キーストロークパターンによるログイン認証*

渡辺幸樹† 福永孝† 塙 敏博‡

東京工科大学 工学部 情報工学科†
東京工科大学 コンピュータサイエンス学部‡

1 はじめに

現在コンピュータ上での認証には，主に ID とパスワードのペアが使用されている．しかしこの方法は，他人によるなりすましの可能性があり安全性に欠ける．そこで，本人以外は同じ特徴を持たず，時間の経過と共に変化しない性質を利用した，バイオメトリクスによる個人認証が使われてきている．

本研究では，バイオメトリクスのうち，キーストロークパターンによる認証技術を検討し，認証精度の向上と実際のログインへの応用を目的とする．

2 キーストロークパターン認証

キーストロークパターン認証とは，キー入力のタイミングの相違により，本人を識別する認証方法である．通常のパスワード入力と操作は変わらず，特別なハードウェアを追加する必要もない．

特徴点の取得 入力文字列のうち，ある 2 文字を 1 組としたものを digraph と呼ぶ．各 digraph において，それぞれキーのプレスとリリースの時刻を取得し，その総当たりの時間差を特徴点とする．従って 1 つの digraph から 6 個の特徴点が取得できる．‘a’ と ‘b’ の digraph の例を図 1 に示す．

入力文字列が “abcdef” の場合，digraph を隣接 2 文字間に限定すると，文字列は，“ab”, “bc”,“ cd”, “de”, “ef” の 5 つに分割され，特徴点は 30 通りになる．重複している ‘a’, ‘b’, ‘c’, ‘d’, ‘e’ のプレスとリリースの差を除くと，特徴点の数は合計で 26 個になる．また，入力文字について総当たりに digraph を構成すると 66 個の特徴点が取得できる．これらの特徴点を数回取得し，その平均値と標準偏差を認証のテンプレートとする．

一致判定 認証時は, テンプレートから平均値 *ave*, 標準偏差 *sd* を用いて認証スケールを N とすると，入力値から求めた T が以下の式を満たすときに特徴点と一致したとみなす．

$$ave - N \times sd < T < ave + N \times sd \qquad (1)$$

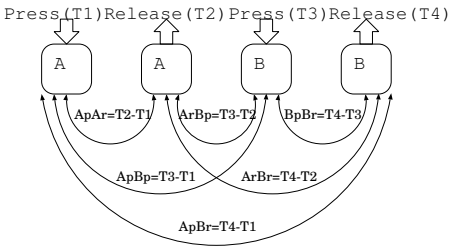

図 1: digraph からの特徴点の求め方

3 認証精度の向上

上に述べた認証方法のままでは，本人拒否率，他人受入率共に大きな値となり，実用性に乏しい．そこで，テンプレートの作成方法を改善することにより認証精度の向上を図る．(1) 式より，テンプレートの標準偏差が大きいと他人受入率が大きくなるため，テンプレート作成時に中央値から 2 倍以上離れているデータは削除し，削除した回数だけ再入力させる．

また，キーボード入力の習熟により入力のタイミングが変化する．これにより本人拒否率が増加するのを防ぐためにテンプレートの更新を行なう．認証が受理された際に，テンプレート内のデータのうち最も古いデータと，認証時の入力データを入れ替える．ただし，更新を繰り返すと，新しく認証を通過したデータは以前のデータより平均に近いことが多く，標準偏差が小さくなり認証率が低下する．そのため，標準偏差の最小値を 15 に設定した．

4 評価

まず，*sd* の値の制限による効果を測定した．表 1 は，6 文字を入力し，digraph が 2 文字間の場合 digraph(2) と総当りの場合 digraph(6)，digraph が 6 文字間の時の *sd* の最小値を 15 に制限した場合，さらに 66 個の特徴点のうちエラー数 3 個，エラー数 5 個までを無視した場合の結果である．認証スケールは 3 である．テスト人数は 5 人，それぞれが自分と他 2 人に対して 40 回ずつ入力し，合計で 3000 回の認証試行を行った．自分以外の他 2 人は平均値と標準偏差の近い者を選んだ．

表 1: パラメータの比較

	本人拒否率	他人受入率
試行回数	40 回	40 回
digraph(2)	67.0	0.0
digraph(6)	62.5	0.0
digraph(6) ($sd \geq 15, error = 0$)	43.7	0.3
digraph(6) ($sd \geq 15, error \leq 3$)	14.5	0.0
digraph(6) ($sd \geq 15, error \leq 5$)	12.5	0.0

*Login Authentication using Keystroke Patterns

†Kouki WATANABE, Takashi FUKUNAGA, Dept. of Information Technology, Tokyo University of Technology

‡HANAWA Toshihiro, School of Computer Science, Tokyo University of Technology

digraph を総当りに取ることで本人拒否率が 5%低下し，*sd* の最小値を制限すると，さらに認証率の低下を抑えることができた．また，この条件では他人のなりすましはほぼ 0 であった．

次に，テンプレートの更新の効果を測定した．入力文字列はそれぞれが普段使い慣れているログイン名とし，登録時のテンプレートに対し認証試行を 10 回，さらにその 10 回の試行で更新されたテンプレートに対し認証試行を 10 回行なった．*sd* の最小値を 15，認証スケールを 3 とし，特徴点が 92%以上一致すれば認証を許可する．テスト人数は 16 人，1 人に対し 3 人がなりすましを行い，なりすましを行なう者には本人の入力を見せた．それぞれが自分への認証試行を 20 回，他人への認証試行を 60 回，合計で 1280 回の認証試行を行なった．

sd と認証率の変化を *sd* の平均により二つに分け，その一部を表 2，表 3 に示す．

表 2: テンプレート登録時の *sd* の平均が 30 未満

被験者 (文字数)		*ave*	*sd*	$sd_{\max}$	本人拒否率	他人受入率
A(5)	登録時	174.8	12.9	29.4	0.0	0.0
	10 回目	168.4	12.7	27.9	0.0	0.0
B(6)	登録時	179.1	18.2	33.4	10.0	3.3
	10 回目	177.1	15.8	26.8	20.0	0.0
C(11)	登録時	342.6	29.0	59.7	20.0	0.0
	10 回目	344.0	24.0	51.9	0.0	0.0

表 3: テンプレート登録時の *sd* の平均が 30 以上

被験者 (文字数)		*ave*	*sd*	$sd_{\max}$	本人拒否率	他人受入率
D(4)	登録時	179.1	34.6	67.7	30.0	53.3
	10 回目	175.0	14.5	26.0	10.0	3.3
E(6)	登録時	200.2	50.2	91.8	20.0	80.0
	10 回目	187.1	26.0	44.5	30.0	10.0

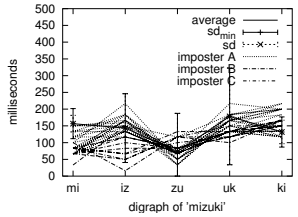

図 2: 登録時のテンプレートに対する他人の試行

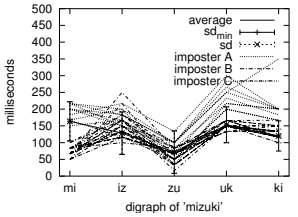

図 3: 試行 10 回目のテンプレートに対する他人の試行

sd の平均が 30 未満の者は 16 件中 11 件であった．表 2 から分かるように，安定した入力ができテンプレート登録時から *sd* の低いものは，本人拒否率も他人受入率も低い．

図 2，図 3 は E 氏のテンプレートに対する他人の試行である．隣接する文字のキープレスの差で，66 個の特徴点のうちの 5 個の特徴点である．E 氏は登録時の *sd* が最も大きく，10 回の試行で他人受入率が最も改善された者である．図 3 を見ると，digraph “uk” の *sd* が減少し他人を拒否しやすくなっているのが分かる．“uk” の *sd* は 48.1 から 17.9 まで下がっていた．表 3 から分かるように，テンプレート登録時の *sd* が高く他人になりすまされやすい者でも，10 回の認証試行でテンプレートが入れ替われば認証率は大きく改善される．

5 実装

5.1 UNIX のログインへの応用

UNIX コンソールからのログイン時は，キーリリースの時刻が取得できない．そのため，キープレスの時刻のみで特徴点を決定する必要がある．総当たりで文字の組を決定し，その時刻の差を特徴点とする．入力文字列が 6 文字の場合，15 個の特徴点が取得できる．

キー入力の直後に時刻を取得するために，端末を cbreak モードに変更し，エコーバックを止めた上で，一度に 1 文字ずつ読み込む．実装の際は PAM のモジュールとして実現する．PAM とは，ログインや他の様々な認証を 1 箇所にまとめるための枠組である．PAM を用いることで，UNIX のログインだけでなく，他の認証にも使用することができる．

表 2，表 3 で使用したデータを，キープレスのみで判定した場合の A 氏と D 氏の結果を表 4 に示す．キーリリースを使用した場合と同様，高い認証率を得ることができることがわかる．

表 4: キープレスの結果

被験者 (文字数)		*ave*	*sd*	$sd_{\max}$	本人拒否率	他人受入率
A(5)	登録時	277.7	17.2	29.4	0.0	0.0
	10 回目	265.7	16.8	27.9	0.0	0.0
D(4)	登録時	324.2	41.0	67.7	30.0	53.3
	10 回目	310.1	13.9	19.7	10.0	6.7

5.2 WEB のログインへの応用

Web のログインには Java アプレットを使用する．Java アプレットのキーイベントにより keyPress と keyRelease の時刻を取得し，サーバに送信する．サーバでは，受け取った時刻を元にテンプレートを作成し，同様に認証の処理を行なう．

6 結論

本研究の結果，テンプレート登録時に *sd* が 30 以下に抑えられていれば，本人拒否率は低く，また他人がキーストロークを真似たとしても，なりすましは困難であることを示した．

テンプレート登録の際に，例外的なデータは削除し *sd* が極端に上がることは防いたが，常に *sd* を 30 以下に抑えることはできていない．*sd* が 30 以下になるまで再入力させる方法が考えられるが．テンプレート登録の負担が大きくなるため，他の有効な手段を検討する必要がある．

参考文献

[1] Rick Joyce and Gopal Gupta. “Identity Authorization Based on Keystroke Latencies,” *Communications of the ACM*, 33(2): 168-176, February 1990.

[2] 粕川正充，角田博保，森裕子 :「アルペジオ打鍵列を利用した個人認証手法の提案」: 情報処理学会論文誌，Vol.34，No.5，pp.1198-1205 (1993)