3W-01

誤入力の傾向を考慮した Levenshtein 距離による柔軟なパスワード認証システム

小林怜央 *　黒米祐馬 †　武田圭史 ‡　村井純 ‡

1 序論

現在，ユーザの ID およびパスワードの完全な一致によりユーザを識別する手法であるパスワード認証が，多くの Web サイトで利用されている．しかし，一般に入力後のパスワードは単一の符号に変換されて表示されるため，自分の誤入力の有無を確認することができない．にも関わらず，現状のほとんどのパスワード認証は，パスワード入力時のタイピングミスを許容していないため, ユーザビリティが低下している．だが，入力後のパスワードの返還表示は，簡単なパスワードを設定する原因の一つであり，パスワードをそのまま表示できるようにするべきという意見もあるものの [1]，この方式は，パスワードの覗き見に対しては効果的な対策であるため，一概に廃止するべきとは言えない．

そこで，本研究ではタイピングミスを考慮した Levenshtein 距離を提案する．これにより，パスワードのタイピングミスを先行研究より柔軟に判別・許容をすることで，一定のセキュリティの基準を満たしながらも，ユーザビリティを向上させる．

2 提案手法

本研究では，パスワード認証を使用して個人を判別する Web サイトに対して，ユーザが誤ったパスワードを入力した時に，その誤りがタイピングミスに起因するものかどうか分析することで，次回以降同じ誤入力をしても，パスワード認証を成功させる．具体的には次の手順を取る．

1. 誤ったパスワードによる入力試行があった場合，誤ったパスワードを破棄せず，暗号化した上で期限付きでデータベースに保存する．
2. 同一ユーザ･同一セッションで正しいパスワードによるログイン試行が行われた場合に，それまでに誤って入力したパスワードをデータベースから参照および復号化する．
3. 正しいパスワードと誤ったパスワードを比較することで，正しいパスワードを入力する意思のもとタイピングミスしたのかどうか判別する．
4. 正しく入力する意思があったと判定された場合には，誤ったパスワードを正しいパスワードと同様にハッシュ化した後データベースに保存し，以降同じタイピングミスをした場合に認証を成功させる．ただし，許容するタイピングミスのパスワードの数を一定数に制限し，また，一定数を超えた場合には最も使用されていないものをデータベースから削除することで，無制限に許容するパスワードが増えることを防ぐ．

3 実装

3.1 Web サイト

Web サイトは次の順序に沿って動作する．

1. ログインするユーザの同一性を担保するために，ログインの試行をしてきたユーザに，ブラウザの Cookie を使用して，32 バイトの乱数をセッション ID として送付する．
2. ユーザはセッション ID を含んだ Cookie とともに ID とパスワードをサーバに送付する．
3. ユーザが入力したパスワードが誤っていた場合には，セッション ID の先頭 16 バイトをユーザの識別子，末尾 16 バイトを鍵として暗号化された誤ったパスワード，そしてデータベースから自動的に削除するまでの時間をデータベースに保存する．
4. ユーザが正しいパスワードを入力できないままにあきらめた場合には，一定時間でデータベースからパスワードの入力ログは削除する．
5. 一定時間以内に正しいパスワードでログインに成功した場合には，成功したユーザのセッション ID を利用してそのユーザがログインするまでに送信してきた誤ったパスワードを抽出及び，復号化して，最後にユーザが入力した正しいパスワードと比較する．

3.2 タイピングミスの判定

Levenshtein 距離は，２つの文字列の違いを定量的にあらわす手法である. 本研究では，Shift の押し間違い・押すキーボードが 1 つずれる・入力する文字列が反転するというタイピングミスに起因する違いの場合の距離を近くすることで，誤入力を許容する．通常の Levenshtein 距離では 1 文字の削除・挿入・置換を 1 コストとした上で，何コストで比較する文字列を変更できるかを測るアルゴリズムであるが，タイピングミスを考慮した Levenshtein 距離では上記のパターンによる置換えのコストを 0.1 とする．そして，誤ったパスワードのうち，この距離の値を正しいパ

*慶應義塾大学 総合政策学部
†慶應義塾大学 環境情報学部
‡慶應義塾大学

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.

スワードの長さで割った値が一定値を下回るものをタイピングミスに起因するミスと判定する．

4 先行研究

Chatterjee らは，タイピングミスを許容するログインシステムを提案している [2]．彼らが提案しているシステムは，先頭が大文字になる場合・CapsLock が意図せず有効になった場合・終端に余計な文字が付いている場合 3 パターンのタイピングミスを許容するというものである．また，彼らは，パスワードの試行回数制限の機能を設けていてかつ，容易に推測できるパスワードを使用していない場合，正しいパスワードに類似したパスワードを許容することは，セキュリティ上の問題とならないことを示している．

本研究は，この手法を拡張し，より柔軟にタイピングミスを許容することで，さらなるユーザビリティの向上に寄与するものである．本研究でもパスワードの試行回数制限を設け，弱いパスワードは適用範囲外とすることでセキュリティを担保する．

5 評価手法

本手法により算出した，タイピングミスした文字列と入力しようとしていた文字列との距離を入力すべき文字列長で割った値と，従来の Levenshtein 距離を入力すべき文字列長で割った値を比較することで，本手法を評価する．なお，タイピングミスした文字列と正しく入力しようとした文字列のデータは以下のような Web サイトを公開することで収集する．

- アルファベット大文字・小文字・数字が全て混在している 10 文字以上の文字列を被験者に入力させる．
- 画面に中身が表示されない password 属性の入力欄 10 個全てに最初に入力した文字列を被験者に入力させる．
- クリップボードの利用・ブラウザの自動補完は無効化する．

6 実験

上述の条件で平成 28 年 12 月 14 日から平成 29 年 1 月 4 日にかけてデータを収集した．ここで，悪戯目的・攻撃目的・途中で断念と考えられる入力が存在したため，通常の Levenshtein 距離を入力するべき文字列長で割った値が 0.5 以上のものを除外することでデータをスクリーニングした．0.5 を閾値とする根拠は，図 1 の通り，0.5 を境として，データが二分されており，また入力の半分以上をタイピングミスすることは通常考えられないためである．

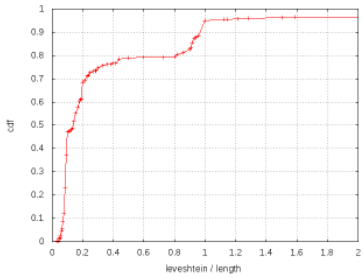

Figure 1: **全体の Levenshtein 距離 (n=569)**

続いて，入力しようとした文字列とタイピングミスした文字列のペアが重複したものを，スクリーニングしたデータを用いて，全てのペアの従来の Levenshtein 距離を入力しようとした文字列長で割った値と，本手法を用いて算出した値を入力すべき文字列長で割った値を計算する．

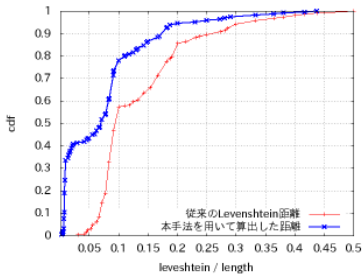

Figure 2: **二種の Levenshtein 距離の比較 (n=223)**

図 2 は青が本手法を用いて算出した距離をあらわしており，赤が従来の Levenshtein 距離をあらわしている．

7 考察

図 2 より，本手法を用いて算出した値を入力しようとした文字列の長さで割った値は，0.045 付近が少なくなっており，その前後でグラフを 2 区間に分けることが可能である． 2 区間のうち後ろの区間は傾きが緩やかなものの，従来の Levenshtein 距離を用いた CDF と類似した形状である．しかし，前半の区間では従来の Levenshtein 距離を用いた場合の CDF は伸びておらず，本手法を用いた場合の CDF のみが大きく伸びている．これらのことから，タイピングミス全体の約 4 割ほどが，本手法を適用すると，従来の Levenshtein 距離を適用する場合と比べ，入力しようとした文字列の長さでそれぞれの距離を割った値が小さくなっていることがわかる．

タイピングミスの残りの 6 割のタイピングミスを抽出して，目視で確認したところ，今回対応していない余計な文字の挿入，および，文字の抜け落ちが原因のものが多く確認することができた．そこで，不必要な文字の削除・抜け落ちた文字の挿入のコストを下げる処理を本手法に実装することにより起こりうる，誤ってタイピングミスでないものをタイピングミスと認識しまうリスクの定量的な分析とその対策を今後の課題としたい．

8 参考文献

References

[1] 徳丸浩『COOKPAD の「伏せ字にせず入力」ボタンは素晴らしい』徳丸浩の日記 （最終閲覧日：2016 年 1 月 13 日）
http://blog.tokumaru.org/2012/06/cookpad-display-password.html

[2] R. Chatterjee et al.: “pASSWORD tYPOS and How to Correct Them Securely”, Security and Privacy (SP), 2016 IEEE Symposium on (May 2016).

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.