can be divided into steps S2021 to S2023 in Fig. 4.

[0077] In step S2011, a first card binding message is sent to the server.

[0078] The first card binding message includes the card identifier of the card to be bound. The user terminal sends the first card binding message to the server, such that the server identifies the card type of the card according to the card identifier of the card to be bound, and feeds back the card type of the card to the user terminal through a response message. For different types of cards, their card identifiers have different formats. The server can identify the card type of the card to be bound based on a format of the card identifier of the card to be bound.

[0079] In step S2021, a first response message sent by the server is received.

[0080] The first response message includes the card type of the card. The server may feedback the card type of the card to the user terminal through the first response message.

[0081] In step S2012, a second card binding message is sent to the server.

[0082] The second card binding message includes card type verification information of the card to be bound. For different types of cards different card type verification information are provided. Specifically, in a case that the card type of the card to be bound is a first-type, the card type verification information includes first-type verification information; and in a case that the card type of the card to be bound is a second-type, the card type verification information includes second-type verification information. The first-type verification information is different from the second-type verification information. For example, in a case that a card of the first-type is a debit card, the first-type verification information may include a withdrawal password; and in a case that a card of the second-type is a credit card, the second-type verification information may include a credit card security code and a credit card validity period.

[0083] Upon reception of the first response message sent by the server, the user terminal sends a second card binding message to the server, such that the server performs card type verification according to the card type verification information in the second card binding message. The card type verification is a verification for that type of cards. For example, in a case that a card of the first-type is a debit card, the verification is performed with respect to a type of debit card; and in a case that a card of the second-type is a credit card, the verification is performed with respect to a type of credit card.

[0084] In step S2022, a second response message sent by the server is received.

[0085] The second response message includes a dynamic verification code. In a case of successful verification of the card type, the server may feedback the dynamic verification code

to the user terminal through the second response message.

[0086] In step S2013, a binding verification message is sent to the server.

[0087] The binding verification message includes an input verification code. The input verification code may be a verification code received by the user terminal, and may specifically be a verification code inputted by the user, or a verification code that is identified and inputted from a short message including a dynamic verification code under authorization of the user terminal, which is not limited herein.

[0088] Upon reception of the second response message sent by the server, a binding verification message is sent to the server, such that the server performs, according to the input verification code and the dynamic verification code, verification of the verification code. Upon successful verification of the card type and successful verification of the verification code, the server determines that verification of the card to be bound is successful.

[0089] In step S2023, a third response message sent by the server is received.

[0090] The third response message includes dedicated card information for the card to be bound. Upon successful verification of the card, the server allocates the dedicated card information for the card to be bound to the user terminal, and may send it to the user terminal through the third response message.

[0091] In order to facilitate management of information of the card, a life state may be set for the card, and data synchronization between the user terminal and the server may be performed to ensure data consistency between the user terminal and the server. Fig. 5 is a flowchart of another embodiment of a card management method according to the first aspect of the present application. Fig. 5 differs from Fig. 3 in that, the card management method shown in Fig. 5 may further include steps S204 and S205.

[0092] In step S204, a life state of the card to be bound is set to a valid state through the security element, and a first mapping relationship is generated or updated.

[0093] The first mapping relationship includes a mapping relationship among the dedicated card information or the card to be bound, the card type of the card to be bound, and the life state of the card to be bound.

[0094] In some embodiments, the card has never been bound with the application in the user terminal before, that is, the current binding is the first binding of the card with the application in the user terminal. The user terminal does not store a mapping relationship among the dedicated card information for the card to be bound, the card type of the card to be bound and the life state of the card to be bound. In this case, the user terminal generates the first

message via multiple processes. Correspondingly, the server may receive the card binding message via multiple processes. Based on the received message in each of the multiple processes, the server can determine the card type of the card to be bound, and perform verification of the card to be bound. The details about these can be found in the detailed description hereinabove and hereinafter.

[00131] In a case that the verification of the card to be bound fails, the server will not allocate the dedicated card information for the card to be bound.

[00132] In some examples, the card authentication information may include a card identifier, card type verification information, and an input verification code.

[00133] The dedicated card information includes a card transaction identifier. The dedicated card information for the card to be bound together with matching universal personalization data are useable to perform transaction verification of the card to be bound. The matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the card to be bound.

[00134] The dedicated card information allocated by the server for different cards of the same card type is different. However, different cards of the same card type have the same matching universal personalization data, that is, different cards of the same card type share the same matching universal personalization data.

[00135] In some examples, in a case that the card to be bound includes a first card and a second card with a same card type, the dedicated card information for the first card together with target matching universal personalization data are useable to perform transaction verification of the first card, and the dedicated card information for the second card together with the target matching universal personalization data are useable to perform transaction verification of the second card. The target matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the first card.

[00136] In some examples, the dedicated card information may further includes dedicated personalization data. The details of the dedicated personalization data can be found in relevant descriptions in the above embodiments, and will not be repeated here.

[00137] In step S302, a card binding response message is sent to the user terminal.

[00138] The card binding response message includes the card type of the card to be bound and dedicated card information for the card to be bound. The card binding response message may include multiple messages, each of which may has different contents. The details about

[00197] In step S417, the TSM server sends the dedicated card information for the card to be bound to the back-end server.

[00198] In step S418, the back-end server sends a third response message to the application.

[00199] The third response message includes the dedicated card information for the card to be bound.

[00200] In step S419, the application sends the dedicated card information for the card to be bound to the security element.

[00201] In step S420, the security element stores the dedicated card information for the card to be bound, sets a life state of the card to be bound, and generates a first mapping relationship.

[00202] In step S421, the security element sends the life state of the card to be bound to the application.

[00203] In step S422, the application sends the life state of the card to be bound to the back-end server.

[00204] In step S423, the back-end server sends a first feedback message to the application.

[00205] The first feedback message indicates that the life state of the card to be bound has been successfully synchronized in the server.

[00206] In step S424, the application sends a prompt message to prompt the user that the card has been successfully bound.

[00207] Fig. 14 is a flowchart of an example of a card deleting process according to an embodiment of the present application. As shown in Fig. 14, the card deleting process may include steps S501 to S509.

[00208] In step S501, the application receives a card deleting input.

[00209] The card deleting input selects a card to be deleted.

[00210] In step S502, the application sends a card deleting request message to the back-end server.

[00211] The card deleting request message indicates the card to be deleted.

[00212] In step S503, the back-end server feeds back a card transaction identifier of the card to be deleted to the application.

[00213] In step S504, the application sends the card transaction identifier of the card to be deleted to the security element.

[00214] In step S505, the security element updates a life state of the card to be deleted to an invalid state according to a mapping relationship corresponding to the card to be deleted.

[00215] In step S506, the security element sends the life state of the card to be deleted to the

application.

[00216] In step S507, the application sends the life state of the card to be deleted to the back-end server.

[00217] In step S508, the back-end server sends a second feedback message to the application.

[00218] The second feedback message indicates that the life state of the card to be deleted has been successfully synchronized in the server.

[00219] In step S509, the application sends a prompt message to prompt a user that the card has been successfully deleted.

[00220] Fig. 15 is a flowchart of another example of a card binding process according to an embodiment of the present application. The card is a card that has ever been bound to the application in the user terminal but has been deleted. As shown in Fig.15, the card binding process may include steps S601 to S610.

[00221] In step S601, the application receives a card binding input.

[00222] The card binding input may include a card identifier of a card to be bound.

[00223] In step S602, the application sends a first card binding message to the back-end server.

[00224] The first card binding message includes the card identifier of the card to be bound.

[00225] In step S603, the back-end server determines, according to the card identifier, that the card to be bound has been deleted, and obtains dedicated card information that has been allocated to the card to be bound, without need to regenerate dedicated card information for the card to be bound.

[00226] In step S604, the back-end server sends the dedicated card information for the card to be bound to the application.

[00227] In step S605, the application sends the dedicated card information for the card to be bound to the security element.

[00228] In step S606, the security element stores the dedicated card information for the card to be bound, and updates a life state of the card to be bound.

[00229] In step S607, the security element sends the life state of the card to be bound to the application.

[00230] In step S608, the application sends the life state of the card to be bound to the back-end server.

[00231] In step S609, the back-end server sends a third feedback message to the application.

verification of the second card. The target matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the first card.

[00241] In the embodiment of the application, the security element of the user terminal stores the first-type universal personalization data and the second-type universal personalization data. In a process of binding a card, the user terminal interacts with the server through the card binding message to obtain the dedicated card information allocated for the card by the server. The obtained dedicated card information is loaded onto the security element. In the process of binding the card, there is no need to download all the personalization data required for transaction verification of the card. When the card is used for a payment transaction, the user terminal and the server can use the matching universal personalization data and the dedicated card information for the card for transaction verification. The user terminal does not need to repeatedly store the universal personalization data required for transaction verification for each card, thereby reducing occupation of storage resources of the user terminal.

[00242] Moreover, except for the case that the application in the user terminal binds a card for the first time, contents needed to be downloaded by the user terminal in the process of binding the card does not include the universal personalization data, which reduces time spent by the user terminal for downloading data. Therefore, binding of the card can be accelerated, and becomes faster and simpler.

[00243] In some examples, the card binding message includes a first card binding message, a second card binding message and a binding verification message. The card authentication information includes a card identifier, card type verification information and an input verification code.

[00244] The sending module 701 may be further configured to: send the first card binding message to the server, the first card binding message including the card identifier of the card to be bound; send the second card binding message to the server upon reception of a first response message sent by the server, the first response message including the card type of the card to be bound, and the second card binding message including the card type verification information of the card to be bound; and send the binding verification message to the server upon reception of a second response message sent by the server, the second response message including a dynamic verification code, and the binding verification message including the input verification code.

[00245] Specifically, in a case that the card type of the card to be bound is a first-type, the

identifier of the card to be deleted fed back by the server.

[00260] The processing module 703 may be further configured to control the security element to update a life state of the card to be deleted to an invalid state.

[00261] The synchronization module 704 may be further configured to synchronize the updated life state of the card to be deleted to the server.

[00262] In some examples, the processing module 703 may be further configured to delete a second mapping relationship, the second mapping relationship including a mapping relationship among dedicated card information for the card to be deleted, a card type of the card to be deleted, and a life state of the card to be deleted.

[00263] A fourth aspect of the present application provides a server. The server stores the first-type universal personalization data and the second-type universal personalization data. Fig. 18 is a schematic structural diagram of an embodiment of the server according to the fourth aspect of the present application. As shown in Fig. 18, the server 800 may include a receiving module 801, a processing module 802 and a sending module 803.

[00264] The receiving module 801 is configured to receive the card binding message sent by the user terminal.

[00265] The card binding message includes card authentication information of the card to be bound.

[00266] The processing module 802 is configured to determine a card type of the card to be bound and verifying the card to be bound in response to the card binding message, and allocate dedicated card information for the card to be bound in a case of successful verification.

[00267] The dedicated card information includes a card transaction identifier. The dedicated card information for the card to be bound together with matching universal personalization data are useable to perform transaction verification of the bound card. The matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the bound card.

[00268] In some examples, the dedicated card information further includes dedicated personalization data.

[00269] The sending module 803 is configured to send a card binding response message to the user terminal.

[00270] The card binding response message includes the card type of the d card to be bound and dedicated card information for the card to be bound.

[00271] In some examples, in a case that the card to be bound includes a first card and a

What is claimed is:

1. A card management method, applicable to a user terminal, the user terminal including a security element which stores first-type universal personalization data and second-type universal personalization data, the method comprising:

sending, to a server, a card binding message including card authentication information of a card to be bound such that the server determines a card type of the card to be bound and allocates dedicated card information for the card to be bound, wherein the dedicated card information comprises a card transaction identifier;

receiving the card type of the card to be bound and the dedicated card information for the card to be bound sent by the server; and

loading the dedicated card information for the card to be bound onto the security element, wherein the dedicated card information for the card to be bound together with matching universal personalization data are useable to perform transaction verification of the card to be bound, and the matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the card to be bound.

2. The method according to claim 1, wherein in a case that the card to be bound comprises a first card and a second card with a same card type, the dedicated card information for the first card together with target matching universal personalization data are useable to perform transaction verification of the first card, and the dedicated card information for the second card together with the target matching universal personalization data are useable to perform transaction verification of the second card; and

wherein the target matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the first card.

3. The method according to claim 1, further comprising setting a life state of the card to be bound to a valid state through the security element, and generating or updating a first mapping relationship, wherein the first mapping relationship comprises a mapping relationship among the dedicated card information for the card to be bound, the card type of the card to be bound, and the life state of the card to be bound.

4. The method according to claim 3, further comprising synchronizing the set life state of the card to be bound and the first mapping relationship to the server.

5. The method according to claim 1, wherein the card binding message comprises a first card binding message, a second card binding message and a binding verification message, and the card authentication information comprises a card identifier of the card to be bound, card type verification information of the bound card and an input verification code; and

wherein the sending of the card binding message to the server comprises:

sending the first card binding message to the server, the first card binding message comprising the card identifier of the card to be bound;

sending the second card binding message to the server upon reception of a first response message sent by the server, the first response message comprising the card type of the card to be bound, and the second card binding message comprising the card type verification information of the bound card; and

sending the binding verification message to the server upon reception of a second response message sent by the server, the second response message comprising a dynamic verification code, and the binding verification message comprising the input verification code.

6. The method according to claim 5, wherein in a case that the card type of the card to be bound is a first-type, the card type verification information comprises first-type verification information;

wherein in a case that the card type of the card to be bound is a second-type, the card type verification information comprises second-type verification information; and

wherein the first-type verification information is different from the second-type verification information.

7. The method according to claim 1, wherein in a case that the user terminal performs card binding for the first time, the method further comprises:

receiving a universal program data package sent by the server; and

loading the universal program data package onto the security element.

8. The method according to claim 3, further comprising:

receiving a default card selection input which indicates a selected default card from multiple cards which have been bound;

controlling the security element to update a life state of an original default card to a valid state, wherein the life state of the original default card before being updated is a default use state;

controlling the security element to update a life state of the selected default card to the default use state; and

synchronizing the updated life state of the selected default card and the updated life state of the original default card to the server.

9. The method according to claim 3, further comprising:

sending a card deleting request message to the server such that the server feeds back a card transaction identifier of a card to be deleted among multiple cards which have been bound , the card deleting request message indicating the card to be deleted;

receiving the card transaction identifier of the card to be deleted fed back by the server;

controlling the security element to update a life state of the card to be deleted to an invalid state; and

synchronizing the updated life state of the card to be deleted to the server.

10. The method according to claim 9, wherein after the receiving of the card transaction identifier of the card to be deleted fed back by the server, the method further comprises:

deleting a second mapping relationship, the second mapping relationship comprising a mapping relationship among dedicated card information for the card to be deleted, a card type of the card to be deleted, and the life state of the card to be deleted.

11. The method according to claim 1, wherein the dedicated card information further comprises dedicated personalization data.

12. A card management method, applicable to a server, the server storing first-type universal personalization data and second-type universal personalization data, the method comprising:

determining a card type of a card to be bound and verifying the card to be bound in response to a received card binding message sent by a user terminal, and allocating dedicated

card information to the card to be bound in a case of successful verification, wherein the card binding message comprises card authentication information of the card to be bound, the dedicated card information comprises a card transaction identifier, the dedicated card information for the card to be bound together with matching universal personalization data are useable to perform transaction verification of the card to be bound, and the matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the bound card; and

sending a card binding response message to the server, the card binding response message comprising the card type of the card to be bound and the dedicated card information for the card to be bound.

13. The method according to claim 12, wherein in a case that the card to be bound comprises a first card and a second card with a same card type, the dedicated card information for the first card together with target matching universal personalization data are useable to perform transaction verification of the first card, and the dedicated card information for the second card together with the target matching universal personalization data are useable to perform transaction verification of the second card; and

wherein the target matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the first card.

14. The method according to claim 12, further comprising keep synchronization with the user terminal in a life state of the card to be bound and a first mapping relationship, the first mapping relationship comprising a mapping relationship among the dedicated card information for the card to be bound, the card type of the card to be bound and the life state of the card to be bound.

15. The method according to claim 12, wherein the card binding response message comprises a first response message, a second response message and a third response message, and the card authentication information comprises a card identifier of the card to be bound, card type verification information of the card to be bound and an input verification code;

wherein the sending of the card binding response message to the server comprises:

sending the first response message to the user terminal upon determining the card type of the card to be bound in response to a first card binding message sent by the user terminal, the first response message comprising the card type of the card to be bound, and the first card binding message comprising the card identifier of the card to be bound;

sending the second response message to the user terminal in a case of successful verification of the card type of the card to be bound in response to a second card binding message sent by the user terminal, the second card binding message comprising the card type verification information that matches the card type of the card to be bound, and the second response message comprising a dynamic verification code; and

sending the third response message to the user terminal in a case of successful verification of a verification code of the card to be bound in response to a binding verification message sent by the user terminal, the binding verification message comprising the input verification code, and the third response message comprising the dedicated card information for the card to be bound.

16. The method according to claim 15, wherein the card binding message comprises the first card binding message, the second card binding message and the binding verification message;

wherein the determining of the card type of the card to be bound, verifying of the card to be bound in response to the card binding message sent by the user terminal and allocating of the dedicated card information for the card to be bound in a case of successful verification comprises:

determining the card type of the card to be bound according to the card identifier of the card to be bound in the first card binding message;

performing card type verification on the card to be bound using the card type verification information of the card to be bound in the second card binding message; and

allocating the dedicated card information for the card to be bound in a case that the input verification code in the binding verification message is consistent with the dynamic verification code.

17. The method according to claim 16, wherein in a case that the card type of the card to be bound is a first type, the card type verification information comprises a first-type verification information;

wherein in a case that the card type of the card to be bound is a second-type, the card type verification information comprises a second-type verification information; and

wherein the first-type verification information is different from the second-type verification information.

18. The method according to claim 12, wherein in a case that the user terminal performs card binding for the first time, the method further comprises sending a universal program data package to the user terminal.

19. The method according to claim 12, further comprising keep in synchronization with the user terminal in an updated life state of a default card and an updated life state of an original default card, wherein a life state of the original default card before being updated is a default use state.

20. The method according to claim 12, further comprising feeding back a card transaction identifier of a card to be deleted among multiple cards that have been bounded to the user terminal in response to a card deleting request message sent by the user terminal, the card deleting request message indicating the card to be deleted; and

keeping in synchronization with the user terminal in an updated life state of the card to be deleted.

21. The method according to claim 20, further comprising deleting a second mapping relationship, wherein the second mapping relationship comprises a mapping relationship among dedicated card information for the card to be deleted, a card type of the card to be deleted, and the life state of the card to be deleted.

22. The method according to claim 12, wherein the dedicated card information further comprises dedicated personalization data.

23. A user terminal, including a security element which stores first-type universal personalization data and second-type universal personalization data;

wherein the user terminal comprises:

a sending module configured to send, to a server, a card binding message including card authentication information of a card to be bound such that the server determines a card type of the card to be bound, and in a case of successful verification of the card, allocates dedicated card information for the card to be bound, wherein the dedicated card information comprises a card transaction identifier;

a receiving module configured to receive the card type of the card to be bound and the dedicated card information for the card to be bound sent by the server; and

a processing module configured to load the dedicated card information for the card to be bound onto the security element, wherein the dedicated card information for the card to be bound together with matching universal personalization data are useable to perform transaction verification of the card to be bound, and the matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the card to be bound.

24. A server storing first-type universal personalization data and second-type universal personalization data;

wherein the server comprises:

a receiving module configured to receive a card binding message sent by a user terminal, wherein the card binding message comprises card authentication information of a card to be bound;

a processing module configured to determine a card type of the card to be bound and verify the to be bound card in response to the card binding message, and allocate dedicated card information to the card to be bound in a case of successful verification, wherein the dedicated card information comprises a card transaction identifier, the dedicated card information for the card to be bound together with matching universal personalization data are useable to perform transaction verification of the card to be bound, and the matching universal personalization data is the first-type universal personalization data or the second-type universal personalization data that matches the card type of the card to be bound; and

a sending module configured to send a card binding response message to the server, wherein the card binding response message comprises the card type of the card to be bound and the dedicated card information for the card to be bound.

25. A user terminal, comprising a processor and a memory storing computer program instructions, wherein the processor executes the computer program instructions to implement the card management method according to any one of claims 1 to 11.

26. A server, comprising a processor and a memory storing computer program instructions, wherein the processor executes the computer program instructions to implement the card management method according to any one of claims 12 to 22.

27. A card management system, comprising the user terminal according to claim 25 and the server according to claim 26.

28. A computer storage medium, storing computer program instructions, wherein the computer program instructions are executed by a processor to perform the card management method according to any one of claims 1 to 22.