

COMPUTER-IMPLEMENTED METHODS AND NON-TRANSITORY COMPUTER-READABLE MEMORIES STORING COMPUTER PROGRAM INSTRUCTIONS FOR PERFORMING SECONDARY AUTHENTICATION

Field of the Invention

5 The present invention relates to the field of two-factor authentication (2FA) and more particularly to computer-implemented methods and non-transitory computer-readable memories for performing secondary authentication.

Background of the Invention

10 Phishing attacks primarily employ a man-in-the-middle mechanism to trick a web application user into accessing a fake or spoof site hosted by bad actors and entering personal authentication credentials. The credentials are then used by the bad actors to gain access to an actual web application on a computing device of the bad actors. The bad actors then trigger a two-factor authentication (2FA) mechanism to gain elevated access. The web application then prompts the actual user to authorize access often with
15 either a Short Messaging Service (SMS) personal identification number (PIN) or an email PIN. On receiving the PIN, the actual user then enters the PIN into the fake web page, thus handing over the PIN to the bad actors, who then enter the PIN into the 2FA authentication page of the web application, allowing the bad actors to gain full access to the web application.

20 Accordingly, it would be desirable to provide computer-implemented methods and non-transitory computer-readable memories for performing secondary authentication that help prevent unauthorised access due to phishing attacks.

Summary of the Invention

25 Accordingly, in a first aspect, the present invention provides a computer-implemented method for performing secondary authentication. The method includes executing on one or more processors of one or more servers the steps of: detecting an operating environment of an authentication page after primary authentication is completed; performing a handshake operation with a pre-authenticated mobile device if a first determination is made that the detected operating environment of the authentication page

is not a mobile environment or if a second determination is made that the secondary authentication is not being performed on the pre-authenticated mobile device; receiving a series of encoded quick response (QR) codes from the pre-authenticated mobile device; decoding the encoded QR codes with the authentication page; displaying the decoded QR codes in the authentication page; and receiving a secondary authentication decision from the pre-authenticated mobile device.

In a second aspect, the present invention provides a computer-implemented method for performing secondary authentication. The method includes executing on one or more processors of a pre-authenticated mobile device the steps of: performing a handshake operation with one or more servers if a first determination is made that a detected operating environment of an authentication page is not a mobile environment or if a second determination is made that the secondary authentication is not being performed on the pre-authenticated mobile device; generating a series of quick response (QR) codes; encoding the QR codes; transmitting the series of encoded QR codes to the one or more servers; scanning the decoded QR codes displayed by the one or more servers; and generating a secondary authentication decision based on the scanned QR codes.

In a third aspect, the present invention provides a non-transitory computer-readable memory storing computer program instructions executable by one or more processors of one or more servers to perform operations for secondary authentication. The operations include: detecting an operating environment of an authentication page after primary authentication is completed; performing a handshake operation with a pre-authenticated mobile device if a first determination is made that the detected operating environment of the authentication page is not a mobile environment or if a second determination is made that the secondary authentication is not being performed on the pre-authenticated mobile device; receiving a series of encoded QR codes from the pre-authenticated mobile device; decoding the encoded QR codes with the authentication page; displaying the decoded QR codes in the authentication page; and receiving a secondary authentication decision from the pre-authenticated mobile device.

In a fourth aspect, the present invention provides a non-transitory computer-readable memory storing computer program instructions executable by one or more processors of a pre-authenticated mobile device to perform operations for secondary authentication. The operations include: performing a handshake operation with one or

more servers if a first determination is made that a detected operating environment of an authentication page is not a mobile environment or if a second determination is made that the secondary authentication is not being performed on the pre-authenticated mobile device; generating a series of QR codes; encoding the QR codes; transmitting the series
5 of encoded QR codes to the one or more servers; scanning the decoded QR codes displayed by the one or more servers; and generating a secondary authentication decision based on the scanned QR codes.

Other aspects and advantages of the invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings,
10 illustrating by way of example the principles of the invention.

Brief Description of the Drawings

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

FIG. 1 is a schematic block diagram illustrating a system architecture for
15 implementing a secondary authentication method in accordance with an embodiment of the present invention;

FIG. 2 is a schematic sequence diagram illustrating computer-implemented methods for performing secondary authentication in accordance with embodiments of the present invention; and

20 FIG. 3 is a schematic block diagram illustrating a computer system for implementing the secondary authentication method disclosed herein.

Detailed Description of Exemplary Embodiments

The detailed description set forth below in connection with the appended drawings is intended as a description of presently preferred embodiments of the invention, and is
25 not intended to represent the only forms in which the present invention may be practiced. It is to be understood that the same or equivalent functions may be accomplished by different embodiments that are intended to be encompassed within the scope of the invention.

Referring now to FIG. 1, a system architecture 10 for implementing secondary authentication methods is shown. The system architecture 10 includes one or more computing devices 12 including a pre-authenticated mobile device and one or more servers 14.

5 The one or more computing devices 12 may be one or a combination of a desktop computer, a laptop, a tablet and a smartphone. The pre-authenticated mobile device may be a laptop, a tablet or a smartphone.

 A login mechanism of a web application may be implemented via a compiled software component such as, for example, a web assembly or a Java applet supported on
10 the one or more computing devices 12. The web assembly runs securely within a web browser such as, for example, Microsoft Edge, Google Chrome, Apple Safari, Mozilla Firefox, etc.

 A mobile application is pre-installed in the pre-authenticated mobile device. When the mobile application is initially installed on the mobile device 12, verification of a user
15 account is performed, thereby pre-authenticating the mobile device 12. This may be by using a secure mechanism to verify the legitimacy of a user such as, for example, by using a combination of username/password together with either a Short Messaging Service (SMS) personal identification number (PIN) or an email PIN. The mobile application may be incorporated into an existing mobile application used by any system for the second
20 factor authentication.

 The one or more servers 14 provide server-side services for communication between an authentication page running on one of the computing devices 12 and the mobile application of the pre-authenticated mobile device.

 Having described the system architecture 10 for implementing the secondary
25 authentication methods, computer-implemented methods for performing secondary authentication will now be described below with reference to FIG. 2.

 Referring now to FIG. 2, computer-implemented methods 50 for performing secondary authentication are shown. The methods 50 for performing secondary authentication may be executed on one or more processors of the pre-authenticated
30 mobile device 12 and one or more processors of the one or more servers 14.

The secondary authentication method 50 begins at step 52 by detecting an operating environment of an authentication page after primary authentication 54 is completed. Using one of the computing devices 12 as a working device, a user may sign in to access a web application using login credentials such as, for example, a username and a password via an authentication page. After primary authentication 54 is successfully accomplished, secondary authentication may be initialised 56 by establishing a secure communication with the one or more servers 14 providing server-side services of the web application. The user may then be presented with the second-factor authentication page.

At step 58, a handshake operation is performed with the one or more servers 14 and the pre-authenticated mobile device 12 if a first determination is made that the detected operating environment of the authentication page is not a mobile environment or if a second determination is made that the secondary authentication is not being performed on the pre-authenticated mobile device 12.

If the first determination is that the authentication page is not running on a mobile platform, the secondary authentication method 50 proceeds to a verification phase. The authentication page may display a quick response (QR) code, for example, a “Beacon Ready” QR code, and prompt the user to manually start a mobile application on the pre-authenticated mobile device 12 to scan the QR code to commence the verification phase. The mobile application was previously installed on the mobile device 12 with proper authorization and activation. The verification phase commences with the mobile application performing a series of handshaking communication with the authentication page.

If the first determination is that the authentication page is running on a mobile platform, a link may be provided at step 60 to activate the mobile application on the pre-authenticated mobile device 12. If the operating environment is a mobile environment such as, for example, iOS or Android, the authentication page may provide a clickable link for the user to click and thereby activate the mobile application via the application link.

The second determination may be performed when the link is activated. When the mobile application is started via the application link, a verification may be made that the authentication page and the mobile application are both operating on the same mobile

device if the working device and the pre-authenticated mobile device 12 have the same IP address.

An option may be provided for user approval if the second determination is that the secondary authentication is being performed on the pre-authenticated mobile device 12.

- 5 More particularly, if verified that both the authentication page and the mobile application are operating on the same mobile device, the mobile application may display a prompt for the user to approve the two-factor authentication (2FA) should the user choose to.

If the second determination is that the secondary authentication is not being performed on the pre-authenticated mobile device 12, the secondary authentication

- 10 method 50 proceeds to the verification phase.

Upon establishing the handshake, a series of QR codes is generated by the pre-authenticated mobile device 12 at step 62. The number of QR codes in the series may be chosen based on a balance of time taken to complete the secondary authentication as opposed to the effectiveness of the secondary authentication method 50. Accordingly, the

- 15 series of QR codes may consist of between about 10 and about 20 QR codes. A minimum of 10 QR codes may cause a minimum delay of 1 second for an attacker in a man-in-the-middle attack to capture all 10 QR codes. If the attacker incurs a 1 second delay in capturing all 10 QR codes, the attacker would not have sufficient time to spoof the QR codes onto a spoofed authentication page for the user to scan as there would be a 1
20 second response time stipulated by the pre-authenticated mobile device 12 for commencement of QR code display. The series of QR codes may be randomly generated for every authentication attempt and never repeated. Each of the QR codes in the series may be a dynamic QR code that is not hardcoded or repeated.

- At step 64, the generated QR codes are encoded by the pre-authenticated mobile
25 device 12. Each of the QR codes in the series may be specially encoded and/or generated in a scrambled format.

The series of encoded QR codes are then transmitted to the one or more servers at step 66.

The authentication page then receives the QR codes and simultaneously proceeds to notify the mobile application that the working device 12 is ready to commence displaying the QR codes.

After the series of encoded QR codes is received from the pre-authenticated mobile device 12 by the one or more servers 14 at step 66, the encoded QR codes are decoded with the authentication page running on the working device 12 at step 68.

At step 70, the decoded QR codes are displayed in the authentication page running on the working device 12. The decoded QR codes may be displayed at a predetermined frame rate. The predetermined frame rate limit may be arrived at by considering practical limits like varying performance level of the processor and camera module of the pre-authenticated mobile device 12. Too high a frame rate would require the pre-authenticated mobile device 12 to be higher performing and thus, older mobile devices may not be supported. Accordingly, the predetermined frame rate at which the series of decoded QR codes is displayed may be between about 11 and about 20 QR codes per second. The authentication page may decode the QR codes and immediately commence display of the QR codes at a rapid frame rate of 100 milliseconds (ms) between QR codes. The decoded QR codes may be repetitively displayed in loop. For example, the QR codes may be displayed in an order as dictated by the mobile application and the display may continue to run in a cycling loop from a first QR code to a last QR code and then starting from the first QR code to the last QR code again. The authentication page may run this loop undisturbed for 10 rounds or until terminated by the mobile application.

Upon being notified by the authentication page that the QR codes are ready to be displayed, the decoded QR codes displayed by the one or more servers are scanned at step 72. This may be by the mobile application immediately activating an onboard camera to scan the QR codes.

At step 74, a secondary authentication decision is generated based on the scanned QR codes. The step 74 of generating the secondary authentication decision based on the scanned QR codes may include determining if a commencement time of a display of the decoded QR codes is within a predetermined time limit. The mobile application may expect the series of QR codes to commence display no more than the predetermined time limit after sending the encoded QR codes to the server-side service. This limit may be

determined based on practical considerations of different network performances experienced by users of the web application. The time limit should not be too long to remove the possibility of any men-in-the-middle attack. Such attacks typically introduce latency into the secondary authentication process, which is mitigated by having this time
5 limit. The predetermined time limit of the commencement time of the display of the decoded QR codes may be between about 1 second (s) and about 2 s from transmission of the series of encoded QR codes to the one or more servers. The mobile application may mandate that the first QR code must be detected within 1 second of commencement of scanning. If the commencement time is determined to be within the predetermined time
10 limit, the series of decoded QR codes is captured at step 72. When the mobile application captures the commencement of the QR codes display within the predetermined time limit, the mobile application continues to rapidly capture all of the decoded QR codes.

The step 74 of generating the secondary authentication decision based on the scanned QR codes may include determining if the decoded QR codes are correctly
15 captured. This may be by determining if each of the decoded QR codes is a correct QR code and/or determining if the decoded QR codes are received in a correct order. The mobile application may check that the correct QR codes are displayed at a specified frame rate and/or that all the QR codes are received in the correct order.

The secondary authentication decision is received from the pre-authenticated
20 mobile device 12 at step 76. This decision may be to grant access if a determination is made that the series of decoded QR codes is correctly captured. Once all the QR codes are correctly captured, the mobile application may trigger a successful second factor authentication decision to the server-side service to grant the user full access to the web application.

25 If the mobile application fails to detect the commencement of display of the QR codes within the predetermined time limit or any of the QR codes fail to be displayed at the required frame rate or are displayed in an incorrect order or with a wrong code, a failed authentication results. This may trigger a re-try. The re-try may involve a brand-new set of QR codes to be transmitted to the web assembly with the same response time
30 requirement. Accordingly, the series of QR codes may be regenerated for encoding and retransmission if the commencement time of the display of the decoded QR codes exceeds the predetermined time limit or if the decoded QR codes are incorrectly captured.

Access may be prevented after a predetermined number of regeneration attempts. The predetermined number of regeneration attempts may be 3 to prevent the second-factor authentication (2FA) process from taking too long and giving a negative user experience. If subsequent re-tries continue to fail, the mobile application may disallow the
5 second factor authentication from succeeding, thus preventing full access to the web application.

Advantageously, the tight time limit between phases of the second factor authentication method 50 coupled with the high frame rate display of the QR codes makes it impossible for a bad actor to run the second factor authentication method 50 on an
10 external computing device as the bad actor would have no means of sending down the QR codes to a spoofed page for the user to scan. The high frame rate makes it practically impossible for the bad actor to simply stream a video of the QR codes from the external computing device to the spoofed page for the user to scan.

The method 50 for performing secondary authentication thus prevents bad actors
15 from triggering second factor authentication on an external computing device by forcing the second factor authentication to work only on the computing device of an actual user accessing an actual web application. Bad actors who manage to steal user credentials would only be able to gain initial access to the web application, but would not be able to be successfully authorized at the secondary authentication stage, thus preventing
20 unauthorised privileged access.

Computer program instructions executable by one or more computer processors to perform operations for secondary authentication described above may be stored in non-transitory computer-readable memory.

Referring now to FIG. 3, a computer system 100 suitable for implementing the
25 method 50 for performing secondary authentication is shown. The computer system 100 includes one or more processors 102 (which may be referred to as a central processor unit or CPU) that is in communication with memory devices including secondary storage 104, read only memory (ROM) 106, random access memory (RAM) 108, input/output (I/O) devices 110, and network connectivity devices 112. The one or more processors 102 may
30 be implemented as one or more CPU chips.

It is understood that by programming and/or loading executable instructions onto the computer system 100, at least one of the CPU 102, the RAM 108, and the ROM 106 are changed, transforming the computer system 100 in part into a particular machine or apparatus having the novel functionality taught by the present disclosure. It is fundamental to the electrical engineering and software engineering arts that functionality that can be implemented by loading executable software into a computer can be converted to a hardware implementation by well-known design rules. Decisions between implementing a concept in software versus hardware typically hinge on considerations of stability of the design and numbers of units to be produced rather than any issues involved in translating from the software domain to the hardware domain. Generally, a design that is still subject to frequent change may be preferred to be implemented in software because re-spinning a hardware implementation is more expensive than re-spinning a software design. Generally, a design that is stable that will be produced in large volume may be preferred to be implemented in hardware, for example in an application specific integrated circuit (ASIC), because for large production runs the hardware implementation may be less expensive than the software implementation. Often a design may be developed and tested in a software form and later transformed, by well-known design rules, to an equivalent hardware implementation in an application specific integrated circuit that hardwires the instructions of the software. In the same manner as a machine controlled by a new ASIC is a particular machine or apparatus, likewise a computer that has been programmed and/or loaded with executable instructions may be viewed as a particular machine or apparatus.

Additionally, after the system 100 is turned on or booted, the CPU 102 may execute a computer program or application. For example, the CPU 102 may execute software or firmware stored in the ROM 106 or stored in the RAM 108. In some cases, on boot and/or when the application is initiated, the CPU 102 may copy the application or portions of the application from the secondary storage 104 to the RAM 108 or to memory space within the CPU 102 itself, and the CPU 102 may then execute instructions that the application is comprised of. In some cases, the CPU 102 may copy the application or portions of the application from memory accessed via the network connectivity devices 112 or via the I/O devices 110 to the RAM 108 or to memory space within the CPU 102, and the CPU 102 may then execute instructions that the application is comprised of. During execution, an application may load instructions into the CPU 102, for example load some of the

instructions of the application into a cache of the CPU 102. In some contexts, an application that is executed may be said to configure the CPU 102 to do something, for example, to configure the CPU 102 to perform the function or functions promoted by the subject application. When the CPU 102 is configured in this way by the application, the
5 CPU 102 becomes a specific purpose computer or a specific purpose machine.

The secondary storage 104 is typically comprised of one or more disk drives or tape drives and is used for non-volatile storage of data and as an over-flow data storage device if RAM 108 is not large enough to hold all working data. Secondary storage 104 may be used to store programs which are loaded into RAM 108 when such programs are
10 selected for execution. The ROM 106 is used to store instructions and perhaps data which are read during program execution. ROM 106 is a non-volatile memory device which typically has a small memory capacity relative to the larger memory capacity of secondary storage 104. The RAM 108 is used to store volatile data and perhaps to store instructions. Access to both ROM 106 and RAM 108 is typically faster than to secondary storage 104.
15 The secondary storage 104, the RAM 108, and/or the ROM 106 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media.

I/O devices 110 may include cameras, printers, video monitors, liquid crystal displays (LCDs), plasma displays, touch screen displays, keyboards, keypads, switches,
20 dials, mice, track balls, voice recognizers, card readers, paper tape readers, or other well-known input devices. The network connectivity devices 112 may take the form of modems, modem banks, Ethernet cards, universal serial bus (USB) interface cards, serial interfaces, token ring cards, fibre distributed data interface (FDDI) cards, wireless local area network (WLAN) cards, radio transceiver cards that promote radio communications using protocols
25 such as code division multiple access (CDMA), global system for mobile communications (GSM), long-term evolution (LTE), worldwide interoperability for microwave access (WiMAX), near field communications (NFC), radio frequency identity (RFID), and/or other air interface protocol radio transceiver cards, and other well-known network devices. These network connectivity devices 112 may enable the processor 102 to communicate
30 with the Internet or one or more intranets. With such a network connection, it is contemplated that the one or more processors 102 might receive information from the network, or might output information to the network in the course of performing the above-

described method steps. Such information, which is often represented as a sequence of instructions to be executed using the one or more processor 102, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave. Such information, which may include data or instructions to be executed
5 using the one or more processors 102 for example, may be received from and outputted to the network, for example, in the form of a computer data baseband signal or signal embodied in a carrier wave. The baseband signal or signal embedded in the carrier wave, or other types of signals currently used or hereafter developed, may be generated according to several methods well-known to one skilled in the art. The baseband signal
10 and/or signal embedded in the carrier wave may be referred to in some contexts as a transitory signal.

The processor 102 executes instructions, codes, computer programs, scripts which it accesses from hard disk, floppy disk, optical disk (these various disk-based systems may all be considered secondary storage 104), flash drive, ROM 106, RAM 108, or the network
15 connectivity devices 112. While only one processor 102 is shown, multiple processors may be present. Thus, while instructions may be discussed as executed by a processor, the instructions may be executed simultaneously, serially, or otherwise executed by one or multiple processors. Instructions, codes, computer programs, scripts, and/or data that may be accessed from the secondary storage 104, for example, hard drives, floppy disks,
20 optical disks, and/or other device, the ROM 106, and/or the RAM 108 may be referred to in some contexts as non-transitory instructions and/or non-transitory information.

In an embodiment, the computer system 100 may comprise two or more computers in communication with each other that collaborate to perform a task. For example, but not by way of limitation, an application may be partitioned in such a way as to permit concurrent
25 and/or parallel processing of the instructions of the application. Alternatively, the data processed by the application may be partitioned in such a way as to permit concurrent and/or parallel processing of different portions of a data set by the two or more computers. In an embodiment, virtualization software may be employed by the computer system 100 to provide the functionality of a number of servers that is not directly bound to the number
30 of computers in the computer system 100. For example, virtualization software may provide twenty virtual servers on four physical computers. In an embodiment, the functionality disclosed above may be provided by executing the application and/or

applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. Cloud computing may be supported, at least in part, by virtualization software. A cloud computing environment may be established by an enterprise and/or may be hired
5 on an as-needed basis from a third-party provider. Some cloud computing environments may comprise cloud computing resources owned and operated by the enterprise as well as cloud computing resources hired and/or leased from a third-party provider.

In an embodiment, some or all of the functionality disclosed above may be provided as a computer program product. The computer program product may comprise one or
10 more computer readable storage medium having computer usable program code embodied therein to implement the functionality disclosed above. The computer program product may comprise data structures, executable instructions, and other computer usable program code. The computer program product may be embodied in removable computer storage media and/or non-removable computer storage media. The removable computer
15 readable storage medium may comprise, without limitation, a paper tape, a magnetic tape, magnetic disk, an optical disk, a solid-state memory chip, for example analog magnetic tape, compact disk read only memory (CD-ROM) disks, floppy disks, jump drives, digital cards, multimedia cards, and others. The computer program product may be suitable for loading, by the computer system 100, at least portions of the contents of the computer
20 program product to the secondary storage 104, to the ROM 106, to the RAM 108, and/or to other non-volatile memory and volatile memory of the computer system 100. The one or more processors 102 may process the executable instructions and/or data structures in part by directly accessing the computer program product, for example by reading from a CD-ROM disk inserted into a disk drive peripheral of the computer system 100.
25 Alternatively, the one or more processors 102 may process the executable instructions and/or data structures by remotely accessing the computer program product, for example by downloading the executable instructions and/or data structures from a remote server through the network connectivity devices 112. The computer program product may comprise instructions that promote the loading and/or copying of data, data structures,
30 files, and/or executable instructions to the secondary storage 104, to the ROM 106, to the RAM 108, and/or to other non-volatile memory and volatile memory of the computer system 100.

In some contexts, the secondary storage 104, the ROM 106, and the RAM 108 may be referred to as a non-transitory computer readable medium or a computer readable storage media. A dynamic RAM embodiment of the RAM 108, likewise, may be referred to as a non-transitory computer readable medium in that while the dynamic RAM receives electrical power and is operated in accordance with its design, for example during a period of time during which the computer system 100 is turned on and operational, the dynamic RAM stores information that is written to it. Similarly, the one or more processors 102 may comprise an internal RAM, an internal ROM, a cache memory, and/or other internal non-transitory storage blocks, sections, or components that may be referred to in some contexts as non-transitory computer readable media or computer readable storage media.

As is evident from the foregoing discussion, the present invention provides computer-implemented methods and non-transitory computer-readable memories for performing secondary authentication that help prevent unauthorised access due to phishing attacks. Because of the scanning requirement, a secondary authentication page must be run in the presence of a pre-authenticated user device for successful secondary authentication. Secondary authentication in accordance with the present invention would fail if performed using a machine of a bad actor as the rapid frame rate requirement ensures that secondary authentication must be physically performed alongside a pre-authenticated mobile device. Accordingly, the present invention makes it impossible for the bad actor to trigger second factor authentication from an external computing device and gain access from the external computing device. Advantageously, the present invention is able to prevent a man-in-the-middle phishing attack even when a user is tricked into clicking on a phishing link containing a spoofed web application. Further advantageously, the present invention requires no additional proprietary hardware.

While preferred embodiments of the invention have been illustrated and described, it will be clear that the invention is not limited to the described embodiments only. Numerous modifications, changes, variations, substitutions and equivalents will be apparent to those skilled in the art without departing from the scope of the invention as described in the claims.

Further, unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise", "comprising" and the like are to be construed in an

inclusive as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to".

CLAIMS

1. A computer-implemented method for performing secondary authentication, comprising executing on one or more processors of one or more servers the steps of:
 - 5 detecting an operating environment of an authentication page after primary authentication is completed;
 - performing a handshake operation with a pre-authenticated mobile device if a first determination is made that the detected operating environment of the authentication page is not a mobile environment or if a second determination is made that the secondary
 - 10 authentication is not being performed on the pre-authenticated mobile device;
 - receiving a series of encoded quick response (QR) codes from the pre-authenticated mobile device;
 - decoding the encoded QR codes with the authentication page;
 - displaying the decoded QR codes in the authentication page; and
 - 15 receiving a secondary authentication decision from the pre-authenticated mobile device.
2. The computer-implemented method for performing secondary authentication according to claim 1, further comprising:
 - 20 providing a link to activate a mobile application on the pre-authenticated mobile device if the first determination is that the authentication page is running on a mobile platform.
3. The computer-implemented method for performing secondary authentication according to claim 2, further comprising:
 - 25 performing the second determination when the link is activated.
4. The computer-implemented method for performing secondary authentication according to any one of claims 1 to 3, further comprising:
 - 30 providing an option for user approval if the second determination is that the secondary authentication is being performed on the pre-authenticated mobile device.

5. The computer-implemented method for performing secondary authentication according to any one of claims 1 to 4, wherein the decoded QR codes are displayed at a predetermined frame rate.
- 5 6. The computer-implemented method for performing secondary authentication according to claim 5, wherein the predetermined frame rate at which the series of decoded QR codes is displayed is between about 11 and about 20 QR codes per second.
7. The computer-implemented method for performing secondary authentication
10 according to any one of claims 1 to 6, wherein the decoded QR codes are repetitively displayed in loop.
8. A computer-implemented method for performing secondary authentication, comprising executing on one or more processors of a pre-authenticated mobile device the
15 steps of:
- performing a handshake operation with one or more servers if a first determination is made that a detected operating environment of an authentication page is not a mobile environment or if a second determination is made that the secondary authentication is not being performed on the pre-authenticated mobile device;
- 20 generating a series of quick response (QR) codes;
- encoding the QR codes;
- transmitting the series of encoded QR codes to the one or more servers;
- scanning the decoded QR codes displayed by the one or more servers; and
- generating a secondary authentication decision based on the scanned QR codes.
- 25
9. The computer-implemented method for performing secondary authentication according to claim 8, wherein the series of QR codes consists of between about 10 and about 20 QR codes.
- 30 10. The computer-implemented method for performing secondary authentication according to claim 8 or 9, wherein the series of QR codes is randomly generated.

11. The computer-implemented method for performing secondary authentication according to any one of claims 8 to 10, wherein each of the QR codes in the series is generated in a scrambled format.
- 5 12. The computer-implemented method for performing secondary authentication according to any one of claims 8 to 11, wherein the step of generating the secondary authentication decision based on the scanned QR codes comprises:
- determining if a commencement time of a display of the decoded QR codes is within a predetermined time limit.
- 10 13. The computer-implemented method for performing secondary authentication according to claim 12, wherein the predetermined time limit of the commencement time of the display of the decoded QR codes is between about 1 second (s) and about 2 s from transmission of the series of encoded QR codes to the one or more servers.
- 15 14. The computer-implemented method for performing secondary authentication according to claim 12 or 13, wherein the step of generating the secondary authentication decision based on the scanned QR codes further comprises:
- determining if the decoded QR codes are correctly captured.
- 20 15. The computer-implemented method for performing secondary authentication according to claim 14, wherein the step of determining if the decoded QR codes are correctly captured comprises:
- determining if each of the decoded QR codes is a correct QR code.
- 25 16. The computer-implemented method for performing secondary authentication according to claim 14 or 15, wherein the step of determining if the decoded QR codes are correctly captured further comprises:
- determining if the decoded QR codes are received in a correct order.
- 30 17. The computer-implemented method for performing secondary authentication according to any one of claims 12 to 16, further comprising:

regenerating the series of QR codes for encoding and retransmission if the commencement time of the display of the decoded QR codes exceeds the predetermined time limit or if the decoded QR codes are incorrectly captured.

5 18. The computer-implemented method for performing secondary authentication according to claim 17, further comprising:

preventing access after a predetermined number of regeneration attempts.

19. A non-transitory computer-readable memory storing computer program
10 instructions executable by one or more processors of one or more servers to perform operations for secondary authentication, the operations comprising:

detecting an operating environment of an authentication page after primary authentication is completed;

performing a handshake operation with a pre-authenticated mobile device if a first
15 determination is made that the detected operating environment of the authentication page is not a mobile environment or if a second determination is made that the secondary authentication is not being performed on the pre-authenticated mobile device;

receiving a series of encoded QR codes from the pre-authenticated mobile device;

decoding the encoded QR codes with the authentication page;

20 displaying the decoded QR codes in the authentication page; and

receiving a secondary authentication decision from the pre-authenticated mobile device.

20. The non-transitory computer-readable memory according to claim 19, further
25 comprising the operation of:

providing a link to activate a mobile application on the pre-authenticated mobile device if the first determination is that the authentication page is running on a mobile platform.

30 21. The non-transitory computer-readable memory according to claim 20, further comprising the operation of:

performing the second determination when the link is activated.

22. The non-transitory computer-readable memory according to any one of claims 19 to 21, further comprising the operation of:

providing an option for user approval if the second determination is that the secondary authentication is being performed on the pre-authenticated mobile device.

5

23. The non-transitory computer-readable memory according to any one of claims 19 to 22, wherein the decoded QR codes are displayed at a predetermined frame rate.

24. The non-transitory computer-readable memory according to claim 23, wherein the
10 predetermined frame rate at which the series of decoded QR codes is displayed is between about 11 and about 20 QR codes per second.

25. The non-transitory computer-readable memory according to any one of claims 19 to 24, wherein the decoded QR codes are repetitively displayed in loop.

15

26. A non-transitory computer-readable memory storing computer program instructions executable by one or more processors of a pre-authenticated mobile device to perform operations for secondary authentication, the operations comprising:

performing a handshake operation with one or more servers if a first determination
20 is made that a detected operating environment of an authentication page is not a mobile environment or if a second determination is made that the secondary authentication is not being performed on the pre-authenticated mobile device;

generating a series of QR codes;

encoding the QR codes;

25 transmitting the series of encoded QR codes to the one or more servers;

scanning the decoded QR codes displayed by the one or more servers; and

generating a secondary authentication decision based on the scanned QR codes.

27. The non-transitory computer-readable memory according to claim 26, wherein the
30 series of QR codes consists of between about 10 and about 20 QR codes.

28. The non-transitory computer-readable memory according to claim 26 or 27, wherein the series of QR codes is randomly generated.

29. The non-transitory computer-readable memory according to any one of claims 26 to 28, wherein each of the QR codes in the series is generated in a scrambled format.
30. The non-transitory computer-readable memory according to any one of claims 26 to 29, wherein the operation of generating the secondary authentication decision based on the scanned QR codes comprises:
- determining if a commencement time of a display of the decoded QR codes is within a predetermined time limit.
31. The non-transitory computer-readable memory according to claim 30, wherein the predetermined time limit of the commencement time of the display of the decoded QR codes is between about 1 second (s) and about 2 s from transmission of the series of encoded QR codes to the server.
32. The non-transitory computer-readable memory according to claim 30 or 31, wherein the operation of generating the secondary authentication decision based on the scanned QR codes further comprises:
- determining if the decoded QR codes are correctly captured.
33. The non-transitory computer-readable memory according to claim 32, wherein the operation of determining if the decoded QR codes are correctly captured comprises:
- determining if each of the decoded QR codes is a correct QR code.
34. The non-transitory computer-readable memory according to claim 32 or 33, wherein the operation of determining if the decoded QR codes are correctly captured further comprises:
- determining if the decoded QR codes are received in a correct order.
35. The non-transitory computer-readable memory according to any one of claims 30 to 34, further comprising the operation of:
- regenerating the series of QR codes for encoding and retransmission if the commencement time of the display of the decoded QR codes exceeds the predetermined time limit or if the decoded QR codes are incorrectly captured.

36. The non-transitory computer-readable memory according to claim 35, further comprising the operation of:

preventing access after a predetermined number of regeneration attempts.