

**SYSTEM AND METHOD TO PERFORM DIGITAL AUTHENTICATION USING
MULTIPLE CHANNELS OF COMMUNICATION**

RELATED APPLICATIONS

[0001] This application claims priority to U.S. Patent Application Serial No. 17/379,537 , entitled “SYSTEM AND METHOD TO PERFORM DIGITAL AUTHENTICATION USING MULTIPLE CHANNELS OF COMMUNICATION ” filed on July 19, 2021. The contents of the aforementioned application are incorporated herein by reference in their entirety.

BACKGROUND

[0002] Today’s banks provide a number of services online through the Internet. For example, a customer may download a banking application on their mobile device and interface with a banking system to perform tasks, such as check their account balance, change their personal information, update their settings, pay bills, and so forth. However, current banking systems require users to setup an online account with personal information to login into to perform these tasks. However, a subset of customers typically performs their banking tasks using traditional methods, e.g., by going to the bank and speaking with a teller in person or calling a banking representative on the phone. The experience for these customers is typically more inconvenient. One downside is that the customers are still required to go through an authentication process that generally requires the user to provide some piece of information to be authenticated. For example, current systems and methods for authenticating a customer include requesting sensitive data from the customer, such as an account number, a transaction card number, a social security number, a mother's maiden name, a password, and/or other personal data. Because certain information may be known by fraudsters, “something you know” authentication techniques force obscure questions such as “What is your grandfather's middle name?” Also, if customers forget the answers to certain questions such as “Who was your favorite teacher?” the customer could be prevented from performing a task. Current authentication processes, therefore, are not only burdensome for customers but also time-consuming and costly for companies providing customer service to these customers. These and other drawbacks exist.

BRIEF SUMMARY

[0003] Embodiments may be generally directed to systems, devices, and techniques including a system configured to authenticate unauthenticated users to perform sensitive actions. The system may include one or more processors, and memory coupled with the one or more processors. The memory is configured store instructions that when executed by the one or more processors, cause the one or more processors to communicate a first communication to a mobile device, the first communication comprising first data to cause the mobile device to download a mobile application, communicate a second communication to the mobile device, the second communication comprising second data comprising a task identifier, the second data to cause the mobile application to perform an authentication task associated with the task identifier, and determine the user is authenticated by the authentication task. In response to the user being authenticated the one or more processors may enable the mobile device to perform sensitive actions.

[0004] Embodiments may be generally directed to systems, devices, and techniques including a computing device configured to authenticate unauthenticated users to perform sensitive actions. The computing device may include processing circuitry, and memory coupled with the processing circuitry. The memory is configured store instructions that when executed by the processing circuitry, cause the processing circuitry to receive a first communication from a system to authenticate a user, the first communication comprising first data to install a mobile application, install the mobile application based on invocation of the first data, receive a second communication from the system, the second communication comprising second data comprising a task identifier, the second data to cause the mobile application to perform an authentication task associated with the task identifier, initiate the authentication task based on invocation of the second data to determine the user is authenticate, and receive an indication that the user is authenticated based on performance of the authentication task, and the user is enabled to perform sensitive actions.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0005] To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced.

[0006] FIG. 1 illustrates an example of a system 100 in accordance with embodiments.

[0007] FIG. 2 illustrates an example configure of a computing device 102 and a system 104 in accordance with embodiments.

[0008] FIG. 3 illustrates an example of sequence flow 300 in accordance with embodiments.

[0009] FIG. 4 illustrates a routine 400 in accordance with one embodiment.

[0010] FIG. 5 illustrates a routine 500 in accordance with one embodiment.

[0011] FIG. 6A illustrates an example of a sequence flow 600 in accordance with one embodiment.

[0012] FIG. 6B illustrates a continuation of the sequence flow 600 in accordance with one embodiment.

[0013] FIG. 7 illustrates a contactless card 700 in accordance with one embodiment.

[0014] FIG. 8 illustrates a transaction card component 800 in accordance with one embodiment.

[0015] FIG. 9 illustrates a sequence flow 900 in accordance with one embodiment.

[0016] FIG. 10 is a diagram of a key system 1000 according to an example embodiment.

[0017] FIG. 11 is a flowchart of a method 1100 of generating a cryptogram according to an example embodiment.

[0018] FIG. 12 illustrates a flowchart of a method 1200 in accordance with embodiments.

[0019] FIG. 13 illustrates a computer architecture 1300 in accordance with embodiments.

DETAILED DESCRIPTION

[0020] Today's technology makes it easier for users and customers to take advantage of many different services in a secure manner. For example, most people have mobile devices configured with different types of mobile applications that allow users to perform many daily tasks. Users may communicate with their friends, family, and work colleagues through applications, such as text messaging applications, phone applications, video calling applications, email applications, and so forth. Users may also spend their leisure time playing games on one of the multitudes of available gaming applications. Mobile devices, when configured correctly, also provide a secure platform to enable users to perform more sensitive tasks, such as interfacing with their bank through a banking application, speaking with their doctor through a medical application, making investments with their broker through their brokerage account application, and so forth. However, these more secure applications typically

need to be linked to the service provider's system through a secure interface and require an established level of trust with the customers to ensure that they are authorized users to perform the task. Typically, a service provider will have a user establish an account with a single sign-on username and password that must be correctly entered into the application each time a user wants to perform a sensitive task, e.g., review/change personal information, perform an action with their bank account, discuss private medical information, etc.

[0021] However, for any number of reasons, a subset of these users never establishes an account and SSO with the service provider. Therefore, these users cannot fully utilize the features of the mobile application to perform the more sensitive tasks. In some instances, the setup process may be too confusing or take too long. Some other users may trust performing these tasks in-person or speaking to another person on the phone or in a chat window. However, these users are generally difficult to authenticate in a quick, secure, and effective manner. Embodiments discussed solve these issues by utilizing a communication channel other than the voice or chat channel established with the customer to authenticate the customer.

[0022] Specifically, embodiments may be directed to systems, devices, and techniques to perform authentication operations for unauthenticated users. Embodiments include providing a mobile application associated with the service provider and then performing an authentication operation with the mobile application to authenticate the user. In one example, a system, such as a banking system or other service-providing system, may generate and communicate a first communication to a user's device in response to the user wanting to perform a sensitive task. The first communication includes an embedded link that may cause the mobile device to download and install a mobile application associated with the system. The system may generate and send a second communication to the device. The second communication may include an embedded link to cause the application to launch or execute on the device. The second communication may also include data, such as a task identifier, that may be provided to the application to locate and execute the instructions to perform the authentication operation.

[0023] Specifically, the application on the mobile device utilizes the task identifier to identify a task to perform for the user, e.g., the authentication operation to authenticate the user. The authentication operation may include the user providing a token or other authentication information that may be authenticated by the system or another authentication server or service. Upon authentication, the application may be enabled to perform the sensitive task for the customer. These and other details will be discussed more fully in the following description.

[0024] FIG. 1 illustrates a schematic of an exemplary system 100, consistent with disclosed embodiments. As shown, system 100 includes a computing device 102 and a system 104 that may communicate with each other via network 106 (e.g., via the Internet). The network 106 may include one or more wireless and/or wired interconnects configured to enable communication between devices, such as the computing device 102 and the system 104. The computing device 102 is representative of any type of network-enabled computing devices, such as mobile devices, mobile phones, smartphones, tablet computers, wearable devices, laptops, portable gaming devices, and the like. Further, FIG. 1 illustrates one computing device 102; however, embodiments are not limited in this manner. In practice, system 100 may include one or more computing device 102, which can communicate with system 104. The system 104 may include one or more servers having processing circuitry (processors), memory, storage, interfaces, etc. System 104 is representative of any type of computing system, such as a server farm, workstation, compute cluster, cloud computing platform, a virtualized computing system, and the like.

[0025] FIG. 2 illustrates examples details of components of the computing device 102 and the system 104 to perform the operations discussed herein. Note that FIG. 2 only illustrates a limited number of components for illustrative purposes and computing device 102 and system 104 can include other components, devices, circuitry, controllers, networking devices, memory, etc. As shown, the computing device 102 may include processor(s) 202, interfaces 204, and memory 206. In some instances, the computing device 102 may include more than one processor 202, each having one or more cores. The processor(s) 202 may be coupled with the interfaces 204 and the memory 206. The memory 206 may be any type of memory, including volatile and non-volatile memory configured to store instructions for processing by the processor(s) 202. The interfaces 204 may include wired and wireless networking interfaces configured to communicate to other devices including computing device 102. Moreover, the interfaces 204 may be configured to communicate in accordance with one or more protocols, such as cellular communication protocols (e.g., WiMAX/LTE/5G), local area network protocols (e.g., 102.11 series), and short-range communication protocols (e.g., near-field communication (NFC)/Bluetooth).

[0026] Further, the computing device 102 includes storage 208 such as a hard disk drive, a solid-state drive, or the like. The storage 208 is configured to store software, such as

application(s) 210 and an operating system 212. Examples of an operating system 212 include the Android® OS, iOS®, macOS®, Linux®, and Windows® operating systems.

[0027] The application(s) 210 may be referred to as mobile applications, mobile apps, apps, applets, or the like. The application(s) 210 includes communication applications, such as a texting application, a web browser application, a telephone application, an electronic mail (e-mail) application, a video chat application, a text chat application, and so forth. The texting application may be configured to send and receive data and messages in accordance with one or more message protocols (e.g., short message service (SMS), rich communication services (RCS), and multimedia messaging service (MMS)). Other applications of the computing device 102 may include gaming applications, multimedia applications, camera or image applications, and so forth. In computing device 102 may also include applications to access services. For example, service provider applications a banking application, a stock management application, a mobile payment service application, human resource application, a medical application, and so forth. In embodiments, the computing device 102 enables users to delete application, update applications, and download and install new applications. For example, the computing device 102 have an application store to control the applications on the computing device 102.

Examples of an application store include the Google® Play Store, the Apple® App Store, the Amazon® Apps & Games Store, etc. In embodiments, the applications 210 enable users of the computing device 102 to interact with the device and perform actions. For example, users may utilize the application(s) 210 to send and receive emails, send and receive text messages, play games, review/edit documents, send and receive money, make phone calls, and so forth. In general, each of the application(s) 210 may provide a service or feature for a user.

[0028] In embodiments, the users can download and install one or more of the application(s) 210 to conduct or perform sensitive actions or high-risk actions. These applications may be provided by service providers, and interface with the service provider's system. A sensitive action may be any action or service that requires the user to be authenticated to perform. A sensitive action may be an action that will enable the user to view, change, and/or interact with private/personal information, financial information, health information, sensitive corporate information, etc. Examples may include providing personal information to the user, enabling the user to wire or send money to another person, applying for a loan, perform checking/saving account operations, and so forth. Embodiments discussed herein enable users that typically do

not take advantage of these services provide through applications 210 to be quickly and easily be authorized.

[0029] As previously discussed, typically users configure an account with a username/password such that they can log into a secure area of a website or application to perform the sensitive actions. However, some users never configure or establish an account or username/password and are unauthenticated. Thus, when these users want to perform a sensitive action, they are not trusted until they go through an authentication operation. Embodiments discussed herein enable system 104 to authenticate users via computing devices 102 and based on a trusted piece of data or information, e.g., providing a token from a contactless card, verifying government-issued identification information, providing a biometric, etc. System 104 first determines that a customer is not authenticated and then performs operations to authenticate the customer, including installing application on the customer's computing device 102 to perform the authentication, causing an authentication operation to be performed, and enabling the user to perform the sensitive task if they are authenticated.

[0030] In embodiments, the system 104 may be a sub-system of the service providers system. For example, the system 104 may be part of a call or help center operated by the service provider and is setup to handle calls and chats to help customers perform the various actions. In one example, the system 104 may enable human or computer operators to speak or chat with users and to perform authentication operations discussed herein including enabling the users to perform sensitive actions. The system 104 may include one or more devices, such as servers having one or more processor(s) 214, interfaces 216, and memory 218. The system 104 also includes storage 220 that may store applications and/or instructions to perform the operations discussed herein.

[0031] The storage 220 may include operation services 222, configuration services 224, and authentication services 226. The operation services 222 include instructions configured to enable operators to interface with customers and provide services. The configuration services 224 include instructions configured to facilitate the configuration of the computing devices for users to install an application and perform authentication operations via the application. The authentication services 226 include instructions to perform authentication operations and/or process the results of authentication operations for the system 104.

[0032] In embodiments, the operation services 222 provides a graphical user interface (GUI) for an operator to interact and communicate with users and computing devices. Moreover, the

operation services 222 provide an interface for the users to communicate with the operators. For example, the operation services 222 may include instructions to handle and process phone calls and chat sessions. The operations services 222 may be configured to receive a phone call or chat initiation and direct the call or chat to an operator.

[0033] The operations services 222 may also enable the operator to communicate with the users, e.g., send communications (voice or data) back and forth between the system 104 and computing devices 102. In some embodiments, at least a portion of the operation services 222 may be automated to automatically communicate with users either via voice communication channels or data communication channels. For example, the operation services 222 may automatically detect requests made by users, generate responses based on the requests, and communicate the responses back to the users. The responses may be interactive responses and are responsive based on the requests made by the users. The operation services 222 may include instructions that are configured to convert the user's spoken word into text and then analyze the text to determine a response. The instructions may also be configured to generate responses based on analyzed text and communicate the response verbally on the phone or through chat in a chat window.

[0034] In embodiments, the operation services 222 may also be configured to determine whether users are authenticated to perform sensitive actions. For example, the operations services 222 may provide an interface for an operator to search and/or perform a lookup in a database to determine whether a particular is authenticated to perform a sensitive and/or their computing device is configured to perform the authentication operations. In some embodiments, the operation services 222 includes instructions to automatically perform the lookup based on the analysis of the user's speech/text. Embodiments discussed herein facilitate instances when the user is not authenticated and their computing device 102 is not configured to perform authentication operation, e.g., the user does not have an account established with an SSO.

[0035] In one specific example, a user may call or initiate a chat to perform a sensitive task. The operation services 222 are configured to provide an operator or agent of the system 104 to interface with the user and to perform task. For example, the operation services 222 may perform a lookup in a database to determine if the user is authenticated to perform the sensitive task. The operator may perform the search through a GUI interface and determine if the customer has configured an account with the service provider and/or has an SSO established. In

some instances, one or more of the operations may be automated. For example, the operation services 222 may include instructions to automatically determine if the user is authenticated (e.g., by interpreting the action to be performed and automatically performing the lookup to determine if the user has an account). If the user is authenticated, the user may be permitted to perform the sensitive task. However, embodiments discussed herein aim to help users that are not unauthenticated with system 104 and/or do not have a trust relationship established with the service provider. The system 104 also includes the configuration services 224 and the authentication services 226 to authenticate the user and enable the user to perform the sensitive task.

[0036] The configuration services 224 include instructions to configure a computing device 102 of the user to authenticate the user. Specifically, the configuration services 224 generate and communicate information to the user's computing device 102 to install a mobile application associated with the service provide. In embodiments, the configuration services 224 may generate a text message including an indication to download the application. For example, the text message may include an embedded link that, when activated, causes the computing device 102 to download the application. The text message may include text, such as "Here's a link to install our application: <https://appstore/application>." A user may select the embedded link and automatically be directed to the application store to download the application. The embedded link may be a location to directly download the application (e.g., a home page to download the app). In some instances, the embedded link may be configured to cause the computing device 102 to automatically redirect to the application store without the user selecting the embedded link.

[0037] The text message may be communicated to the computing device 102 via a text message service, such as SMS, MMS, or RCS. In some instances, the configuration services 224 may automatically detect the phone number from which the user is calling and send the text message to the computing device 102. In other instances, the operator may ask the user for the phone number, the user may provide the number, and the operator may manually and/or automatically enter the provided number into the system 104 to send the message. In some embodiments, the configuration services 224 may apply voice-to-text instructions to the user's provided number and automatically send the message to the given phone number. Note that embodiments are not limited to utilizing text messages. In some instances, a quick response (QR) code including the embedded link may be generated by the configuration services 224 and

provided to the user on a display. The user may use a scanning application and a camera to scan the QR code, which may automatically read the embedded link and redirect to the application download store to download the application.

[0038] In embodiments, the system 104, including the configuration services 224, may determine that the application is installed on the computing device 102. For example, the user may tell the operator that the application is successfully installed, and the operator may input an indication into the system 104. In other instances, the configuration services 224 may receive an indication from the installed application that it is successfully installed. The indication may be provided to system 104 by the application communicating with the system 104 through an application programming interface (API), for example. Once the application is installed, the system 104 may initiate an authentication operation with the computing device 102 to authenticate the user. The authentication operation includes having the user provide a trusted token to the system 104 (or another authentication system), and authenticating the user, e.g., is the person that was previously authenticated with the token.

[0039] In embodiments, the system 104 may utilize the downloaded application to perform the authentication operation. For example, the system 104 may include authentication services 226 that may generate another message to cause the application installed on the computing device 102 to perform the authentication operation. The authentication services 226 may generate or determine a task identifier associated with performing the authentication operation. In some instances, the task identifier may be predetermined, and the application may include instructions to execute the authentication operation upon receiving the task identifier. The task identifier may be a combination of alphanumeric characters to identify the authentication operation.

[0040] In embodiments, the authentication services 226 may generate a text message, embed the task identifier in the text message and communicate the text message to the computing device 102 for the application to process. The text message may include an embedded link to cause the application to execute and provide the task identifier to the application. If the application is already executing on the computing device 102, the embed link may pass the task identifier to the application. In one specific example, the text message may include an embedded link such as "app.launch/task_identifier," that when activated or selected, will launch the application and provide the task identifier (task_identifier) to the application. The task identifier may be an element of a function that will execute when the application is

launched. The application may determine the task (e.g., authentication operation) corresponding to the task identifier and perform the task. Embodiments are not limited to embedding the task identifier in a text message, and the task identifier may also be embedded in a QR code. Again, the user may use QR scanning software to scan the code with a camera to receive the data, including the task identifier.

[0041] In embodiments, the authentication operation may execute on the computing device 102 to authenticate the user. For example, the application executing on the computing device 102 is configured to launch and execute instructions to authenticate the user based on the task identifier. The authentication operation includes the user providing a token or piece of information that can be used to authenticate the user. In one example, the service provider may be a bank, and the application may be a banking application configured to provide banking services for the user. The user may have a contactless card with the bank that can be used to authenticate the user. Specifically, the contactless card includes a token that can be provided to the authentication services 226, a different system, or authentication service to authenticate the user, e.g., a third-party system, a dedicated authentication server, or the like may authenticate the information or token. If a different system or service perform the authentication of the user, a result of the authentication operation may be communicating the application, the authentications services, or a combination therefor. Embodiments are not limited in this manner; however, in this example, the authentication services 226 may perform the authentication of the information or token.

[0042] In embodiments, the authentication services 226 may receive the token from the contactless card via the application on the computing device 102. The computing device 102 may read the token from the contactless card by performing one or more short-range communication read operations. The application may execute the authentication operation, causing a prompt to be displayed on the display of the computing device 102 for the user to provide the contactless card on the computing device 102, e.g., bring the contactless card within short-range communication operating range of the computing device 102. In one example, the computing device 102 may utilize NFC communications having an operating range of approximately 10 centimeters or less. In some embodiments, to ensure that the user brings the contactless card close enough to the computing device 102 to exchange data, the user is prompted to tap the card on the computing device 102, e.g., on the display or other surface of the computing device 102.

[0043] The computing device 102 receives a payload including a cryptogram from the contactless card. The cryptogram may include the token or authentication information, which may be one or more identification numbers, a counter value, a version number, and a shared secret, and be encrypted using diversified keys. The cryptogram may be generated, encrypted, and communicated by the contactless card to the computing device 102. In embodiments, the payload may include additional information, such as an unencrypted identifier. The unencrypted identifier may be a customer identifier to identify a customer associated with and stored on the contactless card. Embodiments are not limited in this manner.

[0044] The computing device 102 may communicate the cryptogram to the system 104 including the authentication services 226 via one or more secure communications (e.g., through one or more secure API communications). The authentication services 226 may decrypt the cryptogram from the contactless card and verify the token and/or authentication information matches stored verified data in the system 104. For example, the system 104 may determine one or more of the identification numbers, counter values, version number, and/or the shared secret match corresponding data in the system 104. If so, the system 104 verifies that the user is authenticated and can perform sensitive tasks. If not, the user is not authenticated and cannot perform the tasks.

[0045] In embodiments, alternative methods may be utilized by the system 104 to authenticate the user. For example, system 104 may utilize a photo verification system to authenticate the user. The application may prompt the user to take a photograph or picture of their government-issued (or other trusted) identification. The computing device 102 may communicate the picture to the system 104 to authenticate the user. The system 104 may use a photo ID verification service, such as BlueCheck®, or a similar service to verify the photo. Other alternative services may include verifying the user via a biometric input, such as a fingerprint, iris, facial picture, etc. Once the user is authenticated, the system 104 may enable the user to perform the sensitive operations. For example, the system 104 may set a flag or setting on the user's account that enables the user to perform tasks, such as wiring money, changing settings, paying bills, etc.

[0046] FIG. 3 illustrates an example sequence flow 300 that may be performed to authenticate an unauthenticated user to perform sensitive tasks. Examples discussed herein are in reference to system 100, and computing device 102 and system 104; however, the techniques discussed may be performed by any system to authenticate a user to perform high-risk or sensitive services. In the illustrated example, the computing device 102 may be a mobile device

configured to make telephone calls via voice channels and communicate data via data channels. The system 104 may include one or more servers configured to communicate with the computing device 102 via voice and/or data channels.

[0047] In embodiments, a user may communicate to an operator and the system 104 that they wish to perform a sensitive task. The communication may be made by the user speaking with the operator of the system 104 over a voice channel or through a chat window over a data channel. The system 104 may determine that the user is not authenticated. For example, the system 104 may determine that the user does not have an electronic online account configured to perform the sensitive task. Thus, the user has not previously provided identifying information to the service provider and system 104 that may be used to authenticate the user. For example, the user does not have a username/password configured to log in to an online account provided through a web browser or mobile application (e.g., a banking app). Moreover, the system 104 may determine that the user does not have the mobile application installed on the computing device 102 configured to perform the sensitive task, e.g., the user may provide the information to the operator.

[0048] At line 302, the system 104 generates a message to communicate to the 102, including information to download the application to perform the sensitive. In some instances, an operator may provide an input to system 104 through a GUI to generate the message to communicate to computing device 102. For example, the operator may be presented with a GUI interface configured to take a user's phone number as input to send the computing device 102 a text message. The phone number may be communicated to the operator via the voice communication channel. In other instances, the system 104 may generate the message automatically. For example, the system 104 may include voice-to-text instructions to convert the user's words to text. The text may be analyzed to determine that the user does not have the application installed on their computing device 102. In some instances, the information may be communicated to the system 104 through a chat or text-based interface and the system 104 may analyze the text to determine the user does not have the application installed on their computing device 102.

[0049] The system 104 may generate the message including an embedded link, that when selected or activated, causes the computing device 102 to redirect to a location to download the application. For example, the link may cause the operating system 212 to launch an application store to download the application. In some instances, the application store may execute and

open to the specific location to perform the download, and the download may occur automatically without human intervention. In another example, the embedded link may direct the operating system 212 to open a website in a web browser to a web-based location to download the application.

[0050] At line 304, the system 104 may communicate the message to the computing device 102. As mentioned, the message may be communicated via a text message service in accordance with a text message protocol based. The text message may include the embedded link which, as discussed, may be an address or link to a location, a QR code, or the like, that includes information to cause the operating system 212 to start downloading the application. In some instances, the message may be communicated to a device other than computing device 102, such as a personal computer, and then the computing device 102 may receive the message by performing a scanning operation utilizing a camera to read a QR code or the like. The message may be displayed in a web browser or other application on the other computing device and the user may use the camera of the computing device 102 to scan or take a picture of the message. Other options may include communicating the message to computing device 102 or another computing device using electronic mail or other message services, such as Teams® or Slack®.

[0051] At line 306, the computing device 102 may process the message, including the embedded link. For example, the user may select or interact with the embedded link, and the operating system 212 may process the selection and launch an application store or web browser to download the application. The operating system 212 may further download the application through the store or browser and install the application to execute on the computing device 102. In some instances, once the application is installed, the application may communicate back to system 104 an indication that it's installed. The indication may be communicated through an API and include information to specifically identify computing device 102, e.g., a phone number, device identifier, etc. In other instances, the user may verbally or through text-based chat communicate to the operator of system 104 that the application is successfully installed.

[0052] At line 308, the system 104 may determine a task identifier to identify an authentication task or operation to be performed by the application. In one example, the system 104 may perform a lookup in a database to determine the task identifier to identify the authentication operation to the application. The task identifier may be a combination of alphanumeric values that uniquely identifies the authentication operation. In some instances,

the task identifier may be a programmed variable for a function of the application. The function may be invoked, and the task identifier may be passed to the function as an element. The function may process the task identifier and perform the authentication operation.

[0053] At line 310, the system 104 may communicate the task identifier to the computing device 102. In one example, the system 104 may generate a second text message or QR code with an embedded link. The embedded link includes a link to launch the mobile application on the computing device 102 and the task identifier. The task identifier is passed to the application when the application is launched. As similarly discussed above, the user may select the embedded link to cause the application to launch. In other instances, the application may automatically launch when the text message is received.

[0054] In some instances, the task identifier may be passed directly to the application on the computing device 102. For example, the application may execute on the computing device 102 and determine that the user is not authenticated to perform operations with the application. The application may communicate to the system 104 through an API, an indication that the authentication operation needs to occur. The system 104 may communicate the task identifier to the computing device 102 through another API call.

[0055] At line 312, the computing device 102 may process the task identifier and perform the operation associated with the task identifier. Specifically, the application will initiate the operation (e.g., authentication operation) based on the task identifier on the computing device 102. The authentication operation includes the user providing a token or identifying information to the computing device 102. The computing device 102, including the application, will perform the authentication by communicating the information to system 104 or another authentication server to compare the information to trusted information. In one example, the user can provide a token on their contactless card 700 to the computing device 102, which may be forwarded to system 104 or another server to authenticate. The method 1100 of FIG. 11 illustrates one possible processing flow that may be performed by the contactless card 700 to generate a cryptogram including the token. FIG. 9 illustrates a sequence flow 900 where the cryptogram is communicated from the contactless card 700 to the computing device 102. In another example, the user may take a picture of a trusted identification (e.g., driver's license), and the picture may be authenticated. Other authentication methods may include authenticating a biometric sample, a trusted certificate, etc.

[0056] In the illustrated example, the computing device 102 may communicate authentication information to the system 104 at line 314 to perform as part of the authentication operation. As mentioned, the authentication information may include a cryptogram from a contactless card 700, a picture of a trusted identification, or other trusted information (biometric sample). The authentication information may be communicated to the system 104 via the application and through an API. The API may provide a secure and encrypted channel to communicate the data to the system 104.

[0057] At line 316, the system 104 performs authentication on the information from the computing device 102. For example, the system 104 may verify that the authentication information is accurate or correct based on a comparison with stored trusted information. If the authentication information matches the trusted information, the user is authenticated. If the authentication information does not match the trusted information, the user is not authenticated. FIG. 12 describes one possible method 1200 that may be performed to authenticate a user utilize a contactless card 700 and a server, e.g., system 104.

[0058] At line 318, the system 104 communicates a result of the authentication to the computing device 102. The result includes an indication as to whether the user is successfully authenticated or not authenticated. At line 320, the system 104 may enable or prevent the user from performing the sensitive task. For example, the system 104 may set a flag or an indication for the application of the computing device 102 to permit or deny the user to perform the sensitive task. The indication may be an API call or communication to the application.

[0059] FIG. 4 illustrates an example routine 400 that may be performed by a system, such as 104, to authenticate a user to perform sensitive actions. The operations discussed herein may be performed by software, hardware, or combination thereof.

[0060] In block 402, the routine 400 includes communicating the first communication to a mobile device. The first communication includes first data to cause the mobile device to download a mobile application. In embodiments, the first communication may be a text message or a QR code and the first data may be an embedded link, such as a Hypertext Transfer Protocol (HTTP) or HTTP secure (HTTPS) link that, when activated, causes the mobile device to go to a location to download a mobile application.

[0061] In block 404, the routine 400 includes communicating a second communication to the mobile device. The second communication includes second data comprising a task identifier, and the second data to cause the mobile application to perform an authentication task

associated with the task identifier. In embodiments, the second communication may also be a text message or QR code, and the second data may be an embedded link to launch the downloaded mobile application. The task identifier may be processed by the mobile application to execute an authentication task to authenticate a user of the mobile device.

[0062] In block 406, the routine 400 includes determining the user is authenticated by the authentication task. In embodiments, the mobile device, an authentication server, the system 104, or a combination thereof may perform the authentication task. One or more of the mobile device, the authentication server, and the system 104 may provide an indication of the result of the authentication task (e.g., authenticated or unauthenticated).

[0063] In block 408, the routine 400 includes enabling the user to perform sensitive actions via the mobile device. For example, the system may set a flag or setting to permit or deny the user from performing the sensitive actions. If the user is authenticated, the system may set a setting to permit the user to perform the sensitive actions, or vice versa.

[0064] FIG. 5 illustrates an example routine 500 that may be performed by a mobile device, such as computing device 102, to authenticate a user to perform sensitive actions. The operations discussed herein may be performed by software, hardware, or combination thereof.

[0065] In block 502, the routine 500 includes receiving a first communication from a system to authenticate a user. In embodiments, the first communication may be generated and communicated by the system based on a determination the mobile device does not have the mobile application to perform the authentication task. The mobile application may be a trusted application provided by the service provider associated with and controlling the system. In embodiments, the first communication includes the first data to install the mobile application. As previously discussed, the first communication may be a text message or a QR code, and the first data may be an embedded link, that when activated, downloads and installs the application on the mobile device.

[0066] In block 504, the routine 500 includes installing the mobile application based on the invocation of the first data. For example, a user of the mobile device may select the link via an interface, such as a touchscreen, and the operating system may be redirected to execute and open the application store to download the application. The link may cause the store to open to the location to directly download the application. In some instances, the first communication, including the first data may cause the operating system to redirect to the application store and

install the application automatically when the communication is received. For example, the operating system may automatically process the link upon reception without the user selecting.

[0067] In block 506, the routine 500 includes receiving a second communication from the system. In embodiments, the second communication includes the second data comprising a task identifier, and the second data may cause the mobile application to perform an authentication task associated with the task identifier. The second communication may also be a text message or QR code, and the second data may also be an embedded link. However, this embedded link may include data to cause the newly downloaded application to execute when it is selected or invoked. Again, this link may also execute based on a user selection or automatically upon reception of the message. The link may also include a task identifier to identify the authentication task that is required to be performed by the application. The identifier may be passed to the application when the link is selected or invoked. The application may initiate and process the task identifier.

[0068] In block 508, the routine 500 includes initiating the authentication task based on the invocation of the second data to determine the user is authenticated. For example, the mobile application utilizes the task identifier to identify instructions to perform the authentication task. The instructions may execute by the mobile device, which may cause the mobile device to perform authentication with one or more of an authentication server or system 104.

[0069] In block 510, the routine 500 includes receiving an indication that the user is authenticated based on the performance of the authentication task. For example, the mobile device may receive an indication from an authentication server or system 104 indicating that information sent to the server or system is authenticated and the user is authenticated. In embodiments, the indication may also indicate that the user capable of performing sensitive actions with the mobile application. The user may now perform the action they desired to perform with the mobile application.

[0070] FIG. 6A illustrates an example sequence flow 600 that may occur between a user and a customer service operator to authenticate a user such that they can perform sensitive in accordance with embodiments discussed herein. The items discussed in this sequence flow 600 may be performed by systems and devices discussed herein, including computing device 102 (e.g., mobile device) and system 104 (e.g., a banking system).

[0071] In the illustrated example, the customer may make a telephone to a service provider, such as a bank. The customer may want to perform a high-risk task using the service provider,

but the customer is not authenticated with the service provider, e.g., the customer is not enrolled with a single sign-on username and password. At 602, the customer may be notified that they are not authenticated to perform the high-risk task but can easily and quickly perform an authentication operation using a mobile application on a mobile device. At 604, the agent or operator communicates with the customer to obtain a phone number associated with the customer's phone to send a text message. The customer may verbally provide the number to the agent.

[0072] At 606, the agent may provide the phone number to a system (e.g., through a GUI) to send the text message to download the application to the customer's mobile phone. The system may generate a text message. The text message includes an embedded link (e.g., <https://appstore/app>). to download the application. At 608, the text message is communicated to the mobile device associated with the phone number. And at 610, the customer receives the text message on their mobile device. At 612, the customer may invoke or select the link in the message, and the message redirects the mobile device to download the application. At 612, the mobile application is downloaded on the mobile device.

[0073] At 614, the customer may communicate an indication that the mobile application is installed on the mobile device and is open. Further and at 616, the agent may confirm with the customer that the mobile application is installed on the mobile device and notify the customer that they are going to send another message. At 618, the agent may interact with a GUI interface to generate another message to communicate to the customer's mobile device. The system may create a task to perform on the mobile device. At 620, the system may determine a task identifier associated with the task. The task identifier may identify the task to the application and the system. The task identifier and link may be embedded in a message and sent to the mobile device at 622. And at block 624, the mobile device may receive the message from the system, including a link to cause the mobile application to execute or initiate with the task identifier.

[0074] In FIG. 6B, the sequence flow 600 continues; and at 626, the mobile application may initiate and execute on the customer's mobile device. In some instances, the customer may select the link to cause the mobile application to execute. In other instances, the mobile application may execute upon reception of the text message. The mobile application may open on the mobile device and present a GUI for the customer to interact with the application.

[0075] At 628, the mobile application evaluates the task identifier embedded in the link and/or provided in the text message. The mobile application may determine the task associated with the task identifier (e.g., perform the authentication operation). At 630, the mobile application may process the task identifier. For example, the mobile application may execute a function with the task identifier to cause the instructions associated with performing the authentication to execute on the mobile device.

[0076] At 634, the mobile application may execute and perform the task (e.g., the authentication operation). In the illustrated example, the customer has a contactless card associated with the service provider and can be authenticated based on the information on the contactless card. To perform authentication via the contactless card, the mobile device may retrieve the information from the card and send the information to a system to perform authentication with the contactless card information. At 632, the customer may be instructed to bring the card within a communication range of the mobile device. In one example, the customer may receive instructions, via a GUI display, to tap the contactless card on the mobile device, ensuring that the card is within a short-range communication range of the mobile device. At 634, the mobile device may process the contactless card information by sending the information to the server to perform the authentication and receiving a result of the authentication performed. At 636, the mobile device may confirm that the authentication was successful.

[0077] At 638, a use case policy analysis may be performed and applied to the task to determine a type of authentication that is available for the user. This process may be performed by the mobile application prior to the authentication operation. In one example, the use case policy may be applied to information of the user and determine that the user has a contactless card for authentication. In another example, the use case policy may determine that the customer does not have a contactless card and an alternative method needs to be used to authenticate the user. In some instances, the user may have more than one authentication method available to use for authentication (e.g., contactless card, biometric, identification), and the use case policy may assign different authenticate methods based on the risk level of the task to be performed by the user. For example, higher-risk tasks may require a stronger authentication method than lower-risk tasks. The authentication methods may be ranked from strongest to weakest, for example.

[0078] In embodiments, the authentication operation may be performed and provide a result to the customer and the mobile device. At 640, the customer may communicate to the agent that the authentication operation is successful. In some instances, at 646, the agent may receive an indication from the mobile application and mobile device confirming that the authentication is successful. The status may be provided in a GUI display at 644, and the agent may confirm to the customer that they are authenticated at 642.

[0079] FIG. 7 illustrates an example configuration of a contactless card 700, which may include a contactless card, a payment card, such as a credit card, debit card, or gift card, issued by a service provider as displayed as service provider indicia 702 on the front or back of the contactless card 700. In some examples, the contactless card 700 is not related to a payment card and may include, without limitation, an identification card. In some examples, the transaction card may include a dual interface contactless payment card, a rewards card, and so forth. The contactless card 700 may include a substrate 708, 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 contactless card 700 may have physical characteristics compliant with the ID-1 format of the ISO/IEC 7816 standard, and the transaction card may otherwise be compliant with the ISO/IEC 14443 standard. However, it is understood that the contactless card 700, according to the present disclosure, may have different characteristics, and the present disclosure does not require a transaction card to be implemented in a payment card.

[0080] The contactless card 700 may also include identification information 706 displayed on the front and/or back of the card, and a contact pad 704. The contact pad 704 may include one or more pads and be configured to establish contact with another client device, such as an ATM, a user device, smartphone, laptop, desktop, or tablet computer via transaction cards. The contact pad may be designed in accordance with one or more standards, such as ISO/IEC 7816 standard, and enable communication in accordance with the EMV protocol. The contactless card 700 may also include processing circuitry, antenna and other components as will be further discussed in FIG. 8. These components may be located behind the contact pad 704 or elsewhere on the substrate 708, e.g. within a different layer of the substrate 708, and may electrically and physically coupled with the contact pad 704. The contactless card 700 may also

include a magnetic strip or tape, which may be located on the back of the card (not shown in FIG. 7). The contactless card 700 may also include a Near-Field Communication (NFC) device coupled with an antenna capable of communicating via the NFC protocol. Embodiments are not limited in this manner.

[0081] As illustrated in FIG. 7, the contact pad 704 of contactless card 700 may include processing circuitry 816 for storing, processing, and communicating information, including a processor 802, a memory 804, and one or more interface(s) 806. It is understood that the processing circuitry 816 may contain additional components, including processors, memories, error and parity/CRC checkers, data encoders, anticollision algorithms, controllers, command decoders, security primitives and tamper-proofing hardware, as necessary to perform the functions described herein.

[0082] The memory 804 may be a read-only memory, write-once read-multiple memory or read/write memory, e.g., RAM, ROM, and EEPROM, and the contactless card 700 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-programmed many times after leaving the factory. A read/write memory may also be read many times after leaving the factory. In some instances, the memory 804 may be encrypted memory utilizing an encryption algorithm executed by the processor 802 to encrypted data.

[0083] The memory 804 may be configured to store one or more applet(s) 808, one or more counter(s) 810, a customer identifier 814, and the account number(s) 812, which may be virtual account numbers. The one or more applet(s) 808 may comprise one or more software applications configured to execute on one or more contactless cards, such as a Java® Card applet. However, it is understood that applet(s) 808 are not limited to Java Card applets and instead may be any software application operable on contactless cards or other devices having limited memory. The one or more counter(s) 810 may comprise a numeric counter sufficient to store an integer. The customer identifier 814 may comprise a unique alphanumeric identifier assigned to a user of the contactless card 700, and the identifier may distinguish the user of the contactless card from other contactless card users. In some examples, the customer identifier 814 may identify both a customer and an account assigned to that customer and may further

identify the contactless card 700 associated with the customer's account. As stated, the account number(s) 812 may include thousands of one-time use virtual account numbers associated with the contactless card 700. An applet(s) 808 of the contactless card 700 may be configured to manage the account number(s) 812 (e.g., to select an account number(s) 812, mark the selected account number(s) 812 as used, and transmit the account number(s) 812 to a mobile device for auto filling by an auto filling service.

[0084] The processor 802 and memory elements of the foregoing exemplary embodiments are described with reference to the contact pad 704, but the present disclosure is not limited thereto. It is understood that these elements may be implemented outside of the contact pad 704 or entirely separate from it or as further elements in addition to processor 802 and memory 804 elements located within the contact pad 704.

[0085] In some examples, the contactless card 700 may comprise one or more antenna(s) 818. The one or more antenna(s) 818 may be placed within the contactless card 700 and around the processing circuitry 816 of the contact pad 704. For example, the one or more antenna(s) 818 may be integral with the processing circuitry 816 and one or more antenna(s) 818 may be used with an external booster coil. As another example, one or more antenna(s) 818 may be external to the contact pad 704 and the processing circuitry 816.

[0086] In an embodiment, the coil of contactless card 700 may act as the secondary of an air-core transformer. The terminal may communicate with the contactless card 700 by cutting power or amplitude modulation. The contactless card 101 may infer the data transmitted from the terminal using the gaps in the contactless card's power connection, which may be functionally maintained through one or more capacitors. The contactless card 700 may communicate back by switching a load on the contactless card's coil or load modulation. Load modulation may be detected in the terminal's coil through interference. More generally, using the antenna(s) 818, processor 802, and/or the memory 804, the contactless card 101 provides a communications interface to communicate via NFC, Bluetooth, and/or Wi-Fi communications.

[0087] As explained above, contactless card 700 may be built on a software platform operable on smart cards or other devices having limited memory, such as JavaCard, and one or more or more applications or applets may be securely executed. Applet(s) 808 may be added to contactless cards to provide a one-time password (OTP) for multifactor authentication (MFA) in various mobile application-based use cases. Applet(s) 808 may be configured to respond to one or more requests, such as near field data exchange requests, from a reader, such as a mobile

NFC reader (e.g., of a mobile device or point-of-sale terminal), and produce an NDEF message that comprises a cryptographically secure OTP encoded as an NDEF text tag.

[0088] One example of an NDEF OTP is an NDEF short-record layout (SR=1). In such an example, one or more applet(s) 808 may be configured to encode the OTP as an NDEF type 4 well-known type text tag. In some examples, NDEF messages may comprise one or more records. The applet(s) 808 may be configured to add one or more static tag records in addition to the OTP record.

[0089] In some examples, one or more applet(s) 808 may be configured to emulate an RFID tag. The RFID tag may include one or more polymorphic tags. In some examples, each time the tag is read, different cryptographic data is presented that may indicate the authenticity of the contactless card. Based on the one or more applet(s) 808, an NFC read of the tag may be processed, the data may be transmitted to a server, such as a server of a banking system, and the data may be validated at the server.

[0090] In some examples, the contactless card 700 and server may include certain data such that the card may be properly identified. The contactless card 700 may include one or more unique identifiers (not pictured). Each time a read operation takes place, the counter(s) 810 may be configured to increment. In some examples, each time data from the contactless card 700 is read (e.g., by a mobile device), the counter(s) 810 is transmitted to the server for validation and determines whether the counter(s) 810 are equal (as part of the validation) to a counter of the server.

[0091] One or more counter(s) 810 may be configured to prevent a replay attack. For example, if a cryptogram has been obtained and replayed, that cryptogram is immediately rejected if the counter(s) 810 has been read or used or otherwise passed over. If the counter(s) 810 has not been used, it may be replayed. In some examples, the counter that is incremented on the card is different from the counter that is incremented for transactions. The contactless card 101 is unable to determine the application transaction counter(s) 810 since there is no communication between applet(s) 808 on the contactless card 700.

[0092] In some examples, the counter(s) 810 may get out of sync. In some examples, to account for accidental reads that initiate transactions, such as reading at an angle, the counter(s) 810 may increment, but the application does not process the counter(s) 810. In some examples, when the mobile device is woken up, NFC may be enabled, and the device 110 may be configured to read available tags, but no action is taken responsive to the reads.

[0093] To keep the counter(s) 810 in sync, an application, such as a background application, may be executed that would be configured to detect when the mobile device 110 wakes up and synchronize with the server of a banking system indicating that a read that occurred due to detection to then move the counter(s) 810 forward. In other examples, Hashed One Time Password may be utilized such that a window of mis-synchronization may be accepted. For example, if within a threshold of 10, the counter(s) 810 may be configured to move forward. But if within a different threshold number, a request for performing re-synchronization may be processed which requests via one or more applications that the user tap, gesture, or otherwise indicate one or more times via the user's device. If the counter(s) 810 increases in the appropriate sequence, then it possible to know that the user has done so.

[0094] The key diversification technique described herein with reference to the counter(s) 810, master key, and diversified key is one example of encryption and/or decryption a key diversification technique. This example key diversification technique should not be considered limiting of the disclosure, as the disclosure is equally applicable to other types of key diversification techniques.

[0095] During the creation process of the contactless card 700, two cryptographic keys may be assigned uniquely per card. The cryptographic keys may comprise symmetric keys which may be used in both encryption and decryption of data. Triple DES (3DES) algorithm may be used by EMV and it is implemented by hardware in the contactless card 700. By using the key diversification process, one or more keys may be derived from a master key based upon uniquely identifiable information for each entity that requires a key.

[0096] In some examples, to overcome deficiencies of 3DES algorithms, which may be susceptible to vulnerabilities, a session key may be derived (such as a unique key per session) but rather than using the master key, the unique card-derived keys and the counter may be used as diversification data. For example, each time the contactless card 101 is used in operation, a different key may be used for creating the message authentication code (MAC) and for performing the encryption. This results in a triple layer of cryptography. The session keys may be generated by the one or more applets and derived by using the application transaction counter with one or more algorithms (as defined in EMV 4.3 Book 2 A1.3.1 Common Session Key Derivation).

[0097] Further, the increment for each card may be unique and assigned either by personalization or algorithmically assigned by some identifying information. For

example, odd-numbered cards may increment by 2, and even-numbered cards may increment by 5. In some examples, the increment may also vary in sequential reads, such that one card may increment in sequence by 1, 3, 5, 2, 2, ... repeating. The specific sequence or algorithmic sequence may be defined at personalization time or from one or more processes derived from unique identifiers. This can make it harder for a replay attacker to generalize from a small number of card instances.

[0098] The authentication message may be delivered as the content of a text NDEF record in hexadecimal ASCII format. In another example, the NDEF record may be encoded in hexadecimal format.

[0099] FIG. 9 is a timing diagram illustrating an example sequence for providing authenticated access according to one or more embodiments of the present disclosure. Sequence flow 900 may include contactless card 700 and computing device 102, which may include an application(s) 210 and processor(s) 202.

[0100] At line 904, the application(s) 210 communicates with the contactless card 700 (e.g., after being brought near the contactless card 700). Communication between the application(s) 210 and the contactless card 700 may involve the contactless card 700 being sufficiently close to a card reader (not shown) of the computing device 102 to enable NFC data transfer between the application(s) 210 and the contactless card 700.

[0101] At line 902, after communication has been established between computing device 102 and contactless card 700, contactless card 700 generates a message authentication code (MAC) cryptogram. In some examples, this may occur when the contactless card 700 is read by the application(s) 210. In particular, this may occur upon a read, such as an NFC read, of a near field data exchange (NDEF) tag, which may be created in accordance with the NFC Data Exchange Format. For example, a reader application, such as application(s) 210, may transmit a message, such as an applet select message, with the applet ID of an NDEF producing applet. Upon confirmation of the selection, a sequence of select file messages followed by read file messages may be transmitted. For example, the sequence may include "Select Capabilities file", "Read Capabilities file", and "Select NDEF file". At this point, a counter value maintained by the contactless card 700 may be updated or incremented, which may be followed by "Read NDEF file." At this point, the message may be generated which may include a header and a shared secret. Session keys may then be generated. The MAC cryptogram may be created from the message, which may include the header and the shared secret. The MAC cryptogram

may then be concatenated with one or more blocks of random data, and the MAC cryptogram and a random number (RND) may be encrypted with the session key. Thereafter, the cryptogram and the header may be concatenated, and encoded as ASCII hex and returned in NDEF message format (responsive to the “Read NDEF file” message).

[0102] In some examples, the MAC cryptogram may be transmitted as an NDEF tag, and in other examples, the MAC cryptogram may be included with a uniform resource indicator (e.g., as a formatted string). In some examples, application(s) 210 may be configured to transmit a request to contactless card 700, the request comprising an instruction to generate a MAC cryptogram.

[0103] At line 906, the contactless card 700 sends the MAC cryptogram to the application(s) 210. In some examples, the transmission of the MAC cryptogram occurs via NFC; however, the present disclosure is not limited thereto. In other examples, this communication may occur via Bluetooth, Wi-Fi, or other means of wireless data communication. At line 908, the application(s) 210 communicates the MAC cryptogram to the processor(s) 202.

[0104] At line 910, the processor(s) 202 verifies the MAC cryptogram pursuant to an instruction from the application 122. For example, the MAC cryptogram may be verified, as explained below. In some examples, verifying the MAC cryptogram may be performed by a device other than computing device 102, such as a server of a banking system in data communication with the computing device 102. For example, processor(s) 202 may output the MAC cryptogram for transmission to the server of the banking system, which may verify the MAC cryptogram. In some examples, the MAC cryptogram may function as a digital signature for purposes of verification and authentication. Other digital signature algorithms, such as public key asymmetric algorithms, e.g., the Digital Signature Algorithm and the RSA algorithm, or zero knowledge protocols, may be used to perform this verification.

[0105] FIG. 10 illustrates a diagram of a system 1000 configured to implement one or more embodiments of the present disclosure. As explained below, during the contactless card creation process, two cryptographic keys may be assigned uniquely for each card. The cryptographic keys may comprise symmetric keys which may be used in both encryption and decryption of data. Triple DES (3DES) algorithm may be used by EMV and it is implemented by hardware in the contactless card. By using a key diversification process, one or more keys may be derived from a master key based upon uniquely identifiable information for each entity that requires a key.

[0106] Regarding master key management, two issuer master keys 1002, 1026 may be required for each part of the portfolio on which the one or more applets is issued. For example, the first master key 1002 may comprise an Issuer Cryptogram Generation/Authentication Key (Iss-Key-Auth) and the second master key 1026 may comprise an Issuer Data Encryption Key (Iss-Key-DEK). As further explained herein, two issuer master keys 1002, 1026 are diversified into card master keys 1008, 1020, which are unique for each card. In some examples, a network profile record ID (pNPR) 522 and derivation key index (pDKI) 1024, as back office data, may be used to identify which Issuer Master Keys 1002, 1026 to use in the cryptographic processes for authentication. The system performing the authentication may be configured to retrieve values of pNPR 1022 and pDKI 1024 for a contactless card at the time of authentication.

[0107] In some examples, to increase the security of the solution, a session key may be derived (such as a unique key per session) but rather than using the master key, the unique card-derived keys and the counter may be used as diversification data, as explained above. For example, each time the card is used in operation, a different key may be used for creating the message authentication code (MAC) and for performing the encryption. Regarding session key generation, the keys used to generate the cryptogram and encipher the data in the one or more applets may comprise session keys based on the card unique keys (Card-Key-Auth 1008 and Card-Key-DEK 1020). The session keys (Aut-Session-Key 1032 and DEK-Session-Key 1010) may be generated by the one or more applets and derived by using the application transaction counter (pATC) 1004 with one or more algorithms. To fit data into the one or more algorithms, only the 2 low order bytes of the 4-byte pATC 1004 is used. In some examples, the four byte session key derivation method may comprise: $F1 := \text{PATC}(\text{lower 2 bytes}) \parallel 'F0' \parallel '00' \parallel \text{PATC}(\text{four bytes})$ $F1 := \text{PATC}(\text{lower 2 bytes}) \parallel '0F' \parallel '00' \parallel \text{PATC}(\text{four bytes})$ $SK := \{(\text{ALG}(\text{MK})[F1]) \parallel \text{ALG}(\text{MK})[F2]\}$, where ALG may include 3DES ECB and MK may include the card unique derived master key.

[0108] As described herein, one or more MAC session keys may be derived using the lower two bytes of pATC 1004 counter. At each tap of the contactless card, pATC 1004 is configured to be updated, and the card master keys Card-Key-AUTH 808 and Card-Key-DEK 1020 are further diversified into the session keys Aut-Session-Key 1032 and DEK-Session-KEY 1010. pATC 1004 may be initialized to zero at personalization or applet initialization time. In some examples, the pATC counter may be initialized at or before personalization, and may be configured to increment by one at each NDEF read.

[0109] Further, the update for each card may be unique, and assigned either by personalization, or algorithmically assigned by pUID or other identifying information. For example, odd numbered cards may increment or decrement by 2 and even numbered cards may increment or decrement by 5. In some examples, the update may also vary in sequential reads, such that one card may increment in sequence by 1, 3, 5, 2, 2, ... repeating. The specific sequence or algorithmic sequence may be defined at personalization time, or from one or more processes derived from unique identifiers. This can make it harder for a replay attacker to generalize from a small number of card instances.

[0110] The authentication message may be delivered as the content of a text NDEF record in hexadecimal ASCII format. In some examples, only the authentication data and an 8-byte random number followed by MAC of the authentication data may be included. In some examples, the random number may precede cryptogram A and may be one block long. In other examples, there may be no restriction on the length of the random number. In further examples, the total data (i.e., the random number plus the cryptogram) may be a multiple of the block size. In these examples, an additional 8-byte block may be added to match the block produced by the MAC algorithm. As another example, if the algorithms employed used 16-byte blocks, even multiples of that block size may be used, or the output may be automatically, or manually, padded to a multiple of that block size.

[0111] The MAC may be performed by a function key (AUT-Session-Key) 1032. The data specified in cryptogram may be processed with javacard.signature method: `ALG_DES_MAC8_ISO9797_1_M2_ALG3` to correlate to EMV ARQC verification methods. The key used for this computation may comprise a session key AUT-Session-Key 1032, as explained above. As explained above, the low order two bytes of the counter may be used to diversify for the one or more MAC session keys. As explained below, AUT-Session-Key 1032 may be used to MAC data 1006, and the resulting data or cryptogram A 1014 and random number RND may be encrypted using DEK-Session-Key 1010 to create cryptogram B or output 1018 sent in the message.

[0112] In some examples, one or more HSM commands may be processed for decrypting such that the final 16 (binary, 32 hex) bytes may comprise a 3DES symmetric encrypting using CBC mode with a zero IV of the random number followed by MAC authentication data. The key used for this encryption may comprise a session key DEK-Session-Key 1010 derived from the

Card-Key-DEK 1020. In this case, the ATC value for the session key derivation is the least significant byte of the counter pATC 1004.

[0113] The format below represents a binary version example embodiment. Further, in some examples, the first byte may be set to ASCII 'A'.

Message Format				
1	2	4	8	8
0x43 (Message Type 'A')	Version	pATC	RND	Cryptogram A (MAC)
Cryptogram A (MAC)	8 bytes			
MAC of				
2	8	4	4	18 bytes input data
Version	pUID	pATC	Shared Secret	

Message Format				
1	2	4	16	
0x43 (Message Type 'A')	Version	pATC		Cryptogram B
Cryptogram A (MAC)	8 bytes			
MAC of				
2	8	4	4	18 bytes input data
Version	pUID	pATC	Shared Secret	
Cryptogram B	16			
Sym Encryption of				

8	8		
RND	Cryptogram A		

[0114] Another exemplary format is shown below. In this example, the tag may be encoded in hexadecimal format.

Message Format				
2	8	4	8	8
Version	pUID	pATC	RND	Cryptogram A (MAC)
8 bytes				
8	8	4	4	18 bytes input data
pUID	pUID	pATC	Shared Secret	

Message Format				
2	8	4	16	
Version	pUID	pATC	Cryptogram B	
8 bytes				
8		4	4	18 bytes input data
pUID	pUID	pATC	Shared Secret	
Cryptogram B		16		
Sym Encryption of				
8	8			

RND	Cryptogram A		
-----	-----------------	--	--

[0115] The UID field of the received message may be extracted to derive, from master keys Iss-Key-AUTH 802 and Iss-Key-DEK 1026, the card master keys (Card-Key-Auth 1008 and Card-Key-DEK 1020) for that particular card. Using the card master keys (Card-Key-Auth 808 and Card-Key-DEK 1020), the counter (pATC) field of the received message may be used to derive the session keys (Aut-Session-Key 1032 and DEK-Session-Key 1010) for that particular card. Cryptogram B 1018 may be decrypted using the DEK-Session-KEY, which yields cryptogram A 1014 and RND, and RND may be discarded. The UID field may be used to look up the shared secret of the contactless card which, along with the Ver, UID, and pATC fields of the message, may be processed through the cryptographic MAC using the re-created Aut-Session-Key to create a MAC output, such as MAC'. If MAC' is the same as cryptogram A 1014, then this indicates that the message decryption and MAC checking have all passed. Then the pATC may be read to determine if it is valid.

[0116] During an authentication session, one or more cryptograms may be generated by the one or more applications by the contactless card 700. For example, the one or more cryptograms may be generated as a 3DES MAC using ISO 9797-1 Algorithm 3 with Method 2 padding via one or more session keys, such as Aut-Session-Key 1032. The input data 1006 may take the following form: Version (2), pUID (8), pATC (4), Shared Secret (4). In some examples, the numbers in the brackets may comprise length in bytes. In some examples, the shared secret may be generated by one or more random number generators which may be configured to ensure, through one or more secure processes, that the random number is unpredictable. In some examples, the shared secret may comprise a random 4-byte binary number injected into the card at personalization time that is known by the authentication service. During an authentication session, the shared secret may not be provided from the one or more applets to the mobile application. Method 2 padding may include adding a mandatory 0x'80' byte to the end of input data and 0x'00' bytes that may be added to the end of the resulting data up to the 8-byte boundary. The resulting cryptogram may comprise 8 bytes in length.

[0117] In some examples, one benefit of encrypting an unshared random number as the first block with the MAC cryptogram, is that it acts as an initialization vector while using CBC

(Block chaining) mode of the symmetric encryption algorithm. This allows the “scrambling” from block to block without having to pre-establish either a fixed or dynamic IV.

[0118] By including the application transaction counter (pATC) as part of the data included in the MAC cryptogram, the authentication service may be configured to determine if the value conveyed in the clear data has been tampered with. Moreover, by including the version in the one or more cryptograms, it is difficult for an attacker to purposefully misrepresent the application version in an attempt to downgrade the strength of the cryptographic solution. In some examples, the pATC may start at zero and be updated by 1 each time the one or more applications generates authentication data. The authentication service may be configured to track the pATCs used during authentication sessions. In some examples, when the authentication data uses a pATC equal to or lower than the previous value received by the authentication service, this may be interpreted as an attempt to replay an old message, and the authenticated may be rejected. In some examples, where the pATC is greater than the previous value received, this may be evaluated to determine if it is within an acceptable range or threshold, and if it exceeds or is outside the range or threshold, verification may be deemed to have failed or be unreliable. In the MAC operation 1012, data 1006 is processed through the MAC using Aut-Session-Key 1032 to produce MAC output (cryptogram A) 1014, which is encrypted.

[0119] In order to provide additional protection against brute force attacks exposing the keys on the card, it is desirable that the MAC cryptogram 1014 be enciphered. In some examples, data or cryptogram A 1014 to be included in the ciphertext may comprise: Random number (8), cryptogram (8). In some examples, the numbers in the brackets may comprise length in bytes. In some examples, the random number may be generated by one or more random number generators which may be configured to ensure, through one or more secure processes, that the random number is unpredictable. The key used to encipher this data may comprise a session key. For example, the session key may comprise DEK-Session-Key 1010. In the encryption operation 1016, data or cryptogram A 1014 and RND are processed using DEK-Session-Key 810 to produce encrypted data, cryptogram B 1018. The data 1014 may be enciphered using 3DES in cipher block chaining mode to ensure that an attacker must run any attacks over all of the ciphertext. As a non-limiting example, other algorithms, such as Advanced Encryption Standard (AES), may be used. In some examples, an initialization vector of 0x'0000000000000000' may be used. Any attacker seeking to brute force the key used for

enciphering this data will be unable to determine when the correct key has been used, as correctly decrypted data will be indistinguishable from incorrectly decrypted data due to its random appearance.

[0120] In order for the authentication service to validate the one or more cryptograms provided by the one or more applets, the following data must be conveyed from the one or more applets to the mobile device in the clear during an authentication session: version number to determine the cryptographic approach used and message format for validation of the cryptogram, which enables the approach to change in the future; pUID to retrieve cryptographic assets, and derive the card keys; and pATC to derive the session key used for the cryptogram.

[0121] FIG. 11 illustrates a method 1100 for generating a cryptogram that may be performed by a contactless card 700. For example, at block 1102, a network profile record ID (pNPR) and derivation key index (pDKI) may be used to identify which Issuer Master Keys to use in the cryptographic processes for authentication. In some examples, the method may include performing the authentication to retrieve values of pNPR and pDKI for a contactless card at the time of authentication.

[0122] At block 1104, Issuer Master Keys may be diversified by combining them with the card's unique ID number (pUID) and the PAN sequence number (PSN) of one or more applets, for example, a payment applet.

[0123] At block 1106, Card-Key-Auth and Card-Key-DEK (unique card keys) may be created by diversifying the Issuer Master Keys to generate session keys which may be used to generate a MAC cryptogram.

[0124] At block 1108, the keys used to generate the cryptogram and encipher the data in the one or more applets may comprise the session keys based on the card unique keys (Card-Key-Auth and Card-Key-DEK). In some examples, these session keys may be generated by the one or more applets and derived by using pATC, resulting in session keys Aut-Session-Key and DEK-Session-Key.

[0125] FIG. 12 depicts an exemplary process 1200 illustrating key diversification according to one example. Initially, a sender and the recipient may be provisioned with two different master keys. For example, a first master key may comprise the data encryption master key, and a second master key may comprise the data integrity master key. The sender has a counter value, which may be updated at block 1202, and other data, such as data to be protected, which it may secure share with the recipient.

[0126] At block 1204, the counter value may be encrypted by the sender using the data encryption master key to produce the data encryption derived session key, and the counter value may also be encrypted by the sender using the data integrity master key to produce the data integrity derived session key. In some examples, a whole counter value or a portion of the counter value may be used during both encryptions.

[0127] In some examples, the counter value may not be encrypted. In these examples, the counter may be transmitted between the sender and the recipient in the clear, i.e., without encryption.

[0128] At block 1206, the data to be protected is processed with a cryptographic MAC operation by the sender using the data integrity session key and a cryptographic MAC algorithm. The protected data, including plaintext and shared secret, may be used to produce a MAC using one of the session keys (AUT-Session-Key).

[0129] At block 1208, the data to be protected may be encrypted by the sender using the data encryption derived session key in conjunction with a symmetric encryption algorithm. In some examples, the MAC is combined with an equal amount of random data, for example each 8 bytes long, and then encrypted using the second session key (DEK-Session-Key).

[0130] At block 1210, the encrypted MAC is transmitted, from the sender (contactless card 700) to the recipient (system 104 via computing device 102), with sufficient information to identify additional secret information (such as shared secret, master keys, etc.), for verification of the cryptogram.

[0131] At block 1212, the recipient uses the received counter value to independently derive the two derived session keys from the two master keys as explained above.

[0132] At block 1214, the data encryption derived session key is used in conjunction with the symmetric decryption operation to decrypt the protected data. Additional processing on the exchanged data will then occur. In some examples, after the MAC is extracted, it is desirable to reproduce and match the MAC. For example, when verifying the cryptogram, it may be decrypted using appropriately generated session keys. The protected data may be reconstructed for verification. A MAC operation may be performed using an appropriately generated session key to determine if it matches the decrypted MAC. As the MAC operation is an irreversible process, the only way to verify is to attempt to recreate it from source data.

[0133] At block 1216, the data integrity derived session key is used in conjunction with the cryptographic MAC operation to verify that the protected data has not been modified.

[0134] Some examples of the methods described herein may advantageously confirm when a successful authentication is determined when the following conditions are met. First, the ability to verify the MAC shows that the derived session key was proper. The MAC may only be correct if the decryption was successful and yielded the proper MAC value. The successful decryption may show that the correctly derived encryption key was used to decrypt the encrypted MAC. Since the derived session keys are created using the master keys known only to the sender (e.g., the transmitting device) and recipient (e.g., the receiving device), it may be trusted that the contactless card which originally created the MAC and encrypted the MAC is indeed authentic. Moreover, the counter value used to derive the first and second session keys may be shown to be valid and may be used to perform authentication operations.

[0135] Thereafter, the two derived session keys may be discarded, and the next iteration of data exchange will update the counter value (returning to block 1202) and a new set of session keys may be created (at block 1210). In some examples, the combined random data may be discarded.

[0136] FIG. 13 illustrates an embodiment of an exemplary computer architecture 1300 suitable for implementing various embodiments as previously described. In one embodiment, the computer architecture 1300 may include or be implemented as part of one or more systems or devices discussed herein.

[0137] As used in this application, the terms “system” and “component” are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution, examples of which are provided by the exemplary computing computer architecture 1300. For example, a component can be, but is not limited to being, a process running on a processor, a processor, a hard disk drive, multiple storage drives (of optical and/or magnetic storage medium), an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a server and the server can be a component. One or more components can reside within a process and/or thread of execution, and a component can be localized on one computer and/or distributed between two or more computers. Further, components may be communicatively coupled to each other by various types of communications media to coordinate operations. The coordination may involve the uni-directional or bi-directional exchange of information. For instance, the components may communicate information in the form of signals communicated over the communications media. The information can be implemented as signals allocated to

various signal lines. In such allocations, each message is a signal. Further embodiments, however, may alternatively employ data messages. Such data messages may be sent across various connections. Exemplary connections include parallel interfaces, serial interfaces, and bus interfaces.

[0138] The computer architecture 1300 includes various common computing elements, such as one or more processors, multi-core processors, co-processors, memory units, chipsets, controllers, peripherals, interfaces, oscillators, timing devices, video cards, audio cards, multimedia input/output (I/O) components, power supplies, and so forth. The embodiments, however, are not limited to implementation by the computer architecture 1300.

[0139] As shown in FIG. 13, the computer architecture 1300 includes a processor 1312, a system memory 1304 and a system bus 1306. The processor 1312 can be any of various commercially available processors.

[0140] The system bus 1306 provides an interface for system components including, but not limited to, the system memory 1304 to the processor 1312. The system bus 1306 can be any of several types of bus structure that may further interconnect to a memory bus (with or without a memory controller), a peripheral bus, and a local bus using any of a variety of commercially available bus architectures. Interface adapters may connect to the system bus 902 via slot architecture. Example slot architectures may include without limitation Accelerated Graphics Port (AGP), Card Bus, (Extended) Industry Standard Architecture ((E)ISA), Micro Channel Architecture (MCA), NuBus, Peripheral Component Interconnect (Extended) (PCI(X)), PCI Express, Personal Computer Memory Card International Association (PCMCIA), and the like.

[0141] The computer architecture 1300 may include or implement various articles of manufacture. An article of manufacture may include a computer-readable storage medium to store logic. Examples of a computer-readable storage medium may include any tangible media capable of storing electronic data, including volatile memory or non-volatile memory, removable or non-removable memory, erasable or non-erasable memory, writeable or re-writable memory, and so forth. Examples of logic may include executable computer program instructions implemented using any suitable type of code, such as source code, compiled code, interpreted code, executable code, static code, dynamic code, object-oriented code, visual code, and the like. Embodiments may also be at least partly implemented as instructions contained in or on a non-transitory computer-readable medium, which may be read and executed by one or more processors to enable the performance of the operations described herein.

[0142] The system memory 1304 may include various types of computer-readable storage media in the form of one or more higher speed memory units, such as read-only memory (ROM), random-access memory (RAM), dynamic RAM (DRAM), Double-Data-Rate DRAM (DDRAM), synchronous DRAM (SDRAM), static RAM (SRAM), programmable ROM (PROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), flash memory, polymer memory such as ferroelectric polymer memory, ovonic memory, phase change or ferroelectric memory, silicon-oxide-nitride-oxide-silicon (SONOS) memory, magnetic or optical cards, an array of devices such as Redundant Array of Independent Disks (RAID) drives, solid-state memory devices (e.g., USB memory, solid-state drives (SSD) and any other type of storage media suitable for storing information. In the illustrated embodiment shown in FIG. 13, the system memory 1304 can include non-volatile 1308 and/or volatile 1310. A basic input/output system (BIOS) can be stored in the non-volatile 1308.

[0143] The computer 1302 may include various types of computer-readable storage media in the form of one or more lower speed memory units, including an internal (or external) hard disk drive 1330, a magnetic disk drive 1316 to read from or write to a removable magnetic disk 1320, and an optical disk drive 1328 to read from or write to a removable optical disk 1332 (e.g., a CD-ROM or DVD). The hard disk drive 1330, magnetic disk drive 1316 and optical disk drive 1328 can be connected to system bus 1306 by an HDD interface 1314, and FDD interface 1318 and an optical disk drive interface 1334, respectively. The HDD interface 1314 for external drive implementations can include at least one or both of Universal Serial Bus (USB) and IEEE 1394 interface technologies.

[0144] The drives and associated computer-readable media provide volatile and/or nonvolatile storage of data, data structures, computer-executable instructions, and so forth. For example, a number of program modules can be stored in the drives and non-volatile 1308, and volatile 1310, including an operating system 1322, one or more applications 1342, other program modules 1324, and program data 1326. In one embodiment, the one or more applications 1342, other program modules 1324, and program data 1326 can include, for example, the various applications and/or components of the systems discussed herein.

[0145] A user can enter commands and information into the computer 1302 through one or more wired/wireless input devices, for example, a keyboard 1350 and a pointing device, such as a mouse 1352. Other input devices may include microphones, infra-red (IR) remote controls,

radio-frequency (RF) remote controls, game pads, stylus pens, card readers, dongles, fingerprint readers, gloves, graphics tablets, joysticks, keyboards, retina readers, touch screens (e.g., capacitive, resistive, etc.), trackballs, track pads, sensors, styluses, and the like. These and other input devices are often connected to the processor 1312 through an input device interface 1336 that is coupled to the system bus 1306 but can be connected by other interfaces such as a parallel port, IEEE 1394 serial port, a game port, a USB port, an IR interface, and so forth.

[0146] A monitor 1344 or other type of display device is also connected to the system bus 1306 via an interface, such as a video adapter 1346. The monitor 1344 may be internal or external to the computer 1302. In addition to the monitor 1344, a computer typically includes other peripheral output devices, such as speakers, printers, and so forth.

[0147] The computer 1302 may operate in a networked environment using logical connections via wire and/or wireless communications to one or more remote computers, such as a remote computer(s) 1348. The remote computer(s) 1348 can be a workstation, a server computer, a router, a personal computer, portable computer, microprocessor-based entertainment appliance, a peer device or other common network node, and typically includes many or all the elements described relative to the computer 1302, although, for purposes of brevity, only a memory and/or storage device 1358 is illustrated. The logical connections depicted include wire/wireless connectivity to a local area network 1356 and/or larger networks, for example, a wide area network 1354. Such LAN and WAN networking environments are commonplace in offices and companies, and facilitate enterprise-wide computer networks, such as intranets, all of which may connect to a global communications network, for example, the Internet.

[0148] When used in a local area network 1356 networking environment, the computer 1302 is connected to the local area network 1356 through a wired and/or wireless communication network interface or network adapter 1338. The network adapter 1338 can facilitate wire and/or wireless communications to the local area network 1356, which may also include a wireless access point disposed thereon for communicating with the wireless functionality of the network adapter 1338.

[0149] When used in a wide area network 1354 networking environment, the computer 1302 can include a modem 1340, or is connected to a communications server on the wide area network 1354 or has other means for establishing communications over the wide area network 1354, such as by way of the Internet. The modem 1340, which can be internal or external and a

wire and/or wireless device, connects to the system bus 1306 via the input device interface 1336. In a networked environment, program modules depicted relative to the computer 1302, or portions thereof, can be stored in the remote memory and/or storage device 1358. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers can be used.

[0150] The computer 1302 is operable to communicate with wire and wireless devices or entities using the IEEE 802 family of standards, such as wireless devices operatively disposed in wireless communication (e.g., IEEE 802.11 over-the-air modulation techniques). This includes at least Wi-Fi (or Wireless Fidelity), WiMax, and Bluetooth™ wireless technologies, among others. Thus, the communication can be a predefined structure as with a conventional network or simply an ad hoc communication between at least two devices. Wi-Fi networks use radio technologies called IEEE 802.11 (a, b, g, n, etc.) to provide secure, reliable, fast wireless connectivity. A Wi-Fi network can be used to connect computers to each other, to the Internet, and to wire networks (which use IEEE 802.3-related media and functions).

[0151] The various elements of the devices as previously described herein may include various hardware elements, software elements, or a combination of both. Examples of hardware elements may include devices, logic devices, components, processors, microprocessors, circuits, processors, circuit elements (e.g., transistors, resistors, capacitors, inductors, and so forth), integrated circuits, application-specific integrated circuits (ASIC), programmable logic devices (PLD), digital signal processors (DSP), field-programmable gate array (FPGA), memory units, logic gates, registers, semiconductor device, chips, microchips, chip sets, and so forth. Examples of software elements may include software components, programs, applications, computer programs, application programs, system programs, software development programs, machine programs, operating system software, middleware, firmware, software modules, routines, subroutines, functions, methods, procedures, software interfaces, application program interfaces (API), instruction sets, computing code, computer code, code segments, computer code segments, words, values, symbols, or any combination thereof. However, determining whether an embodiment is implemented using hardware elements and/or software elements may vary in accordance with any number of factors, such as desired computational rate, power levels, heat tolerances, processing cycle budget, input data rates, output data rates, memory resources, data bus speeds and other design or performance constraints, as desired for a given implementation.

[0152] The components and features of the devices described above may be implemented using any combination of discrete circuitry, application specific integrated circuits (ASICs), logic gates and/or single chip architectures. Further, the features of the devices may be implemented using microcontrollers, programmable logic arrays and/or microprocessors or any combination of the foregoing where suitably appropriate. It is noted that hardware, firmware and/or software elements may be collectively or individually referred to herein as “logic” or “circuit.”

CLAIMS

What is claimed is:

1. A system configured to authenticate unauthenticated users to perform sensitive actions, comprising:
 - one or more processors;
 - memory coupled with the one or more processors, the memory to store instructions that when executed by the one or more processors, cause the one or more processors to:
 - communicate a first communication to a mobile device, the first communication comprising first data to cause the mobile device to download a mobile application;
 - communicate a second communication to the mobile device, the second communication comprising second data comprising a task identifier, the second data to cause the mobile application to perform an authentication task associated with the task identifier;
 - determine the user is authenticated by the authentication task; and
 - enable the mobile device to perform sensitive actions.
2. The system of claim 1, wherein the authentication task comprises prompting a user to tap a contactless card on the mobile device, and the one or more processors to:
 - receive a cryptogram from the contactless card via the mobile device; and
 - verify the user is authenticated with the cryptogram based on stored data.
3. The system of claim 2, wherein the cryptogram comprises third data stored on the contactless card and is received in response to the contactless card being within a near-field communication range of the mobile device.
4. The system of claim 3, the one or more processors to compare the third data of the contactless card to the stored data stored and verify the user if they match.
5. The system of claim 1, wherein the authentication task comprises prompting a user to provide a picture of an identification card, and the one or more processors to:
 - receive the picture of the identification card; and
 - verify the user is authenticated with the picture.
6. The system of claim 1, wherein the first data comprises an embedded link to a location on the mobile device to download the mobile application.

7. The system of claim 6, wherein the first communication is a short message service (SMS) message or a rich communication service (RCS) message, and the embedded link is embedded the SMS message or RCS message.

8. The system of claim 1, wherein the task identifier is encoded in the second data, and the second data comprises an embedded link to launch the mobile application and initiate a function to perform the authentication task based on the task identifier.

9. The system of claim 8, wherein the second communication is a short message service (SMS) message or a rich communication service (RCS) message, and the embedded link is embedded the SMS message or RCS message.

10. The system of claim 1, wherein the security task is performed with the mobile application without the user providing any other authentication information.

11. The system of claim 1, the one or more processors to determine a phone number associated with the mobile device, and the authentication task further comprising verifying the phone number is associated with user.

12. A computing device configured to authenticate unauthenticated users to perform sensitive actions, comprising:

- processing circuitry;

- memory coupled with the processing circuitry, the memory to store instructions that when executed by the processing circuitry, cause the processing circuitry to:

- receive a first communication from a system to authenticate a user, the first communication comprising first data to install a mobile application;

- install the mobile application based on invocation of the first data;

- receive a second communication from the system, the second communication comprising second data comprising a task identifier, the second data to cause the mobile application to perform an authentication task associated with the task identifier;

- initiate the authentication task based on invocation of the second data to determine the user is authenticate; and

- receive an indication that the user is authenticated based on performance of the authentication task, and the user is enabled to perform sensitive actions.

13. The computing device of claim 12, comprising a wireless interface, and the processing circuitry, to perform the authentication task, to:
 - prompt the user to tap a contactless card on a surface of the computing device,
 - receive, via the wireless interface, a cryptogram from the contactless card; and
 - verify the user is authenticated with the cryptogram.
14. The computing device of claim 13, and the processing circuitry, to verify the cryptogram, to communicate the cryptogram to the system.
15. The computing device of claim 13, wherein the wireless interface comprises a near-field communication (NFC) interface, and the cryptogram is received from the contactless card as part of an NFC exchange.
16. The computing device of claim 12, comprising a camera, and the processing circuitry, to perform the authentication task, to:
 - prompting the user to provide a picture of an identification card; and
 - communicate the picture to the system.
17. The computing device of claim 12, wherein the first data comprises an embedded link to a mobile application store to download the mobile application, and the processing circuitry to receive an input on the embedded link, execute the mobile application store to download the mobile application, and download the mobile application.
18. The computing device of claim 12, wherein the task identifier is encoded in the second data, and the second data comprises an embedded link to launch the mobile application and initiate a function to perform the authentication task based on the task identifier.
19. The computing device of claim 12, wherein the security task is performed with the mobile application without the user providing any other authentication information.
20. The computing device of claim 12, the processing circuitry to provide a phone number associated with computing device to the system to perform the authentication task.