METHOD AND APPARATUS FOR SETTING MOBILE DEVICE IDENTIFIER

This application claims priority to Chinese Patent Application No. 201610282544.0 filed on April 29, 2016 and entitled "METHOD AND APPARATUS FOR SETTING MOBILE DEVICE IDENTIFIER," the entire contents of which are incorporated herein by reference in the entirety.

Technical Field

This application relates to the field of communications, and in particular, to methods and apparatuses for setting a mobile device identifier.

Background

With the rapid development of the mobile internet, internet applications are increasingly widespread. Therefore, security and protection have become more and more important, especially in the field of e-commerce, where hacker fraud has caused substantial financial losses.

Device fingerprints, typically comprising certain device identifiers for identifying devices, can be used to determine the identity of a specific device, such as a mobile phone. With a device fingerprint, it is possible to correlate a device with a user's behavior. For example, a device fingerprint can be used to look up the transactions conducted on the most recent day for an account logged onto by a given mobile phone, thereby increasing the dimensionality of transaction security analysis based on device fingerprints to enhance the efficiency of security and protection work and the accuracy of counteractions.

Summary

This application presents a method for setting a mobile device identifier, comprising:

obtaining an APP list of a target mobile device;

performing matching between the APP list of the target mobile device and several APP lists stored in a device library, to determine a number of identical APPs in the APP list of the target mobile device and in the APP lists in the device library, wherein corresponding device identifiers are pre-allocated to the several APP lists;

determining whether the determined number of identical APPs reaches a pre-set threshold;

when the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists in the device library reaches the pre-set threshold, setting the device identifier corresponding to the one APP list in the device library as the target mobile device's device identifier.

Optionally, the APP list includes several designated APPs; the designated APPs comprise APPs capable of characterizing APP installation differences among different mobile devices and differentiating different mobile devices.

Optionally, the pre-set threshold is an optimal threshold obtained by performing a statistical analysis based on a number of identical APPs among the APP lists in the device library;

the method comprises:

conducting a paired comparison of APP quantities in the several APP lists to calculate a set reference value, the set reference value comprising a minimum value, maximum value, an average value, or a difference of the APP quantities of the compared APP lists;

partitioning at least one value interval based on the calculated reference value;

conducting a statistical analysis of the number of identical APPs in the APP lists of the APP list pairs included in each partitioned value interval to set the optimal threshold for the each partitioned value interval.

Optionally, the conducting of a statistical analysis of the number of identical APPs in the APP lists of the APP list pairs included in each partitioned value interval to set the optimal threshold for the each partitioned value interval comprises:

sequentially reading target search thresholds from a set optimal threshold search range;

for the each value interval, tallying a number of APP list pairs for which the number of identical APPs in the APP lists is greater than the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are the same, to obtain a first number value;

for the each value interval, tallying a number of APP list pairs for which the number of identical APPs in the APP lists is less than or equal to the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are not the same, to obtain a second number value;

calculating the first number value to second number value ratio to obtain a best candidate threshold, and finding the maximum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range, to serve as the value interval's corresponding optimal threshold; or,

calculating the second number value to first number value ratio to obtain a best candidate threshold, and finding the minimum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range to serve as the

value interval's corresponding optimal threshold.

Optionally, before determining whether the determined number of identical APPs reaches a pre-set threshold, the method further comprises:

tallying the number of APPs in the APP list of the target mobile device;

separately comparing the number of APPs in the APP list of the target mobile device to the number of APPs in each of the APP lists stored in the device library, and calculating the pre-set reference value;

looking up the value interval where the calculated reference value is located;

obtaining the optimal threshold corresponding to the value interval to serve as the pre-set threshold.

Optionally, the method also comprises:

when the number of identical APPs in the APP list of the target mobile device and the APP lists of the device library does not reach the pre-set threshold, allocating a device identifier to the target mobile device, and saving the APP list of the target mobile device and the device identifier allocated to the target mobile device in the device library.

This application also presents an apparatus for setting a mobile device identifier, comprising:

a collection module, configured to obtain an APP list of a target mobile device;

a matching module, performing matching between the APP list of the target mobile device and several APP lists stored in a device library, to determine a number of identical APPs in the APP list of the target mobile device and in the APP lists in the device library; wherein, corresponding device identifiers are pre-allocated to the several APP lists;

a determination module, configured to determine whether the determined number of identical APPs reaches a pre-set threshold;

a setting module, configured to, when the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists in the device library reaches the pre-set threshold, set the device identifier corresponding to the one APP list in the device library as the target mobile device's device identifier.

Optionally, the APP list includes several designated APPs; the designated APPs comprise APPs capable of characterizing APP installation differences among different mobile devices and differentiating different mobile devices.

Optionally, the pre-set threshold is an optimal threshold obtained by performing a statistical analysis based on a number of identical APPs among the APP lists in the device

library;

the apparatus also comprising:

a statistics module, configured to conduct a paired comparison of APP quantities in the several APP lists to calculate a set reference value; partition at least one value interval based on the calculated reference value; and conduct a statistical analysis of the number of identical APPs in the APP lists of the APP list pairs included in each partitioned value interval.

Optionally, the statistics module is configured to:

sequentially read target search thresholds from the set optimal threshold search range;

for each value interval, tally the number of APP list pairs for which the number of identical APPs in the APP lists is greater than the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are the same, to obtain a first number value;

for each value interval, tally the number of APP list pairs for which the number of identical APPs in the APP lists is less than or equal to the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are not the same, to obtain a second number value;

calculate the first number value to second number value ratio to obtain a best candidate threshold, and find the maximum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range, to serve as the value interval's corresponding optimal threshold; or,

calculate the second number value to first number value ratio to obtain a best candidate threshold, and find the minimum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range to serve as the value interval's corresponding optimal threshold.

Optionally, the apparatus also comprises:

a lookup module, configured to tally the number of APPs in the APP list of the target mobile device before the determination module determines whether the determined number of identical APPs reaches a pre-set threshold; separately compare the number of APPs in the APP list of the target mobile device to the number of APPs in each APP list stored in the device library, and calculate the pre-set reference value; look up the value interval where the calculated reference value is located; and obtain the optimal threshold corresponding to the value interval to serve as the pre-set threshold.

Optionally, the setting module is further configured to:

when the number of identical APPs in the APP list of the target mobile device and the APP lists of the device library does not reach the pre-set threshold, allocate a device identifier to the target mobile device, and save the APP list of the target mobile device and the device identifier allocated to the target mobile device in the device library.

In this application, by collecting the APP list of a target mobile device and performing matching between the APP list of the target mobile device and several APP lists stored in a device library, the number of identical APPs in the APP list of the target mobile device and in the APP lists of the device library is determined, and it is determined whether the determined number of identical APPs reaches a pre-set threshold. When the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists of the device library reaches a pre-set threshold, the device identifier corresponding to the one APP list in the device library is set as the target mobile device's device identifier. Thus, it is possible to use differences in the APPs installed on mobile devices, as obtained by matching APP list information, to identify mobile devices and set mobile device identifiers.

Brief Description of the Drawings

FIG. 1 is a flow chart of the method for setting a mobile device identifier, according to an embodiment of this application;

FIG. 2 is a flow chart of using a statistical analysis of the number of identical APPs in the APP lists of a device library to set the aforementioned pre-set threshold, according to an embodiment of this application;

FIG. 3 is a flow chart of using APP list matching to determine an optimal threshold for a target mobile device, according to an embodiment of this application;

FIG. 4 is a logic block diagram of an apparatus for setting a mobile device identifier, according to an embodiment of this application;

FIG. 5 is a hardware structure diagram of a server carrying the apparatus for setting a mobile device identifier, according to an embodiment of this application.

Detailed Description

In related technologies, identifying a mobile device and setting a device identifier for the mobile device is usually done by collecting the attribute values of multiple device attributes for the mobile device. Based on identical attribute sets, matching is conducted in a device library of previously identified mobile devices, to obtain the difference between the attribute values of this mobile device and the mobile devices in the device library. Then matching is conducted between this difference and a pre-set threshold, to determine whether there is a

mobile device in the device library same as this mobile device.

However, in the increasingly complex internet environment, mobile device identification based on device attributes can no longer meet real security and protection demands. For example, in the field of e-commerce, by altering the device attributes of a mobile device, an unknown device is very likely to be identified as a known device, thereby posing a security risk.

This being the case, this disclosure presents a method for setting a mobile device identifier. By collecting the APP list of a target mobile device and performing matching between the APP list of the target mobile device and several APP lists stored in a device library, the number of identical APPs in the APP list of the target mobile device and in the APP lists of the device library is determined, and it is determined whether the determined number of identical APPs reaches a pre-set threshold. When the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists of the device library reaches a pre-set threshold, the device identifier corresponding to the one APP list in the device library is set as the target mobile device's device identifier. Thus, it is possible to use differences in the APPs installed on mobile devices, as obtained by matching APP list information, to identify mobile devices and set mobile device identifiers.

The following is a description of this application, using embodiments in combination with application scenarios.

Please refer to FIG. 1. FIG. 1 is the method for setting a mobile device identifier, according to an embodiment of this application, applied to a server. The method executes the following steps:

Step 101, obtaining an APP list of a target mobile device;

Step 102, performing matching between the APP list of the target mobile device and several APP lists stored in a device library, to determine a number of identical APPs in the APP list of the target mobile device and in the APP lists in the device library, wherein corresponding device identifiers are pre-allocated to the several APP lists;

Step 103, determining whether the determined number of identical APPs reaches a pre-set threshold, wherein the pre-set threshold is an optimal threshold obtained by performing a statistical analysis based on a number of identical APPs among the APP lists in the device library;

Step 104, when the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists in the device library reaches the pre-set threshold, setting the

device identifier corresponding to the one APP list in the device library as the target mobile device's device identifier.

The mobile device may comprise a smart mobile device, e.g., a smartphone.

The server may comprise a server, server cluster, or cloud platform built upon a server cluster.

The APP list may comprise a list made up of information on the APPs installed on a mobile device. In an exemplary application, the server may collect information on the APPs installed on a mobile device, then use this collected information on the APPs installed on the mobile device to create an APP list corresponding to the mobile device.

The number and types of APPs installed on mobile devices will often differ among different mobile devices. Therefore, in this example, the server may identify a mobile device based on differences in the number and types of installed APPs among mobile devices, by conducting matching of the APP lists of mobile devices. Thus, it is possible to allocate unique device identifiers to different mobile devices.

In this example, the server may collect the information on the APPs installed on a mobile device based on data scraping. "Data scraping" may refer to the process of searching for information hidden in large amounts of data based on pre-set algorithms.

Based on a pre-set algorithm, the server may conduct calculations using large amount of data produced by mobile devices as data samples, search in the large amount of data produced by the mobile devices for information on the APPs installed on the mobile device, and then create an APP list for the mobile device based on the searched information on the APPs installed on the mobile device.

No special limitation is made in this example as to the algorithm employed when the server executes data scraping. A person skilled in the art may reference explanations in related technologies when implementing the technical schemes of this application. For example, in an exemplary application, the algorithm may comprise statistical online analysis processing and machine learning algorithms.

In addition, it should be noted that, in this example, the server uses the differences in the number and types of APPs installed on a mobile device to determine the identity of the mobile device and allocate a unique device identifier to the mobile device. Therefore, when the server uses data scraping to collect information on the APPs installed on the mobile device, priority may be given to collect information on those APPs with a high degree of device distinguishability.

Here, APPs with a high degree of device distinguishability will often comprise APPs that can characterize APP installation differences among different mobile devices. That is, for these APPs with a high degree of distinguishability, the probability of them being installed on different mobile devices at the same time is relatively low, and the probability of them being installed at different times is also relatively low. Therefore, by performing matching of these APPs with a high degree of distinguishability installed on different mobile devices, it is possible to accurately reflect differences in the APPs installed on different mobile devices, thereby making it possible to uniquely determine the identity of different mobile devices.

On one hand, for APPs with relatively widespread applications (such as Alipay), there is a very high probability that this APP is installed on different mobile devices, so this APP has a relatively poor degree of distinguishability for devices.

On the other hand, for APPs with rare applications, there is a very high probability that this APP is not installed on mobile devices, so this APP also has a relatively poor degree of distinguishability for devices.

Apps with a relatively poor degree of distinguishability for devices cannot precisely reflect differences in the APPs installed on mobile devices. Therefore, if the server uses APPs with a relatively poor degree of distinguishability to create an APP list, when the server goes on to identify devices based on differences among installed APPS resulting from APP list matching, it is possible that two different devices will be identified as the same device.

Conversely, APPs with a relatively low probability of being installed on different mobile devices at the same time and with a relatively low probability of being installed at different times have a high degree of device distinguishability.

For example, for different types of bank client Apps, the probability that these APPs will be installed on the mobile devices of different users at the same time and at different times is moderate. Also, the bank APP combinations used by different users are not the same. For example, some users use a combination of China Construction Bank APP + China Merchants Bank APP on their mobile devices, and some users use a combination of ICBC bank APP + Bank of Communications on their mobile devices. Therefore, using these APP combinations can do a very good job of differentiating different devices.

In this example, after the server uses data scraping to collect the APP lists of a large number of mobile devices, the server may conduct encoding and storage of the collected APP lists using a unified format.

In an illustrated implementation manner for encoding collected APP lists, the server may

conduct encoding based on whether the APP list includes designated APPs. Here, the designated APPs comprise APPs with a high degree of distinguishability, as described above; the encoding mode may be 0, 1 encoding.

For example, the server may pre-designate several APPs with a high degree of distinguishability, and then traverse a collected APP list, encoding the APP list's APPs with a high degree of distinguishability as 1, and encoding the APP list's APPs not having a high degree of distinguishability as 0. Then, standard storage is conducted according to the format of the encoded APP list.

In another illustrated implementation manner for encoding collected APP lists, the server may also conduct encoding of the APP list based on the serial numbers of all the APPs in the APP list.

For example, the server can conduct uniform numeration of all the APPs that may be installed by a mobile device. Next, the server uses the serial numbers of the APPs to encode the APP list, and then standard storage is conducted according to the format of the encoded APP list. The encoded APP list will include the serial numbers of all the APPs installed by the current mobile device.

Through encoding of collected APP lists according to a unified format, it is possible to quantify the information in the collected APP lists as standard, straightforward numbers, thereby making the later process of using APP list matching to determine differences in the number and types of APPs installed on different mobile devices faster and easier.

Of course, in actual applications, other forms of encoding may be used when encoding collected APP lists. In this example, they are not listed one by one.

In this example, after the server conducts encoding and storage of the collected APP lists, the server may select some of the APP lists from the encoded APP lists to create a device library (e.g., APP lists for which the identity of the corresponding mobile devices are clearly known may be selected to create a device library). At this point, each record in the device library corresponds to a list of APPs installed on the corresponding mobile device.

Here, corresponding device identifiers may be allocated by the server to the APP lists stored in the device library. This device identifier may comprise information identifying the mobile device corresponding to the APP list. For example, the device identifier may comprise information such as the mobile device's IMEI, MAC, or an identification file on the mobile device.

In addition, it should be noted that there is often a certain temporal relationship among

the collection times for the APP lists in the device library (e.g., a sequential order for the collection times of each APP list), and the APPs installed on a mobile device may change with the passing of time. Therefore, in the APP lists stored in the device library, there may be a situation in which the same device identifier is allocated to APP lists with different recorded information.

In this example, the server creates a complete device library, i.e., a database to conduct APP list matching when it is necessary to determine the identity of, and set a device identifier for, a target mobile device. Here, the target mobile device may comprise an unknown device the APP list of which the server can collect and the identity of which cannot be determined.

When determining the identity of the target mobile device, the server can compare the number of identical APPs in the APP list of the target mobile device and in the APP list of the device library to a pre-set threshold. By determining whether this number of identical APPs reaches the pre-set threshold, whether this target mobile device is the same as the device corresponding to the APP list in the device library is determined.

It should be noted that, in this example, the pre-set threshold is an optimal threshold found after conducting a statistical analysis based on the number of identical APPs in the APP lists in the device library.

Please refer to FIG. 2. FIG. 2 is a flow chart of the process of setting a pre-set threshold by conducting a statistical analysis of the number of identical APPs in the APP lists of a device library, illustrated by this example. The process comprises the following steps:

Step 201, conducting a paired comparison of APP quantities in the several APP lists stored in the device library to calculate a set reference value, the set reference value comprising a minimum value, maximum value, an average value, or a difference of the APP quantities of the compared APP lists;

Step 202, partitioning at least one value interval based on the calculated reference value.

In this example, when conducting a paired comparison of the APP quantities in several APP lists stored in a device library, the APP lists stored in the device library may be compared in pairs to obtain several APP list pairs. Afterward, the APP quantities for each APP list pair are compared.

Here, "pairing" means that from all the APP lists stored in the device library, any two APP lists are grouped together. In this way, it is possible to obtain all the possible APP list pairs from all the APP lists stored in the device library.

For example, it is assumed that the device library comprises the three APP lists A1, A2,

and A3. The APP list pairs generated by pairing comprise the three APP list pairs of (A1, A2), (A1, A3), and (A2, A3).

The number of APP list pairs resulting from pairing is usually determined by the number of APP lists stored in the device library. For example, when a device library stores 1000 APP lists, the number of APP list pairs from pairing may be on the order of C_{1000}^{2}.

After several APP list pairs are obtained through pairing, the server may tally the number of APPs in the APP lists of each APP list pair and compare the number of APPs in the APP lists of the APP list pair. After the comparison, the server may extract a pre-set reference value, and then partition at least one value interval based on the extracted reference value.

Here, in an illustrated implementation manner, the pre-set reference value may comprise the minimum value, maximum value, average value, or difference between the APP quantities of the compared APP lists.

In an example of using the minimum value among the APP quantities of the compared APP lists as the reference value, each APP list pair can include the two APP lists of A1 and A2, and the server can tally up N_{A1} and N_{A2} as the numbers of APPs installed in APP lists A1 and A2, respectively. Next, N_{A1} and N_{A2} are compared for minimum value N_{min}. Here, N_{min} is the reference value. Finally, an N_{min} value can be found for every APP list pair, and the server can conduct interval partitioning based on the N_{min} value obtained for each APP list pair, to obtain at least one value interval.

Similarly, when the pre-set reference value is the maximum value, average value, or difference between the APP quantities of the compared APP lists, the same processing may be performed, using the maximum value, average value, or difference (absolute value) between N_{A1} and N_{A2}as the reference value, and conducting numeric interval partitioning based on this reference value.

No special limitation is made in this example as to the strategy of partitioning numeric intervals based on the pre-set reference value. During implementation, the user may manually conduct static partitioning, or the server may conduct partitioning based on a pre-set interval partitioning algorithm.

In addition, no limitation is made in this example as to the number of partitioned numeric intervals. For example, when a user manually conducts static partitioning, the number of data intervals may be set by the user based on their demands; when partitioning is based on a pre-set interval partition algorithm, the number of data intervals is determined by

this algorithm.

After the server partitions at least one value interval based on the calculated reference value, this partitioned value interval will include several APP list pairs.

Still taking the example of using the minimum value among the APP quantities of the compared APP lists as the reference value, it is assumed that the final partitioned value interval is [1,6]. For a random APP list pair, if this APP list pair includes the two APP lists of A1 and A2, the numbers of installed APPs in APP list A1 and A2 are N_{A1} and N_{A2}, and the minimum value of N_{A1} and N_{A2} is N_{min}. If the value of N_{min} falls within the value interval [1,6], this APP list pair belongs to this value interval. That is, value interval [1,6] will include all the APP list pairs for which the calculated N_{min} value falls within interval [1,6].

Step 203, sequentially reading target search thresholds from the set optimal threshold search range;

Step 204, for each value interval, tallying the number of APP list pairs for which the number of identical APPs in the APP lists is greater than the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are the same, to obtain a first number value;

Step 205, for each value interval, tallying the number of APP list pairs for which the number of identical APPs in the APP lists is less than or equal to the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are not the same, to obtain a second number value.

In this example, after at least one value interval is partitioned based on the calculated reference value, the server can conduct a statistical analysis based on the number of identical APPs in the APP list pairs in each partitioned value interval, to set an optimal threshold corresponding to each value interval.

When setting optimal thresholds for each value interval, the server can pre-set an optimal threshold search range for each numeric interval, and the server can sequentially read the target search thresholds from within this optimal threshold search range. For example, it is assumed that the set optimal threshold search range is [1,10]. The server can sequentially read the target search thresholds 1, 2, 3... in this interval until all the integer thresholds in the interval have been read.

In an illustrated implementation manner, the server reads an optimal search threshold from the optimal threshold search range:

on one hand, the server can tally the number of APP list pairs in each value interval for

which the number of identical APPs in the APP lists that is greater than the target search threshold, and for which the device identifiers corresponding to the APP lists are the same, to obtain a first number value;

on the other hand, the server can tally the number of APP list pairs in each value interval for which the number of identical APPs in the APP lists is less than or equal to the target search threshold, and for which the device identifiers corresponding to the APP lists are not the same, to obtain a second number value.

After the server tallies the first number value and second number value, the server can use this first number value to second number value ratio to set an optimal threshold corresponding to each interval.

Step 206, calculating the first number value to second number value ratio to obtain a best candidate threshold, and finding the maximum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range, to serve as the value interval's corresponding optimal threshold; or,

calculating the second number value to first number value ratio to obtain a best candidate threshold, and finding the minimum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range to serve as the value interval's corresponding optimal threshold.

In this example, when the server sets an optimal threshold for each numeric interval based on the tallied first number and second number values:

in one implementation mode, the server can calculate the first number value to second number value ratio, and let this ratio serve as the best candidate threshold based on the read target search threshold and calculated for the current value interval.

The server can repeat the calculation process, calculating the best candidate threshold for each target search threshold read from the optimal threshold search range. Next, it tallies the maximum value from all the best candidate thresholds calculated for all the search thresholds in this optimal threshold search range, to serve as the optimal threshold corresponding to the current value interval.

In another implementation mode, the server can calculate the second number value to first number value ratio, and let this ratio serve as the best candidate threshold based on the read target search threshold, calculated for the current value interval.

The server can repeat the calculation process, calculating the best candidate threshold for each target search threshold read from the optimal threshold search range. Next, the server

can tally the minimum value from all the best candidate thresholds calculated for all the search thresholds in this optimal threshold search range, to serve as the optimal threshold corresponding to the current value interval.

In this example, based on the aforementioned calculation method, the server can set optimal thresholds corresponding to each partitioned value interval. After an optimal threshold corresponding to each value interval has been set, the server can store the correspondence between each value interval and its optimal threshold in the device library in data table form, to facilitate invoking this correspondence in the future.

This marks the end of the process of the server conducting a statistical analysis based on the number of identical APPs in the APP lists in a device library, to set an optimal threshold.

After the server determines the number of identical APPs in the APP list of the target mobile device and in the several APP lists stored in the device library, the server can compare this number of identical APPs to the optimal threshold to determine whether the target mobile device is an existing device in the device library.

Please refer to FIG. 3. FIG. 3 is a flow chart of the process of using APP list matching to determine an optimal threshold for a target mobile device, illustrated by this example. It comprises the following steps:

Step 301, tallying the number of APPs in the APP list of the target mobile device.

In this example, after the server collects the APP list of the target mobile device, first the server can encode the collected target mobile device APP list using the same encoding mode as the APP lists in the device library.

After encoding is completed, based on the information recorded in the encoded APP list, the server can tally the number of APPs installed on this target mobile device.

For example, it is assumed that the APP lists of target mobile devices employ a 0, 1 encoding mode based on whether the APP list includes designated APPs. The designated APPs included in the APP lists are encoded as 1, and APPs other than those designated are encoded as 0. When the server determines the number of APPs in the target mobile device APP list, the server can directly tally the number of APPs in this APP list encoded as 1.

Similarly, the server can also use the same mode to tally the number of APPs in the APP lists of the device library.

Step 302, separately comparing the number of APPs in the APP list of the target mobile device to the number of APPs in each APP list stored in the device library, and calculating the pre-set reference value.

After the server determines the number of APPs in the target mobile device APP list and in the device library APP lists, the server can conduct matching between the APP quantity of the target mobile device APP list and the APP quantities of the APP lists stored in the device library, one by one, and extract the pre-set reference value.

Here, this extracted reference value needs to be analyzed statistically by the server with the number of identical APPs in the APP lists in the device library, to set the optimal threshold, and consistent with the reference value used when partitioning numeric intervals.

For example, it is assumed that the value interval is partitioned by the server based on the minimum value of compared APP list APP numbers. The reference value extracted by the server while conducting matching between the APP quantity of the target mobile device APP list and, one by one, the APP quantities of the APP lists stored in the device library will be the minimum value of the APP quantity of the target mobile device APP list and the APP quantities of the APP lists stored in the device library.

Step 303, looking up the value interval where this calculated reference value is located, and obtaining the optimal threshold corresponding to this value interval to serve as the pre-set threshold.

In this example, after the server conducts matching, one by one, between the APP quantity of the target mobile device APP list and the APP quantities of the APP lists stored in the device library, and extracts the pre-set reference value, the server can look up the value interval in which the extracted reference value is located.

Prior to this, the server has conducted a statistical analysis based on the number of identical APPs in the APP lists in the device library, has set an optimal threshold for each value interval, and has saved the correspondence between each value interval and its optimal threshold. Therefore, after looking up the value interval in which the reference value is located, the server can use this correspondence to look up the optimal threshold corresponding to the value interval in which the reference value is located.

This looked-up optimal threshold is the pre-set threshold used to determine whether the target mobile device is an existing device in the device library.

In this example, after the server determines the optimal threshold corresponding to the target mobile device, the server can continue to conduct matching, one by one, between the target mobile device APP list and the APP lists stored in the device library, to determine the number of identical APPs in the target mobile device APP list and in the APP lists of the device library.

Here, the number of identical APPs is the number of identical APPs installed on the target mobile device and on the device in the device library.

For example, it is assumed that the target mobile device APP list and the APP lists in the device library both employ a 0, 1 encoding mode based on whether the APP list includes designated APPs. The designated APPs included in the APP lists are encoded as 1, and APPs other than those designated are encoded as 0. When the server determines the number of identical APPs in the target mobile device APP list and in a given APP list in the device library, the server can directly tally the number of APPs encoded as 1 in the two matched APP lists.

After the server tallies the number of identical APPs in the target mobile device APP list and in each of the device library's APP lists, the server can compare this number of identical APPs to the determined optimal threshold, and determine whether the determined number of identical APPs reaches this optimal threshold.

When the number of identical APPs in the target mobile device APP list and in a given APP list of the device library reaches this optimal threshold, it means that the target mobile device may be the same device as the mobile device corresponding to this APP list in the device library. Under these circumstances, the server can directly set the device identifier pre-allocated to this APP list in the device library as the device identifier of the target mobile device.

At this point, based on APP list matching, the server has successfully identified the target mobile device and has set a corresponding device identifier for the target mobile device. Afterward, the server can execute security and protection work for the target mobile device based on its device identifier.

For example, in the field of e-commerce, the server can use the device identifier of the target mobile device as the device fingerprint. Based on this device fingerprint, the server can look up all the recent transactions for an account logged onto using this mobile device, and then conduct a transaction security analysis of all the looked-up transactions, to boost the efficiency of security and protection work.

Of course, if the number of identical APPs in the target mobile device APP list and in all the APP lists of the device library fails to reach the optimal threshold, it means that the target mobile device may be different from all the mobile devices corresponding to the APP lists in the device library. At such a time, the target mobile device is not an existing device in the device library. Under these circumstances, the server can allocate a new device identifier,

which is different from all of the APP lists in the device library, for the wireless identification of this target, based on a pre-set identifier allocation strategy. Afterward, the APP list of this target mobile device and the target mobile device's allocated device identifier are saved to the device library, for future queries.

As the preceding embodiments show, this application uses the collection of the APP list of a target mobile device and matching between the APP list of the target mobile device and several APP lists stored in a device library to determine a number of identical APPs in the APP list of the target mobile device and in the APP lists of the device library, and to determine whether the determined number of identical APPs reaches a pre-set threshold. When the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists of the device library reaches a pre-set threshold, the device identifier corresponding to the one APP list in the device library is set as the target mobile device's device identifier. Thus, it is possible to use differences in the APPs installed on mobile devices, as obtained by matching APP list information, to identify a mobile device and set a device identifier for the mobile device. Because the pre-set threshold in this application is an optimal threshold found after conducting a statistical analysis based on the number of identical APPs in the APP lists in the device library, the accuracy of device identification can be improved.

Corresponding to the aforementioned method embodiment, this application also provides an apparatus embodiment.

Please refer to FIG. 4. This application presents an apparatus for setting a mobile device identifier 40, applied to a server. Here, as shown in FIG. 5, the hardware structure of a server carrying the apparatus for setting a mobile device identifier 40 typically comprises a CPU, memory, nonvolatile memory, network interface, and internal bus. In an example of software implementation, the apparatus for setting a mobile device identifier 40 typically can be understood as a computer program loaded into the memory, forming a combined software-hardware logic apparatus after being run on a CPU, the apparatus 40 comprising:

a collection module 401, configured to obtain an APP list of a target mobile device;

a matching module 402, performing matching between the APP list of the target mobile device and several APP lists stored in a device library, to determine a number of identical APPs in the APP list of the target mobile device and in the APP lists in the device library, wherein corresponding device identifiers are pre-allocated to the several APP lists;

determination module 403, configured to determine whether the determined number of

identical APPs reaches a pre-set threshold;

setting module 404, configured to, when the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists in the device library reaches the pre-set threshold, set the device identifier corresponding to the one APP list in the device library as the target mobile device's device identifier.

In this example, the APP list includes several designated APPs; the designated APPs comprise APPs capable of characterizing APP installation differences among different mobile devices and differentiating different mobile devices.

In this example, the pre-set threshold is an optimal threshold obtained by performing a statistical analysis based on a number of identical APPs among the APP lists in the device library; the apparatus 40 also comprising:

a statistics module 405, configured to conduct a paired comparison of APP quantities in the several APP lists to obtain a set reference value; partition at least one value interval based on the calculated reference value; and conduct a statistical analysis of the number of identical APPs in the APP lists of the APP list pairs included in each partitioned value interval.

In this example, the statistics module 405 is configured to:

sequentially read target search thresholds from the set optimal threshold search range;

for each value interval, tally the number of APP list pairs for which the number of identical APPs in the APP lists is greater than the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are the same, to obtain a first number value;

for each value interval, tally the number of APP list pairs for which the number of identical APPs in the APP lists is less than or equal to the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are not the same, to obtain a second number value;

calculate the first number value to second number value ratio to obtain a best candidate threshold, and find the maximum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range, to serve as the value interval's corresponding optimal threshold; or,

calculate the second number value to first number value ratio to obtain a best candidate threshold, and find the minimum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range to serve as the value interval's corresponding optimal threshold.

In this example, the apparatus 40 also comprises:

a lookup module 406, configured to tally the number of APPs in the APP list of the target mobile device before the determination module determines whether the determined number of identical APPs reaches a pre-set threshold; separately compare the number of APPs in the APP list of the target mobile device to the number of APPs in each APP list stored in the device library, and calculate the pre-set reference value; look up the value interval where this calculated reference value is located; and obtain the optimal threshold corresponding to this value interval to serve as the pre-set threshold.

In this example, the setting module 404 is further configured to:

when the number of identical APPs in the APP list of the target mobile device and the APP lists of the device library does not reach the pre-set threshold, allocate a device identifier to the target mobile device, and save the APP list of the target mobile device and the device identifier allocated to the target mobile device in the device library.

In view of the Description and implementing the invention disclosed here, a person skilled in the art can easily conceive of other implementation schemes for this application. This application is designed to cover any modifications, uses, or adaptive changes of this application. These modifications, uses, or adaptive changes follow the general principles of this application and comprise common knowledge or customary technical means in the art, undisclosed by this application. The Description and embodiments are merely illustrative. The true scope and spirit of this application are indicated by the Claims below.

It should be understood that this application is not limited to the exact structures described above and illustrated in the accompanying drawings, and modifications and changes may be made without departing from the scope of this application. The scope of this application is limited solely by the accompanying Claims.

The preceding are merely the preferred embodiments of this application. They do not limit this application. All modifications, equivalent substitutions, and improvements made within the spirit and principles of this application shall fall within the scope of protection of this application.

Claims

1. A method for setting a mobile device identifier, comprising:

obtaining an APP list of a target mobile device;

performing matching between the APP list of the target mobile device and several APP lists stored in a device library, to determine a number of identical APPs in the APP list of the target mobile device and in the APP lists in the device library, wherein corresponding device identifiers are pre-allocated to the several APP lists;

determining whether the determined number of identical APPs reaches a pre-set threshold;

when the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists in the device library reaches the pre-set threshold, setting the device identifier corresponding to the one APP list in the device library as the target mobile device's device identifier.

2. The method according to Claim 1, wherein: the APP list includes several designated APPs; the designated APPs comprise APPs capable of characterizing APP installation differences among different mobile devices and differentiating different mobile devices.

3. The method according to Claim 1, wherein the pre-set threshold is an optimal threshold obtained by performing a statistical analysis based on a number of identical APPs among the APP lists in the device library;

the method comprises:

conducting a paired comparison of APP quantities in the several APP lists to calculate a set reference value, the set reference value comprising a minimum value, maximum value, an average value, or a difference of the APP quantities of the compared APP lists;

partitioning at least one value interval based on the calculated reference value;

conducting a statistical analysis of the number of identical APPs in the APP lists of the APP list pairs included in each partitioned value interval to set the optimal threshold for the each partitioned value interval.

4. The method according to Claim 3, wherein the conducting of a statistical analysis of the number of identical APPs in the APP lists of the APP list pairs included in each

partitioned value interval to set the optimal threshold for the each partitioned value interval comprises:

sequentially reading target search thresholds from a set optimal threshold search range;

for the each value interval, tallying a number of APP list pairs for which the number of identical APPs in the APP lists is greater than the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are the same, to obtain a first number value;

for the each value interval, tallying a number of APP list pairs for which the number of identical APPs in the APP lists is less than or equal to the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are not the same, to obtain a second number value;

calculating the first number value to second number value ratio to obtain a best candidate threshold, and finding the maximum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range, to serve as the value interval's corresponding optimal threshold; or,

calculating the second number value to first number value ratio to obtain a best candidate threshold, and finding the minimum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range to serve as the value interval's corresponding optimal threshold.

5. The method according to Claim 4, wherein before the determining whether the determined number of identical APPs reaches a pre-set threshold, the method further comprises:

tallying the number of APPs in the APP list of the target mobile device;

separately comparing the number of APPs in the APP list of the target mobile device to the number of APPs in each of the APP lists stored in the device library, and calculating the pre-set reference value;

looking up the value interval where the calculated reference value is located;

obtaining the optimal threshold corresponding to the value interval to serve as the pre-set threshold.

6. The method according to Claim 1, further comprising:

when the number of identical APPs in the APP list of the target mobile device and the

APP lists of the device library does not reach the pre-set threshold, allocating a device identifier to the target mobile device, and saving the APP list of the target mobile device and the device identifier allocated to the target mobile device in the device library.

7. An apparatus for setting a mobile device identifier, comprising:

a collection module, configured to obtain an APP list of a target mobile device;

a matching module, performing matching between the APP list of the target mobile device and several APP lists stored in a device library, to determine a number of identical APPs in the APP list of the target mobile device and in the APP lists in the device library, wherein corresponding device identifiers are pre-allocated to the several APP lists;

a determination module, configured to determine whether the determined number of identical APPs reaches a pre-set threshold;

a setting module, configured to, when the number of identical APPs in the APP list of the target mobile device and in any one of the APP lists in the device library reaches the pre-set threshold, set the device identifier corresponding to the one APP list in the device library as the target mobile device's device identifier.

8. The apparatus according to Claim 7, wherein the APP list includes several designated APPs; the designated APPs comprise APPs capable of characterizing APP installation differences among different mobile devices and differentiating different mobile devices.

9. The apparatus according to Claim 7, wherein the pre-set threshold is an optimal threshold obtained by performing a statistical analysis based on a number of identical APPs among the APP lists in the device library;

the apparatus further comprising:

a statistics module, configured to conduct a paired comparison of APP quantities in the several APP lists to calculate a set reference value; partition at least one value interval based on the calculated reference value; and conduct a statistical analysis of the number of identical APPs in the APP lists of the APP list pairs included in each partitioned value interval to set the optimal threshold for the each partitioned value interval.

10. The apparatus according to Claim 9, wherein the statistics module is configured to:

sequentially read target search thresholds from a set optimal threshold search range;

for the each value interval, tally a number of APP list pairs for which the number of identical APPs in the APP lists is greater than the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are the same, to obtain a first number value;

for the each value interval, tally a number of APP list pairs for which the number of identical APPs in the APP lists is less than or equal to the target search threshold, and for which the mobile device identifiers corresponding to the APP lists are not the same, to obtain a second number value;

calculate the first number value to second number value ratio to obtain a best candidate threshold, and find the maximum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range, to serve as the value interval's corresponding optimal threshold; or,

calculate the second number value to first number value ratio to obtain a best candidate threshold, and find the minimum value of the calculated best candidate thresholds for all the search thresholds in the optimal threshold search range to serve as the value interval's corresponding optimal threshold.

11. The apparatus according to Claim 10, wherein the apparatus also comprises:

a lookup module, configured to tally the number of APPs in the APP list of the target mobile device before the determination module determines whether the determined number of identical APPs reaches a pre-set threshold; separately compare the number of APPs in the APP list of the target mobile device to the number of APPs in each of the APP lists stored in the device library, and calculate the pre-set reference value; look up the value interval where the calculated reference value is located; and obtain the optimal threshold corresponding to the value interval to serve as the pre-set threshold.

12. The apparatus according to Claim 7, wherein the setting module is further configured to:

when the number of identical APPs in the APP list of the target mobile device and the APP lists of the device library does not reach the pre-set threshold, allocate a device identifier to the target mobile device, and save the APP list of the target mobile device and the device identifier allocated to the target mobile device in the device library.