5Z-1

計算式の選択が可能なワンタイムパスワードの実装

Implementation of one time password featuring selectable computation

氏　名† 木津　良太　　氏　名‡ 宇田川　佳久

所　属† ‡ 東京工芸大学工学部コンピュータ応用学科

Kitsu Ryota: Tokyo Polytechnic University　Yoshihisa Udagawa: Tokyo Polytechnic University

1.背景

情報システムが社会基盤となった今日、個人情報保護を始めとしたセキュリティの強化が社会的に重要視されている。個人認証はセキュリティの基本機能であり、本研究では、ワンタイムパスワードによる個人認証に着目した。これまでのワンタイムパスワードは、チャレンジコードからワンタイムパスワードを生成する計算方式が固定されていた為、一度、計算方式が漏えいするとワンタイムパスワードが機能しなくなる危惧がある。本研究では計算方式をボタンで指定出来る様にしたワンタイムパスワードを提案し、Java による実装結果を示す。

2.ワンタイムパスワードについて

従来の固定パスワードによる認証方式では、繰返し、同じパスワードを使うので、パスワード漏洩の危険性を否定できない。ワンタイムパスワードは、ユーザ認証方式の一つであり、パスワードが毎回異なる文字列になることを特徴とするものである。ワンタイムパスワードでは、毎回異なる文字列がパスワードとして使用されることから、パスワード漏洩のリスクを低減できる。

ワンタイムパスワードの方式としては、数学的アルゴリズム方式、時刻同期方式、チャレンジレスポンス方式などがある。本文では、チャレンジレスポンス方式に基づくワンタイムパスワードについて述べている。

3. チャレンジレスポンス方式

典型的なチャレンジレスポンス方式に基づくユーザ認証は、下記の手順で行われる[1]。
(1)ユーザは登録されている ID を入力する
(2)要求を受けたサーバーは毎回異なるチャレンジコードを生成し、クライアント送付する。
(3)ユーザは、あらかじめ定められた方法に基づいて、送られてきたチャレンジコードと固定パスワードからレスポンスコードと呼ばれるワンタイムパスワードを生成し、サーバーに送り返す。(4)サーバー側でもクライアントと同じ手順でレスポンスコードを作成する。(5)送られてきたレスポンスコードと作成したレスポンスコードを比較し、合っていればログインを許可する。

チャレンジレスポンス方式の代表的なものに、数字のマトリックスをチャレンジとし、数字のマトリックスにおける位置情報に基づいてパスワードを生成する方法がある。位置情報に基づくことから、パスワード漏洩のリスクがあると考えられる。他に、時間とともに変化するトークンを使う方法があるが、この方法ではトークンを常に持ち歩かなくてはいけない、紛失・盗難の心配の恐れなどがある。

4.提案するワンタイムパスワード

4.1　ワンタイムパスワードの考察

本研究で提案するワンタイムパスワードは様々な選択が可能な物である。
(1)セキュリティトークンは使用しない。問題点でも書いた通り持ち歩き、紛失・盗難の心配の可能性がある為である。(2)入力後、入力に合ったボタンを押す事で比較・認証する。その為、従来の物と比べ、関連性・規則性を見つけにくくなる。なお、これで、脆弱性が無くなったとは言えないがマトリックス認証のように位置情報だけを覚えて入力する物よりはリスクが少ないと考えられる。

4.2　実装内容

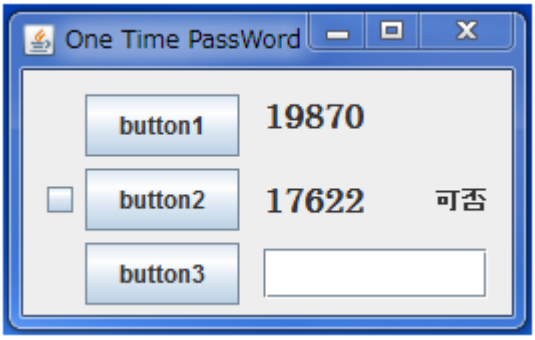
今回、GUI を使ったワンタイムパスワードを目的としている。その為、出力画面はシンプルをベースにし、分かりやすさに重点を置く。

ボタン類に内容を記載するのは、盗み見られる可能性があるので、使用者にだけ分かるよう

な書き方・色にする。そして、ボタン同士で混乱するような物は使用しない。
　数字類はアラビア数字を使い見やすい大きさにする。入力で使う数字と他で使う数字は場所を離す・色を変える等の処置をする。
　入力後、比較をして結果を別のウインドウに出す様にする。認証時には、数秒の後、閉じる。否の時、画面は残す。

5. 実装画面及び実行画面

5.1 画面



・この様に画面はシンプルにし、見やすく・小さく、使いやすさを重要視した。
上記でもあるようにシンプルの観点からCheckbox・Button・Label・Text Field の必要最低限にした。
・画面中央の Label1 及び Label2 に表示する文字は Font：Century, BOLD の 17POINT で表示。
・可否と書かれた Label に結果を表示。

5.2 実行・結果

このワンタイムパスワードは、チャレンジレスポンス方式と取り入れている。従来の物は設定していた一つのやり方でしか出来ない。しかし、本研究では従来の様に一つだけではなく計算の仕方を変え、ボタン一つで認証を変える方式を作った。なぜ複数なのかは前項でも示した通り、他人に関連性・規則性を見つけにくくする為である。相手に盗まれるのを従来の物より遅くする等の効果を期待している。
　さて、本研究では方式を三種類と考えている。これは、認証する時にこれ以上多くしてしまうと覚えているのが困難になってしまう可能性がある為、これ以上あった場合同じ所しか使わなくなってしまう恐れが大きくなってしまう。
人によってはボタンが三種類でも多いと感じてしまうかも知れない。これ以上減らしてしまうと従来の物との大差がなくなってしまう。この事を加味し、三種類とした。

5.2.1 Button 1・2・3 の内容

まず認証の流れを記す。
(1)Checkbox にチェックを押す。するとランダムでボタン一つに色が付く。(使用色 赤・黄・青の三色) (2)Button に色が付いた時、認証画面中央部の Label 2 つにランダムで数字を 5 桁表示させる。(3) 色の付いたボタンに対応するチャレンジの結果をテキストボックスに入れ、色の付いた Button を押す。(4) 結果を可否と書かれた Label に表示。尚、比較後、入力した数値は消去する事とする。

・Button1 の仕様 ---
表示した数字を 1 と 0 にし(奇数は 1,偶数は 0)、その 2 つを加算する。例えば 5.1 で表示されたのを見てみると 19870 と 17622 である。計算すると 11010 と 11000 になる。(使用者は自分で計算する必要あり。) 2 つの結果を加算、Textfield1 に入力、button1 を押すと判定に入る。結果は可否部分に表示する。

・Button2 の仕様---
button2 では単純に加算としている。5.1 で表示されたのを見てみると 19870 と 17622 である。この 2 つの数字を加算する。結果は 37492 となる。この結果を Textfield1 に入力し、button2 を押すと判定に入る。結果は可否部分に表示する。

・Button3 の仕様---
button3 では Label1 から Label2 の下 2 桁を減算する。5.1 で表示されたのをみてみると 19870 と 17622 であるからまず下 2 桁を見ると 70 と 22 の 2 つである事が分かる。Button3 では減算が仕様なので減算すると 48 になる。48 を入力し button3 を押すと判定に入り結果は可否部分に表示する。

6.所感

改めて重要性を学んだ。不注意から他意まで流出の可能性は多いに考えられる。そして、英数字・記号等を使い長くしても絶対的の安全は無いと考える。しかし、だからこそ、注意が必要であり、定期的な変更が重要だとの教訓を学んだ。

参考文献

1) IPA: ユーザ認証システム, http://www.ipa.go.jp/security/awareness/administrator/remote/capter6/5.html