US010171427B2

(12) **United States Patent**
**Dawson**

(10) **Patent No.: US 10,171,427 B2**
(45) **Date of Patent: *Jan. 1, 2019**

(54) **PORTABLE ENCRYPTION AND AUTHENTICATION SERVICE MODULE**

(71) Applicant: **WebCloak, LLC**, Irvine, CA (US)

(72) Inventor: **Martin Dawson**, Laguna Beach, CA (US)

(73) Assignee: **WEBCLOAK, LLC**, Irvine, CA (US)

( * ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 159 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **15/011,068**

(22) Filed: **Jan. 29, 2016**

(65) **Prior Publication Data**

US 2016/0226833 A1 Aug. 4, 2016

**Related U.S. Application Data**

(60) Provisional application No. 62/109,523, filed on Jan. 29, 2015.

(51) **Int. Cl.**

| | |
|---|---|
| ***H04L 29/06*** | (2006.01) |
| ***H04W 12/06*** | (2009.01) |
| *H04L 29/08* | (2006.01) |

(52) **U.S. Cl.**

CPC ..... ***H04L 63/0428*** (2013.01); ***H04L 63/0853*** (2013.01); ***H04W 12/06*** (2013.01); *H04L 69/162* (2013.01); *H04L 69/321* (2013.01)

(58) **Field of Classification Search**

CPC ..... G06F 21/34; H04L 63/0853

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 8,151,262 | B2 | 4/2012 | Challener et al. | |
| 8,522,018 | B2 | 8/2013 | Molina et al. | |
| 9,053,059 | B2 | 6/2015 | Scott-Nash | |
| 2001/0049718 | A1* | 12/2001 | Ozawa | H04N 7/163 709/203 |
| 2003/0218629 | A1* | 11/2003 | Terashima | G06F 17/30905 715/738 |
| 2006/0282678 | A1* | 12/2006 | Ali | G06F 21/34 713/185 |
| 2008/0256536 | A1* | 10/2008 | Zhao | G06F 9/45537 718/1 |
| 2008/0281798 | A1* | 11/2008 | Chatterjee | G06F 17/30902 |

(Continued)

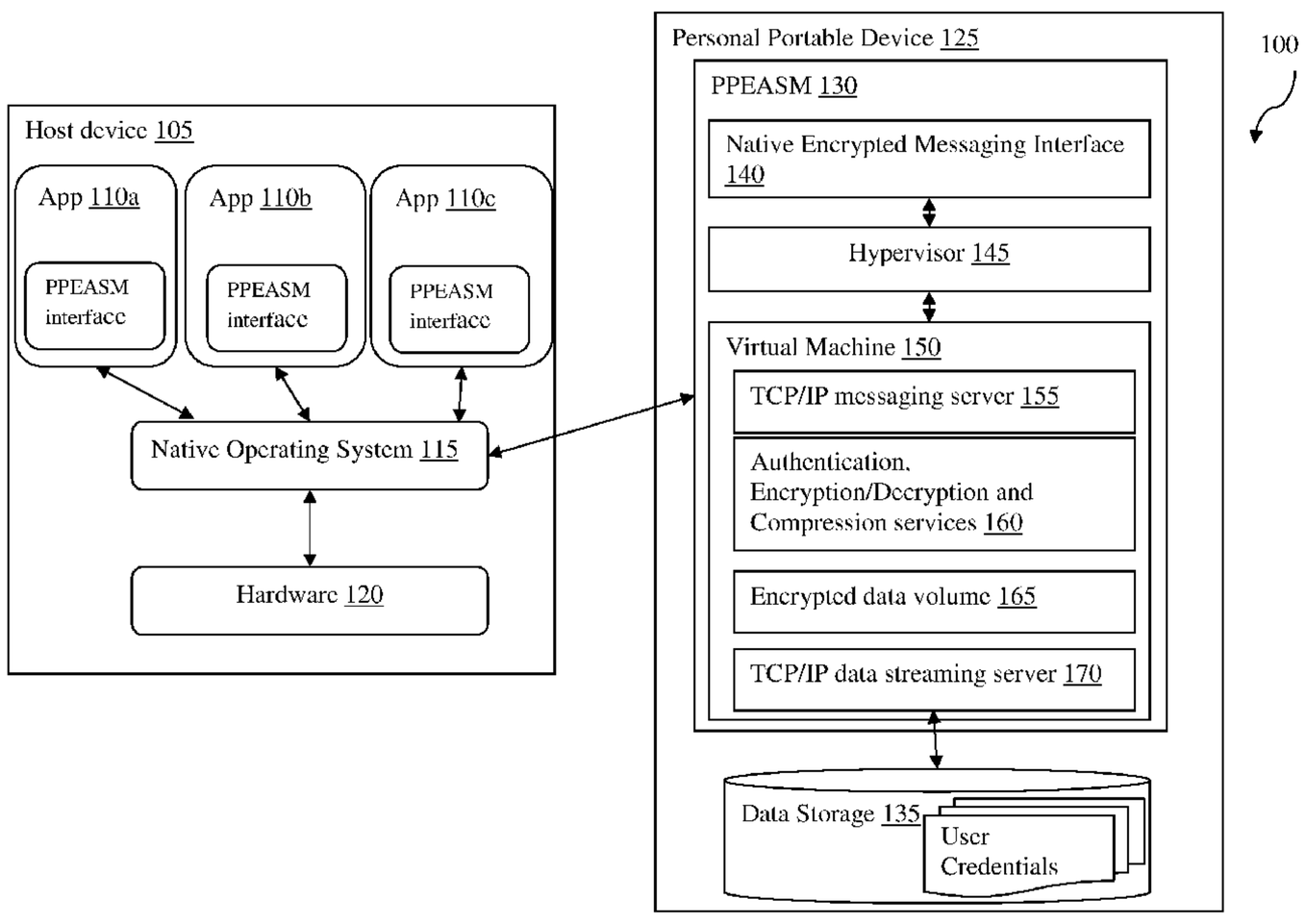
*Primary Examiner* — David J Pearson

(74) *Attorney, Agent, or Firm* — Fish IP Law, LLP

(57) **ABSTRACT**

Portable, hand-held electronic devices and methods to allow a user to anonymously utilize a host device are presented. The host device includes a processor to communicate with an application having a target network address. The portable, hand-held electronic devices includes an onboard database that stores user credential information and a portable encryption and authentication service module (PPEASM) that allows to make a secure communication channel with the host device. The PPEASM configures the processor of the host device to instantiate a virtual machine and render an encrypted messaging interface for communicating between the virtual machine and the application in the host device. Then, PPEASM can also configure the processor to negotiate authentication of the user with the application by utilizing the user credential information and information received through the encrypted messaging interface.

**21 Claims, 4 Drawing Sheets**



(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 2008/0307409 | A1* | 12/2008 | Lu | G06F 9/45537 717/174 |
| 2009/0132816 | A1 | 5/2009 | Lee | |
| 2011/0016382 | A1* | 1/2011 | Cahill | G06F 17/30896 715/234 |
| 2011/0246778 | A1 | 10/2011 | Duane | |
| 2011/0264770 | A1* | 10/2011 | Kim | G06F 17/30893 709/219 |
| 2013/0173759 | A1* | 7/2013 | Herse | G06F 21/34 709/219 |
| 2014/0025726 | A1* | 1/2014 | Chen | G06F 9/54 709/203 |
| 2014/0357227 | A1 | 12/2014 | Lee | |
| 2015/0074764 | A1 | 3/2015 | Stern | |
| 2015/0358328 | A1* | 12/2015 | Kaplan | H04L 67/1095 726/6 |

* cited by examiner

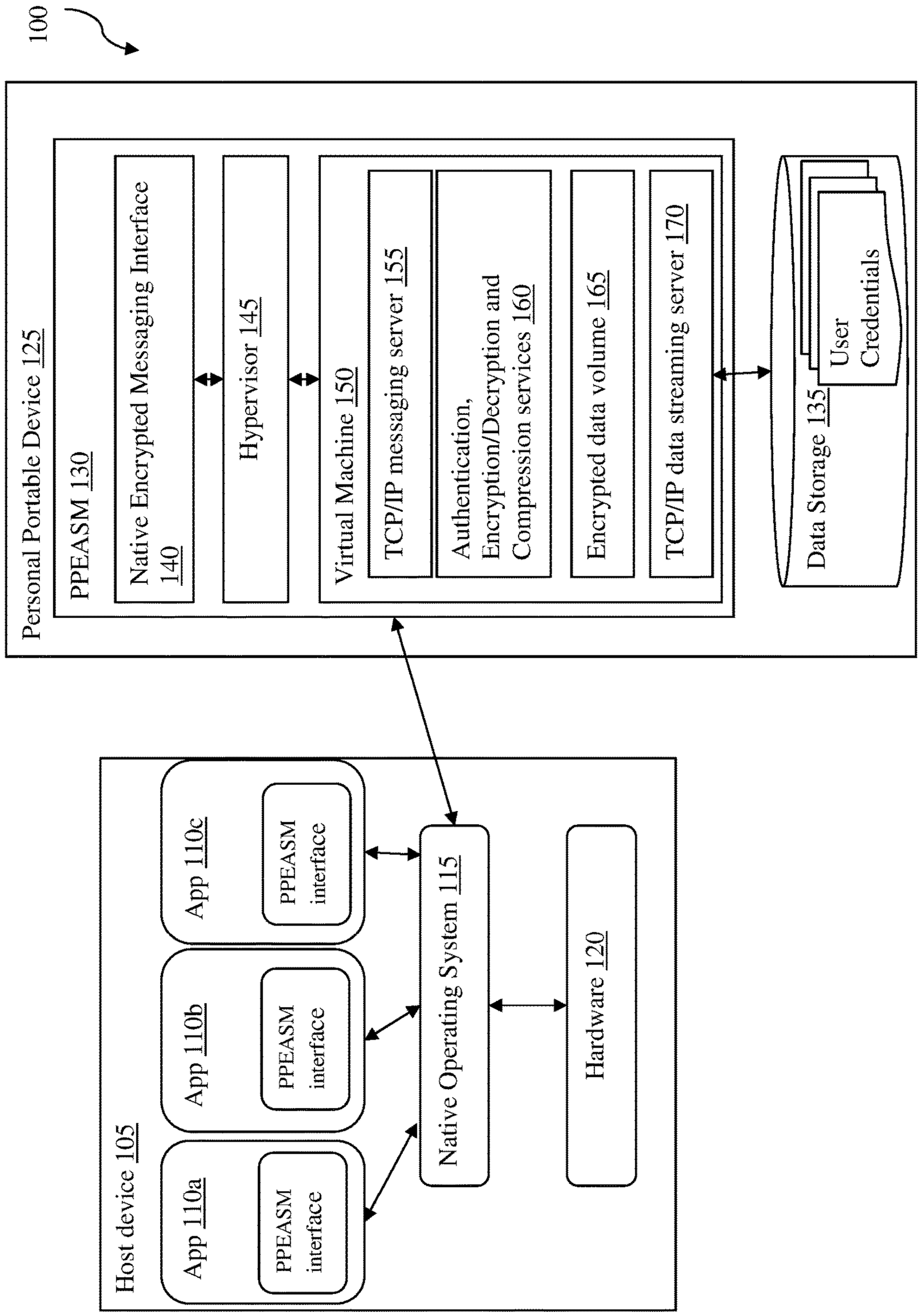

100
Personal Portable Device 125
PPEASM 130
Native Encrypted Messaging Interface 140
Hypervisor 145
Virtual Machine 150
TCP/IP messaging server 155
Authentication, Encryption/Decryption and Compression services 160
Encrypted data volume 165
TCP/IP data streaming server 170
Data Storage 135
User Credentials
Host device 105
App 110a
PPEASM interface
App 110b
PPEASM interface
App 110c
PPEASM interface
Native Operating System 115
Hardware 120

Figure 1

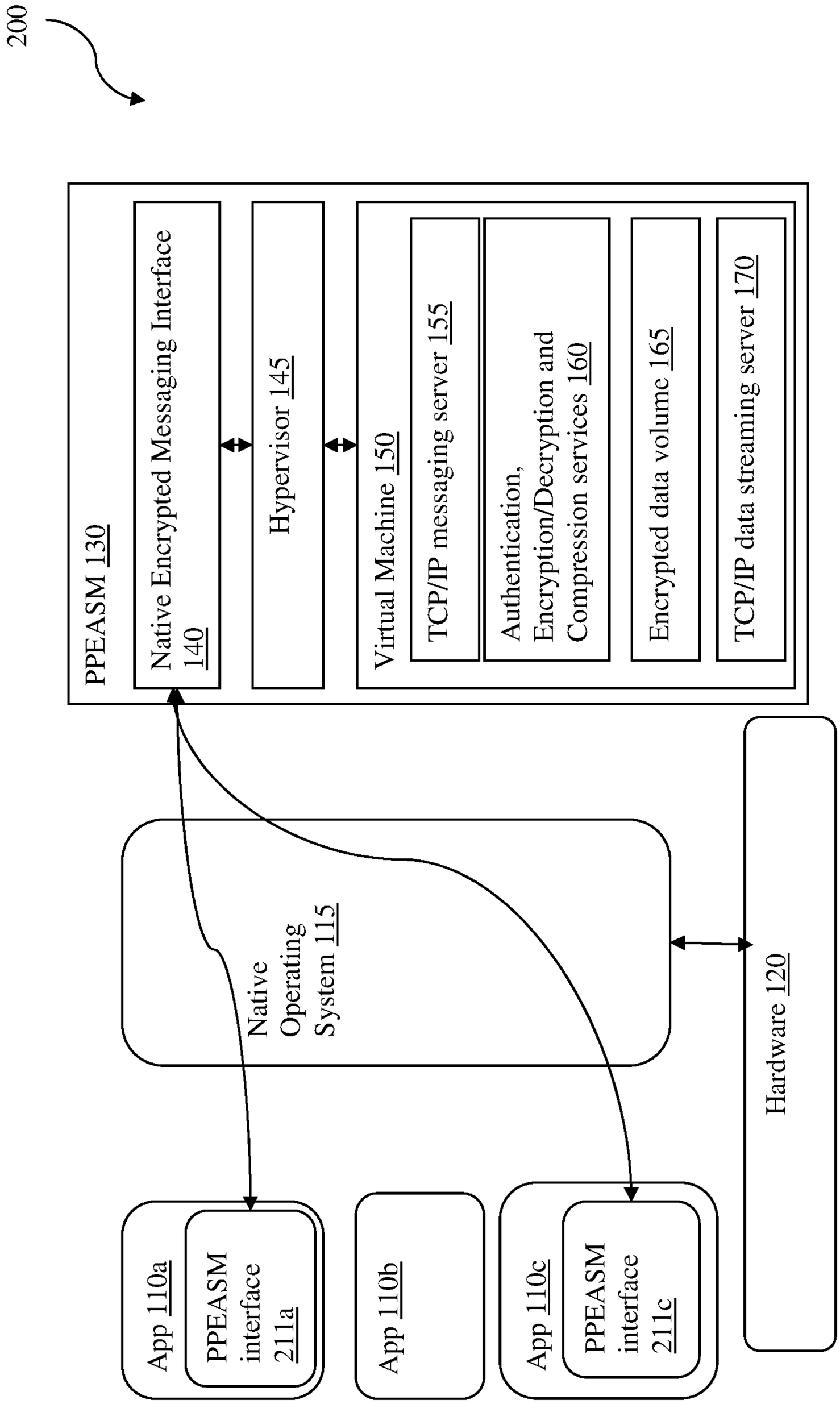

200
PPEASM 130
Native Encrypted Messaging Interface 140
Hypervisor 145
Virtual Machine 150
TCP/IP messaging server 155
Authentication, Encryption/Decryption and Compression services 160
Encrypted data volume 165
TCP/IP data streaming server 170
Native Operating System 115
Hardware 120
App 110a
PPEASM interface 211a
App 110b
App 110c
PPEASM interface 211c

Figure 2

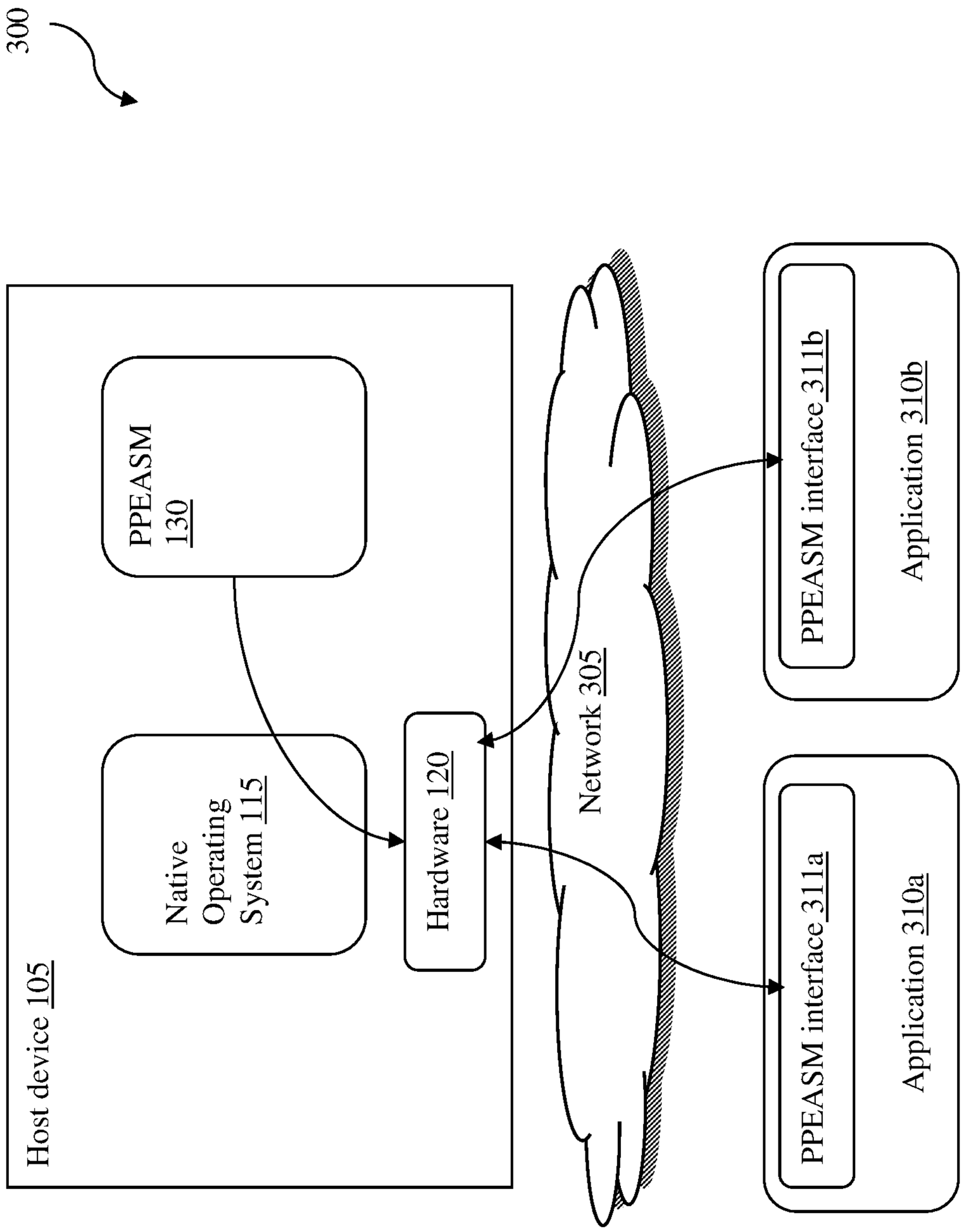

300
Host device 105
Native Operating System 115
PPEASM 130
Hardware 120
Network 305
PPEASM interface 311a
Application 310a
PPEASM interface 311b
Application 310b

Figure 3

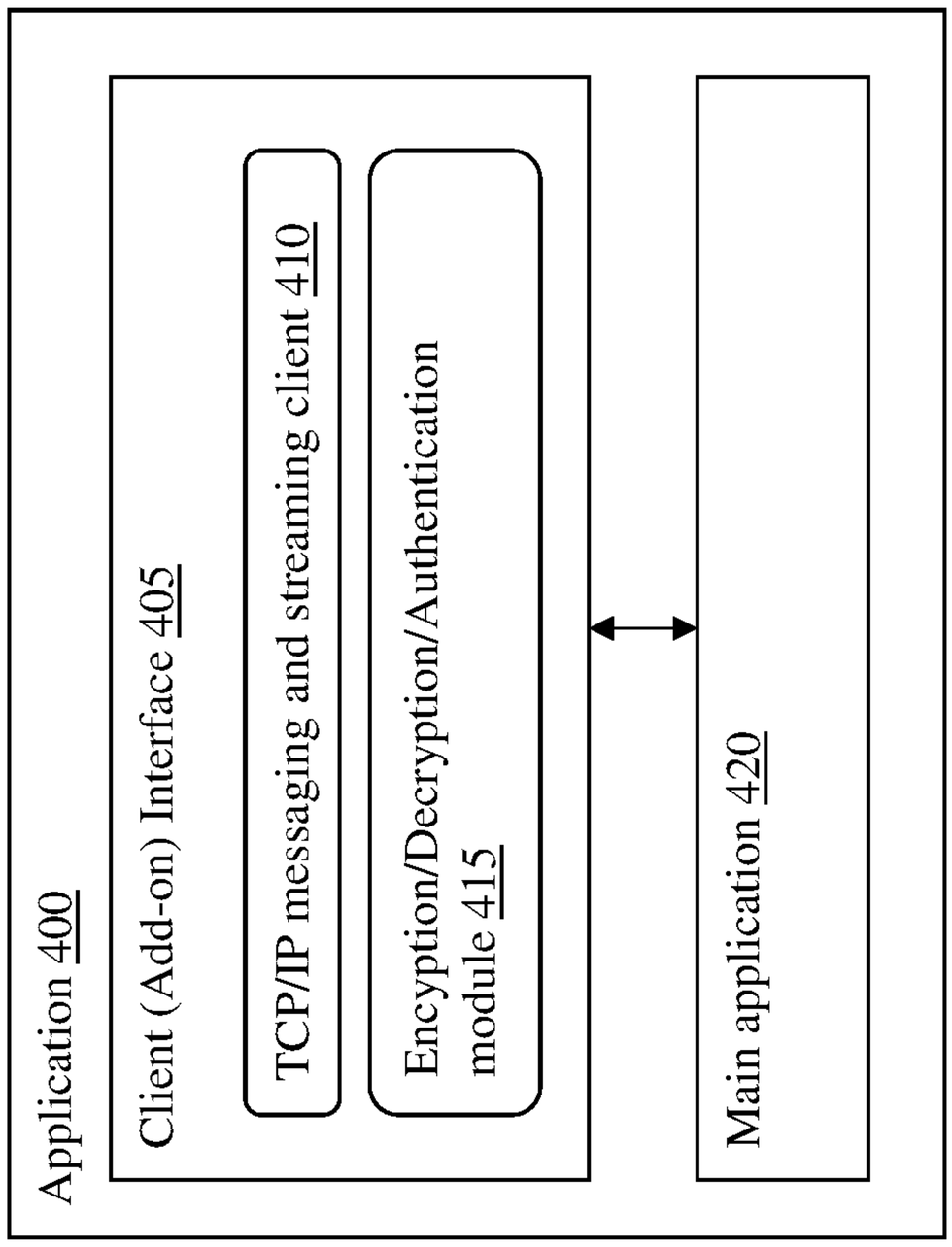

Application 400
Client (Add-on) Interface 405
TCP/IP messaging and streaming client 410
Encyption/Decryption/Authentication module 415
Main application 420

Figure 4

# PORTABLE ENCRYPTION AND AUTHENTICATION SERVICE MODULE

This application claims priority to our U.S. provisional patent application with the Ser. No. 62/109,523 filed Jan. 29, 2015 which is incorporated by reference herein in its entirety.

## FIELD OF THE INVENTION

The field of the invention is secured authentication and communication.

## BACKGROUND

The following description includes information that may be useful in understanding the present invention. It is not an admission that any of the information provided herein is prior art or relevant to the presently claimed invention, or that any publication specifically or implicitly referenced is prior art.

Cloud computing and storage solutions enable users to store and process their data in third-party data centers, which allows easy access and sharing of resources, data, and information among computers and other mobile devices. Generally, Authentication as a service (AaaS) and Encryption as a service (EaaS) are used to provide cloud-based storage of user credentials and access information to facilitate system authentication from a single repository as well as the encryption of data. However, the cloud based authentication and encryption services are volatile to the theft of root keys stored in the cloud-based storage. For example, if the cloud-based storage of service providers is compromised, then individual user's data and passwords can also be compromised.

Many technologies have been developed to provide safer encryption and authentication services in accessing host computer's application or database. For example, U.S. Pat. No. 8,522,018 to Molina discloses a portable or mobile Trusted Platform Module (TPM) based on the specification from the Trusted Computing Group (TCG) that is used to authenticate and help to maintain the security of a system and to provide a computer system with encryption capabilities. However, Molina's system is limited to authentication and encryption in a mobile or virtual environment, and cannot be used as a portable encryption and authentication agent.

Others have sought to solve the problem by providing a portable device that contains security information. For example, US Patent Application Number 2011/0246778 to Duane discloses a USB "Key" that contains information to generate a checksum to validate a virtual machine (VM) host image. When utilizing the VM image, the host system validates the image using the USB "Key". Thus, if the USB drive were missing or if the validation failed, the host system would refuse to load the VM image. For another example, US Patent Application Number 2015/0074764 to Stern discloses a system that allows a portable device running VM to request an authorized access of the VM to the portable devices hardware or other resources via an authenticating server. For example, if a VM running on a mobile phone attempts to access the camera, the VM creates a secure connection to the server by exchanging a key and asks permission. However, these systems does not use VM and secure communications in order to provide encryption and authentication services.



All publications identified herein are incorporated by reference to the same extent as if each individual publication or patent application were specifically and individually indicated to be incorporated by reference. Where a definition or use of a term in an incorporated reference is inconsistent or contrary to the definition of that term provided herein, the definition of that term provided herein applies and the definition of that term in the reference does not apply.

Thus, there is still a need for a portable system and method to provide encryption and authentication services in a remote or guest environment using the users' credentials.

## SUMMARY OF THE INVENTION

The inventive subject matter provides encryption and authentication services in a remote or guest environment using the users' credentials in a portable device and/or application without using the credentials of the remote service provider or host machine.

One aspect of the invention relates a portable, hand-held electronic device. The hand-held electronic device is configured to allow a user to anonymously utilize a host device. The host device includes a processor to communicate with an application having a target network address, and a native operating system (OS) to run applications. The portable device includes an onboard database and an onboard memory. The onboard database stores user credential information, which can include a password, a challenge phrase, or a challenge phrase hash. In some embodiments, the onboard database further includes a static read-only data volume for certificate storage and a read/write data volume for runtime work.

The onboard memory stores software instructions. When executed by the processor, the instructions cause the processor to instantiate a virtual machine that runs on top of the native OS. The virtual machine has a Transmission Control Protocol/Internet Protocol (TCP/IP) messaging server having an IP address different from any IP address of the host device. The instructions further cause the processor to render an encrypted messaging interface that utilizes the TCP/IP messaging server for communication between the virtual machine and the application over a TCP/IP networking layer, and negotiate authentication of the user with the application by utilizing the user credential information and information received through the encrypted messaging interface. Preferably, the virtual machine comprises a secure Linux Kernel.

In some embodiments, the application communicating with the virtual machine is configured to run on top of the native OS. In some other embodiments, it is also contemplated that the application runs on a remote device, which is communicatively coupled to the host device over a network.

In a preferred embodiment, software instructions cause the processor to establish a secured communication channel between the virtual machine and the application over the TCP/IP networking layer. Then the software instructions can also cause the processor to send the user credential information to the application through the secured communication channel. In some embodiments, the processor can encrypt the user credential information before sending the encrypted user credential to the target address. Further, the processor can also decrypt the information.

In some embodiments, the software instructions further configure the processor to provide a user interface that enables the user to interact with the application via the virtual machine.

In some embodiments, the host device further includes a network card. In these embodiments, the software instructions can further cause the processor to enable the virtual machine to utilize the network card to communicate with the application via an interface of the native OS.

Another aspect of the invention includes a computer-implemented method of authenticating a user to access an application having a target network address. The method includes steps of: 1) causing a processor of a host device to instantiate a virtual machine on top of a native OS running on the host device, wherein the virtual machine stores user credential information associated with the user, 2) instantiating, by the virtual machine, a Transmission Control Protocol/Internet Protocol (TCP/IP) messaging server having an IP address different from any IP address of the host device, 3) rendering, by the virtual machine, an encrypted messaging interface that utilizes that. TCP/IP messaging server to communicate with the application over a TCP/IP networking layer, and 4) negotiating, by the virtual machine, authentication of the user with the application by utilizing the user credential information and information received through the encrypted messaging interface.

Various objects, features, aspects and advantages of the inventive subject matter will become more apparent from the following detailed description of preferred embodiments, along with the accompanying drawing figures in which like numerals represent like components.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows one embodiment of private, portable encryption and authentication service module (PPEASM) in a device communicating with applications in a host computer.

FIG. **2** shows one embodiment of PPEASM in a host computer communicating with applications in the host computer.

FIG. **3** shows one embodiment of PPEASM in a host computer communicating with applications in a third party system.

FIG. **4** shows one embodiment of application with a client interface to interact with PPEASM.

## DETAILED DESCRIPTION

Throughout the following discussion, numerous references will be made regarding servers, services, interfaces, engines, modules, machines, clients, peers, portals, platforms, or other systems formed from computing devices. It should be appreciated that the use of such terms is deemed to represent one or more computing devices having at least one processor (e.g., ASIC, FPGA, DSP, x86, ARM, ColdFire, GPU, multi-core processors, etc.) configured to execute software instructions stored on a computer readable tangible, non-transitory medium (e.g., hard drive, solid state drive, RAM, flash, ROM, etc.). For example, a server can include one or more computers operating as a web server, database server, or other type of computer server in a manner to fulfill described roles, responsibilities, or functions. One should further appreciate the disclosed computer-based algorithms, processes, methods, or other types of instruction sets can be embodied as a computer program product comprising a non-transitory, tangible computer readable media storing the instructions that cause a processor to execute the disclosed steps. The various servers, systems, databases, or interfaces can exchange data using standardized protocols or algorithms, possibly based on HTTP, HTTPS, AES, public-private key exchanges, web service APIs, known financial transaction protocols, or other electronic information exchanging methods. Data exchanges can be conducted over a packet-switched network, a circuit-switched network, the Internet, LAN, WAN, VPN, or other type of network.

The terms "configured to" and "programmed to" in the context of a processor refer to being programmed by a set of software instructions to perform a function or set of functions.

The following discussion provides many example embodiments of the inventive subject matter. Although each embodiment represents a single combination of inventive elements, the inventive subject matter is considered to include all possible combinations of the disclosed elements. Thus if one embodiment comprises elements A, B, and C, and a second embodiment comprises elements B and D, then the inventive subject matter is also considered to include other remaining combinations of A, B, C, or D, even if not explicitly disclosed.

As used herein, and unless the context dictates otherwise, the term "coupled to" is intended to include both direct coupling (in which two elements that are coupled to each other contact each other) and indirect coupling (in which at least one additional element is located between the two elements). Therefore, the terms "coupled to" and "coupled with" are used synonymously. Further, the terms "coupled to" and "coupled with" are used euphemistically in a networking context to mean "communicatively coupled with" where two or more devices are configured to exchange data (e.g., uni-directionally, bi-directionally, peer-to-peer, etc.) with each other possibly via one or more intermediary devices.

The recitation of ranges of values herein is merely intended to serve as a shorthand method of referring individually to each separate value falling within the range. Unless otherwise indicated herein, each individual value is incorporated into the specification as if it were individually recited herein. All methods described herein can be performed in any suitable order unless otherwise indicated herein or otherwise clearly contradicted by context. The use of any and all examples, or exemplary language (e.g. "such as") provided with respect to certain embodiments herein is intended merely to better illuminate the invention and does not pose a limitation on the scope of the invention otherwise claimed. No language in the specification should be construed as indicating any non-claimed element essential to the practice of the invention.

In some embodiments, the numbers expressing quantities of properties such as dimensions used to describe and claim certain embodiments of the invention are to be understood as being modified in some instances by the term "about." Accordingly, in some embodiments, the numerical parameters set forth in the written description and attached claims are approximations that can vary depending upon the desired properties sought to be obtained by a particular embodiment. In some embodiments, the numerical parameters should be construed in light of the number of reported significant digits and by applying ordinary rounding techniques. Notwithstanding that the numerical ranges and parameters setting forth the broad scope of some embodiments of the invention are approximations, the numerical values set forth in the specific examples are reported as precisely as practicable. The numerical values presented in some embodiments of the invention may contain certain errors necessarily resulting from the standard deviation found in their respective testing measurements.

Unless the context dictates the contrary, all ranges set forth herein should be interpreted as being inclusive of their endpoints and open-ended ranges should be interpreted to include only commercially practical values. Similarly, all lists of values should be considered as inclusive of intermediate values unless the context indicates the contrary.

As used in the description herein and throughout the claims that follow, the meaning of "a," "an," and "the" includes plural reference unless the context clearly dictates otherwise. Also, as used in the description herein, the meaning of "in" includes "in" and "on" unless the context clearly dictates otherwise.

Groupings of alternative elements or embodiments of the invention disclosed herein are not to be construed as limitations. Each group member can be referred to and claimed individually or in any combination with other members of the group or other elements found herein. One or more members of a group can be included in, or deleted from, a group for reasons of convenience and/or patentability. When any such inclusion or deletion occurs, the specification is herein deemed to contain the group as modified thus fulfilling the written description of all Markush groups used in the appended claims.

The present invention provides systems and devices that allow a user to use user credentials stored on a personal, portable device to securely negotiate authentication with an application running on another device. In some embodiments, the secure negotiation of authentication is performed through a private, portable encryption and authentication service module (PPEASM). One of the advantages of this inventive subject matter is that the PPEASM can provide a user encryption and authentication services in a remote or guest environment using user credentials stored on a personal, portable device. The remote or guest environment in this disclosure refers to a third party device that is not under the user's control (e.g., a public computer in a library or a hotel, etc.). In some embodiments, the third party device can be a host device that is directly connected to the user's private device. In other embodiments, the third party device can be a remote device that is communicatively coupled to the user's private device over a network (e.g., the Internet, a LAN, etc.).

In some embodiments, the secure authentication of user is achieved through establishing secured communication channel between a virtual machine and the application through a SafeChannel as described in the co-pending U.S. Application concurrently filed with this application titled "Safechannel Encrypted Messaging System."

Preferably, the PPEASM is a standalone software application that can be stored in the personal, portable devices such as a USB thumb drive or a memory card. However, it is also contemplated that in some embodiments, the PPEASM is a web service or system service that can be provided via a network.

Another advantage of the inventive subject matter includes a device that allows the user to anonymously access and utilize electronics (e.g., processor, memory, etc.) of a host device. In some embodiments, the host device can be a computing device having one or more processor. Preferably, a native operating system (OS) (e.g., Windows, Mac OS, Linux, UNIX, etc.) is already running on the host device.

In a preferred embodiment, the device is a portable, hand-held electronic device (e.g., a thumb drive, a CD-ROM, a cell phone, a smart phone, an iPod, an iPad, etc.). However, in some embodiments, it is also contemplated the device is a built-in device (e.g., a memory, etc) embedded into another device (e.g., a computer, a server, etc.). In these embodiments, the built-in device is independent from the host device.

The device includes an onboard database and an onboard memory (e.g., random access memory, solid state drive, etc.) storing executable version of the software instructions for the PPEASM software application. The onboard database can comprise various types of data volumes. In a preferred embodiment, the onboard database includes a static read-only data volume for certificate storage and a read/write data volume for runtime work.

The onboard database stores user credential information. As used herein, the user credential information includes any information that can be used to authenticate or validate user's identity and/or authority. For example, the user credential information can be any of a password, a challenge phrase, a challenge phrase hash, or a combination of any of those.

In a preferred embodiment, the user credential information is stored exclusively in the onboard database, but not stored in any remote service providers' systems or host devices. More preferably, the user credential information is stored in an encrypted read-only data volume within the onboard database, which is accessible using an encrypted messaging system. Thus, the user credential information is generally not modifiable and safe from the hostile environment that the onboard database may interact with.

FIG. **1** describes an exemplary environment **100** in which the PPEASM application **130** can be operated. The environment **100** includes a personal, portable device **125** (e.g., a USB thumb drive) and a host device **105**. The host device can be a generic personal computer that is not under control of a user associated with the personal, portable device **125**. The host device **105** usually includes several hardware components **120**. For example, the host device **105** can include one or more processors, memory, persistent data storage such as a hard drive or a solid state drive, ports (e.g., USB sockets, etc.) for connecting with external devices, network cards, network interface for connecting the host device **125** to a network (e.g., the Internet, a LAN, etc.), and many others. The host device **105** can also include a native OS **115** for managing resources of the host device **105**. In addition, the host device **105** can include one or more software applications (e.g., **110***a*, **110***b*, **110***c*, etc.) that run on top of the native OS **115**.

In some embodiments, the personal, portable device **125** includes memory that stores an executable version of the software instructions for the PPEASM application **130** and a database **135**. When the personal, portable device **125** is connected to the host device **105** (e.g., by plugging the USB thumb drive into a USB socket of the host device **105**, etc.), the host device can be triggered to run the PPEASM application **130**. In some embodiments, the PPEASM application **130** is triggered to run on top of the native OS **115** of the host device **105**.

Once the PPEASM application **130** is triggered to be executed by the processor of the host device **105**, the PPEASM application **130** causes the processor to instantiate several modules, engines, or machines to perform functions of the PPEASM application **130**. In some embodiments, the PPEASM application **130** causes the processor of the host device **105** to instantiate a native encrypted messaging interface **140**, a hypervisor **145**, and a virtual machine **150**.

In a preferred embodiment, the virtual machine **150** is a kernel-based virtual machine, which includes a secure Linux Kernel. However, it is contemplated that any suitable types of virtual machine (e.g., any type **2** software based virtual

machine that runs on a host operating system) such as VMware, Xen, VirtualBox, Qemu, etc.) that is capable to perform the functions described below can be used.

In some embodiments, the virtual machine **150** comprises a Transmission Control Protocol/Internet Protocol (TCP/IP) messaging server **155** and a TCP/IP data streaming server **170**. After the TCP/IP messaging server **155** and TCP/IP data streaming server **170** have been instantiated, the virtual machine **150** is programmed to retrieve all of the IP addresses associated with the host device **105** (e.g., IP address associated with the network card of the host device **105**, etc.) by interfacing with the native OS **115**. In some embodiments, the virtual machine **150** is then programmed to assign network addresses (e.g., IP addresses), that are distinct from any of the IP addresses associated with the host device **105**, to the TCP/IP messaging server **155** and to the TCP/IP data streaming server **170**. In some embodiments, the IP addresses assigned to TCP/IP messaging server **155** and the TCP/IP data streaming server **170** are identical. However, it is also contemplated that the IP addresses assigned to the TCP/IP messaging server **155** and the TCP/IP data streaming server **170** are different from each other.

The virtual machine **150** also includes an authentication and encryption module **160**. The authentication and encryption module **160** is configured to negotiate authentication of user with other applications and provide secured communication between the PPEASM **130** application and the other applications. In some embodiment, instead of storing the user credential information in the data storage **135**, the user credential information can be stored within an encrypted data volume **165** of the virtual machine **150**.

In some embodiments, the virtual machine **150** can be instantiated, utilized and then unloaded within a limited time or upon the user's action (e.g., request, etc). In some other embodiments, the virtual machine **150** can be instantiated and loaded in a persistent mode to provide ongoing services.

Once the virtual machine **150** is instantiated, the PPEASM **130** further causes the processor **120** to render an encrypted messaging interface **140** on the host device **105**. The encrypted messaging interface **140** utilizes the TCP/IP messaging server **155** for communicating between the virtual machine **150** and the applications **110***a*, **110***b*, **110***c* over a TCP/IP networking layer.

In some embodiments, in order to enable the applications **110***a*, **110***b*, and **110***c* to be able to communicate with the virtual machine **150** over the TCP/IP networking layer, an add-on (or a plug-in) must be added to the applications **110***a*, **110***b*, and **110***c*. The add-on or plug-in can be implemented as a PPEASM interface that is programmed to interface with the virtual machine **150** via the TCP/IP messaging server **155** and the TCP/IP data streaming server **170**. In some embodiments, the PPEASM interface has a distinct IP address (e.g., a target address) for this communication to occur.

The virtual machine **150** is programmed to interact with the user using a key/value pairs that are delineated using a special character. Through the encrypted messaging interface **140**, the PPEASM **130** and the user establishes a secure connection. For example, the user can interact with the PPEASM **130** using a key/value pairs that are delineated using a special character. In a preferred embodiment, the virtual machine **150** comprises an Open SSL, and Rivest-Shamir-Adleman (RSA) Key based authentication method is used to establish the secure connection. In this embodiment, an encryption key is public and a decryption key is secretly kept in the encrypted data volume **165**. However, in other embodiments, any suitable type of authentication method using user credential information can be used to establish the secure connection.

If the user credential information is accepted by the virtual machine **150**, a token is issued to the virtual machine **150**. In a preferred embodiment, the token is time sensitive such that when the token expires, the authentication process should start over.

Generally, the token is common to all transactions (e.g., authentication, hash function, encryption, compression, decryption that the user makes. In this scenario, a transaction ID is issued per each transaction so that the user and the PPEASM **130** can manage multiple transactions for the same client at one time.

As mentioned before, the virtual machine **150** is programmed to allow the user to interact with applications either running on the host device **105** or running on a remote device communicatively coupled with the host device **105**. Thus, the virtual machine **150** needs to discover what applications running on top of the native OS **115** or running on other devices over a network. In some embodiments, the virtual machine **150** is programmed to broadcast a signal through its TCP/IP messaging server **155** over the TCP/IP networking layer. The signal includes the IP address that has been assigned to the TCP/IP messaging server **155**. When the PPEASM interfaces that have been added onto the applications received such a signal, the PPEASM interfaces are programmed to send a reply to the virtual machine **150** at the IP address included in the broadcast signal. Each reply also includes the IP address that is assigned to the respective PPEASM interface.

Once the secure connection between the user and the PPEASM **130** is established, the PPEASM **130** can configure the hardware (processor) **120** of the host computer **105** to establish a secured communication channel between the virtual machine **150** and the applications **110***a*, **110***b*, **110***c* over the TCP/IP networking layer. In a preferred embodiment, the PPEASM **130** can configure the hardware (processor) **120** to send the user credential information to the applications **110***a*, **110***b*, **110***c* to establish a secured communication channel between the virtual machine **150** and the applications **110***a*, **110***b*, **110***c*. In this embodiment, it is also preferred that the user credential information is encrypted by the authentication, encryption, decryption, and compression services module **160** in the virtual machine **150** before being transmitted to the applications **110***a*, **110***b*, **110***c*.

Either automatically upon instantiation, or upon triggered by the user, the virtual machine **150** is programmed to establish a secured data channel with the applications running on the host device via the PPEASM interface associated with the application **110***a* by negotiating a data encryption protocol. In a preferred embodiment, the virtual machine comprises an Open SSL, and Rivest-Shamir-Adleman (RSA) Key based authentication method is used to establish the secure connection. In this embodiment, an encryption key is public and a decryption key is secretly kept in the encrypted data volume **165**. However, in other embodiments, any suitable type of authentication method using user credential information can be used to establish the secure connection. The secured channel allows the virtual machine **150** and the application **110***a* to transfer encrypted data. In these embodiments, the encrypted messaging interface **140** is responsible for encrypting and decrypting data for the virtual machine **150**, while the PPEASM interface is responsible for encrypting and decrypting data for the application **110***a*.

Once a secured connection between the user and the virtual machine **150** and a secured connection between the virtual machine **150** and the applications **110***a*, **110***b*, and **110***c* are established, the user can begin authentication process with the application by the exchange of user credential information (e.g., challenge phrases) via the encrypted messaging interface **140**. In some embodiments, the virtual machine **150** is programmed to negotiate authentication of the user with one of the applications (e.g., applications **110***a*, **110***b*, and **110***c*) so that the user can access the applications. In some embodiments, the virtual machine **150** is programmed to begin the negotiation process by sending an authentication request along with the user credential (e.g., the user credential stored in the data storage **135**) to the application **110***a*. The virtual machine **150** is preconfigured with user credential information (e.g., a unique password, challenge phrase and challenge phrase hash) that is stored in the encrypted data volume **165**.

If the user is authenticated, the virtual machine **150** is programmed to instantiate a user interface that enables the user to interact (e.g., access, use, send commends, etc.) with the applications **110***a*, **110***b*, and **110***c* via the virtual machine **150**. In a preferred embodiment, the hardware (processor) **120** can generate a plurality of user interface such that a single user interface is specifically used to an individual application **110***a*, **110***b*, **110***c*. In other embodiments, a user interface can be used for more than one application.

As mentioned above, in some embodiments, it is required that the application to have a PPEASM interface in order to communicate with the virtual machine **150**. In FIG. **2**, since only applications **110***a* and **110***c* have the add-ons (PPEASM interfaces **211***a* and **211***c*), the virtual machine can communicate with only applications **110***a* and **110***c*, but not with application **110***b*.

FIGS. **1** and **2** illustrate embodiments in which the user interacts with applications that run locally on the host device **105** via the virtual machine **150**. However, in some other embodiments, the PPEASM application **130** also allows users to interact with applications that run on a remote computing device that is communicatively coupled with the host device **105**.

FIG. **3** illustrate such an approach. In FIG. **3**, environment **300** includes a host device **105** and applications **310***a* and **310***b*. The applications **310***a* and **310***b* may be running on the same or separate computing device, and are communicatively coupled with the host device **105** over a network **305** (e.g., the Internet, a LAN, etc.). It is also contemplated that the applications **310***a*, **310***b* can be a web service or a mobile application. As shown, both applications **310***a* and **310***b* have PPEASM interfaces **311***a* and **311***b*, respectively for communicating with the virtual machine **150**. Using the same method as described above, the virtual machine **150** can establish a secured communication channel with the applications **310***a* and **310***b* over a TCP/IP networking layer via the TCP/IP messaging server **155** and the TCP/IP data streaming server **170**. The virtual machine **150** can then authenticate the user to access and interact with the applications **310***a* and **310***b* using the method described above.

Preferably, when the secure communication channel between the PPEASM **130** and the native operating system **115** is established, the PPEASM **130** can configure the hardware (processor) **120** to establish a secure communication channel with only selected third party applications **310***a*, **310***b* having PPEASM interfaces **311***a*, **311***b*. It is contemplated that the host device **105** is pre-configured to allow the access from the virtual machine **150** only to a pre-selected third party applications. In this case, the hardware (processor) **120** can configure to generate a PPEASM interface only in those pre-selected third party applications.

FIG. **4** illustrates a schematic of an application **400** having a client interface **405** (e.g., PPEASM interface). Generally, the application communicably coupleable with the PPEASM has a client (add-on) interface **405** and the main application part **420**. The client interface **405** includes a TCP/IP messaging and streaming client **410** and a module for encryption, decryption, and authentication **415**. The PPEASM communicates with the application **400** through the TCP/IP messaging and streaming client **410**.

Generally, the client interface **405** is an add-on software, which can be in the form of a driver, service, static or dynamic library. However, any suitable form of software that can be used as an interface is contemplated.

Another aspect of the inventive subject matter includes a method of authenticating a user to access an application having a target network address. The method begins with a step of causing a processor of a host device to instantiate a virtual machine on top of a native operating system (OS) running on the host device. Generally, the virtual machine stores user credential information associated with the user. Then the method continues with a step of instantiating, by the virtual machine, a TCP/IP messaging server having an IP address different from any IP address of the host device. Once the virtual machine is instantiated, then the virtual machine can render an encrypted messaging interface that utilizes that TCP/IP messaging server to communicate with the application over a TCP/IP networking layer. Then, the virtual machine can negotiate authentication of the user with the application by utilizing the user credential information and information received through the encrypted messaging interface.

It should be apparent to those skilled in the art that many more modifications besides those already described are possible without departing from the inventive concepts herein. The inventive subject matter, therefore, is not to be restricted except in the spirit of the appended claims. Moreover, in interpreting both the specification and the claims, all terms should be interpreted in the broadest possible manner consistent with the context. In particular, the terms "comprises" and "comprising" should be interpreted as referring to elements, components, or steps in a non-exclusive manner, indicating that the referenced elements, components, or steps may be present, or utilized, or combined with other elements, components, or steps that are not expressly referenced. Where the specification claims refers to at least one of something selected from the group consisting of A, B, C . . . and N, the text should be interpreted as requiring only one element from the group, not A plus N, or B plus N, etc.

What is claimed is:

**1**. A portable, hand-held electronic device, through which a user can anonymously utilize a host device comprising a processor to communicate with a target application having a target network address, wherein the host device includes a native operating system (OS), the portable, hand-held electronic device comprises:

an onboard database that stores user credential information; and

an onboard memory storing software instructions that, when executed by the processor, configure the processor to perform the steps of

(a) receiving IP addresses associated with the host device;

(b) instantiating a virtual machine that runs on top of the native OS, wherein the virtual machine com-

prises a Transmission Control Protocol/Internet Protocol (TCP/IP) messaging server having an IP address different from any of the received IP address of the host device,

(c) rendering an encryption and decryption service on the virtual machine for encrypting and decrypting data between the onboard database and the virtual machine,

(d) rendering an encrypted messaging interface on the host device that utilizes the TCP/IP messaging server for encrypting and decrypting data between the virtual machine and the target application over a TCP/IP networking layer,

(e) negotiating a data encryption protocol with the target application through a private portable encryption authentication and service module (PPEASM) interface associated with the application to enable encrypting and decrypting data between the target application and a PPEASM application, and

(f) negotiating authentication of the user with the target application by utilizing the user credential information and information received at the encrypted messaging interface from the PPEASM application with user credential information on the onboard database accessed via the encryption and decryption service.

**2**. The portable, hand-held electronic device of claim **1**, wherein the target application runs on top of the native OS.

**3**. The portable, hand-held electronic device of claim **1**, wherein the target application runs on a remote device communicatively coupled to the host device over a network.

**4**. The portable, hand-held electronic device of claim **1**, wherein the software instructions further configure the processor to send the user credential information to the target application through the PPEASM interface.

**5**. The portable, hand-held electronic device of claim **4**, wherein the software instructions further configure the processor to encrypt the user credential information before sending the encrypted user credential to the target address.

**6**. The portable, hand-held electronic device of claim **1**, wherein the host device further comprises a network card, wherein the software instructions further configure the processor to enable the virtual machine to utilize the network card to communicate with the target application via an interface of the native OS.

**7**. The portable, hand-held electronic device of claim **1**, wherein the user credential comprises at least one of the following: a password, a challenge phrase, and a challenge phrase hash.

**8**. The portable, hand-held electronic device of claim **1**, wherein the virtual machine comprises a secure operating system kernel.

**9**. The portable, hand-held electronic device of claim **1**, wherein the onboard database further comprises a static read-only data volume for certificate storage and a read/write data volume for runtime work.

**10**. The portable, hand-held electronic device of claim **1**, wherein the software instructions further configure the processor to decrypt the information.

**11**. The portable, hand-held electronic device of claim **1**, wherein the software instructions further configure the processor to provide a user interface that enables the user to interact with the target application via the virtual machine.

**12**. A method of authenticating a user to access a target application having a target network address, comprising:

causing a processor of a host device to instantiate a virtual machine on top of a native operating system (OS) running on the host device, wherein the virtual machine stores user credential information associated with the user;

receiving IP addresses associated with the host device;

instantiating, by the virtual machine, a Transmission Control Protocol/Internet Protocol (TCP/IP) messaging server having an IP address different from any of the received IP address associated with the host device;

rendering, by the virtual machine, an encryption and decryption service for encrypting and decrypting data between an onboard database storing the user credential information and the virtual machine,

rendering, by the host device, an encrypted messaging interface that utilizes the TCP/IP messaging server to encrypt and decrypt data with the target application over a TCP/IP networking layer;

negotiating a data encryption protocol with the target application through a private portable encryption authentication and service module (PPEASM) interface associated with the application to enable encryption and decryption of data between the target application and a PPEASM application; and

negotiating, by the virtual machine, authentication of the user with the target application by utilizing the user credential information and information received at the encrypted messaging interface from the PPEASM application with user credential information on the onboard database accessed via the encryption and decryption service.

**13**. The method of claim **12**, wherein the target application runs on top of the native OS.

**14**. The method of claim **12**, wherein the target application runs on a remote device communicatively coupled to the host device over a network.

**15**. The method of claim **12**, further comprising sending, by the virtual machine, the user credential information to the target application through the PPEASM interface.

**16**. The method of claim **15**, wherein the software instructions further configure the processor to encrypt the user credential information before sending it the target address.

**17**. The method of claim **15**, wherein sending the user credential information comprises utilizing a network card of the host device to send the user credential information to the target application via an interface of the native OS.

**18**. The method of claim **12**, wherein the user credential comprises at least one of the following: a password, a challenge phrase, and a challenge phrase hash.

**19**. The method of claim **12**, wherein the virtual machine comprises a secure operating system kernel.

**20**. The method of claim **12**, further comprising decrypting the information received through the encrypted messaging interface.

**21**. The method of claim **12**, further comprising enabling the user to interact with the target application via the virtual machine.

* * * * *