[0113] Additionally, the payee information may include a type of transaction identifier informing the mobile payment application 123 that the request is associated with a remote payment transaction. Accordingly, the mobile payment application 123 may select the appropriate public encryption key in which to encrypt the payment information (e.g., remote key manager public key), the appropriate destination or routing address for the remote key manager 140, and the correct communication protocol or message format for communicating with the remote key manager 140.

[0114] At step 502, mobile payment application 123 uses the received payee information to retrieve and/or generate payment information from the secure element 122 of the mobile device 120. "Payment information" may include payment card data (e.g., a payment account number (PAN) and expiration date), a cryptogram (such as a dynamic card verification value (dCVV or dCVV2) or other dynamically generated data), or any other information suitable to conduct a remote payment transaction.

[0115] At step 503, mobile payment application 123 encrypts the payment information using a remote key manager key. The remote key manager key may be either symmetric or a public key associated with a remote key manager public/private key pair. In some embodiments, mobile payment application 123 may determine the remote key manager public key using a remote key manager certificate stored on mobile device 120, may request a certificate from the remote key manager, or may obtain the remote key manager public key on the secure element 122 or general memory of the mobile device 120. Where the remote key manager key is a symmetric key, the symmetric key may be stored securely with the mobile payment application 123.

[0116] At step 504, mobile payment application 123 sends the encrypted payment information to merchant application 121. It should be noted that because the remote key manager private key is not known by merchant application 121, the encrypted payment information cannot be deciphered by the merchant application 121.

[0117] At step 505, merchant application 121 receives the encrypted payment information and generates a payment request including the encrypted payment information, the received encrypted payment information, any other transaction data (e.g., transaction amount, merchant identifier, product identifiers, etc.) associated

with the payment transaction that was not included in the encrypted payment information, a merchant certificate associated with the merchant computer 130, and a signature generated using the merchant computer private key. The merchant application 121 may then transmit the payment request to the remote key manager 140. In some embodiments, the payment request may be encrypted using a session key established between the remote key manager 140 and the mobile device 120. The session key may be established and communicated through any suitable manner.

[0118] At step 506, the remote key manager 140 receives the payment request, may decrypt the payment request using the session key, and parses the encrypted payment information, non-encrypted transaction data, the merchant certificate, and the signature from the payment request. The remote key manager 140 may validate the signature. Typically, the signature may be validated using the public key included in the merchant certificate. Alternatively, in some embodiments, the public key may be registered and stored at the remote key manager as being associated with the merchant certificate. If the signature is not validated, then the remote key manager 140 indicates to the merchant application 121 that the signature is invalid, and the method halts. Otherwise, the method proceeds to step 507. Note that validating the signature is optional (as well as the signing being optional) and may occur on a periodic basis and thus, the signature may not be passed from the merchant computer 130 as well.

[0119] At step 507, remote key manager 140 decrypts the encrypted payment information using a remote key manager key stored at the remote key manager 140. The remote key manager key may include a symmetric or a private key of a public/private key pair. In embodiments using the remote key manager private key, the private key is configured to decrypt information that has been encrypted with a corresponding remote key manager public key and the private key is meant to be stored securely at the remote key manager 140 so that only the remote key manager 140 may be capable of decrypting information encrypted with the remote key manager public key. Similarly, the symmetric key may be stored at the remote key manager 140 and used to decrypt the encrypted payment information.

[0120] At step 508, remote key manager 140 re-encrypts the payment information using the determined merchant public key. As explained above, the merchant public key may be included in the merchant application certificate and

information that would typically be present in card present transactions (e.g., payment credentials, chip data, etc.). Accordingly, the merchant computer 130 may map the decrypted payment information (as well as the other transaction information included in the payment response) to the format associated with an authorization request message of the merchant computer 130, acquirer computer 150, payment processing network 160, and issuer computer 170.

[0125] It should be understood that FIG. 5 is intended to be descriptive and non-limiting. For example, as will be shown below in reference to FIG. 6, any number of different entities may be used to decrypt the re-encrypted payment information. For example, FIG. 6 shows three different options for decrypting and passing payment information. Other transaction entities may also be configured to decrypt the re-encrypted payment information and initiate a payment transaction.

[0126] FIG. 6 shows additional exemplary methods for conducting a remote transaction using a remote key manager 140 and a merchant application 121 of a mobile device 120, according to some embodiments of the invention. For example, as shown in FIG. 6, after remote key manager 140 decrypts the encrypted payment information, remote key manager 140 may re-encrypt the payment information using a public key associated with any number of different transaction processing entities. For instance, instead of using a merchant application public key to encrypt the payment information, a merchant public key (associated with a private key stored at the merchant server computer) or an acquirer public key (associated with a private key stored at an acquirer computer 150) may be used to encrypt the payment information and the re-encrypted payment information may be passed to each respective entity (e.g., merchant server computer or acquirer computer 150) for decrypting and authorization request message generation.

[0127] Steps 601-608 (only some of which are shown in FIG. 6) are similar to those explained in reference to FIG. 5 above. For example, at step 601, a merchant application 121 sends payee information to a mobile payment application 123 of a mobile device 120 in order to identify appropriate payment information for the transaction. Further, at steps 602-604, the mobile payment application 123 encrypts the payment information using a remote key manager public key and sends the encrypted payment information to the merchant application 121.

The consumer 1010 may then initiate a checkout through the merchant application 1021. The merchant application 1021 may provide an option to select a payment card or account for payment through a remote transaction application 1025. The consumer 1010 may select an account in order to initiate payment.

[0197] At step 1102, the merchant application 1021 sends payee information to a remote transaction application 1025 using a remote transaction software development kit (SDK) or other third party service layer or application located on the mobile device 1020. In one embodiment, the remote transaction SDK 1024 or remote transaction application 1025 may include a service layer that is configured to interface between a merchant application 1021 and a remote transaction applet 1024 stored on a secure element 1022 of the mobile device 1020. Alternatively, the remote transaction application 1025 may also be stored in general purpose memory of the mobile device 1020 and may be configured to communicate with a third party server platform 1030 (e.g., mobile wallet provider). Either way, the remote transaction application 1025 may be configured to communicate with a mobile payment application 1023 (e.g., Visa™ Paywave™ application) that is stored on a secure element 1022 of the mobile device 1020.

[0198] At step 1103, the remote transaction SDK 1024 or remote transaction application 1025 communicates the payee information with a mobile payment application 1023 residing in the secure element 1022. The remote transaction application 1025 may then use APIs or other commands to request that the mobile payment application 1023 (e.g., Visa™ Paywave™ application) provide provisioned payment credentials (e.g., a payment token, primary account number (PAN), pseudo-PAN, Ghost PAN, expiration date, etc.) that are stored on the secure element 1022 in a secured manner.

[0199] At step 1104, the mobile payment application 1023 may use the received payee information to retrieve and generate payment information using the secure element 1022. For example, the mobile payment application 1023 may access token information from the secure element 1022 and generate a payment request payload (including token data elements) and a dynamic value (e.g., authentication value) using the payee information, the token, and the information associated with the token (e.g., token expiration date). For instance, payment information may include a token (or PAN, pseudo-PAN, PAN substitute, etc.), a token expiration date, a unique transaction identifier (e.g., a Nonce), token

[0228] At step 1207, the third party server system decrypts the encrypted payment token. In some embodiments, the payment token may be decrypted using the transaction public key, the server private key, the key derivation function, the elliptic curve and the generator function, and the received encrypted payment token. The decrypted payment token may then be processed by third party server system.

[0229] FIG. 13 shows another method for encrypting a payment token using ECC at a mobile payment application 1023 and subsequently decrypting the token at a third party server system. Steps 1301-1303 are substantially the same as described above in relation to steps 1201-1203 of FIG. 12 and include the mobile payment application 1023 retrieving a third party server computer public key associated with the mobile wallet provider computer 1040, a key derivation function (KDF), an elliptic curve (e.g., P256), and a generator function to be used in the encryption process.

[0230] At step 1303, the mobile payment application 1023 generates a random ephemeral elliptic curve (EC) key pair associated with a payment transaction. The key pair may comprise a transaction private key and a transaction public key. In the embodiment of FIG. 13, the key pair may be generated such that the transaction public key is equal to the transaction private key multiplied by the generator function.

[0231] At step 1304, the mobile payment application 1023 generates a shared secret using the product of the server public key and transaction private key. It should be noted that the shared secret is also equal to the product of the server private key, the generator function, and the private transaction key. Similarly, the shared secret is also equal to the product of the server private key and the transaction public key.

[0232] At step 1305, the mobile payment application 1023 generates a symmetric key and a message authentication code (MAC) key using the shared secret determined at step 1304 and the KDF. The symmetric key may be, for example, an AES encryption key. The MAC key may be, for example, a key used to generate a hash-based message authentication code (HMAC).

[0233] At step 1306, the mobile payment application 1023 generates encrypted ciphertext comprising the payment token. In some embodiments, the ciphertext may be generated using the symmetric key determined in step 1305. The

ciphertext comprises the encrypted payment token, and is unreadable without the symmetric key.

[0234] At step 1307, the mobile payment application 1023 calculates a message authentication code (MAC) for the encrypted token. The MAC may be generated using the MAC key determined in step 1305, and any suitable MAC function, such as HMAC-SHA-256. The MAC may be generated in order verify that the encrypted token is not modified during transmission.

[0235] At step 1308, the mobile payment application 1023 sends the encrypted token, the transaction public key, and the MAC for the encrypted token to third party server system.

[0236] At step 1309, the third party server system determines the shared secret using the product of the server private key and the transaction public key received. It should be noted that although the shared secret was computed using different keys, it is the same as the shared secret computed by the mobile payment application 1023 in step 1304.

[0237] At step 1310, the third party server system generates a symmetric key and a message authentication code (MAC) key using the shared secret determined at step 1305 and the KDF determined at step 1303. It should be noted that as the shared secret and KDF are the same as those used by the mobile payment application 1023, the symmetric key and MAC key are the same as used to generate the encrypted token and MAC received by the third party server system at step 1308.

[0238] At step 1311, the third party server system calculates the MAC of the received encrypted token using the MAC key determined at step 1305. If the calculated MAC does not match the MAC received at step 1308, a corruption of the encrypted token is determined and the method results in an error. This may cause, for example, the third party server system to request a retransmission of the encrypted token. If the calculated MAC and received MAC match, then the method proceeds further.

[0239] Then, the third party server system decrypts the encrypted token using the symmetric key determined at step 1310. The decrypted payment token may then be processed by the third party server system.