October 2023

A METHOD FOR FRICTIONLESS SIGNATURE-BASED HIGH-RISK PERMISSIONS MANAGEMENT ON COMPUTING DEVICES AND SYSTEM THEREOF

John Markh
VISA

“A METHOD FOR FRICTIONLESS SIGNATURE-BASED HIGH-RISK PERMISSIONS MANAGEMENT ON COMPUTING DEVICES AND SYSTEM THEREOF”

APPLICANT: VISA INTERNATIONAL SERVICE ASSOCIATION

ADDRESS: P.O. Box 8999, San Francisco, CA, 94128, USA

Nationality: USA

INVENTOR: JOHN MARKH

TECHNICAL FIELD

[0001] The present subject matter is, in general, related to the field of computer software and hardware security and particularly, but not exclusively to a system and method for optimizing the management of permissions for sensitive resources and features within computing devices by utilizing signature-based access control process to streamline and enhance the management of permissions.

BACKGROUND

[0002] In today's world, managing access to sensitive resources and features on computing devices remains a critical concern. The evolution of technology has only amplified the importance of ensuring that applications running on various operating systems can access these resources in a secure and user-friendly manner. Typically, these requests require user authorization. However, the authorization process can be complex and sometimes unclear to users. For instance, on Android devices, applications must declare security permissions in their manifest files to access specific components and features. Some permissions, categorized as "normal," are automatically granted by the operating system during installation without user approval. In contrast, "dangerous" permissions mandate explicit user consent. Various operating systems adopt distinct approaches, with some restricting access to specific features entirely, limiting their use to pre-installed applications.

[0003] Sensitive resources and functions encompass elements that can access private user data or exert control over the computing device, potentially compromising user privacy and security. For example, geo-location access can unveil a user's precise location, while NFC interface access could be exploited for unauthorized data extraction from contactless payment instruments. However, legitimate access to these sensitive resources is vital for providing essential user services, such as real-time navigation and secure contactless payments via digital wallets.

[0004] When users download applications from trusted digital distribution services like Google Play Store or Apple AppStore, they are prompted to grant access to sensitive resources during installation or when the application initially requires such access. Nevertheless, this user-centric authorization model has inherent drawbacks. Users may lack the requisite knowledge to assess security risks adequately, making them vulnerable to deceptive tactics like social engineering.

[0005] Implementing an outright restriction on access to sensitive resources raises concerns related to antitrust and competitiveness regulations and may hinder legitimate use cases, such as utilizing NFC interfaces for secure contactless payments. These challenges underscore the need for a more efficient and secure approach to manage permissions for sensitive resources and features on computing devices.

[0006] To overcome the above-mentioned shortcomings, the present invention introduces a system and method for frictionless signature-based high-risk permissions management on computing devices.

SUMMARY OF THE INVENTION

[0007] To overcome the above-mentioned shortcomings, the present disclosure provides an innovative approach to address the complexities surrounding access control to sensitive resources and features on computing devices. First and foremost, due-diligence review practices are integrated into the solution, emphasizing the importance of meticulous application evaluation before granting access to sensitive resources. This meticulous scrutiny helps ensure that applications adhere to established security standards, reducing the risk of malicious behavior.

[0008] Additionally, the solution harnesses the power of multiple certificates, aligning with advanced security measures like the APK Signature Scheme v3. This approach fortifies the integrity and authenticity of applications by utilizing multiple certificates for signing. By doing so, the method adds layers of verification, making it significantly more challenging for malicious actors to compromise an application's signature, thereby enhancing overall security.

[0009] Furthermore, the adoption of well-established key management practices is central to the solution. Robust key management is essential for safeguarding the storage and distribution of cryptographic keys, which are vital for maintaining the security of sensitive resources and features. By adhering to best practices in key management, the solution further bolsters the overall security of the system.

[0010] Crucially, the solution aims to provide frictionless access control to sensitive resources and features. While prioritizing security, it places a strong emphasis on minimizing user inconvenience during the authorization process. This user-centric approach ensures that individuals can easily and confidently manage resource access without becoming overwhelmed by complex decision-making, ultimately striking a balance between security and user convenience.

[0011] Thus, the present invention offers a comprehensive framework for managing access to sensitive resources and features, aligning with the evolving landscape of computing devices and operating systems. By combining due-diligence review practices, the use of multiple certificates, and robust key management with user-centric frictionless access control, the solution seeks to enhance security while providing a seamless and user-friendly experience within digital ecosystems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, explain the disclosed principles. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference features and components. Some embodiments of device or system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

[0013] FIG. 1 illustrates an environment for frictionless signature-based high-risk management on computing devices, in accordance with some embodiments of the present disclosure.

[0014] FIG. 2 shows a flowchart illustrating a method for frictionless signature-based high-risk management on computing devices, in accordance with some embodiments of the present disclosure.

[0015] FIG. 3 illustrates a detailed block diagram of a recommendation system in accordance with some embodiments of the present disclosure.

[0016] The figures depict embodiments of the disclosure for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the disclosure described herein.

DESCRIPTION OF THE DISCLOSURE

[0017] It is to be understood that the present disclosure may assume various alternative variations and step sequences, except where expressly specified to the contrary. It is also to be understood that the specific devices and processes illustrated in the attached drawings and described in the following specification are simply exemplary and non-limiting embodiments

or aspects. Hence, specific dimensions and other physical characteristics related to the embodiments or aspects disclosed herein are not to be considered as limiting.

[0018] In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

[0019] While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

[0020] The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

[0021] The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

[0022] The terms "including", "comprising", “having” and variations thereof mean "including but not limited to" unless expressly specified otherwise.

[0023] As used herein, the term “computing device” may refer to one or more electronic devices that are configured to directly or indirectly communicate with or over one or more networks. A computing device may be a mobile or portable computing device, a desktop computer, a server, and/or the like. Furthermore, the term “computer” may refer to any computing device that includes the necessary components to receive, process, and output data, and normally includes a display, a processor, a memory, an input device, and a network interface. A “computing system” may include one or more computing devices or computers. An “application” or “Application Program Interface” (API) refers to computer code or other data sorted on a computer-readable medium that may be executed by a processor to facilitate

the interaction between software components, such as a client-side front-end and/or server-side back-end for receiving data from the client.

[0024] The disclosed invention addresses the challenges related to access control for sensitive resources and features on computing devices. It is important to note that the terminology used aligns with the Android OpenSource Project (AOSP) while being adaptable to various computing platforms. The primary innovation centers around the introduction of a signature-based permission model, which is independent of specific operating systems and can be applied universally. Unlike conventional approaches that rely on user consent, the disclosed invention mandates that an application must be signed with a designated key by entities such as the OS vendor or Mobile Device Management (MDM) vendor to gain access to specific sensitive resources or features.

[0025] FIG. 1 illustrates an environment 100 for frictionless signature-based high-risk management on computing devices, in accordance with some embodiments of the present disclosure.

[0026] As shown in FIG. 1, a system for frictionless access control to sensitive resources and features may be implemented in an environment 100 comprising, without limiting to, a software vendor 102, an operating system (OS) vendor or a mobile device management (MDM) vendor 104, and a recommendation system 106.

[0027] The first entity in this environment 100 is the software vendor 102. This entity represents software vendors or developers who create and publish applications for computing devices. They play an important role in the access control process by declaring the required permissions for their applications and signing them with the necessary keys.

[0028] The second critical entity is the Operating System (OS) vendor or Mobile Device Management (MDM) vendor 104. This entity is responsible for the management of the computing device's operating system and security policies. They are central to the implementation of the signature-based access control system, ensuring that applications are authorized to access sensitive resources and features.

[0029] The third element within this environment is the recommendation system 106. The recommendation system 106, potentially hosted on a computing device, serves as a pivotal component. It is configured to facilitate frictionless access control to sensitive resources and features on the computing device. This is achieved through a combination of due-diligence review practices, utilization of multiple certificates for application signing, and adoption of established key management practices.

[0030] Within this environment, the recommendation system 106 acts as an orchestrator, bringing together the necessary processes and practices to ensure secure and efficient access control to sensitive resources and features on computing devices. It collaborates closely with software vendors (102) and the OS or MDM vendor (104) to establish a robust system for managing high-risk permissions through the use of certificates and due-diligence reviews.

[0031] The cooperation and interaction among these entities within this environment are essential to achieve the overarching goal of frictionless, secure, and efficient management of access to sensitive resources and features on computing devices, as described in the disclosed embodiments. This system aims to enhance the security and usability of computing devices while streamlining the authorization process for applications.

[0032] The recommendation system 106 mandates that an application must be signed with a designated key by entities such as the OS vendor or Mobile Device Management (MDM) vendor 104 to gain access to specific sensitive resources or features. By way of example, to use ACCESS_FINE_LOCATION permission on Android distributed through Google Play Store, the application will have to be signed by Google with an appropriate certificate (e.g., certificate A) that will be recognized and trusted by the Android OS. To be allowed BIND_NFC_SERVICE permission, the application will have to be signed by another appropriate certificate (e.g., certificate B) that will be recognized by the Android OS as allowed to use the NFC feature on the device. The approval to access sensitive resources and features can be further supplemented by the device user.

[0033] In one implementation, to enable the signature-based authorization model, the OS or MDM vendors 104 may be required to maintain a Key Management System (KMS). Further, the KMS may store certificates used for application signing, ensuring their trustworthiness. In one non-limiting embodiment, the essential certificates may include Root certificate A_0: that is generated and signed by the OS vendor, serving as a trusted Root Certificate Authority (Root CA) on the target computing device. Further, the certificates may include “app signing” certificate B_1 and “upload” certificate B_2 which are issued exclusively to a single software vendor after the OS vendor conducts a due-diligence process and are also signed by a Root CA. Moreover, for each sensitive permission, the KMS generates a permission-specific certificate correlated to a particular sensitive resource or function, all signed by the Root CA, resulting in a set of certificates $C_1 - C_n$.

[0034] In the event of the loss or compromise of private certificates such as the root certificate A_0, "app signing" certificate B_1, "upload" certificate B_2, or permission-specific keys $C_1 - C_n$,

replacement procedures adhere to established industry best practices, such as those outlined in NIST Special Publication SP 800-57.

[0035] FIG. 2 shows a flowchart 200 illustrating a method for frictionless signature-based high-risk management on computing devices, in accordance with some embodiments of the present disclosure. **FIG. 2** is explained in conjunction with **FIG. 1**.

[0036] At step 202, the method may include onboarding of a new software vendor by the OS vendor or MDM vendor 104. During onboarding, the OS or MDM vendor 104 may implement due diligence processes not only to confirm that the software vendor 104 is a legitimate entity, but also to confirm that it has business and technical justification to access privileged and sensitive resources and features on the end-point computer device. For example, a fintech software developer may request access to READ_PRIVILEGED_PHONE_STATE permission to access mobile phone IMI and BIND_NFC_SERVICE permissions with justification that it is required to uniquely identify the consumer and to support contactless payment functionality.

[0037] The onboarding process may include creating an account (e.g., publish account) with the OS or MDM vendor 104. Based upon successfully creation of the account, the OS or MDM vendor 104 may generate the "upload key" B_2, that is conveyed securely to the software vendor 102. The software vendor may then securely retain and protect by the "upload key" B_2. Further, the OS or MDM vendor 104 may generate the "app signing key" B_1 that is securely retained by the OS or MDM vendor 104 in their key management system (KMS). As a part of sign-up process (i.e., account creation), the OS or MDM vector 104 may perform due-diligence process where the software vendor 102 may indicate the expected sensitive resources and features that may be required by their application(s). For each required access to sensitive resources and features, the software vendor 102 may provide documented justification for why the access to a sensitive resource or feature is required (e.g., real-time navigation for ACCESS_FINE_LOCATION permission, payment application for BIND_NFC_SERVICE permission or banking application for READ_PRIVILEGED_PHONE_STATE privileged permission). The OS or MDM vendor 104 may perform due-diligence process to confirm the software-vendor 102 is a legitimate entity, with a legitimate need to access the requested sensitive resource or feature. Also, the OS or MDM vendor 104 may request the software vendor to provide additional documentation to justify and substantiate the requested access to a sensitive resource or feature,

[0038] At step 204, the method may include rreleasing or publishing the application to a digital distribution service managed by the OS or MDM vendor 104. When a software vendor intends

to release or publish an application requiring access to sensitive resources and features, they must declare the necessary permissions and sign the application using the "upload key." This signed application is then uploaded to a digital distribution service, such as Google Play Store or Apple AppStore. The OS or MDM vendor, upon receiving the application, carries out validation procedures, including confirming the signature's correspondence to the software vendor, checking platform-specific signature rules (e.g., APK v3 signature block), and verifying if the software vendor has received pre-clearance for the declared resource access. If these verifications are successful, the OS or MDM vendor proceeds to sign the application with permission-specific certificates ($C_1 - C_n$) and the software vendor's "app signing key" (B_1). Subsequently, the application is ready for distribution.

[0039] At step 206, the method may include downloading and installing the application on the target end-point computing device. When an end-user downloads the application from a digital distribution service, the operating system on the computing device undertakes several steps. It first verifies the application's signature block, such as APK Signature Scheme 3 verification. If the initial verification is successful, the application is installed with "normal" or "low-risk" permissions without requiring explicit user consent. Additionally, for each requested access to sensitive resources and features, the application is verified to be signed with the corresponding "permission" key (C_i), and "dangerous" or "high-risk" permissions are assigned without explicit user intervention.

[0040] The disclosed invention aims to facilitate a frictionless assignment of application permissions, it also emphasizes the importance of maintaining an audit trail for permission assignments, allowing users the option to review these permissions both before and after application installation. Overall, the key contributions of this solution include enabling frictionless assignment of high-risk permissions for pre-cleared software vendors and introducing a signature-based access control mechanism to regulate access to sensitive resources and features on computing devices.

General computing system:

[0041] FIG. 3 illustrates a detailed block diagram of a computing system in accordance with some embodiments of the present disclosure.

[0042] In an embodiment, FIG. 3 illustrates a block diagram of an exemplary computer system 300 that may be used to implement the recommendation system 106. In some embodiments, the computer system 300 is used to operate the recommendation system 106 for generating

recommendations for suggesting the best location(s) and distribution channel(s) to a new business in accordance with some embodiments of the present disclosure. In some embodiments, the computer system 300 may include a central processing unit ("CPU" or "processor") 302. The processor 302 may include at least one data processor for executing processes in Virtual Storage Area Network. The processor 302 may include at least one data processor for executing program components for executing user (e. g. a new merchant) or system-generated recommendation processes. A user (e. g. a new merchant) may include a person such as a new merchant or seller, a person using the user device 310 for receiving new business recommendations. The processor 302 may include specialized processing units such as integrated system (bus) controllers, memory management control units, floating point units, graphics processing units, digital signal processing units, etc.

[0043] The processor 302 may be disposed in communication with one or more Input/Output (I/O) devices (312 and 313) via I/O interface 301. The I/O interface 301 employ communication protocols/methods such as, without limitation, audio, analog, digital, monoaural, Radio Corporation of America (RCA) connector, stereo, IEEE-1394 high speed serial bus, serial bus, Universal Serial Bus (USB), infrared, Personal System/2 (PS/2) port, Bbayonet Neill-Concelman (BNC) connector, coaxial, component, composite, Digital Visual Interface (DVI), High-Definition Multimedia Interface (HDMI), Radio Frequency (RF) antennas, S-Video, Video Graphics Array (VGA), IEEE 802.11b/g/n/x, Bluetooth, cellular e.g., Code-Division Multiple Access (CDMA), High-Speed Packet Access (HSPA+), Global System for Mobile communications (GSM), Long-Term Evolution (LTE), Worldwide Interoperability for Microwave access (WiMax), or the like, etc.

[0044] Using the I/O interface 301, the computer system 300 may communicate with one or more I/O devices such as input devices 312 and output devices 313. For example, the input devices 312 may be an antenna, keyboard, mouse, joystick, (infrared) remote control, camera, card reader, fax machine, dongle, biometric reader, microphone, touch screen, touchpad, trackball, stylus, scanner, storage device, transceiver, video device/source, etc. The output devices 313 may be a printer, fax machine, video display (e.g., Cathode Ray Tube (CRT), Liquid Crystal Display (LCD), Light-Emitting Diode (LED), plasma, Plasma Display Panel (PDP), Organic Light-Emitting Diode display (OLED) or the like), audio speaker, etc.

[0045] In some embodiments, the processor 302 may be disposed in communication with a communication network 309 via a network interface 303. The network interface 303 may communicate with the communication network 309. The network interface 303 may employ connection protocols including, without limitation, direct connect, ethernet (e.g., twisted pair

10/100/1000 Base T), Transmission Control Protocol/Internet Protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc. The communication network 309 may include, without limitation, a direct interconnection, Local Area Network (LAN), Wide Area Network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, etc. Using the network interface 303 and the communication network 309, the computer system 300 may communicate with a database 314, which may be the enrolled templates database 313. The network interface 303 may employ connection protocols include, but not limited to, direct connect, ethernet (e.g., twisted pair 10/100/1000 Base T), Transmission Control Protocol/Internet Protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc.

[0046] The communication network 309 includes, but is not limited to, a direct interconnection, a Peer-to-Peer (P2P) network, Local Area Network (LAN), Wide Area Network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, Wi-Fi and such. The communication network 309 may either be a dedicated network or a shared network, which represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP), etc., to communicate with each other. Further, communication network 309 may include a variety of network devices, including routers, bridges, servers, computing devices, storage devices, etc.

[0047] In some embodiments, the processor 302 may be disposed of in communication with a memory 305 (e.g., RAM, ROM, etc. not shown in Fig. 3) via a storage interface 304. The storage interface 304 may connect to memory 305 including, without limitation, memory drives, removable disc drives, etc., employing connection protocols such as, Serial Advanced Technology Attachment (SATA), Integrated Drive Electronics (IDE), IEEE-1394, Universal Serial Bus (USB), fiber channel, Small Computer Systems Interface (SCSI), etc. The memory drives may further include a drum, magnetic disc drive, magneto-optical drive, optical drive, Redundant Array of Independent Discs (RAID), solid-state memory devices, solid-state drives, etc. Memory 305 may store a collection of program or database components, including, without limitation, user interface 306, an operating system 307, a web browser 308 etc. In some embodiments, computer system 300 may store user/application data, such as, the data, variables, records, etc., as described in this disclosure. Such databases may be implemented as fault-tolerant, relational, scalable, secure databases such as Oracle or Sybase.

[0048] The operating system 307 may facilitate resource management and operation of the computer system 300. Examples of operating systems include, without limitation, Apple™ Macintosh™ OS X, UNIX™, Unix-like system distributions (e.g., Berkeley Software

Distribution (BSD), FreeBSD™, Net BSD™, Open BSD™, etc.), Linux distributions (e.g., Red Hat™, Ubuntu™, K-Ubuntu™, etc.), International Business Machines (IBM™) OS/2™, Microsoft Windows™ (XP™, Vista/7/8, etc.), Apple iOS™, Google Android™, Blackberry™ operating system (OS), or the like. The User interface 306 may facilitate display, execution, interaction, manipulation, or operation of program components through textual or graphical facilities. For example, user interfaces may provide computer interaction interface elements on a display system operatively connected to the computer system 300, such as cursors, icons, checkboxes, menus, scrollers, windows, widgets, etc. Graphical User Interfaces (GUIs) may be employed, including, without limitation, Apple® Macintosh® operating systems' Aqua®, IBM® OS/2®, Microsoft® Windows® (e.g., Aero, Metro, etc.), web interface libraries (e.g., ActiveX®, Java®, Javascript, AJAX, HTML, Adobe® Flash®, etc.), or the like.

[0049] In some embodiments, the computer system 300 may implement web browser 308 stored program components. Web browser 308 may be a hypertext viewing application, such as Microsoft™ Internet Explorer™, Google Chrome™, Mozilla Firefox™, Apple™ Safari™, etc. Secure web browsing may be provided using secure hypertext transport protocol (HTTPS), Secure Sockets Layer (SSL), Transport Layer Security (TLS), etc. Web browsers 308 may utilize facilities such as AJAX, DHTML, Adobe™ Flash, Javascript, Application Programming Interfaces (APIs), etc. In some embodiments, the computer system 300 may implement a mail server stored program component. The mail server may be an Internet mail server such as Microsoft Exchange, or the like. The mail server may utilize facilities such as ASP, ActiveX, ANSI C++/C#, Microsoft .NET, Common Gateway Interface (CGI) scripts, Java, JavaScript, PERL, PHP, Python, WebObjects, etc. The mail server may utilize communication protocols such as Internet Message Access Protocol (IMAP), Messaging Application Programming Interface (MAPI), Microsoft Exchange, Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), or the like.

[0050] In some embodiments, the computer system 300 may implement a mail client stored program component. The mail client may be a mail viewing application, such as APPLE® MAIL, MICROSOFT® ENTOURAGE®, MICROSOFT® OUTLOOK®, MOZILLA® THUNDERBIRD®, etc.

[0051] Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer-readable storage medium may store instructions for execution by one or more processors, including instructions for causing the

processor(s) to perform steps or stages consistent with the embodiments described herein. The term “computer-readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., be non-transitory. Examples include Random Access Memory (RAM), Read-Only Memory (ROM), volatile memory, non-volatile memory, hard drives, Compact Disc (CD) ROMs, DVDs, flash drives, disks, and any other known physical storage media.

[0052] The described operations may be implemented as a method, system or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The described operations may be implemented as code maintained in a “non-transitory computer readable medium”, where a processor may read and execute the code from the computer readable medium. The processor is at least one of a microprocessor and a processor capable of processing and executing the queries. A non-transitory computer readable medium may include media such as magnetic storage medium (e.g., hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, DVDs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, Flash Memory, firmware, programmable logic, etc.), etc. Further, non-transitory computer-readable media may include all computer-readable media except for transitory. The code implementing the described operations may further be implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.).

[0053] The illustrated steps are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments. Also, the words "comprising," "having," "containing," and "including," and other similar forms are intended to be equivalent in meaning and be open ended in that an item or items following any one of these words is not meant to be an exhaustive listing of such item or items or meant to be limited to only the listed item or items. It must also be noted that as

used herein, the singular forms “a,” “an,” and “the” include plural references unless the context clearly dictates otherwise.

[0054] Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term “computer readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., are non-transitory. Examples include Random Access Memory (RAM), Read-Only Memory (ROM), volatile memory, non-volatile memory, hard drives, CD ROMs, DVDs, flash drives, disks, and any other known physical storage media.

[0055] Finally, the language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the embodiments of the disclosure is intended to be illustrative, but not limiting, of the scope of the disclosure.

[0056] With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for the sake of clarity.

“A METHOD FOR FRICTIONLESS SIGNATURE-BASED HIGH-RISK PERMISSIONS MANAGEMENT ON COMPUTING DEVICES AND SYSTEM THEREOF”

ABSTRACT

The present disclosure provides a method for frictionless signature-based high-risk permissions management on computing devices. The method includes an onboarding process involving the inclusion of new software vendors by the OS or MDM vendor, a process that verifies vendor legitimacy and the necessity of their access to sensitive resources. During onboarding, the OS or MDM vendor generates an "upload key," securely provided to the software vendor, and an "app signing key," securely retained within the vendor's Key Management System (KMS). Further, the method includes releasing or publishing applications to a digital distribution service managed by the OS or MDM vendor involves software vendors declaring necessary permissions and signing applications with the "upload key" before uploading them to distribution services. The OS or MDM vendor validates the application, ensuring vendor correspondence, adherence to platform-specific signature rules, and pre-clearance for resource access. Upon successful verification, the application is signed with permission-specific certificates and the "app signing key," preparing it for distribution. Furthermore, the method includes downloading and installation of applications on end-point computing devices encompass the OS verification process. If successful, the application is installed with "normal" or "low-risk" permissions without user consent. Moreover, for each requested sensitive resource and feature access, the application is verified with the corresponding "permission" key, granting "dangerous" or "high-risk" permissions seamlessly.

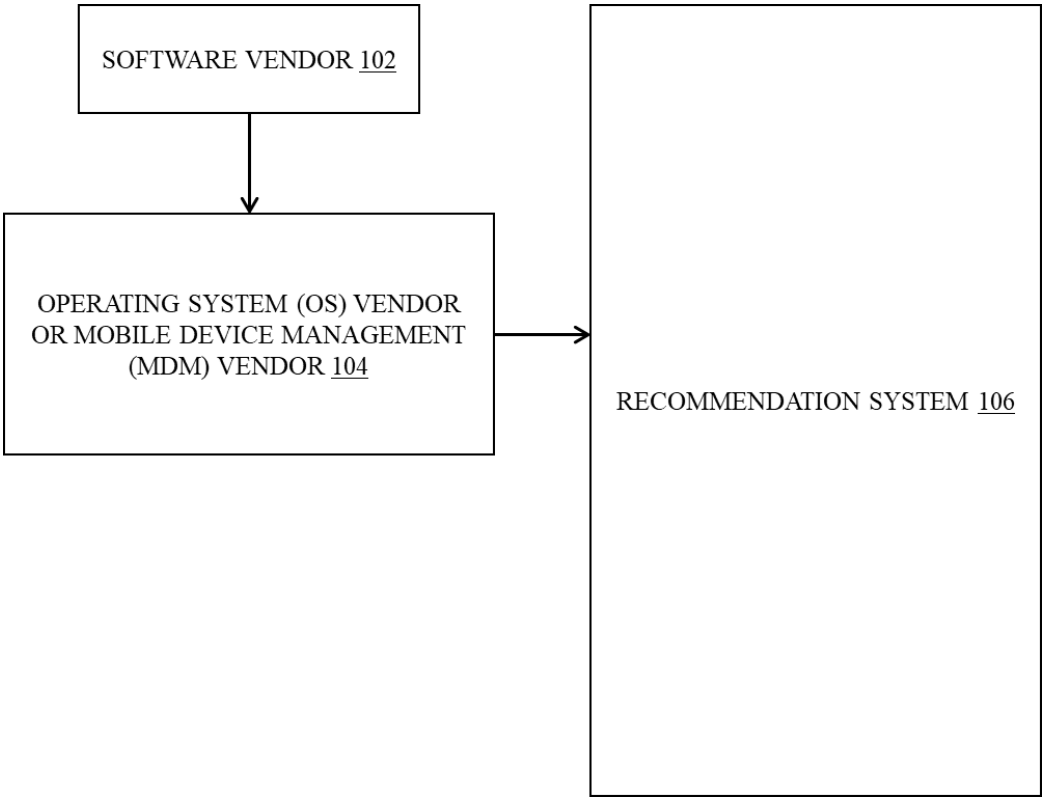
SOFTWARE VENDOR 102
OPERATING SYSTEM (OS) VENDOR OR MOBILE DEVICE MANAGEMENT (MDM) VENDOR 104
RECOMMENDATION SYSTEM 106

FIG. 1

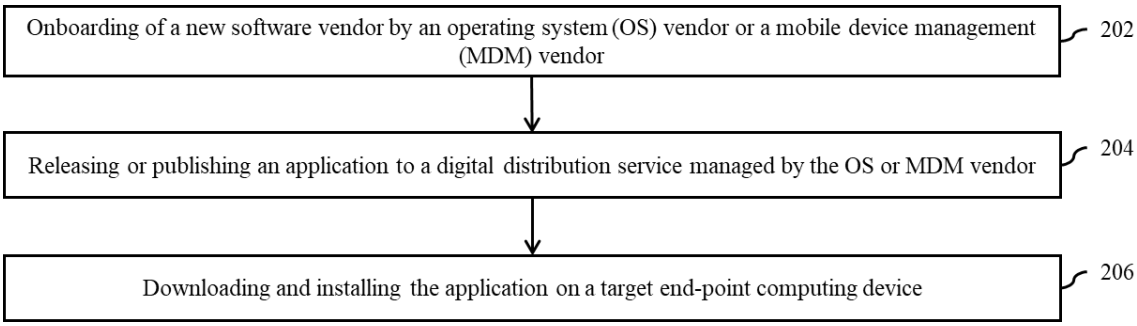
Onboarding of a new software vendor by an operating system (OS) vendor or a mobile device management (MDM) vendor
202
Releasing or publishing an application to a digital distribution service managed by the OS or MDM vendor
204
Downloading and installing the application on a target end-point computing device
206

FIG. 2

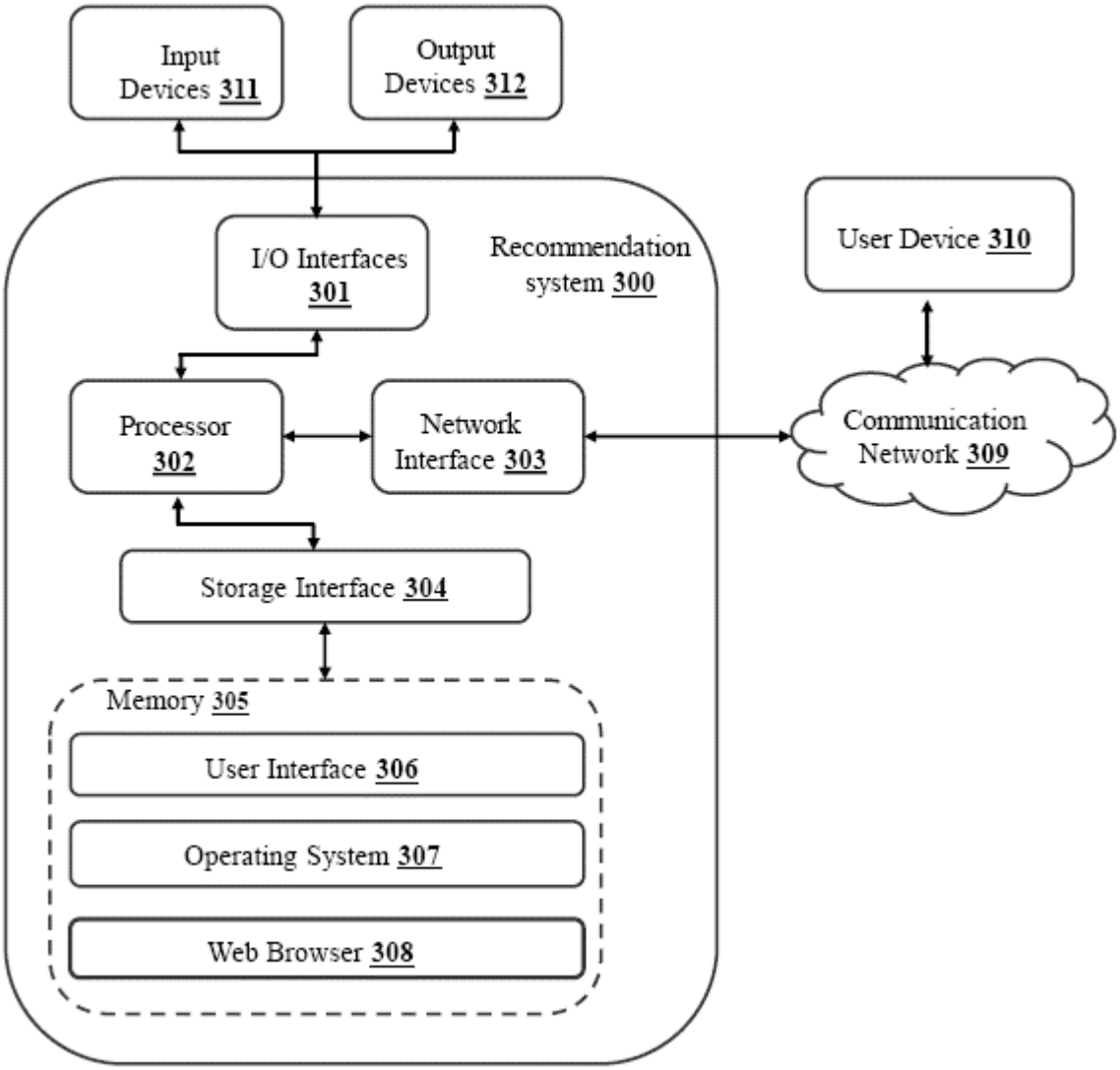
Input Devices 311
Output Devices 312
I/O Interfaces 301
Recommendation system 300
User Device 310
Processor 302
Network Interface 303
Communication Network 309
Storage Interface 304
Memory 305
User Interface 306
Operating System 307
Web Browser 308

FIG. 3