(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 549 304 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
23.09.2020 Bulletin 2020/39

(21) Application number: **17837851.9**

(22) Date of filing: **01.12.2017**

(51) Int Cl.:
H04L 9/08 $^{(2006.01)}$ ***G06F 21/62*** $^{(2013.01)}$
H04W 12/04 $^{(2009.01)}$

(86) International application number:
PCT/EP2017/025349

(87) International publication number:
WO 2018/099606 (07.06.2018 Gazette 2018/23)

(54) PROTECTING USAGE OF KEY STORE CONTENT

SCHUTZ DER VERWENDUNG VON SCHLÜSSELSPEICHERUNGSINHALT

PROTECTION DE L'UTILISATION D'UN CONTENU DE MAGASIN DE CLÉS

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(30) Priority: **02.12.2016 GB 201620553**

(43) Date of publication of application:
09.10.2019 Bulletin 2019/41

(73) Proprietor: **Gurulogic Microsystems OY**
20100 Turku (FI)

(72) Inventors:
- **KÄRKKÄINEN, Tuomas**
  FI-20320 Turku (FI)
- **KALEVO, Ossi**
  FI-37800 Akaa (FI)
- **SAHLBOM, Mikko**
  FI-25500 Perniö (FI)

(74) Representative: **Norris, Timothy Sweyn**
Basck
50 - 60 Station Road
Cambridge CB1 2JH (GB)

(56) References cited:
US-A1- 2014 208 100

- **Woo Commerce: "How to Import Serial Keys from CSV file - StoreApps", , 21 April 2016 (2016-04-21), XP055461145, https://www.storeapps.org/?s=serial+key&post_type=kbe_knowledgebase Retrieved from the Internet: URL:https://www.storeapps.org/docs/wcsk-how-to-import-serial-keys-from-csv-file/ [retrieved on 2018-03-20]**
- **SABT MOHAMED ET AL: "Breaking into the KeyStore: A Practical Forgery Attack Against Android KeyStore", 15 September 2016 (2016-09-15), ECCV 2016 CONFERENCE; [LECTURE NOTES IN COMPUTER SCIENCE; LECT.NOTES COMPUTER], SPRINGER INTERNATIONAL PUBLISHING, CHAM, PAGE(S) 531 - 548, XP047356195, ISSN: 0302-9743 ISBN: 978-3-642-33485-6 [retrieved on 2016-09-15] paragraph [0004]**
- **TIM COOIJMANS ET AL: "Analysis of Secure Key Storage Solutions on Android", SECURITY AND PRIVACY IN SMARTPHONES & MOBILE DEVICES, ACM, 2 PENN PLAZA, SUITE 701 NEW YORK NY 10121-0701 USA, 7 November 2014 (2014-11-07), pages 11-20, XP058061411, DOI: 10.1145/2666620.2666627 ISBN: 978-1-4503-3155-5**

Printed by Jouve, 75001 PARIS (FR)

Description

TECHNICAL FIELD

[0001] The present disclosure relates to systems for protecting usage of key store content at user devices of end users, for example, to data security systems that are reliant upon using key materials to achieve data security. Moreover, the present disclosure also relates to methods of protecting usage of key store content at user devices of end users. Furthermore, the present disclosure also relates to computer program products comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute the aforementioned methods.

BACKGROUND

[0002] There often arises a need to store user-sensitive data on user devices, because there are presently available various services and functionalities that are designed to run as software applications on user devices, for example software applications for making payments. As a first example, there are presently available multiple applications for banking and payment services, wherein the multiple applications require secure arrangements for maintaining strong protection for customers when using the banking and payment services in order to try to avoid malicious third parties from hacking into such services for stealing money. As a second example, a user may need to store secret or private keys to access protected e-mails. For these and many other reasons, it is very desirable to provide a robust solution for handling key store pertaining to key materials.

[0003] There are many security service providers that are accessible via multiple "*eco-system*" platforms, whose key stores are based on software. For example, the Android™ eco-system platform is contemporarily much in public focus, because there are numerous examples of mobile devices worldwide, for example tablet-like computing devices, that utilize the Android™ eco-system platform. Referring to Google documents, the Android™ Keystore system stores cryptographic keys in a container to make it more difficult to extract from a given Android-compatible device. The Android™ Keystore system is most advanced amongst contemporarily available key store systems in security matters, but still unfortunately lacks very important functionalities needed to provide an efficient security solution in contemporary devices that are sold in large numbers in the market. The Android™ Keystore offers a substantially complete set of security algorithms, for example, such as crypto, key generator, key factory, key pair generator, mac, and signature. All of these services run inside hardware that is backed by the key store system to make it efficient and convenient to use, but there is not taken into account real world cryptographic requirements.

[0004] Moreover, in only a few years, there have recently been a huge increase in the use of mobile phones, and there are multiple vendors with different sets of device models, which are powered by different eco-systems, for example, such as Google® Android™, Apple® iOS™, Microsoft® Windows®, and so on. There arises an issue from a security perspective in that every eco-system has its own security solutions to protect user-sensitive data in hardware-based or software-based key stores. This makes it very difficult for application developers to understand security related implementations even theoretically, although it is understood at a basic level. Almost every eco-system has its own key store solution, but from a point of current urgent need, it is desired to focus on mobile platforms, because almost every human will soon have some kind of smartphone and a great amount of different applications that require properly implemented key stores for holding user-sensitive key materials inside. On paper, key stores almost fulfill any known security issues, but in reality, their software implementations do not meet the solid solutions with the System on Chip (SoC) hardware design.

[0005] Firstly, the Android™ Keystore is not designed to import thousands or millions of secret keys (namely, key materials), but has been designed to maintain only a few secret keys. The Android™ Keystore has been designed for a scenario where the same key(s) are used for encryption purposes again and again. Secondly, the Android™ Keystore supports importing only one plain raw key at a time, which is potentially exposed to malicious parties. This is because the security of the Android™ Keystore is based on asymmetric encryption, which is a very slow computation process.

[0006] Moreover, there are some conventionally-known software-based key store solutions, from which *"Bouncy Castle"* alias "*BC*" is the most known provider. When compared to the hardware-based Android™ Keystore, BC supports key store import functionality for protected key materials in Abstract Syntax Notation One (ASN.1) format, which can import securely more than one key at a time into the key store. The main problem with BC is that, the key material is not completely protected against extraction prevention, because the key material is requested from outside of the key store and is provided to another software application that then uses the key material. This makes it possible for a malicious third party to retrieve a key material from the key store in connection with an authenticated request. In particular, the key materials are not indexed in a given BC's key store, which potentially forces revealing the sensitive key materials to malicious parties.

[0007] In a published United States patent document US 2014/0208100 A1 (H. Richard Kendall; "*Provisioning an App* *on a Device and Implementing a Keystore*"), there is described a method of installing a keystore in a mobile app. The app prompts a user to enter a passphrase to create an app keystore. The Keystore has a user section and a Table of Contents (TOC). The files in the Keystore are hashed, creating "*first*" key store hash values. The first keystore hash values are stored in the TOC. The TOC is hashed to create a TOC hash value. The passphrase entered by the user is then combined with the TOC hash value. This creates a "*first*" master passphrase for the keystore. The keystore is then transmitted to the device where it is installed in generic (non-provisioned) wrapped app.

[0008] In a published technical document (Woo Commerce: "How to Import Serial Keys from CSV file - StoreApps", 21 April2016 (2016-04-21), XP055461145), a technical overview of an import of serial keys from a CSV file is described.

[0009] The method includes creating of a CSV file consisting of serial key text to be distributed among customers. The created CSV file is then imported to WooCommerce Serial Key dashboard. Once imported, the CSV file is validated and information of how many serial keys have been imported and ones that are skipped is shown in a table.

[0010] In another published technical document (Mohamed Sabt et al; "Breaking into the KeyStore: A Practical Forgery Attack Against Android KeyStore"), a technical overview of security of an Android KeyStore is described. The document proves that use of non-provably secure cryptographic schemes in complex architectures could cause severe consequences. The document further proves the Authenticated Encryption (AE) scheme Hash-then-CBC-Encrypt does not provide authenticity regardless of the used hash function. Document also presents a selective forgery attack where an adversary exploits this weakness to substantially reduce the length of the symmetric keys protected by the KeyStore.

SUMMARY

[0011] The present disclosure seeks to provide an improved system for protecting usage of key store content at a given user device of an end user.

[0012] Moreover, the present disclosure seeks to provide an improved method of protecting usage of key store content at a given user device of an end user.

[0013] A further aim of the present disclosure is to at least partially overcome at least some of the problems of the prior art, as discussed above.

[0014] In a first aspect, embodiments of the present disclosure provide a method of protecting usage of key store content at a given user device of an end user, characterized in that the method includes steps of:

(i) receiving, at the given user device, the key store content including key materials that are encrypted using encryption credentials compatible with the given user device, the key store content being created by and received from a key service provider in a format that is compatible with the given user device, wherein the key materials are provided as a key code list;

(ii) importing the encrypted key materials of the key store content to a protected key store of the given user device and storing the key materials at the protected key store in the encrypted form, wherein all the encrypted key materials of the key store content are imported at one go, wherein the key store content is stored at the key store in a manner that the key materials are non-exportable from the key store, and wherein the importing comprises:

- decrypting the encrypted key materials to obtain the key code list, and
- generating keys by choosing the keys from the key code list based upon key offsets, byte offsets and/or bit offsets; and

(iii) internally within the protected key store of the given user device, allowing one or more key store integrated services of the given user device to access the non-exportable key materials for use via key references only.

[0015] Embodiments of the present disclosure are of advantage in that complete protection of the key store content against unauthorized access is facilitated by employing a complete end-to-end process from the key service provider to the given user device of the end user, wherein the key materials of the key store content are never exposed or treated unsafely at any step in the process, are non-exportable once stored at the protected key store of the given user device, and are accessible for use by the services that are integrated with the protected key store, via the key references only.

[0016] In a second aspect, embodiments of the present disclosure provide a computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute the method pursuant to the aforementioned first aspect.

[0017] In a third aspect, embodiments of the present disclosure provide a system for protecting usage of key store content at a given user device of an end user, characterized in that the system is operable to:

(i) receive, at the given user device (106), the key store content (102) including key materials that are encrypted using encryption credentials compatible with the given user device (106), the key store content (102) being created by and received from a key service provider (104) in a format that is compatible with the given user device (106), wherein the key materials are provided as a key code list;

(ii) import the encrypted key materials of the key store content (102) to a protected key store (108) of the given user device (106) and store the key materials at the protected key store (108) in the encrypted form, wherein all the key encrypted materials of the key store content (102) are imported at one go, wherein the key store content (102) is stored at the key store in a manner that the key materials are non-exportable from the key store, and wherein the importing com prises:

- decrypting the encrypted key materials to obtain the key code list, and
- generating keys by choosing the keys from the key code list based upon key offsets, byte offsets and/or bit offsets; and

(iii) internally within the protected key store (108) of the given user device (106), allow one or more key store integrated services of the given user device (106) to access the non-exportable key materials for use via key references only.

[0018] Additional aspects, advantages, features and objects of the present disclosure would be made apparent from the drawings and the detailed description of the illustrative embodiments construed in conjunction with the appended claims that follow.

[0019] It will be appreciated that features of the present disclosure are susceptible to being combined in various combinations without departing from the scope of the present disclosure as defined by the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] The summary above, as well as the following detailed description of illustrative embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the present disclosure, exemplary constructions of the disclosure are shown in the drawings. However, the present disclosure is not limited to specific methods and apparatus disclosed herein. Moreover, those in the art will understand that the drawings are not to scale. Wherever possible, like elements have been indicated by identical numbers.

[0021] Embodiments of the present disclosure will now be described, by way of example only, with reference to the following diagrams wherein:

FIG. 1A is a schematic illustration of a system for protecting usage of key store content at a given user device of an end user, in accordance with an embodiment of the present disclosure;

FIG. 1B is a schematic illustration of a complete end-to-end process flow of protecting the usage of the key store content at the given user device, in accordance with an embodiment of the present disclosure;

FIG. 1C is a schematic illustration of how the key store content is imported and loaded to a protected key store of the given user device, in accordance with an embodiment of the present disclosure; and

FIG. 2 is a flow chart depicting steps of a method of protecting usage of key store content at a given user device of an end user, in accordance with an embodiment of the present disclosure.

[0022] In the accompanying drawings, an underlined number is employed to represent an item over which the underlined number is positioned or an item to which the underlined number is adjacent. When a number is non-underlined and accompanied by an associated arrow, the non-underlined number is used to identify a general item at which the arrow is pointing.

DETAILED DESCRIPTION OF EMBODIMENTS

[0023] The following detailed description illustrates embodiments of the present disclosure and ways in which they can be implemented. Although some modes of carrying out the present disclosure have been disclosed, those skilled in the art would recognize that other embodiments for carrying out or practising the present disclosure are also possible.

[0024] In a first aspect, embodiments of the present disclosure provide a method of protecting usage of key store content at a given user device of an end user, characterized in that the method includes steps of:

(i) receiving, at the given user device, the key store content including key materials that are encrypted using encryption credentials compatible with the given user device, the key store content being created by and received from a key service provider in a format that is compatible with the given user device, wherein the key materials are provided as a key code list;

(ii) importing the encrypted key materials of the key store content to a protected key store of the given user device and storing the key materials at the protected key store in the encrypted form, wherein all the encrypted key materials of the key store content are imported at one go, wherein the key store content is stored at the key store in a manner that the key materials are non-exportable from the key store, and wherein the importing comprises:

- decrypting the encrypted key materials to obtain the key code list, and
- generating keys by choosing the keys from the key code list based upon key offsets, byte offsets and/or bit offsets; and

(iii) internally within the protected key store of the given user device, allowing one or more key store integrated services of the given user device to access the non-exportable key materials for use via key references only.

[0025] Optionally, in addition to the key materials, the protected key store also includes information about the end user and/or a group of end users that are authorized to use the key materials stored at the protected key store. Additionally or alternatively, optionally, the protected key store includes other information relevant for usage of the key materials.

[0026] Throughout the present disclosure, the term "*end user*" encompasses a human user as well as a machine. As an example, the end user could be a registered relay machine. This is particularly beneficial for cases where the aforementioned method is used to recognize and verify servers that perform machine-to-machine communication.

[0027] It will be appreciated that the key store is protected only for its usage by the end user, namely against unauthorized usage by anyone other than the authorized end user, and such protection of the key store is not related to the encryption of the imported key materials. In other words, the key materials are encrypted using one or more different secret keys, for example, such as one or more pre-shared keys.

[0028] Optionally, the key materials are received at the step (i) in a symmetrically-encrypted form. Optionally, in this regard, the key materials are encrypted using symmetric keys.

[0029] Optionally, the method includes, internally within the protected key store of the given user device, decrypting one or more of the key materials to be used by the one or more key store integrated services of the given user device.

[0030] Throughout the present disclosure, the term "*key reference*" generally refers to a given reference that refers to and identifies a given key material stored at the protected key store. In other words, with a given key reference, it is known which key material (for example, a key or a certificate) is to be used, and optionally, in which location of the protected key store the key material to be used is located. Pursuant to embodiments of the present disclosure, the key material itself is never extracted from the key store.

[0031] According to an embodiment, the key references are implemented by way of indices of the key materials. Optionally, the indices are ordinal numbers of the key materials in an order of their occurrence. Optionally, the method includes receiving, within the key store content, the indices to be used for referencing the key materials via the key references. Alternatively, optionally, the method includes generating, at the given user device, the indices to be used for referencing the key materials via the key references. As an example, the indices may be generated in a consecutive manner corresponding to an order in which the key materials are included in the key store content. The indices can be generated, for example, at an initial registration of the given user device to the key service provider or during the decryption of the key materials.

[0032] According to another embodiment, the key references are implemented by way of offsets based upon which the key materials are to be identified. It will be appreciated that a given offset refers to and identifies a given key material stored at the protected key store. Some examples of the offsets have been provided later for illustration purposes.

[0033] Pursuant to embodiments of the present disclosure, complete protection of the key store content against unauthorized access is facilitated by employing a complete end-to-end process from the key service provider to the given user device of the end user, wherein the key materials of the key store content are never exposed or treated unsafely at any step in the process. The key store content is created and delivered to the given user device in the encrypted form. This potentially prevents eavesdropping by third parties. At the given user device, the key store content is imported to the key store of the given user device at one go.

[0034] The key store content is received as a single list containing all the key materials (hereafter referred to as the "key code list", for the sake of convenience only). Alternatively, optionally, the key store content is received as a plurality of key code lists. It will be appreciated that the plurality of key code lists can be imported simultaneously. Irrespective of the form in which different key code lists, namely the key materials, are encrypted, all the key materials are imported at one go.

[0035] Optionally, at the step (ii), the key store content is imported to the given user device as a single data file. It will be appreciated that the number of key materials included within the key store content can be as large as thousands, potentially millions. In such a case, importing the key store content as a single data file has several advantages, as compared to conventional key store techniques.

[0036] For illustration purposes only, there will now be considered an example implementation of the aforementioned system with Gurulogic Microsystem Oy's proprietary product "*Starwindow®*". In such an implementation, once the key store content is received at the given user device, a "*Load*" function provided by a key store of the "*Starwindow®*" product is used to import all the key materials to the key store at one go. Optionally, the "*Load*" function also decrypts the key materials securely within the key store to enable fast usage thereof. It will be appreciated that the "*Load*" function can be used to import all the key materials at one go, even when there are millions of key materials included in the key store content.

[0037] The aforementioned importing of a huge amount of key materials simultaneously is enabled by different implementation solutions, for example, such as the following options:

Option A:

[0038] A key code list comprises eight different 128-bit keys; this key code list consumes 128 bytes of a storage space. As the smallest unit size is one byte, which represents eight bits, a 128-bit key consumes 16 bytes of the storage space. It will be appreciated that the keys typically represent highest-possible entropy, thereby strengthening the protection achieved therefrom, and therefore these keys cannot be compressed with traditional compression techniques.

[0039] An example key code list may be represented as follows:

```
      KeyCodeListA : array [0..7] of array [0..15] of UInt8 =
(0x4B, 0xDA, 0x72, 0x44, 0xB3, 0x12, 0x07, 0x43, 0x6F, 0x65, 0x83,
0x8C, 0xF5, 0x3F, 0xF1, 0x08, //key1
0xCA, 0x1E, 0x7F, 0xDF, 0x5C, 0x7F, 0x78, 0x0C, 0x55, 0x88, 0x96, 0x0B,
0xA9, 0xD9, 0x22, 0x6F, //key2
0xB6, 0x43, 0x73, 0x84, 0x57, 0x86, 0x66, 0xF8, 0x79, 0xB0, 0xCC, 0xA0,
0x16, 0x13, 0x42, 0xDF, //key3
0xF0, 0x6B, 0x2B, 0xF8, 0x68, 0x5A, 0x31, 0xCF, 0x9A, 0x65, 0xF1, 0xC7,
0x94, 0x62, 0xDD, 0x9B, //key4
0xB1, 0x28, 0x68, 0xEE, 0x1B, 0x4D, 0x43, 0x07, 0xE4, 0x97, 0xFF, 0x00,
0x01, 0xFF, 0x00, 0xE0, //key5
0xEE, 0x1F, 0xFD, 0xA9, 0x69, 0xE5, 0xFF, 0x00, 0xDF, 0x67, 0x67, 0xF7
0xB0, 0xB9, 0xAA, 0x77, //key6
0x9E, 0x55, 0xAC, 0xE3, 0xFE, 0x16, 0x27, 0xD9, 0xED, 0xE1, 0x2B, 0xFF
0x00, 0x13, 0xFF, 0x00, //key7
0xB5, 0xE2, 0x28, 0x56, 0x2D, 0xBF, 0xE9, 0x39, 0x1F, 0xF0, 0x74, 0x9F,
0x95, 0x19, 0x05, 0x07, //key8);
```

, wherein consecutive sequences of 16 bytes each represent the keys. In this example, the keys are generated based upon key offsets, namely by increasing the offsets by the size of the keys (which is 16 bytes in this exam ple).

[0040] Optionally, the importing at one go is made by "compressing" the keys to be delivered. There are at least two different ways, namely options 1) "*B*" and "*C*", and 2) "*D*" to implement this:

Option B:

[0041] For illustration purposes only, the option "*B*" will now be described with respect to the same example key code list. Optionally, 128 keys (= 8 x 16) are generated from the same key code list, by choosing the keys based upon byte offsets instead of key offsets. As an example, first three keys can be generated as follows:

```
      KeyCodeListB : array [0..127] of Ulnt8 =
(0x4B, 0xDA, 0x72, 0x44, 0xB3, 0x12, 0x07, 0x43, 0x6F, 0x65, 0x83,
0x8C, 0xF5, 0x3F, 0xF1, 0x08, // key1 from a byte offset "0"
(0xDA, 0x72, 0x44, 0xB3, 0x12, 0x07, 0x43, 0x6F, 0x65, 0x83, 0x8C,
0xF5, 0x3F, 0xF1, 0x08, 0xCA // key2 from a byte offset "1"
(0x72, 0x44, 0xB3, 0x12, 0x07, 0x43, 0x6F, 0x65, 0x83, 0x8C, 0xF5, 0x3F,
0xF1, 0x08, 0xCA, 0x1 E, // key3 from a byte offset "2"
```

It will be appreciated that the keys can be generated by selecting an offset in any predefined order, and is not necessarily always generated by increasing the offset by one as illustrated in the abovementioned example for the example key code list provided in the option "*A*".

Option C:

[0042] Using the same key code list, it is possible to generate 1024 keys (= 8 x 16 x 8 keys) instead of the abovementioned eight (8) and 128 keys by choosing the keys based on bit offsets instead of key and byte offsets. For example, the first three 16-byte keys (generated in the option "*B*") can be converted to bits as follows:

```
0100 1011 1101 1010 0111 0010 0100 0100 1011 0011 0001 0010
0000 0111 0100 0011
0110 1111 0110 0101 1000 0011 1000 1100 1111 0101 0011 1111
1111 0001 0000 1000
1101 1010 0111 0010 0100 0100 1011 0011 0001 0010 0000 0111
0100 0011 0110 1111
0110 0101 1000 0011 1000 1100 1111 0101 0011 1111 1111 0001
0000 1000 1100 1010
0111 0010 0100 0100 1011 0011 0001 0010 0000 0111 0100 0011
0110 1111 0110 0101
1000 0011 1000 1100 1111 0101 0011 1111 1111 0001 0000 1000
1100 1010 0001 1110
```

[0043] Thus, a first 128-bit key from a bit offset "*0*" is:

```
0100 1011 1101 1010 0111 0010 0100 0100 1011 0011 0001 0010
0000 0111 0100 0011
0110 1111 0110 0101 1000 0011 1000 1100 1111 0101 0011 1111
1111 0001 0000 1000
```

and a second 128-bit key from a bit offset "*1*" is:

```
1001 0111 1011 0100 1110 0100 1000 1001 0110 0110 0010 0100
0000 1110 1000 0110
1101 1110 1100 1011 0000 0111 0001 1001 1110 1010 0111 1111
1110 0010 0001 0001
```

and a third 128-bit key from a bit offset "2" is:

```
0010 1111 0110 1001 1100 1001 0001 0010 1100 1100 0100 1000
0001 1101 0000 1101
1011 1101 1001 0110 0000 1110 0011 0011 1101 0100 1111 1111
1100 0100 0010 0011
```

[0044] In other words, when compared to the original keys, this technique is capable of generating 128 times more keys with the same amount of key materials.

[0045] It will be appreciated that instead of using the aforementioned key offset, the aforementioned byte offset or the aforementioned bit offset, other type of offsets, for example, such as word offsets can also be used, depending on whether it is more important to increase the processing speed or to generate a larger amount of keys.

Option D:

[0046] According to an embodiment, 65535 keys are expanded in device memory of the given user device, for example, as follows:

(a) a 128-bit key is expanded to, for example, a 352-bit key;
(b) a 192-bit key is expanded to, for example, a 432-bit key; and
(c) a 256-bit key is expanded to, for example, a 512-bit key.

[0047] These keys can then be used in respect of the key store integrated services, for example, using algorithms such as AES, Salsa20 and ChaCha20, but not limited thereto.

[0048] The key materials may be used for various purposes, for example, such as cryptography, data protection (for example, encryption and decryption), signing, integrity, verification, authentication, authorization and similar. Advantageously, the key materials are made accessible for use, internally within the protected key store, to the key store integrated services, which access the key materials for use via the key references only. In other words, the key materials are not accessible by software applications or ecosystem processes from outside of the key store.

[0049] In an event that a malicious party makes an attempt to use a key reference to access, evaluate or debug its corresponding key material, an exception is optionally raised. As an example, if the key store is implemented on Android™ and technical interfaces are built using Java, where a technical implementation of security solutions is mixed between Sun Microsystem®'s Java and Google Android™'s Java, the key store should support exactly required interfaces defined in Google Android™ developer's reference, so that the technical implementation may be done using already existing Java Application Programming Interface (API). However, the technical implementation of the key store does not allow to access, evaluate or debug a key material referenced by a given key reference.

[0050] In embodiments of the present disclosure, the key store content is created by the key service provider, in the aforementioned format, namely in the format that is compatible with the given user device, so as to be compliant with an import function of the key store of the given user device. This remarkably speeds up the import procedure at the step (ii) at the given user device. Optionally, the key store content is created by the key service provider in a format that is compliant with a key-store import function of a wide spectrum of user devices; for example, the user devices employ various types of proprietary secure key stores implemented in hardware, such as aforementioned TEE, and employ a software-supported interface to provide a portal of standardized functionality presented by the secure key store to a received encrypted key store content file sent by the key service provider. Optionally, in this regard, at the key service provider, the key store content is individually customized to be compatible with various different types of user devices.

[0051] Examples of such user devices include, but are not limited to, mobile phones, smart telephones, Mobile Internet Devices (MIDs), tablet computers, Ultra-Mobile Personal Computers (UMPCs), phablet computers, Personal Digital Assistants (PDAs), web pads, Personal Computers (PCs), handheld PCs, laptop computers, desktop computers, and interactive entertainment devices, such as game consoles, Television (TV) sets and Set-Top Boxes (STBs).

[0052] Moreover, it will be appreciated that the key store of the given user device may be either hardware-based or software-based, for example implemented using hardware as in TEE ("*trusted execution environment*"), that prevents export of data therefrom after initial importing and loading of the key store content to the protected key store of the given user device.

[0053] According to an embodiment of the present disclosure, the key store is hardware-based. Optionally, in such a case, the importing at the step (ii) includes binding the key materials stored at the hardware-based key store to a secure area of processing hardware of the given user device. Subsequently, in use, key materials stored in the key store are accessed for use, via use of their references, but are not exportable from the key store. Optionally, a pointer is used to transfer a key reference of a key material to be used by a key store integrated service.

[0054] One or more trusted software applications, for example encryption algorithms and/or decryption algorithms that require use of the key materials in the key store are protected in operation by a kernel layer of the given user device. The kernel layer of the given user device, for example, is implemented in a mixture of hardware and software, and is often proprietary to the given user device, for example proprietary to a manufacturer of the given user device. The trusted software applications interface to other software applications supported in other software layers supported in operation on the given user device. Beneficially, the one or more trusted software applications are downloaded in encrypted form from a trusted software service provider. Optionally, the key service provider and the trusted software service provider are the same party. Alternatively, optionally, the key service provider and the trusted software service provider are mutually different parties.

[0055] Thus, it will be appreciated, in a given user device including a hardware-implemented key store, that there is also a kernel layer and one or more software layers hosted in the device. Software applications can be imported and then executed in the one or more software layers. Moreover, other trusted software applications provided by the trusted software service provider can be executed in the kernel layer, in which case the trusted software applications are protected by security provisions of the kernel layer that are generally more secure than the one or more software layers; the software applications protected by security provisions of the kernel layer are referred to as "*key store integrated services*" for purposes of the present disclosure. In operation, various data exchanges occur between applications supported in the one or more software layers and the "*key store integrated services*" hosted in the kernel layer.

[0056] Optionally, the secure area of the processing hardware is implemented by way of dedicated hardware that is configured to disallow externally-loaded software applications or programs, namely in the aforementioned one or more software layers, to operate on the dedicated hardware. It will be appreciated that such externally-loaded software applications or programs could be maliciously loaded by hostile third parties. More optionally, the secure area of the processing hardware is implemented by way of Trusted Execution Environment (TEE; see reference [1]), for example as aforementioned.

[0057] In this way, the method facilitates a solid and strong integration between software and security hardware of the given user device.

[0058] According to an embodiment of the present disclosure, the "key materials" include at least one of:

(a) secret keys for symmetric data encryption,

(b) private keys and public keys for a Public Key Infrastructure (PKI)-equivalent usage,

(c) certificates to be used for cryptography, signing, integrity, verification, authentication, authorization and similar,

(d) one or more key generators for generating keys.

[0059] Optionally, in this regard, the one or more key generators are used to generate the keys reproducibly. In other words, each time a same input is used, a same key is generated by a given key generator.

[0060] Optionally, one or more of the key materials are individually protected with additional encryption. This is particularly beneficial for certain security applications.

[0061] It will be appreciated that even if PKI as such uses asymmetric encryption, the key materials can still be imported using symmetric encryption.

[0062] Moreover, optionally, the key materials include disposable keys that are to be used only once and are to be discarded after use. Such disposable keys may, for example, be used for signing-in to a given service. Additionally or alternatively, optionally, at least some of the keys are reusable encryption keys.

[0063] Furthermore, optionally, the key store is capable of acting as a key generator, and is capable of generating new keys reproducibly.

[0064] Moreover, according to an embodiment of the present disclosure, the method includes integrating, with the key store, one or more trusted software applications or ecosystem processes hosted at the given user device that are authorized to use the key store of the given user device. Such integrated software applications or ecosystem processes are referred to as "*key store integrated services*" throughout the present disclosure, as aforementioned. Examples of the key store integrated services include, but are not limited to, data delivery services, content delivery services, banking services, and financial transaction services; such services typically involve encrypting and/or decrypting data using one or more keys.

[0065] Optionally, in this regard, the method includes importing, from the trusted software service provider, the one or more trusted software applications for providing key store integrated services, wherein the one or more trusted software applications when executed at the given user device are operable to provide one or more key store integrated services and are provided with protection from a kernel of the given user device.

[0066] Moreover, optionally, when the key store is hardware-based, the key materials are encrypted using symmetric encryption that is compliant with the hardware-based key store. Optionally, in this regard, the method includes encrypting the key materials, at the key service provider, by employing symmetric Advanced Encryption Standard (AES; see reference [2]) encryption, for example, using a 128-bit key or a 256-bit key.

[0067] Alternatively, optionally, when the key store is software based, the key store content is encrypted using asymmetric encryption that is compliant with the software-based key store.

[0068] It will be appreciated here that for the given user device to be able to decrypt the encrypted key store content, the encryption credentials used during encryption must be known to the given user device. It will be appreciated that it is not relevant in embodiments of the present disclosure, which encryption algorithm or which kind of encryption credentials are used to encrypt the key store content, because different device vendors and ecosystem providers may implement multiple different security solutions, which may then be implemented by multiple different security service providers on different platforms with their own hardware-based or software-based key stores.

[0069] Moreover, as mentioned previously, the encryption credentials that were used to encrypt the key materials are compatible with the given user device. Such compatible encryption credentials may be provided by the given user device or by the key service provider. Optionally, in this regard, the method includes encrypting the key store content, at the key service provider, using encryption key data provided by the given user device or by the key service provider.

[0070] According to an embodiment of the present disclosure, the method includes protecting the key store, at the given user device, using a token of the end user's bio-credential. Optionally, in this regard, the end user's bio-credential includes at least one of: a fingerprint of the end user, facial features of the end user, a DNA profile of the end user, iris recognition of the end user, a walking manner of the end user, a writing manner of the end user, a heartbeat pattern of the end user. It will be appreciated that the end user's bio-credential used for protecting the key store are provided by way of hardware-based functionalities of the end user's device. These hardware-based functionalities may, for example, be implemented by way of TEE. As an example, the facial features of the end user could be captured using a camera of the end user's device and verified against a reference template using image correlation or use of neural network algorithms. It will be appreciated that the end user's bio-credential may alternatively correspond to any other type of biometrical verification feasible in the future.

[0071] According to another embodiment of the present disclosure, the method includes protecting the key store, at the given user device, using a personal identification code (PIN) associated with the end user. It will be appreciated that the PIN is provided by way of the hardware-based functionalities of the end user's device.

[0072] According to yet another embodiment of the present disclosure, the method includes protecting the key store, at the given user device, using an application-specific identification (ID). Optionally, the application-specific ID is provided by way of the hardware-based functionalities of the end user's device. Alternatively, optionally, the application-specific ID is provided by way of platform-based functionalities. Optionally, in such a case, the application-specific ID is an instance identifier (namely, instanceID).

[0073] Furthermore, according to an embodiment of the present disclosure, the key store content is received at the step (i) via unsecured transportation. As an example, the encrypted key store content can be communicated via non-secured public Internet connection, because when properly-protected the encrypted key store content does not reveal any user-sensitive data. This is possible because the key materials are protected using encryption, and therefore, the transportation of the key materials is not necessary to be protected.

[0074] In a second aspect, embodiments of the present disclosure provide a computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute the method pursuant to the aforementioned first aspect.

[0075] Optionally, the computer-readable instructions are downloadable from a software application store, for example, from an "*App store*" to the computerized device.

[0076] In a third aspect, embodiments of the present disclosure provide a system for protecting usage of key store content at a given user device of an end user, characterized in that the system is operable to:

(i) receive, at the given user device (106), the key store content (102) including key materials that are encrypted using encryption credentials compatible with the given user device (106), the key store content (102) being created by and received from a key service provider (104) in a format that is compatible with the given user device (106), wherein the key materials are provided as a key code list;

(ii) import the encrypted key materials of the key store content (102) to a protected key store (108) of the given user device (106) and store the key materials at the protected key store (108) in the encrypted form, wherein all the key encrypted materials of the key store content (102) are imported at one go, wherein the key store content (102) is stored at the key store in a manner that the key materials are non-exportable from the key store, and wherein the importing com prises:

- decrypting the encrypted key materials to obtain the key code list, and
- generating keys by choosing the keys from the key code list based upon key offsets, byte offsets and/or bit offsets; and

(iii) internally within the protected key store (108) of the given user device (106), allow one or more key store integrated services of the given user device (106) to access the non-exportable key materials for use via key references only.

[0077] Optionally, in addition to the key materials, the protected key store also includes information about the end user and/or a group of end users that are authorized to use the key materials stored at the protected key store. Additionally or alternatively, optionally, the protected key store includes other information relevant for usage of the key materials.

[0078] Optionally, the key materials are received at (i) in a symmetrically-encrypted form.

[0079] Optionally, the system is operable to, internally within the protected key store of the given user device, decrypt one or more of the key materials to be used by the one or more key store integrated services of the given user device.

[0080] According to an embodiment, the key references are implemented by way of indices of the key materials. Optionally, the system is operable to receive, within the key store content, the indices to be used for referencing the key materials via the key references. Alternatively, optionally, the system is operable to generate, at the given user device, the indices to be used for referencing the key materials via the key references. As an example, the indices may be generated in a consecutive manner corresponding to an order in which the key materials are included in the key store content. The indices can be generated, for example, at an initial registration of the given user device to the key service provider or during the decryption of the key materials.

[0081] According to another embodiment, the key references are implemented by way of offsets based upon which the key materials are to be identified.

[0082] Optionally, when importing at (ii), the system is operable to import the key store content to the given user device as a single data file.

[0083] It will be appreciated here that embodiments of the present disclosure are suitable for various different types of user devices. Examples of such user devices include, but are not limited to, mobile phones, smart telephones, MIDs, tablet computers, UMPCs, phablet computers, PDAs, web pads, PCs, handheld PCs, laptop computers, desktop computers, and interactive entertainment devices, such as game consoles, TV sets and STBs.

[0084] According to an embodiment of the present disclosure, the key store is hardware-based. Optionally, in such a case, when importing at (ii), the system is operable to bind the key materials stored at the hardware-based key store to a secure area of processing hardware of the given user device.

[0085] Optionally, the secure area of the processing hardware is implemented by way of dedicated hardware that is configured to disallow externally-loaded software applications or programs to operate on the dedicated hardware; for example software applications that are externally loaded are operable to interface via key store integrated services provided by trusted software applications that are protected by the kernel layer of the end user's device, wherein the key store integrated services shield the key store from direct access by the externally loaded software applications. It will be appreciated that such externally-loaded software applications or programs could be maliciously loaded by hostile third parties. However, it will be appreciated that the key store integrated services are implemented using trusted software applications provided from a trusted software service provider, as aforementioned. More optionally, the secure area of the processing hardware is implemented by way of TEE (see reference [1]).

[0086] According to an embodiment of the present disclosure, the "key materials" include at least one of:

(a) secret keys for symmetric data encryption,

(b) private keys and public keys for a PKI-equivalent usage,

(c) certificates to be used for cryptography, signing, integrity, verification, authentication, authorization and similar,

(d) one or more key generators for generating keys.

[0087] Moreover, according to an embodiment of the present disclosure, the system is operable to integrate, with the key store, one or more trusted software applications or ecosystem processes hosted at the given user device that are authorized to use the key store of the given user device. Examples of such key store integrated services include, but are not limited to, data delivery services, content delivery services, banking services, and financial transaction services.

[0088] Optionally, in this regard, the system is operable to import, from a trusted software service provider, the one or more trusted software applications for providing key store integrated services, wherein the one or more trusted software applications when executed at the given user device are operable to provide one or more key store integrated services and are provided with protection from a kernel of the given user device.

[0089] Furthermore, according to an embodiment of the present disclosure, the key service provider is operable to encrypt the key store content using encryption key data provided by the given user device or by the key service provider.

[0090] According to an embodiment of the present disclosure, the key store is protected at the given user device using a token of the end user's bio-credential. Optionally, in this regard, the end user's bio-credential includes at least one of: a fingerprint of the end user, facial features of the end user, a DNA profile of the end user, iris recognition of the end user, a walking manner of the end user, a writing manner of the end user, a heartbeat pattern of the end user. It will be appreciated that the end user's bio-credential may alternatively correspond to any other type of biometrical verification feasible in the future.

[0091] According to another embodiment of the present disclosure, the key store is protected at the given user device using a PIN associated with the end user.

[0092] According to yet another embodiment of the present disclosure, the key store is protected at the given user device using an application-specific ID.

[0093] Optionally, the key service provider is operable to encrypt the key materials by employing symmetric AES encryption (see reference [2]), for example, using a 128-bit key or a 256-bit key.

[0094] Moreover, according to an embodiment of the present disclosure, when receiving at (i), the system is operable to receive the key store content via unsecured transportation.

[0095] Next, embodiments of the present disclosure will be described with reference to figures.

[0096] Referring to FIG. 1A, there is provided a schematic illustration of a system **100** for protecting usage of key store content **102,** in accordance with an embodiment of the present disclosure. The system **100** includes a key service provider **104** and a given user device **106** of an end user, wherein the key service provider **104** and the given user device **106** are coupled in communication via a data communication arrangement.

[0097] The key service provider **104** creates the key store content **102** in a format that is compatible with the given user device **106,** encrypts key materials included in the key store content **102,** and sends the key store content **102** to the given user device **106.** Optionally, the key store content **102** is importable to a protected key store **108** of the given user device **106** as a single data file.

[0098] At the given user device **106,** the key store content **102** (namely, all the key materials) is imported to the protected key store **108** of the given user device **106** at one go, wherein the key materials are stored in the encrypted form and in a manner that the key materials are non-exportable from the protected key store **108,** and are accessible for use by key store integrated services via key references only.

[0099] As described earlier, the key references may be implemented by way of indices or offsets. Optionally, the indices are received in the key store content **102;** alternatively, optionally, the indices are generated at the given user device **106,** for example in a consecutive manner corresponding to an order in which the key materials are included in the key store content **102.**

[0100] A trusted software service provider **120** provides one or more trusted software applications **122** that are imported in encrypted form to the protected key store **108** of the given user device **106,** wherein the one or more trusted software applications **122** are executable upon the given user device **106** in a manner that is protected by a kernel **124,** for example a kernel layer, of the given user device **106.** The one or more trusted software applications **122** are operable to use the key references to access the key materials of the key store **108** for various purposes, for example encryption, decryption, verification, authentication, but are prevented from divulging the key materials to other software applications that are supported in one or more software layers of the given user device **106.** As the key materials are stored in the encrypted form, the key materials are required to be decrypted prior to use. Optionally, the encrypted key materials are decrypted securely within the key store **108,** when loading the key materials to the key store **108.**

[0101] Optionally, the trusted software service provider **120** is a same party as the key service provider **104.** Alternatively, optionally, the trusted software service provider **120** is a mutually different party to the key service provider **104.**

[0102] In FIG. 1B, there is provided a schematic illustration of a complete end-to-end process flow of protecting the usage of the key store content **102** at the given user device **106,** in accordance with an embodiment of the present disclosure.

Step 1: The key service provider **104** creates the key store content **102** in the format that is compatible with the given user device **106.**

Step 2: The key service provider **104** encrypts the key materials included in the key store content **102.**

Step 3: The given user device **106** receives the key store content **102** from the key service provider **104.**

Step 4: The encrypted key materials are imported to the protected key store **108** of the given user device **106.** Optionally, the key store **108** is protected using encryption credentials from the end user.

Step 5: The key materials are loaded to the key store, wherein the key materials are decrypted securely within the key store, in order to enable easy and fast usage thereof.

Step 6: The key store integrated services **122** access the key materials, internally within the protected key store, via key references only.

Prohibited step 7: The key materials are non-exportable, and cannot be exported from the key store **108.**

[0103] Furthermore, in FIG. 1C, there is provided a schematic illustration of how the key store content **102** is imported and loaded to the protected key store **108,** in accordance with an embodiment of the present disclosure.

[0104] Upon receiving the key store content **102** from the key service provider **104,** the encrypted key materials included therein are imported to the protected key store **108** of the user device **106** at one go.

[0105] Notably, the key materials may be provided as a single key code list or as a plurality of different key code lists. It will be appreciated that different key code lists can be imported simultaneously. Irrespective of the form in which different key code lists, namely the key materials, are encrypted, all the key materials are imported at one go.

[0106] The encrypted key materials are then decrypted securely within the key store **108,** when the key materials are loaded at the key store **108.**

[0107] FIGs. 1A, 1B and 1C are merely examples, which should not unduly limit the scope of the claims herein. It is to be understood that the specific designation for the system **100** is provided as an example and is not to be construed as limiting the system **100** to specific numbers, types, or arrangements of service providers and user devices; specifically, a single user device has been shown for the sake of simplicity only. A person skilled in the art will recognize many variations, alternatives, and modifications of embodiments of the present disclosure.

[0108] It will be appreciated that even though FIGs. 1B and 1C show the indices of the key materials, the key references are not necessarily always implemented by way of such indices. Notably, in alternative implementations, the key references can be implemented using offsets, as described earlier.

[0109] Referring next to FIG. 2, there is provided a flow chart depicting steps of a method of protecting usage of key store content at a given user device of an end user, in accordance with an embodiment of the present disclosure. The method is depicted as a collection of steps in a logical flow diagram, which represents a sequence of steps that can be implemented in hardware, software, or a combination thereof, for example as aforementioned.

[0110] At a step **202,** the key store content is received at the given user device. In accordance with the step **202,** the key store content includes key materials that are encrypted using encryption credentials compatible with the given user device. The key store content is created by and received from a key service provider in a format that is compatible with the given user device.

[0111] At a step **204,** the encrypted key materials of the key store content are imported to a protected key store of the given user device at one go, and the key materials are stored at the protected key store in the encrypted form. In accordance with the step **204,** the key store content is stored at the protected key store in a manner that the key materials are non-exportable from the key store.

[0112] At a step **206,** internally within the protected key store of the given user device, one or more key store integrated services of the given user device are allowed to access the non-exportable key materials for use, via key references only. As aforementioned, such integrated services are provided by executable software that is run within protection of the kernel layer, for example a kernel structure, of the given user device. Optionally, the kernel structure includes hardware, for achieving an enhanced degree of security.

[0113] The steps **202** to **206** are only illustrative and other alternatives can also be provided where one or more steps are added without departing from the scope of the claims herein.

[0114] Modifications to embodiments of the present disclosure described in the foregoing are possible without departing from the scope of the present disclosure as defined by the accompanying claims. Expressions such as "including", "comprising", "incorporating", "consisting of", "have", "is" used to describe and claim the present invention are intended to be construed in a non-exclusive manner, namely allowing for items, components or elements not explicitly described also to be present. Reference to the singular is also to be construed to relate to the plural; as an example, *"at least one of"* indicates "*one of*" in an example, and "*a plurality of*" in another example; moreover, "*two of*", and similarly "one or more" are to be construed in a likewise manner. Numerals included within parentheses in the accompanying claims are intended to assist understanding of the claims and should not be construed in any way to limit subject matter claimed by these claims.

[0115] The phrases "in an embodiment", "according to an embodiment" and the like generally mean the particular feature, structure, or characteristic following the phrase is included in at least one embodiment of the present disclosure, and may be included in more than one embodiment of the present disclosure. Importantly, such phrases do not necessarily refer to the same embodiment.

REFERENCES

[0116]

[1] Trusted execution environment - Wikipedia, the free encyclopedia (accessed November 28, 2016); URL: https://en.wikipedia.org/wiki/Trusted_execution_environment

[2] Advanced Encryption Standard - Wikipedia, the free encyclopedia (accessed November 28, 2016); URL: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Claims

1. A method of protecting usage of key store content (102) at a given user device (106) of an end user, the method includes steps of:

   (i) receiving, at the given user device (106), the key store content (102) including key materials that are encrypted using encryption credentials compatible with the given user device (106), the key store content (102) being created by and received from a key service provider (104) in a format that is compatible with the given user device (106), wherein the key materials are provided as a key code list;
   (ii) importing the encrypted key materials of the key store content (102) to a protected key store (108) of the

given user device (106) and storing the key materials at the protected key store (108) in the encrypted form, wherein all the encrypted key materials of the key store content (102) are imported at one go, wherein the key store content (102) is stored at the key store in a manner that the key materials are non-exportable from the key store, and wherein the importing comprises:

- decrypting the encrypted key materials to obtain the key code list, and
- generating keys by choosing the keys from the key code list based upon key offsets, byte offsets and/or bit offsets; and

(iii) internally within the protected key store (108) of the given user device (106), allowing one or more key store integrated services of the given user device (106) to access the non-exportable key materials for use via key references only.

2. A method of claim 1, **characterized in that** at the step (ii), the key store content (102) is imported as a single data file.

3. A method of claim 1 or 2, **characterized in that** the method includes receiving, within the key store content (102), indices to be used for referencing the key materials via the key references.

4. A method of claim 1 or 2, **characterized in that** the method includes generating, at the given user device (106), indices to be used for referencing the key materials via the key references.

5. A method of any one of claims 1 to 4, **characterized in that** the key store is hardware-based, and the importing at the step (ii) includes binding the key materials stored at the hardware-based key store to a secure area of processing hardware of the given user device (106).

6. A method of any one claims claim 1 to 5, **characterized in that** the method includes integrating, with the key store, one or more trusted software applications (122) or ecosystem processes hosted at the given user device (106) that are authorized to use the key store of the given user device (106).

7. A method of claim 6, **characterized in that** the method includes importing, from a trusted software service provider (120), the one or more trusted software applications (122) for providing key store integrated services, wherein the one or more trusted software applications (122) when executed at the given user device (106) are operable to provide one or more key store integrated services and are provided with protection from a kernel (124) of the given user device (106).

8. A method of any one of claims 1 to 7, **characterized in that** the method includes protecting the key store, at the given user device (106), using a token of the end user's bio-credential.

9. A method of any one of claims 1 to 8, **characterized in that** the key materials are received at the given user device (106) at the step (i) in a symmetrically-encrypted form.

10. A method of any one of claims 1 to 9, **characterized in that** the key materials include at least one of:

(a) secret keys for symmetric data encryption,
(b) private keys and public keys for a Public Key Infrastructure (PKI)-equivalent usage,
(c) certificates to be used for cryptography, signing, integrity, verification, authentication, authorization,
(d) one or more key generators for generating keys.

11. A computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute a method of any one of claims 1 to 10.

12. A system (100) for protecting usage of key store content (102) at a given user device (106) of an end user, the system (100) is operable to:

(i) receive, at the given user device (106), the key store content (102) including key materials that are encrypted using encryption credentials compatible with the given user device (106), the key store content (102) being created by and received from a key service provider (104) in a format that is compatible with the given user

device (106), wherein the key materials are provided as a key code list;
(ii) import the encrypted key materials of the key store content (102) to a protected key store (108) of the given user device (106) and store the key materials at the protected key store (108) in the encrypted form, wherein all the key encrypted materials of the key store content (102) are imported at one go, wherein the key store content (102) is stored at the key store in a manner that the key materials are non-exportable from the key store, and wherein the importing com prises:

- decrypting the encrypted key materials to obtain the key code list, and
- generating keys by choosing the keys from the key code list based upon key offsets, byte offsets and/or bit offsets; and

(iii) internally within the protected key store (108) of the given user device (106), allow one or more key store integrated services of the given user device (106) to access the non-exportable key materials for use via key references only.

13. A system (100) of claim 12, **characterized in that** the system (100) is operable to receive, within the key store content (102), indices to be used for referencing the key materials via the key references.

14. A system (100) of claim 12, **characterized in that** the system (100) is operable to generate, at the given user device (106), indices to be used for referencing the key materials via the key references.

15. A system (100) of any one of claims 12 to 14, **characterized in that** the key store is hardware-based, and when importing at (ii), the system (100) is operable to bind the key materials stored at the hardware-based key store to a secure area of processing hardware of the given user device (106).

16. A system (100) of any one of claims 12 to 15, **characterized in that** the system (100) is operable to integrate, with the key store, one or more trusted software applications (122) or ecosystem processes hosted at the given user device (106) that are authorized to use the key store of the given user device (106).

17. A system (100) of claim 16, **characterized in that** the system (100) is operable to import, from a trusted software service provider (120), the one or more trusted software applications (122) for providing key store integrated services, wherein the one or more trusted software applications (122) when executed at the given user device (106) are operable to provide one or more key store integrated services and are provided with protection from a kernel (124) of the given user device (106).

18. A system (100) of any one of claims 12 to 17, **characterized in that** the key store is protected at the given user device (106) using a token of the end user's bio-credential.

Patentansprüche

1. Verfahren zum Schutz der Verwendung von Schlüsselspeicherungsinhalt (102) an einer gegebenen Benutzervorrichtung (106) eines Endbenutzers, wobei das Verfahren die folgenden Schritte einschließt:

(i) Empfangen, an der gegebenen Benutzervorrichtung (106), des Schlüsselspeicherungsinhalts (102) einschließlich Schlüsselmaterialien, die unter Verwendung von Verschlüsselungsberechtigungsnachweisen verschlüsselt werden, die mit der gegebenen Benutzervorrichtung (106) kompatibel sind, wobei der Schlüsselspeicherungsinhalt (102) durch einen Schlüsseldienstanbieter (104) in einem Format, das mit der gegebenen Benutzervorrichtung (106) kompatibel ist, erzeugt und empfangen wird, wobei die Schlüsselmaterialien als eine Schlüsselcodeliste bereitgestellt werden;
(ii) Importieren der verschlüsselten Schlüsselmaterialien des Schlüsselspeicherungsinhalts (102) in einen geschützten Schlüsselspeicher (108) der gegebenen Benutzervorrichtung (106) und Speichern der Schlüsselmaterialien in dem geschützten Schlüsselspeicher (108) in der verschlüsselten Form, wobei alle verschlüsselten Schlüsselmaterialien des Schlüsselspeicherungsinhalts (102) gleichzeitig importiert werden, wobei der Schlüsselspeicherungsinhalt (102) derart in dem Schlüsselspeicher gespeichert wird, dass die Schlüsselmaterialien nicht aus dem Schlüsselspeicher exportierbar sind, und wobei das Importieren umfasst:

- Entschlüsseln der verschlüsselten Schlüsselmaterialien, um die Schlüsselcodeliste zu erhalten, und

- Erzeugen von Schlüsseln durch Auswählen der Schlüssel aus der Schlüsselcodeliste basierend auf Schlüsselversätzen, Byteversätzen und/oder Bitversätzen; und

(iii) intern innerhalb des geschützten Schlüsselspeichers (108) der gegebenen Benutzervorrichtung (106), Zulassen, dass einer oder mehrere integrierte Schlüsselspeicherdienste der gegebenen Benutzervorrichtung (106) auf nicht exportierbare Schlüsselmaterialien zur Verwendung nur über Schlüsselreferenzen zugreifen.

2. Verfahren nach Anspruch 1, **dadurch gekennzeichnet, dass** in dem Schritt (ii) der Schlüsselspeicherungsinhalt (102) als eine einzelne Datendatei importiert wird.

3. Verfahren nach Anspruch 1 oder 2, **dadurch gekennzeichnet, dass** das Verfahren ein Empfangen, innerhalb des Schlüsselspeicherungsinhalts (102), von Indizes, die zum Referenzieren der Schlüsselmaterialien über die Schlüsselreferenzen verwendet werden sollen, einschließt.

4. Verfahren nach Anspruch 1 oder 2, **dadurch gekennzeichnet, dass** das Verfahren ein Erzeugen, an der gegebenen Benutzervorrichtung (106), von Indizes, die zum Referenzieren der Schlüsselmaterialien über die Schlüsselreferenzen verwendet werden sollen, einschließt.

5. Verfahren nach einem der Ansprüche 1 bis 4, **dadurch gekennzeichnet, dass** der Schlüsselspeicher hardwarebasiert ist und das Importieren in dem Schritt (ii) ein Binden der in dem hardwarebasierten Schlüsselspeicher gespeicherten Schlüsselmaterialien an einen sicheren Bereich von Verarbeitungshardware der gegebenen Benutzervorrichtung (106) einschließt.

6. Verfahren nach einem der Ansprüche 1 bis 5, **dadurch gekennzeichnet, dass** das Verfahren ein Integrieren, mit dem Schlüsselspeicher, von einer/einem oder mehreren an der gegebenen Benutzervorrichtung (106) gehosteten vertrauenswürdigen Softwareanwendungen (122) oder Ökosystemprozessen umfasst, die autorisiert sind, den Schlüsselspeicher der gegebenen Benutzervorrichtung (106) zu verwenden.

7. Verfahren nach Anspruch 6, **dadurch gekennzeichnet, dass** das Verfahren ein Importieren, von einem vertrauenswürdigen Softwaredienstanbieter (120), der einen oder der mehreren vertrauenswürdigen Softwareanwendungen (122) zum Bereitstellen von integrierten Schlüsselspeicherdiensten einschließt, wobei die eine oder die mehreren vertrauenswürdigen Softwareanwendungen (122), wenn sie an der gegebenen Benutzervorrichtung (106) ausgeführt werden, betreibbar sind, um einen oder mehrere integrierte Schlüsselspeicherdienste bereitzustellen, und mit einem Schutz von einem Kern (124) der gegebenen Benutzervorrichtung (106) bereitgestellt werden.

8. Verfahren nach einem der Ansprüche 1 bis 7, **dadurch gekennzeichnet, dass** das Verfahren ein Schützen des Schlüsselspeichers an der gegebenen Benutzervorrichtung (106) unter Verwendung eines Tokens des Bio-Berechtigungsnachweises des Endbenutzers einschließt.

9. Verfahren nach einem der Ansprüche 1 bis 8, **dadurch gekennzeichnet, dass** die Schlüsselmaterialien an der gegebenen Benutzervorrichtung (106) in dem Schritt (i) in einer symmetrisch verschlüsselten Form empfangen werden.

10. Verfahren nach einem der Ansprüche 1 bis 9, **dadurch gekennzeichnet, dass** die Schlüsselmaterialien mindestens eines einschließen von:

(a) geheimen Schlüsseln zur symmetrischen Datenverschlüsselung,
(b) privaten Schlüsseln und öffentlichen Schlüsseln für eine Public-Key-Infrastruktur-äquivalente (PKI-äquivalente) Verwendung,
(c) Zertifikaten, die für Kryptographie, Signierung, Integrität, Verifizierung, Authentifizierung, Autorisierung zu verwenden sind,
(d) einem oder mehreren Schlüsselgeneratoren zum Erzeugen von Schlüsseln.

11. Computerprogrammprodukt, umfassend ein nicht-transitorisches computerlesbares Speichermedium, auf dem computerlesbare Anweisungen gespeichert sind, wobei die computerlesbaren Anweisungen durch eine computerisierte Vorrichtung ausführbar sind, die Verarbeitungshardware umfasst, um ein Verfahren nach einem der Ansprüche 1 bis 10 auszuführen.

12. System (100) zum Schutz der Verwendung von Schlüsselspeicherungsinhalt (102) an einer gegebenen Benutzervorrichtung (106) eines Endbenutzers, wobei das System (100) betreibbar ist zum:

(i) Empfangen, an der gegebenen Benutzervorrichtung (106), des Schlüsselspeicherungsinhalts (102), einschließlich Schlüsselmaterialien, die unter Verwendung von Verschlüsselungsberechtigungsnachweisen verschlüsselt werden, die mit der gegebenen Benutzervorrichtung (106) kompatibel sind, wobei der Schlüsselspeicherungsinhalt (102) durch einen Schlüsseldienstanbieter (104) in einem Format, das mit der gegebenen Benutzervorrichtung (106) kompatibel ist, erzeugt und empfangen wird, wobei die Schlüsselmaterialien als eine Schlüsselcodeliste bereitgestellt werden;
(ii) Importieren der verschlüsselten Schlüsselmaterialien des Schlüsselspeicherungsinhalts (102) in einen geschützten Schlüsselspeicher (108) der gegebenen Benutzervorrichtung (106) und Speichern der Schlüsselmaterialien in dem geschützten Schlüsselspeicher (108) in der verschlüsselten Form, wobei alle verschlüsselten Schlüsselmaterialien des Schlüsselspeicherungsinhalts (102) gleichzeitig importiert werden, wobei der Schlüsselspeicherungsinhalt (102) derart in dem Schlüsselspeicher gespeichert wird, dass die Schlüsselmaterialien nicht aus dem Schlüsselspeicher exportierbar sind, und wobei das Importieren umfasst:

- Entschlüsseln der verschlüsselten Schlüsselmaterialien, um die Schlüsselcodeliste zu erhalten, und
- Erzeugen von Schlüsseln durch Auswählen der Schlüssel aus der Schlüsselcodeliste basierend auf Schlüsselversätzen, Byteversätzen und/oder Bitversätzen; und

(iii) intern innerhalb des geschützten Schlüsselspeichers (108) der gegebenen Benutzervorrichtung (106), Zulassen, dass einer oder mehrere integrierte Schlüsselspeicherdienste der gegebenen Benutzervorrichtung (106) auf nicht exportierbare Schlüsselmaterialien zur Verwendung nur über Schlüsselreferenzen zugreifen.

13. System (100) nach Anspruch 12, **dadurch gekennzeichnet, dass** das System (100) betreibbar ist, um, innerhalb des Schlüsselspeicherungsinhalts (102), Indizes zu empfangen, die zum Referenzieren der Schlüsselmaterialien über die Schlüsselreferenzen verwendet werden sollen.

14. System (100) nach Anspruch 12, **dadurch gekennzeichnet, dass** das System (100) betreibbar ist, um, an der gegebenen Benutzervorrichtung (106), Indizes zu erzeugen, die zum Referenzieren der Schlüsselmaterialien über die Schlüsselreferenzen verwendet werden sollen.

15. System (100) nach einem der Ansprüche 12 bis 14, **dadurch gekennzeichnet, dass** der Schlüsselspeicher hardwarebasiert ist und, beim Importieren in (ii), das System (100) betreibbar ist, um die in dem hardwarebasierten Schlüsselspeicher gespeicherten Schlüsselmaterialien an einen sicheren Bereich von Verarbeitungshardware der gegebenen Benutzervorrichtung (106) zu binden.

16. System (100) nach einem der Ansprüche 12 bis 15, **dadurch gekennzeichnet, dass** das System (100) betreibbar ist, um, mit dem Schlüsselspeicher, eine/einen oder mehrere an der gegebenen Benutzervorrichtung (106) gehostete vertrauenswürdige Softwareanwendungen (122) oder Ökosystemprozesse, die berechtigt sind, um den Schlüsselspeicher der gegebenen Benutzervorrichtung (106) zu verwenden, zu integrieren.

17. System (100) nach Anspruch 16, **dadurch gekennzeichnet, dass** das System (100) betreibbar ist, um, von einem vertrauenswürdigen Softwaredienstanbieter (120), die eine oder die mehreren vertrauenswürdigen Softwareanwendungen (122) zum Bereitstellen von integrierten Schlüsselspeicherdiensten zu importieren, wobei die eine oder die mehreren vertrauenswürdigen Softwareanwendungen (122), wenn sie an der gegebenen Benutzervorrichtung (106) ausgeführt werden, betreibbar sind, um einen oder mehrere integrierte Schlüsselspeicherdienste bereitzustellen, und mit einem Schutz von einem Kern (124) der gegebenen Benutzervorrichtung (106) bereitgestellt werden.

18. System (100) nach einem der Ansprüche 12 bis 17, **dadurch gekennzeichnet, dass** der Schlüsselspeicher an der gegebenen Benutzervorrichtung (106) unter Verwendung eines Tokens des Bio-Berechtigungsnachweises des Endbenutzers geschützt wird.

Revendications

1. Procédé de protection d'utilisation d'un contenu de magasin de clés (102) au niveau d'un dispositif utilisateur donné (106) d'un utilisateur final, le procédé inclut des étapes consistant à :

(i) recevoir, au niveau du dispositif utilisateur donné (106), le contenu de magasin de clés (102) incluant des matériaux de clé qui sont chiffrés en utilisant des informations d'identification de chiffrement compatibles avec le dispositif utilisateur donné (106), le contenu de magasin de clés (102) étant créé par et reçu d'un fournisseur de services de clés (104) dans un format qui est compatible avec le dispositif utilisateur donné (106), dans lequel les matériaux de clé sont fournis en tant que liste de codes de clés ;

(ii) importer les matériaux de clé chiffrés du contenu de magasin de clés (102) vers un magasin de clés protégé (108) du dispositif utilisateur donné (106) et stocker les matériaux de clé au niveau du magasin de clés protégé (108) sous la forme chiffrée, dans lequel tous les matériaux de clé chiffrés du contenu de magasin de clés (102) sont importés en une fois, dans lequel le contenu de magasin de clés (102) est stocké au niveau du magasin de clés d'une manière selon laquelle les matériaux de clé ne peuvent pas être exportés du magasin de clés, et dans lequel l'importation comprend :

- le déchiffrement des matériaux de clé chiffrés pour obtenir la liste de codes de clés, et
- la génération de clés en choisissant les clés à partir de la liste de codes de clés sur la base de décalages de clé, décalages d'octet et/ou décalages de bit ; et

(iii) de façon interne au magasin de clés protégé (108) du dispositif utilisateur donné (106), permettre à un ou plusieurs services intégrés de magasin de clés du dispositif utilisateur donné (106) d'accéder aux matériaux de clé non exportables pour une utilisation par l'intermédiaire de références de clés uniquement.

2. Procédé selon la revendication 1, **caractérisé en ce qu'**à l'étape (ii), le contenu de magasin de clés (102) est importé en tant que fichier de données unique.

3. Procédé selon la revendication 1 ou 2, **caractérisé en ce que** le procédé inclut la réception, à l'intérieur du contenu de magasin de clés (102), d'indices à utiliser pour référencer les matériaux de clé par l'intermédiaire des références de clés.

4. Procédé selon la revendication 1 ou 2, **caractérisé en ce que** le procédé inclut la génération, au niveau du dispositif utilisateur donné (106), d'indices à utiliser pour référencer les matériaux de clé par l'intermédiaire des références de clés.

5. Procédé selon l'une quelconque des revendications 1 à 4, **caractérisé en ce que** le magasin de clés est matériel, et l'importation à l'étape (ii) inclut la liaison des matériaux de clé stockés au niveau du magasin de clés matériel à une zone sécurisée de matériel de traitement du dispositif utilisateur donné (106).

6. Procédé selon l'une quelconque des revendications revendication 1 à 5, **caractérisé en ce que** le procédé inclut l'intégration, avec le magasin de clés, d'une ou plusieurs applications logicielles (122) ou processus d'écosystème, de confiance, hébergés au niveau du dispositif utilisateur donné (106) qui sont autorisés à utiliser le magasin de clés du dispositif utilisateur donné (106).

7. Procédé selon la revendication 6, **caractérisé en ce que** le procédé inclut l'importation, à partir d'un fournisseur de services logiciels de confiance (120), de la ou des applications logicielles de confiance (122) pour fournir des services intégrés de magasin de clés, dans lequel la ou les applications logicielles de confiance (122) lorsqu'elles sont exécutées au niveau du dispositif utilisateur donné (106) sont opérationnelles pour fournir un ou plusieurs services intégrés de magasin de clés et sont pourvues d'une protection à partir d'un noyau (124) du dispositif utilisateur donné (106).

8. Procédé selon l'une quelconque des revendications 1 à 7, **caractérisé en ce que** le procédé inclut la protection du magasin de clés, au niveau du dispositif utilisateur donné (106), en utilisant un jeton d'identification biométrique de l'utilisateur final.

9. Procédé selon l'une quelconque des revendications 1 à 8, **caractérisé en ce que** les matériaux de clé sont reçus au niveau du dispositif utilisateur donné (106) à l'étape (i) sous une forme chiffrée symétriquement.

10. Procédé selon l'une quelconque des revendications 1 à 9, **caractérisé en ce que** les matériaux de clé incluent au moins l'un parmi :

(a) des clés secrètes pour chiffrement symétrique de données,

(b) des clés privées et des clés publiques pour un usage équivalent à une infrastructure à clé publique (ICP),
(c) des certificats à utiliser pour chiffrement, signature, intégrité, vérification, authentification, autorisation,
(d) un ou plusieurs générateurs de clés pour générer des clés.

11. Produit programme informatique comprenant un support de stockage non transitoire lisible par un ordinateur sur lequel sont stockées des instructions lisibles par ordinateur, les instructions lisibles par ordinateur étant exécutables par un dispositif informatisé comprenant un matériel de traitement pour exécuter un procédé selon l'une quelconque des revendications 1 à 10.

12. Système (100) de protection d'utilisation d'un contenu de magasin de clés (102) au niveau d'un dispositif utilisateur donné (106) d'un utilisateur final, le système (100) est opérationnel pour :

(i) recevoir, au niveau du dispositif utilisateur donné (106), le contenu de magasin de clés (102) incluant des matériaux de clé qui sont chiffrés en utilisant des informations d'identification de chiffrement compatibles avec le dispositif utilisateur donné (106), le contenu de magasin de clés (102) étant créé par et reçu d'un fournisseur de services de clés (104) dans un format qui est compatible avec le dispositif utilisateur donné (106), dans lequel les matériaux de clé sont fournis en tant que liste de codes de clés ;
(ii) importer les matériaux de clé chiffrés du contenu de magasin de clés (102) vers un magasin de clés protégé (108) du dispositif utilisateur donné (106) et stocker les matériaux de clé au niveau du magasin de clés protégé (108) sous la forme chiffrée, dans lequel tous les matériaux chiffrés de clé du contenu de magasin de clés (102) sont importés en une fois, dans lequel le contenu de magasin de clés (102) est stocké au niveau du magasin de clés d'une manière selon laquelle les matériaux de clé ne peuvent pas être exportés du magasin de clés, et dans lequel l'importation comprend :

- le déchiffrement des matériaux de clé chiffrés pour obtenir la liste de codes de clés, et
- la génération de clés en choisissant les clés à partir de la liste de codes de clés sur la base de décalages de clé, décalages d'octet et/ou décalages de bit ; et

(iii) de façon interne au magasin de clés protégé (108) du dispositif utilisateur donné (106), permettre à un ou plusieurs services intégrés de magasin de clés du dispositif utilisateur donné (106) d'accéder aux matériaux de clé non exportables pour une utilisation par l'intermédiaire de références de clés uniquement.

13. Système (100) selon la revendication 12, **caractérisé en ce que** le système (100) est opérationnel pour recevoir, au sein du contenu de magasin de clés (102), des indices à utiliser pour référencer les matériaux de clé par l'intermédiaire des références de clés.

14. Système (100) selon la revendication 12, **caractérisé en ce que** le système (100) est opérationnel pour générer, au niveau du dispositif utilisateur donné (106), des indices à utiliser pour référencer les matériaux de clé par l'intermédiaire des références de clés.

15. Système (100) selon l'une quelconque des revendications 12 à 14, **caractérisé en ce que** le magasin de clés est de type matériel, et lors de l'importation en (ii), le système (100) est opérationnel pour lier les matériaux de clé stockés au niveau du magasin de clés matériel vers une zone sécurisée de matériel de traitement du dispositif utilisateur donné (106).

16. Système (100) selon l'une quelconque des revendications 12 à 15, **caractérisé en ce que** le système (100) est opérationnel pour intégrer, avec le magasin de clés, une ou plusieurs applications logicielles (122) ou processus d'écosystème, de confiance, hébergés au niveau du dispositif utilisateur donné (106) qui sont autorisés à utiliser le magasin de clés du dispositif utilisateur donné (106).

17. Système (100) selon la revendication 16, **caractérisé en ce que** le système (100) est opérationnel pour importer, à partir d'un fournisseur de services logiciels de confiance (120), la ou les applications logicielles de confiance (122) pour fournir des services intégrés de magasin de clés, dans lequel la ou les applications logicielles de confiance (122) lorsqu'elles sont exécutées au niveau du dispositif utilisateur donné (106) sont opérationnelles pour fournir un ou plusieurs services intégrés de magasin de clés et sont pourvues d'une protection à partir d'un noyau (124) du dispositif utilisateur donné (106).

18. Système (100) selon l'une quelconque des revendications 12 à 17, **caractérisé en ce que** le magasin de clés est protégé au niveau du dispositif utilisateur donné (106) en utilisant un jeton d'identification biométrique de l'utilisateur final.

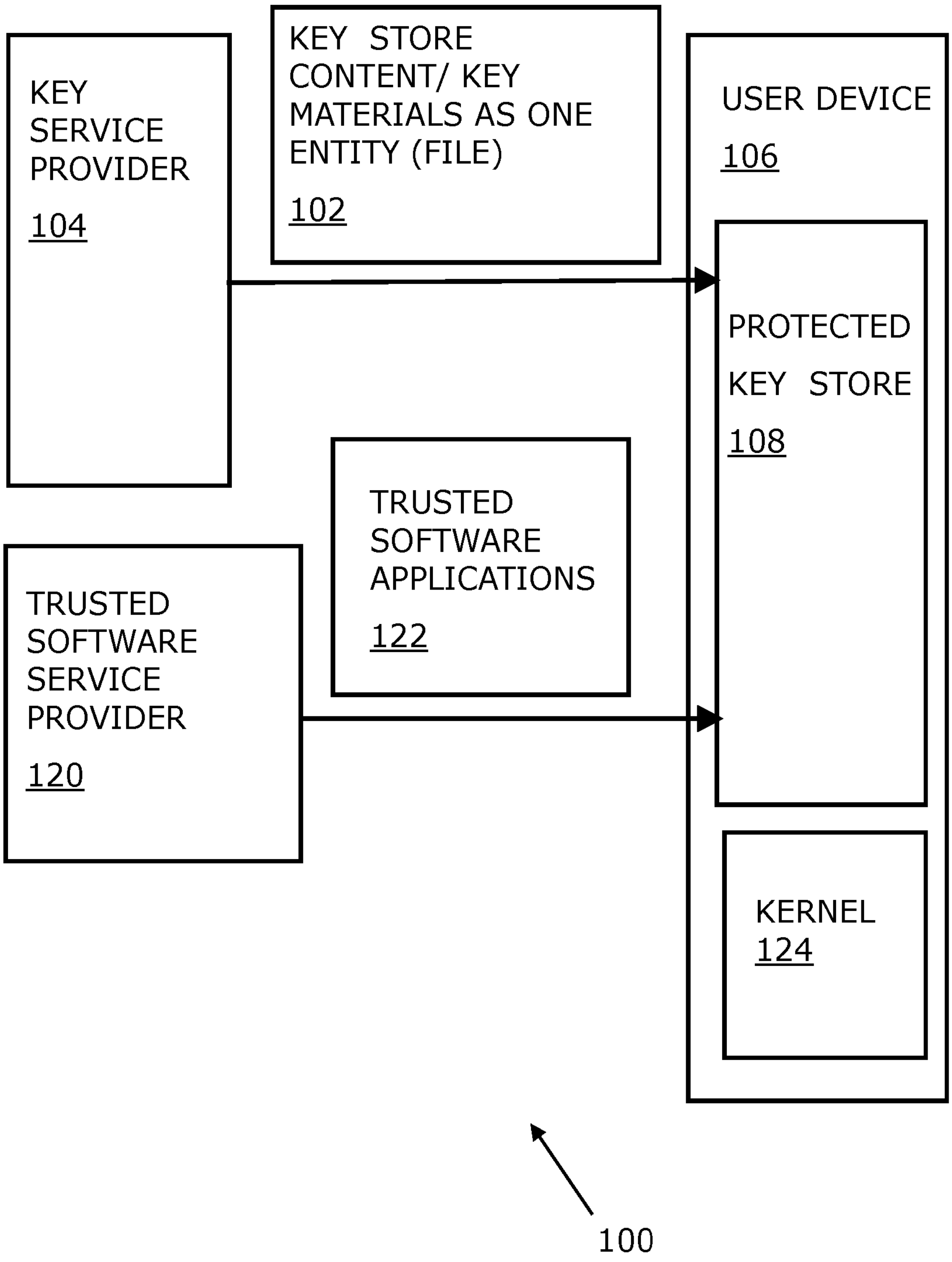
KEY SERVICE PROVIDER
104
KEY STORE CONTENT/ KEY MATERIALS AS ONE ENTITY (FILE)
102
USER DEVICE
106
PROTECTED KEY STORE
108
TRUSTED SOFTWARE APPLICATIONS
122
TRUSTED SOFTWARE SERVICE PROVIDER
120
KERNEL
124
100

FIG. 1A

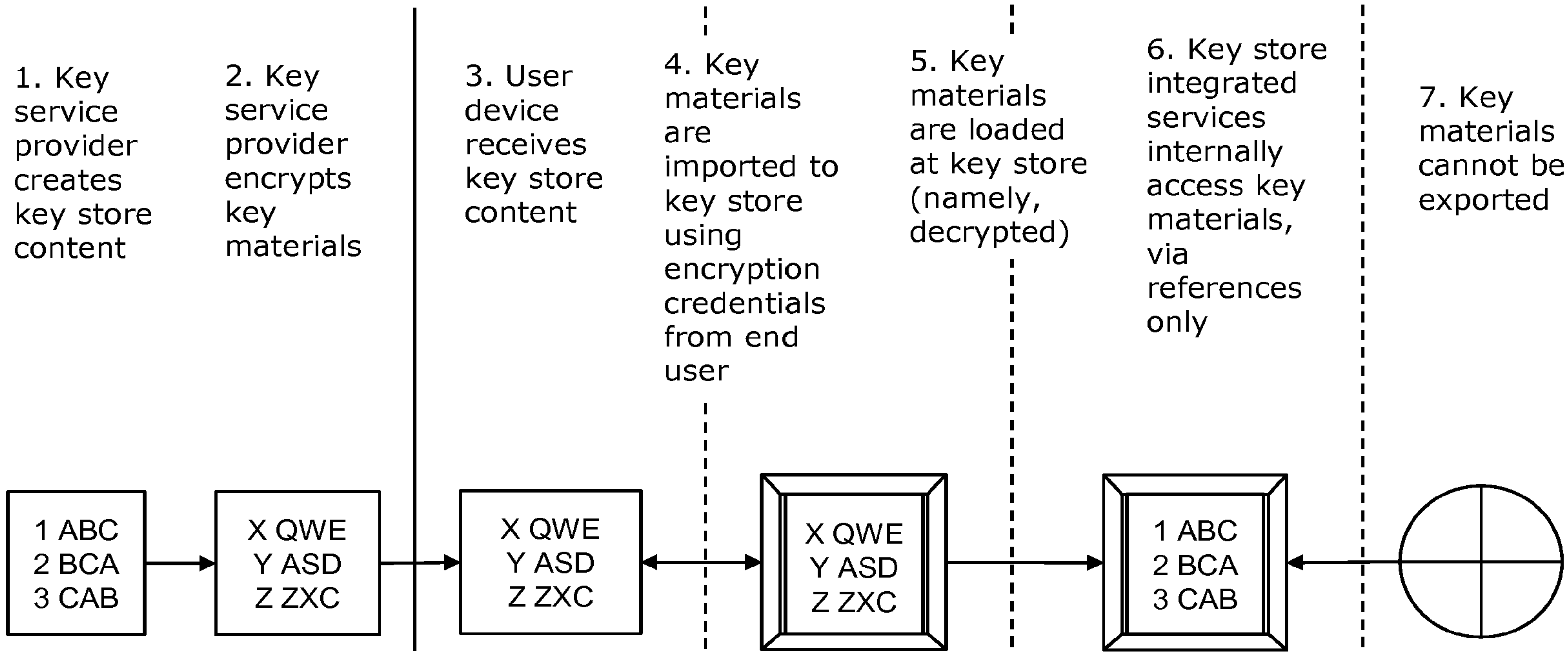
1. Key service provider creates key store content
2. Key service provider encrypts key materials
3. User device receives key store content
4. Key materials are imported to key store using encryption credentials from end user
5. Key materials are loaded at key store (namely, decrypted)
6. Key store integrated services internally access key materials, via references only
7. Key materials cannot be exported
1 ABC
2 BCA
3 CAB
X QWE
Y ASD
Z ZXC
X QWE
Y ASD
Z ZXC
X QWE
Y ASD
Z ZXC
1 ABC
2 BCA
3 CAB

FIG. 1B

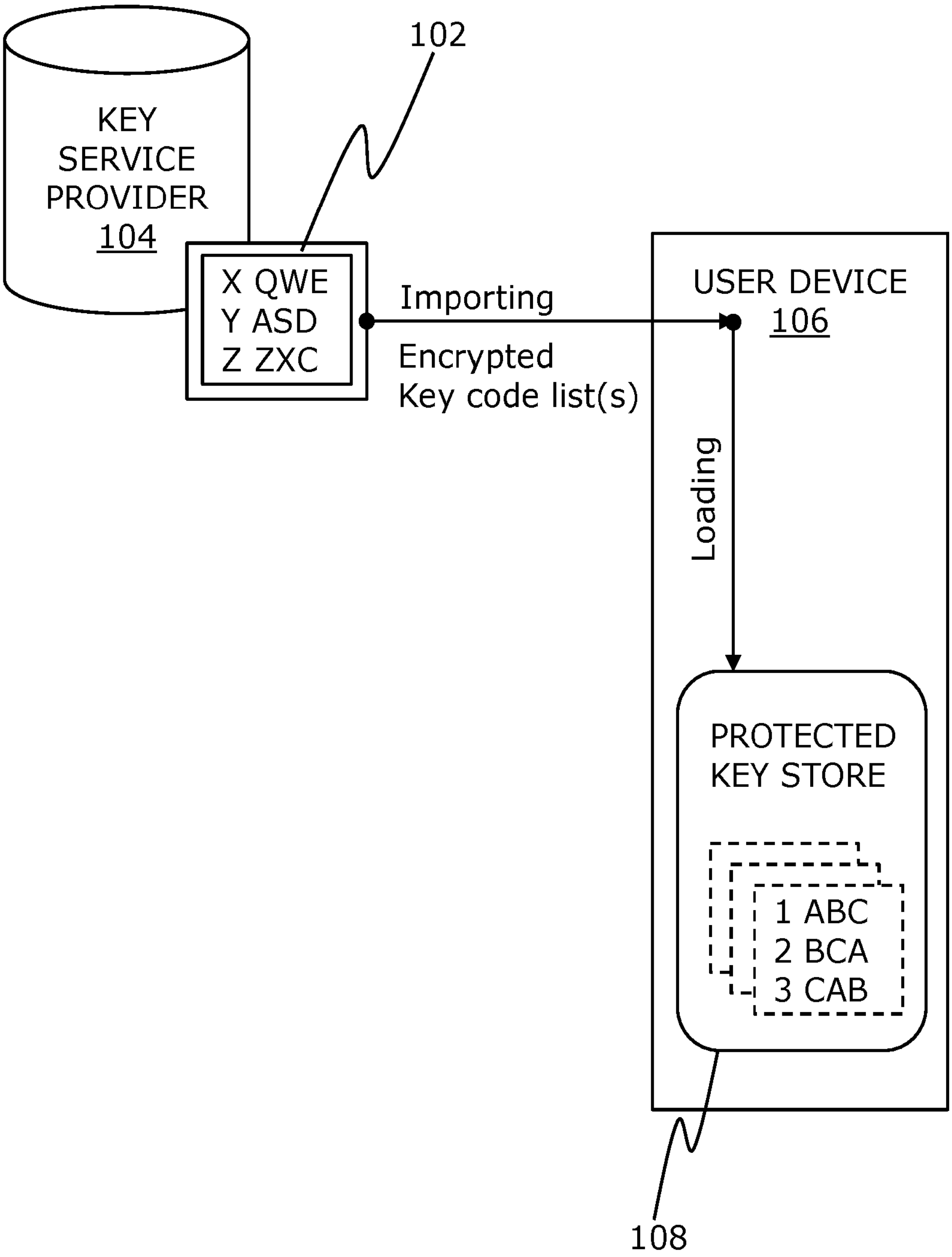
102
KEY
SERVICE
PROVIDER
104
X QWE
Y ASD
Z ZXC
Importing
Encrypted
Key code list(s)
USER DEVICE
106
Loading
PROTECTED
KEY STORE
1 ABC
2 BCA
3 CAB
108

FIG. 1C

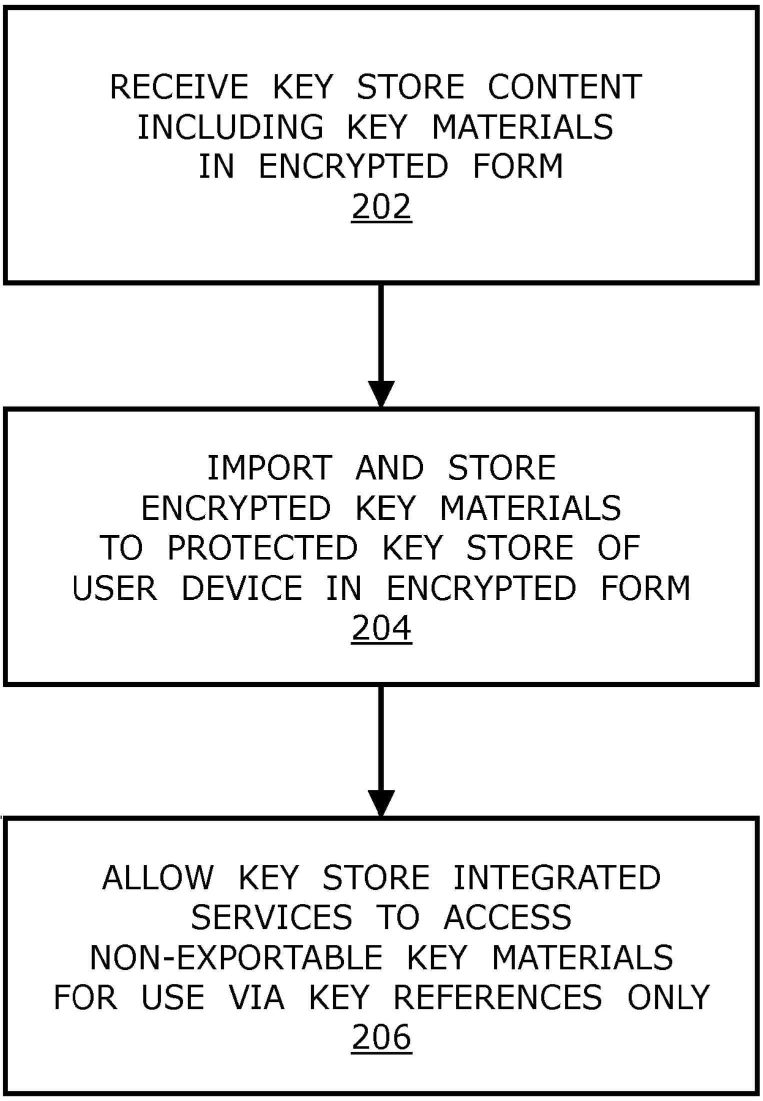
RECEIVE KEY STORE CONTENT INCLUDING KEY MATERIALS IN ENCRYPTED FORM
202
IMPORT AND STORE ENCRYPTED KEY MATERIALS TO PROTECTED KEY STORE OF USER DEVICE IN ENCRYPTED FORM
204
ALLOW KEY STORE INTEGRATED SERVICES TO ACCESS NON-EXPORTABLE KEY MATERIALS FOR USE VIA KEY REFERENCES ONLY
206

FIG. 2

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20140208100 A1, H. Richard Kendall **[0007]**

Non-patent literature cited in the description

- How to Import Serial Keys from CSV file - StoreApps. *Woo Commerce,* 21 April 2016 **[0008]**
- **MOHAMED SABT et al.** *Breaking into the KeyStore: A Practical Forgery Attack Against Android KeyStore* **[0010]**
- Trusted execution environment - Wikipedia, the free encyclopedia. 28 November 2016 **[0116]**
- Advanced Encryption Standard - Wikipedia, the free encyclopedia. 28 November 2016 **[0116]**