストレージプログラムのインストール。
- (III) eToken ランタイムのインストール。
- (IV) eToken 内部のストレージに初期連鎖用データをセット。
- (V) ロガープログラム、ログストレージプログラムを起動した状態でコンピュータを操作者に渡す。

なお、ロガープログラムとログストレージプログラムは、操作者がコンピュータにログインを行うと起動するようにした。

(2) 蓄積フロー

操作者は、eToken を挿入しコンピュータロック機能を解除してから、コンピュータの操作を行う。操作

者が操作を行うとロガープログラムによりテンポラリーログファイルにデータが蓄積される。テンポラリーログファイルの増分をログストレージプログラムが検出し、ハッシュ値を計算し eToken の暗号機能を用いてヒステリシス署名を行う。署名データはログストレージプログラムによって、eToken の耐タンパ領域に書き込まれ、また、ログデータとしてログファイルに追記される。操作者が eToken を取り外すと、ログストレージプログラムはコンピュータをロックする。

(3) 検証フロー

検証者は、操作者からコンピュータを受け取り、ロガープログラムとログストレージプログラムを終了させた後に、ログファイルの検証を行う。検証プログラムは、eToken 内部に保存されている連鎖用データとログファイル上の最後に記録されたログデータのハッシュ値とを比較して一致してからヒステリシス署名の検証を開始する。途中で検証に失敗すると警告を出し、ログファイルが信頼のできないものであることを伝える。また、最後まで検証が成功した場合は、ログファイルが信頼のできることを報告する。

4.4 評価

(1) ログファイルの正当性の検証

ログファイルに含まれる各ログデータは、ヒステリシス署名によって相関関係が構築されている。したがって、どのログデータに対して改変しても、検証プログラムによって後からそれを検出することができ、(要件 3) を満たすことができた。また、(機能目標) であるセキュリティデバイスの耐タンパ領域を必要最小限にできた。さらに、蓄積フロー時のハッシュ値を計算する時間を必要最小限にできた。以上のことから、筆者らが提案するスタンドアロン型 DF システムとして、ヒステリシス署名は有効であると判断できた。

(2) 操作記録性

logol2 で採取できるログの範囲で操作者の行った操作を追跡することができた。たとえば、何のウィンドウを開いて何を入力したのか、といった細かい作業まで確認することができた。特に、作業中にウェブを見た際、入力した URL や検索エンジンでのキーワードも確認できた。そのため、(要件 1) を十分に満たすことができ、加えて業務に関係のないウェブを見ないようにする抑止力になると考えられる。同様に、セキュリティデバイスを抜くとコンピュータの操作ができないことから、(前提条件 4) により特定の人以外が操作者になりえないので、(要件 2) を満たすことができた。このことは、操作者が不正などの疑いが持たれた際、このログデータによって自らの正当性を主張す

ることもできると考えられる。

(3) 処理性能

eToken を用いてログを作成する際の処理時間について、以下の方法で測定を行った。(a) ログストレージプログラムがハッシュ値を計算して eToken に送る直前の時刻を取得し、(b) eToken で署名処理を行わせ、(c) eToken から処理結果がログストレージプログラムに返ってきた直後の時刻を取得し、(d) (c)-(a) を計算して処理時間とした。上記のフローを 5 回繰り返し、その平均時間を測定した。PC は東芝社製 libretto U100 (超低電圧版 Pentium M 733, RAM 1 GB) を用いた。その結果、1 回あたりの処理時間は 0.55 秒であった。ログストレージプログラムは、4.2 節で述べたとおり 1 秒ごとに eToken にデータを転送しているため、処理時間およびキー操作を追跡するうえでは実用上問題のない範囲である。

5. 今後の課題

プロトタイプを通じて運用上および処理性能については問題がないことが分かったが、同時に以下の点についての課題が明確となった。

(1) ログの対象

今回はオペレーションデータとして logol2 が生成したデータを対象としてきたが、キーボードの操作情報だけでなく、マウスの操作内容やスクリーンショット、ネットワークのパケットなども記録として蓄積していく必要があると考えられる。しかしながら、証拠性を高めようとすればするほど、単位時間あたりのオペレーションデータの容量も増加してくる。そのため、どのようなシーンでどのデータをオペレーションデータとして取り扱えば DF システムとして必要十分な証拠性を確保できるのかについての検討が必要であると考えられる。

(2) プログラムの安全な動作

本提案システムでは、ロガープログラムやログストレージプログラムなどがだれからも変更されないことを前提条件としていたが、実際の運用を考えた際、この条件を満たせない場合も多いと考えられる。

万一、DF システムの各プログラムが不正に書き換えられた場合に備えて、プログラムの改ざんを検出しコンピュータを使えないようにする方式が必要である。現在筆者らは、TCG⁹⁾により策定されている TPM (Trusted Platform Module) などのハードウェアのほか、デバイスドライバなどを用いてプログラムの改ざんを検出した場合コンピュータの操作をできなくする方式を検討中である。

6. おわりに

本稿では、セキュリティデバイスとヒステリシス署名が DF システムとしての有効性をプロトタイプの開発を介して評価を行い、基本的な有効性を確認できた。

今後は、前述した 2 つの課題について取り組んでいき、より実用的な DF システムの実現を目指していきたいと考えている。

謝辞 本稿作成に関連し、プログラムの作成などに協力いただいた卒業生の青井尚幸君、粉川寛人君、佐藤史君、ならびに同研究室の藤田圭祐君に謹んで感謝の意を表する。

参 考 文 献

- 1) 佐々木良一、芦野佑樹、増淵孝延：デジタル・フォレンジックの体系化の試みと必要技術の提案、JSSM 学会誌, Vol.20, No.2, pp.49-61 (2006).
- 2) 安東 学、松浦幹太、馬場 章：分散環境で保存されるログファイルにおける各ログエントリ間の順序関係保証方法に関する考察、コンピュータセキュリティシンポジウム (CSS) 2002 論文集, 情報処理学会シンポジウムシリーズ, Vol.2002, No.16, pp.1-6 (2002).
- 3) 芦野佑樹、佐々木良一：セキュリティデバイスとヒステリシス署名を用いたデジタルフォレンジック技術の提案と評価、コンピュータセキュリティシンポジウム (CSS) 2006 論文集, 情報処理学会シンポジウムシリーズ, Vol.2006, No.11, pp.363-368 (2006).
- 4) Schneier, B. and Kelsey, J.: Cryptographic Support for Secure Logs on Untrusted Machine, *The 7th USENIX Security Symposium Proceedings*, pp.53-62, USENIX Press (Jan. 1998).
- 5) 岩村 充、宮崎邦彦、松本 勉、佐々木良一、松木武：電子署名におけるアリバイ証明問題と経時証明問題—ヒステリシス署名とデジタル古文書の概念、コンピュータサイエンス誌 bit, Vol.32, No.11, 共立出版 (2000).
- 6) 洲崎誠一、松本 勉：電子署名アリバイ実現機構—ヒステリシス署名と履歴交差、情報処理学会論文誌, Vol.43, No.8, pp.2381-2393 (2002).
- 7) Aladdin Knowledge Systems, Ltd.
<http://www.aladdin.com/>
- 8) 作者 folia.
<http://hp.vector.co.jp/authors/VA026310/>
- 9) TCG (Trusted Computing Group).
<https://www.trustedcomputinggroup.org/home/>

(平成 19 年 5 月 9 日受付)

(平成 19 年 11 月 6 日採録)



芦野 佑樹（学生会員）

平成 14 年北海道東海大学工学部
電子情報工学科卒業。卒業後、会社
員としてシステム開発に従事しながら、平成 16 年東京電機大学大学院
工学研究科情報メディア学専攻修士
課程に入学。平成 18 年 3 月同課程を修了し会社を退
職。同年 4 月同大学院先端科学技術研究科情報通信メ
ディア工学専攻博士課程に入学し現在に至る。デジタル
フォレンジックシステムを中心に研究を行っている。



佐々木良一（フェロー）

昭和 46 年 3 月東京大学卒業。同
年 4 月日立製作所入所。システム開
発研究所にてシステム高信頼化技術、
セキュリティ技術、ネットワーク管
理システム等の研究開発に従事。平
成 13 年 4 月より東京電機大学工学部教授、平成 19 年
4 月より未来科学部教授。工学博士（東京大学）。平
成 10 年電気学会著作賞受賞。平成 14 年情報処理学会
論文賞受賞。平成 19 年総務大臣表彰等。著書に、『イ
ンターネットセキュリティ入門』（岩波新書、1999 年）
等。情報処理学会フェロー。情報処理学会コンピュ
ータセキュリティ研究会顧問。日本セキュリティ・マネ
ジメント学会常任理事。情報ネットワーク法学会理事
長。日本学術会議連携会員。日本ネットワークセキュ
リティ協会会長。