

PRIVACY CENTRIC DATA SECURITY IN CLOUD ENVIRONMENT**BACKGROUND**

[0001] The present invention relates to computer security, and more specifically, to privacy centric data security in a cloud environment.

SUMMARY

[0002] Embodiments include a method. The method includes receiving, at a cloud data privacy service, a request from a user device to permit a first data processor to access private data associated with the user. The request includes a request for a first data access block relating to the private data, and a data filter describing one or more access parameters relating to the first data processor and the private data. The method further includes generating the first data access block, at the cloud data privacy service, based on the private data and the data filter. The method further includes transmitting the first data access block from the cloud privacy service to the user device. The user device is configured to transmit the first data access block to the first data processor. The method further includes receiving, at the cloud data privacy service, a request from the first data processor for the private data, the request including the first data access block. The method further includes determining, at the cloud data privacy service, that the first data access block received from first data processor is valid, and in response granting the first data processor at least partial access to the private data.

[0003] Embodiments further include a system. The system includes a processor, and a memory storing a program, which, when executed on the processor, performs an operation. The operation includes receiving, at a cloud data privacy service, a request from a user device to permit a first data processor to access private data associated with the user. The request includes a request for a first data access block relating to the private data, and a data filter describing one or more access parameters relating to the first data processor and the private data. The operation further includes generating the first data access block, at the cloud data privacy service, based on the private data and the data filter. The operation further includes transmitting the first data access block from the cloud privacy service to the user device. The user device is configured to transmit the first data access block to the first data processor.

The operation further includes receiving, at the cloud data privacy service, a request from the first data processor for the private data, the request including the first data access block. The operation further includes determining, at the cloud data privacy service, that the first data access block received from first data processor is valid, and in response granting the first data processor at least partial access to the private data.

[0004] Embodiments further include a non-transitory computer program product, the computer program product including a computer-readable storage medium having computer-readable program code embodied therewith, the computer-readable program code executable by one or more computer processors to perform an operation. The operation includes receiving, at a cloud data privacy service, a request from a user device to permit a first data processor to access private data associated with the user. The request includes a request for a first data access block relating to the private data, and a data filter describing one or more access parameters relating to the first data processor and the private data. The operation further includes generating the first data access block, at the cloud data privacy service, based on the private data and the data filter. The operation further includes transmitting the first data access block from the cloud privacy service to the user device. The user device is configured to transmit the first data access block to the first data processor. The operation further includes receiving, at the cloud data privacy service, a request from the first data processor for the private data, the request including the first data access block. The operation further includes determining, at the cloud data privacy service, that the first data access block received from first data processor is valid, and in response granting the first data processor at least partial access to the private data.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0005] Figure 1 illustrates a system for privacy centric data security in a cloud environment, according to at least one embodiment.

[0006] Figures 2A-C illustrate a user device, a cloud server, and a data processor in a system for privacy centric data security in a cloud environment, according to at least one embodiment.

[0007] Figures 3A-B are a flow diagram illustrating data flow for privacy centric data security in a cloud environment, according to at least one embodiment.

[0008] Figures 4A-C illustrate data structures for privacy centric data security in a cloud environment, according to at least one embodiment.

[0009] Figures 5A-C illustrate privacy filters for privacy centric data security in a cloud environment, according to at least one embodiment.

[0010] Figures 6A-B illustrate data access block structures for privacy centric data security in a cloud environment, according to at least one embodiment.

[0011] Figure 7 illustrate techniques for navigating data access block structures in a privacy centric data security in a cloud environment, according to at least one embodiment.

[0012] Figures 8A-B illustrate a user interface for privacy centric data security in a cloud environment, according to at least one embodiment.

[0013] Figure 9 depicts a cloud computing environment, according to at least one embodiment.

[0014] Figure 10 depicts abstraction model layers, according to at least one embodiment.

DETAILED DESCRIPTION

[0015] In the modern connected world, data privacy is becoming increasingly important. In an embodiment, data privacy can be improved through a privacy-centric system, in which personal data is stored where the owner decides, and access to the data is under real-time control of the owner at any time. To facilitate this, a user can own a virtual data safe to manage the user's personal data and the scope of access allowed for the data. In an embodiment, this can be a secure service in the cloud, operated by a trusted service provider, and installed in the region chosen by the user. The exchange of information is negotiated through this virtual data safe. Further, unlike a physically secure safe (which can be opened by a key or code, but is either open or locked) the virtual data safe may unlock partial access to its content.

[0016] In an embodiment, access to read the user's private data is implemented by a secure access token. Each data requestor (e.g., a user's medical provider, financial institution, social media service, etc.) is given a token allowing the requestor to read the user

data. The user, further, can govern precisely which data each requestor is allowed to access, for example providing a medical provider with access to a relevant portion of the user's medical data and a financial provider with access to a relevant portion of the user's financial data. A user can withdraw the token, revoking access to the requestor. In an embodiment, the requestor can access the data (stored in the virtual data safe) through a web service, or suitable API (e.g., a RESTful API). Further, the user can choose to be notified whenever his or her data is accessed. Notification can be configured per item, based on the requestor's token.

[0017] In an embodiment, service providers who want to store data about the user are required to store the data in the user's virtual data safe. Trusted authorities may request access to the data, under control of the user. The user can be informed in real time about the access request and can approve or deny (via, for example, push notification or SMS to her mobile device). Further, to avoid overwhelming the user with notifications, the user can decide how often or at which time or event she receives the next notification for the specific requestor or data element. Further, in case the user is not able to react in time to approve or deny the access, temporary access rules can be applied for data elements for which the user has provided a pre-approval for the temporary access.

[0018] In an embodiment, a user can further choose to allow broader access to data that has been anonymized. This can allow a user to participate in public polls but limit disclosure of specifics within his private data. For example, specific information about the person can be anonymized so that the requestor receives only limited, or aggregate, results. In an embodiment, the provider of the virtual data safe acts as a trusted party with which the user has engaged into a trust relationship based on the service contract the two parties have engaged into.

[0019] For example, a user may also choose to tag specific data elements with a type of use such as "temporary read only," as opposed to "own/copy," in which case the requestor of a data element may process the data temporarily but not store it permanently. The decision of the user can be implemented by means of a privacy token which the user obtains in real-time from the trusted service provider. The user (holder of the privacy token) might forward this token to a third party which might in turn approach the trusted service provider and

present the privacy token. The trusted service provider can then hand over requested data associated with the privacy token.

[0020] Further, one privacy token can be used to generate a secondary privacy token (with permission from the user), allowing another entity to access the same, or a subset, of the user's private information. In an embodiment, all involved parties adhere to policies which are bound to the data under the control of the data owner. All accesses, transactions and interactions are logged on the virtual data safe.

[0021] Figure 1 illustrates a system for privacy centric data security in a cloud environment, according to at least one embodiment. One or more user devices 100 are communicatively coupled with a cloud system 150. In an embodiment, the user devices 100 can include any suitable user device, including a laptop computer, a desktop computer, a smartphone, a tablet, etc. The cloud system 150 can be any suitable cloud system (e.g., a public cloud, a private cloud, a hybrid cloud, etc.). The cloud system 150 is described in more detail below with regard to Figures 9 and 10. While one or more disclosed embodiments describe a cloud-based system as one example, any suitable system, including a non-cloud-based system can be used.

[0022] The user devices 100 can communicate with the cloud system 150 using any suitable communication technique, including a WiFi connection, a cellular connection, a wired connection, etc. In an embodiment, one or more of the user devices 100 communicate with the cloud system 150 using a cellular connection via the mobile gateway 120. Further, the user devices 100 can communicate with the cloud system 150 using any suitable communication network, including a local area network, a wide area network, the Internet, etc.

[0023] One or more data processors 180A-N are further communicatively coupled with the cloud system 150 and, in an embodiment, with each other. As described further below, the data processors 180A-N can be systems that use private data associated with the users of the user devices 100. For example, the data processors 180A-N can include social networks, financial institutions, medical institutions, businesses, educational institutions, electronic games, etc. The data processors 180A-N can communicate with the cloud system 150 using any suitable communication technique, including a WiFi connection, a cellular connection,

a wired connection, etc. Further, the data processors 180A-N can communicate with the cloud system 150 using any suitable communication network, including a local area network, a wide area network, the Internet, etc.

[0024] In an embodiment, the cloud system 150 includes a cloud privacy service 162. As discussed further below with regard to subsequent figures, the cloud privacy service 162 can facilitate privacy centric data security by governing interactions between the user devices 100 and the data processors 180A-N. For example, in an embodiment private data for the users of the user devices 100 can be maintained in a secure repository at the cloud system 150 and the cloud privacy service 162 can govern access to this data by the data processors 180A-B. As another example, private data for the users of the user devices can be maintained at the user devices 100 or at another storage location maintained by the users (e.g., a storage location in the user's home or office) and the cloud privacy service 162 can govern access to this data. This is discussed in further detail with regard to Figures 2A-C and later, below.

[0025] As discussed further below, according to one or more disclosed embodiments a user can grant access to his or her private data to one or more of the data processors 180A-N, using the cloud privacy service 162. This access can be temporary, and can be limited (e.g., requiring notification to the user or approval by the user). The data processors 180A-N do not retain the user's private data, beyond the allowed temporary usage. Instead, the data remains stored in a repository controlled by the user (e.g., in the cloud 150 or at the user device 100).

[0026] In an embodiment, one or more of the data processors 180A-N can communicate with each other. For example, the data processor 180A may be a financial services provider, which sub-contracts some functions to the data processor 180N. As another example, the data processor 180A may be a medical provider and the data processor 180N may be an insurer. These are merely examples, and any suitable data processors can be used.

[0027] As discussed further below, the user can allow the data processor 180A to grant access to the user's private data the data processor 180N (e.g., by generating a data access block, as discussed with regard to Figures 3A-B and Figures 6A-6B, below). The data processor 180A can grant the data processor 180N the same access allowed for the data

processor 180A, including any notification and approval requirements, or more limited access (e.g., a user might allow a medical provider to grant an insurer access to a user's biographical and credit card information, but not the user's medical records). This is discussed in more detail below.

[0028] In an embodiment, the user devices 100 and the data processors 180A-N can communicate with the cloud privacy service 162 using a suitable application programming interface (API). For example, a RESTful API, or any other suitable API, can be used. Alternatively, the user devices 100 and/or the data processors 180A-N can communicate with the cloud privacy service 162 using any other suitable means, including a webpage, network messages, e-mail, SMS messaging, social network messages, etc. For example, a user wishing to grant access to his or her private data to a medical provider could authorize the cloud privacy service 162 to send an e-mail or SMS message to the medical provider granting access and providing a link to the data to allow the medical provider authorized access to the data. In this way the data processor (e.g., the medical provider) can access the data without specifically implementing direct communication with the cloud privacy service 162.

[0029] Figures 2A-C illustrate a user device, a cloud server, and a data processor in a system for privacy centric data security in a cloud environment, according to at least one embodiment. Figure 2A is a block diagram illustrating a user device 200 (e.g., one of the user devices 100 illustrated in Figure 1). The user device 200 includes a processor 202, a memory 210, and network components 220. The processor 202 generally retrieves and executes programming instructions stored in the memory 210. The processor 202 is included to be representative of a single central processing unit (CPU), multiple CPUs, a single CPU having multiple processing cores, graphics processing units (GPUs) having multiple execution paths, and the like.

[0030] The network components 220 include the components necessary for the user device 200 to interface with the cloud system 150, as discussed above in relation to Figure 1. For example, the network components 220 can include wired, WiFi or cellular network interface components and associated software to facilitate communication between the user device 200 and the cloud system 150.

[0031] Although the memory 210 is shown as a single entity, the memory 210 may include one or more memory devices having blocks of memory associated with physical addresses, such as random access memory (RAM), read only memory (ROM), flash memory, or other types of volatile and/or non-volatile memory. The memory 210 generally includes program code for performing various functions related to use of the user device client 200. The program code is generally described as various functional “applications” or “modules” within the memory 210, although alternate implementations may have different functions and/or combinations of functions.

[0032] Within the memory 210, a data owner service 212 facilitates privacy centric data security, as discussed in subsequent figures. For example, in an embodiment the user device 200 includes a data repository 214. The data repository 214 can store private data for the user of the user device 200. The data owner service 212 can be used to interact with a privacy service in a cloud system (e.g., the cloud privacy service 162 illustrated in Figure 1 and the privacy service 262 illustrated in Figure 2B) in order to ensure privacy centric data security. This is discussed in further detail in subsequent figures.

[0033] Figure 2B is a block diagram illustrating a server (e.g., a virtual machine or a physical server, as discussed below with regard to Figure 9) in a cloud system 250 (e.g. the cloud system 150 illustrated in Figure 1). The cloud system 250 includes a processor 252, a memory 260, and network components 270. The processor 252 generally retrieves and executes programming instructions stored in the memory 260. The processor 252 is included to be representative of a single central processing unit (CPU), multiple CPUs, a single CPU having multiple processing cores, graphics processing units (GPUs) having multiple execution paths, and the like.

[0034] The network components 270 include the components necessary for the cloud system 250 to interface with a wireless communication network, as discussed above in relation to Figure 1. For example, the network components 270 can include wired, WiFi or cellular network interface components and associated software to facilitate communication between the user device 200, the cloud system 250, and the data processor 280 illustrated in Figure 2C.

[0035] Although the memory 260 is shown as a single entity, the memory 260 may include one or more memory devices having blocks of memory associated with physical addresses, such as random access memory (RAM), read only memory (ROM), flash memory, or other types of volatile and/or non-volatile memory. The memory 260 generally includes program code for performing various functions related to use of the cloud system 250. The program code is generally described as various functional “applications” or “modules” within the memory 260, although alternate implementations may have different functions and/or combinations of functions.

[0036] Within the memory 260, a cloud privacy service 262 (e.g., the cloud privacy service 162 illustrated in Figure 1) facilitates privacy centric data security, as discussed in subsequent figures. For example, in an embodiment the memory 260 includes a data repository 264. The data repository 264 can store private data for the user of the user device 200 (e.g., instead of, or in addition to, the data repository 214 illustrated in Figure 2A). The cloud privacy service 262 can be used to interact with the data owner service 212 illustrated in Figure 2A and the data processor service 292 illustrated in Figure 2C, in order to ensure privacy centric data security.

[0037] Further, in an embodiment the cloud privacy service 262 can include a filter generation service 263. In an embodiment, the filter generation service 263 is used to generate user-defined filters for private data that can specify access parameters to the data (e.g., whether notification or approval is required, and what data can be accessed). This is discussed in more detail with regard to Figure 5A, below. Further, the memory 260 can store one or more privacy filters 266, used to filter private data before providing the data to data processors (e.g., the data processor 280 illustrated in Figure 2C). The cloud privacy service 262, data repository 264, and privacy filters 266 are all discussed in further detail in subsequent figures.

[0038] Figure 2C is a block diagram illustrating a data processor 280 (e.g., one of the data processors 180A-N illustrated in Figure 1). The data processor 280 includes a processor 282, a memory 290, and network components 296. The processor 282 generally retrieves and executes programming instructions stored in the memory 290. The processor 282 is included to be representative of a single central processing unit (CPU), multiple CPUs, a

single CPU having multiple processing cores, graphics processing units (GPUs) having multiple execution paths, and the like.

[0039] The network components 296 include the components necessary for the data processor 280 to interface with a wireless communication network, as discussed above in relation to Figure 1. For example, the network components 296 can include wired, WiFi or cellular network interface components and associated software to facilitate communication between the cloud system 250 and the data processor 280, and between the data processor 280 and another data processor.

[0040] Although the memory 290 is shown as a single entity, the memory 290 may include one or more memory devices having blocks of memory associated with physical addresses, such as random access memory (RAM), read only memory (ROM), flash memory, or other types of volatile and/or non-volatile memory. The memory 290 generally includes program code for performing various functions related to use of the data processor 280. The program code is generally described as various functional “applications” or “modules” within the memory 290, although alternate implementations may have different functions and/or combinations of functions.

[0041] Within the memory 290, a data processor service 292 facilitates privacy centric data security, as discussed in subsequent figures. For example, in an embodiment the data processor service 292 can be used to interact with the cloud privacy service 262 illustrated in Figure 2B, in order to ensure privacy centric data security. This is discussed in further detail in subsequent figures.

[0042] Figures 3A-B are a flow diagram 300 illustrating data flow for privacy centric data security in a cloud environment, according to at least one embodiment. At block 302 a user uploads a filter to the cloud privacy service 262 (e.g., using the data owner service 212 illustrated in Figure 2A). In an embodiment, filters are used to govern access by a data processor to the user’s private data. A filter can be generated by the user (e.g., a customer filter), or can be a standard filter (e.g., providing read-only access, read and write access, requiring verification before access, etc). Further, in an embodiment, a filter can be code (e.g., source code or compiled code) for execution by the cloud privacy service. Filters are discussed in further detail with regard to Figures 5A-C, below.

[0043] At block 304, the cloud privacy service 262 compiles the received filter (e.g., if necessary) and stores the filter (e.g., in the privacy filter repository 266 illustrated in Figure 2B). For example, the user device 200 can upload to the cloud privacy service source code for a filter providing read access to the user's private data (or a particular subset of the user's private data). The cloud privacy service 262 can compile the received filter and store the compiled filter in the privacy filter repository 266, for later execution. This is again discussed in more detail with regard to Figures 5A-C, below.

[0044] At block 306, the user device 200 uploads private data to the cloud privacy service 262 (e.g., using the data owner service 212 illustrated in Figure 2A). As discussed above with regard to Figures 2A-2B, in one embodiment a cloud system (e.g., the cloud system 250 illustrated in Figure 2B) stores the user's private data (e.g., in the data repository 264 illustrated in Figure 2B). In this embodiment, the user device 200 uploads the private data to the cloud privacy service. In another embodiment, the user's private data is maintained elsewhere (e.g., at the user device 200) instead of, or in addition to, at the cloud system 250.

[0045] At block 308 the cloud privacy service 262 stores the private data in a data repository (e.g., the data repository 264 illustrated in Figure 2B). At block 310, the user device 200 initiates a session with the first data processor (e.g., the data processor 280A). For example, as discussed above, the data processor 280A can be a medical provider, a financial institution, or any other entity seeking access to the user's private data.

[0046] At block 312, the data processor 280A requests access to specific private data elements relating to the user. In an embodiment, the data processor 280A is communicatively coupled to the user device 200, either directly, through a cloud service (e.g., the cloud service 150 illustrated in Figure 1), or through another communication path. The data processor 280A transmits the data access request to the user device (e.g., through the cloud system 250 using the cloud privacy service 262).

[0047] At block 314, the user device 200 requests from the cloud privacy service 262 a first data access block relating to the requested data and the requesting entity. Data access blocks are described in more detail with regard to Figures 6A-B. In an embodiment, a data access block describes what data a requestor can access, and how the requestor can access

that data (e.g., always allow access, require permission from the user on each access, require permission daily/weekly/monthly, etc.). A data access block can further describe how a sub-requestor related to the initial requestor can access the data. For example, the requestor could be a medical provider, and the data access block could describe both who at the medical provider can access the information and what entities can receive the information from the medical provider (e.g., what information can be provided to insurers, or to other medical providers). In an embodiment, data access blocks provide granular control of private data to the user. Data access blocks are discussed in further detail with regard to Figures 6A-6B, below.

[0048] Further, at block 314, in an embodiment the user device 200 can specify additional detail level filtering for the data access block. In an embodiment, a privacy filter (e.g., a privacy filter uploaded to the cloud privacy service at block 302) specified filtering based on data type. At block 314, the user device can specify additional filtering for the specific data requested by the data processor 280A (e.g., in addition to the existing filters). Figures 8A-8B discuss a user interface to facilitate block 314.

[0049] At block 316, the cloud privacy service generates a data access block and stores the data access block in a data access block registry (e.g., as part of the data repository 264 illustrated in Figure 2B). Data access blocks are discussed in further detail with regard to Figures 6A-B.

[0050] At block 318, the cloud privacy service 262 sends to the user device 200 a data access block for the requested private data. This is again discussed further with regard to Figures 6A-B, below. Alternatively, or in addition, the cloud privacy service 262 can notify the user device 200 of the location of the appropriate data access block (e.g., if the data access block has previously been generated and is maintained at the user device, or at another location). For example, the cloud privacy service 262 could notify the user device 200 of an identifier associated with a data access block stored locally at the user device, or at a location accessible to the user device.

[0051] At block 320, the user device 200 sends to the data processor 280A (e.g., the first data requestor) an address for the cloud privacy service 262, along with the data access block. For example, the user device 200 can transmit to the data processor 280A a Uniform

Resource Locator (URL) identifying the cloud privacy service 262. The data processor 280A can then use the URL to access the cloud privacy service 262. A URL is merely one example, and any suitable address or identifier can be used. Further, the user device 200 sends to the data processor 280A the data access block describing the allowed access for the data processor 280A. Data access blocks are discussed in further detail with regard to Figures 6A-B.

[0052] At block 322, the data processor 280A sends the data access block (e.g., received from the user device 200) to the cloud privacy service 262 in order to retrieve the desired data. In an embodiment, the data processor 280A is permitted to access the data (e.g., as described in the data access block) and at block 330 the cloud privacy service 262 transmits the data to the data processor 280A.

[0053] Alternatively, or in addition, the user can specify that she must receive notice, and/or provide approval, before the data processor 280A is permitted to access the requested data. This can be defined by the user, for example, using a suitable filter and the requirement can be specified in the data access block. As discussed below, the user can define differing controls (e.g., notice required, approval required, or neither required) for different data processors (e.g., approval may be required for one merchant to access a user's credit card information but not for another merchant) and for different data (e.g., approval might be required for access to medical information but not for a user's e-mail address). At block 324, in this scenario, the cloud privacy service 262 notifies the user device 200 about the data access and, if necessary, seeks approval. At block 326, the cloud privacy service generates a subsequent requestor access block (e.g., denoting the request) and stores the access block in the registry (e.g., the data repository 264 illustrated in Figure 2B). At block 328, the user acknowledges, approves, or rejects the data access request, as appropriate depending on the data access requirements and the user's preference. Figures 8A-B describe a user interface to allow the user to acknowledge, approve, or reject the data access block. Assuming the user does not reject the request, the flow proceeds to block 330 and the cloud privacy service sends the data and the subsequent access block to the data processor 280A.

[0054] In an embodiment, multiple data processors can seek access to the private data (e.g., a general medical practitioner and a specialist medical practitioner, a medical facility and an insurer, a financial institution and a credit agency, or any other suitable data

processors). At block 332, the data processor 280A sends to the second data processor 280B the URL of the cloud privacy service 262 and the subsequent access block.

[0055] At block 334, the second data processor 280B sends to the cloud privacy service 262 a request for the desired data along with the subsequent access block. In an embodiment, the user has authorized access for the second data processor 280B as well as for the first data processor 280A. In this embodiment, the flow proceeds to block 342 and the cloud privacy service 362 sends the data to the second data processor 280B.

[0056] Alternatively, or in addition, the user can specify that he or she must receive notice, and/or provide approval, before the second data processor 280B is permitted to access the requested data. This can be defined by the user, for example, using a suitable filter and the requirement can be specified in the data access block. At block 336, in this scenario, the cloud privacy service 262 notifies the user device 200 about the data access and, if necessary, seeks approval. At block 338, the cloud privacy service generates a next subsequent requestor access block (e.g., denoting the request) and stores the access block in the registry (e.g., the data repository 264 illustrated in Figure 2B). At block 340, the user acknowledges, approves, or rejects the data access request, as appropriate depending on the data access requirements and the user's preference. Figures 8A-B describe a user interface to allow the user to acknowledge, approve, or reject the data access block. Assuming the user does not reject the request, the flow proceeds to block 342 and the cloud privacy service sends the data and the subsequent access block to the data processor 280A.

[0057] In an embodiment, the cloud privacy service 262 can further request that the data processors 280A and 280B delete the user's private data after a duration specified by the user. As discussed above, private data can be deleted automatically by the data processors 280A and 280B (e.g., to be allowed access to the private data the data processor must agree to delete the data as required), could be enforced by providing the data processor with a version of the data that will expire after a specified time period (e.g., self-enforcing), and by the cloud privacy service 262 transmitting a request to the data processor to delete the data.

[0058] Figures 4A-C illustrate data structures for privacy centric data security in a cloud environment, according to at least one embodiment. Figure 4A illustrates a hierarchical view of private user data 402-438, along with cumulative keys associated with the data, according

to at least one embodiment. In an embodiment, private user data is stored in a hierarchical relationship (e.g., in a data repository 264 in the cloud system 250, as illustrated in Figure 2B, or in a data repository 214 in the user device 200, as illustrated in Figure 2A). For example, a user's name is stored in a field 402. The user's social security number is stored in a field 404 that is a child of the name field 402. The user's e-mail address 406 is stored in a field 406 which is a sibling of the social security number field 404 and a child of the name field 402. The user's home address is stored in a field 414 that is a child of the e-mail address field 406.

[0059] Further, in an embodiment, a given field of private information can be accessed using a key. These keys can be given to data processors (e.g., the data processor 280 illustrated in Figure 2C), and the data processor can present the key to the service providing data access (e.g., the cloud privacy service 262 illustrated in Figures 2B and 3). If the key is valid, the service provides the data. In an embodiment, each key is signed (e.g., using public/private key encryption or another suitable digital signature) or secured to verify that it is valid.

[0060] Further, different data processors can be provided with different signatures, and the differing signatures can be used to track who is attempting to access the data. For example, as illustrated in Figure 3, the data processor 280A can be provided with a key with different signature from the signature provided to the data processor 280B, and the cloud privacy service 262 can use these differing signatures to track which data processor is attempting to access a given data field. For example, the cloud privacy service 262 can log accesses to data using a given signature.

[0061] Further, in an embodiment, the keys can be hierarchical and can build on each other. For example, shorter keys can allow access to higher level data, while longer keys can allow access to more specific information. As illustrated in Figure 4A, a key "A1" provides access to the name field 402. A key "B2A1" (e.g., the name field key with the addition of "B2") provides access to the e-mail address field 406. A key "C1B2A1" provides access to the home address field 414. Similarly, a key "B3A1" provides access to a user's birthdate. But a longer key is required for access to more specific (e.g., more private) information. A key "C2B3A1" is required for access to a health reports field 416, a key

“D2C2B3A1” is required for access to a general health data field 424, and a key “E2D2C2B3A1” is required for access to allergy data.

[0062] Further, as discussed above and illustrated 4A, a longer key has within it a shorter key to allow access to more general data. This allows one data processor that has been granted to a specific data field (e.g., a medical provider with access to the general health data field 424 using the key “D2C2B3A1”) to provide a shorter key to another entity to allow access to more general data and not the specific data (e.g., providing an insurer with a key “B3A1” to allow access to the birth data field 408 but not general health data field 424). Hierarchical data, and hierarchical keys, as discussed above, is merely one example embodiment. Private data can also be stored in any other suitable fashion (e.g., not hierarchical).

[0063] Figures 4B-C illustrate storage of private data in a table 450, according to at least one embodiment. For example, the table 450 illustrated in Figures 4B-C can be a database table (e.g., in a relational database or any other suitable database) storing the private data illustrated in Figure 4A (e.g., in a data repository 264 as illustrated in Figure 2B or 214 as illustrated in Figure 2A). The item index column 462 provides an index for the data (e.g., to allow the database to identify the data). The item label column 464 describes the data.

[0064] The item key column 466 provides the key used for a data processor to access the data. In an embodiment, the keys can be hierarchical as discussed above in relation to Figure 4A. The value column 468 provides the data itself. The default filters column 470 provides a default filter level for access to the data. For example, the default filter for name data could be to allow “read always” to data processors (allowing access without specific approval, by default), while the default filter for general health data could be “approve by me”, requiring a user to approve any access to this data (e.g., by default). In an embodiment, the default filter levels can be encoded (e.g., using constant integers to represent levels of filtering) instead of using human-readable language. In an embodiment, specific filters can also be provided by users, as discussed below with regard to Figures 5A-C.

[0065] Figures 5A-C illustrate privacy filters (e.g., data shaping filters) for privacy centric data security in a cloud environment, according to at least one embodiment. Figure 5A illustrates generating, storing, and using, privacy filters for privacy centric data security

in a cloud environment, according to at least one embodiment. At block 502, a user (e.g., the data owner 500) selects the desired filter level. In an embodiment, the user compiles a filter term with instructions and requirements for accessing his private data. For example, the user could specify “approve by me and read once by <dentist> and valid until august 2019.” A user interface to provide this is discussed in relation to Figures 8A-B, below.

[0066] At block 504, a filter generation service (e.g., the filter generation service 263 illustrated in Figure 2B) generates the filter. In an embodiment, the filter generation receives and parses the filter term (e.g., from block 502). The filter generation service then uses the privacy filter language dictionary 506 to tokenize the filter.

[0067] For example, the privacy filter language dictionary can specify:

Filterterm := Action [Denominator [<Parameter>]] [| Condition Filterterm].

[0068] The Action can be any suitable action, including “approve,” “create,” “read,” “update,” “delete,” “forward,” “validate,” or “default.” The Denominator can be any suitable denominator, including, “by,” “to,” “once,” “twice,” “never,” “always,” “until,” or “only.” The Condition can be any suitable condition, including “and,” “or,” or “and not.”

[0069] The filter generation service can then analyze the tokens, generate suitable program code, compile the code to machine executable code (if necessary), and store the filter (e.g., the compiled filter) in the filter repository 566. Figure 5B illustrates examples of suitable filters, including human-readable program code.

[0070] At block 512, a data processor (e.g., the data processor 280 illustrated in Figure 2C) requests private data. In an embodiment, the data processor transmits the request to the cloud privacy service, and provides a valid data access block. This is described in more detail with regard to Figure 3, above.

[0071] At block 514, a privacy service (e.g., the cloud privacy service 262 illustrated in Figure 2B) processes the request. For example, the privacy service can receive the data access block from the data processor (e.g., transmitted at block 512). The privacy service can access a data access block registry 564B (e.g., a data access block registry included as part of the data repository 264 illustrated in Figure 2B) and can validate the data access block

using the registry. For example, the privacy service can determine whether a matching data access block is included in the data access block registry 564B. In an embodiment, the data access block is suitably protected to ensure that it is valid (e.g., signed, encrypted, etc.).

[0072] Assuming the data access block is valid, the privacy service can access the requested data from the private data repository 564A (e.g., part of the data repository 264 illustrated in Figure 2B). Further, the privacy service can retrieve a default filter for the data from the private data repository 564A (e.g., the detail filters column 470 illustrated in Figure 4B).

[0073] The privacy service can then retrieve any filter program modules corresponding to the requested data, from the filter repository 566, and can execute the filter program modules. This may require the privacy service to contact the data owner 500 (e.g., the user of the user device 200 illustrated in Figure 2A) to provide notice of the request from the data processor, and to receive acknowledgment, approval, or denial of the request (as appropriate, depending on the filter) from the data owner 500. In an embodiment, as discussed above in relation to Figure 3, the privacy service can then generate a next data access block, update the data access block registry 564B, and send the data to the requesting data processor.

[0074] Figures 5B-C illustrate example privacy filters for privacy centric data security in a cloud environment, according to at least one embodiment. For example, the table 550 illustrated in Figures 5B-C can be a database table (e.g., in a relational database or any other suitable database) storing filters (e.g., in a filter repository 566 as illustrated in Figure 5A). The filter identifier column 552 provides an index for the data (e.g., to allow the database to identify the data). The filter term column 554 identifies the short term for the filter (e.g., the user-friendly term to use in a user interface, for example as described below with regard to Figures 8A-B). The program module column 556 illustrates human readable program code implementing the filter for a given row. This is merely one example, and other suitable program code can be used.

[0075] Figures 6A-B illustrate data access block structures for privacy centric data security in a cloud environment, according to at least one embodiment. Figure 6A illustrates an example structure 600 for a data access block (e.g., as discussed above), according to at least one embodiment. In an embodiment, a data access block can be used to track, and

enforce, what data a particular data processor is allowed to access, and for how long. For example, a data access block can limit access by data type (e.g., by item) and by duration (e.g., allowing access for a specified period of time). A type column 602 illustrates the type of a field in the data access block and a content column 604 illustrates the content of the corresponding field in that row.

[0076] For example, a data access block can be made up of a signature, followed by a series of blocks. Each block can be made up of a date, a series of one or more items, and a series of one or more filters. The date can be a timestamp denoting the creation of the block. The item(s) can be identifiers for the privacy data item in the data repository. For example, the item(s) can be the value in the item index column 462, illustrated in Figure 4B, corresponding with the data field. The filter can be a filter term (e.g., as illustrated in column 554 illustrated in Figure 5B) or an identifier for the filter in the filter repository (e.g., as illustrated in column 552 illustrated in Figure 5B). The signature can be a signature over the list of blocks. In an embodiment, as discussed above, the signature is generated by the privacy service (e.g., the cloud privacy service 262 illustrated in Figure 2B) using an owner specific secret key (e.g., using public key encryption).

[0077] Figure 6B illustrates an example data access block registry 650 (e.g., the data access block registry 564B illustrated in Figure 5A), according to at least one embodiment. A unique identifier column 652 provides a unique identifier for each data access block. A requestor column 654 provides an identifier for the data processor that requested the data that led to creation of the data access block. A data access block column 656 provides the data access block (e.g., as illustrated in Figure 6A). A block usage column 658 provides a log of usage of the data access block. For example, the block usage column 658 can track each use of the data access block column 656 by the requestor column 654.

[0078] In an embodiment, a log can be used to provide reporting to the user, to identify errors or unauthorized data access, or for other suitable purposes. For example, in an embodiment a privacy service (e.g., the cloud privacy service 262 illustrated in Figure 2) could provide a report to a user of which data processor(s) have accessed the user's private data. This could be provided to the user periodically (e.g., using e-mail or physical mail) or could be accessible upon request at a webpage, within a mobile application, or at any other suitable location.

[0079] Figure 7 illustrate techniques for navigating data access block structures in a privacy centric data security in a cloud environment, according to at least one embodiment. As illustrated, at block 702 a privacy service (e.g., the cloud privacy service 262 illustrated in Figure 2B) receives input parameters. At block 704, the privacy service uses these parameters to generate a subsequent data access block (e.g., as illustrated at blocks 326 and 338 in Figure 3).

[0080] For example, the currentBlock input parameter could be [`<signature>|<timestamp>|<item_index>|<filter_index>`] (e.g., using the data access block structure illustrated in Figure 6A. This represents the previous data access block. The signKey input parameter could be the user's specific private key. The currentDate input parameter could be a timestamp representing the current date and time. The filter parameter could be the new filter to use (e.g., an index to the filter in the filter repository, or a shortname for the filter). The output could be a subsequent block generated based on these inputs, using the human readable code illustrated in block 704. This is merely one example, and other suitable techniques can be used.

[0081] Figures 8A-B illustrate a user interface for privacy centric data security in a cloud environment, according to at least one embodiment. Figure 8A illustrates a user interface to allow a user to set data-processor specific privacy filter controls for privacy centric data security in a cloud environment, according to at least one embodiment. A user device 800 includes an interface 802 that allows a user to graphically set desired security parameters for various data processors. Using graphical icons, text, or any other suitable visual or audio representation, the user can define how entities are allowed to access the user's private data.

[0082] For example, a user can specify that a dentist is allowed to read the user's biographical information (e.g., name, address, phone number) without notification, and the user's dental records without notification, but the dentist must receive approval before accessing the user's medical information. As another example, the user can specify that an insurance company can access the user's biographical and automobile information without notification, but cannot access the user's medical information.

[0083] The user device 800 further includes an interface 804 to provide notification to the user of an access request, and to allow the user to approve or deny the request. For

example, the illustrated embodiment shows an entity “Hypermarket 24/7” (e.g., a grocery store) seeking access to the user’s credit card information. The user can approve or deny the request.

[0084] Figure 8B illustrates a user interface to allow a user to set data specific privacy filter controls for privacy centric data security in a cloud environment, according to at least one embodiment. For example, a user device 800 includes an interface 852 that depicts the user’s private data. The user can use graphical icon, textual representations, audio indications, etc. to control default access to this data. For example, the user can determine that, by default, any entity can access the user’s name and address without notification or approval. The user can further determine that, by default, an entity must receive approval before accessing the user’s credit card information. The user can further determine that, by default, no entity can access the user’s health record.

[0085] As discussed above, each of these defaults can be overridden by a specific privacy filter applied to a specific entity. For example, the user could create a filter (e.g., using the interface illustrated in Figure 8A) to allow a medical provider to access medical data without approval, but to require the medical provider to seek approval before accessing credit card data.

[0086] Figure 9 depicts a cloud computing environment, according to at least one embodiment. It is to be understood that although this disclosure includes a detailed description on cloud computing, implementation of the teachings recited herein are not limited to a cloud computing environment. Rather, embodiments are capable of being implemented in conjunction with any other type of computing environment now known or later developed.

[0087] Cloud computing is a model of service delivery for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, network bandwidth, servers, processing, memory, storage, applications, virtual machines, and services) that can be rapidly provisioned and released with minimal management effort or interaction with a provider of the service. This cloud model may include at least five characteristics, at least three service models, and at least four deployment models.

[0088] Characteristics are as follows:

[0089] On-demand self-service: a cloud consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with the service's provider.

[0090] Broad network access: capabilities are available over a network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

[0091] Resource pooling: the provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to demand. There is a sense of location independence in that the consumer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).

[0092] Rapid elasticity: capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

[0093] Measured service: cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

[0094] Service Models are as follows:

[0095] Software as a Service (SaaS): the capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based e-mail). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application

capabilities, with the possible exception of limited user-specific application configuration settings.

[0096] Platform as a Service (PaaS): the capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including networks, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.

[0097] Infrastructure as a Service (IaaS): the capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

[0098] Deployment Models are as follows:

[0099] Private cloud: the cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on-premises or off-premises.

[00100] Community cloud: the cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on-premises or off-premises.

[00101] Public cloud: the cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.

[00102] Hybrid cloud: the cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

[00103] A cloud computing environment is service oriented with a focus on statelessness, low coupling, modularity, and semantic interoperability. At the heart of cloud computing is an infrastructure that includes a network of interconnected nodes.

[00104] Referring now to Fig. 9, illustrative cloud computing environment 950 is depicted. As shown, cloud computing environment 950 includes one or more cloud computing nodes 910 with which local computing devices used by cloud consumers, such as, for example, personal digital assistant (PDA) or cellular telephone 954A, desktop computer 954B, laptop computer 954C, and/or automobile computer system 954N may communicate. Nodes 910 may communicate with one another. They may be grouped (not shown) physically or virtually, in one or more networks, such as Private, Community, Public, or Hybrid clouds as described hereinabove, or a combination thereof. This allows cloud computing environment 950 to offer infrastructure, platforms and/or software as services for which a cloud consumer does not need to maintain resources on a local computing device. It is understood that the types of computing devices 954A-N shown in Fig. 9 are intended to be illustrative only and that computing nodes 910 and cloud computing environment 950 can communicate with any type of computerized device over any type of network and/or network addressable connection (e.g., using a web browser).

[00105] Referring now to Fig. 10, a set of functional abstraction layers provided by cloud computing environment 950 (Fig. 9) is shown. It should be understood in advance that the components, layers, and functions shown in Fig. 10 are intended to be illustrative only and embodiments of the invention are not limited thereto. As depicted, the following layers and corresponding functions are provided:

[00106] Hardware and software layer 1060 includes hardware and software components. Examples of hardware components include: mainframes; RISC (Reduced Instruction Set Computer) architecture based servers; servers; blade servers; storage devices; and networks and networking components. In some embodiments, software components include network application server software and database software.

[00107] Virtualization layer 1062 provides an abstraction layer from which the following examples of virtual entities may be provided: virtual servers; virtual storage; virtual

networks, including virtual private networks; virtual applications and operating systems; and virtual clients.

[00108] In one example, management layer 1064 may provide the functions described below. Resource provisioning provides dynamic procurement of computing resources and other resources that are utilized to perform tasks within the cloud computing environment. Metering and Pricing provide cost tracking as resources are utilized within the cloud computing environment, and billing or invoicing for consumption of these resources. In one example, these resources may include application software licenses. Security provides identity verification for cloud consumers and tasks, as well as protection for data and other resources. User portal provides access to the cloud computing environment for consumers and system administrators. Service level management provides cloud computing resource allocation and management such that required service levels are met. Service Level Agreement (SLA) planning and fulfillment provide pre-arrangement for, and procurement of, cloud computing resources for which a future requirement is anticipated in accordance with an SLA.

[00109] Workloads layer 1066 provides examples of functionality for which the cloud computing environment may be utilized. Examples of workloads and functions which may be provided from this layer include: mapping and navigation; software development and lifecycle management; virtual classroom education delivery; data analytics processing; transaction processing; and a cloud privacy service. In an embodiment, some or all of the modules of the cloud system 250 could be implemented in the workloads layer 1066. For example, the cloud privacy service 262, the filter generation service 263, the data repository 264, and the privacy filter repository 266, could be implemented in the workloads layer 1066. In an embodiment, the cloud privacy service 262 could execute on a computing system in the cloud (e.g., in the workloads layer 1066) and the data repository 264, and the privacy filter repository 266 stored on the cloud. Doing so allows access to this information from any computing system attached to a network connected to the cloud (e.g., the Internet).

[00110] The descriptions of the various embodiments of the present invention have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described

embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed herein.

[00111] In the following, reference is made to embodiments presented in this disclosure. However, the scope of the present disclosure is not limited to specific described embodiments. Instead, any combination of the following features and elements, whether related to different embodiments or not, is contemplated to implement and practice contemplated embodiments. Furthermore, although embodiments disclosed herein may achieve advantages over other possible solutions or over the prior art, whether or not a particular advantage is achieved by a given embodiment is not limiting of the scope of the present disclosure. Thus, the following aspects, features, embodiments and advantages are merely illustrative and are not considered elements or limitations of the appended claims except where explicitly recited in a claim(s). Likewise, reference to “the invention” shall not be construed as a generalization of any inventive subject matter disclosed herein and shall not be considered to be an element or limitation of the appended claims except where explicitly recited in a claim(s).

[00112] Aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “circuit,” “module” or “system.”

[00113] The present invention may be a system, a method, and/or a computer program product. The computer program product may include a computer readable storage medium (or media) having computer readable program instructions thereon for causing a processor to carry out aspects of the present invention.

[00114] The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-

exhaustive list of more specific examples of the computer readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing. A computer readable storage medium, as used herein, is not to be construed as being transitory signals *per se*, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.

[00115] Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing/processing device receives computer readable program instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing/processing device.

[00116] Computer readable program instructions for carrying out operations of the present invention may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Smalltalk, C++ or the like, and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the

user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present invention.

[00117] Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

[00118] These computer readable program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. These computer readable program instructions may also be stored in a computer readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the function/act specified in the flowchart and/or block diagram block or blocks.

[00119] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute

on the computer, other programmable apparatus, or other device implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

[00120] The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

[00121] While the foregoing is directed to embodiments of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.