

UNITED STATES NON -PROVISIONAL PATENT APPLICATION
FOR

**SERVER-SIDE REDIRECT OF UNIFORM RESOURCE LOCATOR GENERATED BY
CONTACTLESS CARD**

**SERVER-SIDE REDIRECT OF UNIFORM RESOURCE LOCATOR GENERATED
BY CONTACTLESS CARD**

RELATED APPLICATIONS

[0001] This application claims priority to U.S. Patent Application Serial No. 17/359,220, entitled “SERVER-SIDE REDIRECT OF UNIFORM RESOURCE LOCATOR GENERATED BY CONTACTLESS CARD ” filed on June 25, 2021. The contents of the aforementioned application are incorporated herein by reference in their entirety.

BACKGROUND

[0002] Contactless cards may include logic to generate and/or store uniform resource locators (URLs). However, contactless cards may have limited resources compared to other computing devices. As such, there may be a limit to the number of different URLs a given contactless card can generate and/or store. Furthermore, the logic of the contactless card may need to be updated periodically. This may pose challenges, as the number of contactless cards may number in the thousands, millions, or more.

SUMMARY

[0003] Systems, methods, apparatuses, and computer-readable media for server-side redirect of uniform resource locators (URLs) generated by contactless cards. In one aspect, a server may receive, from a client device, a first hypertext transfer protocol (HTTP) request comprising a uniform resource locator (URL), where a first parameter of the URL includes a cryptogram generated by a contactless card and a second parameter of the URL includes a customer identifier of an account associated with the contactless card. The server may decrypt the cryptogram and determine a context of the account based on one or more attributes of the account. The server may select, based on the decryption of the cryptogram and the determined context of the account, a first redirect URL of a plurality of redirect URLs. The server may transmit, to the client device, an HTTP response including the redirect URL. The server may receive, from the client device, a second HTTP request including the redirect URL. The server may transmit, to the client device, an HTTP response including a resource at the redirect URL.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0004] 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.

[0005] FIG. 1A illustrates an aspect of the subject matter in accordance with one embodiment.

[0006] FIG. 1B illustrates an aspect of the subject matter in accordance with one embodiment.

[0007] FIG. 1C illustrates an aspect of the subject matter in accordance with one embodiment.

[0008] FIG. 1D illustrates an aspect of the subject matter in accordance with one embodiment.

[0009] FIG. 1E illustrates an aspect of the subject matter in accordance with one embodiment.

[0010] FIG. 2A illustrates an aspect of the subject matter in accordance with one embodiment.

[0011] FIG. 2B illustrates an aspect of the subject matter in accordance with one embodiment.

[0012] FIG. 2C illustrates an aspect of the subject matter in accordance with one embodiment.

[0013] FIG. 3 illustrates a routine 300 in accordance with one embodiment.

[0014] FIG. 4 illustrates a routine 400 in accordance with one embodiment.

[0015] FIG. 5 illustrates a routine 500 in accordance with one embodiment.

[0016] FIG. 6A illustrates a contactless card in accordance with one embodiment.

[0017] FIG. 6B illustrates a contactless card in accordance with one embodiment.

[0018] FIG. 7 illustrates a data structure in accordance with one embodiment.

[0019] FIG. 8 illustrates a computer architecture in accordance with one embodiment.

DETAILED DESCRIPTION

[0020] Embodiments disclosed herein provide techniques for secure server-side redirection of URLs generated by contactless cards. Generally, a user may tap a contactless card to a computing device, thereby bringing the card within wireless communications range of the computing device. In response, the contactless card may generate a uniform resource locator (URL) that includes a cryptogram as a first parameter and a customer identifier as a second

parameters. The computing device 102 may receive the URL, which may cause the computing device to open an application. The application may transmit the URL to an authentication server for processing. The server may attempt to decrypt the cryptogram. If the server is able to decrypt the cryptogram, the server may determine a redirect URL based on a context of an account associated with the card. The context may be determined based at least in part on one or more attributes of the account associated with the card. The server may determine the one or more attributes based on the customer ID specified in the URL. The server may transmit the redirect URL to the computing device, which causes the application to request and load the resource specified at the redirect URL.

[0021] In some embodiments, a machine learning (ML) model may be used to select the redirect URL and/or determine the context of the account. Generally, the ML model may be trained based on training data. The training data may describe redirect URLs selected for a received URL generated by a plurality of contactless cards 104 based on different account contexts and/or account attributes. Doing so may improve the accuracy in selecting redirect URLs relative to selecting redirect URLs without using machine learning.

[0022] Advantageously, embodiments disclosed herein provide secure, server-side redirect of URLs generated by contactless cards. By leveraging cryptograms generated by contactless cards, embodiments of the disclosure may securely verify the identity of the user with minimal risk of fraudulent activity. Furthermore, doing so ensures that redirect operations are only performed when the user has access to a contactless card that facilitates the cryptogram verification with the server. Furthermore, by providing the server-side redirect logic, fewer resources are needed in the contactless cards, thereby improving performance and reducing costs of the contactless cards. For example, the contactless card may not have enough storage and/or processing resources to have a one-to-one relationship between URLs and available web resources. However, by configuring the contactless card to generate a limited number of URLs that can be redirected by the server based on context, the contactless card can support a one-to-many relationship between URLs and available web resources. Furthermore, doing so may remove the need to periodically update the logic in the contactless card to support new URLs, e.g., as new web resources are added, removed, and/or modified.

[0023] With general reference to notations and nomenclature used herein, one or more portions of the detailed description which follows may be presented in terms of program procedures executed on a computer or network of computers. These procedural descriptions and representations are used by those skilled in the art to most effectively convey the substances of their work to others skilled in the art. A procedure is here, and generally,

conceived to be a self-consistent sequence of operations leading to a desired result. These operations are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be noted, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to those quantities.

[0024] Further, these manipulations are often referred to in terms, such as adding or comparing, which are commonly associated with mental operations performed by a human operator. However, no such capability of a human operator is necessary, or desirable in most cases, in any of the operations described herein that form part of one or more embodiments. Rather, these operations are machine operations. Useful machines for performing operations of various embodiments include digital computers as selectively activated or configured by a computer program stored within that is written in accordance with the teachings herein, and/or include apparatus specially constructed for the required purpose or a digital computer. Various embodiments also relate to apparatus or systems for performing these operations. These apparatuses may be specially constructed for the required purpose. The required structure for a variety of these machines will be apparent from the description given.

[0025] Reference is now made to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for the purpose of explanation, numerous specific details are set forth in order to provide a thorough understanding thereof. It may be evident, however, that the novel embodiments can be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate a description thereof. The intention is to cover all modification, equivalents, and alternatives within the scope of the claims.

[0026] Figure 1A depicts an exemplary computing architecture 100, also referred to as a system, consistent with disclosed embodiments. Although the computing architecture 100 shown in Figures 1A-1E has a limited number of elements in a certain topology, it may be appreciated that the computing architecture 100 may include more or less elements in alternate topologies as desired for a given implementation.

[0027] The computing architecture 100 comprises one or more computing devices 102, one or more servers 106, and one or more contactless cards 104. The contactless card 104 is representative of any type of card, such as a credit card, debit card, ATM card, gift card, payment card, smart card, and the like. The contactless card 104 may comprise one or more communications interfaces 122, such as a radio frequency identification (RFID) chip, configured to communicate with a communications interface 122 (also referred to herein as a “card reader”, a “wireless card reader”, and/or a “wireless communications interface”) of the computing devices 102 via NFC, the EMV standard, or other short-range protocols in wireless communication. Although NFC is used as an example communications protocol herein, the disclosure is equally applicable to other types of wireless communications, such as the EMV standard, Bluetooth, and/or Wi-Fi.

[0028] The computing device 102 is representative of any number and type of computing device, such as smartphones, tablet computers, wearable devices, laptops, portable gaming devices, virtualized computing system, merchant terminals, point-of-sale systems, servers, desktop computers, and the like. A mobile device may be used as an example of the computing device 102 but should not be considered limiting of the disclosure. The server 106 is representative of any type of computing device, such as a server, workstation, compute cluster, cloud computing platform, virtualized computing system, and the like. Although not depicted for the sake of clarity, the computing device 102, contactless card 104, and server 106 each include one or more processor circuits to execute programs, code, and/or instructions.

[0029] As shown, a memory 108 of the contactless card 104 includes an applet 110, a counter 112, a master key 114, a diversified key 116, a unique customer identifier (ID) 118, and one or more URLs 120. The applet 110 is executable code configured to perform the operations described herein. The counter 112, master key 114, diversified key 116, and customer ID 118 are used to provide security in the system 100 as described in greater detail below.

[0030] As shown, a memory 132 of the computing device 102 includes an instance of an operating system 134. Example operating systems include the Android® OS, iOS®, macOS®, Linux®, and Windows® operating systems. As shown, the operating system 134 includes an account application 136 and a web browser 138. The account application 136 allows users to perform various account-related operations, such as activating payment cards, viewing account balances, purchasing items, processing payments, and the like. In some embodiments, a user may authenticate using authentication credentials to access certain features of the account application 136. For example, the authentication credentials

may include a username (or login) and password, biometric credentials (e.g., fingerprints, Face ID, etc.), and the like. The web browser 138 is an application that allows the computing device 102 to access information via the network 156 (e.g., via the Internet). For example, using the web browser 138, the user may access one or more resources in the web server database 146 of the server 106. The resources may include pages for activating contactless cards 104, viewing account balances, purchasing items, processing payments, onboarding experiences for newly activated contactless cards 104, and the like. Similar to the account application 136, the user may authenticate using authentication credentials prior to accessing such web pages.

[0031] As shown, a memory 124 of the server 106 includes an authentication application 142, a machine learning (ML) model 150, and a web server 144. Although depicted as separate components of the server 106, in some embodiments, the authentication application 142, ML model 150, and/or the web server 144 may be integrated into a single component. Furthermore, the authentication application 142, ML model 150, and/or the authentication application 142 may be implemented in hardware, software, and/or a combination of hardware and software. The memory 124 further includes an account database 126, the web server database 146, and a data store of training data 152. The account database 126 generally includes information related to an account holder (e.g., one or more users), one or more accounts of the account holder, and one or more contactless cards 104 of the account.

[0032] In some embodiments, a user may need to perform an operation and/or access a resource in the web server database 146. For example, the user may need to activate the contactless card 104 from an inactive payment state to an active payment state such that the contactless card 104 may be used to process payments. As another example, the user may need customer support, e.g., when unable to access their account via the account application 136, when a purchase using the contactless card 104 is declined, etc. Often, many resources in the web server database 146 may assist the user to perform any requested operations and/or receive assistance. However, because there may be any number of resources in the web server database 146, the contactless card 104 may not have sufficient resources to store a URL 120 for each resource in the web server database 146. As another example, the contactless card 104 may not have been updated to reflect the most recent state of the resources in the web server database 146. Advantageously, the system 100 implements secure server-side redirect of URLs 120 generated by the contactless cards 104.

[0033] In the embodiment depicted in Figure 1A, the user may tap the contactless card 104 to the computing device 102 (or otherwise bring the contactless card 104 within communications range of the communications interface 122 of the device 102). The applet

110 of the contactless card 104 may then generate a URL 120 that is directed to a resource, such as the server 106, the authentication application 142, web server 144, and/or a resource in the web server database 146. In some embodiments, the URL 120 is a generic URL that may be interpreted by the server 106 to determine a particular resource in the web server database 146. Stated differently, the server 106 may need to identify a redirect URL for the generic URL 120 and return the redirect URL to the computing device 102 such that the computing device 102 can access the resource at the redirect URL. Therefore, in such embodiments, the URL 120 may be directed to a redirect component of the server 106, e.g., a page and/or application that handles redirects as described herein. In some embodiments, the applet 110 constructs the URL 120 according to one or more rules. In some embodiments, the contactless card 104 stores a plurality of generic URLs 120 and the applet 103 selects the URL 120 from the plurality of URLs 120 based on one or more rules. In some embodiments, the applet 110 may generate the URL 120 by selecting a URLs 120 and adding dynamic data, such as a cryptogram 130, as one or more parameters of the URL.

[0034] The cryptogram 130 may be based on the customer ID 118 of the contactless card 104. The cryptogram 130 may be generated based on any suitable cryptographic technique. In some embodiments, the applet 110 may include the URL 120, the cryptogram 130, and an unencrypted identifier (e.g., the customer ID 118, an identifier of the contactless card 104, and/or any other unique identifier) as part of a data package. In at least one embodiment, the data package is an NDEF file.

[0035] As stated, the computing architecture 100 is configured to implement key diversification to secure data, which may be referred to as a key diversification technique herein. Generally, the server 106 (or another computing device) and the contactless card 104 may be provisioned with the same master key 114 (also referred to as a master symmetric key). More specifically, each contactless card 104 is programmed with a distinct master key 114 that has a corresponding pair in the server 106. For example, when a contactless card 104 is manufactured, a unique master key 114 may be programmed into the memory 108 of the contactless card 104. Similarly, the unique master key 114 may be stored in a record of a customer associated with the contactless card 104 in the account database 126 of the server 106 (and/or stored in a different secure location, such as the hardware security module (HSM) 128). The master key 114 may be kept secret from all parties other than the contactless card 104 and server 106, thereby enhancing security of the system 100. In some embodiments, the applet 110 of the contactless card 104 may encrypt and/or decrypt data (e.g., the customer ID 118) using the master key 114 and the data as input a cryptographic algorithm. For example, encrypting the customer ID 118 with the master key 114 may result

in the cryptogram 130. Similarly, the server 106 may encrypt and/or decrypt data associated with the contactless card 104 using the corresponding master key 114.

[0036] In other embodiments, the master keys 114 of the contactless card 104 and server 106 may be used in conjunction with the counters 112 to enhance security using key diversification. The counters 112 comprise values that are synchronized between the contactless card 104 and server 106. The counter 112 may comprise a number that changes each time data is exchanged between the contactless card 104 and the server 106 (and/or the contactless card 104 and the computing device 102). When preparing to send data (e.g., to the server 106 and/or the device 102), the applet 110 of the contactless card 104 may increment the counter 112. The applet 110 of the contactless card 104 may then provide the master key 114 and counter 112 as input to a cryptographic algorithm, which produces a diversified key 116 as output. The cryptographic algorithm may include encryption algorithms, hash-based message authentication code (HMAC) algorithms, cipher-based message authentication code (CMAC) algorithms, and the like. Non-limiting examples of the cryptographic algorithm may include a symmetric encryption algorithm such as 3DES or AES107; a symmetric HMAC algorithm, such as HMAC-SHA-256; and a symmetric CMAC algorithm such as AES-CMAC. Examples of key diversification techniques are described in greater detail in United States Patent Application 16/205,119, filed November 29, 2018. The aforementioned patent application is incorporated by reference herein in its entirety.

[0037] Continuing with the key diversification example, the applet 110 may then encrypt the data (e.g., the customer ID 118 and/or any other data) using the diversified key 116 and the data as input to the cryptographic algorithm. For example, encrypting the customer ID 118 with the diversified key 116 may result in an encrypted customer ID (e.g., a cryptogram 130). In some embodiments, the cryptogram 130 is included in as a parameter of the URL 120. In other embodiments, the cryptogram 130 is not a parameter of the URL 120 but is transmitted with the URL 120 in a data package such as an NDEF file. The authentication application 142 may then read the data package including the URL 120 and cryptogram 130 via the communications interface 122 of the computing device 102.

[0038] As stated, the cryptogram 130 and the customer ID 118 may be parameters of the URL 120. For example, the URL 120 may be “http://www.example.com/redirect?param=ABC123&custID=123”. In such an example, the cryptogram 130 may correspond to the parameter “ABC123” and the customer ID 118 may correspond to the parameter “custID”. Once received by the operating system 134, the operating system 134 may open an application to process the URL 120. In some

embodiments, the URL 120 may be registered with the account application 136, which causes the operating system 134 to launch the account application 136 and provide the URL 120 as input to the account application 136. However, in other examples, the operating system 134 may launch the web browser 138 and provide the URL 120 as input to the web browser 138. Regardless of the entity launched by the operating system 134, the account application 136 and/or web browser 138 may generate a hypertext transfer protocol (HTTP) request that includes the URL 120. For the sake of clarity, the embodiments of Figs. 1A-1E are discussed using the account application 136 as a reference example. However, the embodiments are equally applicable to the web browser 138, and the use of the account application 136 should not be considered limiting of the disclosure.

[0039] Figure 1B depicts an embodiment where the account application 136 transmits an HTTP request 140 comprising the cryptogram 130 and the unencrypted customer ID 118 to the server 106. The web server 144 may generally receive the HTTP request 140 and provide the cryptogram 130 to the authentication application 142 for verification. For example, the authentication application 142 may attempt to decrypt the cryptogram 130 using a copy of the master key 114 stored by the server 106. In some embodiments, the authentication application 142 may identify the master key 114 and counter 112 using the unencrypted customer ID 118 (or other identifier) provided by the account application 136 to the server 106. In some examples, the authentication application 142 may provide the master key 114 and counter 112 as input to the cryptographic algorithm, which produces a diversified key 116 as output. The resulting diversified key 116 may correspond to the diversified key 116 of the contactless card 104, which may be used to decrypt the cryptogram 130.

[0040] Regardless of the decryption technique used, the authentication application 142 may successfully decrypt the cryptogram 130, thereby verifying or authenticating the cryptogram 130 in the request 140 (e.g., by comparing the customer ID 118 that is produced by decrypting the cryptogram 130 to a known customer ID stored in the account database 126, and/or based on an indication that the decryption using the master key 114 and/or diversified key 116 was successful). Although the keys 114, 116 are depicted as being stored in the memory 124, the keys may be stored elsewhere, such as in a secure element and/or the HSM 128. In such embodiments, the secure element and/or the HSM 128 may decrypt the cryptogram 130 using the master key 114 and/or diversified key 116 and a cryptographic function. Similarly, the secure element and/or HSM 128 may generate the diversified key 116 based on the master key 114 and counter 112 as described above.

[0041] If the decryption is successful, the authentication application 142, web server 144, and/or the ML model 150 may determine a context of the account associated with the contactless card 104. The context of the account may be based on one or more attributes of the account. The one or more attributes may include any metadata for the account holder, the account, and/or the contactless card 104 in the account database 126. Example attributes include, but are not limited to, an amount of time the account has been open, a payment state of the contactless card 104, an age of the contactless card 104, a balance of the account, a balance of the contactless card 104, and recent events associated with the account and/or contactless card 104. The events may include any type of event, such as a declined transaction, approved transaction, transaction history, activation of the contactless card, mailing (but not activation) of the contactless card 104 to the customer, etc. The context may further be determined based on an analysis of the request 140, e.g., an application that transmitted the request, metadata of the request, etc. For example, if the account application 136 generates the request 140, the account application 136 may include some metadata that allows the server 106 to determine the context. The metadata may include a current page of the account application 136 that was displayed on the computing device 102 when the contactless card 104 was tapped to the computing device 102. Embodiments are not limited in this context.

[0042] Based on the context, a resource from the web server database 146 may be selected for the server-side redirect of the URL 120. For example, as stated, the URL 120 may be a generic URL that is directed to a redirect resource of the server 106. By determining the context, a specific resource in the web server database 146 may be identified, and a redirect URL directed to the identified resource may be returned to the computing device 102. For example, if the user cannot activate their contactless card 104, the server 106 may determine to select a card activation web page from the web server database 146 and transmit a URL to the card activation web page to the computing device 102.

[0043] In some embodiments, the ML model 150 may determine the context and/or select a resource from the web server database 146 for the redirect. Generally, the ML model 150 is representative of any type of computing model. The ML model 150 may be trained to determine contexts and select resources for redirection based on the training data 152. The training data 152 includes data to train the ML model 150. For example, the training data 152 may include data describing a plurality of accounts, attributes of each account, attributes of one or more contactless cards 104 associated with each account, contexts of the accounts, and one or more redirect URLs selected for the account based on receiving a request from a client device associated with each account, where the request specifies a

URL such as the URL 120, e.g., a generic URL generated by a contactless card 104 that needs interpretation and redirect by the server 106. The redirect includes determination of a context of the account and selection of a redirect URL directed to a specific resource. Stated differently, the training data 152 includes a plurality of redirect URLs to be selected based on a plurality of URLs 120, one or more attributes of the accounts, attributes of the contactless cards 104, and/or contexts of the accounts and/or contactless cards 104. In some embodiments, the training data 152 generally includes data reflecting different times in a life cycle of an account (and/or contactless card 104) that certain account-related operations occur. Therefore, based on the training, the ML model 150 may accurately determine contexts of the accounts and/or accurately select resources in the web server database 146 as resources for redirect operations based on the stage of the life cycle of the account (and/or contactless card 104). Stated differently, the trained ML model 150 may predict what the customer is trying to do when tapping their contactless cards 104 to a computing device 102. Once trained, the ML model 150 selects redirect URLs for a URL 120 generated by the contactless cards 104 based on the prediction of what the customer is trying to do.

[0044] Returning to the decryption, if the authentication application 142 is unable to decrypt the cryptogram 130 to yield the expected result (e.g., the customer ID 118 of the account associated with the contactless card 104), the authentication application 142 does not validate the cryptogram 130. In such an example, the authentication application 142 determines to refrain from determining the context of the account and/or selecting a redirect URL. The authentication application 142 may transmit an indication of the failed decryption to the computing device 102.

[0045] Figure 1C depicts an embodiment where the authentication application 142 selects and transmits a redirect URL 158 to the computing device 102 based on successfully decrypting the cryptogram 130 and determining the context of the account. For example, if the context indicates the user is unable to activate the contactless card 104, the redirect URL 158 may be directed to a help webpage in the web server database 146 to assist users attempting to activate the contactless card 104. The redirect URL 158 may be included in an HTTP response (not pictured). In some embodiments, the HTTP response includes an HTTP response status code with a location header field, where the redirect URLs 158 is specified as at least a portion of the location header field. The HTTP response may include any suitable HTTP status code that includes a location header field. Examples of HTTP status codes that include location header fields include, but are not limited to, the HTTP 301 status code, HTTP 302 status code, HTTP 303 status code, HTTP 307 status code, HTTP 308 status code, and the like. However, in other embodiments, the redirect URL 158 is

transmitted according to a different format and/or protocol. In some embodiments, the redirect URL 158 includes one or more additional parameters, such as an authentication token and/or an identifier of a page of the account application 136 to be opened responsive to receiving the redirect URL 158.

[0046] As stated, the authentication application 142 may generate an authentication token to be included as a parameter of the redirect URL 158. The authentication token may be generated responsive to the decryption of the cryptogram 130. In some embodiments, the authentication application 142 further generates the authentication token based on a determination that the account associated with the contactless card 104 has been active for a period of time that exceeds a time threshold (e.g., 1 month, 3 months, etc.). As another example, the authentication application 142 may further generate the authentication token based on a device identifier of the computing device 102, received from the account application 136 and/or web browser 138, matching a device identifier associated with the account in the account database 126. The device identifier may be any unique identifier, such as a media access control (MAC) address of the computing device 102, a unique alphanumeric string, and the like. Stated differently, if the device identifier matches the stored identifier in the account database 126, the computing device 102 may be a “trusted” device, and the authentication application 142 may generate the authentication token based on the computing device 102 being a trusted device.

[0047] The authentication token may generally allow the account application 136 to log into the account associated with the contactless card 104 in the account database 126. In some embodiments, the authentication token includes a hash value and one or more attributes of the account. For example, the account application 136 may extract the authentication token. The account application 136 may then use the token to access the account, e.g., by verifying the token locally and/or transmitting the authentication token to the server 106. For example, the server 106 may compare the authentication token received from the account application 136 to the account token generated by the authentication application 142. If the comparison results in a match, the server 106 may transmit one or more attributes of the account in the account database 126 to the account application 136, which may process and/or otherwise display the one or more attributes on the computing device 102.

[0048] As shown, the authentication application 142 may further transmit a decryption result 148 to the computing device 102. The decryption result 148 generally reflects whether or not the cryptogram 130 was decrypted, e.g., with a flag, bit, or any other suitable indication. In the example depicted in Figure 1C, the decryption result 148 may indicate the

server 106 decrypted the cryptogram 130. Doing so may allow the account application 136 to determine that the cryptogram 130 was successfully decrypted prior to following the redirect URL 158, thereby improving security. The account application 136 and/or the web browser 138 may receive the redirect URL 158 and/or the decryption result 148. In at least one embodiment, the account application 136 and/or web browser 138 automatically follows the redirect URL 158 upon receipt, e.g., by generating a new request comprising the redirect URL 158.

[0049] Figure 1D depicts an embodiment where the account application 136 transmits a request 154 comprising the redirect URL 158 to the server 106. Once received, the web server 144 may identify a resource in the web server database 146 associated with the redirect URL 158. For example, the resource in the web server database 146 may include the help page for activating the contactless card. Embodiments are not limited in this context. If the redirect URL 158 includes an authentication token and/or page identifier, the account application 136 may extract the token for account authentication purposes and/or open the page of the account application 136 associated with the page identifier. In some embodiments, the account application 136 may decode the authentication token to identify one or more attributes of the account in the account database 126.

[0050] Figure 1E depicts an embodiment where the web server 144 identifies a resource 160 associated with the redirect URL 158. The web server 144 may then transmit the resource 160 to the computing device 102. The resource 160 may be transmitted as part of an HTTP response. As shown, the account application 136 may then access the resource 160, thereby displaying the resource 160 on a display of the computing device 102. As stated, in some embodiments, the web browser 138 may receive and render the resource 160.

[0051] Advantageously, the system 100 provides server-side redirection of the URL 120 generated by the contactless card 104. Doing so improves the performance of the system 100. For example, the performance of the contactless card 104 may be improved by reducing the amount of processing resources and/or memory resources needed to generate and/or store URLs 120. Furthermore, by configuring the contactless card to generate a limited number of URLs 120 that can be redirected by the server 106 based on context, the contactless card 104 can support a one-to-many relationship between URLs and available resources in the web server database 146. Furthermore, doing so may remove the need to periodically update the logic in the contactless card 104 to support new URLs, e.g., as new web resources are added, removed, and/or modified. Similarly, by leveraging cryptograms 130 included in the URLs 120 by the contactless cards, the system 100 may securely verify

the identity of the user with minimal risk of fraudulent activity. Furthermore, the system 100 ensures that redirect operations are only performed when the user has access to a contactless card that generates a valid cryptogram for verification by the server 106.

[0052] Figure 2A is a schematic 200a illustrating an embodiment where a contactless card 104 is tapped to a computing device 102. While the computing device 102 is depicted as outputting a screen (e.g., a home screen) of the operating system 134, the computing device 102 may generally be in any state. For example, the user may be using another application, such as the account application 136 and/or the web browser 138, when tapping the contactless card 104 to the computing device 102.

[0053] As stated, when the contactless card 104 is tapped to the computing device 102, the applet 110 may generate a cryptogram 130 and URL 120. In some embodiments, the cryptogram 130 is a parameter of the URL 120. The applet 110 may further include an identifier, such as an unencrypted customer ID 118, an identifier of the contactless card 104, and the like, as a parameter of the URL 120. Regardless of whether the cryptogram 130 and/or unencrypted identifier are parameters of the URL 120, the cryptogram 130, unencrypted identifier, and the URL 120 may be included in a data package, such as an NDEF file, that is read by the computing device 102. As shown, responsive to receiving the data package, the operating system 134 may launch an application, as the URL 120 (or a portion thereof) may be registered with the account application 136 and/or web browser 138.

[0054] Figure 2B is a schematic 200b illustrating an embodiment where the account application 136 is opened responsive to the operating system 134 receiving the URL 120 from the contactless card 104. As shown, the account application 136 transmits the URL 120, cryptogram 130, and unencrypted customer ID 118 to the server 106. In at least one embodiment, the account application 136 transmits the URL 120, cryptogram 130, and customer ID 118 to the server 106 as at least a portion of a first HTTP request. The authentication application 142 may then attempt to decrypt the cryptogram 130 as described in greater detail above. If the decryption is successful, the authentication application 142 may instruct the ML model 150 to determine a context of the account associated with the customer ID 118 and output a redirect URL that is directed to a resource 160 in the web server database 146. The server 106 may then transmit the redirect URL 158 to the computing device 102.

[0055] In other embodiments, the ML model 150 may output the resource 160, and the web server 144 determines the redirect URL associated with the resource 160, e.g., based on a

mapping of URLs to resources (not pictured) stored by the web server 144. In another embodiment, the ML model 150 may output the context, such as “account activation,” “transaction declined,” etc. The web browser 138 may then identify one or more resources 160 in the web server database 146 that are associated with the context, e.g., based on a mapping of contexts to resources (not pictured) stored by the web server 144. For example, a plurality of resources 160 may be associated with the “transaction declined” context generated by the ML model 150. In such an example, the web server 144 may select one of the plurality of resources and return a redirect URL 158 directed to the selected resource to the computing device 102.

[0056] Figure 2C is a schematic 200c illustrating an embodiment where the account application 136 displays a page 202. The page 202 may correspond to a resource 160 associated with a redirect URL 158 received from the server 106 as described above with reference to Figure 2B. Once received, the account application 136 may generate a new request specifying the redirect URL 158 and transmit the new request to the server 106. The server 106 may respond with the resource 160, e.g., the page 202. In some embodiments, however, the account application 136 may store the page 202 locally. In such embodiments, the account application 136 need not generate a new request for the page 202. Instead, the redirect URL 158 may include an identifier of the page 202, and the account application 136 may open the page 202 based on identifying the identifier of the page 202 in the redirect URL 158. Illustratively, as shown, the page 202 corresponds to an account detail page, where one or more attributes of one or more accounts are displayed on the computing device 102.

[0057] Operations for the disclosed embodiments may be further described with reference to the following figures. Some of the figures may include a logic flow. Although such figures presented herein may include a particular logic flow, it can be appreciated that the logic flow merely provides an example of how the general functionality as described herein can be implemented. Further, a given logic flow does not necessarily have to be executed in the order presented unless otherwise indicated. In addition, the given logic flow may be implemented by a hardware element, a software element executed by a processor, or any combination thereof. The embodiments are not limited in this context.

[0058] Figure 3 illustrates an embodiment of a logic flow, or routine, 300. The logic flow 300 may be representative of some or all of the operations executed by one or more embodiments described herein. For example, the logic flow 300 may include some or all of the operations for server-side redirect of a URL generated by a contactless card 104. Embodiments are not limited in this context.

[0059] In block 302, routine 300 receives, by a server 106 from a client computing device 102, a first hypertext transfer protocol (HTTP) request comprising a URL 120, wherein a first parameter of the URL 120 comprises a cryptogram 130 generated by a contactless card 104 and a second parameter of the URL 120 comprises a customer ID 118 of an account associated with the contactless card 104. In block 304, the routine 300 decrypts, by the server 106, the cryptogram 130. In block 306, routine 300 determines, by the server 106, a context of the account based on one or more attributes of the account. In at least one embodiment, the ML model 150 determines the context. In block 308, routine 300 selects, by the server 106 based on the decryption of the cryptogram and the determined context of the account, a first redirect URL 158 of a plurality of redirect URLs. In block 310, routine 300 transmits, by the server 106 to the computing device 102, an HTTP response comprising the redirect URL 158. In block 312, routine 300 receives, by the server 106 from the computing device 102, a second HTTP request comprising the redirect URL 158. In block 314, routine 300 transmits, by the server 106 to the computing device 102, an HTTP response comprising a resource 160 at the redirect URL 158.

[0060] Figure 4 illustrates an embodiment of a logic flow, or routine, 400. The logic flow 400 may be representative of some or all of the operations executed by one or more embodiments described herein. For example, the logic flow 400 may include some or all of the operations for server-side redirect of a URL generated by a contactless card 104. Embodiments are not limited in this context.

[0061] In block 402, routine 400 analyzes, by the server 106, a URL specified in a request. For example, the URL may be the URL 120 comprising the cryptogram 130 received at block 302 of Figure 3. In block 404, routine 400 determines, by the server 106, based on the analysis, an application generating the request. For example, the account application 136 and/or the web browser 138 may transmit the request, and the server 106 may determine that the account application 136 or the web browser 138 transmitted the request. In block 406, routine 400 determines, by the server 106, one or more attributes of an account associated with the request, e.g., based on the customer ID 118 in the request. In block 410, routine 400 selects a redirect URL 158 based on the analysis, the application generating the request, and the one or more attributes of the account. The redirect URL 158 may be transmitted to the computing device 102, which may follow the redirect URL 158 to load the corresponding resource 160.

[0062] Figure 5 illustrates an embodiment of a logic flow, or routine, 500. The logic flow 500 may be representative of some or all of the operations executed by one or more embodiments described herein. For example, the logic flow 500 may include some or all of

the operations for server-side redirect of a URL using machine learning, where the URL is generated by a contactless card 104. Embodiments are not limited in this context.

[0063] In block 502, routine 500 receives, by an ML model 150, a request specifying a URL, and one or more attributes of an account associated with the request. In block 504, routine 500 determines, by the ML model 150, a context of the account. At block 506, routine 500 processes, by the ML model 150, the URL, the one or more attributes of the account, and the context of the account. In block 508, routine 500 generates, by the ML model based on the processing, a first redirect URL 158 of a plurality of redirect URLs. The redirect URL 158 may be transmitted to the computing device 102, which may follow the redirect URL 158 to load the corresponding resource 160.

[0064] FIG. 6A is a schematic 600 illustrating an example configuration of a contactless card 104, which may include a payment card, such as a credit card, debit card, or gift card, issued by a service provider as displayed as service provider indicia 602 on the front or back of the contactless card 104. In some examples, the contactless card 104 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 104 may include a substrate 604, 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 104 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 104 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.

[0065] The contactless card 104 may also include identification information 606 displayed on the front and/or back of the card, and a contact pad 608. The contact pad 608 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 104 may also include processing circuitry, antenna and other components as will be further discussed in FIG. 6B. These components may be located behind the contact pad 608 or elsewhere on the substrate 604, e.g. within a different

layer of the substrate 604, and may electrically and physically coupled with the contact pad 608. The contactless card 104 may also include a magnetic strip or tape, which may be located on the back of the card (not shown in FIG. 6A). The contactless card 104 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.

[0066] As illustrated in FIG. 2, the contact pad 608 of contactless card 104 may include processing circuitry 610 for storing, processing, and communicating information, including a processor 612, a memory 108, and one or more communications interfaces 122. It is understood that the processing circuitry 610 may contain additional components, including processors, memories, error and parity/CRC checkers, data encoders, anticollision algorithms, controllers, command decoders, security primitives and tamperproofing hardware, as necessary to perform the functions described herein.

[0067] The memory 108 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 104 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 108 may be encrypted memory utilizing an encryption algorithm executed by the processor 612 to encrypted data.

[0068] The memory 108 may be configured to store one or more applet 110, one or more counters 112, a customer ID 118, the master key 114, diversified key 116, and Redirect URLs 120. The one or more applet 110 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 110 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 112 may comprise a numeric counter sufficient to store an integer. The customer ID 118 may comprise a unique alphanumeric identifier assigned to a user of the contactless card 104, and the identifier may distinguish the user of the contactless card from other contactless card users. In some examples, the customer ID 118 may identify both a customer and an account assigned to that customer and may further identify the contactless card 104 associated with the customer's account.

[0069] The processor 612 and memory elements of the foregoing exemplary embodiments are described with reference to the contact pad 608, but the present disclosure is not limited thereto. It is understood that these elements may be implemented outside of the contact pad 608 or entirely separate from it, or as further elements in addition to processor 612 and memory 108 elements located within the contact pad 608.

[0070] In some examples, the contactless card 104 may comprise one or more antenna(s) 614. The one or more antenna(s) 614 may be placed within the contactless card 104 and around the processing circuitry 610 of the contact pad 608. For example, the one or more antenna(s) 614 may be integral with the processing circuitry 610 and the one or more antenna(s) 614 may be used with an external booster coil. As another example, the one or more antenna(s) 614 may be external to the contact pad 608 and the processing circuitry 610.

[0071] In an embodiment, the coil of contactless card 104 may act as the secondary of an air core transformer. The terminal may communicate with the contactless card 104 by cutting power or amplitude modulation. The contactless card 104 may infer the data transmitted from the terminal using the gaps in the power connection of the contactless card 104, which may be functionally maintained through one or more capacitors. The contactless card 104 may communicate back by switching a load on the coil of the contactless card 104 or load modulation. Load modulation may be detected in the terminal's coil through interference. More generally, using the antenna(s) 614, processor 612, and/or the memory 108, the contactless card 104 provides a communications interface to communicate via NFC, Bluetooth, and/or Wi-Fi communications.

[0072] As explained above, contactless card 104 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 110 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 110 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 computing device 102 or point-of-sale terminal), and produce an NDEF message that comprises a cryptographically secure OTP encoded as an NDEF text tag. The NDEF message may include the URL 120, the cryptogram 130, and any other data.

[0073] One example of an NDEF OTP is an NDEF short-record layout (SR=1). In such an example, one or more applet 110 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 110 may be configured to add one or more static tag records in addition to the OTP record.

[0074] In some examples, the one or more applet 110 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 110, 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.

[0075] In some examples, the contactless card 104 and server may include certain data such that the card may be properly identified. The contactless card 104 may include one or more unique identifiers (not pictured). Each time a read operation takes place, the counter 112 may be configured to increment. In some examples, each time data from the contactless card 104 is read (e.g., by a mobile device), the counter 112 is transmitted to the server for validation and determines whether the counter 112 are equal (as part of the validation) to a counter of the server.

[0076] The one or more counter 112 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 112 has been read or used or otherwise passed over. If the counter 112 has not been used, it may be replayed. In some examples, the counter that is incremented on the contactless card 104 is different from the counter that is incremented for transactions. The contactless card 104 is unable to determine the application transaction counter 112 since there is no communication between applets 110 on the contactless card 104. In some examples, the contactless card 104 may comprise a first applet 440-1, which may be a transaction applet, and a second applet 440-2. Each applet 440-1 and 440-2 may comprise a respective counter 112.

[0077] In some examples, the counter 112 may get out of sync. In some examples, to account for accidental reads that initiate transactions, such as reading at an angle, the counter 112 may increment but the application does not process the counter 112. In some examples, when the mobile device 10 is woken up, NFC may be enabled and the computing device 102 may be configured to read available tags, but no action is taken responsive to the reads.

[0078] To keep the counter 112 in sync, an application, such as a background application, may be executed that would be configured to detect when the computing device 102 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 112 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 112 may be configured to move forward. But if within a different threshold number, for example within 10 or 1000, 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 112 increases in the appropriate sequence, then it possible to know that the user has done so.

[0079] The key diversification technique described herein with reference to the counter 112, 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.

[0080] During the creation process of the contactless card 104, 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 104. 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.

[0081] 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 104 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).

[0082] 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.

[0083] 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.

[0084] FIG. 7 illustrates an NDEF short-record layout (SR=1) data structure 700 according to an example embodiment. One or more applets 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 applets may be configured to add one or more static tag records in addition to the OTP record. Exemplary tags include, without limitation, Tag type: well known type, text, encoding English (en); Applet ID: D2760000850101; Capabilities: read-only access; Encoding: the authentication message may be encoded as ASCII hex; type-length-value (TLV) data may be provided as a personalization parameter that may be used to generate the NDEF message. In an embodiment, the authentication template may comprise the first record, with a well-known index for providing the actual dynamic authentication data. The data structure 700 may include the URL 120, the cryptogram 130, and any other data provided by the applet 110.

[0085] FIG. 8 illustrates an embodiment of an exemplary computer architecture 800 suitable for implementing various embodiments as previously described. In one embodiment, the computer architecture 800 may include or be implemented as part of computing architecture 100.

[0086] 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 800. 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.

[0087] The computer architecture 800 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 computing architecture 800.

[0088] As shown in FIG. 8, the computer architecture 800 includes a computer 812 comprising a processor 802, a system memory 804 and a system bus 806. The processor 802 can be any of various commercially available processors. The computer 812 may be representative of the computing device 102 and/or the server 106.

[0089] The system bus 806 provides an interface for system components including, but not limited to, the system memory 804 to the processor 802. The system bus 806 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 806 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.

[0090] The computer architecture 800 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-writeable 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 performance of the operations described herein.

[0091] The system memory 804 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. 8, the system memory 804 can include non-volatile 808 and/or volatile 810. A basic input/output system (BIOS) can be stored in the non-volatile 808.

[0092] The computer 812 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 814, a magnetic disk drive 816 to read from or write to a removable magnetic disk 818, and an optical disk drive 820 to read from or write to a removable optical disk 822 (e.g., a CD-ROM or DVD). The hard disk drive 814, magnetic disk drive 816 and optical disk drive 820 can be connected to system bus 806 the by an HDD interface 824, and FDD interface 826 and an optical disk drive interface 828, respectively. The HDD interface 824 for external drive implementations can include at least one or both of Universal Serial Bus (USB) and IEEE 1394 interface technologies.

[0093] 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 808, and volatile 810, including an operating system 830, one or more applications 832, other program modules 834, and program data 836. In one embodiment, the one or more applications 832, other program modules 834, and program data 836 can include, for example, the various applications and/or components of the system 100.

[0094] A user can enter commands and information into the computer 812 through one or more wire/wireless input devices, for example, a keyboard 838 and a pointing device, such

as a mouse 840. Other input devices may include microphones, infra-red (IR) remote controls, radio-frequency (RF) remote controls, game pads, stylus pens, card readers, dongles, finger print 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 802 through an input device interface 842 that is coupled to the system bus 806 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.

[0095] A monitor 844 or other type of display device is also connected to the system bus 806 via an interface, such as a video adapter 846. The monitor 844 may be internal or external to the computer 812. In addition to the monitor 844, a computer typically includes other peripheral output devices, such as speakers, printers, and so forth.

[0096] The computer 812 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) 848. The remote computer(s) 848 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 812, although, for purposes of brevity, only a memory and/or storage device 850 is illustrated. The logical connections depicted include wire/wireless connectivity to a local area network 852 and/or larger networks, for example, a wide area network 854. 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.

[0097] When used in a local area network 852 networking environment, the computer 812 is connected to the local area network 852 through a wire and/or wireless communication network interface or network adapter 856. The network adapter 856 can facilitate wire and/or wireless communications to the local area network 852, which may also include a wireless access point disposed thereon for communicating with the wireless functionality of the network adapter 856.

[0098] When used in a wide area network 854 networking environment, the computer 812 can include a modem 858, or is connected to a communications server on the wide area network 854 or has other means for establishing communications over the wide area network 854, such as by way of the Internet. The modem 858, which can be internal or

external and a wire and/or wireless device, connects to the system bus 806 via the input device interface 842. In a networked environment, program modules depicted relative to the computer 812, or portions thereof, can be stored in the remote memory and/or storage device 850. 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.

[0099] The computer 812 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, ac, ax, 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).

[0100] The various elements of the devices as previously described with reference to FIGS. 1-7 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.

[0101] One or more aspects of at least one embodiment may be implemented by representative instructions stored on a machine-readable medium which represents various logic within the processor, which when read by a machine causes the machine to fabricate logic to perform the techniques described herein. Such representations, known as “IP cores” may be stored on a tangible, machine readable medium and supplied to various customers or manufacturing facilities to load into the fabrication machines that make the logic or processor. Some embodiments may be implemented, for example, using a machine-readable medium or article which may store an instruction or a set of instructions that, if executed by a machine, may cause the machine to perform a method and/or operations in accordance with the embodiments. Such a machine may include, for example, any suitable processing platform, computing platform, computing device, processing device, computing system, processing system, computer, processor, or the like, and may be implemented using any suitable combination of hardware and/or software. The machine-readable medium or article may include, for example, any suitable type of memory unit, memory device, memory article, memory medium, storage device, storage article, storage medium and/or storage unit, for example, memory, removable or non-removable media, erasable or non-erasable media, writeable or re-writeable media, digital or analog media, hard disk, floppy disk, Compact Disk Read Only Memory (CD-ROM), Compact Disk Recordable (CD-R), Compact Disk Rewriteable (CD-RW), optical disk, magnetic media, magneto-optical media, removable memory cards or disks, various types of Digital Versatile Disk (DVD), a tape, a cassette, or the like. The instructions may include any suitable type of code, such as source code, compiled code, interpreted code, executable code, static code, dynamic code, encrypted code, and the like, implemented using any suitable high-level, low-level, object-oriented, visual, compiled and/or interpreted programming language.

[0102] The foregoing description of example embodiments has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the present disclosure to the precise forms disclosed. Many modifications and variations are possible in light of this disclosure. It is intended that the scope of the present disclosure be limited not by this detailed description, but rather by the claims appended hereto. Future filed applications claiming priority to this application may claim the disclosed subject matter in a different manner, and may generally include any set of one or more limitations as variously disclosed or otherwise demonstrated herein.

CLAIMS

What is claimed is:

1. A method, comprising:
 - receiving, by a server from a client device, a first hypertext transfer protocol (HTTP) request comprising a uniform resource locator (URL), wherein a first parameter of the URL comprises a cryptogram generated by a contactless card and a second parameter of the URL comprises a customer identifier of an account associated with the contactless card;
 - decrypting, by the server, the cryptogram;
 - determining, by the server, a context of the account based on one or more attributes of the account;
 - selecting, by the server based on the decryption of the cryptogram and the determined context of the account, a first redirect URL of a plurality of redirect URLs;
 - transmitting, by the server to the client device, an HTTP response comprising the redirect URL;
 - receiving, by the server from the client device, a second HTTP request comprising the redirect URL; and
 - transmitting, by the server to the client device, a second HTTP response comprising a resource at the redirect URL.
2. The method of claim 1, wherein a machine learning (ML) model selects the redirect URL based on input comprising the first HTTP request, the URL, and the context of the account, wherein the ML model is generated based on training data describing a plurality of redirect operations and a plurality of contexts for a plurality of accounts.
3. The method of claim 1, wherein the one or more attributes of the account comprise one or more of: (i) a payment state of the contactless card, wherein the payment state comprises one of an active payment state or an inactive payment state, (ii) an amount of time the account has been active, and (iii) a rejected transaction using the contactless card in a transaction history of the account.
4. The method of claim 1, further comprising:
 - determining, by the server based on the first HTTP request, whether the first HTTP request was generated by a web browser of the client device or an application of the client device, wherein the web browser is distinct from the application, wherein the server selects

the redirect URL based on whether the request was generated by the client device or the application.

5. The method of claim 4, wherein the server determines that the first HTTP request was generated by the application, the method further comprising:

generating, by the server based on the determination that the application generated the first HTTP request, an authentication token for the account, wherein the server includes the authentication token as a first parameter of the redirect URL;

receiving, by the server from the application, the authentication token; and

transmitting, by the server to the application, one or more attributes of the account based on the receipt of the authentication token from the application.

6. The method of claim 5, further comprising:

determining, by the server based on the context and the redirect URL, an identifier of a page of the application, wherein the server includes the identifier of the page of the application as a second parameter of the redirect URL, wherein the application opens the page of the application based at least in part on the second parameter of the URL.

7. The method of claim 1, wherein the decrypting comprises:

incrementing, by the server, a counter value associated with the contactless card;

encrypting, by the server, the counter value and a master key to generate a diversified key; and

decrypting, by the server, the cryptogram based on the diversified key.

8. A non-transitory computer-readable storage medium having computer-readable program code embodied therewith, the computer-readable program code executable by a processor to cause the processor to:

receive, by a server from a client device, a first hypertext transfer protocol (HTTP) request comprising a uniform resource locator (URL), wherein a first parameter of the URL comprises a cryptogram generated by a contactless card and a second parameter of the URL comprises a customer identifier of an account associated with the contactless card;

decrypt, by the server, the cryptogram;

determine, by the server, a context of the account based on one or more attributes of the account;

select, by the server based on the decryption of the cryptogram and the determined context of the account, a first redirect URL of a plurality of redirect URLs;

transmit, by the server to the client device, an HTTP response comprising the redirect URL;

receive, by the server from the client device, a second HTTP request comprising the redirect URL; and

transmit, by the server to the client device, a second HTTP response comprising a resource at the redirect URL.

9. The medium of claim 8, wherein a machine learning (ML) model selects the redirect URL based on input comprising the first HTTP request, the URL, and the context of the account, wherein the ML model is generated based on training data describing a plurality of redirect operations and a plurality of contexts for a plurality of accounts.

10. The medium of claim 8, wherein the one or more attributes of the account comprise one or more of: (i) a payment state of the contactless card, wherein the payment state comprises one of an active payment state or an inactive payment state, (ii) an amount of time the account has been active, and (iii) a rejected transaction using the contactless card in a transaction history of the account.

11. The medium of claim 8, the computer-readable program code executable by the processor to cause the processor to:

determine, by the server based on the first HTTP request, whether the first HTTP request was generated by a web browser of the client device or an application of the client device, wherein the web browser is distinct from the application, wherein the server selects the redirect URL based on whether the request was generated by the client device or the application.

12. The medium of claim 11, the computer-readable program code executable by the processor to cause the processor to:

generate, by the server based on a determination that the application generated the first HTTP request, an authentication token for the account, wherein the server includes the authentication token as a first parameter of the redirect URL;

receive, by the server from the application, the authentication token; and

transmit, by the server to the application, one or more attributes of the account based on the receipt of the authentication token from the application.

13. The medium of claim 12, the computer-readable program code executable by the processor to cause the processor to:

determine, by the server based on the context and the redirect URL, an identifier of a page of the application, wherein the server includes the identifier of the page of the application as a second parameter of the redirect URL.

14. The medium of claim 8, the computer-readable program code executable by the processor to decrypt the cryptogram to cause the processor to:

increment, by the server, a counter value associated with the contactless card;

encrypt, by the server, the counter value and a master key to generate a diversified key; and

decrypt, by the server, the cryptogram based on the diversified key.

15. An apparatus, comprising:

a processor; and

a memory storing instructions which when executed by the processor, cause the processor to:

receive, by a server from a client device, a first hypertext transfer protocol (HTTP) request comprising a uniform resource locator (URL), wherein a first parameter of the URL comprises a cryptogram generated by a contactless card and a second parameter of the URL comprises a customer identifier of an account associated with the contactless card;

decrypt, by the server, the cryptogram;

determine, by the server, a context of the account based on one or more attributes of the account;

select, by the server based on the decryption of the cryptogram and the determined context of the account, a first redirect URL of a plurality of redirect URLs;

transmit, by the server to the client device, an HTTP response comprising the redirect URL;

receive, by the server from the client device, a second HTTP request comprising the redirect URL; and

transmit, by the server to the client device, a second HTTP response comprising a resource at the redirect URL.

16. The apparatus of claim 15, wherein a machine learning (ML) model executing on the processor selects the redirect URL based on input comprising the first HTTP request, the URL, and the context of the account, wherein the ML model is generated based on training

data describing a plurality of redirect operations and a plurality of contexts for a plurality of accounts.

17. The apparatus of claim 15, wherein the one or more attributes of the account comprise one or more of: (i) a payment state of the contactless card, wherein the payment state comprises one of an active payment state or an inactive payment state, (ii) an amount of time the account has been active, and (iii) a rejected transaction in a transaction history of the account.

18. The apparatus of claim 15, the memory storing instructions which when executed by the processor, cause the processor to:

determine, by the server based on the first HTTP request, whether the first HTTP request was generated by a web browser of the client device or an application of the client device, wherein the web browser is distinct from the application, wherein the server selects the redirect URL based on whether the request was generated by the client device or the application.

19. The apparatus of claim 18, the memory storing instructions which when executed by the processor, cause the processor to:

generate, by the server based on a determination that the application generated the first HTTP request, an authentication token for the account, wherein the server includes the authentication token as a first parameter of the redirect URL;

receive, by the server from the application, the authentication token; and

transmit, by the server to the application, one or more attributes of the account based on the receipt of the authentication token from the application.

20. The apparatus of claim 19, the memory storing instructions which when executed by the processor, cause the processor to:

determine, by the server based on the context and the redirect URL, an identifier of a page of the application, wherein the server includes the identifier of the page of the application as a second parameter of the redirect URL.