

# METHOD AND DEVICE FOR CONCEALING USER INFORMATION CONTAINED IN APPLICATION

## TECHNICAL FIELD

[0001] The present application relates to the field of terminal technologies, and in particular, to a method and a device for concealing user information contained in an application.

## BACKGROUND

[0002] With the rapid development of various wireless mobile devices and the mobile Internet, more and more wireless mobile devices are used in people's daily life, and many wireless mobile devices have become a part of life. Currently, regardless of the iOS, Android, or Windows system, the speed, convenience, stability, and security of data propagation are important indicators for measuring a wireless mobile device. In particular, when it relates to important private user information, the security indicator becomes more important.

[0003] An application used in a wireless mobile device can implement various life and work functions such as shopping, chatting, making friends, and traveling. It is greatly helpful to a user. When using the application, it generates various data, such as a login account and a login password for the application, and a historical chat record, shopping order information, payment information, etc., that relate to sensitive and confidential user information. The user information is usually stored in a local folder of the mobile device in a form of a text file and a database file. Therefore, the text file mentioned above and the database file are easily found, and consequently, the user information is likely to be disclosed. To avoid likely disclosure of user information, sensitive and confidential user information generated in the running application are encrypted before being stored in the local folder of the mobile device. However, although the information has been encrypted, the encrypted information ciphertext string is still completely exposed. For example, a text file and a database file (such as a .txt, .db, or .plist file) can be found by using a relatively common third-party tool (a

database tool such as iTools and Peapod), to obtain the encrypted information. As such, the probability of cracking the encrypted information to obtain sensitive and confidential information is increased, the sensitive and confidential user information is then disclosed, and the security level is relatively low.

5

## SUMMARY

**[0004]** Implementations of the present application provide a method and a device for concealing user information contained in an application, to resolve an existing-technology problem that user information generated in an application running process is easily discovered and obtained and consequently user information is disclosed.

10

**[0005]** An implementation of the present application provides a method for concealing user information contained in an application, and the method includes the following: obtaining operation rights to an attribute list of a media file in an application; reading the attribute list of the media file in the application and writing user information generated in the application into the attribute list of the media file; and re-saving the media file.

15

**[0006]** Further, the user information includes sensitive or confidential information.

**[0007]** Further, obtaining operation rights to an attribute list of a media file in an application includes the following: obtaining a file handle of the media file in the application; and obtaining read/write permission for the attribute list of the media file in the application by using the file handle of the media file.

20

**[0008]** Further, writing user information generated in the application into the attribute list of the media file includes the following: writing the user information generated in the application into the attribute list of the media file in a form of a key-value pair.

25

**[0009]** Further, the media file includes a photo file, an audio file, and a video file.

**[0010]** Further, the user information is encrypted user information.

**[0011]** An implementation of the present application further provides a method for concealing user information contained in an application, and the method includes the following: obtaining operation rights to an attribute list of a media file in an application; performing logic processing on user information generated in the

30

application; reading the attribute list of the media file in the application and writing user information that undergoes logic processing in the application into the attribute list of the media file; and re-saving the media file.

5 [0012] Further, the user information includes sensitive or confidential information.

[0013] Further, obtaining operation rights to an attribute list of a media file in an application includes the following: obtaining a file handle of the media file in the application; and obtaining read/write permission for the attribute list of the media file in the application by using the file handle of the media file.

10 [0014] Further, writing user information generated in the application into the attribute list of the media file includes the following: writing the user information generated in the application into the attribute list of the media file in a form of a key-value pair.

[0015] Further, the media file includes a photo file, an audio file, and a video file.

15 [0016] Further, the user information is encrypted user information.

[0017] Further, performing logic processing on user information generated in the application includes the following: performing reversible logic processing on the user information generated in the application.

20 [0018] Further, performing logic processing on user information generated in the application includes the following: performing symmetric encryption processing on the user information generated in the application by using a symmetric encryption algorithm.

[0019] An implementation of the present application further provides a device for concealing user information contained in an application, and the device includes the following: an acquisition module, configured to obtain operation rights to an attribute list of a media file in an application; a read/write module, configured to read the attribute list of the media file in the application and write user information generated in the application into the attribute list of the media file; and a saving module, configured to re-save the media file.

30 [0020] Further, the device further includes the following: a logic processing module, configured to perform logic processing on the user information generated in the application.

[0021] In comparison with the existing technology, according to the method and the device for concealing user information contained in an application provided in the

implementations of the present application, the user information generated in the application is written into the attribute list of the media file in the application. As such, the user information is concealed and can hardly be found and obtained in a local folder of a mobile device, and the user information generated in the application is  
5 prevented from being disclosed.

## **BRIEF DESCRIPTION OF DRAWINGS**

[0022] The accompanying drawings described here are intended to provide a further understanding of the present application, and constitute a part of the present application. The illustrative implementations of the present application and  
10 descriptions of the implementations are intended to describe the present application, and do not constitute limitations on the present application. In the accompanying drawings:

[0023] FIG. 1 is a schematic flowchart illustrating a method for concealing user information contained in an application, according to Implementation 1 of the present  
15 application;

[0024] FIG. 2 is a schematic flowchart illustrating a method for concealing user information contained in an application, according to Implementation 2 of the present application; and

[0025] FIG. 3 is a schematic structural diagram illustrating a device for  
20 concealing user information contained in an application, according to Implementation 3 of the present application.

## **DESCRIPTION OF EMBODIMENTS**

[0026] To make the objectives, technical solutions, and advantages of the present application clearer, the following describes the technical solutions of the present  
25 application with reference to specific implementations of the present application and corresponding accompanying drawings. Apparently, the described implementations are merely some rather than all of the implementations of the present application. Other implementations obtained by a person of ordinary skill in the art based on the implementations of the present application without creative efforts shall fall within  
30 the protection scope of the present application.

**[0027]** FIG. 1 is a schematic flowchart illustrating a method for concealing user information contained in an application, according to an implementation of the present application. The following describes in detail the method for concealing user information contained in an application provided in the present implementation of the present application with reference to FIG. 1. As shown in FIG. 1, the method for concealing user information contained in an application provided in the present implementation of the present application includes the following steps.

**[0028]** Step S101: Obtain operation rights to an attribute list of a media file in an application.

**[0029]** In the present implementation of the present application, in a running process, an application of a mobile device generates user information corresponding to a user operation. The user information includes sensitive and confidential user information, and the user information is automatically stored in the mobile device. The previous application can be various applications that are installed on the mobile device, for example, an application for payment or for making friends. A payment application is used as an example. When the user uses the payment application, the application generates sensitive and confidential user information such as a login account, a login password, an ID card number, a mobile number, a bank card number, and a payment password. In such user information, the login password and the payment password are confidential user information, and the login account, the ID card number, the mobile number, and the bank card number are sensitive information.

**[0030]** Further, in application running processes, different service types or application scenarios correspond to different user information. Continuing with the example of the payment application, in a payment application scenario, sensitive and confidential user information related to payment is generated. In addition, user information related to chat information is generated in a chat application scenario.

**[0031]** In the present step, the media file in the application can be located in an installation directory of the application, and can be used as a carrier to store the previous user information. Different types of user information can be stored in different media files. Further, the previous user information can be stored in the attribute list of the media file, and the attribute list includes attribute information used to describe the media file. The previous media file can be any one of a picture file, an audio file, and a video file. Preferably, the previous media file can be a picture file. The previous media file can be pre-packaged into an application installation package

as a part of an application icon, and after the application is installed, the previous media file is located in an installation directory of the application.

**[0032]** The user can move a mouse pointer to the media file, click on the right mouse button, click on a media file attribute option, and then click on the detailed information to see an attribute list of the media file, including attribute identifiers and corresponding attribute values. For example, an attribute list of a photo includes attribute identifiers and attribute values corresponding to the attribute identifiers, such as a name, a project type, a creation date, and a modification date.

**[0033]** Further, obtaining operation rights to an attribute list of a media file in an application includes the following: obtaining a file handle of the media file in the application; and obtaining read/write permission for the attribute list of the media file in the application by using the file handle of the media file.

**[0034]** The file handle of the media file can also be referred to as an object pointer of the media file. A file handle is a unique integer value, namely, a 4-byte value (an 8-byte value in a 64-bit program) that is used to identify different objects in the application and different instances of the same type of object. The file handle of the media file is obtained, and the application can use the file handle to read information from or write information into the media file, that is, the attribute list of the media file.

**[0035]** Step S102: Read the attribute list of the media file in the application and write user information generated in the application into the attribute list of the media file.

**[0036]** In the present implementation, because the file handle of the media file is obtained, the attribute list of the media file can be used as an object that can be read or written. The attribute list of the media file includes the attribute information used to describe the media file.

**[0037]** The attribute list of the media file is read into a memory of the mobile device, and the previous user information is written in the memory of the mobile device into the attribute list of the media file.

**[0038]** Writing user information generated in the application into the attribute list of the media file includes the following: writing the user information generated in the application into the attribute list of the media file in a form of a key-value pair.

**[0039]** In an application running process, user information related to a user operation is generated, such as sensitive information and confidential information.

The user information can be an unencrypted plaintext, and the user information can be directly written into the attribute list of the media file.

**[0040]** For example, in an application running process, an account password is generated, and is locally stored. To prevent the user's account password from being stolen, the account password can be written into an attribute list of a media file in a form of a key-value pair to conceal the account password. An attribute identifier Key corresponding to the account password can be named Password. An attribute value Value of the account password can be a string representing a password, and the string can be a plaintext. Certainly, to better conceal and disguise the account password, the attribute identifier Key corresponding to the account password can be set to an attribute identifier Key of a system type that is unlikely to be identified.

**[0041]** In addition, it is worthwhile to note that the user information related to the user operation and generated in the application can alternatively be an encrypted ciphertext. Encryption methods for user information include encryption algorithms such as MD5, DES, RSA, SM2, and SM3.

**[0042]** Further, user information generated in different application scenarios can be written into different media files. For example, user information (such as a payment password and order information) generated in an application payment scenario can be written into a media file, for example, into picture a, and user information generated in a chat application scenario (such as a phone number and a bank card number in chat information) can be written into another media file, for example, into picture b.

**[0043]** Step S103: Re-save the media file.

**[0044]** In the present implementation, after all user information that needs to be concealed is written into the attribute list of the media file, the previous media file is to be re-saved. The user can later move a mouse pointer to the media file, clicks on the right mouse button, clicks on a media file attribute option, and then clicks on the detailed information to see that the user information written into the attribute list has been stored in the attribute list of the media file.

**[0045]** If the media file is successfully saved, the media file is stored in an installation path of the application, and in such case, the user information is successfully disguised.

**[0046]** If the media file fails to be saved, the media file can be re-saved automatically at intervals of one predetermined time period (for example, three to five

seconds), and the predetermined number of re-saving times can be three. If the media file is successfully saved during the three times of re-saving, it indicates that the user information is successfully disguised. If the media file fails to be saved, after being re-saved more than three times, such saving failure is saved into a failure queue.

5   **[0047]**     When user information related to sensitive and confidential information is generated in the application again, after the user information is written into the media file and the media file is successfully saved, the failure queue is automatically invoked, and the media file is re-saved until the media file is saved successfully. If the file still fails to be saved, such task is added to a new failure queue, and the previous  
10   process is repeated until the media file is successfully saved.

**[0048]**     FIG. 2 is a schematic flowchart illustrating a method for concealing user information contained in an application, according to Implementation 2 of the present application. The following describes in detail the method for concealing user information contained in an application provided in Implementation 2 of the present  
15   application with reference to FIG. 2. As shown in FIG. 2, the method for concealing user information contained in an application provided in Implementation 2 of the present application includes the following steps.

**[0049]**     Step S201: Obtain operation rights to an attribute list of a media file in an application.

20   **[0050]**     In the present implementation, the present step is basically the same as step S101 in the method for concealing user information contained in an application provided in Implementation 1 of the present application, and details are omitted here for simplicity.

**[0051]**     Step S202: Perform logic processing on user information generated in the  
25   application.

**[0052]**     In the present implementation, to avoid likely disclosure of the user information (such as sensitive and confidential user information) generated in the application, logic processing needs to be performed on the user information in the application in advance. It is worthwhile to note that the previous logic processing is  
30   reversible logic processing. To be specific, a logic processing process for user information is reversible.

**[0053]**     Performing logic processing on user information generated in the application includes the following: performing symmetric encryption processing on the user information generated in the application by using a symmetric encryption

algorithm.

[0054] Certainly, it is worthwhile to note that, before logic processing, the user information can also be encrypted by using an encryption algorithm such as MD5, DES, RSA, SM2, or SM3.

5 [0055] Step S203: Read the attribute list of the media file in the application and write user information that undergoes logic processing in the application into the attribute list of the media file.

[0056] It is worthwhile to note that, in the present implementation, the user information that undergoes logic processing can also be written into the attribute list  
10 of the media file in a form of a key-value pair to conceal the user information. For example, the user information is an account password, an attribute identifier Key corresponding to the account password can be named Password. An attribute value Value of the account password can be a string or a character string obtained after logic processing.

15 [0057] Step S204: Re-save the media file.

[0058] In the present implementation, after all user information that needs to be concealed is written into the attribute list of the media file, the previous media file is to be re-saved. The present step is basically the same as step S103 in the method for concealing user information contained in an application provided in Implementation 1  
20 of the present application, and details are omitted here for simplicity.

[0059] The previous description provides the method for concealing user information contained in an application in the implementation of the present application. Based on the same idea, Implementation 3 of the present application further provides a device for concealing user information contained in an application.  
25 FIG. 3 shows a structure of the device for concealing user information contained in an application, according to Implementation 3 of the present application. The device includes the following: an acquisition module 301, configured to obtain operation rights to an attribute list of a media file in an application; a logic processing module 302, configured to perform logic processing on user information generated in the application; a read/write module 303, configured to read the attribute list of the media  
30 file in the application and write the user information generated in the application or user information that undergoes logic processing into the attribute list of the media file; and a saving module 304, configured to re-save the media file.

[0060] It is worthwhile to note that the previous attribute list includes attribute

information used to describe the media file. The user information is sensitive and confidential information, and the previous media file includes a photo file, an audio file, and a video file. In addition, the previous user information can alternatively be pre-encrypted user information.

- 5   **[0061]**     Performing logic processing on the user information generated in the application is reversible logic processing, for example, the logic processing can be symmetric encryption.

10   **[0062]**     In conclusion, according to the method and the device for concealing user information contained in an application provided in the implementations of the present application, user information generated in an application is written into an attribute list of a media file in the application. As such, the user information is concealed and can hardly be found and obtained in a local folder of a mobile device, and the user information generated in the application is prevented from being disclosed.

- 15   **[0063]**     In the 1990s, whether a technology improvement is hardware improvement (for example, improvement of a circuit structure, such as a diode, a transistor, or a switch) or software improvement (improvement of a method procedure) can be obviously distinguished. However, as technologies develop, the current improvement for many method procedures can be considered as direct improvement of a hardware circuit structure. A designer usually programs an improved method procedure to a hardware circuit, to obtain a corresponding hardware circuit structure. Therefore, a method procedure can be improved by using a hardware entity module. For example, a programmable logic device (PLD) (for example, a field programmable gate array (FPGA)) is such an integrated circuit, and a logical function of the PLD is determined by a user through device programming. The designer performs programming to "integrate" a digital system to a PLD without requesting a chip manufacturer to design and produce an application-specific integrated circuit chip 2. In addition, the programming is mostly implemented by using "logic compiler" software instead of manually making an integrated circuit chip. It is similar to a software compiler used for program development and compiling. However, original code before compiling is also written in a specific programming language, which is referred to as a hardware description language (HDL). There are many HDLs, such as an Advanced Boolean Expression Language (ABEL), an Altera Hardware Description Language (AHDL), Confluence, a Cornell University Programming Language (CUPL), HDcAl, a Java

Hardware Description Language (JHDL), Lava, Lola, MyHDL, PALASM, and a Ruby Hardware Description Language (RHDL). Currently, a Very-High-Speed Integrated Circuit Hardware Description Language (VHDL) and Verilog2 are most commonly used. A person skilled in the art should also understand that a hardware circuit that implements a logical method procedure can be readily obtained once the method procedure is logically programmed by using the several described hardware description languages and is programmed into an integrated circuit.

**[0064]** A controller can be implemented in any appropriate way. For example, the controller can be a microprocessor, a processor, or a computer readable medium, a logic gate, a switch, an application-specific integrated circuit (ASIC), a programmable logic controller, or an embedded microcontroller that stores computer readable program code (for example, software or firmware) that can be executed by the processor (or the microprocessor). Examples of the controller include but are not limited to the following microcontrollers: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, or Silicon Labs C8051F320. A memory controller can also be implemented as a part of control logic of the memory. A person skilled in the art also knows that a controller can be implemented using pure computer readable program code, and the steps in the method can be logically programmed to enable the controller to implement same functions in forms of a logical gate, a switch, an application-specific integrated circuit, a programmable logic controller, an embedded microcontroller, etc. Therefore, the controller can be considered as a hardware component, and a device configured to implement various functions in the controller can also be considered as a structure in a hardware component. Alternatively, a device configured to implement various functions can be considered as a software module that can implement the method or a structure in a hardware component.

**[0065]** The system, device, module, or unit illustrated in the previous implementations can be implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function.

**[0066]** For ease of description, the described device is described by dividing functions into various units. Certainly, when the present application is implemented, the functions of each unit can be implemented in one or more pieces of software and/or hardware.

**[0067]** A person skilled in the art should understand that the implementations of the present disclosure can be provided as a method, a system, or a computer program

product. Therefore, the present disclosure can use a form of hardware only implementations, software only implementations, or implementations with a combination of software and hardware. In addition, the present disclosure can use a form of a computer program product implemented on one or more computer-usable storage media (including but not limited to a magnetic disk memory, a CD-ROM, and an optical memory) that include computer-usable program code.

**[0068]** The present disclosure is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product based on the implementations of the present disclosure. It should be understood that computer program instructions can be used to implement each procedure and/or each block in the flowcharts and/or the block diagrams and a combination of a procedure and/or a block in the flowcharts and/or the block diagrams. These computer program instructions can be provided for a general-purpose computer, a dedicated computer, an embedded processor, or a processor of another programmable data processing device to generate a machine. As such, the instructions are executed by the computer or the processor of another programmable data processing device to generate a device for implementing a specified function in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

**[0069]** These computer program instructions can be stored in a computer readable memory that can instruct the computer or another programmable data processing device to work in a particular way. As such, the instructions stored in the computer readable memory generate an artifact that includes an instruction device. The instruction device implements a particular function in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

**[0070]** These computer program instructions can be loaded to a computer or another programmable data processing device so that a series of operations and steps are performed on the computer or another programmable device to generate computer-implemented processing. Therefore, the instructions executed on the computer or another programmable device provide steps for implementing a specified function in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

**[0071]** In a typical configuration, a computing device includes one or more processors (CPU), an input/output interface, a network interface, and a memory.

**[0072]** The memory can include a non-persistent memory, a random access

memory (RAM), and/or a nonvolatile memory, etc. in a computer readable medium, such as a read-only memory (ROM) or a flash memory (flash RAM). The memory is an example of the computer readable medium.

**[0073]** The computer readable medium includes persistent, non-persistent,  
5 movable, and unmovable media that can store information by using any method or technology. The information can be a computer readable instruction, a data structure, a program module, or other data. Examples of a computer storage medium include but are not limited to a phase change random access memory (PRAM), a static RAM (SRAM), a dynamic RAM (DRAM), a RAM of another type, a read-only memory  
10 (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory technology, a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD) or another optical storage, a magnetic tape, a magnetic disk storage, another magnetic storage device, or any other non-transmission medium. The computer storage medium can be used to store  
15 information that can be accessed by a computing device. As described in the present specification, the computer readable medium does not include transitory media, for example, a modulated data signal and a carrier.

**[0074]** It is worthwhile to further note that, the term "include", "contain", or any other variant is intended to cover non-exclusive inclusion so that a process, a method,  
20 a product, or a device that includes a series of elements not only includes these elements, but also includes other elements that are not expressly listed, or further includes elements inherent to such a process, method, product, or device. An element preceded by "includes a ..." does not, without more constraints, exclude the existence of additional identical elements in the process, method, product, or device that  
25 includes the element.

**[0075]** A person skilled in the art should understand that the implementations of the present application can be provided as a method, a system, or a computer program product. Therefore, the present application can use a form of hardware only implementations, software only implementations, or implementations with a  
30 combination of software and hardware. In addition, the present application can use a form of a computer program product implemented on one or more computer-usable storage media (including but not limited to a magnetic disk memory, a CD-ROM, and an optical memory) that include computer-usable program code.

**[0076]** The present application can be described in common contexts of computer

executable instructions executed by a computer, such as a program module. Generally, the program module includes a routine, a program, an object, a component, a data structure, etc. executing a specific task or implementing a specific abstract data type. The present application can also be practiced in distributed computing environments.

5 In these distributed computing environments, tasks are executed by remote processing devices that are connected by using a communications network. In the distributed computing environments, the program module can be located in local and remote computer storage media that include storage devices.

**[0077]** The implementations in the present specification are described in a progressive way. For same or similar parts in the implementations, references can be made to the implementations. Each implementation focuses on a difference from other implementations. Particularly, a system implementation is similar to a method implementation, and therefore, is described briefly. For related parts, references can be made to related descriptions in the method implementation.

15 **[0078]** The previous descriptions are merely implementations of the present application, and are not intended to limit the present application. A person skilled in the art can make various modifications and changes to the present application. Any modifications, equivalent replacements, improvements, etc. made within the spirit and principle of the present application shall fall within the protection scope of the claims  
20 of the present application.

## CLAIMS

1. A computer-implemented method, comprising:
  - obtaining operation rights to an attribute list of a media file in a software application, including obtaining a file handle of the media file and obtaining read or write permission for the attribute list of the media file;
  - identifying the attribute list of the media file, the attribute list including attribute information that describes the media file;
  - writing user information that has been generated during execution of the software application into the attribute list of the media file in a key-value pair format; and
  - storing the media file in an installation path of the software application.
2. The computer-implemented method of claim 1, wherein the user information comprises sensitive and confidential information, wherein the user information is encrypted.
3. The computer-implemented method of claim 1, wherein the media file comprises a photo file, an audio file, or a video file.
4. The computer-implemented method of claim 1, further comprising writing user information that undergoes logic processing by the software application into the attribute list.
5. The computer-implemented method of claim 4, wherein performing logic processing on user information generated by the software application comprises:
  - performing reversible logic processing on the user information; and
  - performing symmetric encryption processing on the user information by using a symmetric encryption algorithm.
6. A non-transitory, computer-readable medium storing one or more instructions executable by a computer system to perform operations comprising:

obtaining operation rights to an attribute list of a media file in a software application, including obtaining a file handle of the media file and obtaining read or write permission for the attribute list of the media file;

identifying the attribute list of the media file, the attribute list including attribute information that describes the media file;

writing user information that has been generated during execution of the software application into the attribute list of the media file in a key-value pair format; and

storing the media file in an installation path of the software application.

7. The non-transitory, computer-readable medium of claim 6, wherein the user information comprises sensitive and confidential information, wherein the user information is encrypted.

8. The non-transitory, computer-readable medium of claim 6, wherein the media file comprises a photo file, an audio file, or a video file.

9. The non-transitory, computer-readable medium of claim 6, further comprising one or more instructions to write user information that undergoes logic processing by the software application into the attribute list.

10. The non-transitory, computer-readable medium of claim 9, wherein performing logic processing on user information generated by the software application comprises one or more instructions to:

perform reversible logic processing on the user information; and

perform symmetric encryption processing on the user information by using a symmetric encryption algorithm.

11. A computer-implemented system, comprising:  
one or more computers; and

one or more computer memory devices interoperably coupled with the one or more computers and having tangible, non-transitory, machine-readable media storing one or more instructions that, when executed by the one or more computers, perform one or more operations comprising:

- obtaining operation rights to an attribute list of a media file in a software application, including obtaining a file handle of the media file and obtaining read or write permission for the attribute list of the media file;

- identifying the attribute list of the media file, the attribute list including attribute information that describes the media file;

- writing user information that has been generated during execution of the software application into the attribute list of the media file in a key-value pair format; and

- storing the media file in an installation path of the software application.

12. The computer-implemented system of claim 11, wherein the user information comprises sensitive and confidential information, wherein the user information is encrypted.

13. The computer-implemented system of claim 11, wherein the media file comprises a photo file, an audio file, or a video file.

14. The computer-implemented system of claim 11, further comprising writing user information that undergoes logic processing by the software application into the attribute list, wherein performing logic processing on user information generated by the software application comprises one or more operations to:

- perform reversible logic processing on the user information; and

- perform symmetric encryption processing on the user information by using a symmetric encryption algorithm.