

AUTHORITY REVOKING METHOD AND DEVICE

TECHNICAL FIELD

[0001] The present application relates to the field of authority control technologies, and in particular, to an authority revoking method and device.

5

BACKGROUND

[0002] The development of computer and Internet technologies enables users to enjoy many convenient functions. However, as users' requirements getting higher, a single functional entity sometimes cannot satisfy users' various requirements. In this case, a plurality of relatively independent functional entities need to cooperate in a certain way to jointly provide services for the users. For example, to satisfy users' shopping requirement, a plurality of parties such as an e-commerce platform, a payment platform, and a logistics platform need to cooperate.

[0003] To implement cooperation between functional entities that are originally independent of each other, in addition to exchanging necessary data, the functional entities usually need to grant some necessary authority to each other. For example, a user needs to modify, by using application A, information that is stored by the user in a storage directory of website B (here, assumed that application A and website B are provided by different vendors). To satisfy this requirement, website B needs to grant personal information data modification authority to application A.

[0004] In terms of security, an authorizing party should further have a function of actively revoking granted authority. A general practice is as follows: The authorizing party writes each piece of authority granted to an authorized party as a record to a database, adds one attribute field used to indicate whether the authority is valid to each record, and actively revokes the authority by modifying a value of the field to "invalid". A problem of this solution is as follows: Each time revoking a piece of authority, a field update operation needs to be performed on the database. If authority needs to be revoked in batches, related authorization records need to be updated one by one. In the database operation environment, actual efficiency of such processing is

relatively low. In addition, operation logic of "updating one by one" also needs to be compiled and implemented by high labor and time costs. If different authorization records are stored in different libraries or tables, corresponding data update operations further need to be related to library separation or table separation, further increasing
5 technology implementation difficulty.

SUMMARY

[0005] For the previous technical problem, the present application provides the following technical solutions:

[0006] According to a first aspect of the present application, a database
10 authorization record batch-revoking method is provided, where a time validity attribute is configured for an authorization record set needing revoking, the time validity attribute is used to describe a time point, so an authorization record whose authorization validation moment is earlier than the time point in the set becomes invalid, and the method includes: determining an authorization record set needing
15 revoking; and setting a time validity attribute value for the determined set, and setting the time validity attribute value T_0 to a current moment or a future moment.

[0007] According to a second aspect of the present application, a database authorization record batch-revoking method is provided, where a first time validity attribute or a second time validity attribute is configured for an authorization record
20 set needing revoking, the first time validity attribute is used to describe a first time point, so an authorization record whose authorization validation moment is earlier than the first time point in the set becomes invalid; the second time validity attribute is used to describe a second time point, so an authorization record whose authorization validation moment is later than the second time point in the set becomes
25 invalid, and the method includes: determining an authorization record set needing revoking; when the first time validity attribute is configured for the set, determining the latest validation moment of authorization records in the set and setting a first time validity attribute value for the set, so the first time validity attribute value T_1 > the latest validation moment of the authorization records in the set; when the second time
30 validity attribute is configured for the set, determining the earliest validation moment of authorization records in the set, and setting a second time validity attribute value for the set, so the second time validity attribute value T_2 < the earliest validation

moment of the authorization records in the set.

[0008] According to a third aspect of the present application, a database authorization record validity verification method is provided, where a first time validity attribute or a second time validity attribute is configured for an authorization record set needing revoking, the first time validity attribute is used to describe a first time point, so an authorization record whose authorization validation moment is earlier than the first time point in the set becomes invalid, the second time validity attribute is used to describe a second time point, so an authorization record whose authorization validation moment is later than the second time point in the set becomes invalid; and the method includes: determining an authorization record to be verified; obtaining a validation moment of the authorization record to be verified; obtaining a first time validity attribute value or a second time validity attribute value of an authorization record set that includes the authorization record to be verified; and determining that the authorization record to be verified is invalid, when the obtained validation moment is earlier than the obtained first time validity attribute value; or determining that the authorization record to be verified is invalid, when the obtained validation moment is later than the obtained second time validity attribute value.

[0009] According to a fourth aspect of the present application, a database authorization record batch-revoking device is provided, where a time validity attribute is configured for an authorization record set needing revoking, the time validity attribute is used to describe a time point, so an authorization record whose authorization validation moment is earlier than the time point in the set becomes invalid, and the device includes: a revoking task determining unit, configured to determine an authorization record set needing revoking; and a revoking execution unit, configured to set a time validity attribute value for the determined set, and set the time validity attribute value T_0 to a current moment or a future moment.

[0010] According to a fifth aspect of the present application, a database authorization record batch-revoking device is provided, where a first time validity attribute or a second time validity attribute is configured for an authorization record set needing revoking, the first time validity attribute is used to describe a first time point, so an authorization record whose authorization validation moment is earlier than the first time point in the set becomes invalid, the second time validity attribute is used to describe a second time point, so an authorization record whose authorization validation moment is later than the second time point in the set becomes invalid, and

the device includes: a revoking task determining unit, configured to determine an authorization record set needing revoking; and a revoking execution unit, configured to determine the latest validation moment of authorization records in the set and set a first time validity attribute value for the set when the first time validity attribute is configured for the set, so the first time validity attribute value $T_1 >$ the latest validation moment of the authorization records in the set; or determine the earliest validation moment of authorization records in the set and set a second time validity attribute value for the set, when the second time validity attribute is configured for the set, so the second time validity attribute value $T_2 <$ the earliest validation moment of the authorization records in the set.

[0011] According to a sixth aspect of the present application, a database authorization record validity verification device is provided, where a first time validity attribute or a second time validity attribute is configured for an authorization record set needing revoking, the first time validity attribute is used to describe a first time point, so an authorization record whose authorization validation moment is earlier than the first time point in the set becomes invalid, the second time validity attribute is used to describe a second time point, so an authorization record whose authorization validation moment is later than the second time point in the set becomes invalid, and the device includes: a verification object determining unit, configured to determine an authorization record to be verified; a first acquisition unit, configured to obtain a validation moment of the authorization record to be verified; a second acquisition unit, configured to obtain a first time validity attribute value or a second time validity attribute value of an authorization record set that includes the authorization record to be verified; and a verification unit, configured to determine that the authorization record to be verified is invalid, when the obtained validation moment is earlier than the obtained first time validity attribute value; or determine that the authorization record to be verified is invalid, when the obtained validation moment is later than the obtained second time validity attribute value.

[0012] According to a seventh aspect of the present application, an authorization token batch-revoking method for a third-party application is provided, and the method includes: determining a third-party application needing authorization token revoking; obtaining a current moment, and setting a time validity attribute value for the determined third-party application, so the time validity attribute value $T_0 =$ current moment; receiving an authorization token provided by the third-party application;

obtaining a validity attribute value of the authorization token; and further obtaining a validation moment of the authorization token and the time validity attribute value of the third-party application if the validity attribute value indicates valid, and determining that the authorization token is invalid if the obtained validation moment is earlier than the obtained time validity attribute value; otherwise, determining that the authorization token is valid.

[0013] According to the technical solutions provided in the present application, a time validity attribute is configured for an authority record set needing batch-revoking. As such, an authority record can be updated to an invalid state by modifying a value of the attribute, thereby revoking the authority record. This newly added attribute does not affect the meaning of an existing attribute. In addition, because the attribute is configured specifically for a set of records, states of all records in the set can be updated by performing one modification operation instead of updating authorization records in a database one by one, and different storage locations of a plurality of authorization records have no impact on the complexity of updating and/or revoking the records in a set. Therefore, authority batch-revoking processing efficiency is effectively improved, and technology implementation difficulty is reduced.

[0014] It should be understood that the previous general description and the following detailed description are merely an example for explanation, and do not limit the present application.

BRIEF DESCRIPTION OF DRAWINGS

[0015] To describe the technical solutions in the implementations of the present application or in the existing technology more clear, the following briefly introduces the accompanying drawings needed for describing the implementations or the existing technology. Apparently, the accompanying drawings in the following description merely show some implementations recorded in the present application, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0016] FIG. 1 is a schematic flowchart illustrating an authority revoking method, according to the present application;

[0017] FIG. 2 is a schematic flowchart illustrating an authority verification method, according to the present application;

[0018] FIG. 3 is a combined schematic flowchart illustrating an authority granting, revoking, and verification method, according to the present application;

[0019] FIG. 4 is a schematic structural diagram illustrating an authority revoking device, according to the present application;

5 [0020] FIG. 5 is a first schematic structural diagram illustrating an authority verification device, according to the present application; and

[0021] FIG. 6 is a second schematic structural diagram illustrating an authority verification device, according to the present application.

DESCRIPTION OF IMPLEMENTATIONS

10 [0022] To make a person skilled in the art better understand the technical solutions in the present application, the following describes the technical solutions in the implementations of the present application in detail with reference to the accompanying drawings in the implementations of the present application. Apparently, the described implementations are merely some rather than all of the implementations
15 of the present application. All other implementations obtained by a person of ordinary skill in the art based on the implementations of the present application shall fall within the protection scope of the present application.

[0023] For an authorizing party, each piece of authority that has been granted to an authorized party is stored in a database in the form of a data record. Table 1 shows
20 an example of a data structure of an authority record. It is worthwhile to note that a field attribute shown in the table is merely used for illustrative description, and constitutes no limitation on the solution.

Table 1

Authority ID	Authority type	Authorized party	User ID	Validation moment	Validity
100001	Read	Application 1	User 001	2016/06/01	Valid
100002	Write	Application 2	User 002	2016/06/10	Valid
100003	Write	Application 1	User 002	2016/06/10	Valid
...	...	...		...	...

[0024] According to an authority management method in the existing technology, the authorizing party controls whether authority is valid by using an attribute field "validity" in each authority record. When authority needs to be revoked in batches, assume that n authority records need to be revoked, attribute fields "validity" corresponding to the authority records need to be modified one by one. Theoretical algorithm complexity in completing all revoking processing is $O(n)$. Actual processing difficulty is further improved if the n records are stored in different libraries or tables.

[0025] For the previous problem, a solution provided in the present application is as follows: Authority records that need to be revoked in batches are considered as one set, and an attribute used for overall validity control is configured for the set, so authority can be revoked in batches through one-time modification.

[0026] Based on actual demands, authorization records that need to be revoked in batches generally has certain "generality", and the "generality" indicates that a plurality of authority records have one or more same attribute values. For example, "the authorization records that need to be revoked in batches belong to application 1", or "the authorization records that need to be revoked in batches belong to user 001".

[0027] After a set whose authority needs to be revoked in batches is determined, a new attribute is configured for the set, and the attribute is used to collectively indicate validity of all authority records in the set. To avoid a meaning conflict between the newly added attribute and an existing attribute "validity" in a single record, the newly added attribute is described by adding a time dimension in the present application. A newly added attribute of a certain given set can indicate the earliest authorization validation moment allowed to validate an authorization record set. Based on actual demands, a data type of the newly added attribute can be "date", "time", or time data types. Regardless of the data type, a value of the newly added attribute is considered as a specific time point corresponding to the value.

[0028] To be specific, the value of the attribute describes a time point T_0 . When authority validity verification is performed subsequently, if a validation moment of a certain authority record in the set is earlier than the time point, the authority record is considered to be invalid. In this case, an attribute value of "validity" of the record no longer works. Otherwise, if a validation moment of a certain authority record in the set is equal to or later than the time point, validity of the record can be further determined based on an attribute value of "validity" of the record. As such, one-time

invalidation processing can be performed on states of all records in the set by modifying the attribute value of the newly added attribute.

[0029] Table 1 is used as an example. Assume that the three current valid authority records shown in Table 1 form a set (actual meanings corresponding to the set are temporarily not considered here, and the set is used for illustrative description purpose only), if "the earliest authorization validation moment" of the set is assigned to "2016/06/12", validation moments of the three authority records in Table 1 are earlier than the moment. Therefore, all the three authority records are considered to be invalid, and the three authorization records are revoked in batches through one-time data modification.

[0030] In conclusion, as shown in FIG. 1, an authority batch-revoking method provided in the present application can include the following steps.

[0031] S101. Determine an authorization record set needing revoking.

[0032] "Set" here is described based on actual demands. For example, if authority needs to be revoked in batches based on an "authorized party" category, authority records for different authorized parties can be described as different sets; or if authority needs to be revoked in batches based on a "user" category, authority records for different users can be described as different sets. Certainly, "set" can also be jointly described by using a plurality of attribute dimensions. Examples are omitted here for simplicity.

[0033] S102. Set a time validity attribute value for the determined set, to revoke authorization records in the set in batches.

[0034] For ease of description, "time validity attribute" is used to represent the previous "earliest authorization validation moment" below. It is worthwhile to note that "time validity attribute" is specific to the set, and should not be confused with an attribute such as "validation moment" or "invalidation moment" specific to a single record.

[0035] It can be seen from the previous definition of "time validity attribute" that theoretically, the authorization records in the set can be revoked in batches provided that a modified time validity attribute value satisfies the following condition: $T_0 > \text{the latest validation moment of the authorization records in the set}$.

[0036] It is worthwhile to note that, mathematical expressions ">" and "<" of data such as date or time can be respectively explained as "later" and "earlier". Table 1 is used as an example. In the three authorization records, the latest validation moment is

2016/06/10. Therefore, authority can be revoked in batches provided that the value of T_0 is later than 2016/06/10, for example, $T_0=2016/06/11$, $T_0=2016/06/12$, or $T_0=2016/06/13$.

[0037] According to an implementation of the present application, when a batch-revoking operation needs to be performed, the latest validation moment of the authorization records in the set can be first determined, and then T_0 can be set to any value provided that " $T_0 > \text{the latest validation moment of the authorization records in the set}$ ".

[0038] In practice, T_0 can be directly set to a current moment or a future moment to avoid the step of "determining the latest validation moment of the authorization records in the set".

[0039] If authority corresponding to the set is only revoked temporarily, and can still be re-granted subsequently, the value range of T_0 can be further limited: Assume that the next earliest authorization validation moment of each record in the set can be predetermined, the time validity attribute value T_0 needs to further satisfy the following condition: $T_0 < \text{the next earliest authorization validation moment}$. Table 1 is still used as an example. Assume that it is known that authority starts to be re-granted on 2016/06/20 after authority is revoked in batches this time, T_0 needs to be earlier than 2016/06/20 to ensure that authorization after 2016/06/20 is not affected by the time validity attribute value.

[0040] According to an implementation of the present application, when a batch-revoking operation needs to be performed, obtain a current moment and directly set $T_0 = \text{current moment}$. Because "current moment" satisfies both " $T_0 < \text{the next earliest authorization validation moment}$ " and " $T_0 > \text{the latest validation moment of the authorization records in the set}$ ", the value of T_0 can not only ensure that authority in the set can be revoked in batches, but also have no impact on subsequent reauthorization.

[0041] It can be understood that, to implement a moment comparison, the time of the attribute value can be more precise based on actual demands, for example, precise to an hour/a minute/a second. Further details are omitted here.

[0042] Corresponding to the previous authority batch-revoking method, the present application further provides a database authorization record validity verification method. As shown in FIG. 2, the method can include the following steps:

[0043] S201. Determine an authorization record to be verified.

[0044] S202. Obtain a validation moment of the authorization record to be verified.

[0045] S203. Obtain a time validity attribute value of an authorization record set that includes the authorization record to be verified.

5 [0046] S204. Determine that the authorization record to be verified is invalid, when the obtained validation moment is earlier than the obtained time validity attribute value.

[0047] According to the solution of the present application, an authorizing party can determine, based on authority verification information provided by a
10 to-be-verified party, an authorization record corresponding to the verification information and a set that includes the authorization record, and then further obtain related data for comparison to verify authority validity. In comparison with the existing technology, in the present application, to verify the validity of an authorization record, a set that includes the authorization record needs to be identified
15 and then whether the authorization record is valid is determined based on "time validity attribute" of the set. Two cases can be involved here:

[0048] (1) Only "time validity attribute" added in the present application is used to verify whether the authorization record is valid:

[0049] In this case, an attribute "validity" specific to a single authorization record
20 in the existing technology is no longer considered. If a validation moment of a single record is earlier than a time validity attribute value of the set, the authorization record is determined to be invalid; otherwise, if a validation moment of a single record is later than or equal to a time validity attribute value of the set, the authorization record is determined to be valid.

25 [0050] (2) "Validity" specific to a single authorization record and "time validity attribute" are jointly used to verify whether the authorization record is valid:

[0051] In this case, the authorization record is determined to be valid only when two conditions "a state of a single record is valid" and "a validation moment of a single record is later than or equal to a time validity attribute value of a set" are
30 simultaneously satisfied. Otherwise, the authorization record is considered to be invalid. In practice, a sequence of determining the two conditions does not need to be limited.

[0052] In the previous implementation, "time validity attribute" is used to describe a time point, so an authorization record whose authorization validation moment is

earlier than the time point in the set becomes invalid. Based on a similar idea, "time validity attribute" can have another definition. For example, in an implementation of the present application, "time validity attribute" can be described as the latest authorization validation moment allowed to validate the authorization record set.

5 **[0053]** To be specific, the value of the attribute describes a time point T_0 . When authority validity verification is subsequently performed, an authority record in the set is considered to be invalid if a validation moment of the authority record is later than the time point.

10 **[0054]** Correspondingly, when a batch-revoking operation needs to be performed, a processing method changes (relative to S102) to the following method: The earliest validation moment of authorization records in the set is first determined, and then T_0 can be set to any value provided that " $T_0 >$ the earliest validation moment of authorization records in the set".

15 **[0055]** Further, when record validity needs to be verified, determining logic changes (relative to S204) to the following: The authorization record to be verified is determined to be invalid when the obtained validation moment is later than the obtained time validity attribute value.

20 **[0056]** In this implementation, although the definition of the time validity attribute is changed, overall processing logic is basically similar to that of the previous implementation. Therefore, this implementation only focuses on differences, and the other parts are omitted for simplicity.

25 **[0057]** In addition, it is worthwhile to note that "validation moment" described in the solution of the present application is a time point at which certain authority is generated or starts to function, and should not be construed as a field "validation moment" in a database in a narrow sense. For example, in different databases, an attribute field name used to indicate the previous time point can be "authority validation moment/time/date", "authority creation moment/time/date", "authorization moment/time/date", etc. It does not affect implementation of the solutions of the present application.

30 **[0058]** In addition, in some cases, a plurality of different fields in a database can have similar functions. For example, there are two attribute fields: "authority creation moment" and "authority validation moment" in some databases. The latter can be slightly delayed than the former. However, based on actual demands, both "creation" and "validation" are usually completed when authority needs to be performed in

batches. In this case, when the solution of the present application is applied, using the two attribute fields as "validation moment" in the previous solution can achieve the same authority batch-revoking effects and authority validity verification effects. These cases should also fall within the protection scope of the present application.

5 **[0059]** The following describes the technical solutions provided in the present application with reference to a specific application example.

[0060] The Open Authorization (OAUTH) protocol provides a secure, open and simple standard for authorization of user resources. For an authorizing party, authority is granted to an authorized party by using OAUTH, so the authorized party can visit
10 user resources of the authorizing party without visiting user account information (such as a user name and a password).

[0061] Assume that a current scenario is that a user modifies, by using a certain image processing application A1, a picture stored by the user on website B. To satisfy the previous requirement, website B needs to grant picture data modification authority
15 to the image processing application A1. An authorization process is shown in FIG. 3 according to the OAUTH protocol.

[0062] S301. A user requests to modify, by using application A1, a picture stored by the user on website B.

[0063] S302. Application A1 requests website B to authorize application A1 to
20 visit the picture of the user.

[0064] S303. Website B queries the user whether to authorize application A1 to visit the picture.

[0065] S304. The user determines authorization.

[0066] S305. Website B returns a token to application A1.

25 **[0067]** S306. Application A1 visits the picture of the user by using the token.

[0068] One token corresponds to one authority record granted by an authorizing party to an authorized party. For the authorizing party, token data is stored in a database, and a key attribute of one piece of token data can include the following:

[0069] Token identifier: used to uniquely distinguish between tokens.

30 **[0070]** Authorized party identifier: indicates an authorized party that is issued with a token, for example, "image processing application A1".

[0071] User identifier: indicates a user who has a token, for example, "Zhang San".

[0072] Token validation moment: A value is a specific time point, for example,

2016/06/01 and 2016/06/01-13:00:00.

[0073] Token invalidation moment: A value is a specific time point, for example, 2017/06/01 and 2017/06/01-13:00:00.

[0074] Current token validity: A value indicates valid or invalid.

- 5 [0075] In actual practice, a large number of users modify their pictures on website B by using application A1. Therefore, website B needs separately issue tokens specific to the users to application A1. As shown in Table 2, issued tokens are recorded in a token data table:

Table 2

Token identifier	Authorized party	User	Validation moment	Invalidation moment	Validity
100001	Application A1	Zhang San	2016/06/01	2017/06/01	Valid
100002	Application A1	Li Si	2016/06/10	2017/06/10	Valid
100003	Application A1	Wang Wu	2016/06/12	2017/06/12	Valid
100004	Application A2	Zhao Liu	2016/06/02	2017/06/30	Valid
100005	Application A3	Xiaoming	2016/06/05	2017/07/05	Valid
...	...	...	...	...	...

10

[0076] Assume that website B wants to revoke, based on "authorized party"-category, tokens in batches that have been issued to a certain authorized application or some authorized applications, according to the solution of the present application, different token sets can be divided based on different authorized parties, and a corresponding attribute "earliest authorization validation moment" can be configured for each token set.

15

[0077] A specific implementation is as follows: As shown in Table 3, an attribute "earliest authorization validation moment" is added to an authorized application attribute table (the table can be created if not exist):

20

Table 3

Authorized party	Earliest authorization validation moment
Application A1	

Application A2	
Application A3	
...	...

[0078] Assume that for a purpose of security, website B needs to revoke all tokens in batches authorized to application A1, according to the solution of the present application, as shown in S307 in FIG. 3, website B can directly modify an attribute value of "earliest authorization validation moment" corresponding to "application A1" in the authorized application attribute table. A method in which the attribute value is directly modified to a current moment is used here. Assume that the current moment is 2017/06/15, a result obtained after the attribute value is written is shown in Table 4:

Table 4

Authorized party	Earliest authorization validation moment
Application A1	2017/06/15
Application A2	
Application A3	
...	...

10

[0079] As such, a batch-revoking operation on the tokens of application A1 is completed. The operation affects the first three data records (the authorized party is application A1) in Table 2 without modifying attributes "validity" of the three records one by one.

15 [0080] As shown in S308 and S309 in FIG. 3, when application A1 re-visits website B subsequently by using the token, website B obtains a validation moment of the token by querying a token attribute table, and obtains the earliest authorization validation moment of A1 by querying the authorized application attribute table. "Validation moments" of the first three token records shown in Table 2 are all earlier
20 than "earliest authorization validation moment" of application A1. Therefore, all the three tokens are actually considered to be invalid.

[0081] In addition, if A1 needs to be reauthorized subsequently, an authorization validation moment of A1 is later than the currently recorded "earliest authorization

validation moment" (here, assume that token revoking and token reissuing do not occur on the same day, and data precision of related attribute values can be adaptively adjusted if on the same day). Therefore, validity of a token subsequently reissued to application A1 is not affected by "earliest authorization validation moment".

5 **[0082]** In practice, comprehensive determination can be performed by combining "validity" of the token and "earliest authorization validation moment" of the token set. The token is determined to be valid only when two conditions "a state of the token is valid" and "a validation moment of the token is later than or equal to the earliest authorization validation moment of the token set" are both satisfied. If any of the two
10 conditions cannot be satisfied, the token is considered to be invalid. Certainly, a sequence of determining the previous two conditions does not need to be limited.

[0083] It can be seen that in comparison with the existing technology, in the solution of the present application, for n authority records that need to be revoked in batches, theoretical algorithm complexity corresponding to revoking processing can
15 be reduced from $O(n)$ to $O(1)$. In addition, because authority batch-revoking can be implemented by performing one-time data modification on the database in the solution of the present application, a plurality of complex cases in which authority records are stored in separate libraries or separate tables do not need to be considered, effectively reducing actual technology implementation difficulty of the solution.

20 **[0084]** Corresponding to the previous method implementation, the present application further provides a database authorization record batch-revoking device. As shown in FIG. 4, the device can include: a revoking object determining unit 110, configured to determine an authorization record set needing revoking; and a revoking execution unit 120, configured to set a time validity attribute value for the determined
25 set, and set the time validity attribute value T_0 to a current moment or a future moment.

[0085] In a specific implementation of the present application, the authorization record set can include authorization records for the same authorized party.

[0086] In a specific implementation of the present application, when the next
30 earliest authorization validation moment of the authorization record set needing revoking can be determined, the revoking execution unit 120 can be configured to set the time validity attribute value for the determined set, so the modified attribute value T_0 further satisfies the following condition: the latest validation moment of authorization records in the set $<T_0<$ the next earliest authorization validation moment.

[0087] In a specific implementation of the present application, the revoking execution unit 120 can be configured to obtain the current moment, and set the time validity attribute value for the determined set, so the modified attribute value T_0 =current moment.

5 [0088] In another specific implementation of the present application, the revoking execution unit 120 can be alternatively configured to: determine the latest validation moment of authorization records in the set, and set a first time validity attribute value for the set when a first time validity attribute is configured for the set, so the first time validity attribute value T_1 >the latest validation moment of the authorization records in
10 the set; or when a second time validity attribute is configured for the set, determine the earliest validation moment of authorization records in the set, and set a second time validity attribute value for the set, so the second time validity attribute value T_2 <the earliest validation moment of the authorization records in the set.

[0089] The first time validity attribute is used to describe a first time point, so an
15 authorization record whose authorization validation moment is earlier than the first time point in the set becomes invalid, and the second time validity attribute is used to describe a second time point, so an authorization record whose authorization validation moment is later than the second time point in the set becomes invalid.

[0090] The present application further provides a database authorization record
20 validity verification device. As shown in FIG. 5, the device can include: a verification object determining unit 210, configured to determine an authorization record to be verified; a first acquisition unit 220, configured to obtain a validation moment of the authorization record to be verified; a second acquisition unit 230, configured to obtain a first time validity attribute value or a second time validity attribute value of an
25 authorization record set that includes the authorization record to be verified; and a verification unit 240, configured to determine that the authorization record to be verified is invalid, when the obtained validation moment is earlier than the obtained first time validity attribute value; or determine that the authorization record to be verified is invalid, when the obtained validation moment is later than the obtained
30 second time validity attribute value.

[0091] As shown in FIG. 6, in a specific implementation of the present application, the previous database authorization record validity verification device can further include a third acquisition unit 250, configured to obtain a validity attribute value of the authorization record to be verified.

[0092] Correspondingly, the verification unit 240 can be further configured to determine that the authorization record to be verified is valid, when the obtained validity attribute value indicates valid and the obtained validation moment is not earlier than the obtained first time validity attribute value; or determine that the authorization record to be verified is invalid, when the obtained validity attribute value indicates valid and the obtained validation moment is not later than the obtained second time validity attribute value.

[0093] It can be seen from the descriptions of the implementations that a person skilled in the art can clearly understand that the present application can be implemented by using software and a necessary general hardware platform. Based on such an understanding, the technical solutions in the present application essentially or the part contributing to the existing technology can be implemented in a form of a software product. The computer software product can be stored in a storage medium, such as a ROM/RAM, a magnetic disk, or an optical disc, and includes several instructions for instructing a computer device (can be a personal computer, a server, or a network device) to perform the methods described in the implementations or in some parts of the implementations of the present application.

[0094] The implementations in the present specification are described in a progressive way. For same or similar parts of the implementations, references can be made to the implementations. Each implementation focuses on a difference from other implementations. Particularly, a device implementation is similar to a method implementation, and therefore is described briefly. For a related part, references can be made to some descriptions in the method implementation. The previously described device implementations are merely examples. The modules described as separate parts can be physically separate or not. During implementation of the present application, functions of the modules can be implemented in one or more pieces of software and/or hardware. Some or all of the modules can be selected based on an actual need to implement the solutions of the implementations. A person of ordinary skill in the art can understand and implement the implementations of the present application without creative efforts.

[0095] The previous descriptions are merely specific implementations of the present application. It is worthwhile to note that a person of ordinary skill in the art can make several improvements or polishing without departing from the principle of the present application. The improvements and polishing shall fall within the

protection scope of the present application.

CLAIMS

1. An authorization token batch-revoking method for a third-party application, wherein the method comprises:

determining a third-party application needing a plurality of authorization tokens to be revoked;

obtaining a current moment, and setting a time validity attribute value for the determined third-party application, so the time validity attribute value equals the current moment;

receiving the plurality of authorization tokens provided by the third-party application;

obtaining a validity attribute value of each of the plurality of authorization tokens;

determining whether the obtained validity attribute value is valid or not;

in response to determining that the obtained validity attribute value is valid, obtaining a validation moment of each of the plurality of the authorization tokens and the time validity attribute value of the third-party application; and

determining that the plurality of the authorization tokens are invalid if the obtained validation moment of each of the plurality of the authorization tokens are earlier than the obtained time validity attribute value, otherwise, determining that the plurality of the authorization tokens are valid.

2. An authority revoking device configured to perform the steps of claim 1.