December 12, 2016

Method Of Password Entering And Storing

Dimitri Kanevsky

Marcel Yung

METHOD OF PASSWORD ENTERING AND STORING

ABSTRACT

A system and method are disclosed in which a user adds random characters or produces random gestures while s/he enters a password/pin. The password/pin input string becomes a combination of substrings--some of which are part of the valid password string and some of which consist of random characters or gestures. The correct password characters are then easily detected by a security password extracting system that matches characters in the correct password string to those that appear in the garbled string. The method disclosed can be performed with touchscreen gesture passwords, security questions, voice passwords or similar security lock. The method is easy to use and increases protection for passwords without adding to mental workload of the user, while making it difficult for an observer to hack passwords. It also allows entry of passwords without having to re-enter if mistakes are made.

BACKGROUND

When people enter a password or pin in a public place, these passwords can be seen by a stranger, sometimes called ‘shoulder phishing’. This problem is especially acute when people enter passwords via gestures on a touch screen in mobile devices or pins in an ATM machine since people tend to choose relatively simple gesture passwords or pins and their activities happen often in crowded places. Another problem is that when people enter passwords, if they mistype some characters or make a wrong gesture turn, they need to go back and start the password/pin entry process again. This problem is especially acute for people who have tremors or if they require a secured access in some 'shaking' environment (e.g. while they are walking or driving). There is always the security risk that hackers manage to get passwords/pins in communication channels and reuse them by copying password utterances and re-sending these copies to get illegal entry to secured assets. To address the above problems, users are usually required to use long or complicated passwords. This problem is especially acute if users need to enter passwords many times per day (for example to access their mobile phones).

DESCRIPTION

A system and method are disclosed in which a user adds random characters or produces random gestures while s/he enters a password/pin. The password/pin input string thus becomes a combination of substrings--some of which are part of the valid password string and some of which are random characters or gestures. The security system that validates password entry strings checks whether the original password string could be extracted from the garbled input (i.e. a mixture of correct and random strings of characters/gestures). For example a password is 1AB2CD. And a user enters a password input as: SDG1ANMB3D2C%D, in which the string 1AB2CD is split into components by some random substrings. The correct password characters are then easily detected by a security password extracting system that matches characters in the correct password string to those that appear in the correct sequence in the garbled string. This process can be shown in this example in detail as the following: The security extract gets the first character from the password 1AB2CD. It is 1. It scans the garbled utterance SDG1ANMB3D2C%D until it finds 1. After that it gets the second character from the password: A. Then it gets a next entry from the password: B and continues to scan the garbled utterance until it finds the entry B. This process continues until the security system extracts the full password string (in this case the access is approved). Alternatively, it scans the whole garbled string until the end without recovering the full password - in which case the system does not allow access to a user who entered this garbled string.

Similarly, the method disclosed can be performed with gesture passwords in the following way. In usual mobile touch screen panel numbers are not displayed and points or symbols are displayed, but for convenience we use numbers here. As illustrated in FIG. 1, the user passes a finger through 1->2->3->6->8 on a touch panel for the correct unlock pattern. The user then can add random gestures and make a sequence as for example: 1->4->5->2->3->6->5->7->8 Then, the correct password string of gestures can be easily extracted from this garbled sequence. And it is very easy for a user to produce a random garbled utterance that contains all pieces of correct password utterance gesture elements by moving from some point to point randomly but always returning back to the next correct entry in the password. For instance, in the above example the user moved randomly from the correct entry 2 to 5 but then s/he returns back to the next entry 3. And similarly, s/he jumps randomly from the entry 6 to 5 and then returns to the next correct entry 8 via 7.

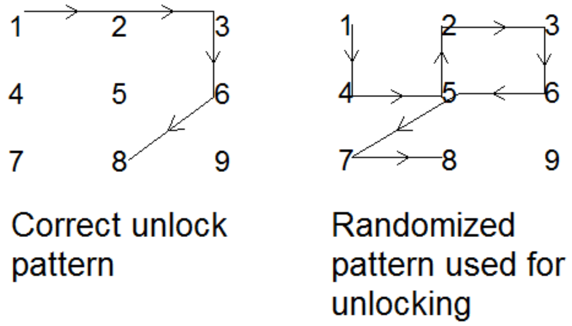

FIG. 1: Method for randomizing unlock patterns on a touchscreen

This concept could be extended for systems that require answers to security questions. Users who answer questions could insert a number of random wrong answers in addition to correct answers. For example for the question “where were you born?” the user can answer: Moscow Kiev Princeton (and the correct city is Kiev). There could be variants of this concept for voice entries. Users could add random sounds or phrases when they are asked for voice entry passwords. In another variation of the method, entry of the randomized voice password could be coupled with colors displayed on the mobile device. The user gives a correct answer when s/he sees an indicator color (e.g. red) but gives a wrong answer when seeing other colors (e.g. blue or green). In this situation, even someone overhearing a voice answer may have no access to what color is being displayed on the user mobile device screen and cannot obtain the password. In order to prevent an observer from copying a garbled password string, the method can add rejection of a password string with random characters, if the string is identical or too close to previously used password entry strings. There can also be a variant of the method in which a user can enter for each password element exactly k elements, one of which must be correct and the others random. If the size of the random string is *m* and the length of the password is *n* then the likelihood of guessing the password becomes $(k/m)^n$. If *m* is much bigger than *k* then the probability is small for large *n*.

The method is easy to use and increases protection for passwords without adding to mental workload of the user. It allows entry of passwords without going back and re-entering if mistakes are made. It makes it difficult for an observer to guess a password (shoulder phishing). It also makes breaking passwords more difficult since the length of a password increases.