

ONE-TAP PAYMENT USING A CONTACTLESS CARD

TECHNICAL FIELD

[0001] Embodiments herein generally relate to computing platforms, and more specifically, to tapping a contactless card to a computing device to perform one-tap payment.

RELATED APPLICATIONS

[0002] This application claims priority to U.S. Patent Application Serial No. 16/265,974, titled “ONE-TAP PAYMENT USING A CONTACTLESS CARD” filed on February 1, 2019. The contents of the aforementioned application are incorporated herein by reference in their entirety.

BACKGROUND

[0003] Users often complete transactions using online software platforms. Conventionally, users must manually enter all relevant payment information in the online software platform to complete a transaction. However, users often make mistakes when entering their payment information. Furthermore, transmitting payment information via a user device may pose security risks.

SUMMARY

[0004] According to an aspect, there is provided a method comprising: receiving, by an authentication application executing on a processor from a merchant application executing on a client device: (i) a merchant identifier of a merchant associated with the merchant application, (ii) a transaction identifier of a transaction, and (iii) encrypted data, the encrypted data generated by a contactless card; verifying, by the authentication application, the encrypted data by decrypting the encrypted data; generating, by a virtual account number generator executing on the processor, a virtual account number based on the verification of the encrypted data; and transmitting, by the virtual account number generator, the merchant identifier, the transaction identifier, the virtual account number, an expiration date associated with the virtual account number, and a card verification value

(CVV) associated with the virtual account number to a merchant server associated with the merchant to process the transaction using the transaction identifier, the virtual account number, the expiration date, and the CVV.

[0005] According to another aspect, there is provided a computer - readable storage medium including instructions that when executed by a processor, cause the processor to: receive, by an authentication application executing on the processor from a merchant application executing on a client device: (i) a merchant identifier of a merchant associated with the merchant application, (ii) a transaction identifier of a transaction, and (iii) encrypted data, the encrypted data generated by a contactless card; verify, by the authentication application, the encrypted data by decrypting the encrypted data; generate, by a virtual account number generator executing on the processor, a virtual account number based on the verification of the encrypted data; and transmit, by the virtual account number generator, the merchant identifier, the transaction identifier, the virtual account number, an expiration date associated with the virtual account number, and a card verification value (CVV) associated with the virtual account number to a merchant server associated with the merchant to process the transaction using the transaction identifier, the virtual account number, the expiration date, and the CVV.

[0006] According to another aspect, there is provided a computing apparatus comprising a processor and a memory storing instructions that, when executed by the processor, cause the processor to: receive, by an authentication application executing on a processor from a merchant application executing on a client device: (i) a merchant identifier of a merchant associated with the merchant application, (ii) a transaction identifier of a transaction, and (iii) encrypted data, the encrypted data generated by a contactless card; verify, by the authentication application, the encrypted data by decrypting the encrypted data; generate, by a virtual account number generator executing on the processor, a virtual account number based on the verification of the encrypted data; and transmit, by the virtual account number generator, the merchant identifier, the transaction identifier, the virtual account number, an expiration date associated with the virtual account number, and a card verification value (CVV) associated with the virtual account number to a merchant server associated with the merchant to process the transaction using the transaction identifier, the virtual account number, the expiration date, and the CVV.

[0007] Embodiments disclosed herein provide systems, methods, articles of manufacture, and computer-readable media for tapping a contactless card to a computing device for one-tap payment. In one example, an authentication application may receive, from a merchant application of a client device, a merchant identifier, a transaction identifier of a transaction, encrypted data, and a location of the client device. The authentication application may verify the encrypted data by decrypting the encrypted data based at least in part on a private key for a contactless card. The authentication application may determine that the contactless card has been used to make a previous purchase with the merchant and that the location of the client device is within a threshold distance of a known location associated with the contactless card. A virtual account number generator may generate a virtual account number and transmit the merchant identifier, the transaction identifier, the virtual account number, an expiration date, and a card verification value (CVV) to a merchant server associated with the merchant to process the transaction using the transaction identifier, the virtual account number, the expiration date, and the CVV.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Figures 1A-1C illustrate embodiments of a system to tap a contactless card to a computing device for one-tap payment.

[0009] Figures 2A-2D illustrate embodiments of tapping a contactless card to a computing device for one-tap payment.

[0010] Figure 3 illustrates an embodiment of a first logic flow.

[0011] Figure 4 illustrates an embodiment of a second logic flow.

[0012] Figure 5 illustrates an embodiment of a third logic flow.

[0013] Figure 6 illustrates an embodiment of a fourth logic flow.

[0014] Figure 7 illustrates an embodiment of a computing architecture.

DETAILED DESCRIPTION

[0015] Embodiments disclosed herein provide secure techniques to use a contactless card for one-tap payment. For example, when using a merchant application executing on a computing device, a user may select one or more items for purchase. The user may then tap their contactless card to the computing device, which may cause the contactless card to come within communications range of the computing device. Doing so causes the contactless card to generate encrypted data which is transmitted to the computing device. The merchant application may receive the encrypted data generated by the contactless card and transmit the encrypted data to an authentication server for validation. The merchant application may further transmit an indication of a merchant identifier and/or a transaction identifier with the encrypted data. Once validated, the authentication server may instruct a virtual account number server to generate card data for the account associated with the contactless card. The card data may include a virtual account number, an expiration date, and a card verification value (CVV). A virtual account number may be an account number that is different than the account number associated with the contactless card. The generated card data may then be transmitted with the merchant identifier and the transaction identifier to a merchant server associated with the merchant. The virtual account number server and/or the authentication server may also provide a name of the user, a billing address of the user, and/or a shipping address of the user to the merchant server. The merchant server may then complete the transaction using the received data. The merchant server may then transmit a confirmation to the computing device which is outputted for display by the merchant application to the user. From the point of view of the user of the computing device, once the card is tapped to the computing device, the transaction is completed without requiring additional user input, and the user receives a positive confirmation message reflecting that the transaction has been completed.

[0016] Advantageously, embodiments disclosed herein improve the security of all devices and associated data. For example, by transmitting card data between secure servers, the risks associated with storing data on a user device and transmitting the data using the user device are avoided. Furthermore, conventional approaches require the user to manually enter card data into a form. However, doing so may allow other users or devices to capture the card data as the user enters the card data into the form. By

eliminating the need for the user to manually enter card data, the security of the card data is enhanced. However, doing so may allow other users or devices to capture the card data as the user enters the card data into the form. By eliminating the need for the user to manually enter card data into the form, the security of the card data is enhanced. Furthermore, the validation performed by the authentication server provides an additional safeguard to ensure that the correct card data is being entered into the form. Additionally, the generation and filling of virtual card numbers into the form protects the security of the actual account number of the contactless card, as conventional solutions require providing the actual account number of the contactless card into the form.

[0017] 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.

[0018] 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.

[0019] 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.

[0020] Figure 1A depicts a schematic of an exemplary system 100, consistent with disclosed embodiments. As shown, the system 100 includes one or more contactless cards 101, one or more mobile devices 110, an authentication server 120, a virtual account number server 140, and one or more merchant servers 150. The contactless cards 101 are representative of any type of payment cards, such as a credit card, debit card, ATM card, gift card, and the like. The contactless cards 101 may comprise one or more communications interfaces 107, such as a radio frequency identification (RFID) chip, configured to communicate with the mobile devices 110 via NFC, the EMV standard, or other short-range protocols in wireless communication. Although NFC is used as an example communications protocol, the disclosure is equally applicable to other types of wireless communications, such as the EMV standard, Bluetooth, and/or Wi-Fi. The mobile devices 110 are representative of any type of network-enabled computing devices, such as smartphones, tablet computers, wearable devices, laptops, portable gaming devices, and the like. The servers 120, 140, 150 are representative of any type of computing device, such as a server, workstation, compute cluster, cloud computing platform, virtualized computing system, and the like.

[0021] As shown, a memory 111 of the mobile device 110 includes an instance of an operating system (OS) 112. Example operating systems 112 include the Android® OS, iOS®, Linux®, and Windows® operating systems. As shown, the OS 112 includes one or more merchant applications 113. The merchant applications 113 are representative of any type of application where a user may provide payment information to complete a transaction. For example, the merchant applications 113 may allow the users to select and purchase goods, products, and/or services. In one embodiment, the merchant applications 113 may be web-based applications that are accessed using a web browser (not pictured). For example, in web-based applications, the web browser may access a website of the merchant and/or progressive web applications provided by the merchant.

[0022] As shown, the merchant applications 113 include a one-tap module 114, a merchant identifier 115, and a transaction identifier 116. As described in greater detail herein, the one-tap module 114 is software, hardware, and/or a combination of software and hardware that allows users to pay for transactions initiated in the merchant applications 113 using a single tap of the contactless card 101 to the mobile device 110. The merchant identifier 115 is a unique identifier associated with a merchant and/or a merchant server 150 associated with the merchant. The transaction identifier 116 uniquely identifies a given transaction. For example, the transaction identifier 116 may be a unique alphanumeric identifier, a unique session alphanumeric identifier, a file, etc. The transaction identifier 116 may be generated by the corresponding merchant server 150 and transmitted to the merchant application 113 and/or generated by the merchant application 113 and transmitted to the corresponding merchant server 150.

[0023] Generally, a user may use the merchant application 113 (and/or a merchant website in embodiments where the application 113 is a web browser) to select one or more items and/or services for purchase. When the user has selected the desired items and/or services, the user may encounter an interface for completing the transaction (e.g., a cart page, a checkout page, etc.) in the merchant application 113. Conventionally, the user is required to manually enter their name, card number, expiration date, CVV, and/or address information into the merchant application 113 to complete the purchase. Furthermore, conventional solutions may require the user to log into their account in the

merchant application 113 (e.g., using a login/password, biometric credentials, etc.). Advantageously, however, the one-tap module 114 may facilitate one-tap checkout (e.g., payment) using the contactless card 101, regardless of whether the user has logged into their account in the merchant application 113. Generally, the one-tap module 114 may output a notification in the interface for completing the transaction. The notification may instruct the user to tap the contactless card 101 to the mobile device 110, thereby bringing the contactless card 101 sufficiently close to the card reader 119 of the mobile device 110 to enable data transfer (e.g., NFC data transfer, Bluetooth data transfer, etc.) between the communications interface 107 of the contactless card 101 and the card reader 119 of the mobile device 110. In some embodiments, the mobile device 110 may trigger the card reader 119 via an application program interface (API) call. In one example, the mobile device 110 triggers the card reader via an API call responsive to the user tapping or otherwise selecting an element of the user interface, such as a form field. In addition and/or alternatively, the mobile device 110 may trigger the card reader 119 based on periodically polling the card reader 119. More generally, the mobile device 110 may trigger the card reader 119 to engage in communications using any feasible method.

[0024] After communication has been established between mobile device 110 and contactless card 101, the applet 103 executing on a processor (not pictured) of the contactless card 101 generates and transmits encrypted data 106 to the mobile device 110 via the communications interface 107. For example, the applet 103 of the contactless card 101 may use a cryptographic algorithm to generate a cryptographic payload of encrypted data 106 based at least in part on the private key 104 stored in the memory 102 of the contactless card 101. In such an embodiment, the private key 104 and some other piece of data (e.g., the customer identifier 105, an account identifier, etc.) may be provided as the input to the cryptographic algorithm, which outputs the encrypted data 106. Generally, the applet 103 may use any type of cryptographic algorithm and/or system to generate the encrypted data 106, and the use of a specific cryptographic algorithm as an example herein should not be considered limiting of the disclosure. In some embodiments, the applet 103 may perform encryption using a key diversification technique to generate the encrypted data 106. Examples of key

diversification techniques are described in United States Patent Application 16/205,119, filed November 29, 2018. The aforementioned patent application is incorporated by reference herein in its entirety.

[0025] Once generated, the applet 103 may transmit the encrypted data 106 to the one- tap module 114 of the mobile device 110, e.g., via NFC. In some embodiments, the one- tap module 114 may require the user to log into their account in the account data 124 associated with the contactless card 101 (e.g., provide a login/password and/or biometric credentials). The one-tap module 114 may transmit the encrypted data 106, the merchant identifier 115, and the transaction identifier 116 to the authentication application 123 of the authentication server 120. In some embodiments, the one-tap module 114 may transmit additional data describing the mobile device 110, such as a software fingerprint of the mobile device 110, location (e.g., global positioning system (GPS)) data of the mobile device 110, a unique identifier of the mobile device 110, etc.

[0026] Once received, the authentication application 123 may then authenticate the encrypted data 106. For example, the authentication application 123 may attempt to decrypt the encrypted data 106 using a copy of the private key 104 stored in the memory 122 of the authentication server 120. The private key 104 may be identical to the private key 104 stored in the memory 102 of the contactless card 101, where each contactless card 101 is manufactured to include a unique private key 104 (and the authentication server 120 stores a corresponding copy of each unique private key 104). Therefore, the authentication application 123 may successfully decrypt the encrypted data 106, thereby verifying the encrypted data 106. Although the private key 104 is depicted as being stored in the memory 122, the private key 104 may be stored elsewhere, such as in a secure element and/or a hardware security module (HSM). In such embodiments, the secure element and/or the HSM may decrypt the encrypted data 106 using the private key 104 and a cryptographic function.

[0027] For example, as stated, the customer identifier 105 may be used to generate the encrypted data 106. In such an example, the authentication application 123 may decrypt the encrypted data 106 using the private key 104 of the authentication server 120. If the result of the decryption yields the customer identifier 105 associated with the

account in the account data 124, the authentication application 123 verifies the encrypted data 106 and instructs the VAN generator 142 to generate virtual card data for the account associated with the contactless card 101. If the authentication application 123 is unable to decrypt the encrypted data to yield the expected result (e.g., the customer identifier 105 of the account associated with the contactless card 101), the authentication application 123 does not validate the encrypted data 106. Due to the failed verification, the authentication application 123 does not instruct the VAN generator 142 to generate virtual card data to preserve the security of the associated account.

[0028] In some embodiments, the authentication application 123 processes the additional data received from the mobile device 110 as a condition to instructing the VAN generator 142 to generate the virtual card data. Some such embodiments include where the user is not logged into the merchant account and/or the account in the account data 124. For example, the authentication application 123 may determine whether the device identifier of the mobile device 110 is specified as a known device identifier for the associated account in the account data 124. If the device identifier is not a known identifier, the authentication application 123 may refrain from instructing the VAN generator 142 to generate the virtual card data. Otherwise, the authentication application 123 may instruct the VAN generator 142 to generate the virtual card data. As another example, the authentication application 123 may determine whether the software fingerprint matches a known software fingerprint for the associated account in the account data 124. If the software fingerprint is not a known software fingerprint, the authentication application 123 may refrain from instructing the VAN generator 142 to generate the virtual card data. Otherwise, the authentication application 123 may instruct the VAN generator 142 to generate the virtual card data. As yet another example, the authentication application 123 may determine whether the GPS coordinates indicate the user is at home, at work, or some other known address associated with the account in the account data 124. If the location of the device is not within a threshold distance of the known address, the authentication application 123 may refrain from instructing the VAN generator 142 to generate the virtual card data. Otherwise, the authentication application 123 may instruct the VAN generator 142 to generate the virtual card data. Further still, the authentication application 123 may determine whether the user has

previously made purchases with the merchant associated with the received merchant identifier 115. If the user has not made a previous purchase with the merchant, the authentication application 123 may refrain from instructing the VAN generator 142 to generate the virtual card data. Otherwise, the authentication application 123 may instruct the VAN generator 142 to generate the virtual card data.

[0029] Figure 1B depicts an embodiment where the authentication application 123 validates the encrypted data 106 and instructs the virtual account number (VAN) generator 142 in the memory 141 of the virtual account number server 140 to generate virtual card data 126 comprising a virtual account number, expiration date, and CVV for the account associated with the contactless card 101. In some embodiments, the VAN generator 142 generates the virtual account number, the expiration date, and the CVV. In other embodiments, the VAN generator 142 generates the virtual account number and selects an existing expiration date and/or CVV (e.g., from the account data 124). For example, the existing expiration date and/or CVV may be the expiration date and/or CVV of the contactless card 101, or another card associated with the account in the account data 124. The VAN generator 142 may further receive the name of the account holder and one or more known addresses (e.g., a billing address, shipping address, etc.) from the account in the account data 124 associated with the contactless card 101. In at least one embodiment, the card data 126 including the virtual account number generated by the VAN generator 142 is restricted to the merchant associated with the merchant identifier 115. The virtual account number and/or card data 126 may further include other restrictions (e.g., time restrictions, amount restrictions, etc.). Once generated, the VAN generator 142 may transmit the virtual card data 126, the merchant identifier 115, and the transaction identifier 116 to the merchant server 150. The VAN generator 142 and/or the authentication server 120 may further transmit the name of the account holder, the billing address, and the shipping address to the merchant server 150. In one embodiment, the VAN generator 142 determines a location of the merchant server 150 based on the merchant identifier 115 (e.g., in a lookup table associating merchant identifiers 115 with a corresponding merchant server 150).

[0030] Figure 1C depicts an embodiment where the merchant server 150 receives the virtual card data 126, the merchant identifier 115, the transaction identifier 116, and

any other data (e.g., the account holder's name and addresses) from the VAN generator 142 and/or the authentication server 120. Generally, the merchant server 150 may process the transaction using the data received from the VAN generator 142 and/or the authentication server 120. More specifically, the merchant server 150 may generate a transaction record 152 in the transaction database 151 using the received virtual account number, expiration date, CVV, name, and/or addresses. The transaction record 152 in the transaction database 151 may further specify the transaction identifier 116. Doing so uses the virtual card data 126 to pay for the transaction without requiring the user to input any payment information using the merchant application 113 executing on the mobile device 110. Instead, the transaction is paid for and processed using the securely generated and transmitted virtual card data 126.

[0031] As shown, the merchant server 150 may generate a confirmation 153 indicating that payment for the transaction was received and the transaction has been processed. The merchant server 150 may transmit the confirmation 153 to the merchant application 113, which may output the confirmation 153 on a display of the mobile device 110. In other embodiments, the confirmation 153 may be an email confirmation, text message confirmation, push notification, etc. Doing so informs the user that the transaction has been completed and may include additional information regarding the delivery of items, rendering of services, etc.

[0032] Figure 2A is a schematic 200 depicting an example embodiment of tapping the contactless card 101 for one-tap payment. As shown, the merchant application 113 executing on the mobile device 110 displays a graphical user interface (GUI) to complete a purchase. For example, the GUI may be a shopping cart page, checkout page, or any other page of the merchant application 113. As shown, a notification 201 instructs a user to tap the contactless card 101 to the mobile device 110 to pay for the items in their shopping cart using one-tap payment. Once the tap is detected, as shown in the schematic 210 of Figure 2B, the merchant application 113 may output a notification 202 indicating that a secure virtual number is being generated.

[0033] As stated, once the user taps the contactless card 101 to the mobile device 110, the contactless card 101 generates the encrypted data 106 and transmits the

encrypted data 106 to the mobile device 110. Once received, the one-tap module 114 transmits the encrypted data 106, merchant ID 115, and transaction ID 116 to the authentication server 120. The authentication application 123 may then verify the encrypted data 106 by decrypting the encrypted data 106 with the private key 104 stored in the memory 122 of the server 120. Once verified, the authentication application 123 instructs the VAN generator 142 to generate the virtual card data 126 (e.g., a virtual card number, expiration date, and/or CVV). As stated, in some embodiments, the VAN generator 142 may generate the virtual card number and retrieve the expiration date and/or CVV from the account data 124. The VAN generator 142 may also receive the merchant ID 115, the transaction ID 116, and additional data describing the user (e.g., names, billing address, shipping address, etc.) from the account data 124. The VAN generator 142 may then transmit the virtual card data 126, the merchant ID 115, transaction ID 116, and any usernames and/or addresses to the merchant server 150. The merchant server 150 may then process the transaction using the data received from the VAN generator 142, e.g., by generating a transaction record 152 in the transaction database 151 using at least the received virtual card number, expiration date, CVV. The transaction record 152 may further include the user's name, billing address, shipping address, and an indication of each item and/or service purchased. The merchant server 150 may then transmit an order confirmation to the mobile device 110. Figure 2D is a schematic 230 depicting an example confirmation outputted by the merchant application 113 on the mobile device 110.

[0034] In some embodiments, the merchant server 150 may require resolution of data conflicts before processing the transaction using the data received from the VAN generator 142. For example, the account data 124 may specify a billing address of "123 Main St." for the user's account, whereas the user's profile in the merchant server 150 may specify "300 Bowles Dr." as the billing address of the user's account. Therefore, as depicted in the schematic 220 of Figure 2C, the merchant server 150 may cause the merchant application 113 to output the notification 203 on the mobile device 110. As shown, the notification 203 allows the user to select one of the two different addresses. Once selected, the merchant application 113 transmits an indication of the selected address to the merchant server 150. The merchant server 150 may then process the

transaction using the selected address. Although an address is used as an example, the merchant server 150 and/or the merchant application 113 may similarly resolve data conflicts for other data types (e.g., names, shipping addresses, etc.).

[0035] Figure 3 illustrates an embodiment of a logic flow 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 to use a contactless card to perform one-tap payment. Embodiments are not limited in this context.

[0036] As shown, the logic flow 300 begins at block 305, where a user optionally provides authentication credentials to authenticate a user account in the merchant application 113, where the merchant application 113 includes the one-tap module 114 for supporting one-tap checkout as described herein. However, as stated, in some embodiments, the user may not authenticate into their account with the merchant application 113. At block 310, the user may select one or more items and/or services to purchase as part of a transaction. For example, the user may specify to request a car service to drive them from their home to the airport. As another example, the user may specify to order food items for delivery to their office. As stated, the merchant application 113 may be associated with a unique merchant ID 115. Similarly, the user's selected items and/or services may be associated with a transaction ID 116. As the user selects and/or removes items and/or services from their shopping cart, the merchant application 113 may transmit an indication of the transaction ID 116 and the selected and/or removed items to the merchant server 150, thereby allowing the merchant server 150 to be aware of what items and/or services are associated with a given transaction ID 116. At block 315, the one-tap module 114 of the merchant application 113 outputs a notification specifying to tap a contactless card 101 to the mobile device 110. In one embodiment, the one-tap module 114 outputs the notification upon determining the merchant application 113 outputs an element (e.g., a form, a form field, etc.) that is associated with completing a transaction and/or paying for a transaction.

[0037] At block 320, a user taps the contactless card 101 to the mobile device 110 to cause the contactless card 101 to generate and transmit encrypted data 106. The applet

103 of the contactless card 101 may then generate the encrypted data 106 using the private key 104, input data (e.g., the customer identifier 105), and a cryptographic algorithm. The applet 103 may then transmit the encrypted data 106 to the mobile device 110. At block 325, the one-tap module 114 and/or the merchant application 113 may receive the encrypted data 106 from the communications interface 107 of the contactless card. At block 330, the one-tap module 114 and/or the merchant application 113 transmits the encrypted data 106, merchant identifier 115, and the transaction identifier 116 to the authentication server 120.

[0038] At block 335, the authentication application 123 decrypts the encrypted data 106 using the private key 104 in the memory 122 of the authentication server 120 to validate the encrypted data 106. At block 340, the authentication application 123 transmits an indication to the VAN generator 142 specifying to generate card data comprising a virtual account number, expiration date, and CVV. The authentication application 123 may further transmit the merchant identifier 115 and/or the transaction identifier 116 to the VAN generator 142. At block 345, the VAN generator 142 generates the virtual account number, expiration date, and CVV. At block 350, the VAN generator 142 transmits the virtual account number, expiration date, CVV, merchant ID 115, and transaction ID 116 to the mobile device 110. The VAN generator 142 may further include the name, billing address, and shipping address of the account holder, which may be stored locally by the VAN generator 142 and/or received from the authentication server 120.

[0039] At block 355, the merchant server 150 processes the transaction associated with the transaction ID 116 using the data received from the VAN generator 142. More specifically, the merchant server 150 processes payment for the transaction using at least the virtual card number, expiration date, and CVV generated by the VAN generator 142. As stated, the merchant server 150 may process the transaction payment by generating a transaction record 152 in the transaction database 151 using at least the received virtual card number, expiration date, CVV. The transaction record 152 may further include the user's name, billing address, shipping address, and an indication of each item and/or service associated with the transaction ID 116. At block 360, the merchant server 150 may transmit a payment confirmation to the merchant application 113 on the mobile

device. The confirmation may generally indicate that the payment has been processed using the virtual card data generated by the VAN generator.

[0040] Figure 4 illustrates an embodiment of a logic flow 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 500 may include some or all of the operations performed to authorize a one-tap payment using the contactless card 101. Embodiments are not limited in this context.

[0041] As shown, the logic flow 400 begins at block 405, where the one-tap module 114 and/or the merchant application 113 transmits the encrypted data 106 generated by the contactless card 101, the merchant ID 115, transaction ID 116, and a device identifier to the authentication server 120. As stated, the device identifier may be a software fingerprint of the mobile device 110, a unique identifier of the mobile device 110 (e.g., a medium access control (MAC) address), and the like. At block 410, the authentication application 123 may determine that the received device identifier is not associated with the account of the contactless card 101. For example, the account data 124 may include known device identifiers for a given account. If the received device identifier is not specified as a known device identifier in the account data 124, the authentication application 123 may require additional information to preserve account security. As such, the authentication application 123 may instruct the one-tap module 114 to output an indication to receive authentication credentials for the account in the account data 124. For example, the one-tap module 114 may output a notification specifying that the user provides their fingerprint as biometric input to authenticate into their account associated with the account in the account data 124. As another example, the one-tap module 114 may output a notification specifying that the user provides the login and password associated with the account in the account data 124. In other embodiments, the user may log into the account using FaceID, other biometric identifiers, or any other type of credential.

[0042] At block 415, the user provides credentials (e.g., biometric input, login/password, etc.) for the account in the account data 124 to the one-tap module 114 and/or the merchant application 113. The one-tap module 114 may include logic to

verify the received credentials. In other embodiments, the one-tap module 114 may transmit the received credentials to the authentication server 120. At block 420, the authentication application 123 authenticates the received account credentials and the encrypted data 106 generated by the contactless card 101. The authentication application 123 may determine whether the received account credentials match credentials for the account stored in the account data 124 and decrypt the encrypted data 106 using the private key 104. At block 425, the authentication server 120 may determine to instruct the VAN generator 142 to generate a virtual card number, expiration date, and CVV. The authentication server 120 may further transmit, with the instruction, a name, billing address, and shipping address from the account data 124 to the VAN generator 142.

[0043] Figure 5 illustrates an embodiment of a logic flow 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 to provide one-tap payment for transactions using a contactless card. Embodiments are not limited in this context.

[0044] As shown, the logic flow 500 begins at block 505, where the merchant server 150 generates a transaction identifier 116 for a transaction. As stated, the transaction identifier 116 may be any type of identifier, such as a unique alphanumeric identifier, a unique session alphanumeric identifier, a file, etc. At block 510, the merchant application 113 receives the transaction identifier 116 from the merchant server 150 via the network 130. The merchant application 113 and/or the one-tap module 114 may transmit the received transaction identifier 116 with the encrypted data 106 to the authentication server 120. At block 515, the merchant application 113 transmits indications of items and/or services added and/or removed from the user's shopping cart for the transaction associated with the transaction identifier 116. At block 520, the merchant server 150 receives the virtual card data 126, user information, and transaction identifier 116 from the VAN generator 142. At block 525, the merchant server processes payment for the transaction associated with the transaction identifier 116 by generating a transaction record in the transaction database 151.

[0045] Figure 6 illustrates an embodiment of a logic flow 600. The logic flow 600 may be representative of some or all of the operations executed by one or more embodiments described herein. For example, the logic flow 600 may include some or all of the operations performed by the merchant application 113 and/or the merchant server 150. Embodiments are not limited in this context.

[0046] As shown, the logic flow 600 begins at block 605, where the merchant server 150 receives the virtual card data 126 (e.g., virtual account number, expiration date, and CVV), account holder data (e.g., name, billing address, shipping address, etc.), and transaction ID 116 from the VAN generator 142. At block 610, the merchant server 150 determines that at least one element of the received data is not like at least one element of user data in a user profile associated with the merchant. For example, the received mailing and/or billing addresses may not match any addresses specified in the user profile.

[0047] At block 615, the merchant server 150 instructs the merchant application 113 to output indications of the data elements identified at block 610. For example, the merchant application 113 may output a list of addresses. At block 620, the merchant application 113 receives a user selection of one of the data elements (e.g., a first address outputted at block 615). At block 625, the merchant application 113 transmits an indication of the data element selected at block 620 to the merchant server 150. At block 630, the merchant server 150 processes the transaction using the data received from the VAN generator 142 and the data element selected by the user at block 620. More specifically, the merchant server 150 generates a transaction record for the transaction in the transaction database 151.

[0048] Figure 7 illustrates an embodiment of an exemplary computing architecture 700 comprising a computing system 702 that may be suitable for implementing various embodiments as previously described. In various embodiments, the computing architecture 700 may comprise or be implemented as part of an electronic device. In some embodiments, the computing architecture 700 may be representative, for example, of a system that implements one or more components of the system 100. In some embodiments, computing system 702 may be representative, for example, of the

contactless card 101, mobile devices 110, authentication server 120, the virtual account number server 140, and/or the merchant server 150 of the system 100. The embodiments are not limited in this context. More generally, the computing architecture 700 is configured to implement all logic, applications, systems, methods, apparatuses, and functionality described herein with reference to Figures 1-6.

[0049] As used in this application, the terms “system” and “component” and “module” 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 architecture 700. For example, a component can be, but is not limited to being, a process running on a computer processor, a computer 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.

[0050] The computing system 702 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 system 702.

[0051] As shown in Figure 7, the computing system 702 comprises a processor 704, a system memory 706 and a system bus 708. The processor 704 can be any of various commercially available computer processors, including without limitation an AMD® Athlon®, Duron® and Opteron® processors; ARM® application, embedded and secure processors; IBM® and Motorola® DragonBall® and PowerPC® processors; IBM and Sony® Cell processors; Intel® Celeron®, Core®, Core (2) Duo®, Itanium®, Pentium®, Xeon®, and XScale® processors; and similar processors. Dual microprocessors, multi core processors, and other multi processor architectures may also be employed as the processor 704.

[0052] The system bus 708 provides an interface for system components including, but not limited to, the system memory 706 to the processor 704. The system bus 708 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 708 via a 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.

[0053] The system memory 706 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 (e.g., one or more flash arrays), 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 Figure 7, the system memory 706 can include non-

volatile memory 710 and/or volatile memory 712. A basic input/output system (BIOS) can be stored in the non-volatile memory 710.

[0054] The computing system 702 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 (HDD) 714, a magnetic floppy disk drive (FDD) 716 to read from or write to a removable magnetic disk 718, and an optical disk drive 720 to read from or write to a removable optical disk 722 (e.g., a CD-ROM or DVD). The HDD 714, FDD 716 and optical disk drive 720 can be connected to the system bus 708 by a HDD interface 724, an FDD interface 726 and an optical drive interface 728, respectively. The HDD interface 724 for external drive implementations can include at least one or both of Universal Serial Bus (USB) and IEEE 1394 interface technologies. The computing system 702 is generally configured to implement all logic, systems, methods, apparatuses, and functionality described herein with reference to Figures 1-6.

[0055] 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 memory units 710, 712, including an operating system 730, one or more application programs 732, other program modules 734, and program data 736. In one embodiment, the one or more application programs 732, other program modules 734, and program data 736 can include, for example, the various applications and/or components of the system 100, e.g., the applet 103, private keys 104, encrypted data 106, operating system 112, merchant applications 113, the one-tap module 114, the authentication application 123, the VAN generator 142, and/or the merchant server 150.

[0056] A user can enter commands and information into the computing system 702 through one or more wire/wireless input devices, for example, a keyboard 738 and a pointing device, such as a mouse 740. 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, trackpads, sensors, styluses, and the like. These and other input devices are often

connected to the processor 704 through an input device interface 742 that is coupled to the system bus 708, 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.

[0057] A monitor 744 or other type of display device is also connected to the system bus 708 via an interface, such as a video adaptor 746. The monitor 744 may be internal or external to the computing system 702. In addition to the monitor 744, a computer typically includes other peripheral output devices, such as speakers, printers, and so forth.

[0058] The computing system 702 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 748. The remote computer 748 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 of the elements described relative to the computing system 702, although, for purposes of brevity, only a memory/storage device 750 is illustrated. The logical connections depicted include wire/wireless connectivity to a local area network (LAN) 752 and/or larger networks, for example, a wide area network (WAN) 754. 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. In embodiments, the network 130 of Figure 1 is one or more of the LAN 752 and the WAN 754.

[0059] When used in a LAN networking environment, the computing system 702 is connected to the LAN 752 through a wire and/or wireless communication network interface or adaptor 756. The adaptor 756 can facilitate wire and/or wireless communications to the LAN 752, which may also include a wireless access point disposed thereon for communicating with the wireless functionality of the adaptor 756.

[0060] When used in a WAN networking environment, the computing system 702 can include a modem 758, or is connected to a communications server on the WAN 754, or has other means for establishing communications over the WAN 754, such as by way

of the Internet. The modem 758, which can be internal or external and a wire and/or wireless device, connects to the system bus 708 via the input device interface 742. In a networked environment, program modules depicted relative to the computing system 702, or portions thereof, can be stored in the remote memory/storage device 750. 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.

[0061] The computing system 702 is operable to communicate with wired and wireless devices or entities using the IEEE 802 family of standards, such as wireless devices operatively disposed in wireless communication (e.g., IEEE 802.16 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 802.1x (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).

[0062] Various embodiments may be implemented using hardware elements, software elements, or a combination of both. Examples of hardware elements may include processors, microprocessors, circuits, 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), logic gates, registers, semiconductor device, chips, microchips, chip sets, and so forth. Examples of software may include software components, programs, applications, computer programs, application programs, system 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. 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.

[0063] 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.

[0064] 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

1. A method, comprising:

receiving, by an authentication application executing on a processor from a merchant application executing on a client device: (i) a merchant identifier of a merchant associated with the merchant application, (ii) a transaction identifier of a transaction, (iii) encrypted data, and (iv) a software fingerprint of the client device, wherein the encrypted data is generated by a contactless card and read by the client device;

verifying, by the authentication application, the encrypted data by decrypting the encrypted data;

determining, by the authentication application, that the software fingerprint matches a stored software fingerprint of the client device;

instructing, by the authentication application based on the verification of the encrypted data and the determination that the software fingerprint matches the stored software fingerprint, a virtual account number generator to generate a virtual account number for the transaction;

generating, by the virtual account number generator the virtual account number, an expiration date associated with the virtual account number, and a card verification value (CVV) associated with the virtual account number based on the instruction; and

transmitting, by the virtual account number generator, the merchant identifier, the transaction identifier, the virtual account number, the expiration date, and the CVV to a merchant server associated with the merchant to process the transaction using the transaction identifier, the virtual account number, the expiration date, and the CVV.

2. The method of claim 1, further comprising:

receiving, by the authentication application, a location of the client device; and

determining, by the authentication application, that the location of the client device is within a threshold distance of a known location associated with the contactless card, wherein the authentication application further instructs the virtual account number generator to generate the virtual account number based on the determination that the client device is within the threshold distance of the known location.

3. The method of claim 1, further comprising:
determining, by the authentication application, that the contactless card has been used to make a previous purchase with the merchant, wherein the authentication application further instructs the virtual account number generator to generate the virtual account number based on the determination that the contactless card has previously been used to make a purchase with the merchant.
4. The method of claim 1, further comprising:
receiving, by the virtual account number generator from an account database, an account holder name of an account associated with the contactless card and an address associated with the account; and
transmitting, by the virtual account number generator, the account holder name and address to the merchant server.
5. The method of claim 1, further comprising:
receiving, by the authentication application, authentication information for a user account associated with the contactless card, the authentication information comprising one or more of: (i) a username and a password, (ii) one or more biometric credentials; and
verifying, by the authentication application, the authentication information for the user account.
6. The method of claim 1, further comprising:
determining, by the authentication application, a device identifier received from the client device is associated with a user account associated with the contactless card.
7. The method of claim 1, wherein the virtual account number is different than an account number associated with the contactless card.
8. A computer-readable storage medium including instructions that when executed by a processor, cause the processor to:
receive, by an authentication application executing on the processor and from a merchant application executing on a client device: (i) a merchant identifier of a merchant associated with the merchant application, (ii) a transaction identifier of a transaction, (iii)

encrypted data, and (iv) a software fingerprint of the client device, wherein the encrypted data is generated by a contactless card and read by the client device;

verify, by the authentication application, the encrypted data by decrypting the encrypted data;

determine, by the authentication application, that the software fingerprint matches a stored software fingerprint of the client device;

instruct, by the authentication application based on the verification of the encrypted data and the determination that the software fingerprint matches the stored software fingerprint, a virtual account number generator to generate a virtual account number for the transaction;

generate, by the virtual account number generator the virtual account number, an expiration date associated with the virtual account number, and a card verification value (CVV) associated with the virtual account number based on the instruction; and

transmit, by the virtual account number generator, the merchant identifier, the transaction identifier, the virtual account number, the expiration date, and the CVV to a merchant server associated with the merchant to process the transaction using the transaction identifier, the virtual account number, the expiration date, and the CVV.

9. The computer-readable storage medium of claim 8, wherein the instructions further cause the processor to:

receive, by the authentication application, a location of the client device; and

determine, by the authentication application, that the location of the client device is within a threshold distance of a known location associated with the contactless card, wherein the authentication application further instructs the virtual account number generator to generate the virtual account number based on the determination that the client device is within the threshold distance of the known location.

10. The computer-readable storage medium of claim 8, wherein the instructions further cause the processor to:

determine, by the authentication application, that the contactless card has been used to make a previous purchase with the merchant, wherein the authentication application further instructs the virtual account number generator to generate the virtual account number

based on the determination that the contactless card has previously been used to make a purchase with the merchant.

11. The computer-readable storage medium of claim 8, wherein the instructions further cause the processor to:

receive, by the virtual account number generator from an account database, an account holder name of an account associated with the contactless card and an address associated with the account; and

transmit, by the virtual account number generator, the account holder name and address to the merchant server.

12. The computer-readable storage medium of claim 8, wherein the instructions further cause the processor to:

receive, by the authentication application, authentication information for a user account associated with the contactless card, the authentication information comprising one or more of: (i) a username and a password, (ii) one or more biometric credentials; and

verify, by the authentication application, the authentication information for the user account.

13. The computer-readable storage medium of claim 8, wherein the instructions further cause the processor to:

determine, by the authentication application, a device identifier received from the client device is associated with a user account associated with the contactless card.

14. The computer-readable storage medium of claim 8, wherein the virtual account number is different than an account number associated with the contactless card.

15. A computing apparatus comprising:

a processor; and

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

receive, by an authentication application executing on the processor and from a merchant application executing on a client device: (i) a merchant identifier of a

merchant associated with the merchant application, (ii) a transaction identifier of a transaction, (iii) encrypted data, and (iv) a software fingerprint of the client device, wherein the encrypted data is generated by a contactless card and read by the client device;

verify, by the authentication application, the encrypted data by decrypting the encrypted data;

determine, by the authentication application, that the software fingerprint matches a stored software fingerprint of the client device;

instruct, by the authentication application based on the verification of the encrypted data and the determination that the software fingerprint matches the stored software fingerprint, a virtual account number generator to generate a virtual account number for the transaction;

generate, by the virtual account number generator the virtual account number, an expiration date associated with the virtual account number, and a card verification value (CVV) associated with the virtual account number based on the instruction; and

transmit, by the virtual account number generator, the merchant identifier, the transaction identifier, the virtual account number, the expiration date, and the CVV to a merchant server associated with the merchant to process the transaction using the transaction identifier, the virtual account number, the expiration date, and the CVV.

16. The computing apparatus of claim 15, wherein the instructions further cause the processor to:

receive, by the authentication application, a location of the client device; and

determine, by the authentication application, that the location of the client device is within a threshold distance of a known location associated with the contactless card, wherein the authentication application further instructs the virtual account number generator to generate the virtual account number based on the determination that the client device is within the threshold distance of the known location.

17. The computing apparatus of claim 15, wherein the instructions further cause the processor to:

determine, by the authentication application, that the contactless card has been used to make a previous purchase with the merchant, wherein the authentication application further instructs the virtual account number generator to generate the virtual account number based on the determination that the contactless card has previously been used to make a purchase with the merchant.

18. The computing apparatus of claim 15, wherein the instructions further cause the processor to:

receive, by the virtual account number generator from an account database, an account holder name of an account associated with the contactless card and an address associated with the account; and

transmit, by the virtual account number generator, the account holder name and address to the merchant server.

19. The computing apparatus of claim 15, wherein the instructions further cause the processor to:

receive, by the authentication application, authentication information for a user account associated with the contactless card, the authentication information comprising one or more of: (i) a username and a password, (ii) one or more biometric credentials; and

verify, by the authentication application, the authentication information for the user account.

20. The method of claim 1, further comprising prior to instructing the virtual account number generator to generate the virtual account number for the transaction:

determining, by the authentication application, to process the transaction without receiving user input specifying payment information associated with the contactless card via the merchant application based on: the verification of the encrypted data, and the determination that the software fingerprint matches the stored software fingerprint,

wherein the authentication application instructs the virtual account number generator to generate the virtual account number for the transaction further based on the determination

to process the transaction without receiving the user input specifying the payment information associated with the contactless card.