# SYSTEMS AND METHODS FOR CONTACTLESS CARD COMMUNICATION AND MULTI-DEVICE KEY PAIR CRYPTOGRAPHIC AUTHENTICATION

## CROSS-REFERENCE TO RELATED APPLICATION

**[0001]** This application claims priority to U.S. Patent Application No. 17/352,181 filed June 18, 2021, the disclosure of which is incorporated herein by reference in its entirety.

## FIELD OF THE DISCLOSURE

**[0002]** The present disclosure relates to systems and methods for contactless card communication and, more specifically, for key pair cryptographic authentication of contactless cards using multiple devices.

## BACKGROUND

**[0003]** Card-based operations often involve the use of a card in communication with a server or other device. It is necessary to protect such communications from interception and unauthorized access. However, there are numerous sophisticated methods for data interception that may be employed by hackers or other fraudulent actors.

**[0004]** For example, transmission of data without encryption or other protection is susceptible to man-in-the-middle attacks, phishing attacks, replay attacks, and other attacks, and may be subject to other vulnerabilities, resulting in increased security risks and increased risks of account or card misuse. These risks may be further increased through the use of contactless cards, which communicate with other devices wirelessly, thus exposing data to potential interception during transmission.

**[0005]** Measures taken to address security risk, such as encryption, may consume system resources and hinder operational efficiency. For large numbers of operations, the consumption of system resources and the hindrance of operation efficiency can increase, which may result in a

failure to perform operations or unsatisfactory performance. In addition, there may be situations in which access to secure authenticator devices is limited or unavailable, thereby adversely impacting secure encryption, data transmission, and access to secure registration and authentication, which may lead to a compromised account that can be used to initiate fraudulent transactions or an account takeover.

**[0006]** These and other deficiencies exist. Accordingly, there is a need for systems and methods for authentication that overcome these deficiencies by protecting communications from interception and unauthorized access in a secure and reliable manner while effectively utilizing multi-device key pair cryptographic authentication and securely and quickly identifying the originator and determining their authenticity.

## SUMMARY OF THE DISCLOSURE

**[0007]** Embodiments of the present disclosure provide an authenticator. The authenticator may include a processor and a memory. The processor may be configured to receive one or more challenges. The processor may be configured to generate a first instruction, the first instruction including a request to retrieve a first Fast Identity Online (FIDO) key. The processor may be configured to transmit the first instruction. The processor may be configured to receive the first FIDO key. The processor may be configured to sign the one or more challenges using the first FIDO key. The processor may be configured to transmit one or more signed challenges for validation using a second FIDO key.

**[0008]** Embodiments of the present disclosure provide a method of authentication. The method may include receiving, at a processor, one or more challenges. The method may include generating, by the processor, a first instruction including a request to retrieve a first Fast Identity Online

(FIDO) key. The method may include transmitting, by the processor, the first instruction. The method may include receiving the first FIDO key. The method may include signing, by the processor, the one or more challenges using the first FIDO key. The method may include transmitting, by the processor, one or more signed challenges for validation using a second FIDO key.

**[0009]** Embodiments of the present disclosure provide a computer readable non-transitory medium comprising computer executable instructions that, when executed on a processor, perform procedures comprising the steps of: receiving one or more challenges; generating a first instruction including a request to retrieve a first Fast Identity Online (FIDO) key; transmitting the first instruction; receiving the first FIDO key; signing the one or more challenges using the first FIDO key; and transmitting one or more signed challenges for validation using a second FIDO key.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[00010]** Various embodiments of the present disclosure, together with further objects and advantages, may best be understood by reference to the following description taken in conjunction with the accompanying drawings.

**[00011]** Figure 1 depicts an authentication system according to an exemplary embodiment.

**[00012]** Figure 2A is an illustration of a first device according to an exemplary embodiment.

**[00013]** Figure 2B is an illustration of a contact pad of a first device according to an exemplary embodiment.

**[00014]** Figure 3 depicts an authentication method according to an exemplary embodiment.

**[00015]** Figure 4 depicts a sequence diagram of an authentication process according to an exemplary embodiment.

**[00016]** Figure 5 depicts an authentication method according to an exemplary embodiment.

**[00017]** Figure 6 depicts an authentication method according to an exemplary embodiment.

## DETAILED DESCRIPTION

**[00018]** The following description of embodiments provides non-limiting representative examples referencing numerals to particularly describe features and teachings of different aspects of the invention. The embodiments described should be recognized as capable of implementation separately, or in combination, with other embodiments from the description of the embodiments. A person of ordinary skill in the art reviewing the description of embodiments should be able to learn and understand the different described aspects of the invention. The description of embodiments should facilitate understanding of the invention to such an extent that other implementations, not specifically covered but within the knowledge of a person of skill in the art having read the description of embodiments, would be understood to be consistent with an application of the invention.

**[00019]** The systems and methods disclosed herein may be used to supplement authentication frameworks, including without limitation, Fast Identity Online (FIDO) authentication, Fast Identity Online 2 (FIDO2) authentication, WebAuthn, Client to Authenticator Protocol (CTAP) FIDO, other authentication implementations. Systems and methods employed herein may be implemented with distributed storage, cloud-based storage, and other forms of storage in support of this functionality. The systems and methods disclosed herein allow a device to serve as a roaming authenticator in a Fast Identity Online 2 (FIDO2) authentication in a browser. As further explained below, a user may authenticate with a website in a passwordless environment using the implemented systems and methods. To log into the website, a private key may be stored on the

device during the website registration process. When the user desires to log into the website again, the device may enter a communication field, such as that of a mobile device, via one or more gestures including but not limited to a tap, a swipe, a wave, or the like. The private key may then be transmitted from the first device to an application comprising instructions for execution on the mobile device, including but not limited to an authenticator. The authenticator may be configured to sign the challenge issued by the website using the private key. Once the challenge has been signed, it may be transmitted to a relying party server and validated using the public key stored. In this manner, this authentication may serve as primary authentication to log in the user. Furthermore, this authentication may be used in conjunction with a secondary factor authentication, such as biometrics or credential, entered on the mobile device.

**[00020]** Benefits of the systems and methods disclosed herein include improved authentication by protecting communications from interception and unauthorized access and securely and quickly identifying the originator and determining their authenticity. The systems and methods disclosed herein allow for the avoidance of man-in-the-middle attacks and phishing attacks, prevention of replay attacks, improved accessibility and security of authenticator devices and secure encryption and data transmission, and the reduction of other security vulnerabilities by using, without limitation iOS® and Android® executing on devices, as an authenticator in a FIDO2 authentication in a browser, such as a laptop browser or tablet browser or desktop browser while implementing conditional multi-factor authentication.

**[00021]** Further, a concern with the FIDO2 framework and other authentication frameworks is establishing the identity of the user seeking to undertake the authentication process. Systems and methods described herein may reduce this vulnerability by confirming the user attempting to

register credentials and authenticate via the framework is the user he or she claims to be and is authorized to perform the authentication process. By doing so, framework security and the ability to exclude unauthorized users may be increased. Accordingly, security risks may be further mitigated and compatibility across various devices and transaction efficiency may be further improved. In some examples, computation processing of the authenticator is reduced when instructing the first device to retrieve or generate one or more FIDO keys instead of generating the one or more FIDO keys in the authenticator.

**[00022]** These features can be implemented without degrading the user experience by burdening the user with unnecessary security tasks. Further, these features may be performed in a manner that allows for the time-efficient performance of transactions, in order to comply with user expectations and transaction requirements.

**[00023]** Accordingly, the systems and methods disclosed herein reduce the risk of fraudulent activity, such as misuse of the card or an account associated with the card, while also improving secure access to authenticator devices and encrypted data transmissions. Systems and methods disclosed herein improve upon implementations that lack secure authentication. These benefits may be advantageously achieved while promoting system efficiency, avoiding degradation of the user experience, and promoting compatibility across numerous devices.

**[00024]** Figure 1 illustrates an authentication system 100. The system 100 may comprise a browser extension 101, first device 105, an authenticator 110 or a second device, a network 115, a relying party or server 120, and a database 125. Although Figure 1 illustrates single instances of components of system 100, system 100 may include any number of components.

**[00025]** System 100 may include a browser extension 101. The browser extension 101 may

comprise Chrome ®, Internet Explorer ®, Firefox ®, or Safari ®. It is understood that software applications other than a browser extension may be utilized, including stand-alone software applications. Without limitation, an authentication request, such as a website registration may take place on any device, including but not limited to a laptop or a desktop associated with the browser extension 101. A mobile-based browser extension 101, or a tablet-based browser extension 101, or a laptop-based browser extension 101, or a desktop-based browser extension 101, may be configured to transmit and receive one or more requests, as further explained below.

**[00026]** System 100 may include a first device 105. Without limitation, the first device 105 may comprise a contactless card, a contact-based card, a network-enabled computer, or other device described herein. As referred to herein, a network-enabled computer may include, but is not limited to a computer device, or communications device including, e.g., a server, a network appliance, a personal computer, a workstation, a phone, a handheld PC, a personal digital assistant, a contactless card, a thin client, a fat client, an Internet browser, a kiosk, a tablet, a terminal, or other device. As further explained below in FIGs. 2A-2B, first device 105 may include one or more processors 102, and memory 104. Memory 104 may include one or more applets 106 and one or more counters 108. Each counter 108 may include a counter value. Memory 104 may include the counter value, transmission data, and at least one key.

**[00027]** First device 105 may include a communication interface 107. The communication interface 107 may comprise communication capabilities with physical interfaces and contactless interfaces. For example, the communication interface 107 may be configured to communicate with a physical interface, such as by swiping through a card swipe interface or inserting into a card chip reader found on an automated teller machine (ATM) or other device configured to

communicate over a physical interface. In other examples, the communication interface 107 may be configured to establish contactless communication with a card reading device via a short-range wireless communication method, such as NFC, Bluetooth, Wi-Fi, RFID, and other forms of contactless communication. As shown in FIG. 1, the communication interface 107 may be configured to communicate directly with the authenticator 110 or second device, relying party or server 120, and/or database 125 via network 115.

**[00028]** First device 105 may be in data communication with any number of components of system 100. For example, first device 105 may transmit data via network 115 to the authenticator 110 or second device, and/or relying party or server 120. First device 105 may transmit data via network 115 to database 125. In some examples, first device 105 may be configured to transmit data via network 115 after entry into one or more communication fields of any device. Without limitation, each entry may be associated with a tap, a swipe, a wave, and/or any combination thereof.

**[00029]** System 100 may include an authenticator 110. The authenticator 110 may comprise a roaming authenticator to another client device. In some examples, the authenticator 110 may include a mobile device that serves as a roaming authenticator to a laptop, a desktop, or a tablet. It is understood that the client devices are not limited to such devices, and that other client devices are within the scope of the invention.

**[00030]** By way of example, the authenticator 110 may comprise a second device. The authenticator 110 may include one or more processors 112, and memory 114. Memory 114 may include one or more applications, including but not limited to application 116. The authenticator 110 may be in data communication with any number of components of system 100. For example,

the authenticator 110 may transmit data via network 115 to server 120. The authenticator 110 may transmit data via network 115 to database 125. Without limitation, the authenticator 110 may be a network-enabled computer. As referred to herein, a network-enabled computer may include, but is not limited to a computer device, or communications device including, e.g., a server, a network appliance, a personal computer, a workstation, a phone, a handheld PC, a personal digital assistant, a contactless card, a thin client, a fat client, an Internet browser, a kiosk, a tablet, a terminal, or other device. The authenticator 110 also may be a mobile device; for example, a mobile device may include an iPhone, iPod, iPad from Apple® or any other mobile device running Apple's iOS® operating system, any device running Microsoft's Windows® Mobile operating system, any device running Google's Android® operating system, and/or any other smartphone, tablet, or like wearable mobile device.

**[00031]** The authenticator 110 may include processing circuitry and may contain additional components, including processors, memories, error and parity/CRC checkers, data encoders, anticollision algorithms, controllers, command decoders, security primitives and tamperproofing hardware, as necessary to perform the functions described herein. The authenticator 110 may further include a display and input devices. The display may be any type of device for presenting visual information such as a computer monitor, a flat panel display, and a mobile device screen, including liquid crystal displays, light-emitting diode displays, plasma panels, and cathode ray tube displays. The input devices may include any device for entering information into the user's device that is available and supported by the user's device, such as a touch-screen, keyboard, mouse, cursor-control device, touch-screen, microphone, digital camera, video recorder or camcorder. These devices may be used to enter information and interact with the software and

other devices described herein.

**[00032]** System 100 may include a network 115. In some examples, network 115 may be one or more of a wireless network, a wired network or any combination of wireless network and wired network, and may be configured to connect to any one of components of system 100. For example, first device 105 may be configured to connect to the relying party or server 120 via network 115. In some examples, network 115 may include one or more of a fiber optics network, a passive optical network, a cable network, an Internet network, a satellite network, a wireless local area network (LAN), a Global System for Mobile Communication, a Personal Communication Service, a Personal Area Network, Wireless Application Protocol, Multimedia Messaging Service, Enhanced Messaging Service, Short Message Service, Time Division Multiplexing based systems, Code Division Multiple Access based systems, D-AMPS, Wi-Fi, Fixed Wireless Data, IEEE 802.11b, 802.15.1, 802.11n and 802.11g, Bluetooth, NFC, Radio Frequency Identification (RFID), Wi-Fi, and/or the like.

**[00033]** In addition, network 115 may include, without limitation, telephone lines, fiber optics, IEEE Ethernet 902.3, a wide area network, a wireless personal area network, a LAN, or a global network such as the Internet. In addition, network 115 may support an Internet network, a wireless communication network, a cellular network, or the like, or any combination thereof. Network 115 may further include one network, or any number of the exemplary types of networks mentioned above, operating as a stand-alone network or in cooperation with each other. Network 115 may utilize one or more protocols of one or more network elements to which they are communicatively coupled. Network 115 may translate to or from other protocols to one or more protocols of network devices. Although network 115 is depicted as a single network, it should be appreciated that

according to one or more examples, network 115 may comprise a plurality of interconnected networks, such as, for example, the Internet, a service provider's network, a cable television network, corporate networks, such as credit card association networks, and home networks.

**[00034]** System 100 may include a server 120. For example, and without limitation, the server 120 may comprise a relying party. In some examples, the server 120 may include one or more processors 122 coupled to memory 124. Without limitation, the server 120 may comprise a cloud-based authenticator. The server 120 may be configured as a central system, server or platform to control and call various data at different times to execute a plurality of workflow actions. The server 120 may be configured to connect to the first device 105. The server 120 may be in data communication with the applet 106 and/or application 116. For example, a server 120 may be in data communication with applet 106 via one or more networks 115. The first device 105 may be in communication with one or more servers 120 via one or more networks 115, and may operate as a respective front-end to back-end pair with server 120. The first device 105 may transmit, for example from applet 106 executing thereon, one or more requests to the server 120. The one or more requests may be associated with retrieving data from the server 120. The server 120 may receive the one or more requests from the first device 105. Based on the one or more requests from applet 106, the server 120 may be configured to retrieve the requested data. The server 120 may be configured to transmit the received data to applet 106, the received data being responsive to one or more requests.

**[00035]** In some examples, the server 120 can be a dedicated server computer, such as bladed servers, or can be personal computers, laptop computers, notebook computers, palm top computers, network computers, mobile devices, wearable devices, or any processor-controlled

device capable of supporting the system 100. While FIG. 1 illustrates a single relying party or server 120, it is understood that other embodiments can use multiple servers or multiple computer systems as necessary or desired to support the users and can also use back-up or redundant servers to prevent network downtime in the event of a failure of a particular server.

**[00036]** The server 120 may include an application comprising instructions for execution thereon. For example, the application may comprise instructions for execution on the server 120. The application of the server 120 may be in communication with any components of system 100. For example, server 120 may execute one or more applications that enable, for example, network and/or data communications with one or more components of system 100 and transmit and/or receive data. Without limitation, the server 120 may be a network-enabled computer. As referred to herein, a network-enabled computer may include, but is not limited to a computer device, or communications device including, e.g., a server, a network appliance, a personal computer, a workstation, a phone, a handheld PC, a personal digital assistant, a contactless card, a thin client, a fat client, an Internet browser, or other device. The server 120 also may be a mobile device; for example, a mobile device may include an iPhone, iPod, iPad from Apple® or any other mobile device running Apple's iOS® operating system, any device running Microsoft's Windows® Mobile operating system, any device running Google's Android® operating system, and/or any other smartphone, tablet, or like wearable mobile device.

**[00037]** The server 120 may include processing circuitry and may contain additional components, including processors, memories, error and parity/CRC checkers, data encoders, anticollision algorithms, controllers, command decoders, security primitives and tamperproofing hardware, as necessary to perform the functions described herein. The server 120 may further

include a display and input devices. The display may be any type of device for presenting visual information such as a computer monitor, a flat panel display, and a mobile device screen, including liquid crystal displays, light-emitting diode displays, plasma panels, and cathode ray tube displays. The input devices may include any device for entering information into the user's device that is available and supported by the user's device, such as a touch-screen, keyboard, mouse, cursor-control device, touch-screen, microphone, digital camera, video recorder or camcorder. These devices may be used to enter information and interact with the software and other devices described herein.

**[00038]** System 100 may include one or more databases 125. The database 125 may comprise a relational database, a non-relational database, or other database implementations, and any combination thereof, including a plurality of relational databases and non-relational databases. In some examples, the database 125 may comprise a desktop database, a mobile database, or an in-memory database. Further, the database 125 may be hosted internally by any component of system 100, such as the first device 105 or server 120, or the database 125 may be hosted externally to any component of the system 100, such as the first device 105 or server 120, by a cloud-based platform, or in any storage device that is in data communication with the first device 105 and server 120. In some examples, database 125 may be in data communication with any number of components of system 100. For example, server 120 may be configured to retrieve the requested data from the database 125 that is transmitted by applet 106. Server 120 may be configured to transmit the received data from database 125 to applet 106 via network 115, the received data being responsive to the transmitted one or more requests. In other examples, applet 106 may be configured to transmit one or more requests for the requested data from database 125 via network

115.

**[00039]** In some examples, exemplary procedures in accordance with the present disclosure described herein can be performed by a processing arrangement and/or a computing arrangement (e.g., computer hardware arrangement). Such processing/computing arrangement can be, for example entirely or a part of, or include, but not limited to, a computer/processor that can include, for example one or more microprocessors, and use instructions stored on a computer-accessible medium (e.g., RAM, ROM, hard drive, or other storage device). For example, a computer-accessible medium can be part of the memory of the first device 105, authenticator 110, server 120, and/or database 125, or other computer hardware arrangement.

**[00040]** In some examples, a computer-accessible medium (e.g., as described herein above, a storage device such as a hard disk, floppy disk, memory stick, CD-ROM, RAM, ROM, etc., or a collection thereof) can be provided (e.g., in communication with the processing arrangement). The computer-accessible medium can contain executable instructions thereon. In addition or alternatively, a storage arrangement can be provided separately from the computer-accessible medium, which can provide the instructions to the processing arrangement so as to configure the processing arrangement to execute certain exemplary procedures, processes, and methods, as described herein.

**[00041]** The processor 122 of server 120 may be configured to receive an authentication request. For example, the authentication request may be received from browser extension 101. In some examples, the authentication request may comprise a request for Fast Identity Online 2 (FIDO2) website registration. The processor 122 may be configured to generate one or more challenges. For example, the processor 122 may be configured to generate a first challenge. In

some examples, the processor 122 may be configured to challenge the authenticator 110 to login. The first challenge may include an identifier, such as a user identifier or a site identifier, that may be used to select an appropriate FIDO key pair. The first challenge may further include an unpredictable number, which may be given by the server 120, that is used to prevent replay. For example, for each instance of authentication, a new unpredictable number will be needed and thus the unpredictable number will be different each time. In this manner, this avoids utilizing a signature of the old unpredictable number and instead utilizes the number for the instant session of authentication.

**[00042]** The authenticator 110, such as processor 112 of a mobile device, may be configured to prompt one or more entries of the first device 105. For example, the one or more entries may include at least one selected from the group of a tap, a swipe, a wave, or the like, and/or any combination thereof. Thus, the communication interface 107 of the first device 105 may enter a communication field of the authenticator 110, such as the communication field of the mobile device. The first device 105 may be configured to, depending on the first instruction received from the authenticator 110, generate or retrieve a FIDO key pair associated with a specific user or site. The generated or retrieved FIDO key pair may be read by the authenticator 110. The FIDO key pair may include a FIDO private key which may be read by the authenticator 110 from the first device 105 and may be used by the authenticator 110 to sign the first challenge resulting in a signed challenge data. The FIDO private key may be retrieved via entry of a communication interface 107 of the first device 105 into a communication field. In some examples, the FIDO private key may be encrypted by the first device 105 prior to its transfer to the authenticator 110, in which case the processor 112 of the mobile device may be configured to decrypt the received encrypted FIDO

private key. The server 120 may receive the signed challenge data from the authenticator 110, such as the processor 112 of the mobile device. The server 120 may perform validation of the signed challenge data. For example, the server 120 may validate the signed challenge data with the FIDO public key that was stored when the user registered to the site. The validation may comprise the result of the authentication process, which includes signature verification.

**[00043]** In some examples, transmission of the FIDO private key from the first device 105 to the authenticator 110 may be prevented. For example, over the air (OTA) transmission of the FIDO private key is avoided when the first device 105 performs the signing of the first challenge. Thus, the first challenge and the site or user identifier information is transmitted to the first device 105 from the authenticator, which thereby distributes its computation to the first device 105 through near field communication (NFC) via a proxy protocol.

**[00044]** In some examples, the FIDO private key may be generated based on a master key and an identifier associated with the authentication request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce another key pair. The first device 105 may be configured to store a finite number of keys in memory 104 so as to avoid having to regenerate keys. In other examples, the first instruction from the authenticator 110 may include a request to regenerate the FIDO private key, such as a private key which is part of a public/private key pair, by the first device 105.

**[00045]** In some examples, the master key be stored in memory 104 on the first device 105, and the master key may be combined with the site identifier on the first device 105 to generate the FIDO private key. In other examples, the master key may be transferred from the first device 105 to the authenticator 110, in which the combining of the master key and the site identifier is

performed by the authenticator 110 to generate the FIDO private key.

**[00046]** In some examples, rather than generating or storing the FIDO private key in the first device 105, the FIDO private key may be generated or stored in a secure element belonging to the authenticator 110. For example, the FIDO private key may be stored in a secure element maintained by an authenticator 110. In some examples, the secure element may comprise a tamper-resistant secure storage area in which one or more keys may be securely stored and retrieved by the server 120.

**[00047]** After instruction generation, the authenticator 110 may be configured to transmit the first instruction to the first device 105. The authenticator 110 may be configured to receive, based on the first instruction, the FIDO private key from the first device 105. The first instruction may be transmitted to the processor 102 of the first device 105 by the authenticator 110. For example, the processor 102 of the first device 105 may be configured to generate the FIDO private key. The processor 102 of the first device 105 may be configured to transmit the FIDO private key. For example, the processor 102 of the first device 105 may be configured to transmit the FIDO private key to the authenticator 110, such as an application 116 comprising instructions for execution on the authenticator 110. As previously explained above, the authenticator 110 may be, without limitation, a mobile device such as a laptop or a tablet or a phone. In some examples, the FIDO private key may be transmitted and received via one or more channels. For example, the FIDO private key may be transmitted and received via an out-of-band channel. The processor 102 of the first device 105 may be configured to transmit the FIDO private key to the authenticator 110 via entry of the communication interface 107 into a communication field of the authenticator 110. In some examples, the entry may be associated with one or more gestures, including but not limited

to one or more taps, swipes, waves, and/or any combination thereof. The authenticator 110 may be configured to transmit the first FIDO key to the processor 122 of the server 120 for validation, and thus the authenticator 110 may be configured to serve as an intermediary device between the first device 105 and the processor 122 of the server 120.

**[00048]** The processor 122 may be configured to validate the signed challenge data transmitted from the authenticator 110 using a second FIDO key, such as a FIDO public key, which may be part of the public/private key pair as previously explained. For example, the application 116 comprising instructions for execution on the authenticator 110 may be configured to sign the first challenge issued by the website using the FIDO private key. Once the challenge has been signed by authenticator 110, it may be transmitted to the processor 122 for validation using the FIDO public key stored in memory 124 of the server 120.

**[00049]** The processor 122 may be configured to generate a second instruction. For example, the second instruction may include a second request to transmit input data. After instruction generation, the processor 122 may be configured to transmit the second instruction. For example, the processor 122 may be configured to transmit the second instruction to the authenticator 110. In some examples, the second instruction may be forwarded by the processor 122 to the application 116 via one or more push notifications. In some examples, the second instruction may be transmitted by the processor 122 after evaluation by processor 122 and/or database 125 of one or more conditions. For example, at least one of the one or more conditions may include determining a threshold number of authentication requests over a predetermined time period. For example, the processor 122 and/or database 125 may be configured to determine whether an abnormal number of transactions or requests were performed within any number of seconds, minutes, hours, days,

weeks, months, years, etc. In another example, at least one of the one or more conditions may include determining whether misuse or fraud associated with an account and/or user occurred. For example, the processor 122 and/or database 125 may be configured to determine whether a user transaction history is indicative of excessive amount of purchases or abnormal locations. In this manner, conditional multifactor authentication may be implemented to improve security, which may trigger transmittal of the second instruction for receiving the input data.

**[00050]** The processor 122 may be configured to receive, based on the second instruction, the input data. The processor 122 may be configured to receive the input data from the authenticator 110, such as from application 116 comprising instructions for execution on the second device. The input data may include at least one selected from the group of biometric data and credential data. For example, the input data may include biometric data, credential data, and/or any combination thereof. Without limitation, the biometric data may include at least one selected from the group of a fingerprint, a facial scan, a retinal scan, a voice recognition, and/or any combination thereof. In some examples, the input data may additionally and/or alternatively include credential data. Without limitation, the login data may include at least one selected from the group of an entry of a user name, a password, an account number, a security code, a one-time passcode, an answer to a security question, and/or any combination thereof.

**[00051]** The processor 122 may be configured to complete the authentication request by authenticating the input data. For example, the processor 122 may be configured to generate one or more outcomes by comparing the received input data with reference input data. In some examples, the reference input data may be stored in memory 124 of the server 120. In other examples, the reference input data may be requested by the processor 122. For example, the

processor 122 may be configured to receive, via one or more requests, reference input data, or alternatively transmit the input data for comparison with the reference input data to a database 125. For example, the processor 122 of the server 120 may be configured to generate an outcome indicative of successful authentication when it results in a successful match based on the comparison between the received input data with reference input data. In another example, the processor 122 of the server 120 may be configured to generate an outcome of unsuccessful authentication when it results in an unsuccessful match based on the comparison between the received input data with reference input data. Upon determination of an unsuccessful authentication, the processor 122 of the server 120 may be configured to re-authenticate the input data up to and including a predetermined number of attempts before successfully authenticating the input data so as to complete the authentication request, or aborting the completion of the authentication request. In this manner, system 100 may implement with distributed storage, cloud-based storage, and other forms of storage in support of the aforementioned functionalities.

**[00052]** FIG. 2A illustrates one or more first devices 200. First device 200 may reference the same or similar components of first device 105, as explained above with respect to FIG. 1. Although Figure 2A and 2B illustrate single instances of components of first device 200, any number of components may be utilized.

**[00053]** First device 200 may be configured to communicate with one or more components of system 100. First device 200 may comprise a contact-based card (e.g., a card read by a swipe of a magnetic stripe or by insertion into a chip reader) or a contactless card, and the first device 200 may comprise a payment card, such as a credit card, debit card, or gift card, issued by a service provider 205 displayed on the front or back of the first device 200. In some examples, the first

device 200 is not related to a payment card, and may comprise, without limitation, an identification card, a membership card, and a transportation card. In some examples, the payment card may comprise a dual interface contactless payment card.

**[00054]** First device 200 may comprise a substrate 210, which may include a single layer or one or more laminated layers composed of plastics, metals, and other materials. Exemplary substrate materials include polyvinyl chloride, polyvinyl chloride acetate, acrylonitrile butadiene styrene, polycarbonate, polyesters, anodized titanium, palladium, gold, carbon, paper, and biodegradable materials. In some examples, the first device 200 may have physical characteristics compliant with the ID-1 format of the ISO/IEC 7810 standard, and the first device 200 may otherwise be compliant with the ISO/IEC 14443 standard. However, it is understood that the first device 200 according to the present disclosure may have different characteristics, and the present disclosure does not require implementation in a payment card.

**[00055]** The first device 200 may also include identification information 215 displayed on the front and/or back of the card, and a contact pad 220. The contact pad 220 may be configured to establish contact with another communication device, including but not limited to a user device, smartphone, laptop, desktop, or tablet computer. The first device 200 may also include processing circuitry, antenna and other components not shown in FIG. 2A. These components may be located behind the contact pad 220 or elsewhere on the substrate 210. The first device 200 may also include a magnetic strip or tape, which may be located on the back of the card (not shown in FIG. 2A).

**[00056]** As illustrated in FIG. 2B, the contact pad 220 of FIG. 2A may include processing circuitry 225 for storing and processing information, including a processor 230, such as a microprocessor, and a memory 235. It is understood that the processing circuitry 225 may contain

additional components, including processors, memories, error and parity/CRC checkers, data encoders, anticollision algorithms, controllers, command decoders, security primitives and tamperproofing hardware, as necessary to perform the functions described herein.

**[00057]** The memory 235 may be a read-only memory, write-once read-multiple memory or read/write memory, e.g., RAM, ROM, and EEPROM, and the first device 200 may include one or more of these memories. A read-only memory may be factory programmable as read-only or one-time programmable. One-time programmability provides the opportunity to write once then read many times. A write once/read-multiple memory may be programmed at a point in time after the memory chip has left the factory. Once the memory is programmed, it may not be rewritten, but it may be read many times. A read/write memory may be programmed and re-programed many times after leaving the factory. It may also be read many times.

**[00058]** The memory 235 may be configured to store one or more applets 240, one or more counters 245, and a customer identifier 250. The one or more applets 240 may comprise one or more software applications configured to execute on one or more contact-based or contactless cards, such as Java Card applet. However, it is understood that applets 240 are not limited to Java Card applets, and instead may be any software application operable on contact-based or contactless cards or other devices having limited memory. The one or more counters 245 may comprise a numeric counter sufficient to store an integer. The customer identifier 250 may comprise a unique alphanumeric identifier assigned to a user of the first device 200, and the identifier may distinguish the user of the contactless card from other contactless card users. In some examples, the customer identifier 250 may identify both a customer and an account assigned to that customer and may further identify the contactless card associated with the customer's account.

**[00059]** The processor and memory elements of the foregoing exemplary embodiments are described with reference to the contact pad, but the present disclosure is not limited thereto. It is understood that these elements may be implemented outside of the contact pad 220 or entirely separate from it, or as further elements in addition to processor 230 and memory 235 elements located within the contact pad 220.

**[00060]** In some examples, the first device 200 may comprise one or more antennas 255. The one or more antennas 255 may be placed within the first device 200 and around the processing circuitry 225 of the contact pad 220. For example, the one or more antennas 255 may be integral with the processing circuitry 225 and the one or more antennas 255 may be used with an external booster coil. As another example, the one or more antennas 255 may be external to the contact pad 220 and the processing circuitry 225.

**[00061]** In an embodiment, the coil of first device 200 may act as the secondary of an air core transformer. The terminal may communicate with the first device 200 by cutting power or amplitude modulation. The first device 200 may infer the data transmitted from the terminal using the gaps in the first device's power connection, which may be functionally maintained through one or more capacitors. The first device 200 may communicate back by switching a load on the first device's coil or load modulation. Load modulation may be detected in the terminal's coil through interference.

**[00062]** Figure 3 depicts a method 300 of authentication. Figure 3 may reference the same or similar components of system 100 of FIG. 1, and first device 200 of FIG. 2A and FIG. 2B.

**[00063]** At block 305, the method 300 may include receiving, at a processor, one or more challenges. For example, a relying party or server may be configured to generate a first challenge.

In some examples, the relying party may be configured to challenge the authenticator, such as a processor of a mobile device, to login. The first challenge may include an identifier, such as a user identifier or a site identifier, that may be used to select an appropriate FIDO key pair. The first challenge may further include an unpredictable number, which may be given by the relying party, that is used to prevent replay. For example, for each instance of authentication, a new unpredictable number will be needed and thus the unpredictable number will be different each time. In this manner, this avoids utilizing a signature of the old unpredictable number and instead utilizes the number for the instant session of authentication.

**[00064]** In some examples, the first request may comprise an authentication request. The authentication request may comprise a request for Fast Identity Online 2 (FIDO2) website registration. The processor may be configured to generate one or more instructions. For example, the processor may be configured to generate a first instruction. Each instruction may include one or more requests. The first instruction may include a first request to retrieve a first key from the processor of the first device. The key may comprise a FIDO key. The first key may be retrieved via entry of a communication interface of the first device into a communication field. In some examples, the first instruction may include a request to generate the first key by the processor of the first device. In other examples, the first instruction may include a request to regenerate the first FIDO key, such as a private key which is part of a public/private key pair, by the processor of the first device.

**[00065]** At block 310, the method 300 may include generating, by the processor, a first instruction including a request to retrieve a first Fast Identity Online (FIDO) key from a first device. For example, based on the first challenge received from the relying party, the authenticator,

such as processor of a mobile device, may be configured to prompt one or more entries of a first device by generating one or more instructions. For example, the one or more entries may include at least one selected from the group of a tap, a swipe, a wave, or the like, and/or any combination thereof. The first instruction may include a request to generate the first FIDO key by the first device. In some examples, the first FIDO key may be generated based on a master key and an identifier associated with the first request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce a key pair. In some examples, the master key be stored on the first device, and the master key may be combined with the site identifier on the first device to generate the private key. In other examples, the master key may be transferred from the first device to the application comprising instructions for execution on the second device, in which the combining of the master key and the site identifier is performed by the application of the second device to generate the private key. In some examples, rather than generating or storing the FIDO private key in the first device, the first FIDO key may be generated or stored in a secure element belonging to the second device. For example, the FIDO private key may be stored in a secure element maintained by the second device. In some examples, the secure element may comprise a tamper-resistant secure storage area in which one or more keys may be securely stored and retrieved by the server.

**[00066]** At block 315, the method 300 may include transmitting, by the processor, the first instruction. For example, the processor may be configured to transmit the first instruction to the first device. In some examples, the first instruction may be transmitted to the processor of the first device by the authenticator. After instruction generation, the authenticator may be configured to transmit the first instruction to the first device.

**[00067]** At block 320, the method 300 may include receiving the first FIDO key. For example, a communication interface of the first device may enter a communication field of the authenticator, such as the communication field of the mobile device, to transfer the first FIDO key. The first device may be configured to, depending on the first instruction received from the authenticator, generate or retrieve a FIDO key pair associated with a specific user or site. The generated or retrieved FIDO key pair may be read by the authenticator. The FIDO key pair may include a FIDO private key which may be read by the authenticator from the first device and may be used by the authenticator to sign the first challenge resulting in a signed challenge data. The FIDO private key may be retrieved via entry of a communication interface of the first device into a communication field. In some examples, the FIDO private key may be encrypted by the first device prior to its transfer to the authenticator, in which case the processor of the mobile device may be configured to decrypt the received encrypted FIDO private key.

**[00068]** In some examples, the FIDO private key may be generated based on a master key and an identifier associated with the authentication request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce another key pair. The first device may be configured to store a finite number of keys in its memory so as to avoid having to regenerate keys. In other examples, the first instruction from the authenticator may include a request to regenerate the FIDO private key, such as a private key which is part of a public/private key pair, by the first device.

**[00069]** In some examples, the master key be stored in memory on the first device, and the master key may be combined with the site identifier on the first device to generate the FIDO private key. In other examples, the master key may be transferred from the first device to the authenticator,

in which the combining of the master key and the site identifier is performed by the authenticator to generate the FIDO private key.

**[00070]** In some examples, transmission of the FIDO private key from the first device to the authenticator may be prevented. For example, over the air (OTA) transmission of the FIDO private key is avoided when the first device performs the signing of the first challenge. Thus, the first challenge and the site or user identifier information is transmitted to the first device from the authenticator, which thereby distributes its computation to the first device through near field communication (NFC) via a proxy protocol.

**[00071]** In some examples, rather than generating or storing the FIDO private key in the first device, the FIDO private key may be generated or stored in a secure element belonging to the authenticator. For example, the FIDO private key may be stored in a secure element maintained by a second device. In some examples, the secure element may comprise a tamper-resistant secure storage area in which one or more keys may be securely stored and retrieved by the relying party.

**[00072]** In some examples, the processor of the first device may be configured to generate the FIDO private key. The processor of the first device may be configured to transmit the FIDO private key. For example, the processor of the first device may be configured to transmit the FIDO private key to the authenticator, such as an application comprising instructions for execution on a second device. As previously explained above, the authenticator may be, without limitation, a mobile device such as a laptop or a tablet or a phone. In some examples, the FIDO private key may be transmitted and received via one or more channels. For example, the FIDO private key may be transmitted and received via an out-of-band channel. The processor of the first device may be configured to transmit the FIDO private key to the authenticator via entry of the communication

interface of the first device into a communication field of the authenticator. In some examples, the entry may be associated with one or more gestures, including but not limited to one or more taps, swipes, waves, and/or any combination thereof.

**[00073]** For example, the processor of the first device may be configured to generate the first FIDO key. The processor of the first device may be configured to transmit the first FIDO key. For example, the processor of the first device may be configured to transmit the first FIDO key to an application comprising instructions for execution on a second device. As previously explained above, the second device may be, without limitation, a mobile device such as a laptop or a tablet or a phone. In some examples, the first FIDO key may be transmitted and received via one or more channels. For example, the first FIDO key may be transmitted and received via an out-of-band channel. The processor of the first device may be configured to transmit the first FIDO key to the application of the second device via entry of the communication interface into a communication field of the second device. In some examples, the entry may be associated with one or more gestures, including but not limited to one or more taps, swipes, waves, and/or any combination thereof. The application of the second device may be configured to transmit the first FIDO key to the processor of the server for validation, and thus the second device may be configured to serve as an intermediary device between the first device and the processor.

**[00074]** At block 325, the method 300 may include signing, by the processor, the one or more challenges using the first FIDO key.

**[00075]** At block 330, the method 300 may include transmitting, by the processor, one or more signed challenges for validation using a second FIDO key. For example, the processor may be configured to transmit one or more signed challenges to the relying party. The relying party may

receive the signed challenge data from the authenticator, such as the processor of the mobile device. The relying party may perform validation of the signed challenge data. For example, the relying party may validate the signed challenge data with the FIDO public key that was stored when the user registered to the site. The validation may comprise the result of the authentication process, which includes signature verification. The authenticator may be configured to transmit the first FIDO key to the processor of the relying party for validation, and thus the authenticator may be configured to serve as an intermediary device between the first device and the processor of the relying party.

**[00076]** The relying party may be configured to validate the FIDO private key via a website FIDO service. The relying party may be configured to validate the signed challenge data transmitted from the authenticator using a second FIDO key, such as a FIDO public key, which may be part of the public/private key pair as previously explained. For example, the application comprising instructions for execution on the second device may be configured to sign the first challenge issued by the website using the FIDO private key. Once the challenge has been signed by authenticator, it may be transmitted to the processor for validation using the FIDO public key stored in memory of the server.

**[00077]** The relying party may be configured to generate a second instruction. For example, the second instruction may include a second request to transmit input data. After instruction generation, the relying party may be configured to transmit the second instruction. For example, the relying party may be configured to transmit the second instruction to the authenticator. In some examples, the second instruction may be forwarded by the relying party to the application via one or more push notifications. In some examples, the second instruction may be transmitted by the

relying party after evaluation by the relying party and/or a database of one or more conditions. For example, at least one of the one or more conditions may include determining a threshold number of authentication requests over a predetermined time period. For example, the relying party and/or database may be configured to determine whether an abnormal number of transactions or requests were performed within any number of seconds, minutes, hours, days, weeks, months, years, etc. In another example, at least one of the one or more conditions may include determining whether misuse or fraud associated with an account and/or user occurred. For example, the relying party and/or the database may be configured to determine whether a user transaction history is indicative of excessive amount of purchases or abnormal locations. In this manner, conditional multifactor authentication may be implemented to improve security, which may trigger transmittal of the second instruction for receiving the input data.

**[00078]** The relying party may be configured to receive, based on the second instruction, the input data. The relying party may be configured to receive the input data from the authenticator, such as from the application comprising instructions for execution on the second device. The input data may include at least one selected from the group of biometric data and credential data. For example, the input data may include biometric data, credential data, and/or any combination thereof. Without limitation, the biometric data may include at least one selected from the group of a fingerprint, a facial scan, a retinal scan, a voice recognition, and/or any combination thereof. In some examples, the input data may additionally and/or alternatively include credential data. Without limitation, the login data may include at least one selected from the group of an entry of a user name, a password, an account number, a security code, a one-time passcode, an answer to a security question, and/or any combination thereof.

**[00079]** The relying party may be configured to complete the authentication request by authenticating the input data. For example, the relying party may be configured to generate one or more outcomes by comparing the received input data with reference input data. In some examples, the reference input data may be stored in a memory of the relying party. In other examples, the reference input data may be requested by the relying party. For example, the relying party may be configured to receive, via one or more requests, reference input data, or alternatively transmit the input data for comparison with the reference input data to the database. For example, the processor of the relying party may be configured to generate an outcome indicative of successful authentication when it results in a successful match based on the comparison between the received input data with reference input data. In another example, the processor of the relying party may be configured to generate an outcome of unsuccessful authentication when it results in an unsuccessful match based on the comparison between the received input data with reference input data. Upon determination of an unsuccessful authentication, the processor of the relying party may be configured to re-authenticate the input data up to and including a predetermined number of attempts before successfully authenticating the input data so as to complete the authentication request, or aborting the completion of the authentication request. In this manner, the authentication system may implement with distributed storage, cloud-based storage, and other forms of storage in support of the aforementioned functionalities.

**[00080]** Figure 4 depicts a sequence diagram 400 of an authentication process according to an exemplary embodiment. Figure 4 may reference the same or similar components of system 100, first device 200 of FIG. 2A and FIG. 2B, and method 300 of FIG. 3.

**[00081]** At step 405, a processor may be configured to receive an authentication request. As an

example, the processor may be part of a relying party or server. In some examples, the authentication request may comprise a request for Fast Identity Online 2 (FIDO2) website registration. For example, the request may be received from a browser extension.

**[00082]** At step 410, the processor may be configured to generate, responsive to the authentication request, one or more challenges. For example, the processor may be configured to generate a first challenge. In some examples, the relying party may be configured to challenge the authenticator, such as a processor of a mobile device, to login. The first challenge may include an identifier, such as a user identifier or a site identifier, that may be used to select an appropriate FIDO key pair. The first challenge may further include an unpredictable number, which may be given by the relying party, that is used to prevent replay. For example, for each instance of authentication, a new unpredictable number will be needed and thus the unpredictable number will be different each time. In this manner, this avoids utilizing a signature of the old unpredictable number and instead utilizes the number for the instant session of authentication.

**[00083]** At step 415, the processor may be configured to transmit the first challenge. For example, the processor may be configured to transmit the first instruction to an authenticator, such as the application comprising instructions for execution on a device, including but not limited to a mobile device.

**[00084]** At step 420, a first instruction may be transmitted to a first device. For example, the first instruction may be generated and/or transmitted by the authenticator to a processor of a card. The processor may be configured to generate one or more instructions. For example, the processor may be configured to generate a first instruction. Each instruction may include one or more requests. The first instruction may include a first request to retrieve a first key from the processor

of the first device. The key may comprise a FIDO key. The first key may be retrieved via entry of a communication interface of the first device into a communication field, such as the communication field of the authenticator. In some examples, the first instruction may include a request to generate the first key by the processor of the first device. In other examples, the first instruction may include a request to regenerate the first FIDO key, such as a private key which is part of a public/private key pair, by the processor of the first device.

**[00085]** For example, based on the first challenge received from the relying party, the authenticator, such as processor of the mobile device, may be configured to prompt one or more entries of a first device by generating one or more instructions. For example, the one or more entries may include at least one selected from the group of a tap, a swipe, a wave, or the like, and/or any combination thereof. The first instruction may include a request to generate the first FIDO key by the first device. In some examples, the first FIDO key may be generated based on a master key and an identifier associated with the first request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce a key pair. In some examples, the master key be stored on the first device, and the master key may be combined with the site identifier on the first device to generate the private key.

**[00086]** At step 425, the processor of the first device may be configured to generate and transmit the first FIDO key. For example, a communication interface of the first device may enter a communication field of the authenticator, such as the communication field of the mobile device, to transfer the first FIDO key. The first device may be configured to, depending on the first instruction received from the authenticator, generate or retrieve a FIDO key pair associated with a specific user or site. The generated or retrieved FIDO key pair may be read by the authenticator.

The FIDO key pair may include a FIDO private key which may be read by the authenticator from the first device and may be used by the authenticator to sign the first challenge resulting in a signed challenge data. The FIDO private key may be retrieved via entry of a communication interface of the first device into a communication field. In some examples, the FIDO private key may be encrypted by the first device prior to its transfer to the authenticator, in which case the processor of the mobile device may be configured to decrypt the received encrypted FIDO private key.

**[00087]** In some examples, the FIDO private key may be generated based on a master key and an identifier associated with the authentication request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce another key pair. The first device may be configured to store a finite number of keys in its memory so as to avoid having to regenerate keys. In other examples, the first instruction from the authenticator may include a request to regenerate the FIDO private key, such as a private key which is part of a public/private key pair, by the first device.

**[00088]** In some examples, the master key be stored in memory on the first device, and the master key may be combined with the site identifier on the first device to generate the FIDO private key. In other examples, the master key may be transferred from the first device to the authenticator, in which the combining of the master key and the site identifier is performed by the authenticator to generate the FIDO private key.

**[00089]** In some examples, transmission of the FIDO private key from the first device to the authenticator may be prevented. For example, over the air (OTA) transmission of the FIDO private key is avoided when the first device performs the signing of the first challenge. Thus, the first challenge and the site or user identifier information is transmitted to the first device from the

authenticator, which thereby distributes its computation to the first device through near field communication (NFC) via a proxy protocol.

**[00090]** In some examples, rather than generating or storing the FIDO private key in the first device, the FIDO private key may be generated or stored in a secure element belonging to the authenticator. For example, the FIDO private key may be stored in a secure element maintained by a second device. In some examples, the secure element may comprise a tamper-resistant secure storage area in which one or more keys may be securely stored and retrieved by the relying party.

**[00091]** In some examples, the processor of the first device may be configured to generate the FIDO private key. The processor of the first device may be configured to transmit the FIDO private key. For example, the processor of the first device may be configured to transmit the FIDO private key to the authenticator, such as an application comprising instructions for execution on a second device. As previously explained above, the authenticator may be, without limitation, a mobile device such as a laptop or a tablet or a phone. In some examples, the FIDO private key may be transmitted and received via one or more channels. For example, the FIDO private key may be transmitted and received via an out-of-band channel. The processor of the first device may be configured to transmit the FIDO private key to the authenticator via entry of the communication interface of the first device into a communication field of the authenticator. In some examples, the entry may be associated with one or more gestures, including but not limited to one or more taps, swipes, waves, and/or any combination thereof.

**[00092]** At step 430, the authenticator may be configured to receive the first FIDO key. For example, the application of the second device may be configured to receive the first FIDO private key from the processor of the first device. The authenticator may be configured to sign the first

challenge using the received first FIDO key. The authenticator may be configured to transmit the signed first challenge to the relying party for validation.

**[00093]** At step 435, the processor of the relying party or server may be configured to validate the signed first challenge. For example, the processor of the relying party or server may be configured to receive the signed first challenge from the authenticator. For example, the relying party may validate the signed challenge data with the FIDO public key that was stored when the user registered to the site. The validation may comprise the result of the authentication process, which includes signature verification. The authenticator may be configured to transmit the first FIDO key to the processor of the relying party for validation, and thus the authenticator may be configured to serve as an intermediary device between the first device and the processor of the relying party.

**[00094]** At step 440, the processor of the relying party or server may be configured to generate a second instruction. For example, the second instruction may include a second request to transmit input data. In some examples, the processor may be configured to evaluate one or more conditions. For example, the second instruction may be transmitted by the processor after evaluation of one or more conditions. For example, at least one of the one or more conditions may include determining a threshold number of authentication requests over a predetermined time period. For example, the processor may be configured to determine whether an abnormal number of transactions or requests were performed within any number of seconds, minutes, hours, days, weeks, months, years, etc. In another example, at least one of the one or more conditions may include determining whether misuse or fraud associated with an account and/or user occurred. For example, the processor may be configured to determine whether a user transaction history is indicative of excessive amount of

purchases or abnormal locations. In this manner, conditional multifactor authentication may be implemented to improve security, which may trigger transmittal of the second instruction for receiving the input data.

**[00095]** At step 445, the authenticator may be configured to receive the second instruction. For example, the processor of the mobile device may be configured to receive, after evaluation of the one or more conditions by the relying party or server, the second instruction from the processor of the relying party or server. In some examples, the second instruction may be forwarded by the processor of the relying party or server to the authenticator, such as the application comprising instructions for execution of the mobile device via one or more push notifications.

**[00096]** At step 450, the authenticator may be configured to transmit, based on the second instruction, the input data. For example, the processor or application of the mobile device may be configured to transmit the input data to the relying party, such as the processor of the server. The input data may include at least one selected from the group of biometric data and credential data. For example, the input data may include biometric data, credential data, and/or any combination thereof. Without limitation, the biometric data may include at least one selected from the group of a fingerprint, a facial scan, a retinal scan, a voice recognition, and/or any combination thereof. In some examples, the input data may additionally and/or alternatively include credential data. Without limitation, the login data may include at least one selected from the group of an entry of a user name, a password, an account number, a security code, a one-time passcode, an answer to a security question, and/or any combination thereof.

**[00097]** At step 455, the relying party or server may be configured to complete the authentication request by authenticating the input data. For example, the processor of the relying

party or server may be configured to receive the input data from the authenticator, such as the processor or application comprising instructions for execution of the mobile device. In some examples, the processor of the relying party or server may be configured to generate one or more outcomes by comparing the received input data with reference input data. In some examples, the reference input data may be stored by the relying party or server. In other examples, the reference input data may be requested by the relying party or server. For example, the processor of the relying party or server may be configured to receive, via one or more requests, reference input data, or alternatively transmit the input data for comparison with the reference input data to a database. For example, the relying party or server may be configured to generate an outcome indicative of successful authentication when it results in a successful match based on the comparison between the received input data with reference input data. In another example, the relying party or server may be configured to generate an outcome of unsuccessful authentication when it results in an unsuccessful match based on the comparison between the received input data with reference input data. Upon determination of an unsuccessful authentication, the relying party or server may be configured to re-authenticate the input data up to and including a predetermined number of attempts before successfully authenticating the input data so as to complete the authentication request, or aborting the completion of the authentication request. In this manner, sequence diagram 400 of authentication process may implement with distributed storage, cloud-based storage, and other forms of storage in support of the aforementioned functionalities.

**[00098]** Figure 5 depicts a method of 500 of authentication according to an exemplary embodiment. Figure 5 may reference the same or similar components of system 100, first device 200 of FIG. 2A and FIG. 2B, method 300 of FIG. 3, and sequence diagram 400 of FIG. 4.

**[00099]** At block 505, the method 500 may include receiving, at a processor, a first instruction. The processor may be coupled to a memory comprising an application which includes instructions for execution on a first device, such as a card. In some examples, the first instruction may be associated with completion of an authentication request. For example, the authentication request may comprise a request for Fast Identity Online 2 (FIDO2) website registration. The first instruction may be a part of one or more instructions that are generated and/or transmitted by a processor of an authenticator, such as an application comprising instructions for execution on a second device, such as a client device or mobile device. For example, the processor of the authenticator may be configured to generate a first instruction. Each instruction may include one or more requests. The first instruction may include a first request to generate or retrieve a first key from the processor of a the first device. The key may comprise a FIDO key, such as a FIDO private key.

**[000100]** The first instruction may include a request to retrieve a first Fast Identity Online (FIDO) key from a first device, including but not limited to a card. The first instruction may include a request to generate the first FIDO key by the first device. In some examples, the first FIDO key may be generated based on a master key and an identifier associated with the first request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce a key pair. In some examples, the master key be stored on the first device, and the master key may be combined with the site identifier on the first device to generate the private key. In some examples, rather than generating or storing the FIDO private key in the first device, the first FIDO key may be generated or stored in a secure element belonging to an authenticator or second device, such as a client device that may include a mobile device. For

example, the FIDO private key may be stored in a secure element maintained by the second device. In some examples, the secure element may comprise a tamper-resistant secure storage area in which one or more keys may be securely stored and retrieved by the authenticator or relying party server. For example, the authenticator may be configured to transmit the first instruction to the first device, such that the authenticator may be configured to act as an intermediary device between the first device and the relying party or server.

**[000101]** At block 510, the method 500 may include generating a first key. For example, the first key may be retrieved via entry of a communication interface of the second device into a communication field of the authenticator or client device, such as a mobile device. In some examples, the first instruction may include a request to generate the first key by the first device. In other examples, the first instruction may include a request to regenerate the first FIDO key, such as a private key which is part of a public/private key pair, by the first device. The authenticator, such as the processor of the mobile device, may be configured to prompt one or more entries of the first device. For example, the one or more entries may include at least one selected from the group of a tap, a swipe, a wave, or the like, and/or any combination thereof. Thus, a communication interface of the first device may enter a communication field of the authenticator, such as the communication field of the mobile device. The first device may be configured to, based on the first instruction received from the authenticator, generate or retrieve a FIDO key pair associated with a specific user or site. The processor of the first device may be configured to receive the first instruction from the authenticator via a first entry and transmit the FIDO private key to the authenticator via a second entry of the communication interface of the first device into a communication field of the authenticator. In some examples, the one or more entries may be

associated with one or more gestures, including but not limited to one or more taps, swipes, waves, and/or any combination thereof.

**[000102]** In some examples, the FIDO private key may be generated based on a master key and an identifier associated with the authentication request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce another key pair. The first device may be configured to store a finite number of keys in its memory so as to avoid having to regenerate keys. In other examples, the first instruction from the authenticator may include a request to regenerate the FIDO private key, such as a private key which is part of a public/private key pair, by the first device.

**[000103]** In some examples, the master key be stored in memory on the first device, and the master key may be combined with the site identifier on the first device to generate the FIDO private key. In other examples, the master key may be transferred from the first device to the authenticator, in which the combining of the master key and the site identifier is performed by the authenticator to generate the FIDO private key.

**[000104]** In some examples, rather than generating or storing the FIDO private key in the first device, the FIDO private key may be generated or stored in a secure element belonging to the authenticator. For example, the FIDO private key may be stored in a secure element maintained by a second device.

**[000105]** At block 515, the method 500 may include encrypting the first key. For example, the FIDO private key may be encrypted by the first device prior to its transfer to the authenticator. Thus, the processor of the mobile device may be configured to decrypt the received encrypted FIDO private key prior to signing one or more challenges.

**[000106]** At block 520, the method 500 may include transmitting the encrypted first key for validation. For example, the processor of the first device may be configured to transmit the encrypted FIDO private key to the authenticator for validation. The generated or retrieved FIDO key pair may be read by the authenticator, including but not limited to via near field communication (NFC). The FIDO key pair may include a FIDO private key which may be read by the authenticator from the first device. The FIDO private key may be retrieved via one or more entries of a communication interface of the first device into a communication field of the authenticator.

**[000107]** In some examples, transmission of the encrypted FIDO private key from the first device to the authenticator may be prevented. For example, over the air (OTA) transmission of the FIDO private key is avoided when the first device is configured to perform signing of a first challenge. Thus, the first challenge and the site or user identifier information is transmitted to the first device from the authenticator, which thereby distributes its computation to the first device through near field communication (NFC) via a proxy protocol.

**[000108]** In some examples, the FIDO private key may be transmitted and received via one or more channels. For example, the FIDO private key may be transmitted and received via an out-of-band channel. In some examples, the authenticator may be configured to receive the FIDO private key from the first device via an out-of-band channel. The authenticator may be configured to transmit the first FIDO key to the processor of the relying party for validation, and thus the authenticator may be configured to serve as an intermediary device between the first device and the processor of the relying party.

**[000109]** In some examples, the relying party may be configured to generate and/or transmit one

or more second instructions. For example, the second instruction may include a request to transmit input data. The authenticator, such as the processor of the second device, may be configured to receive the second instruction. For example, the processor of the second device may be configured to receive the request to transmit input data from the relying party or processor of the server. In some examples, the second instruction may be forwarded by the relying party or processor of the server to the authenticator, such as the processor of the second device, via one or more push notifications. In some examples, the second instruction may be received by the authenticator after evaluation of one or more conditions by the relying party or server. For example, at least one of the one or more conditions may include determining a threshold number of authentication requests over a predetermined time period. For example, the relying party may be configured to determine whether an abnormal number of transactions or requests were performed within any number of seconds, minutes, hours, days, weeks, months, years, etc. In another example, at least one of the one or more conditions may include determining whether misuse or fraud associated with an account and/or user occurred. For example, the relying party may be configured to determine whether a user transaction history is indicative of excessive amount of purchases or abnormal locations. In this manner, conditional multifactor authentication may be implemented to improve security, which may trigger transmittal of the second instruction for receiving the input data.

**[000110]** Figure 6 depicts an authentication method 600 according to an exemplary embodiment. Figure 6 may reference the same or similar components of system 100, first device 200 of FIG. 2A and FIG. 2B, method 300 of FIG. 3, sequence diagram 400 of FIG. 4, and method 500 of FIG. 5.

**[000111]** At block 610, the method 600 may include receiving, at a processor, an authentication

request. For example, the processor may belong to a relying party or server. In some examples, the authentication request may comprise a request for Fast Identity Online 2 (FIDO2) website registration.

**[000112]** At block 620, the method 600 may include transmitting, by the processor, a first challenge. For example, the processor may be configured to transmit the first challenge to an authenticator, such as an application comprising instructions for execution on an intermediary second device or client device. In some examples, the processor may be configured to generate, responsive to the authentication request, one or more challenges. For example, the processor may be configured to generate a first challenge. In some examples, the relying party may be configured to challenge the authenticator, such as a processor of a mobile device, to login. The first challenge may include an identifier, such as a user identifier or a site identifier, that may be used to select an appropriate FIDO key pair. The first challenge may further include an unpredictable number, which may be given by the relying party, that is used to prevent replay. For example, for each instance of authentication, a new unpredictable number will be needed and thus the unpredictable number will be different each time. In this manner, this avoids utilizing a signature of the old unpredictable number and instead utilizes the number for the instant session of authentication.

**[000113]** At block 630, the method 600 may include validating the first challenge. For example, a first instruction may be transmitted to a first device by the authenticator. In some examples, the first instruction may be generated and/or transmitted by the authenticator to a processor of a card. The authenticator, such as the processor of the mobile device, may be configured to generate one or more instructions. For example, the processor of the mobile device may be configured to generate a first instruction. Each instruction may include one or more requests. The first instruction

may include a first request to retrieve a first key from the processor of a first device, such as a card. The key may comprise a FIDO key. The FIDO private key may be retrieved via entry of a communication interface of the first device into a communication field, such as the communication field of the authenticator. In some examples, the first instruction may include a request to generate the FIDO private key by the processor of the first device. In other examples, the first instruction may include a request to regenerate the first FIDO key, such as the FIDO private key which is part of a public/private key pair, by the processor of the first device.

**[000114]** For example, the authenticator may be configured to prompt, based on the first challenge received from the relying party or processor of the server, one or more entries of a first device by generating one or more instructions. For example, the one or more entries may include at least one selected from the group of a tap, a swipe, a wave, or the like, and/or any combination thereof. The first instruction may include a request to generate the first FIDO key by the first device. In some examples, the first FIDO key may be generated based on a master key and an identifier associated with the first request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce a key pair. In some examples, the master key be stored on the first device, and the master key may be combined with the site identifier on the first device to generate the private key.

**[000115]** For example, thee processor of the first device may be configured to generate and transmit the first FIDO key. For example, a communication interface of the first device may enter a communication field of the authenticator, such as the communication field of the mobile device, to transfer the first FIDO key. The first device may be configured to, depending on the first instruction received from the authenticator, generate or retrieve a FIDO key pair associated with a

specific user or site. The generated or retrieved FIDO key pair may be read by the authenticator. The FIDO key pair may include a FIDO private key which may be read by the authenticator from the first device and may be used by the authenticator to sign the first challenge resulting in a signed challenge data. The FIDO private key may be retrieved via entry of a communication interface of the first device into a communication field. In some examples, the FIDO private key may be encrypted by the first device prior to its transfer to the authenticator, in which case the processor of the mobile device may be configured to decrypt the received encrypted FIDO private key.

**[000116]** In some examples, the FIDO private key may be generated based on a master key and an identifier associated with the authentication request using one or more cryptographic algorithms. For example, the identifier may include a site identifier and combined with the master key to produce another key pair. The first device may be configured to store a finite number of keys in its memory so as to avoid having to regenerate keys. In other examples, the first instruction from the authenticator may include a request to regenerate the FIDO private key, such as a private key which is part of a public/private key pair, by the first device.

**[000117]** In some examples, the master key be stored in memory on the first device, and the master key may be combined with the site identifier on the first device to generate the FIDO private key. In other examples, the master key may be transferred from the first device to the authenticator, in which the combining of the master key and the site identifier is performed by the authenticator to generate the FIDO private key.

**[000118]** In some examples, transmission of the FIDO private key from the first device to the authenticator may be prevented. For example, over the air (OTA) transmission of the FIDO private key is avoided when the first device performs the signing of the first challenge. Thus, the first

challenge and the site or user identifier information is transmitted to the first device from the authenticator, which thereby distributes its computation to the first device through near field communication (NFC) via a proxy protocol.

**[000119]** In some examples, rather than generating or storing the FIDO private key in the first device, the FIDO private key may be generated or stored in a secure element belonging to the authenticator. For example, the FIDO private key may be stored in a secure element maintained by a second device. In some examples, the secure element may comprise a tamper-resistant secure storage area in which one or more keys may be securely stored and retrieved by the relying party.

**[000120]** In some examples, the processor of the first device may be configured to generate the FIDO private key. The processor of the first device may be configured to transmit the FIDO private key. For example, the processor of the first device may be configured to transmit the FIDO private key to the authenticator, such as an application comprising instructions for execution on a second device. As previously explained above, the authenticator may be, without limitation, a mobile device such as a laptop or a tablet or a phone. In some examples, the FIDO private key may be transmitted and received via one or more channels. For example, the FIDO private key may be transmitted and received via an out-of-band channel. The processor of the first device may be configured to transmit the FIDO private key to the authenticator via entry of the communication interface of the first device into a communication field of the authenticator. In some examples, the entry may be associated with one or more gestures, including but not limited to one or more taps, swipes, waves, and/or any combination thereof.

**[000121]** In some examples, the authenticator may be configured to receive the first FIDO key. For example, the application of the second device may be configured to receive the first FIDO

private key from the processor of the first device. The authenticator may be configured to sign the first challenge using the received first FIDO key. The authenticator may be configured to transmit the signed first challenge to the relying party for validation. The processor of the relying party or server may be configured to validate the signed first challenge. For example, the processor of the relying party or server may be configured to receive the signed first challenge from the authenticator. For example, the relying party may validate the signed challenge data with the FIDO public key that was stored when the user registered to the site. The validation may comprise the result of the authentication process, which includes signature verification. The authenticator may be configured to transmit the first FIDO key to the processor of the relying party for validation, and thus the authenticator may be configured to serve as an intermediary device between the first device and the processor of the relying party.

**[000122]** At block 640, the method 600 may include evaluating, by the processor, one or more conditions. For example, at least one of the one or more conditions may include determining a threshold number of authentication requests over a predetermined time period. For example, the processor may be configured to determine whether an abnormal number of transactions or requests were performed within any number of seconds, minutes, hours, days, weeks, months, years, etc. In another example, at least one of the one or more conditions may include determining whether misuse or fraud associated with an account and/or user occurred. For example, the processor may be configured to determine whether a user transaction history is indicative of excessive amount of purchases or abnormal locations. In this manner, conditional multifactor authentication may be implemented to improve security, which may trigger transmittal of the second challenge for receiving the input data.

**[000123]** At block 650, the method 600 may include transmitting, by the processor, a second challenge. For example, the processor of the relying party or server may be configured to generate a second challenge. For example, the second challenge may include a second request to transmit input data. In some examples, the processor of the relying party or server may be configured to evaluate one or more conditions, as previously explained. In some examples, the second challenge may be transmitted by the processor after evaluation of one or more conditions.

**[000124]** In some examples, the authenticator may be configured to receive the second challenge. For example, the processor of the mobile device may be configured to receive, after evaluation of the one or more conditions by the relying party or server, the second challenge from the processor of the relying party or server. In some examples, the second challenge may be forwarded by the processor of the relying party or server to the authenticator, such as the application comprising instructions for execution of the mobile device via one or more push notifications.

**[000125]** The authenticator may be configured to transmit, based on the second challenge, the input data. For example, the processor or application of the mobile device may be configured to transmit the input data to the relying party, such as the processor of the server. The input data may include at least one selected from the group of biometric data and credential data. For example, the input data may include biometric data, credential data, and/or any combination thereof. Without limitation, the biometric data may include at least one selected from the group of a fingerprint, a facial scan, a retinal scan, a voice recognition, and/or any combination thereof. In some examples, the input data may additionally and/or alternatively include credential data. Without limitation, the login data may include at least one selected from the group of an entry of a user name, a password, an account number, a security code, a one-time passcode, an answer to a

security question, and/or any combination thereof.

**[000126]** At block 660, the method 600 may include authenticating, by the processor, received input data. For example, the processor of the relying party or server may be configured to complete the authentication request by authenticating the input data. For example, the processor of the relying party or server may be configured to receive the input data from the authenticator, such as the processor or application comprising instructions for execution of the mobile device. In some examples, the processor of the relying party or server may be configured to generate one or more outcomes by comparing the received input data with reference input data. In some examples, the reference input data may be stored by the relying party or server. In other examples, the reference input data may be requested by the relying party or server. For example, the processor of the relying party or server may be configured to receive, via one or more requests, reference input data, or alternatively transmit the input data for comparison with the reference input data to a database. For example, the relying party or server may be configured to generate an outcome indicative of successful authentication when it results in a successful match based on the comparison between the received input data with reference input data. In another example, the relying party or server may be configured to generate an outcome of unsuccessful authentication when it results in an unsuccessful match based on the comparison between the received input data with reference input data. Upon determination of an unsuccessful authentication, the relying party or server may be configured to re-authenticate the input data up to and including a predetermined number of attempts before successfully authenticating the input data so as to complete the authentication request, or aborting the completion of the authentication request. In this manner, sequence diagram 400 of authentication process may implement with distributed storage, cloud-

based storage, and other forms of storage in support of the aforementioned functionalities.

**[000127]** It is further noted that the systems and methods described herein may be tangibly embodied in one of more physical media, such as, but not limited to, a compact disc (CD), a digital versatile disc (DVD), a floppy disk, a hard drive, read only memory (ROM), random access memory (RAM), as well as other physical media capable of data storage. For example, data storage may include random access memory (RAM) and read only memory (ROM), which may be configured to access and store data and information and computer program instructions. Data storage may also include storage media or other suitable type of memory (e.g., such as, for example, RAM, ROM, programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), magnetic disks, optical disks, floppy disks, hard disks, removable cartridges, flash drives, any type of tangible and non-transitory storage medium), where the files that comprise an operating system, application programs including, for example, web browser application, email application and/or other applications, and data files may be stored. The data storage of the network-enabled computer systems may include electronic information, files, and documents stored in various ways, including, for example, a flat file, indexed file, hierarchical database, relational database, such as a database created and maintained with software from, for example, Oracle® Corporation, Microsoft® Excel file, Microsoft® Access file, a solid state storage device, which may include a flash array, a hybrid array, or a server-side product, enterprise storage, which may include online or cloud storage, or any other storage mechanism. Moreover, the figures illustrate various components (e.g., servers, computers, processors, etc.) separately. The functions described as being performed at various components may be performed at other components, and the various

components may be combined or separated. Other modifications also may be made.

**[000128]** In the preceding specification, various embodiments have been described with references to the accompanying drawings. It will, however, be evident that various modifications and changes may be made thereto, and additional embodiments may be implemented, without departing from the broader scope of the invention as set forth in the claims that follow. The specification and drawings are accordingly to be regarded as an illustrative rather than restrictive sense.

**WE CLAIM:**

1. An authenticator, comprising:

a memory and a processor, wherein the processor is configured to:

receive one or more challenges;

generate a first instruction, the first instruction including a request to retrieve a first Fast Identity Online (FIDO) key;

transmit the first instruction;

receive the first FIDO key;

sign the one or more challenges using the first FIDO key; and

transmit one or more signed challenges for validation using a second FIDO key.

2. The authenticator of claim 1, wherein the first instruction includes a request to generate the first FIDO key.

3. The authenticator of claim 2, wherein the first key is generated based on a master key and an identifier associated with an authentication request using one or more cryptographic algorithms.

4. The authenticator of claim 1, wherein the first FIDO key is transmitted via an out-of-band channel.

5. The authenticator of claim 1, wherein the first instruction includes a request to regenerate the first FIDO key.

6. The authenticator of claim 1, wherein the one or more challenges are received based on an authentication request for FIDO website registration.

7. The authenticator of claim 1, wherein the processor is further configured to receive input data including at least one selected from the group of biometric data and credential data.

8. The authenticator of claim 1, wherein the first FIDO key is retrieved via entry of a communication interface into a communication field.

9. The authenticator of claim 7, wherein the input data is transmitted after determination of one or more conditions.

10. A method of authentication, the method comprising the steps of:

receiving, at a processor, one or more challenges;

generating, by the processor, a first instruction including a request to retrieve a first Fast Identity Online (FIDO) key;

transmitting, by the processor, the first instruction;

receiving the first FIDO key;

signing, by the processor, the one or more challenges using the first FIDO key; and

transmitting, by the processor, one or more signed challenges for validation using a second FIDO key.

11. The method of claim 10, wherein the first instruction includes a request to generate the first FIDO key.

12. The method of claim 11, wherein the first FIDO key is generated based on a third key and an identifier associated with an authentication request using one or more cryptographic algorithms.

13. The method of claim 10, further comprising transmitting the first FIDO key via an out-of-band channel.

14. The method of claim 10, wherein the first instruction includes a request to regenerate the first FIDO key.

15. The method of claim 10, wherein the one or more challenges are received based on an authentication request for FIDO website registration.

16. The method of claim 10, further comprising receiving, at the processor, input data including at least one selected from the group of biometric data and credential data.

17. The method of claim 10, wherein the first FIDO key is retrieved via entry of a communication interface into a communication field.

18. The method of claim 16, wherein the input data is transmitted after determination of one or more conditions.

19. The method of claim 18, wherein at least one condition comprises determining a threshold number of authentication requests over a predetermined time period.

20. A computer readable non-transitory medium comprising computer executable instructions that, when executed on a processor, perform procedures comprising the steps of:

receiving one or more challenges;

generating a first instruction including a request to retrieve a first Fast Identity Online (FIDO) key;

transmitting the first instruction;

receiving the first FIDO key;

signing the one or more challenges using the first FIDO key; and

transmitting one or more signed challenges for validation using a second FIDO key.