February 24, 2017

PASSWORD CHECKING SYSTEM

Omri Amarilio

Albert Bodenhamer

PASSWORD CHECKING SYSTEM

ABSTRACT

A password checking system can be used to log into a device or service by entering a password in the form of a character sequence on a first keyboard layout. In case a user enters a password that does not match the correct password, the system retrieves a list of alternate keyboard layouts. Then the system uses the user's touch location pattern from the password entry on one of the alternate keyboard layouts in the list, and the system converts the touch location pattern to another character sequence. The system then checks whether the converted character sequence matches the correct password. Then, if the converted character sequence matches the correct password, the system unlocks the device or service. Thus, if a user attempts to unlock a device or service fails due to incorrect password entry, the system tries the converted character codes based on other keyboard layouts and checks for a password match.

PROBLEM STATEMENT

The use of passwords for logging into a device or a service is generally known in the art. A user usually inputs a password using a keyboard by typing a character sequence for the password. Furthermore, there are a wide variety of keyboard layouts available that can be used for inputting the password into a device. A user may set the password in one keyboard layout but may have to press/tap different keystroke patterns on another keyboard layout for inputting the correct password in order to log into a service or the device. For example, the password was set on a keyboard with a different language and with a multilingual use case a different language keyboard is presented to input the password. This implies that the password is used only in the

final form, i.e., the character sequence which was initially set by the user. Generally, a user develops a muscle memory for typing passwords on one form of keyboard layout which was used to set the password. In many cases, the user may inadvertently input a wrong password on a different kind of keyboard layout by tapping in the keystrokes pattern as per the keyboard that was used to set the password. Because there is no way to know if the user tapped the right keystroke pattern on the keyboard but was just using the wrong keyboard layout, the device will interpret the input as a wrong password entry. This should lead to authentication failure and may cause unnecessary inconvenience to the user who then has to convert to the correct keyboard layout and then re-enter a password.

DETAILED DESCRIPTION

The systems and techniques described in this disclosure relate to a password checking system that checks the correctness of a password based on a tapping pattern of the password on various types of keyboard layouts. The system can be implemented for use in an Internet, an intranet, or another client and server environment. The system can be implemented locally on a client device or implemented across a client device and server environment. The client device can be any electronic device such as a mobile device, a smartphone, a tablet, a handheld electronic device, a wearable device, a laptop, a desktop computer, etc. The client device can further include a password-enabled locking apparatus having a virtual/hardware keyboard to input the password. The system can further be implemented for use in a password protected gateway implemented on a server or locally on a client electronic device. The system can be implemented to any computing environment where a password is required.

Fig. 1 illustrates an example method 100 to determine the correctness of a password based on the tapping pattern of the password on different types of keyboard layouts. The method 100 starts with setting a password. The system receives (102) a first character sequence. The first character sequence can be any alphanumeric character sequence which a user wants to set as a password. For example, the first character sequence may contain a combination of letters (A-Z, a-z), numbers (0-9), and special characters (@, *, _, etc.). The system may receive the first character sequence through a keyboard. The keyboard can be an on-screen virtual QWERTY keyboard of a smartphone or an externally attached hardware keyboard. The types of keyboards may further include, but are not limited to, AZERTY type, Dvorak, ADB keyboard, KALQ for touch screen devices, T9 keypad, calculator keypad, a multilingual keypad, and telephone keypad.

The system receives (104) a second character sequence through the same keyboard. The second sequence can be an alphanumeric character sequence. The system then determines (106) whether the first and the second character sequences match. If the first and the second character sequence match with each other, the system stores (108) a non-reversible “secret” corresponding to the matching character sequence to be identified as the password. If the first and the second character sequences do not match, the system starts again with receiving (102) a first character sequence and receiving (104) a second character sequence until the first and second character sequence match.

After the password storage phase is complete, the system receives (110) a third character sequence. The system may receive a third sequence as an attempt by a user to unlock a device or a service/application. The user may input the third character sequence as an alphanumeric

character sequence using any of various implementation options such as an on-screen virtual keyboard of a smartphone or an externally attached hardware keyboard as described previously with respect to steps 102, 104. The system then proceeds to determine (112) whether the third character sequence matches the password. If the third character sequence matches the password, the system unlocks (114) the device or service/application protected by the corresponding correct password.

If the third character sequence does not match the password, the system identifies (116) the user's keyboard touch-location pattern. The system may identify the tapping pattern of the keystrokes that the user created while entering the third character sequence on the keyboard. The system then retrieves (118) an ordered list of alternate keyboard layouts. The alternate keyboard layouts may include various Latin keyboard layouts, e.g., QWERTY layout, AZERTY layout, or T9 layout, etc. The system sets (120) a pointer to a (first) alternate keyboard layout in the retrieved ordered list of alternate keyboard layouts. The system then converts (122) the user's keyboard touch-location pattern to another character sequence based on the pointed-to first alternate keyboard layout. Thus, in this step, the system determines how the third character sequence entered by the user would look like if that pattern were entered on the pointed-to first alternate keyboard layout.

The system then determines (124) whether the resulting character sequence on the first alternate keyboard matches the correct password. If yes, the system unlocks (114) the intended device or service/application which the user was attempting to access by entering the third character sequence. If the resulting character sequence on the first alternate keyboard layout does not match the correct password, the system advances (126) the pointer to the next alternate

keyboard layout and proceeds to set the pointer to the second (next) alternate keyboard layout in the ordered list of alternate keyboard layouts. The system then repeats the steps of converting (122) the user's keyboard touch-location pattern to another character sequence based on the pointed-to alternate keyboard layout and determining (124) whether the resulting sequence from the pointed-to alternate keyboard layout matches the correct password until all the keyboard layouts in the retrieved list of keyboard layouts have been exhausted, or the correct sequence criteria has been met with one of the keyboard layouts in the list.

In an alternate embodiment, the system simultaneously determines and adds each character of all layouts in a different input field as the user types the third character sequence for inputting the password. The resulting sequences in different input fields are matched for correctness of the entered password with the original password. In this way, the process of matching the entered password with the original password, keeping in view the possible representations of the original password on various alternate keyboard layouts, may save the user an inconvenience of re-entering the password based on the current keyboard layout or switching the keyboard layout and then re-entering the password. Further, if the system runs out of various keyboard layouts (126), an error message is displayed to the user for prompting a wrong password input. The system allows limited number of attempts for determining (124) that the entered password is correct in order to keep the security guarantee of a password. This stops a random user from brute-forcing the password through an excessively large amount of keyboard layouts.

Fig. 2 is a block diagram of an exemplary environment that shows components of a system for implementing the techniques described in this disclosure. The environment includes

client devices 210, servers 230, and network 240. Network 240 connects client devices 210 to servers 230. Client device 210 is an electronic device. Client device 210 may be capable of requesting and receiving data/communications over network 240. Example client devices 210 are personal computers (e.g., laptops), mobile communication devices, (e.g. smartphones, tablet computing devices), set-top boxes, game-consoles, embedded systems. The other devices 210' that can send and receive data/communications over network 240 may include a password-enabled locking apparatus having a virtual/hardware keyboard for inputting the password, etc. Client device 210 may execute a password protected application, such as a web browser 212 or 214 or a native application 216. Web applications 213 and 215 may be displayed via a web browser 212 or 214. Server 230 may be a web server capable of sending, receiving and storing web pages 232. Web page(s) 232 may be stored on or accessible via server 230. Web page(s) 232 may be associated with web application 213 or 215 and accessed using a web browser, e.g., 212. When accessed, webpage(s) 232 may be transmitted and displayed on a client device, e.g., 210. Resources 218 and 218' are resources available to the client device 210 and/or applications thereon, or server(s) 230 and/or web pages(s) accessible therefrom, respectively. Resources 218' may be, for example, memory or storage resources; a text, image, video, audio, JavaScript, CSS, or other file or object; or other relevant resources. Network 240 may be any network or combination of networks that can carry data communication.

DRAWINGS

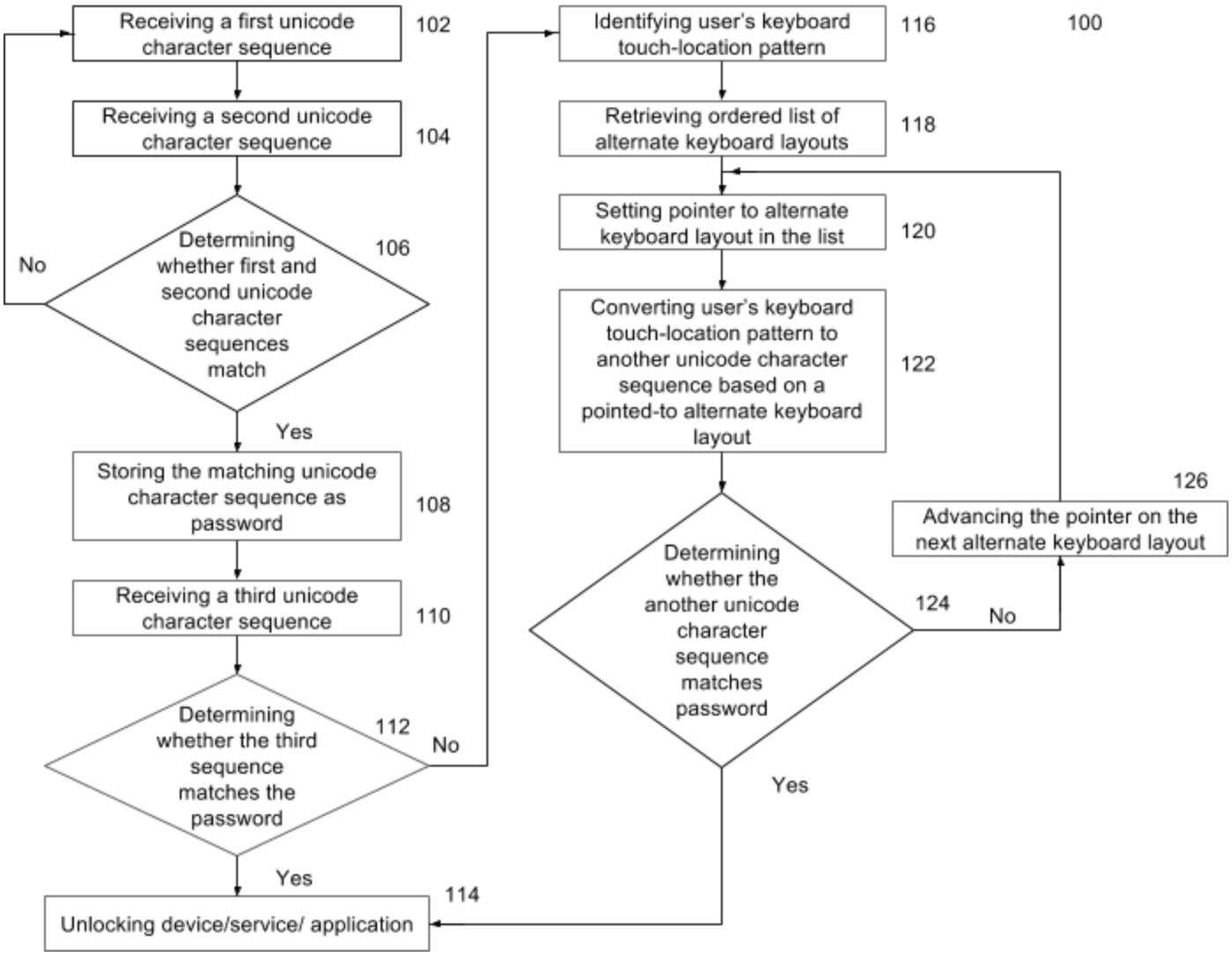

Fig. 1

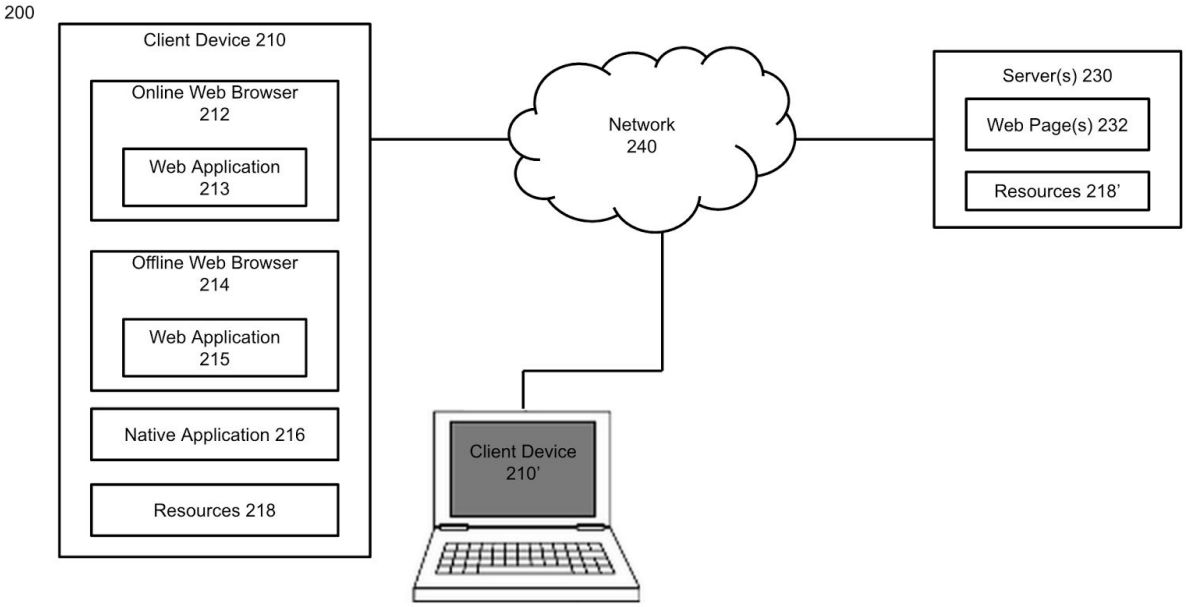
200
Client Device 210
Online Web Browser 212
Web Application 213
Offline Web Browser 214
Web Application 215
Native Application 216
Resources 218
Network 240
Server(s) 230
Web Page(s) 232
Resources 218'
Client Device 210'

Fig. 2