情報処理学会論文誌　データベース　Vol. 4　No. 4　63–77 (Dec. 2011)

端末操作ログからの情報漏えい検出

豊田　真智子[†1]　櫻井　保志[†1]
小林　透[†2]　市川　裕介[†2]

近年，内部犯行による企業や自治体などの組織内情報の漏えいが社会問題となっており，その抑止対策としてユーザの端末操作ログを定期的にセキュリティ監査する方法が注目されている．現状のセキュリティ監査は，一定期間のログをサンプリングにより選択し，それらを調査する方法で実施されており，その多くが監査人のスキルに依存している．監査ログ量の増加により監査人への負担が増大するため，効率的に監査を行う手法が求められる．そこで本論文では，監査人の支援を行う危険行動検出方式を提案する．本方式は，危険行動に該当するパターンを端末操作ログから高速に検出するものであり，監査対象ログの全件探索を達成する．本方式を自治体職員の端末操作ログに適用し，効率的に危険行動パターンを検出することが確認された．

Information Leak Detection on Terminal Operation Logs

Machiko Toyoda,[†1] Yasushi Sakurai,[†1]
Toru Kobayashi[†2] and Yusuke Ichikawa[†2]

Information leak by internal threats is a big problem in companies and local governments. To deter them, security audits of terminal operation logs are attracting an increasing amount of interest. The usual security audit procedure is to sample logs at certain periods and then to check them visually. However, analysis accuracy depends on the skill of the auditor and the massive growth in the number of audited logs now threatens to overwhelm the auditors. We present an effective method that can automatically discover dangerous behavior that predicts an information leak. Our method detects the patterns representative of dangerous behavior, at high speed, and covers logs of all scales. We empirically demonstrate its usefulness on terminal operation logs collected at a local government and confirm that our method can correctly discriminate the patterns.

1. まえがき

近年，企業や自治体などの個人情報の流出が後を絶たず，組織内の情報漏えいが社会問題となっている．これまで，ネットワーク機器やサーバに対する外部からの不正アクセス対策が進められる一方[26),34),43)]，最近の情報漏えいの原因の多くは内部犯行によるものであるという傾向が指摘されている[21),41)]．これらの内部犯行に対して，たとえ外部記憶媒体の利用制限や印刷の管理を実施したとしても，システム的に完全に防止することは困難であり，正当なアクセス権限を持つユーザに対する抑止策が重要となる．

2008 年に施行された日本版 SOX 法により，企業におけるログ管理の重要性が唱えられ，様々なログが蓄積されるようになった．これらのログを効率的に管理するためのログ収集ツールも多く市販されており[1),5)]，組織内部からの情報漏えいに対し，セキュリティポリシに違反するような，明らかなルール違反の検出によりレポートを作成する機能が備わっている．しかしながら，セキュリティポリシに違反しない通常業務に紛れた情報漏えいを検出することは非常に困難となっており，問題が発生した後に監査人が過去のログをチェックして原因を調査するという程度にとどまっている．

このような状況から，内部統制や運用の可視化を目的としてユーザの端末操作ログが注目されている．具体的な対策としては，これらのログを定期的にセキュリティ監査することが求められている[42)]．これまで，単一の違反行動判定や違反行動の実行回数が閾値を超えたかどうかの判定によって，危険行動の検出が行われてきた．しかしながら，セキュリティ監査において，通常業務にまぎれた危険行動を特定するためには，一連の操作行動単位でログをとらえ，そのうえで通常業務と危険行動を区別して判定することが必要となる．これは，ログを個々のイベント単位で解析するのではなく，複数のイベントの時間的順序を考慮した行動パターンとして解析することを意味する．ここで，イベントとはユーザの端末操作にともなってログとして記録される行動の証跡を指す．現状のセキュリティ監査では，特定期間のログをサンプリングにより抽出し，危険行動につながる可能性のある特定のイベント（たとえば，印刷）を検索する．そして，それらの前後のイベントも含めた一連の行動パターンを目視により解析することで，監査人が危険行動かどうかを判定してきた．この方法は，監

†1 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, NTT Corporation
†2 日本電信電話株式会社 NTT サイバーソリューション研究所
NTT Cyber Solutions Laboratories, NTT Corporation

査人の人的稼動に大きく依存しており，すべてのログを監査対象とすることを困難にしている．また，人員増員を行う場合においても，解析判断のノウハウが共有されておらず，監査結果にばらつきがでてしまうという課題を残している．

そこで本論文では，セキュリティ監査において監査人の支援を行うための危険行動検出方式を提案する[44]．本方式は，端末操作ログを記号シーケンスとして表し，危険行動パターンに該当するパターンを高速に検出するものである．パターンの検出においては，文献 47) の手法を記号シーケンスのために改良したアルゴリズムを使用し，すべての操作ログを対象とした解析を行う．提案方式の有効性を評価するため，自治体職員から収集した端末操作ログを対象に危険行動パターンの検出を行った．その結果，高い精度で効率的に該当するパターンを検出することが確認された．

本論文の構成は以下のとおりである．まず 2 章において，危険行動検出のためのセキュリティ監査について述べ，方式に求められる要件を整理する．3 章で問題設定を行い，4 章で危険行動検出方式について述べる．5 章で提案方式の実験を行い，その有用性を示し，6 章で関連研究について議論する．最後に 7 章でまとめを述べる．

2. 危険行動パターン検出のためのセキュリティ監査

セキュリティ監査システムに関する検討が進められ[32),39)]，ユーザの端末操作ログを対象とするツールも増加している[2)–4)]．また一方で，端末操作ログを Web の操作支援[45)] や業務行動モニタリング[46)] のために解析する研究も見られ，端末操作ログの分析は非常に重要となってきている．しかしながら，通常業務と危険行動を判断したり，セキュリティの観点で行動パターンの分析を行ったりするツールや研究はほとんど行われていない．

現在，情報漏えいにつながる危険行動パターンを検出するためのセキュリティ監査は，以下の手順で進められている（**図 1**）．

(1)　端末操作ログ収集
　ログ収集ツールにより全端末の操作ログを収集する．

(2)　監査対象ログの決定
　1 カ月などの監査期間を決定し，サンプリングによりその期間中の監査対象端末操作ログを決定する．

(3)　危険行動の絞り込み
　添付ファイルのあるメール送信などの個別の危険イベントを検索ツールなど利用して特定し，そのイベントの前後を含めた端末操作ログを抽出する．

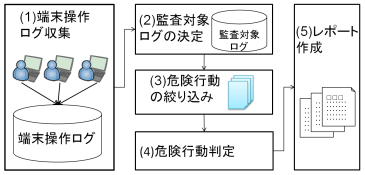

図 1　従来のセキュリティ監査手順
Fig. 1　Procedure of traditional security audit.

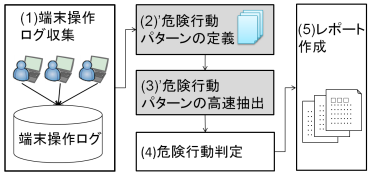

図 2　提案するセキュリティ監査手順
Fig. 2　Procedure of proposed security audit.

(4)　危険行動判定
　抽出した端末操作ログを目視によりチェックし，通常行動か危険行動かを判定する．

(5)　レポート作成
　監査結果を集計し，適合性報告やガバナンス違反アラートをレポート化する．

上記から確認されるとおり，危険行動パターンの分析においては，監査人のスキルに負う部分が多い．特に，危険行動の絞り込み (3) とその判定 (4) においては，監査人の大幅な労力を必要とする．そこで，これらを機械的に処理するためには，従来のセキュリティ監査を，**図 2** のようなセキュリティ監査に置き換える必要がある．最初に，監査人のノウハウとして蓄えられている危険行動のパターンを統一的に定義する (2)'（これを定義パターンと呼ぶ）．これは，単一の行動のみでなく，その前後にどのような行動が行われていた場合に危険行動と判断するかを詳細に示すものである．定義パターンの各要素が 1 つ 1 つの行動を表しており，自然言語で表記される．定義パターンが決定されると，定義パターンに該当する危険行動パターンを検出アルゴリズムにより自動的に検出する (3)'．そして，検出されたパターンが真の危険行動であるかの判定を監査人が行う (4)．ここで監査人が判定するパターンは，行動パターン単位で検出されたものであるため，従来のセキュリティ監査時の判定より候補数は大幅に少なく，判定コストの削減が見込まれる．

危険行動パターンの検出においては，以下の 2 つのケースを考慮する必要がある．

(1)　実際の危険行動パターンが定義パターンとほぼ一致している．

(2)　実際の危険行動パターンが定義パターンの一部と一致している．

ケース (1) において重要となるのは，定義パターンで定義されている以外のイベント，すなわち，ノイズイベントを考慮することである．たとえば，新規にメールを作成し，送信するという一連の行動が定義パターンであるとする．これらの行動の合間に行う，Web を閲覧する，ファイルを作成するなどの行動は，定義パターンとは関係のないノイズイベントである．ユーザによって，様々なノイズイベントが挿入される可能性があるため，検出におい

てノイズイベントにロバストであることが求められる．しかしながら，ケース (1) はほぼ理想的な検出であり，悪意を持つユーザが情報を持ち出そうとした場合，定義パターンとよく似た行動パターンをとるとは限らない．そのため，実環境においては，ケース (2) となることが多いと予想される．もしこれらの行動パターンを排除すると，本来検出すべき多くのパターンを検出漏れさせてしまうことになるため，定義パターンの一部に一致する行動パターンも検出対象とすべき必要がある，

上記で示した 2 つのケースを考慮することにより，定義パターンに類似するより多くのパターンを検出することができるようになった．しかしながら，検出したパターンが危険行動に該当するかどうかを最終的に判定するのは監査人であり，監査人が意図しない行動パターンを検出することは，判定の負荷を増やしかねない．そこで，検出に際し，さらに次の 2 つを考慮する．まず第 1 に，定義パターンへの適合度を導入する．定義パターンの一部に該当するパターンの中には，その適合度合いがかなり低いパターンも含まれる．これは特に定義パターンが長い場合に，その中のごく短い一部に該当するパターンが検出されることを意味する．適合度合いは危険行動か否かを判定するための重要な指標であり，定義パターンに合わせて監査人が自由に設定できることが望ましい．第 2 に，一連の行動を表す端末操作ログからは，最も定義パターンに類似する行動パターンのみを検出する．定義パターンのどの要素に該当するかにより，同一の行動を表すパターンが複数検出される可能性がある．そのため，最も類似度が高い行動パターンのみを検出することにより，重複する危険行動パターンを排除する．

以上より，図 2 のセキュリティ監査を実現するための必要要件は，次のとおりである．

- 要件 1：ノイズイベントに対するロバスト性
- 要件 2：定義パターンとログの部分的な適合の許容
- 要件 3：監査人のための適合度合いの設定
- 要件 4：重複する行動パターンの排除

そこで我々は，監査人の支援を目的とした監査の半自動化のための危険行動検出方式を提案する．提案方式は，上記で述べた要件を満たし，図 2 のセキュリティ監査を実現するものである．危険行動の検出には，記号シーケンス間の類似度を測定する編集距離[30),36)] を利用する．編集距離は，バイオインフォマティクス[15)] や音声処理[20)]，文字認識[33)] など，様々な分野で利用されている有用な距離尺度である．定義パターンの要素と端末操作ログのイベントを記号として表し，記号のシーケンスとして編集距離で処理することで，ログの時間的な順序性を考慮した危険行動パターンの検出が可能となる．編集距離は 2 つの記号シーケンス全体を比較するものであり，定義パターンと端末操作ログの部分的な適合を行うためには処理時間が大幅に増加する．一方，シーケンス間の部分的な適合を行う手法が文献 47) で提案されているが，時系列データを対象としており，本論文の問題に適用することが困難である．そこで，文献 47) で提案されている手法を編集距離に基づくアルゴリズムに改良し，記号間の部分シーケンス検出を可能とする手法を考案した．まず次章において，編集距離について述べ，具体的に問題を定義する．

3. 準　　備

3.1 編 集 距 離

2 つの記号シーケンスが与えられたとき，これらの類似度を計算するために用いられるのが，編集距離[30),36)] である．これは，2 つの記号シーケンス X，Y 間の距離を，X を Y に変換するために必要な編集操作（挿入，削除，置換）に要するコストの最小値として表す．距離計算には動的計画法が用いられ，レーベンシュタインによって提案されたアルゴリズム[24)] が最も代表的である．

長さ n の記号シーケンス $X = (x_1, \ldots, x_t, \ldots, x_n)$ と長さ m の記号シーケンス $Y = (y_1, \ldots, y_i, \ldots, y_m)$ を考える．これらの編集距離 $D(X,Y)$ は，次のように計算される．

$$D(X,Y) = d(n,m)$$

$$d(t,i) = min \begin{cases} d(t,i-1) + \gamma(\phi \to y_i) \\ d(t-1,i) + \gamma(x_t \to \phi) \\ d(t-1,i-1) + \gamma(x_t \to y_i) \end{cases} \quad (1)$$

$$d(0,0) = 0, \quad d(t,0) = t \ (t = 1, \ldots, n), \quad d(0,i) = i \ (i = 1, \ldots, m)$$

ここで，$\gamma(\phi \to y_i)$ は挿入，$\gamma(x_t \to \phi)$ は削除，$\gamma(x_t \to y_i)$ は置換の各コストを表す．通常，挿入，削除コストは $\gamma(\phi \to y_i) = \gamma(x_t \to \phi) = 1$，置換コストは $x_t = y_i$ であれば $\gamma(x_t \to y_i) = 0$，そうでなければ $\gamma(x_t \to y_i) = 1$ が用いられる．2 つのシーケンスが適合しているかどうかの判定は，距離の閾値 ε を用いて行われる．すなわち，

$$D(X,Y) \leq \varepsilon \quad (2)$$

を満たすとき，シーケンス X と Y は適合していることになる．

図 3 は，編集距離を計算する行列を例示したものである．シーケンス $X = (FTCGTFC)$ は 1 回の挿入操作と 2 回の削除操作を得て $Y = (FTGTTF)$ に一致するため，編集距離は 3 となる．図において色付けしたセルは，距離を計算するために対応付けられた要素のパス（適合パスと呼ぶ）を示す．編集距離を求めるために必要とする行列は $(n+1)(m+1)$ 個の

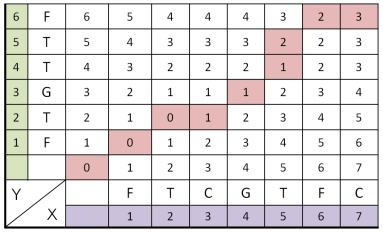

図 3　編集距離による記号列マッチング
Fig. 3　Illustration of symbol matching with edit distance.

セルから構成されるため，その計算時間は $O(nm)$ となる．メモリ使用量については行列の 2 列（現在の列と直前の列）だけで計算可能であるため，$O(m)$ または $O(n)$ となる．

3.2 問題設定

長さ n の端末操作ログ $X = (x_1, \ldots, x_t, \ldots, x_n)$ と長さ m の定義パターン $Y = (y_1, \ldots, y_i, \ldots, y_m)$ を考える．編集距離を用いた危険行動パターンの検出は，以下の条件を満たす部分シーケンスを検出することである．

$$D(X[t_s : t_e], Y) \leq \varepsilon \tag{3}$$

$X[t_s : t_e]$ は t_s 番目から t_e 番目までの X の部分シーケンス（すなわち，危険行動パターン）であり，$D(X[t_s : t_e], Y)$ は $X[t_s : t_e]$ と Y 間の編集距離を表す．ここで，ノイズイベントによる影響を考慮する（要件 1）．式 (1) から確認されるとおり，ログにノイズイベントが混入している場合，削除コストが追加されることにより全体の編集距離は増加する．すなわち，ノイズイベントの後に定義パターンに該当するイベントが連続していたとしても，ノイズイベント混入前までの距離の方が小さくなる．そのため，短い部分シーケンス，つまり，短い行動パターンの方が長い行動パターンより検出されやすくなってしまう．ノイズイベントが含まれている場合でも，全体的に定義パターンに近い行動パターンを検出することが好ましいため，この設定はノイズイベントへのロバストさを欠くものである．そこで，本論文においては，次の条件を満たす部分シーケンスペアを検出する．

$$D(X[t_s : t_e], Y) \leq \varepsilon l_x \tag{4}$$

l_x は X の部分シーケンスの長さを表し，$l_x = t_e - t_s + 1$ である．これにより長さに比例して閾値が設定されるため，短い部分シーケンスペアが検出されやすくなるのを防ぐことができる．

次に，危険行動パターンの検出の精度を高めるため，定義パターンの一部に該当するパターンを検出可能とする（要件 2）．

$$D(X[t_s : t_e], Y[i_s : i_e]) \leq \varepsilon L(l_x, l_y) \tag{5}$$

$Y[i_s : i_e]$ は i_s 番目から i_e 番目までの Y の部分シーケンスであり，その長さは $l_y = i_e - i_s + 1$ である．L は 2 つの部分シーケンスペアの長さを規定する関数であり，$X[t_s : t_e]$ と $Y[i_s : i_e]$ の両方の長さを考慮した検出を可能とする．本論文では，2 つの部分シーケンスの平均長である $L(l_x, l_y) = (l_x + l_y)/2$ を用いる[⋆1]．

式 (5) により，部分シーケンスの長さに依存することなく適合する部分シーケンスペアを検出できるようになった．ここでさらに，適合する部分シーケンスペアの長さを導入する（要件 3）．これは，監査人が意図しない，適合度合いの低いパターンを排除するためのものであり，セキュリティ監査における監査人の利便性を高めるものである．適合する部分シーケンスペアの長さを l_{min} とすると，以下の条件を満たす部分シーケンスペアを検出する．

$$D(X[t_s : t_e], Y[i_s : i_e]) \leq \varepsilon (L(l_x, l_y) - l_{min}) \tag{6}$$

部分シーケンスペアの長さ $L(l_x, l_y)$ が l_{min} より小さい場合，右辺は負の値となる．部分シーケンスペア $X[t_s : t_e]$ と $Y[i_s : i_e]$ が完全に一致した場合の距離は 0 であり，距離が負の値となることはない．そのため，l_{min} より短い部分シーケンスペアが検出されることがないことが保証される．

一方，部分シーケンスペア $X[t_s : t_e]$ と $Y[i_s : i_e]$ が適合するとき，距離が最小値となる部分シーケンスペアと重複する多くの部分シーケンスペアが存在しており，これらも適合してしまう可能性がある（要件 4）．ここで，重複するとは部分シーケンスペア間の適合パスが交わることであり，次のように定義される．

定義 1　X と Y の部分シーケンスペアの適合パスが与えられたとき，重複とはそれらの適合パスの少なくとも 1 つの要素が共有されることであり，重複する部分シーケンスペアとは，適合パスの要素が共有された部分シーケンスペアのグループを指す．

これらをすべて検出することは，冗長な検出結果を提示するだけでなく，監査人の危険行動判定の効率を低下させる可能性がある．そのため，これらを排除し，編集距離が最小値となるペア，すなわち，最も定義パターンに類似する行動パターンのみを検出する．

以上より，本論文において解決したい問題は以下のように定義される．

問題 1　端末操作ログ X と定義パターン Y，距離の閾値 ε，適合する部分シーケンスペ

⋆1 関数 L は，2 つの部分シーケンスの最大値である $L(l_x, l_y) = max(l_x, l_y)$ や最小値である $L(l_x, l_y) = min(l_x, l_y)$ を使用することも可能である．

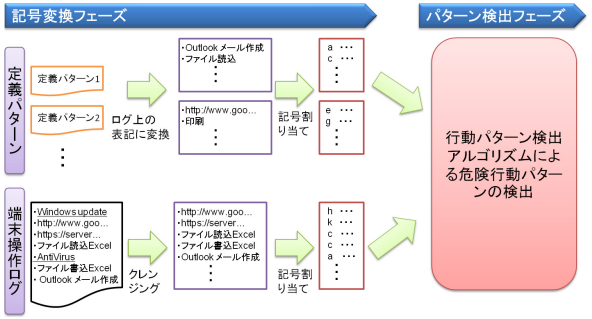

図 4　処理の流れ
Fig. 4　Flow of processing.

アの長さの閾値 l_{min} が与えられたとき，次の条件を満たす部分シーケンスペア $X[t_s : t_e]$ と $Y[i_s : i_e]$ を検出する．

(1)　$X[t_s : t_e]$ と $Y[i_s : i_e]$ が式 (6) を満たす．

(2)　重複する部分シーケンスペアの中で，$D(X[t_s : t_e], Y[i_s : i_e]) - \varepsilon(L(l_x, l_y) - l_{min})$ が最小値をとる．

これ以降，問題 1 の条件 (1) を満たすものを適合する部分シーケンスペア，(1) と (2) の条件を満たすものを最適な部分シーケンスペアと呼び，特に (1) と (2) の条件を満たすペアを危険行動パターンとして検出する．

本論文で提案する危険行動検出方式は，記号変換とパターン検出の 2 つのフェーズで構成される（**図 4**）．本章で示した問題は，パターン検出フェーズにおいて解決する問題であり，記号変換フェーズでは，検出を効率的に進めるための処理を担う．次章において，各処理の詳細を述べ，どのように危険行動パターンを検出するのかを示す．

4. 危険行動検出方式

4.1 記 号 変 換

記号変換フェーズでは，テキスト形式で表記された定義パターンと端末操作ログを記号シーケンスに変換し，機械的に危険行動パターンを検出するための前処理を行う重要なフェーズである．変換にあたり，まず最初に定義パターンをログ上の表記に対応付ける処理を行う．これは，定義パターンの各要素（すなわち，行動）がどのようなログとして取得されるかを厳密に調べていくことを意味する．行動によっては複数のログとして記録されるものもあり，この場合，1 つの行動が複数のイベントに対応付けられていると判断し，それらすべてを定義パターンの要素に含める．

表 1　属性値の例
Table 1　Examples of attribute values.

収集するログの種類	属性値	属性値の詳細
共通	端末エージェント名	端末エージェントの名前
	マシン名	端末の名前
	ログオンユーザ ID	ログオンしているユーザ名
	時刻	各操作の時間
	ログ種別	収集するログの種類
アクティブウィンドウ	ウィンドウタイトル	アクティブになっているウィンドウのタイトル
	EXE パス	アプリケーションの実行パス
	アプリケーション	実行中のアプリケーション名
ファイルアクセス	ファイル操作	ファイルに対して行われた操作
	ファイル名	操作されたファイルの名前
	EXE パス	アプリケーションの実行パス
Web アクセス	ウィンドウタイトル	Web アプリケーションのウィンドウのタイトル
	URL	アクセスした URL
	アプリケーション	Web アプリケーション名
送受信メール	プロトコル	使用したプロトコル
	メールタイトル	メールのタイトル
	メールアドレス	送受信のメールアドレス
	添付ファイル	添付ファイル情報
印刷対象	ドキュメント名	印刷したドキュメントの名前
	印刷対象	印刷に使用したプリンタやアプリケーション名
	印刷部数	印刷した部数
	印刷種別	モノクロ印刷かカラー印刷かの情報
	出力サイズ	印刷した用紙のサイズ

次に，ログ上の表記に対応付けられた各要素に記号を割り当てることによって記号シーケンスに変換する．端末操作ログは複数の属性値から構成される．本論文で使用する属性値の例を**表 1** に示す．収集するログの種類が「共通」に含まれる属性値は各ログの収集において必ず記録される属性値であることを意味する．これらの属性値のうち，定義パターンに含まれる属性値に対して，各項目を記号に変換するルールを設定する（ルールの設定例については 5.2 節で示す）．その後，設定したルールに基づき，定義パターンの各要素に対し，属性値ごとに記号を割り当てていく．このとき変換された定義パターンは，属性値の数に対応

する多次元の記号シーケンスとなる⋆1．なお，属性値名は，ログを収集するツールによって異なる場合があるが，その場合においても同様に処理することで記号変換可能である．

定義パターンの変換後，端末操作ログの変換に移る．まず最初に，ログのクレンジングを行う．これは，ユーザの操作によらずに収集されたログの削除を行うことであり，このログには，システムに起因するものやユーザが意図せずに実行されたアプリケーションのログ（ウィルスチェックやスクリーンセーバなど）が含まれる．クレンジングしたログは，定義パターンと同様の変換ルールに基づき，記号を割り当てていく．このとき，変換ルールの存在しない属性値についてはすべて削除することで，定義パターンと同次元のデータを作成する．

以上の処理を経て，定義パターンと端末操作ログの記号変換処理が完了する．次に，これらのデータを入力として，行動パターンの検出処理を行う．

4.2　行動パターン検出

行動パターン検出フェーズでは，編集距離に基づく検出アルゴリズムにより，行動パターンを検出する．アルゴリズムにおいて使用するのは，純粋な編集距離ではなく，計算の効率化を目的とした独自の距離計算である．これは，スコア関数と開始点行列によって実現され，$O(m^2n^2)$ の計算コストを $O(mn)$ にまで削減する．

端末操作ログ $X = (x_1, \ldots, x_t, \ldots, x_n)$ と定義パターン $Y = (y_1, \ldots, y_i, \ldots, y_m)$ が与えられたとき，これらの部分シーケンスペアを見つけるためには，端末操作ログ，定義パターンともに 1 つずつずらした部分シーケンスを作成し，これらのすべての組合せについて，編集距離を計算する必要がある．そのため，編集距離を求める行列は $O(mn)$ 個必要となり，各行列ごとに $O(mn)$ 個の値を更新するため，全体で $O(m^2n^2)$ もの計算時間が必要となる（具体的な計算方法は付録 A.1 で示す.）．そこで，これらの計算を効率化するために，直接編集距離を求めるのではなく，スコア関数を用いて間接的に編集距離を計算する．スコア関数は，編集距離と同様に動的計画法に基づいて類似度の計算を行うが，結果をスコアとして出力する点が異なる．スコアとは類似度を計算する行列（スコア行列と呼ぶ）の各要素の累積値として表され，スコア値が大きいほど類似度が高いという性質を持っている．部分シーケンスペア $X[t_s : t_e]$ と $Y[i_s : i_e]$ 間のスコア $V(X[t_s : t_e], Y[i_s : i_e])$ は，次のように計算される．

$$V(X[t_s : t_e], Y[i_s : i_e]) = v(t_e, i_e)$$

$$v(t,i) = max \begin{cases} 0 \\ \varepsilon b_v - \gamma(\phi \to y_i) + v(t, i-1) \\ \varepsilon b_h - \gamma(x_t \to \phi) + v(t-1, i) \\ \varepsilon b_d - \gamma(x_t \to y_i) + v(t-1, i-1) \end{cases} \tag{7}$$

$$v(0,0) = v(t,0) = v(0,i) = 0$$

スコア関数は，距離の閾値 ε と編集操作コスト（挿入，削除，置換）の差を累積することにより決定されるため，部分シーケンスペアが条件を満たさない場合には，スコアは負の値を示す．そのため，不適合な部分シーケンスペアをスコアの値から判断することができる．また，負の値となった要素のスコアをゼロでリセットすることで，その要素から新たにスコア計算を開始する．これにより，不適合な部分シーケンスペアを枝刈りし，適合する可能性の高い部分シーケンスペアのみを効率的に計算することができる．ここで，b_v，b_h，b_d は部分シーケンスの長さの重みを表している．本実験においては，$L(l_x, l_y) = (l_x + l_y)/2$ を使用するため，$b_v = b_h = 1/2$，$b_d = 1$ となる．これは，垂直または水平方向の要素が引き継がれた場合，シーケンス長は 1/2 増加し，対角方向の要素が引き継がれた場合は 1 増加するためである．スコア行列の中での各パス上の重み (b_v, b_h, b_d) の合計は，$L(l_x, l_y)$ と等しくなるように設計されているため，スコアと編集距離の変換が可能であることが保証される（詳細な証明は付録 A.3 で示す）．部分シーケンスペアの編集距離は，次の式によりスコアから計算される．

$$D(X[t_s : t_e], Y[i_s : i_e]) = \varepsilon L(l_x, l_y) - V(X[t_s : t_e], Y[i_s : i_e]) \tag{8}$$

スコア関数では，スコア行列の各セルに部分シーケンスペアのスコアとその終了点 t_e と i_e の情報が保持される．どの点から適合したかという，部分シーケンスペアの開始点 $(t_s, i_s) = s(t,i)$ は開始点行列によって保持され，次のように求められる．

$$s(t,i) = \begin{cases} s(t, i-1) & (v(t,i-1) > 0 \ \wedge \ v(t,i) \\ & \quad = \varepsilon b_v - \gamma(\phi \to y_i) + v(t, i-1)) \\ s(t-1, i) & (v(t-1,i) > 0 \ \wedge \ v(t,i) \\ & \quad = \varepsilon b_h - \gamma(x_t \to \phi) + v(t-1, i)) \\ s(t-1, i-1) & (v(t-1,i-1) > 0 \ \wedge \ v(t,i) \\ & \quad = \varepsilon b_d - \gamma(x_t \to y_i) + v(t-1, i-1)) \\ (t,i) & (otherwise) \end{cases} \tag{9}$$

⋆1 属性値ごとに記号に変換するのではなく，1 要素単位で記号に変換することも可能であるが，定義パターンをより詳細に設定できるといった利点を考え，本論文では属性値ごとの変換とする．

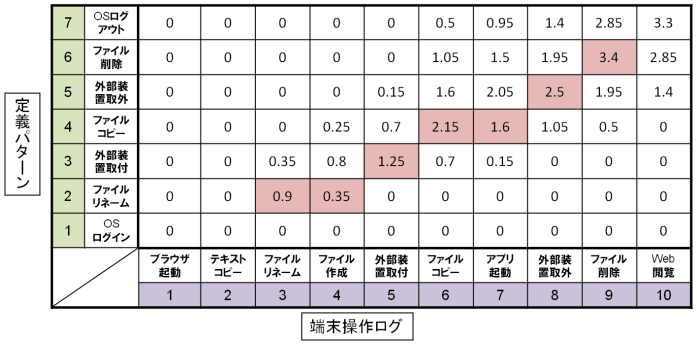

図 5　スコア行列の処理例
Fig. 5　Example of score matrix.

行動パターン検出アルゴリズムは，スコア行列と開始点行列の 2 つの行列を用いて，効率的に類似する部分シーケンスペアを検出する．計算時間は各行列に対し $O(mn)$ 個の値を更新するため，全体でも $O(mn)$ で処理可能である．メモリ使用量は各行列において 2 列（現在の列と直前の列）が必要となり，$O(m)$ となる．アルゴリズムの詳細は付録 A.2 に示す．

図 5 は，外部記憶媒体による情報漏えいを想定した場合のスコア行列を示している．処理の理解を容易にするために，定義パターン，端末操作ログともにテキスト形式で表示しているが，実際の処理においてはこれらがすべて記号変換されているものとする．図 5 のスコア行列において，濃く色付けしたセルが定義パターンと端末操作ログの適合を表す．すなわち，定義パターンの要素 2 から 6 までと端末操作ログのイベント 3 から 9 までが高いスコアで類似している．端末操作ログには，イベント 4 と 7 に，定義パターンに含まれていないイベント（すなわち，ノイズイベント）が存在する．そのため，パターン間の適合を示すセル $(4, 2)$ と $(7, 4)$ において，一時的にスコアが低下している．しかしながら，全体的なパターンの類似が，このノイズによる影響を吸収し，結果として高いスコアを獲得する．

5. 評 価 実 験

危険行動検出方式の評価を行うため，実データを用いた実験を行った．実験は，Intel Xeon W3565 3.2 GHz の CPU と 12 GB のメモリを持つ Linux マシンで行った．

5.1　実験データと定義パターン

本実験では，自治体職員の端末から収集した操作履歴の実データを用いて評価を行った．データ収集には，MaLion [⋆1] という市販のログ収集ツールを使用した．これは，ユーザ端末にインストールするエージェント型のツールであり，アプリケーションの利用やファイル操作，印刷，メールの送受信などを監視し，監視対象ごとに操作ログとして出力する．操作ログはサーバ側で集約されるため，各操作ログには複数ユーザの操作履歴が混在している．本論文では，ユーザごとの危険行動を検出することが目的であるため，ログをユーザごとに分割し，操作時間でソートしたものを実験データとして使用した．このとき，複数の操作ログ間で共通して含まれるものについては同じ属性値として集約した（たとえば，表 1 のアクティブウィンドウとファイル操作の EXE パス）．実験データは 2 種類あり，2009 年 7 月 8 日から 9 日に職員 43 名を対象に取得されたものをデータ 1，2009 年 9 月 14 日に職員 33 名を対象に取得されたものをデータ 2 とする．**図 6** に，実験データの具体例を示す．

一方，定義パターンについては，監査人により定義された 25 個の危険行動パターンのうち，特に自治体職員に関連が高い，次の 3 つのパターンを検証した（**表 2**）．

- 定義パターン 1：業務システムから該当者の一覧を検索し印刷する．
- 定義パターン 2：業務システムのデータを抽出し，ローカルドライブに保存する．
- 定義パターン 3：Web メールを用いてファイルを転送する．

定義パターン 1 は，業務システムにアクセスし，該当者を探し出して，それらを印刷するという行動である．これは，単純な印刷という行動ではなく，特定の対象者を探し出し，それらを確認したうえで印刷をするという一連の行動が，情報を持ち出す悪意を持っている可能性があると判断される．定義パターン 2 は，業務システム上に保存されているデータをユーザ端末に保存するという行動である．業務システム上のデータは，そのシステム上において参照されるものであるため，これらのデータを別のデバイスに保存していることが情報漏えいの可能性を含んでいる．定義パターン 3 は，ユーザ端末に保存されているファイルを，Web メールを用いて送信するという行動である．この Web メールは，業務アプリケーションとして提供されているメールの利用を想定しており，業務システム上の情報が持ち出されている可能性がある．

5.2　前　処　理

危険行動パターンの検出に先立ち，定義パターンと端末操作ログの記号変換処理を行う．まずはじめに，定義パターンの各要素をログ上の表記に変換した．定義パターン 1 を用いて，表記変換の具体例を示す．表 2 に示すとおり，定義パターン 1 は 5 つの要素から構成

⋆1 http://www.intercom.co.jp/malion/

	端末エージェント名	マシン名	ログオンユーザID	時刻	ログ種別	ウィンドウタイトル	ファイル操作	ファイル名	印刷種別	印刷対象	EXEパス	アプリケーション	ドライブ	URL
1														
2	b513e48c	7d62ff61	297a1762	2009/9/14 8:28	アクティブウィンドウ監視	予約検索					C:¥Program Files¥Trend Micro¥OfficeScan Client¥PccNTMon.exe	PccNTMon.exe		
3	b513e48c	7d62ff61	297a1762	2009/9/14 8:29	アクティブウィンドウ監視	Program Manager					C:¥WINDOWS¥explorer.exe	explorer.exe		
4	b513e48c	7d62ff61	297a1762	2009/9/14 8:30	ファイルアクセス監視		読込み	[redacted]			c:¥windows¥system32¥wscript.exe	wscript.exe	固定 ハードディスク	
5	b513e48c	7d62ff61	297a1762	2009/9/14 8:30	ファイルアクセス監視		読込み	[redacted]			c:¥windows¥system32¥wscript.exe	wscript.exe	固定 ハードディスク	
6	b513e48c	7d62ff61	297a1762	2009/9/14 8:30	ファイルアクセス監視		書込み	http:¥¥avsv02city.****.****.jp			c:¥program files¥trend micro¥officescan client¥pccntmon.exe	pccntmon.exe	N/A	
7	b513e48c	7d62ff61	297a1762	2009/9/14 8:31	アクティブウィンドウ監視	Q-Printianavi								
8	b513e48c	7d62ff61	297a1762	2009/9/14 8:31	アクティブウィンドウ監視	IPWATP								
9	b513e48c	7d62ff61	297a1762	2009/9/14 8:34	アクティブウィンドウ監視	ハードコピー設定					C:¥ハードコピーツール¥HCOPY.EXE	HCOPY.EXE		
10	b513e48c	7d62ff61	297a1762	2009/9/14 8:34	ファイルアクセス監視		読込み	[redacted]			c:¥windows¥system32¥wscript.exe	wscript.exe	固定 ハードディスク	
11	b513e48c	7d62ff61	297a1762	2009/9/14 8:34	ファイルアクセス監視		読込み	[redacted]			c:¥windows¥system32¥wscript.exe	wscript.exe	固定 ハードディスク	
12	b513e48c	7d62ff61	297a1762	2009/9/14 8:34	ファイルアクセス監視		読込み	[redacted]			c:¥windows¥system32¥wscript.exe	wscript.exe	固定 ハードディスク	
13	b513e48c	7d62ff61	297a1762	2009/9/14 8:35	ファイルアクセス監視		書込み	http:¥¥avsv02city.****.****.jp			c:¥program files¥trend micro¥officescan client¥pccntmon.exe	pccntmon.exe	N/A	
14	b513e48c	7d62ff61	297a1762	2009/9/14 8:35	アクティブウィンドウ監視	Program Manager					C:¥WINDOWS¥explorer.exe	explorer.exe		
15	b513e48c	7d62ff61	297a1762	2009/9/14 8:36	アクティブウィンドウ監視	Microsoft Internet Explorer					C:¥Program Files¥Internet Explorer¥iexplore.exe	iexplore.exe		
16	b513e48c	7d62ff61	297a1762	2009/9/14 8:36	Webアクセス監視									http://pac.city.****.****.jp/proxypac
17	b513e48c	7d62ff61	297a1762	2009/9/14 8:36	Webアクセス監視									http://portal.city.****.****.jp/user.ori/startup.ins
18	b513e48c	7d62ff61	297a1762	2009/9/14 8:36	アクティブウィンドウ監視	[redacted]					C:¥Program Files¥Internet Explorer¥iexplore.exe	iexplore.exe		
19	b513e48c	7d62ff61	297a1762	2009/9/14 8:36	アクティブウィンドウ監視	STARTUP – Microsoft Internet Explorer					C:¥Program Files¥Internet Explorer¥iexplore.exe	iexplore.exe		
20	b513e48c	7d62ff61	297a1762	2009/9/14 8:36	アクティブウィンドウ監視	Microsoft Internet Explorer					C:¥Program Files¥Internet Explorer¥iexplore.exe	iexplore.exe		
21	b513e48c	7d62ff61	297a1762	2009/9/14 8:36	Webアクセス監視									[redacted]

図 6　実験データの例
Fig. 6　Example of dataset.

表 2　定義パターンの詳細
Table 2　Details of defined patterns.

定義パターン 1	1. 業務システムに接続する 2. 業務アプリケーションを起動する 3. 該当者の絞込みを行う 4. 選択した該当者の一覧を画面に表示する 5. ツールを用いて印刷する
定義パターン 2	1. 業務システムに接続する 2. 業務アプリケーションを起動する 3. データ抽出のためのバッチジョブを選択する 4. バッチジョブを起動する 5. 抽出したデータを画面に表示する 6. ローカルドライブへの保存を選択する 7. ローカルドライブに書き込む 8. 保存先のフォルダを表示する
定義パターン 3	1. ポータルページを開く 2. メールサービスを起動する 3. 添付ファイルを選択する 4. メールを送信する

される．各イベントのログ表記は以下のとおりである．

＜1. 業務システムに接続する＞
ログ種別：「アクティブウィンドウ監視」，ウィンドウタイトル：「COKAS − X に接続しています」，アプリケーション：「wfcrun32.exe」

＜2. 業務アプリケーションを起動する＞
ログ種別：「アクティブウィンドウ監視」，ウィンドウタイトル：「業務ネット業務担当者コーナー」，アプリケーション：「ieplore.exe」または「firefox.exe」

＜3. 該当者の絞り込みを行う＞
ログ種別：「アクティブウィンドウ監視」，ウィンドウタイトル「検索キー入力 - ¥¥リモート」，アプリケーション：「wfica32.exe」

＜4. 選択した該当者の一覧を画面に表示する＞
ログ種別：「アクティブウィンドウ監視」，ウィンドウタイトル「該当者一覧 - ¥¥リモート」，アプリケーション：「wfica32.exe」

＜5. ツールを用いて印刷する＞
ログ種別：「印刷監視」，印刷対象：「HardCopy」

ここで，要素 1，2 はアプリケーション起動時に発生するイベントである．これは通常，端末起動時，すなわち，業務開始時にのみ起動されていることが多いことが確認された．そのため，定義パターン 1 においてつねに発生するものではないため，3 から 5 の要素を定義パターン 1 の表記変換の対象とした．定義パターン 1 と同様の処理を経て，定義パターン 2 と 3 もログ上の表記に対応付ける．詳細な対応付けについては省略するが，定義パターン 2，3 ともに 6 つの要素として表記変換される．

表 3　記号変換ルールの例
Table 3　Examples of symbol conversion rules.

属性値	対象項目	記号
ログ種別	アクティブウィンドウ監視	a
	ファイルアクセス監視	b
	Web アクセス監視	c
	印刷監視	d
ウィンドウタイトル	ポータル [メインメニュー]	a
	メールサービス	b
	入力ファイルの指定	c
	外部ファイルから入力	c
	ファイルを開く	c
	抽出 - ¥¥リモート	d
	ファイル転送中	e
	検索キー入力 - ¥¥リモート	f
	該当者一覧 - ¥¥リモート	g
ファイル操作	書込み	a
ファイル名	c:¥notexist.htm	a
印刷対象	HardCopy	a
アプリケーション	wfcrun32.exe	a
	wfica32.exe	b
	explorer.exe	c
	ieplore.exe	d
	firefox.exe	d
ドライブ	固定 ハードディスク	a

次に，表記変換によって得られた属性値とそれらの項目に対し，記号変換する処理を行う．3 つの定義パターンに必要となるのは，表 1 の属性値の中で，ログ種別，ウィンドウタイトル，ファイル操作，ファイル名，印刷対象，アプリケーション，ドライブ，URL の 8 つであった．属性値の各項目に対し，一意の記号を設定する．変換ルールの一部を**表 3** に示す．なお，変換ルール表で指定されていない項目については，すべて z に設定した．

たとえば，定義パターン 1 の場合，3 から 5 の各要素に含まれる属性値に対して，変換ルールに基づいた記号の割当てを行う．要素 3 は，ログ種別，ウィンドウタイトル，アプリケーションの 3 つの属性値からなり，変換ルール適用後，それぞれ，a，f，b の記号に変換される．要素 3 に含まれていない残りの 5 つの属性値についてはすべて z が割り当てられるため，要素 3 は $(a, f, z, z, z, b, z, z)^{T}$ の 8 次元の記号となる．同様に，要素 4 は $(a, g, z, z, z, b, z, z)^{T}$，要素 5 は $(d, z, z, z, a, z, z, z)^{T}$ となる．

定義パターンと同様の操作により，端末操作ログも記号変換処理を行う．ログのクレンジングにおいて，ウィルスチェックの起動などシステム起因のログを削除し，表 3 に基づいて，記号を割り当てる．このとき，対象の属性値以外の項目は削除され，8 次元の記号シーケンスとなる．たとえば，図 6 の実験データの場合，端末エージェント名，マシン名，ログオンユーザ ID，時刻，印刷種別，EXE パスの 6 つの属性値を削除し，残った 8 つの属性値に変換ルールを設定する．最初のイベントは，ログ種別：「アクティブウィンドウ監視」，ウィンドウタイトル：「予約検索」，アプリケーション：「PccNTMon.exe」であるため，$(a, z, z, z, z, z, z, z)^{T}$ のように変換される．

表 4　精度評価結果
Table 4　Results of accuracy.

データ No.	定義パターン No.	正解数	提案手法検出数（正解数）	再現率	適合率
1	3	45	85 (44)	97.8%	51.8%
2	1	1	21 (1)	100%	5%
	2	1	12 (1)	100%	8%
	3	10	11 (9)	90%	82%

上記の処理により，定義パターン，端末操作ログともに記号列データに変換され，行動パターン検出アルゴリズムを用いた行動パターンの検出を行う．式 (1) において，8 次元の記号列における置換コストは，次のように計算する．

$$\gamma(x_t \to y_i) = \begin{cases} 0 & \forall x_{t,k} = y_{i,k} \\ 1 & otherwise \end{cases}$$

ここで，$x_{t,k}$ は x_t の k 次元の記号を指し，$y_{i,k}$ についても同様である．

検出されたパターンの精度については，監査人が目視により検出したパターンを正解とし，それらと比較することで評価した．なお，収集したデータの性質上，データ 1 では定義パターン 3 を，データ 2 ではすべての定義パターンについての検証を行った．

5.3 実験結果

まずはじめに，提案方式の精度評価についての結果を**表 4** に示す．表 4 において，再現率は正解数に対するアルゴリズム検出正解数の割合を，適合率はアルゴリズム検出総数に対するアルゴリズム検出正解数の割合を示す．適合率にばらつきがあるものの，再現率は 90%以上の高い値で検出が可能であることが分かった．セキュリティ監査においては，危険行動パターンを漏れなく検出することが求められるため，再現率の高さが重要となる．本実験結果から，本方式は監査人の監査をサポートするうえで重要なツールとなりうることが確

	A	B	C	D	E	F	G	H
1	ログ種別	ウィンドウタイトル	ファイル操作	ファイル名	印刷対象	アプリケーション	ドライブ	URL
1906	アクティブウィンドウ監視	システム起動 - ¥¥リモート				wfica32.exe		
1907	アクティブウィンドウ監視	業務ネット 業務担当者コーナー - Mozilla Firefox				firefox.exe		
1908	アクティブウィンドウ監視	業務ネット 業務担当者コーナー - Mozilla Firefox				firefox.exe		
1909	アクティブウィンドウ監視	連携認証確認ダイアログ - ¥¥リモート				wfica32.exe		
1910	アクティブウィンドウ監視	業務ネット 業務担当者コーナー - Mozilla Firefox				firefox.exe		
1911	アクティブウィンドウ監視	メニュー - ¥¥リモート				wfica32.exe		
1912	Webアクセス監視							
1913	アクティブウィンドウ監視	検索キー入力 - ¥¥リモート				wfica32.exe		
1914	アクティブウィンドウ監視	業務ネット 業務担当者コーナー - Mozilla Firefox				firefox.exe		
1915	アクティブウィンドウ監視	該当者一覧 - ¥¥リモート				wfica32.exe		
1916	印刷監視				HardCopy			
1917	アクティブウィンドウ監視					explorer.exe		
1918	アクティブウィンドウ監視	ハードコピー設定				HCOPY.EXE		
1919	アクティブウィンドウ監視					explorer.exe		
1920	アクティブウィンドウ監視	該当者一覧 - ¥¥リモート				wfica32.exe		
1921	アクティブウィンドウ監視	TF_FloatingLangBar_WndTitle - ¥¥リモート				wfica32.exe		
1922	アクティブウィンドウ監視					explorer.exe		
1923	アクティブウィンドウ監視	無題 - ペイント				mspaint.exe		
1924	アクティブウィンドウ監視	ペイント				mspaint.exe		
1925	アクティブウィンドウ監視	該当者一覧 - ¥¥リモート				wfica32.exe		

図 7　検出されたパターンの例
Fig. 7　Example of detected patterns.

認された．

図 7 は，定義パターン 1 の検出結果である．実線で囲まれた部分が，適合した正解パターンを示している．このパターンには，定義パターンに含まれていないノイズイベントが含まれているが（1,914 行目），検出に成功している様子が確認される．一方，正解パターンではなく検出されたパターンは，該当者の一覧を絞り込んで表示したが，印刷には至らなかったという行動であった．本方式では，定義パターンの一部に適合するパターンも検出可能であり，その他の定義パターンにおける検出結果においても同様の傾向が見られた．これらのパターンの分析は，危険行動の取りこぼしを防ぐだけでなく，新たな危険行動パターンの検出にもつながる可能性があり，本方式のセキュリティ監査への適用を高めるものであると考えている．

一方，検出できなかったパターンは 2 つあり，データ 1，2 ともに定義パターン 3 に該当するパターンであった．1 つ目のパターンは，メール送信の前に，ファイルを読み込むという行動が複数回ノイズとして挿入されており，かつ，メール送信の同一ログが複数回取得されていたことが原因であると考えられる．本方式は，ノイズイベントに対して比較的ロバストな手法ではあるが，想定以上のノイズイベントの挿入により大幅にスコアが減少してしまう．さらに，編集距離は，一方のシーケンスをもう一方のシーケンスに一致させるための操作コストとして定義されるため，同一のイベントが連続している場合，挿入または削除コストが加算され，スコアは大幅に減少する．これらの影響については，スコア関数における挿入，削除，置換の各編集操作のコストの重みを変えることで改善される可能性があると考えられる．また，もう 1 つのパターンを検出できなかった原因としては，2 つの行動が同時に

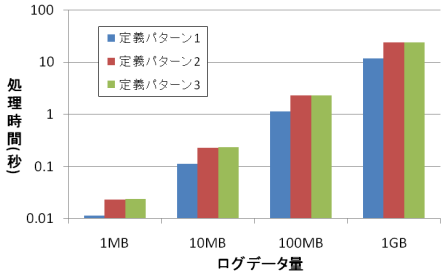

図 8　性能評価結果
Fig. 8　Results of performance evaluation.

行われており，一方の行動がもう一方のノイズイベントとなってしまった可能性が考えられる．検出できなかったパターンは，検出できた正解パターンのすぐ直後に行われたメール送信行動であり，2 つのメール送信行動のログが並存して取得されていた．一方の行動の最中に別の行動が行われた場合，定義パターンに含まれるイベントであってもそのイベントの取得位置が異なり，イベントの順序性が失われてしまうためノイズイベントとなってしまう．並存する行動については，同じフォルダや同じ Web ブラウザ単位で操作を 1 つにまとめることで区別することができる可能性もあるため，ログの前処理についてさらに検討するつもりである．

次に，提案方式の性能評価実験を行った．実験は，ログ量に対する各定義パターンの処理時間を測定している．結果を**図 8** に示す．定義パターンの長さや検出されたパターン数などの影響で，定義パターンごとに若干の違いは見られるが，1 GB のログに対してもわずか数分で処理可能であるとの結果が得られた．本実験で使用した操作ログは合計 2.3 MB であり，約 1 カ月分を想定したとしても 100 MB に満たないため，十分に処理可能なスケールであるといえる．また，本方式はリアルタイムなパターン検出も可能であり，この場合，ログの総量にかかわらず一定の時間で処理を行う．そのため，パターンの検出と監査人による危険行動判定を分けて行うことができ，監査の効率化につながると考えられる．

5.4　実用化に向けた課題

記号変換フェーズについて，定義パターンの各要素が厳密にログ上の表記に対応付けられないことが確認された．たとえば，「アプリケーションを終了する」という行動は，ログとして記録されない．そのため，危険行動として定義した行動が，ログから取得されないという可能性が起こりうる．これは，ログ収集ツールの変更により改善される場合もあるが，

多くのツールが，ユーザの詳細な行動把握を目的としてログを取得しているわけではないことに起因する．行動パターン分析においては行動ログを正しく取得することが重要であり，監査のためのログ収集ツールの検討が必要となると考えている．また，すでに定義しているパターンの対応付けについても，まだ不完全である可能性も考えられる．これは，定義パターンに含まれているにもかかわらず，ノイズイベントとして処理してしまっていることになる．検出精度を上げるためにも，より詳細な分析が必要になると考えている．

行動パターン検出フェーズについては，ノイズイベントや同一イベントにより強固なアルゴリズムに改善する必要があることが分かった．スコア関数における編集操作コストの重みが結果にどのように影響するのかを調べ，危険行動パターンの検出に有効な重みを決定していくつもりである．

6. 関 連 研 究

本論文の関連研究は大きく 2 種類に分類することができる．第 1 のグループはシーケンスマッチングに関するものであり，第 2 のグループは異常検出に関するものである．

シーケンスマッチング　編集距離[30),36)] は，2 つの記号シーケンス間全体の類似性を調べるために利用されてきた．記号シーケンス間の部分的な類似性を調べるアルゴリズムとしては，バイオインフォマティクスの分野における，遺伝子やたんぱく質配列のための探索アルゴリズムである Smith-Waterman アルゴリズムが広く利用されている[31)]．発見的アルゴリズムによる近似解を求めることで高速化を目指した FASTA[27)] や BLAST[6),7)]，並列化による高速化をめざした手法[28)] など，Smith-Waterman アルゴリズムの改良も多くみられる．遺伝子やたんぱく質など，バイオシーケンスが 1 次元の記号シーケンスで表現されるのに対し，本論文で扱う端末操作ログは多次元の記号シーケンスとして表現される点で異なっており，その処理はより複雑なものとなる．さらに，提案するアルゴリズムはデータストリーム処理の中で，適合する部分シーケンスを探索漏れを発生することなく検出する．すなわち過去のデータに遡ることなしに逐次的に処理を行い，距離値だけでなく部分シーケンスの位置を特定することができる．

データベースやデータマイニング分野においても，古くからシーケンスマッチングの問題が扱われている[11),16),37)]．これらの手法は，スライディングウィンドウを用いてあらかじめ部分シーケンスに分割する．一方，我々の手法はスコアの値によって自動的に部分シーケンスが決定され，あらかじめシーケンスを分割する必要がない．また，スライディングウィンドウを用いず，効率的に部分シーケンスを特定する手法も提案されているが[8),29),40)]，データベースやデータマイニング分野で提案されている手法は，主に数値で表されるシーケンスを対象としており，記号シーケンスへの適用は困難である．

異常検出　異常検出は，データから予期しない振舞いのパターンを見つけることであり，なりすましや詐欺，侵入検知などに応用されている[12)]．距離ベース[25)] やクラスタリングベース[9),35)]，統計ベース[14),17),38)] など様々な手法が提案されている．また，ストリーム処理に対応した手法[18),22)] も検討されている．しかしながら，これらの手法は単一のイベントに対する異常検出を目的とするものであり，本論文で扱う問題に適用することは困難である．

文献 19) では，潜在変数空間を利用した異常行動検出手法が提案されている．彼らは異常検出のための 2 つの隠れ変数を示し，それらの変化を追跡することで観測データから異常行動を検出する．また，文献 10) は，株式市場などにおける異常な振舞いの取引を，株の売り買いと通常の取引の 3 つシーケンスを対として扱うことにより検出するものである．これらの研究は，HMM ベースの手法を用いて異常行動を学習により検出しようというものであり，我々のアプローチとは異なる．

文献 23) では，シーケンスマッチングを用いた異常検出の問題が扱われている．UNIX コマンドにおけるなりすましを検出するために，ユーザごとにパターンを学習し，学習したパターンと入力データとのマッチングを行う．しかしながら，独自に考案された距離関数を用いている点や，シーケンスの長さが学習したパターンと同じものを対象としている点が大きく異なる．

文献 13) は，監査ログからのなりすましを検出するために Smith-Waterman アルゴリズムを改良し，その評価を行っている．シーケンスマッチングに基づく部分的なパターンの検出を目的としているところは本論文と共通するが，端末操作ログという多次元の記号列として表される監査データを対象としている点や，シーケンスの適合度合いの導入し，セキュリティ監査のためにより特化した手法を検討している点が異なる．

7. ま と め

本論文では，情報漏えいにつながる危険行動の高速検出が可能な方式を提案した．提案方式は，記号シーケンス間の類似度を評価する編集距離に基づく手法により，定義パターンと端末操作ログの部分的な適合を可能にするものである．自治体職員の端末操作ログを対象に提案方式の評価を行い，精度や性能の面で効率的に危険行動パターンを検出することが確認され，監査人の支援ツールとなりうる可能性があることが示された．一方，課題としては，提案方式自体や端末操作ログの抽出方法に改善の余地があることが分かった．今後は，実用

化に向けて，これらの課題の解決を図る予定である．

謝辞　本研究を進めるにあたり，NTT コミュニケーションズ株式会社の山口伸弥様に多くのご助言とサポートをいただきました．ここに感謝申し上げます．

参 考 文 献

1) ALog ConVerter, available from ⟨http://www.amiya.co.jp/solutions/alog_converter/⟩.
2) CWAT, available from ⟨http://www.iwi.co.jp/product/cwat.htm⟩.
3) MylogStar, available from ⟨http://www.mylogstar.net/⟩.
4) QOH, available from ⟨http://www.quality.co.jp/products/QOH/index.html⟩.
5) RSA enVision, available from ⟨http://japan.rsa.com/node.aspx?id=3170⟩.
6) Altschul, S., Gish, W., Miller, W., Myers, E. and Lipman, D.: Basic local alignment search tool, *Journal of Molecular Biology*, Vol.215, pp.403–410 (1990).
7) Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J.: Gapped Blast and PsiBlast: A new generation of protein database search programs, *Nucleic Acids Research*, Vol.25, No.17, pp.3389–3402 (1997).
8) Assent, I., Wichterich, M., Krieger, R., Kremer, H. and Seidl, T.: Anticipatory DTW for Efficient Similarity Search in Time Series Databases, *PVLDB*, Vol.2, No.1, pp.826–837 (2009).
9) Böhm, C., Faloutsos, C. and Plant, C.: Outlier-robust clustering using independent components, *SIGMOD*, pp.185–198 (2008).
10) Cao, L., Ou, Y., Yu, P.S. and Wei, G.: Detecting abnormal coupled sequences and sequence changes in group-based manipulative trading behaviors, *KDD*, pp.85–94 (2010).
11) Chan, K.-P. and Chee Fu, A.W.: Efficient time series matching by wavelets, *ICDE*, pp.126–133 (1999).
12) Chandola, V., Banerjee, A. and Kumar, V.: Anomaly detection: A survey, *ACM Comput. Surv.*, Vol.41, pp.15:1–15:58 (2009).
13) Coull, S.E. and Szymansk, B.K.: Sequence alignment for masquerade detection, *Computational Statistics and Data Analysis*, Vol.52, pp.4116–4131 (2008).
14) Das, S., Matthews, B.L., Srivastava, A.N. and Oza, N.C.: Multiple kernel learning for heterogeneous anomaly detection: Algorithm and aviation safety case study, *KDD*, pp.47–56 (2010).
15) Durbin, R., Eddy, S.R., Krogh, A. and Mitchison, G.: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press (1997).
16) Faloutsos, C., Ranganathan, M. and Manolopoulos, Y.: Fast subsequence matching in time-series databases, *SIGMOD*, pp.419–429 (1994).
17) Fujimaki, R., Nakata, T., Tsukahara, H. and Sato, A.: Mining Abnormal Patterns from Heterogeneous Time-Series with Irrelevant Features for Fault Event Detection, *SDM*, pp.472–482 (2008).
18) Gu, X. and Wang, H.: Online Anomaly Prediction for Robust Cluster Systems, *ICDE*, pp.1000–1011 (2009).
19) Hirose, S. and Yamanishi, K.: Latent Variable Mining with Its Applications to Anomalous Behavior Detection, *SDM*, pp.231–242 (2008).
20) Huang, X., Acero, A. and Hon, H.-W.: *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall (2001).
21) Institute, C.S.: 2010 CSI Computer Crime and Security Survey (2010).
22) Kontaki, M., Gounaris, A., Papadopoulos, A.N., Tsichlas, K. and Manolopoulos, Y.: Continuous monitoring of distance-based outliers over data streams, *ICDE*, pp.135–146 (2011).
23) Lane, T. and Brodley, C.E.: Sequence matching and learning in anomaly detection for computer security, *AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management* (1997).
24) Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics Doklady*, Vol.10, No.8, pp.707–710 (1966).
25) Müller, E., Schiffer, M. and Seidl, T.: Statistical selection of relevant subspace projections for outlier ranking, *ICDE*, pp.434–445 (2011).
26) Paxson, V.: Bro: A System for Detecting Network Intruders in Real-Time, *Computer Networks*, pp.2435–2463 (1999).
27) Pearson, W.R. and Lipman, D.J.: Improved Tools for Biological Sequence Comparison, *Proc. National Academy of Sciences of the United States of America*, Vol.85, No.8, pp.2444–2448 (1988).
28) Rognes, T. and Seeberg, E.: Six-fold speed-up of Smith-Waterman sequence database searches using parallel processing on common microprocessors, *Bioinformatics*, Vol.16, pp.699–706 (2000).
29) Sakurai, Y., Faloutsos, C. and Yamamuro, M.: Stream Monitoring under the Time Warping Distance, *ICDE*, pp.1046–1055 (2007).
30) Sankoff, D. and Kruskal, J.B.: *Time warps, string edits, and macromolecules: The Theory and Practice of Sequence Comparison*, Cambridge University Press (1999).
31) Smith, T.F. and Waterman, M.S.: Identification of common molecular subsequences, *Journal of molecular biology*, Vol.147, pp.195–197 (1981).
32) Stefan, B. and Rita, S.: Finding the Leak: A Privacy Audit System for Sensitive XML Databases, *International Conference on Data Engineering Workshops*, p.100 (2006).

33) Tsay, Y. and Tsai, W.: Attributed String Matching by Split-and-Merge for On-Line Chinese Character Recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.15, pp.180–185 (1993).
34) Vigna, G. and Kemmerer, R.A.: NetSTAT: A Network-based Intrusion Detection System, *Journal of Computer Security*, Vol.7, pp.37–71 (1999).
35) Wang, Y., Parthasarathy, S. and Tatikonda, S.: Locality Sensitive Outlier Detection: A ranking driven approach, *ICDE*, pp.410–421 (2011).
36) Wei, J.: Markov Edit Distance, *IEEE Trans. Pattern Analysis and Machine Intelligence* (*TPAMI*), Vol.26, No.3, pp.311–321 (2003).
37) Wong, T.S.F. and Wong, M.H.: Efficient Subsequence Matching for Sequences Databases under Time Warping, *IDEAS*, pp.139–148 (2003).
38) Yamanishi, K. and Maruyama, Y.: Dynamic syslog mining for network failure monitoring, *KDD*, pp.499–508 (2005).
39) You, P. and Yan-Zhang, W.: Research about Security Audit Platform in E-Government System, *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp.235–239 (2008).
40) Zhou, M. and Wong, M.H.: Efficient Online Subsequence Searching in Data Streams under Dynamic Time Warping Distance, *ICDE*, pp.686–695 (2008).
41) サイボウズメディアアンドテクノロジー株式会社：日本情報漏えい年鑑 2008 (2008).
42) 経済産業省：情報セキュリティ監査基準 (2008).
43) 高田哲司，小池英樹：ログ情報視覚化システムを用いた集団監視による不正侵入対策手法の提案，情報処理学会論文誌，Vol.41, No.8, pp.2216–2227 (2000).
44) 小林　透，豊田真智子，市川裕介：端末操作ログを対象にした情報漏洩につながる危険行動高速抽出方式，電子情報通信学会技術研究報告 LOIS，ライフインテリジェンスとオフィス情報システム，Vol.110, No.450, pp.19–24 (2011).
45) 中村友洋，新谷隆彦，恵木正史，櫻井隆雄：操作ログを利用した Web 操作支援システム，電子情報通信学会技術研究報告 LOIS，ライフインテリジェンスとオフィス情報システム，Vol.109, No.39, pp.55–62 (2009).
46) 鳥羽美奈子，森　靖英，恵木正史，櫻井隆雄：PC 操作ログと映像ログを用いた業務行動モニタリングシステムの初期検討，情報処理学会研究報告 CVIM，コンピュータビジョンとイメージメディア，Vol.2010, No.1, pp.1–8 (2010).
47) 豊田真智子，櫻井保志，石川佳治：部分シーケンスマッチングのためのストリームアルゴリズム，電子情報通信学会論文誌 D，Vol.94, No.7, pp.1058–1070 (2011).

付　　録

A.1　ナイーブな手法

端末操作ログ X と定義パターン Y 間の部分シーケンスペアを検出する最も素直な方法は，考えられるすべての部分シーケンスの組合せについて，それらの編集距離を計算することである．ここで，考えられるすべての部分シーケンスとは，$X[t_s : t_e]$ については $1 \le t_s \le n-1 \wedge t_s < t_e \le n$，$Y[i_s : i_e]$ については $1 \le i_s \le m-1 \wedge i_s < i_e \le m$ の範囲の部分シーケンスのことであり，すべての要素を開始点とする編集距離の行列を作成し，その距離を計算することを意味する．この手法をナイーブな手法と呼ぶ．

端末操作ログ $X = (x_1, \ldots, x_t, \ldots, x_n)$ と定義パターン $Y = (y_1, \ldots, y_i, \ldots, y_m)$ が与えられたとき，X の t 番目の要素と Y の i 番目の要素から始まる編集距離行列におけるセル (p, q) が示す距離を $d_{t,i}(p, q)$ とする．X と Y 間の部分シーケンスマッチングの編集距離は次のように計算される．

$$
\begin{aligned}
&D(X[t_s : t_e], Y[i_s : i_e]) = d_{t_s,i_s}(l_x, l_y) \\
&d_{t,i}(p,q) = min \begin{cases} d_{t,i}(p, q-1) + \gamma(\phi \rightarrow y_i) \\ d_{t,i}(p-1, q) + \gamma(x_t \rightarrow \phi) \\ d_{t,i}(p-1, q-1) + \gamma(x_t \rightarrow y_i) \end{cases} \qquad (10) \\
&d_{t,i}(0,0) = 0 \\
&d_{t,i}(p,0) = p \ (p = 1, \ldots, n-t+1) \\
&d_{t,i}(0,q) = q \ (q = 1, \ldots, m-i+1)
\end{aligned}
$$

ナイーブな手法は，シーケンスの各要素ごとに新たな行列を作成するため，$O(mn)$ 個の行列を扱うことになる．さらに，その 1 つ 1 つの行列において，すべての要素の距離値 $O(mn)$ 個を計算する．そのため合計すると，$O(m^2n^2)$ 個もの値を計算する必要がある．

A.2　行動パターン検出アルゴリズム

スコア行列，開始点行列において，スコア $v(t,i)$ と開始点 $s(t,i)$ を式 (7) と (9) を用いて計算する．アルゴリズムの詳細を**図 9** に示す．問題 1 の条件 (1) を満たす部分シーケンスペア（適合する部分シーケンスペア）を検出すると，そのスコア $C'_v := v(t,i)$，開始点 $C'_s := s(t,i)$，終了点 $C'_e := (t,i)$ を候補集合の配列 $\mathcal{S}$ に格納する．重複する部分シーケンスペアが存在する場合，候補配列には重複する候補ペアのグループの中で $v(t,i) - \varepsilon l_{min}$ が最大値となるペアのみが格納される．部分シーケンスペアの重複は，配列 $\mathcal{S}$ に含まれるすべての開始点（この集合を S_{c_s} とする）と現在の開始点 C'_s が一致するかどうかによって判断される．これは，開始点行列において，同じ要素を共有した時点でその要素の開始点が引き継がれていくためである．

候補配列には，開始点が異なる複数の最適な部分シーケンスペア（この時点では候補部分

```
Algorithm
Input:  端末操作ログ X，定義パターン Y
Output: 最適な部分シーケンスペアとその編集距離
for t := 1 to n do
  //最適な部分シーケンスペアの検出
  for i := 1 to m do
    C'_v := v(t,i);    //式 (7) から計算されるスコア
    C'_s := s(t,i);    //式 (9) から計算される開始点
    C'_e := (t,i);     //終了点
    if C'_v ≥ εl_min then
      if C'_s ∉ S_{C_s} then //新たな候補として追加
        配列 S に C' を追加;
      else
        for 配列 S の各候補 C do
          if C'_s = C_s ∧ C'_v ≥ C_v then //最大スコアを更新
            C_v := C'_v;
            C_e := C'_e;
          endif
        endfor
      endif
    endif
  endfor
  //最適な部分シーケンスペアの報告
  for 配列 S の各候補 C do
    if ∀_i, s(t,i) ≠ C_s then
      d_min := εL(l_x,l_y) − C_v;
      d_min, C_s, C_e を報告;
      配列 S から C を削除;
    endif
  endfor
endfor
```

図 9　行動パターン検出アルゴリズム
Fig. 9　Pattern detection algorithm.

シーケンスペア）の情報 C が保持されている（すなわち，スコア C_v，開始点 C_s，終了点 C_e）．アルゴリズムは以下の条件を満たすとき，候補部分シーケンスペアを最適解として報告する．

$$\forall_i, s(t,i) \neq C_s$$

これは，候補部分シーケンスペアが今後出現する部分シーケンスペアによって置き換わることがないことを意味する．最終的な部分シーケンスペアの類似度は式 (8) により編集距離 d_{min} として報告される．

A.3　理論的な分析

補題 1　端末操作ログ X と定義パターン Y が与えられたとき，問題 1 の 2 つの条件は次の条件と等価である．

(1)　$V(X[t_s : t_e], Y[i_s : i_e]) \geq \varepsilon l_{min}$

(2)　重複する部分シーケンスペアのグループの中で，$V(X[t_s : i_e], Y[j_s : j_e]) - \varepsilon l_{min}$ が最大値をとる．

証明 1　編集距離行列とスコア行列の双方で，開始点 (t_s, i_s)，終了点 (t_e, i_e) である適合パスが (t_e, i_e-1) を通るとするならば，式 (10) より

$$\gamma(\phi \to y_i) = d_{t_s,i_s}(l_x, l_y) - d_{t_s,i_s}(l_x, l_y-1).$$

また，$b_v = L(l_x, l_y) - L(l_x, l_y-1)$ から，式 (7) より，

$$\begin{aligned}\gamma(\phi \to y_i) &= \varepsilon L(l_x, l_y) - v(t_e, i_e) \\ &\quad - \varepsilon L(l_x, l_y-1) + v(t_e, i_e-1)\end{aligned}$$

が成り立つ．同様に，(t_e-1, i_e) と (t_e-1, i_e-1) を通るときは各々

$$\begin{aligned}\gamma(x_t \to \phi) &= d_{t_s,i_s}(l_x, l_y) - d_{t_s,i_s}(l_x-1, l_y) \\ &= \varepsilon L(l_x, l_y) - v(t_e, i_e) \\ &\quad - \varepsilon L(l_x-1, l_y) + v(t_e-1, i_e) \\ \gamma(x_t \to y_i) &= d_{t_s,i_s}(l_x, l_y) - d_{t_s,i_s}(l_x-1, l_y-1) \\ &= \varepsilon L(l_x, l_y) - v(t_e, i_e) \\ &\quad - \varepsilon L(l_x-1, l_y-1) + v(t_e-1, i_e-1)\end{aligned}$$

が成り立つ．

$$\gamma(x_t \to y_i) = d_{t_s,i_s}(1,1) = \varepsilon - v(t_s, i_s)$$

であるため，編集距離行列とスコア行列は同じ適合パスを共有する．また，適合パス上の重みの合計は $L(l_x, l_y)$ であるため，式 (7) より

$$v(t_e, i_e) = \varepsilon L(l_x, l_y) - d_{t_s,i_s}(t_e, i_e)$$

が成り立ち，編集距離とスコアの関係式

$$d_{t_s,i_s}(l_x, l_y) = \varepsilon L(l_x, l_y) - v(t_e, i_e) \tag{11}$$

が導かれる．さらに，式 (6) と (11) より，

$$v(t_e, i_e) \geq \varepsilon l_{min}$$

が得られる．

編集距離行列では，問題 1 の条件 (2) より，(t_s, i_s) から (t_e, i_e) までの最適な適合パスは適合パスが交差する部分シーケンスペアのグループの中で最小距離を示すことが明らかである．また式 (7) と式 (11) より，スコア行列でも同じ適合パスが選択され，最大スコアを出力することが分かる．よって，問題 1 の 2 つの条件と等価となる．□

補題 2　行動パターン検出アルゴリズムは，条件を満たす最適な部分シーケンスペアを検出する．

証明 2　最適な部分シーケンスペア $X[t_s : t_e]$ と $Y[i_s : i_e]$ のワーピングパスの開始点を $C_s = (t_s, i_s)$ とする．ログにおける t 番目のイベントにおいて，以下の条件を満たすとき，今後出現する部分シーケンスペアは候補配列の中の部分シーケンスペアと重複することはない．

$\forall_i, s(t, i) \neq C_s$

行動パターン検出アルゴリズムは上記の条件を満たすときのみ $X[t_s : t_e]$ と $Y[i_s : i_e]$ を最適な部分シーケンスペアとして報告する．よって，最適な部分シーケンスペアを見落とすことはない．□

(平成 23 年 6 月 20 日受付)
(平成 23 年 10 月 8 日採録)

(担当編集委員　新谷 隆彦)

豊田真智子（学生会員）

2004 年お茶の水女子大学理学部情報科学科卒業．2006 年同大学大学院人間文化研究科修士課程修了．同年日本電信電話（株）入社．現在，名古屋大学大学院情報科学研究科博士後期課程在学中．データストリーム処理の研究開発に従事．本会 2005 年度大会奨励賞受賞．電子情報通信学会，日本データベース学会各会員．

櫻井　保志（正会員）

1991 年同志社大学工学部電気工学科卒業．同年日本電信電話（株）入社．1999 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．2004〜2005 年カーネギーメロン大学客員研究員．本会平成 16 年度および平成 19 年度論文賞，平成 18 年度長尾真記念特別賞，電子情報通信学会平成 19 年度論文賞，日本データベース学会上林奨励賞，ACM KDD Best Paper Awards（2008 年および 2010 年）等受賞．索引技術，データストリーム処理，センサーデータ処理技術の研究に従事．ACM，電子情報通信学会，日本データベース学会各会員．

小林　　透（正会員）

1985 年東北大学工学部精密機械工学科卒業．1987 年同大学大学院工学研究科修士課程修了．同年日本電信電話（株）入社．以来，ソフトウェア生産技術，ユビキタスコンピューティング，情報セキュリティ等の研究開発に従事．現在，NTT サイバーソリューション研究所，グループリーダ，主幹研究員．電子情報通信学会，IEEE 各会員，博士（工学）．

市川　裕介（正会員）

1994 年慶應義塾大学理工学部計測工学科卒業．1996 年同大学大学院理工学研究科修士課程修了．同年日本電信電話（株）入社．通信履歴活用サービスの研究開発に従事．現在，NTT サイバーソリューション研究所，主任研究員．本会平成 6 年度学術奨励賞，平成 17 年度山下記念研究賞受賞．