入力コマンド系列の評価による侵入者検知手法 -GUIシステムへのアプローチ-

宮崎 小玉* 白井 治彦** 黒岩 丈介*** 小高 知宏* 小倉 久和***

Intruder Detection Method By Evaluating The Input Command Sequence -New Approach To GUI System-

Kodama MIYAZAKI*, Haruhiko SHIRAI**, Josuke KUROIWA***, Tomohiro ODAKA* and Hisakazu OGURA***

(Received February 8, 2013)

In this research, we focused on the characteristics that some people how use the mouse or keyboard when they using interactive software. Especially when people use a mouse device, how do they move on different from other people, such as coordinates, trajectory, speed and movement clicking speed. We saved up mouse clicking's log data and used it to extract features of the individual. We propose a new intruder detection method that used mouse clicking's log data.

In this intruder detection method, we evaluate the input command sequence for authentication using the transition probability table. This method has two steps. First step, we divided the collected into two sets , as the data in the first half of the training data, and used it to created a transition probability table. Next step, make the second half data into validation data, and used the transition probability table to take the average of the total. We appropriate the average value's height to judge who is real user.

This method is easy to implement to an ordinary computer. Using this method in conjunction with traditional authentication methods such as password system. And it can help system security be stronger than before.

Key words : Mouse Clicking, Intruder Detection Method, Command Sequence, Transition Probability Table, Authentication, Security

1. はじめに

近年コンピュータやタブレット端末，スマートフォンなどの情報通信機器の普及により，クラウドサービスも普及しつつある．こういったサービスでは，同じ端末機器でなくても，アカウントがあればどこでも同じサービスを受けることができる．しかし，クラウドでのデータの集約や，サービス認証でのセキュリティ脆弱性により，サイバー犯罪の格好な攻撃標的となっている．実際最近では大規模な個人情報漏洩が大きな問題となっている．

このような問題を解決するために，我々は，ユーザがマウスやキーボードなどを利用する際に見られる個人の特徴に注目した．例えば，マウスを利用するときに，人によって軌跡や座標が違ったり，クリックするときの早さが違ったり，無意識に読みたい文字を反転するなどの特徴がある．こういったものを認証システムと併用すれば，従来のシステムよりも安全度が向上

*大学院工学研究科 原子力・エネルギー安全工学専攻
**工学部技術部
*** 大学院工学研究科 知能システム工学専攻
*Nuclear Power and Energy Safety Engineering Course, Graduate School of Engineering
**Dept. of Technical, Dept. Engineering
***Human and Artificial Intelligent Systems Course, Graduate School of Engineering

すると考えられる．

また，近年では入力コマンド系列における個人の特徴抽出と，それを利用する認証手法 [1] の研究が進められてきた．しかし，コンピュータオペレーティングシステムで主流をとっているのは，マイクロソフト社が出している Windows（GUI）シリーズである．そこで本稿は，この入力コマンド系列で提案してきた認証手法を応用して，GUI システムでの効果を検証すべく，GUI システムの性格を分析し，入力コマンド系列の評価による侵入者手法の GUI システムに対する効果 [2] を検証を行なった．

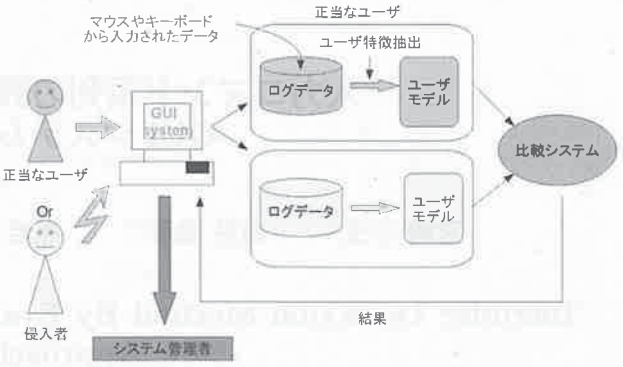

図 1: システム構成図

2. 侵入者検知手法

2.1 コマンド系列の特徴を用いた個人認証手法

ユーザがコンピュータと対話的に作業を行うとき，そのコマンド系列には何らかの個人的特徴が含まれている．これまでに，我々は入力コマンド系列の評価による侵入者検出手法をいくつか提案して来た．例えば，遷移確率表を用いる手法 [2] [4]，ファジィ測度的手法や HMM（隠れマルコフモデル）手法 [4] などがある．今回 GUI へ応用する手法は，遷移確率表による個人認証手法である．

遷移確率表による個人認証手法では，ユーザがあるコマンド入力に続いて，次にどのようなコマンドが入力されるかを絶えず監視する．入力されつつあるコマンド系列の遷移状況が，以前に入力されたコマンド系列を基に，遷移確率表を構成し，次に入力されつつあるコマンド系列の遷移状況と似ているかを調べることで本人であるかどうかを見分ける．この手法の利点は，既存のシステムへの実装が容易で，従来のパスワードによるユーザの認証方法との併用が可能であ．我々が提案してきたシステムの概要を以下の図 1. にあげる．ユーザがコンピュータ，主に Linux システムを利用する際に，コマンドラインを用いる．そのコマンドラインをログファイルとして溜め込み，遷移確率表を作成し，それに基づいて次に入力されるコマンドが正当なユーザであるかどうか，前のコマンドラインで作成した遷移確率表に照合して，比較をする．次に比較システムで判断させ，もし正当なユーザであればそのままシステムを利用することが許される．もし正当なユーザでなければ，システム管理者に知らせる，という仕組みである [2] [4]．

2.2 GUI システムへの応用

本稿では，既存する GUI システムに，上記の手法を適用させて，有効であるかどうか検証する．GUI システムでは主にマウスやキーボードによる直感操作が多い．マウス操作ならマウスポインタの軌跡や座標ログを検証し，キーボード操作ならキーボード入力する際，入力速度やミスの多いキーに着目し，ユーザの特徴を抽出できると考えられる．今回は主にマウスに注目して，検証を行なった．

3. GUI での認証手法

本稿では，Windows システムの利用ログを用いて，個人の特徴を抽出する試みをし，2 章で述べた遷移確率表による個人認証手法を用いた．この手法は 2 つのステップから構成されている．詳しくは，3.2 章で説明する．

3.1 検証データ抽出

先ほど述べたように，今回利用するログデータは，Windows システムでのマウスによるクリックイベントログを抽出したものである．マウスクリックイベントログとは，マウスポインタがシステム上で，どこをクリックしたのかをキャッチし，履歴，操作時間やアクションなどの情報をログとして溜め込む．

図 2. がその採取したデータログの一例を示している．履歴と操作時間は付加情報として，アクションがイベントログとして加工する．今回はアクションのログのみを使用し，さらにこのイベントログの中の実行プログラムだけを文字列として抽出する．

また，ログデータが文字列であるため，Java プログラミング言語が直接文字列を扱う String クラスを持っているので，Java を用いて今回の実験を行なった．

図 2. から抽出した実行プログラムをまとめると，表 1. のようなシンプルなログデータとなる．このログデータを Java プログラミング言語 [5] を用いて遷移確率表を作成し，ユーザモデルを構築して，個人認証を行う．

```
履歴      操作時間      アクション
[13:33:11] (5:37.103sec) バッテリ メーター -
[13:38:48] (0:07.534sec) Windows Media Player - C:\Program Files (x86)\Windows Media Player\wmplayer.exe
[13:38:55] (0:02.840sec) WinShell - [yoko4*] - C:\Program Files\WinShell\WinShell.exe
[13:38:58] (0:00.109sec) Program Manager -
[13:38:58] (0:00.109sec) Windows エクスプローラー -
[13:38:58] (0:07.972sec) activeWindowRecorder -
[13:39:06] (0:00.327sec) へちまで - Mozilla Firefox - C:\Program Files (x86)\Mozilla Firefox\firefox.exe
[13:39:07] (1:35.660sec) みんなで牧場物語 - Mozilla Firefox - C:\Program Files (x86)\Mozilla Firefox\firefox.exe
[13:40:42] (0:03.495sec) Twitter / @bebelockがフォローしているユーザ - Mozilla Firefox - C:\Program Files (x86)\Mozilla Firefox\firefox.exe
[13:40:46] (0:06.879sec) Twitter / ホーム - Mozilla Firefox - C:\Program Files (x86)\Mozilla Firefox\firefox.exe
[13:40:53] (0:02.730sec) C:\Users\chan\Desktop\地方会2011\yoko4.dvi(1/1) - dviout - c:\tex\dviout\dviout.exe
[13:40:55] (14:02.047sec) WinShell - [yoko4*] - C:\Program Files\WinShell\WinShell.exe
                                    ⋮
```

図 2: 採集データログ例

表 1: 実行プログラム採取例

wmplayer.exe
WinShell.exe
Program Manager
Windows エクスプローラー
activeWindowRecorder
firefox.exe
firefox.exe
firefox.exe
firefox.exe
dviout.exe
WinShell.exe

3.2 認証手法

本研究で用いる手法は，2 章で紹介した入力コマンド系列で使われている遷移確率表による個人認証手法を応用する．本手法は詳しくは 2 つのステップから構成されている．

ステップ 1 では，3.1 章で紹介したようなデータを抽出して，学習させて，各ユーザのログ遷移の特性を計算し，遷移確率表を作成する．ステップ 2 では検証データを用いて，ステップ 1 で得られた遷移確率表と比べて，データが正当なユーザのものであるかどうか判別する．以下各ステップについて詳しく説明する．

3.2.1 Step1-遷移確率表の作成

仮に表 1. を正規ユーザによるログデータ連鎖例とする．次に‘wmplayer.exe’を連鎖を引き起こすログ A とすると，表 1. によれば‘WinShell.exe’が‘wmplayer.exe’の後に続いたので，‘WinShell.exe’が‘wmplayer.exe’の後継するログ B となる．次に‘WinShell.exe’が新たな連鎖を引き起こすログ A となって，‘Program Manager’がその後に続いたので，‘Program Manager’が‘WinShell.exe’の後継するログ B となる．これを、一番最後のデータログの‘WinShell.exe’まで続かせる．今回の例では合計 11 個のログデータがあるから，トータルで 10 回のログデータ連鎖が起こる．

また，‘wmplayer.exe’が今回の連鎖の中での出現回数が 1 回，これを A の合計出現回数とする．‘wmplayer.exe’から‘WinShell.exe’の出現回数が 1 回となっているので，これを A → B の合計出現回数とする．例えば‘firefox.exe’の出現回数が全体で 4 回あったので、この場合，‘firefox.exe’(A) の合計出現回数は 4 となる．‘firefox.exe’から‘firefox.exe’の出現回数が合計で 3 回あったので，この場合、‘firefox.exe → firefox.exe’(A → B) の合計出現回数は 3 となる．ここでは，すべてのデータログ連鎖が一定回数なので，A の合計出現回数と，A → B の合計出現回数の数が一緒でなければならない．これをまとめたものが図 3. の連鎖出現率表である．

連鎖を引き起こすログ(A)	後継するログ(B)	(A)の合計出現回数	(A→B)の合計出現回数
wmplayer.exe	WinShell.exe	1	1
WinShell.exe	Program Manager	1	1
Program Manager	Windows エクスプローラー	1	1
Windows エクスプローラー	activeWindowRecorder	1	1
activeWindowRecorder	firefox.exe	1	1
firefox.exe	firefox.exe	4	3
	dviout.exe		1
dviout.exe	WinShell.exe	1	1

図 3: 連鎖出現率表

$$P(A \to B) = \frac{N_{AB}}{N_A} \tag{1}$$

式 1. は，あるログ A からログ B への全体のログデータ連鎖の中での出現確率を表したものである．N_{AB} はログ A からログ B への連鎖出現回数、つまりログ A →

B の合計出現回数．N_A はあるログ A の合計出現回数を表したものである．例えばこの例の場合では，あるログ A を ‘ firefox.exe ’ と考えると，図 3 を参照して，‘ firefox.exe ’ から ‘ firefox.exe ’ の連鎖は全部で 3 回出現している．また，‘ firefox.exe ’ の出現は 4 回出現している．つまり，‘ firefox.exe ’ が N_A となり，‘ firefox.exe → firefox.exe ’ が N_{AB} となる．式 (1) に代入すると，

$$P(firefox.exe \rightarrow firefox.exe) = \frac{3}{4} = 0.75 \quad (2)$$

これと対照に，‘ firefox.exe ’ から ‘ dviout.exe ’ の連鎖は全部で 1 回出現しているから，

$$P(firefox.exe \rightarrow dviout.exe) = \frac{1}{4} = 0.25 \quad (3)$$

このように，すべてのデータログ連鎖を式 1. から，式 2.3. のように求めると，図 4. のような遷移確率表を求めることができる．

後継するログ

連鎖を引き起こすログ	wmplayer.exe	WinShell.exe	Program Manager	Windows エクスプローラー	activeWindowRecorder	firefox.exe	dviout.exe
wmplayer.exe	0	1	0	0	0	0	0
WinShell.exe	0	0	1	0	0	0	0
Program Manager	0	0	0	1	0	0	0
Windows エクスプローラー	0	0	0	0	1	0	0
activeWindow Recorder	0	0	0	0	0	1	0
firefox.exe	0	0	0	0	0	0.75	0.25
dviout.exe	0	1	0	0	0	0	0

図 4: 遷移確率表

3.2.2 Step2-個人認証

ステップ 2 では，ステップ 1 で求めた図 4. の遷移確率表を用いて，検証用データから正当なユーザであるかどうか判別を行う．

表 2: 検証用データログ例

1.firefox.exe
2.dviout.exe
3.WinShell.exe
4.firefox.exe
5.dviout.exe
6.firefox.exe
7.scp.pdf

例えば表 2. のような検証用データログの例があるとする．表 2. から図 4. で求めた数値を引用して．1 番目の ‘ firefox.exe ’ から 2 番目の ‘ dviout.exe ’ の連鎖出現確率が 0.25 であるから．それを代入すると，$P(firefox.exe \rightarrow dviout.exe) = 0.25$ となる（式 3. で求めた数値）．これと同様に，この検証用ログデータの連鎖にすべての数値を代入すると，表 3. にまとめることができる．

表 3: 図 3 より抽出した数値表

回数 n	データログ連鎖	遷移確率 K
1	firefox.exe → dviout.exe	0.25
2	dviout.exe → WinShell.exe	1
3	WinShell.exe → firefox.exe	0
4	firefox.exe → dviout.exe	0.25
5	dviout.exe → firefox.exe	0
6	firefox.exe → scp.pdf	0

$$P_n = \sum^{n} K/n \quad (4)$$

式 4. は，表 3. でまとめた数値の平均値を撮ったものである．ここでは，この P_n を遷移確率平均値と呼ぶこととする．n は連鎖の出現回数，K はそのデータログ連鎖の図 4. で取れた遷移確率の数値を表している．すべての K の和を n 回で割ると，検証用データログ連鎖の繊維確率の平均値を求めることができる．今回の連鎖回数は全部で 6 回あり、$n = 6$ となり，その遷移確率の和を求め，6 で割ると式 5. のようになる．

$$P_6 = \frac{0.25 + 1 + 0 + 0.25 + 0 + 0}{6} = 0.333 \quad (5)$$

ここで，式 5. で求められた遷移確率平均値が高ければ高いほど，マウスクリックが似たようなパターンである可能性が高いので，本人の可能性も高い．反対に，この値が低ければ，本人の利用パターンと違うものが多いということになるので，他人である可能性が高い．従ってこの値に下限値を儲ければ，下限値以上であれば正当なユーザと判断し，下限値よりしたの値になれば不当なユーザであると判断させることができる．これによって，本人であるかどうかシステムに判断させる．

下限値の決定は，比較する対象によって変動するので，また 4 章と 5 章で詳しく説明することとする．

4. 実験

今回の実験では，本研究室の学生 6 名と機械工学科の学生 6 名，一般の方 4 名の計 16 名からそれぞれのロ

グデータを採集し，3 章で提案した手法を用いて検証した．この 16 名 (被験者 1 〜16 とする) のコンピュータ利用率と熟練度を以下の表 4. にまとめた．

この実験では，まず採取したログデータを二分し，それぞれのファイルに収納した．前半のログデータをステップ 1 (data1) の学習ログデータとして保存し，そのデータログ連鎖の遷移がどの程度の確率で生じたかを調べる．3 章の図 4. のような遷移確率表を，被験者ごとに予め計算しておき，求めた数値をそれぞれの遷移確率表として保存しておく．後半のログデータをステップ 2 (data2) の検証用ログデータとし，ステップ 1 で作られた遷移確率表より，3.2.2 章で述べた表 3. で方法で，平均値を計算する．

表 4: 被験者の熟練度

ユーザ	グループ	熟練度	PC 利用度合い
被験者 1 被験者 2 被験者 3 被験者 4 被験者 5 被験者 6	本研究室の学生	上級	研究内容が主にコンピュータ関係なので、PC 利用度は高い
被験者 7 被験者 8 被験者 9 被験者 10 被験者 11 被験者 12	機械工学科の学生	中級	学校では主に実験データの整理や、文書作成
被験者 13 被験者 14 被験者 15 被験者 16	一般の方	初級	主にインターネットや、ドキュメント作成

ログデータは Windows システムをマウスで操作する際に発生するイベントログを採取した．この実験では、まず採取したログデータを二分し、それぞれのファイルに収納した．前半のログデータをステップ 1 (data1) の学習ログデータとして保存し，そのデータログ連鎖の遷移がどの程度の確率で生じたかを調べる．3 章の図 7. のような遷移確率表を，被験者ごとに予め計算しておき，求めた数値をそれぞれの遷移確率表として保存しておく．後半のログデータをステップ 2 (data2) の検証用ログデータとし，ステップ 1 で作られた遷移確率表より，3 章で述べた式 4. を用いて，平均値を計算する．

また今回は，16 人の被験者から算出された遷移確率表から，それぞれのグループ内で交互にステップ 2 で遷移確率平均値を求め，比較をする．例えばグループ 1 の被験者 1 で得られた遷移確率表を，被験者 1 だけの P_n(式 4.) を求めるのでなく，被験者 1 から 6 までそれぞれ遷移確率平均値を求める．グループ 2 の被験者 7 では，被験者 7 〜12 までのそれぞれの遷移確率平均値を求める．

ここで，被験者 1 の遷移確率表で得られた結果を被験者 1 の検証用ログデータで P_n を $P1_{n_1}$ と表現することにする．この時，遷移確率表を得た被験者のことを i とする．次に，被験者 1 の遷移確率表で得られた結果を被験者 2 の検証用ログデータの遷移確率平均値は $P1_{n_2}$ と表現できる．この時，検証用ログデータを用いて遷移確率平均値を求めたユーザのことを j とすると，被験者 i から被験者 j のデータでも求めた P_n は，「Pi_{n_j}」と表すことができる．

表 5: ログデータ採取量

ユーザ	data1	data2	合計
被験者 1	2719	2720	5439
被験者 2	1600	1601	3201
被験者 3	2414	2414	4828
被験者 4	1756	1757	3513
被験者 5	2821	2820	5641
被験者 6	2262	2262	4524
被験者 7	1851	1851	3702
被験者 8	1920	1920	3840
被験者 9	2068	2067	4135
被験者 10	2361	2361	4722
被験者 11	1729	1730	3459
被験者 12	2266	2266	4532
被験者 13	1673	1672	3345
被験者 14	1178	1178	2356
被験者 15	2006	2006	4012
被験者 16	1601	1601	3202
合計	32225	32226	64451

なお，今回の実験で使用したログデータの採取期間は 3 日間で，作業内容はそれぞれのグループによって大きな違いがある．被験者 1 〜6 は，プログラム，システム開発および資料作成が中心である．被験者 7 〜12 は，インターネット閲覧や，実験で得たデータの編集，文章作成が中心となっている．また，被験者 13〜16 では，主にインターネットの利用が多かったように見受

けられる．表 5. は，実験を行うためのログのデータの具体的な採取量である．ここでの数値の単位は，マウスが一回クリックした数を 1 回としている．表 6. はこれらのデータを計算した実際の計算環境をまとめたものである．

表 6: 計算環境

CPU	Intel(R) Core(TM) i7 CPU 920
Memory	8192MB RAM
Operating System	Windows 7 Professional 64-bit
resolution	1980 × 1200
HDD	1T
PG language	Java

5. 実験結果

前節で述べた実験を行ない，各グループの結果が表 7.〜9. にまとめた．

網表示した数値は，それぞれ被験者自身での検出遷移確率平均値 (Pi_{n_j}) を表している (図 5. のひし型の点)．この数値が高ければ高いほど，被験者自身での類似度が高いということになる．つまり，検出しやすいということである．図 5. で示した四角の点は，各被験者での他人に対する遷移確率平均値での最低値を表している．

下限値の決定については，システム管理者がユーザの利用状況と類似度を総じて判断し，決めなければならない．しかし利用者が増えればその分，類似度も似てくるので，下限値をあまり高く設定すると，正当なユーザもはじかれるかもしれない．そのため，図 5. のグラフによれば，下限値が 0.3 にすることで，多くの不正利用を検出できると考えられる．

また，本研究で提案した検知手法を普段の ID とパスワードによる認証方式と併用すれば，システムの安全度が上がると考えられる．

6. 考察

この検証手法では，Windows システムを利用する際に用いるマウス操作からクリックイベントログを採集することができた．また，採集したログデータを Java プログラミング言語で実行ファイルの部分だけを抽出した．Linux システムでの入力コマンド系列連鎖の遷移確率表を用いた個人認証と同じような方法を用いて検証実験を行なった．そのため，入力コマンド系列と

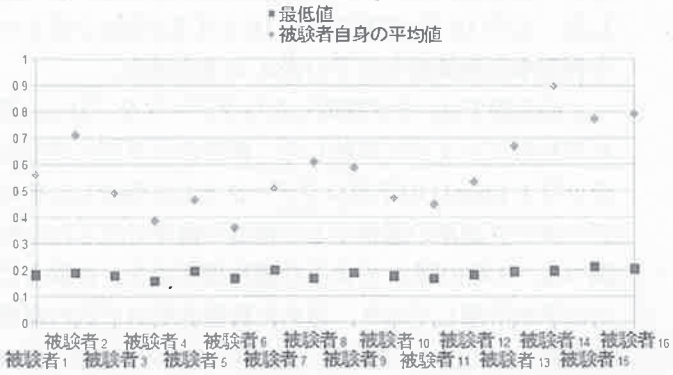

図 5: 移動量グラフ

似たような結果が得られ、期待した値が得られたとも言える．

今回の実験では，GUI システムで抽出したログデータは，入力コマンド系列とはまたいくつもの異なる性格を持っていることがわかった．両者とも文字列として処理している点では同じだが，入力コマンド系列では，コマンドを入力するため，覚えているコマンド数に個人差がある．ユーザそれぞれの熟練度によっては使っているコマンドが限られている．従って，初心者と熟練者では大きな差があり，ある程度個人判別の精度が高いことが多い [2]．それに対して Windows システムでは，直感的な操作が多いため，ユーザによってはどういう風に操作するかは，人それぞれの違いが生じ安く，どういう風にまたはどのようなログを採集したらいいか，まだまだ課題が多いことを感じた．これは Windows システムの欠点であり，特徴抽出においては利点であるとも言える．ユーザの特徴の最適な抽出方法を見つかれば，個人認証の効果が飛躍的に上がると思われる．

下限値の決定については，すべての利用者で同じ下限値だと検出率が悪いことも，今回の実験でわかった．一番下の基準にしてしまえば，正当なユーザに対する誤検出率こそは高いものの，不当なユーザの検出率も下がり，不正利用を見落としてしまう．逆に下限値の基準を一番上に設定すると，今度は正当なユーザまでもが不正ユーザであると判断され，システムを使えなくなってしまう．従って，システムの利用対象によって，システム管理者が下限値を見極める必要がある．しかし，この手法を従来の ID とパスワードによる認証方式と併用すれば，ある程度下限値が低くても，効果が見られると考えられる．そのため，この提案手法は有用であると言える．

一方，本実験では 16 人の被験者しか使わなかったの

表 7: グループ 1 での計算結果

data1data2	被験者 1	被験者 2	被験者 3	被験者 4	被験者 5	被験者 6
被験者 1	0.564	0.305	0.419	0.223	0.340	0.234
被験者 2	0.327	0.712	0.215	0.356	0.232	0.282
被験者 3	0.368	0.211	0.495	0.347	0.217	0.332
被験者 4	0.158	0.260	0.201	0.387	0.163	0.208
被験者 5	0.221	0.331	0.194	0.243	0.466	0.331
被験者 6	0.323	0.298	0.177	0.333	0.212	0.361

表 8: グループ 2 での計算結果

data1data2	被験者 7	被験者 8	被験者 9	被験者 10	被験者 11	被験者 12
被験者 7	0.512	0.379	0.421	0.323	0.416	0.366
被験者 8	0.419	0.610	0.390	0.338	0.459	0.478
被験者 9	0.510	0.412	0.588	0.392	0.356	0.465
被験者 10	0.389	0.340	0.322	0.476	0.401	0.299
被験者 11	0.331	0.321	0.284	0.371	0.449	0.268
被験者 12	0.419	0.292	0.367	0.338	0.444	0.535

表 9: グループ 3 での計算結果

data1data2	被験者 13	被験者 14	被験者 15	被験者 16
被験者 13	0.672	0.446	0.523	0.492
被験者 14	0.688	0.901	0.701	0.693
被験者 15	0.416	0.556	0.774	0.498
被験者 16	0.525	0.497	0.539	0.793

で，実験の収束性を見いだせることはできなかった．そのため，今後被験者とログデータを増やして再実験する必要があると考えられる．

今後の課題として，データ量をもっと増やして，同じような実験を行うと，どのような結果になるか，検証する必要がある．そして，データセッションをどういう風に扱うか，ほかにどんなログデータをとって，どんな結果になるかも検証しなければならない．また，２章で紹介したファジィ測度による検出手法を用いて，類似度に比重を置くと結果がどのように変化するか，あるいはもっとうまく個人特徴を抽出出来るか，様々な検証が必要である．

7. まとめ

本稿では対話的計算機環境下で，ユーザがマウスやキーボードを利用する際に見られる個人の特徴に注目した．特にマウスを利用する時は，人によっては移動する座標や軌跡が違ったり，移動速度やクリックする速さが違ったりする．今回の実験は，マウスクリックする時に発生するイベントに注目した．ユーザがマウスを利用する際のマウスクリックイベントをデータとして溜め込み，そこから個人の特徴を抽出し，それを利用した侵入者検出法の一つを提案した．

その提案手法に基づいて，ユーザがGUIシステムの利用するときに用いるマウスから，マウスクリックイベントを抽出した．そこから入力コマンド連鎖の頻度によるユーザ認証手法を利用して，GUIシステム上でも実装し，効果を検証した．本稿で提案したのは，遷移確率表を用いる認証手法である．本手法では二つのステップから認証を行う．まず採取したデータを２分割し，ステップ１として前半のデータを学習データとして，遷移確率表を作成した．ステップ２では分割した後半のデータを検証用データとして，遷移確率表から得た遷移確率の合計の平均値を取った．その平均値の高低で正規なユーザであるかどうか判断することである．

この認証方法の有効性をを検証するため，パソコンに対する熟練度の違う16人の被験者のログデータを採集し，実験を行なった．実験では，上記手法を用いて二つのステップで検証を行い，Linuxシステムと似たような結果が得られた．これは，この手法の有用性を証明したとも言える．

また，今回の認証手法では，作業内容が限定されていれば個人の遷移確率も比較的高くなるが，多岐にわたればそれだけ確率が下がることを結果から確認できた．本手法は比較的システムでの実装が容易で，従来のパスワードなどの認証手法と併用すれば，システムセキュリティの強化に役立てると考えられる．また，遷移確率を生成する最適条件を見つけることや，工夫することにより，認証の精度も向上する．

参考文献

[1] 宮崎小玉, 白井治彦, 黒岩丈介, 小高知宏, 小倉久和. 入力コマンド系列の評価による侵入者検出手法のGUIシステムへのアプローチ. 平成23年度電気関係学会北陸支部連合大会,E-21

[2] 白井治彦, 西野順次, 小高知宏, 小倉久和. 対話的計算機環境におけるコマンド入力連鎖を用いた認証手法の提案. 信学論 (A),Vol.182-A.No.10,pp.1602-1611.Oct.1999

[3] 井上善夫, 白井治彦, 高橋勇, 黒岩丈介, 小高知宏, 小倉久和. 入力コマンド連鎖の出現頻度に基づく個人認証手法の提案. 福井大学工学部研究報告,Vol.54.No.1,March.2006

[4] 白井治彦, 黒岩丈介, 小高知宏, 小倉久和. ファジィ測度に基づいた入力コマンド系列の評価による侵入者検出手法のSchonlauデータセットに対する効果. 知能と情報 (日本知能情報ファジィ学会誌),Vol.21,No.5,pp.804-814,2009

[5] 柴田望洋. 解明—Javaによるアルゴリズムとデータ構造. SoftBank Creative,2010