TPM と TSS を用いたエンドホストのロギング機構

福田 洋治 I)　白石 善明 II)　毛利 公美 III)　溝渕 昭二 IV)　野口 亮司 V)

I) 愛知教育大学
448-8542 刈谷市井ヶ谷町広沢 1
yfukuta@auecc.aichi-edu.ac.jp

II) 名古屋工業大学
466-8555 名古屋市昭和区御器所町
zenmei@nitech.ac.jp

III) 岐阜大学
501-1193 岐阜市柳戸 1-1
mmohri@gifu-u.ac.jp

IV) 近畿大学
577-8502 東大阪市小若江 3-4-1
mizo@info.kindai.ac.jp

V) 株式会社豊通シスコム
450-0002 名古屋市中村区名駅 4-5-28 近鉄新名古屋ビル
ict_research@tsyscom.co.jp

あらまし　本稿では，コンピュータやネットワークに関する犯罪や法的紛争が生じた際の調査，解析の技術であるデジタルフォレンジックに注目して，ログの法的証明力を考慮したエンドホストにおけるロギングの機構を示す．法的紛争・訴訟への対応を想定した場合，当該組織に属する人間が，フォレンジックシステムの管理者であると，自身の組織に都合の悪いログが意図的に改変，削除されることが考えられる．本稿で示すロギングの機構は，エンドホストの TPM(trusted platform module) セキュリティチップと，信頼できる第三者機関により提供される TSS(time stamp service) を併用し，これを困難にする．

An End-host based Logging Mechanism Using TPM and TSS

Youji Fukuta I)　Yoshiaki Shiraishi II)　Masami Mohri III)　Shoji Mizobuchi IV)　Ryoji Noguchi V)

I) Aichi University of Education
1 Hirosawa Igaya-Cho Kariya 448-8542 Japan
yfukuta@auecc.aichi-edu.ac.jp

II) Nagoya Institute of Technology
Gokiso-Cho Showa-Ku Nagoya 466-8555 Japan
zenmei@nitech.ac.jp

III) Gifu University
1-1 Yanagido Gifu 501-1193 Japan
mmohri@gifu-u.ac.jp

IV) Kinki University
3-4-1 Kowakae Higashi-Osaka 577-8502 Japan
mizo@info.kindai.ac.jp

V) Toyotsu Syscom Corporation
4-5-28 Meieki Nakamura-Ku Nagoya 450-0002 Japan
ict_research@tsyscom.co.jp

Abstract　In this paper, we focus on digital forensics and show a end-host based logging mechanism considering probative value of digital evidence. In case of supposing legal conflicts and lawsuits, when the administrators of a forensics system belong to an organization which have that system, there is the issue that some logs, which are detrimental to themselves, can be modified and removed. Our logging mechanism deal with the issue using both end-host's functionsrealized by TPM(trusted platform module) security chip and TTS(time stamp service) provided by trusted third party.

1　はじめに

インシデントレスポンスや法的紛争・訴訟に対して，電磁的記録の証拠保全や調査，分析，電磁的記録の改竄・毀損等についての分析，情報収集等を行う手法・技術として，デジタルフォレンジックがある [1]．本稿では，コンピュータにおいて各種のログを取得して，障害や不正行為の証拠を保全するためのフォレンジックシステムに注目して，ログの法的証明力の要件を考慮したエンドホストのロギングの機構を示す．

フォレンジックシステムでは，コンピュータ上のデジタルデータの取り扱いの痕跡，作業履歴を確保する際，法的紛争や訴訟の場で証拠能力および証明力を認められるための，妥当な仕組みが必要となる．間形らは，デジタル証拠の証明力について，根拠性と安定性の 2 つの概念を導入して，証明の難易度が高い不存在の証明をすることを目標に，これらの証明力を高めるための要件を導いている [2]．芦野らは，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ヒステリシス署名技術を用いて，エンドホストにおいて利用者の操作ログを取得，保全するフォレンジックシステムを提案している [3]．

本稿で示すロギングの機構は，間形らの要件を満足するための仕組みを検討したものであり，また，芦野らのシステムの課題として挙げられているプログラムの安定動作，管理者の不正操作への対処を検討した仕組みを含んでいる．

2 節では，我々が想定するエンドホストにおけるロギング，関連するエンティティについて，間形らのデジタル証拠の法的証明力を高めるための要件，芦野らのフォレンジックシステムについて述べる．3 節では，TPM（trusted platform module）[5] と TSS（time stamp service）[6] を用いたエンドホストのロギング機構として，エンティティ間の連携，動作について述べ，またここで示したロギングの機構に関して考察を与える．

2　準備

2.1　エンドホストにおけるロギング

エンドホストのロギングを，ロギングを実施する監視対象ホストとそのログを集めるログ収集ホストの 2 つをおいて，図 1 のように，ログの取得から提出までの 5 つのフェーズで定義する．

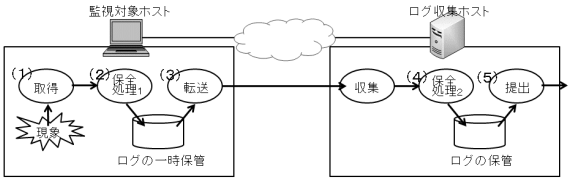

図 1: エンドホストにおけるロギング

(1) 取得フェーズ 現象が発生するデバイス，プログラムをモニタして，ログとして現象の生データやイベントを取得し，それを保全処理 1 フェーズの主体に送る．

(2) 保全処理 1 フェーズ 取得フェーズの主体からログデータを受けとり，保全のための加工を施し，ログデータとして監視対象ホストのストレージに一時的に保管する．

(3) 転送・収集フェーズ 監視対象ホストのストレージからログデータを読み出し，ログ収集ホストに転送，ログ収集ホストでそれを受け取り，保全処理 2 フェーズの主体に送る．

(4) 保全処理 2 フェーズ 収集フェーズの主体からログデータを受け取り，保全のための加工を施し，ログ収集ホストのストレージに保管する．

(5) 提出フェーズ ログ収集ホストのストレージからログデータを読み出し，提出媒体に書き込み，それを提出先の主体に提示する．

監視対象ホストは，組織内の情報システムの利用者が使用するホストであり，ログ収集ホストおよび外部ネットワーク上のホストに常にアクセス可能とする．

ログ収集ホストは，個々の監視対象ホストのログを収集するホストであり，外部ネットワーク上のホストからのアクセスが制限されているものとする．

監視対象ホストを扱う者は，当該組織に所属しており，利用者と管理者が別々の人物である場合，利用者と管理者が同一人物である場合の両方を考慮する．

ログ収集ホストを扱う者は，当該組織に所属しており，この管理はフォレンジックシステムの管理者が負うものとする．

2.2 関連研究

間形らにより，証拠性と安定性という 2 つの概念に基づき，訴訟の際に提出される電子データの法的証明力を高めるための要件が示されている [2].

要件 1 記録により必要な注意義務に従った運用実績を示すこと.
要件 2 故意過失の記録が含まれていないこと.
要件 3 記録の実在を証明できること.
要件 4 記録した主体が何かを証明できること.
要件 5 記録した日時を証明できること.
要件 6 記録の完全性を証明できること.
要件 7 記録の発生契機を証明できること.
要件 8 記録解釈の妥当性を証明できること.
要件 9 記録の正確性を証明できること.
要件 10 記録の網羅性を証明できること.
要件 11 記録保管の継続性を証明できること.
要件 12 記録の整合性があること.
要件 13 異常時の検出と対処が記録されていること.

要件 1,2 は証拠の内容が要証事実の裏づけになること（証拠性）に関する要件であり，要件 3〜13 は訴訟相手から証拠に対する反論を受けても再反論が可能で裁判官の心証に動揺を与えないこと（安定性）に関する要件と定義されている．本稿では，システムによる支援が可能な安定性の要件（要件 3〜7, 要件 9〜11）に注目し，3.3 節で，これを満足するロギングの機構の実現方法について考察する．現象の表現方法や各種ログの関係，異常時の対処に関係する要件 8，要件 12，13 に関しては，今回は議論の対象外とする．

芦野らにより，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ヒステリシス署名技術を用いた，エンドホストにおけるログの取得，保全の方式が示されている [3]．フォレンジックシステムを，コンピュータが管理されたネットワークに接続されたかたちで設置するネットワーク型と，コンピュータを管理されたネットワーク以外で利用するスタンドアロン型の 2 つに分類しており，このうちのスタンドアロン型に適したログ保全方式が示されている．ロガープログラムによって取得されたログのデータを，ログストレージプログラムにより形式化し，これに対してセキュリティデバイスに格納される秘密鍵でデジタル署名を作成する．このとき，過去に署名したデータ（連鎖用データ）と現在の署名対象のデータを結合したものに対して署名処理を行い，連鎖用データによって過去にわたる署名の連鎖構造を構築することで，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ヒステリシス署名 [4] を用いている．ヒステリシス署名では，署名が完了しているログデータを末尾から任意の個数削除した場合，それを検出することができないため，ログデータを末尾から削除するという攻撃が想定され，これを困難にするためにセキュリティデバイスに末尾の連鎖用データを保存するという方法が示されている．芦野らのログ保全方式では，プログラムが不正に変更されないこと，管理者が使用するプログラムは予め指定した人間だけがアクセスできること，コンピュータの管理者は不正を行わないこと，セキュリティデバイスを他人に渡さないことを前提としている．本稿では，3.2 節で，TPM を用いたログのヒステリシス署名を利用しながら，プログラムが不正に変更されない，コンピュータの管理者は不正を行わないとする 2 つの前提を外したとしてもロギングの真正性を確保できる，ネットワーク型のログ保全の仕組みを示す.

3 ロギング機構の提案

3.1 連携するホストの役割と動作

エンドホストのロギング機構と実現するにあたり，連携するホストとその役割を図 2 のようにおく.

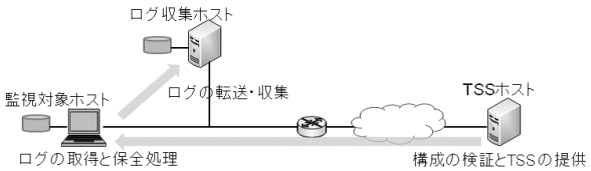

図 2: 連携するホストとその役割

TSS ホストは，信頼できる第三者機関が管理，運用するホストとし，監視対象ホストの構成を検証，ホスト上で意図したエージェントの動作を強制し，ロギングの開始時刻およびそれ以降のロギング，ログが信頼できることを証明するタイムスタンプを発行するものとする.

監視対象ホスト，ログ収集ホスト，TSS ホストの間の連携動作を図 3 のようにおく.

ロギングの準備は，監視対象ホストと TSS ホストの間の動作であり，監視対象ホストにエージェントを導入し，TPM 固有の RSA 鍵，機器構成を第三

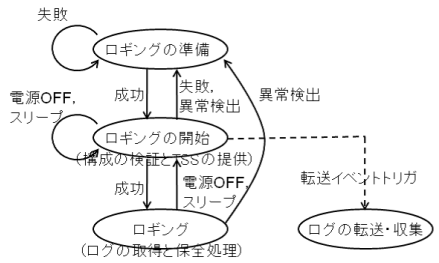

図 3: ホスト間の連携動作

者機関の TSS ホストに登録，ロギングを実施するための連鎖データを共有する動作とする．

ロギングの開始は，監視対象ホストと TSS ホストの間の動作であり，監視対象ホストの TPM を用いて，ホストの構成を検証，エージェントの動作を確認し，ログのヒステリシス署名の起点となる連鎖データに対して，タイムスタンプを発行する動作とする．

ロギングは，監視対象ホストにおいて実施される動作であり，監視対象ホストの TPM を用いて，ロギングの開始において TSS ホストにより署名を受けた連鎖データを用いてログのヒステリシス署名を逐次行い，ログの取得，保全処理を実施する動作とする．

ログの転送・収集は，監視対象ホストのストレージの空スペースを確保し，ロギングの継続性を維持するための動作であり，監視対象ホストにより保全処理されたログをログ収集ホストで集めて，保管する動作とする．

3.2 動作の詳細

エンドホストのロギングの機構について，ロギングの準備，ロギングの開始，ロギングのそれぞれの動作を示す．ログの転送・収集については，本稿では省略する．

ロギングの準備

(1) 監視対象ホストの管理者は，TPM を有効化，セットアップ（TPM オーナの取得）して，ロギングのエージェントを導入する．

(2) 監視対象ホストの管理者は，TPM の AIK（attestation identity key）のペア AIK_{pub}, AIK_{sec} を作成，TPM 内に格納する．AIK は，TPM チップ固有の鍵である EK（endorsement key）と紐付けされたかたちで生成される．

(3) 監視対象ホストの管理者は，AIK_{pub} と，EK（endorsement key）証明書 EK_{cer}，構成情報 $Host_{inf}$ を TSS ホストに送る．EK 証明書は TPM チップのベンダから発行されるものとする．

(4) TSS ホストは，EK_{cer} を検証，AIK_{pub} が TPM 依存の鍵であることを確認する．正しい場合は，AIK 証明書 AIK_{cer} と初期連鎖データ C_{ini} を作成する．その他の場合は，処理を中断する．

(5) TSS ホストは，AIK_{pub} と AIK_{cer} を証明書 DB へ，$Host_{inf}$ を構成情報 DB へ，C_{ini} を対象データ DB へ格納し，監視対象ホストに AIK_{cer} と C_{ini} を送る．

(6) 監視対象ホストは，$i \leftarrow 0, j \leftarrow 0$ として，初期ログレコード $L_{i,j} = \{i, j, null, null, C_{ini}, null, null\}$ と AIK_{cer} を自身のストレージに，C_{ini} を TPM 内に格納して，ロギングの開始に移行する．

ロギングの開始（図 4）

(1) 監視対象ホストが起動されると，CRTM（core root of trust for measurement），BIOS，OS ローダ，OS，エージェントの順番でプログラムが動作し，TPM 内の PCR（platform configuration register）に，各プログラムから計算したダイジェストが格納される．

(2) 監視対象ホストのエージェントは，TPM 内の PCR の値 R と AIK 秘密鍵 AIK_{sec} から署名 $S = Sign(Hash(R), AIK_{sec})$ を計算し，Integrity report を作成して，TSS ホストの PTV（platform trust verifier）に送る．

(3) TSS ホストの PTV は，Integrity report を検証，監視対象ホストの構成が信頼できるかどうかを確認する．登録済みの構成である場合は，TSA （time stamp authority）にサービスを許可する．その他の場合は，処理を中断する．

(4) 監視対象ホストのエージェントは，TPM 内から連鎖データ C_0 と Tick カウンタの値 T_0 を取

り出し，タイムスタンプの対象データとして TSS ホストの TSA（time stamp authority）に送る．構成検証に失敗した場合は，ロギングの準備に移行する．

(5) TSS ホストの TSA は，TA（time authority）から時刻 CT を取得して，C_0 と T_0 と CT と自身の秘密鍵 K_{sec} から署名 $TS = Sign(Hash(C_0||T_0||CT), K_{sec})$ を作成して，これらを対象データ DB に格納し，TS を監視対象ホストのエージェントに送る．

(6) 監視対象ホストのエージェントは，$j \leftarrow 0, i \leftarrow i{+}1$ として，ログレコード $L_{i,j} = \{i, j, T_0, CT, C_0, TS, null\}$ を格納して，ロギングに移行する．

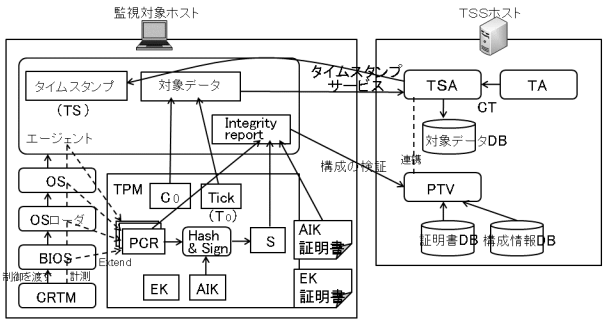

図 4: ロギングの開始（構成の検証と TSS）

ロギング（図 5）

(1) 監視対象ホストのエージェントは，デバイスの現象を取得，ログデータ D_{j+1} を作成して，そのハッシュ値 $H_{j+1} = Hash(D_{j+1})$ を計算する．

(2) H_{j+1} と TPM 内の Tick カウンタの値 T_{j+1} と連鎖データ C_j と AIK 秘密鍵 AIK_{sec} から署名 $S_{j+1} = Sign(Hash(T_{j+1}||H_{j+1}||C_j), AIK_{sec})$ を作成する．

(3) S_{j+1} と T_{j+1} と H_{j+1} と C_j から，ハッシュ値 $C_{j+1} = Hash(S_{j+1}||T_{j+1}||H_{j+1}||C_j)$ を計算して，TPM 内の C_j を C_{j+1} で置き換える．

(4) $j \leftarrow j{+}1$ として，ログレコード $L_{i,j} = \{i, j, T_j, D_j, C_j, S_j, AIK_{cer}\}$ をストレージに格納して，ロギングを継続する．電源 OFF やスリープの際は，ロギングの開始に移行する．

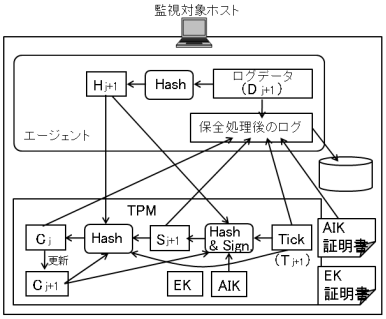

図 5: ロギング（ログの取得と保全処理）

監視対象ホストに格納される保全処理後のログデータは，ログ収集ホストへのログの転送，インシデントや法的訴訟が発生した際のログの提出の場面で，次のように検証される．

ログの検証

(1) 検証者は，初期ログレコード $L_{0,0}$ から，初期連鎖データ C_{ini} を取り出し，これが TSS ホストの対象データ DB に登録されているかどうかを確認する．登録されていない場合，初期ログレコードの改ざんを検出，処理を中断する．

(2) 検証者は，$i = 1, 2, \ldots$ について，次の処理を行う．ログレコード $L_{i,0}$ から，Tick カウンタの値 T_0 と TSS ホストから得た時刻 CT，連鎖データ C_0，TSS ホストから得たタイムスタンプ TS を取り出し，TSS ホストの公開鍵 K_{pub} を取得して，次式を確認する．式が成立しない場合は，ログレコードの改ざんを検出，処理を中断する．

$$Hash(C_0||T_0||CT) \stackrel{?}{=} Veri(TS, K_{pub})$$

ログレコード $L_{i,j}, j = 1, 2, \ldots$ から，Tick カウンタの値 T_j とログデータ D_j と連鎖データ C_j と署名 S_j を取り出し，当該の監視対象ホストの AIK 公開鍵 AIK_{pub} を取得して，次式を確認する．式が成立しない場合は，ログレコードの改ざんを検出，処理を中断する．

$$Hash(S_j||T_j||Hash(D_j)||C_{j-1}) \stackrel{?}{=} C_j,$$
$$Hash(T_j||Hash(D_j)||C_{j-1}) \stackrel{?}{=} Veri(S_j, AIK_{pub}).$$

3.3 考察

本稿で示したロギングの機構は，ロギングの開始時に，監視対象ホストのTPMを用いたtrusted bootにより，信頼できる第三者機関により構成を検証し，構成が正しい場合にログのヒステリシス署名を作成するための連鎖データに対してタイムスタンプを発行する．これにより，監視対象ホストの上で，信頼できるエージェントの動作を強制するとともに，監視対象ホストのTPM内のTickカウンタとTSSホストの時刻同期が可能となり，デジタル証拠の法的証明力を高める要件の中の，記録取得の日時（要件5），記録の発生契機（要件7），正確性（要件9），網羅性（要件10）の証明が可能になると考えられる．

監視対象ホストで取得したログの保全処理は，ログデータに，TPMのTickカウンタの値を付加したものに対して，TPMのAIK秘密鍵で署名を作成することにより，記録の実在（要件3），主体（要件4），完全性（要件6）の証明が可能になると考えられる．ただし，第三者機関により構成の検証を行ったときのPCRの値と，エージェントの実行時点でのPCRの値にはタイムラグがあり，PCRを取得した時点からエージェントが攻撃を受けると，検証したPCRの値と現在のシステムの状態が必ずしも一致するとは限らない．このことについては，ホストの管理者であっても操作が限定されているカーネルモードプログラムにより，エージェントプログラムへの不正操作を監視することで，対処することを考えている．

記録保管の継続性の要件（要件11）に関しては，監視対象ホストのストレージの空きが無くなる前に，ログ収集ホストにログ転送要求を出し，適切にスケジューリングした上で，ログ収集ホストへログレコードを転送することが考えられる．監視対象ホストのストレージの空き容量が不足していたり，ログ収集ホストの負荷が高いためにログレコードの転送が困難な場合は，デバイスをロックしてログ生成を停止させることで，監視対象ホストの記録保管の継続性（要件11）を達成することが考えられる．

本稿で示したロギングの機構では，監視対象ホストの中にロギングのエージェントを導入しないまま，TSSホストによる構成の検証を受けずに，組織内にホストを設置し，組織内の情報システムにアクセスするような場合が考えられる．このことについては，監視対象ホストを組織内のネットワークに接続する際に，監視対象ホストの構成を検証し，ロギングを実施するエージェントプログラムが導入されたホストだけに接続を許可する仕組みが考えられる．

4 おわりに

本稿では，コンピュータにおいて各種のログを記録して，障害や不正行為の証拠を保全するためのフォレンジックシステムに注目し，TPMとTSSを用いたエンドホストのロギングの機構を提案した．

インシデントレスポンスの他に，法的紛争・訴訟への対応を想定した場合，エンドホストでのロギングには，デジタル証拠の法的証明力を高める要件を考慮した仕組みが求められる．

提案したロギングの機構は，エンドホストのTPMのtrusted bootを利用して信頼できる第三者機関によりエンドホストの構成を検証することで，エンドホスト上で意図したエージェントの動作を強制し，エンドホストの管理者であってもログの削除や改ざんを困難にしている．

今後の課題として，本稿で示したロギングの機構を実装し，エンドホストのロギングの負荷の評価，第三者機関で提供する構成検証とTSSの負荷の評価，エンドホストのログの転送方式の検討と実装，評価等が挙げられる．

参考文献

[1] 佐々木良一，芦野祐樹，増渕孝延，“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.46-61，2006年．

[2] 間形文彦，高橋克巳，金井敦，“デジタル証拠の法的証明力を高めるための要件に関する一考察，”電子情報通信学会 SCIS2008予稿集，4E1-6，2008年1月．

[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システムの提案と評価，”情報処理学会論文誌，Vol.49，No.2，pp.999–1009，2008年．

[4] 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年．

[5] TCG，Trusted Platform Module(TPM)，*http://www.trustedcomputinggroup.org/developers/trusted_platform_module*

[6] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato, “Internet X.509 Public Key Infrastructure Time-Stamp Protocol(TSP),” IETF RFC3161, 2001.