



(51) International Patent Classification:  
**G06F 21/00** (2006.01) **G06F 7/06** (2006.01)

(21) International Application Number:  
PCT/US2009/065019

(22) International Filing Date:  
18 November 2009 (18.11.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
61/115,633 18 November 2008 (18.11.2008) US

(71) Applicant (for all designated States except US): **WORK-SHARE TECHNOLOGY, INC.** [US/US]; 208 Utah Street, Suite 350, San Francisco, California 94103 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **MORE, Scott** [US/JP]; 4-29-405 Ichigayanakanochi, Shinjuku-Ku, Tokyo 162-0064 (JP). **BEYER, Ilya** [US/US]; 3705 Kingridge Drive, San Mateo, California 94403 (US).

(74) Agents: **MCKEE, Ian A.** et al.; Perkins Coie LLP, P.O. Box 1208, Seattle, Washington 98111-1208 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: METHODS AND SYSTEMS FOR EXACT DATA MATCH FILTERING

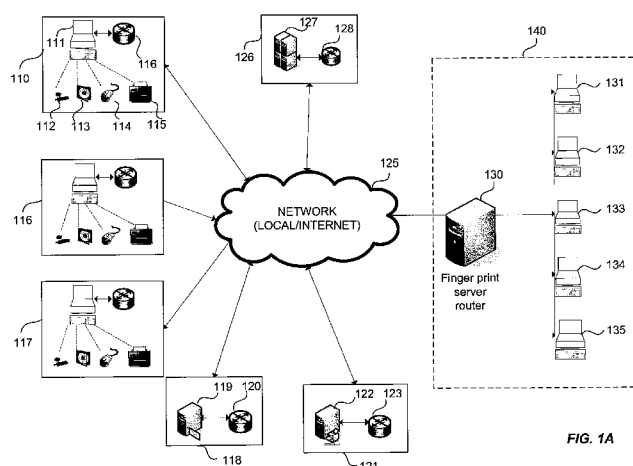


FIG. 1A

(57) Abstract: A technique for efficiently preventing exact data words ("entities") from unauthorized disclosure is disclosed. Protect agents installed at various egress points identify candidate entities from digital information desired to be disclosed by a user. The candidate entities are compared against registered entities stored in a lightweight entity database (LWED). If a candidate entity matches against a registered entity in the LWED, the protect agent initiates a security action. Alternately, the protect agent transmits the matching candidate entity to a global entity database (GED) server to receive additional confirmation on whether the candidate entity matches a registered entity. In some instances, the protect agent also receives (from the GED server) metadata information associated with the matching candidate entity. The protect agent utilizes the metadata information to initiate suitable security actions.

**METHODS AND SYSTEMS FOR EXACT DATA MATCH FILTERING****CROSS-REFERENCE TO RELATED APPLICATIONS**

**[0001]** This application claims priority to U.S. Provisional Patent Application  
5 No. 61/115,633 filed November 18, 2008, which is hereby incorporated by  
reference in its entirety.

**[0002]** This application is related to U.S. Application no. 12/177,043,  
entitled "METHODS AND SYSTEMS TO FINGERPRINT TEXTUAL  
INFORMATION USING WORD RUNS," filed July 21, 2008, and to U.S.

10 Application no. 12/209,082, entitled "METHODS AND SYSTEMS FOR PROTECT  
AGENTS USING DISTRIBUTED LIGHTWEIGHT FINGERPRINTS," filed  
September 11, 2008, both of which are incorporated by reference as if fully set  
forth herein.

15 **FIELD OF THE INVENTION**

**[0003]** The present invention relates to information security and more  
specifically relates to systems and methods for detecting and preventing  
unauthorized disclosure of secure information. Furthermore, the present invention  
pertains to methods and systems for exact data match filtering for structured data.

20

**BACKGROUND**

**[0004]** With the rapid increase and advances in digital documentation  
services and document management systems, organizations are increasingly  
storing important, confidential, and secure information in the form of digital  
25 documents. Unauthorized dissemination of this information, either by accident or  
by wanton means, presents serious security risks to these organizations.

Therefore, it is imperative for the organizations to protect such secure information and detect and react to any secure information from being disclosed beyond the perimeters of the organization.

**[0005]** Additionally, the organizations face the challenge of categorizing and

5 maintaining the large corpus of digital information across potentially thousands of data stores, content management systems, end-user desktops, etc. It is therefore important to the organization to be able to store concise and lightweight versions of fingerprints corresponding to the vast amounts of image data. Furthermore, the organizations face the challenge of categorizing and maintaining the large corpus  
10 of digital information across potentially thousands of data stores, content management systems, end-user desktops, etc. One solution to this challenge is to generate fingerprints from all of the digital information that the organization seeks to protect. These fingerprints tersely and securely represent the organization's secure data, and can be maintained in a database for later  
15 verification against the information that a user desires to disclose. When the user wishes to disclose any information outside of the organization, fingerprints are generated for the user's information, and these fingerprints are compared against the fingerprints stored in the fingerprint database. If the fingerprints of the user's information matches with fingerprints contained in the fingerprint server, suitable  
20 security actions are performed.

**[0006]** However, the user has at his disposal myriad options to disclose the information outside of the organization's protected environment. For example, the user could copy the digital information from his computer to a removable storage medium (e.g., a floppy drive, a USB storage device, etc.), or the user could email  
25 the information from his computer through the organization's email server, or the

user could print out the information by sending a print request through the organization's print server, etc.

**[0007]** Additionally, in many organizations, sensitive data is stored in databases, including account numbers, patient IDs, and other well-formed, or “structured”, data. The amount of this structured data can be enormous and ease of unwanted distribution across the egress points creates security problems for organizations.

**[0008]** The exact data match problem can be thought of as a massive, multi-keyword search problem. Methods for exact keyword match include Wu-Manber and Aho-Corasick. However, these methods are disadvantageous because they do not scale beyond several thousand keywords in space or time.

**[0009]** Full blown databases can be employed for exact data matches, but they do not scale down to Agents residing on Laptops. There are also security concerns with duplicating all the confidential cell data within an organization directly.

**[0010]** A more general approach can be taken where the pattern of each category of structured data is inferred and searched via regular expressions or a more complex entity extraction technique. However, without the actual values being protected, this approach would lead to many false positives.

**SUMMARY**

**[0011]** Introduced here and described below in detail are methods and systems for exact data match filtering. In one embodiment, an organization's digital information is scanned to retrieve "sensitive" candidate entities. These sensitive entities correspond to structured data words (e.g., social security numbers, patient IDs, etc.) that the organization desires to protect from unauthorized disclosure. In some instances, the candidate entities are identified on the basis of word-patterns and/or heuristic rules. The identified candidate entities are optionally converted to a canonical format to enable the data match inspection engine to be impervious to changes in character encoding, digital format, etc. The candidate entities are then stored as registered entities in an entity database. In some instances, the entity database is a lightweight entity database (LWED) that supports a compressed version of the registered entities. The database compression can be achieved by storing the candidate entities in a data structure that supports membership query while introducing a small error of false positives in the membership query (e.g., a Bloom filter). In some instances, the entity database is a global entity database (GED) that is stored in association with a remote server. The GED includes an uncompressed version of the registered entities (or corresponding hash-values of the entities), and also includes metadata information associated with each of the registered entities.

**[0012]** Protect agents are installed across several egress points (laptop, mail server, etc.) to monitor information being disclosed by a user. The protect agents receive digital information (e.g., textual information) that a user wishes to disclose using the egress point, and identifies candidate entities from the textual information. In one embodiment, the protect agent looks up the candidate entities against registered entities stored in the LWED. If the protect agent detects any

matching candidate entities, the protect agent initiates an appropriate security action. In some embodiments, the protect agent communicates with a remote GED server (containing the GED). In such embodiments, the protect agent transmits the matching candidate entities to the GED server, where the candidate entities are again matched against the registered entities in the GED. The results of the GED comparison eliminate or reduce any false positives that may have resulted from the comparison of the candidate entities against the LWED. In some instances, the GED also supplies the protect agent with metadata associated with the matching candidate entities. The metadata information is useful in initiating various types of security actions.

**[0013]**      BRIEF DESCRIPTION OF THE DRAWINGS

**[0014]**      One or more embodiments of the present invention are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements.

5    **[0015]**      Fig. 1A illustrates an example of an overall setup to implement protect agents for exact data match filtering.

**[0016]**      Fig. 1B is a high-level block diagram showing an example of the architecture of an egress point or an entity server.

10 **[0017]**      Fig. 2 is a flow diagram depicting a process for registering the entities for the digital information maintained by an organization.

**[0018]**      Figure 3A is a block diagram illustrating an exemplary architecture of an egress point configured to operate a protect agent.

**[0019]**      Figure 3B is a flow diagram illustrating an exemplary inspection process performed by the protect agent

15 **[0020]**      Fig. 4 is a flow diagram illustrating various mechanisms used by the registration process and the inspection process to identify a candidate entity.

**[0021]**      Figure 5 is a flow diagram depicting an exemplary process 500 for comparison of the received candidate entities.

**DETAILED DESCRIPTION**

**[0022]** The present invention may be embodied in several forms and manners. The description provided below and the drawings show exemplary embodiments of the invention. Those of skill in the art will appreciate that the invention may be embodied in other forms and manners not shown below. It is understood that the use of relational terms, if any, such as first, second, top and bottom, and the like are used solely for distinguishing one entity or action from another, without necessarily requiring or implying any such actual relationship or order between such entities or actions. References in this specification to “an embodiment”, “one embodiment”, or the like, mean that the particular feature, structure or characteristic being described is included in at least one embodiment of the present invention. Occurrences of such phrases in this specification do not necessarily all refer to the same embodiment.

**[0023]** Fig. 1A illustrates an example of an overall setup to implement protect agents for exact data match filtering. One of the means by which a user can disclose digital information outside of the organization's perimeter is by disclosing the information through his computer system 110. Examples of such a computer system include a desktop computer, a laptop, a PDA or any such device that allows a user to access the organization's information. In one embodiment, the computing system 110 is connected to a network 125. Here, the computing system 110 comprises the desktop/laptop computer 111 through which the user accesses the organization's secure information. The user would be able to transfer information outside of the organization by transferring the information to any medium connected to the computer.

**[0024]** Such points (i.e., computer hardware) through which information can be transferred outside of the organization's protected environment are called

egress points. Examples of transferring data at egress points include copying the information from the computer to a CD disk 112 or any other optical storage medium, copying the information to a floppy drive 113 or any other tape medium, copying the information to a USB key 114 or other flash based storage medium, transferring the information by printing the information using a printer 115, copying information to the clipboard 115a of the local operating system, etc. In such an event, all the information that is transmitted through the computer 111 needs to be monitored to ensure that secure or sensitive information does not get transferred.

**[0025]** The information to be monitored may include digital textual data,

image data, multimedia data etc. Such digital information can be monitored by using, for example, fingerprinting technology to be enable registration and inspection of a large corpus of data. Examples of such fingerprinting technology are described in detail in related applications U.S. Application no. 12/177,043, entitled "METHODS AND SYSTEMS TO FINGERPRINT TEXTUAL

INFORMATION USING WORD RUNS," filed July 21, 2008, and U.S. Application no. 12/209,082, entitled "METHODS AND SYSTEMS FOR PROTECT AGENTS USING DISTRIBUTED LIGHTWEIGHT FINGERPRINTS," filed September 11, 2008, both of which are incorporated by reference in their entireties herein. The fingerprinting technology described in the above applications uses various

techniques to protect the large corpus an organization's confidential information.

In one example, the fingerprinting technology detects sentences, or even paragraphs, in original or derivative forms, and prevents such textual information from being disclosed. However, such fingerprinting technology may not be an effective tool for protection of "exact data words." An "exact data word," as

described herein, refers to any combination of characters (e.g., alphabets, numbers, symbols, etc.) that form a structured word. Such exact data words may

exist, for example, in the form of patient IDs in a hospital database, social-security numbers, or employees' date-of-birth information, phone numbers, etc. In some instances, such exact data words have a well-structured format or pattern (e.g., social security numbers have a pattern that includes seven numerical characters and two "-" symbols separating groups of the numerical characters). These exact data words may be spread across various documents that constitute the organization's digital information (e.g., in textual data, embedded in images, etc.). When such exact data words are confidential they need to be protected from unauthorized disclosure. To achieve this, the following sections describe techniques for identifying such exact data words, and preventing the exact data words from unauthorized disclosure through any of the egress points.

**[0026]** Returning to Fig. 1A, the various egress points of the computer 111 are monitored to detect any activity that purports to disclose information through the egress points. A software agent, called the protect agent 116, is run on the computer 111 to monitor activity at the egress points (112, 113, 114, 115, 115a) associated with the computer 111. If the organization supports more than one computer system, each of these computer systems (110, 116, 117, 118) have protect agents installed on them to ensure that the activity on each of the computer systems is monitored. In one embodiment, the protect agent 116 is a set of computer instructions or a computer implemented program available on a memory location (e.g., on a magnetic tape drive, a flash memory drive, etc.) at the site of the protect agent 116. In some instances, the protect agent can be implemented by using programmable circuitry programmed by software and/or firmware, or by using special-purpose hardwired circuitry, or by using a combination of such embodiments.

**[0027]** In addition to being installed in every computer system (110, 116, 117, 118) in the network, the protect agents are also installed on other vulnerable egress points across the organization. One example of such a vulnerable egress point includes one or more email server systems 118 connected to the network.

5 The email server 119 handles and routes the emails sent out and received by the organization. The protect agent 120 installed on the email server 119 monitors the emails desired to be sent out of the organization through the email server.

Another example of a vulnerable egress point could be a print server 121 connected to the organization's network. A protect agent 123 connected to the  
10 print server 122 monitors print jobs sent by the users to the printers connected to the network.

**[0028]** Additional examples of vulnerable egress points include network appliance systems 126. Here, a protect agent 128 is installed in each network appliance 127 to ensure that information disclosed through a particular network  
15 appliance 127 is monitored. Examples of using network appliances 126 to transfer data include sharing of data over a network share medium, data transferred at the socket or TCP layer of the network, etc. It is understood that in addition to these examples, the egress points also include other porous environments through which information can be disclosed by the user beyond the  
20 secure environment of the organization.

**[0029]** In one embodiment, a lightweight entity database (LWED) 118 is provided locally at the site at which each of the protect agents is installed (e.g., the user's desktop/laptop computer, one of the network appliances, etc.). As will be explained in detail below, in one embodiment, the LWED is a compressed  
25 database that includes registered entities. An entity, as described herein, refers to an exact data word. A registration process scans the organization's digital

information to extract entities (i.e., exact data words that need to be protected against unauthorized disclosure) and registers them in a database. The entities registered into such a database are referred to as "registered entities." As will be described in detail below, the database may be a global database (GED), or an  
5 LWED (which is, for example, a compressed version of the GED).

**[0030]** In one embodiment, at least one redundant copy of the LWED is stored locally at the site of each protect agent 116 such that the protect agent can access or communicate with the LWED even when the protect agent is not connected to any network. For example, a protect agent 116 implemented on a  
10 user's laptop computer monitors the activity at all egress points of the user's laptop computer (e.g., 112, 113, 114, etc.) and prevents unauthorized disclosure of information from the laptop computer through the egress points, even if the laptop computer is not connected to any network (e.g., the organization's local network, the public internet, etc.).

**[0031]** In one illustrative embodiment, the computer systems and all other systems representing egress points (the egress point systems) are centrally connected to a network 125. In one embodiment, the network includes a local network. This includes a network that is managed and maintained locally by the organization. In another embodiment, the network could also be the internet. In  
20 the case of the internet, each of the egress point systems could be directly and individually connected to the internet, or could be connected to a local network or a cluster of local networks, with each of the local networks communicating with each other through the internet. Other combinations of the egress point systems within the local network and the internet are possible and such combinations will  
25 be apparent to a person of skill in the art.

**[0032]** In one embodiment where the egress point systems are connected to the network, one or more entity servers (e.g., 131, 132, 133, 134, 135) are connected to the network. The entity server (e.g., 131) is coupled to the GED (that holds the uncompressed version of the registered entities). In one example, 5 each of the entity servers (131, 132, 133, 134, 135) is connected directly to the network. In another example, each of the entity servers (131, 132, 133, 134, 135) is connected to a entity server router 130.

**[0033]** The functions of the entity server router 130 may include, for example, routing requests from a protect agent 116 to the least busy entity server, 10 collecting performance statistics of the entity servers (131, 132, 133, 134, 135) to determine the load on each entity server (such that a request from a protect agent can be routed to the least busy entity server, synchronization and version control of the GED at each entity server, etc.).

**[0034]** In one embodiment, the entity servers (131, 132, 133, 134, 135) 15 could be located at different geographical locations (not shown in Fig. 1A) and connect to the entity server router 130 through the network. This distributed model would allow organizations to run protect agents with minimal performance lag across geographically diverse locations, such that information from the protect agents are routed to the most optimal entity server. It should be noted that the 20 entity server router is not imperative to maintaining a distributed entity server array. Any other means known in the art through which a distributed network can be achieved can be employed in the place of the entity server router 130.

**[0035]** In the case of the public internet, the entity servers (e.g., 131) function as hosted entity servers. A hosted entity server is publicly accessible 25 over the internet. One advantage of using a hosted entity server is that an organization does not have to deploy and manage one or more server appliances

within its networks (for the purpose of holding a GED). Some small organizations may not even have infrastructure to maintain a network and host an entity server, but may still require their secure information to be protected. In such cases, the support and manageability of the entity server can be done by even a third party provider that provides the service of a hosted entity server.

**[0036]** A provider offering a hosted registered entity service can also support multi-tenancy services, whereby the provider shares the hosted entity server's resources across different organizations. In one embodiment, this would allow GEDs for multiple organizations to reside on the same server.

**[0037]** It is emphasized that the network 125 and entity servers 140 depicted in Fig. 1A are for illustrative purposes only, and that a network 125 or a entity server setup 140 is not essential for a protect agent 116 to perform an entity lookup. For example, the protect agent 116 may purely rely on the LWED 118 to perform the entity lookup.

**[0038]** Now refer to Fig. 1B, which is a high-level block diagram showing an example of the architecture of an egress point (e.g., 111) or an entity server (e.g., 131). The egress ping (e.g., 111) or the entity server (e.g., 131) includes one or more processors 1201 and memory 1202 coupled to an interconnect 1203. The interconnect 1203 shown in Fig. 1B is an abstraction that represents any one or more separate physical buses, point to point connections, or both, connected by appropriate bridges, adapters, or controllers. The interconnect 1203, therefore, may include, for example, a system bus, a Peripheral Component Interconnect (PCI) bus or PCI-Express bus, a HyperTransport or industry standard architecture (ISA) bus, a small computer system interface (SCSI) bus, a universal serial bus (USB), IIC (I2C) bus, or an Institute of Electrical and Electronics Engineers (IEEE) standard 1394 bus, also called "Firewire".

**[0039]** The processor(s) 1201 is/are the central processing unit (CPU) of egress point (e.g., 111) or the entity server (e.g., 131) and, thus, control the overall operation of the egress point (e.g., 111) or the entity server (e.g., 131). In certain embodiments, the processor(s) 1201 accomplish this by executing  
5 software or firmware stored in memory 1202. The processor(s) 1201 may be, or may include, one or more programmable general-purpose or special-purpose microprocessors, digital signal processors (DSPs), programmable controllers, application specific integrated circuits (ASICs), programmable logic devices (PLDs), trusted platform modules (TPMs), or the like, or a combination of such  
10 devices.

**[0040]** The memory 1202 is or includes the main memory of the egress point (e.g., 111) or the entity server (e.g., 131). The memory 1202 represents any form of random access memory (RAM), read-only memory (ROM), flash memory, or the like, or a combination of such devices. In use, the memory 1202 may  
15 contain, among other things, code 1207 embodying the the protect agent 116.

**[0041]** Also connected to the processor(s) 1201 through the interconnect 1203 are a network adapter 1204 and a storage adapter 1205. The network adapter 1204 provides the egress point (e.g., 111) or the entity server (e.g., 131) with the ability to communicate with remote devices over the interconnect 1203  
20 and may be, for example, an Ethernet adapter or Fiber Channel adapter.

**[0042]** Detailed information on how the protect agents at each egress point secure the entities from unauthorized disclosure is provided with reference to Figs. 2-5 below.

**[0043]** Figure 2 is a flow diagram depicting a process 200 for registering the  
25 entities for the digital information maintained by an organization. At step 201, the process 200 receives digital information (e.g., textual information, image

information, etc.). At step 203, the process 200 then parses the received information to identify any potential entities (i.e., candidate entities). In one embodiment, where the received information is in the form of image data, image conversion techniques (e.g., optical character recognition, etc.) may be used to retrieve the text information from such data. The process 200 may use one of several techniques (e.g., pattern recognition, regular expression matching, etc.) to identify the entities. These techniques are described in detail with reference to Fig. 4 below.

**[0044]** In some instances, as indicated in step 205, the candidate entities are optionally normalized to a canonical format. This can be done by converting the candidate entities into one of several raw text formats (e.g., UTF-16 format). By doing this, the protect agent (at a later inspection stage) will be impervious to differences in character encodings, data formats, case folding, etc. in the candidate entities identified during the inspection stage.

**[0045]** The process 200 then proceeds to register the candidate entities within the LWED and/or the GED. At step 207, the process 200 registers the candidate entities into the LWED. Since the LWED is stored at each egress point, the overall size of the database is controlled using one or more techniques. In one embodiment, the candidate entities are converted to hash values before being registered into the LWED. One example of generating a value hash is to compute a hash based function over every character of a word and generating an integer value corresponding to that word. In another embodiment, the candidate entities are compressed by storing them in a data structure that supports membership query while introducing a small probability of false positives in the membership query. An example of such a data structure is a Bloom filter, where a large bit vector and multiple hash functions are used to determine whether a candidate

entity being inspected may potentially be present in the LWED. The Bloom filter is implemented using a sequence of software instructions as indicated by an algorithm, and such software is physically stored at a physical memory location at the site of the protect agent. The implementation of the Bloom filter itself is widely known in the art and a person of ordinary skill in the art would be able to reproduce the functions of a Bloom filter to generate the LWED as indicated in this embodiment.

**[0046]** The process 200 also optionally includes the generation of a GED which may be stored, for example, in a remote server (e.g., 131 of Fig. 1A). At step 209, the process 200 stores the candidate entities within the GED. In some instances, the process 200 may convert the candidate entities to hash values before registering the candidate entities. The process 200 may also incorporate metadata information along with the candidate entities while registering the candidate entities. Such metadata information is valuable for auditing and self-remediation purposes. Examples of the uses of metadata information include, for example, identifying the source document associated with the candidate entity, categorizing the entities according to entity type (e.g., patient ID numbers, social security numbers, etc.), associating the entity with a particular risk level, etc.

**[0047]** Figure 3A is a block diagram illustrating an exemplary architecture of an egress point 110 configured to operate a protect agent 116 to inspect data being disclosed through the egress point 110. The protect agent 116 includes a receiving module 302, candidate ID module 304, comparison module 306, a communication module 308, and a security action module 310. As described above, the protect agent 116 can be implemented by using programmable circuitry programmed by software and/or firmware, or by using special-purpose hardwired circuitry, or by using a combination of such embodiments. In some instances, the

protect agent 116 is implemented as a unit in the processor 1201 of the egress point 110.

**[0048]** The receiving module 302 is configured to receive the data a user desires to disclose through the egress point 110. This data includes, for example, digital text information. The candidate ID module 304 of the protect agent 116 receives the digital information, and identifies candidate entities from the digital information. Detailed information on identifying candidate entities is provided with reference to Figure 4 below. The candidate entities may be every word identified in the digital information, or may be words that match a particular format (as, for example, identified by a regular expression matcher). In one embodiment, the candidate ID module 304 may optionally convert the candidate entities to a canonical format, to ensure that the candidate entities are impervious to digital format, character encoding, case-folding, etc. In some embodiments, the candidate ID module 304 may also optionally convert the candidate entities to equivalent hash values (e.g., with the same hashing algorithm used during the registration of the organization's candidate entities). In some instances, the candidate ID module 304 may optionally detect and record an entity type (e.g., social security number type, patient ID type, etc.) of the candidate entity based on the format of the candidate entity

**[0049]** The comparison module 306 receives the candidate entities from the candidate ID module 304 and compares the candidate entities with registered entities stored in an entity database. In some instances, the entity database is the LWED stored locally at the site of the egress point 110. The comparison module 206 detects the presence of candidate entities that match any of the registered entities. In some instances, the comparison module 306 directly supplies the list of matching entities to the security action module 310 for further action. In some

instances, the comparison module 306 may communicate with a remote server (containing the GED) using a communication module 308 to compare the matching entities (received from the comparison against the LWED) against the registered entities stored in the GED. In this manner, the comparison module 306  
5 can eliminate or at least reduce any false positives that may result from the comparison against the LWED. Additionally, by sending only those candidate entities identified as matching entities to the GED, the server holding the GED has to process only a limited number of candidate entities (as opposed to processing all the candidate entities identified in a textual information). This results in  
10 reduced latency time in receiving the final matching results from the GED server. Additionally, in such instances, the GED supplies the comparison module 306 with metadata information associated with the matching entities for further processing.

**[0050]** In some instances, the comparison module 306 may directly communicate with the remote server (i.e., the GED) in lieu of comparing the  
15 candidate entities with the LWED. In some instances, the comparison module 306 may utilize the entity type recorded by the candidate ID module 304 to compare the candidate entity only against a subset of registered entities (instead of the entire database of registered entities) that are tagged (e.g., according to their metadata information in the GED) under a similar entity type. This  
20 comparison, according to entity type of the candidate entity, further helps in reducing latency/processing time of the comparison process.

**[0051]** The results of the comparison are provided to the security action module 310, which proceeds to initiate an appropriate security action. In some instances, the security action module 310 utilizes metadata retrieved from, for  
25 example, the GED, to initiate various types of security actions. Examples of such security actions include preventing the information from being transmitted out

through the associated egress point, sending out a security alert to a system administrator, revoking the user's access to the particular information, alerting the user of the security violation, etc. The security actions may also include integration with third party software to offer security solutions (e.g., integration with

5 Microsoft Windows® RMS to apply rights management to the information being disclosed). It is understood that these examples of security actions are provided for illustrative purposes only, and that other security actions known to people skilled in the art are equally applicable here.

**[0052]** Figure 3B is a flow diagram illustrating an exemplary inspection

10 process 300 performed by the protect agent 116. At step 312, the process 300 receives digital information (e.g., textual data). At step 314, the process 300 identifies one or more candidate entities from the received textual information. As described above, the process 300 may optionally record the entity type of the candidate entities, and may also convert the candidate entities to a canonical

15 format and/or hash values. If the process 300 does not identify any candidate entities at step 314, the process 300 returns and may repeat the process of receiving textual information for inspection.

**[0053]** At step 316, the process 300 looks up the candidate entities against registered entities in an entity database. As described above, the entity database

20 may be an LWED and/or the GED. At 318, the process 300 determines whether the candidate entities match against one of the registered entities in the entity database. If the process 300 determines that at least one of the candidate entities matches against the registered entities, the process 300 proceeds to step 320 to perform a security action. As discussed above, the process 300 may use

25 metadata information retrieved from the entity database to initiate appropriate security actions.

**[0054]** Figure 4 is a flow diagram illustrating various mechanisms used by the registration process (e.g., step 205 of Figure 2) and the inspection process (e.g., step 314 of Figure 3B) to identify a candidate entity. The process receives textual information at step 402, and proceeds to identify candidate entities at step 5 404. As illustrated in Fig. 4, the candidate entity may be chosen according to one or more of the following identification schemes. Step 406 represents identification scheme 1, where the process employs an entity format checker to identify word-patterns or word-formats. For example, a regular expression matcher may be used to identify regular expressions (e.g., a social security number expression 10 structured according to a particular pattern) in the received textual information. Accordingly, at step 412, the process records any entity in the textual data that satisfies the particular word-pattern or word-format targeted by the entity format checker.

**[0055]** Step 408 represents identification scheme 2, where the process 15 employs one or more heuristic rules to exclude or skip over non-entity words. In a first example, the heuristic rule may define stop words that can be skipped over. Examples of stop words include words that commonly occur in the language (e.g., prepositions, etc.), common words considered non-confidential by the organization (e.g., address information, disclaimer language included by default in patient 20 admittance forms, etc.). In a second example, the heuristic rule may require any words shorter than the shortest word in the entity database (or longer than the longest word in the entity database) to be excluded from consideration as a candidate entity. Other similar heuristic rules, as appreciated by a person of ordinary skill in the art, can also be employed in implementing the identification 25 scheme 2 described herein. As indicated in step 414, the words that are not excluded by the heuristic rule are submitted as candidate entities.

**[0056]** Step 410 represents identification scheme 3, where every word (e.g., every set of characters demarcated by one or more spaces) in the received textual information is treated as a candidate entity. It is understood that a person of ordinary skill in the art may combine one or more of these identification schemes, or add other identification schemes that are readily apparent to such a person, to improve the efficiency of the candidate entity identification process.

**[0057]** Figure 5 is a flow diagram depicting an exemplary process 500 for comparison of the received candidate entities. At step 500, the process identifies the candidate entities that need to be compared against the registered entities in the entity database. At step 504, the process 500 matches the candidate entities against registered entities in an LWED located at the site of the egress point. Using this lookup, the process 500 generates a list of candidate entities that match against any of the registered entities. In some instances, the process 500 directly communicates this information to a security action module to initiate appropriate security action. In other instances, as indicated in step 506, the process 500 determines whether the egress point is connected to the network. If the egress point is not connected to the network, the process 500 proceeds to step 512, where the process 500 initiates an appropriate security action.

**[0058]** If the egress point is connected to the network, then the process 500 transmits the matching candidate entities to the remote server holding the GED. The GED server compares the received candidate entities against the registered entities in the GED. This allows the process 500 to eliminate or reduce the number of false positives that may have been identified by the comparison against LWED. Additionally, by sending only those candidate entities identified as matching entities to the GED, the GED server has to process only a limited number of candidate entities (as opposed to processing all the candidate entities

identified in a textual information). This results in reduced latency time in receiving the final matching results from the GED server. Additionally, the GED server may also return metadata information associated with the matching candidate entities. The process 500 then proceeds to step 512 to initiate one or  
5 more security actions.

**[0059]** It is emphasized, however, that in some embodiments, the process 500 may operate by matching the candidate entities exclusively against the LWED (i.e., by initiating the security action subsequent to comparison of the candidate entities against the registered entities in the LWED). In other embodiments, the  
10 process 500 may operate by matching the candidate entities exclusively against the GED (i.e., by directly comparing the candidate entities against the GED instead of the LWED).

**[0060]** The techniques introduced above can be implemented by programmable circuitry programmed or configured by software and/or firmware, or  
15 entirely by special-purpose circuitry, or in a combination of such forms. Such special-purpose circuitry (if any) can be in the form of, for example, one or more application-specific integrated circuits (ASICs), programmable logic devices (PLDs), field-programmable gate arrays (FPGAs), etc.

**[0061]** Software or firmware for implementing the techniques introduced  
20 here may be stored on a machine-readable storage medium and may be executed by one or more general-purpose or special-purpose programmable microprocessors. A "machine-readable medium", as the term is used herein, includes any mechanism that can store information in a form accessible by a machine (a machine may be, for example, a computer, network device, cellular  
25 phone, personal digital assistant (PDA), manufacturing tool, any device with one or more processors, etc.). For example, a machine-accessible medium includes

recordable/non-recordable media (e.g., read-only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; etc.), etc.

**[0062]** The term "logic", as used herein, can include, for example, special-  
5 purpose hardwired circuitry, software and/or firmware in conjunction with  
programmable circuitry, or a combination thereof.

**[0063]** Although the present invention has been described with reference to  
specific exemplary embodiments, it will be recognized that the invention is not  
limited to the embodiments described, but can be practiced with modification and  
10 alteration within the spirit and scope of the appended claims. Accordingly, the  
specification and drawings are to be regarded in an illustrative sense rather than a  
restrictive sense.

CLAIMS

1. A method for preventing unauthorized disclosure of secure information, the method comprising:

- 5 receiving, by a protect agent installed at a first egress point, digital information including a first text, the first text including a plurality of words;
- identifying, by the protect agent, a first candidate entity, the first candidate entity corresponding to a particular word of the plurality of words;
- comparing, by the protect agent, the first candidate entity against a plurality
- 10 of registered entities stored in an entity database; and
- performing, by the protect agent, a security action when the first candidate entity matches against a particular registered entity of the plurality of registered entities.

- 15 2. The method of claim 1, wherein the identification of the first candidate entity from the plurality of words further comprises:
- utilizing an entity format matcher to identify a first word from the plurality of words that matches a particular word-pattern.

- 20 3. The method of claim 1, wherein the identification of the first candidate entity from the plurality of words further comprises:
- utilizing a heuristics engine to skip over one or more words from the plurality of words based on a heuristic rule.

- 25 4. The method of claim 4, wherein the heuristic rule includes one or more of:
- skipping over a first word from the plurality of words when the first word matches a first stop word of a plurality of stop words;

skipping over a second word from the plurality of words when the second word has a word-length that is shorter than a first word-length of a shortest registered entity of the plurality of registered entities; or

5 skipping over a third word from the plurality of words when the third word has a word-length that is longer than a second word-length of the longest registered entity of the plurality of registered entities.

5. The method of claim 1, wherein the identification of the first candidate entity from the plurality of words further comprises identifying a first entity type  
10 associated with the candidate entity.

6. The method of claim 5, wherein the plurality of registered entities stored in the entity database are categorized according an entity type associated with each of the plurality of registered entities.

15

7. The method of claim 6, wherein the comparison of the first candidate entity against the plurality of registered entities further comprises:

identifying a subplurality of registered entities that are categorized based on the first entity type;

20 comparing the first candidate entity against the subplurality of registered entities that are categorized based on the first entity type.

8. The method of claim 1, wherein the first candidate entity is converted to a canonical format prior to being compared against the plurality of registered  
25 entities, wherein the canonical format causes the protect agent to be impervious to differences in digital format and character encoding of the first candidate entity.

9. The method of claim 1, wherein the plurality of entities stored in the entity database correspond to entity words that are desired to be secured from unauthorized distribution.

5

10. The method of claim 9, wherein the entity database is a lightweight entity database (LWED) located at the site of the first egress point.

11. The method of claim 10, wherein the LWED includes a compressed  
10 database to hold the plurality of registered entities.

12. The method of claim 11, wherein the compressed database is generated by:

generating hash-values for each registered entity of the plurality of  
15 registered entities; and  
storing the generated hash-values in a probabilistic data structure.

13. The method of claim 12, wherein the probabilistic data structure is a Bloom filter or a derivative thereof.

20

14. The method of claim 11, wherein, when the first candidate entity matches against a first registered entity in the LWED, the method further comprises:

transmitting, by the protect agent, the first candidate entity to a global entity database (GED), the GED including an uncompressed version of the plurality of  
25 registered entities; and

receiving, by the protect agent, a confirmation on whether the first candidate entity matches against a second registered entity in the LWED.

15. The method of claim 9, wherein the entity database is a global entity  
5 database (GED), the GED including the plurality of registered entities.

16. The method of claim 15, wherein the GED is stored in association with a remote server, and wherein the protect agent at the first egress point communicates with the GED utilizing a network.

10

17. The method of claim 16, wherein each of the plurality of registered entities in the GED is represented using a corresponding hash-value.

18. The method of claim 1, wherein each of the plurality of registered entities in  
15 the GED includes metadata information, the metadata information for a given registered entity including one or more of:

an entity type associated with the given registered entity;  
a location of the given registered entity within a particular document; or  
an origin information of the particular document.

20

19. The method of claim 1, wherein the security action includes one or more of:  
preventing the first text from being disclosed through the first egress point;  
logging the event as a security violation;  
requiring a password from a user to allow the first text to be disclosed;  
25 blocking the user's access to the first text;  
sending out a security alert; or

integration of the first text with rights management information.

20. A method for preventing unauthorized disclosure of secure information, the method comprising:

5 receiving, by a protect agent installed at a first egress point, digital information including a first text, the first text including a plurality of words; identifying, by the protect agent, a plurality of candidate entities, each of the plurality of candidate entities corresponding to a particular word of the plurality of words;

10 identifying, by the protect agent, one or more matching candidate entities from the plurality of candidate entities that match against one of a plurality of lightweight entities stored in a lightweight entity database (LWED);

transmitting, by the protect agent, the one or more matching candidate entities to a global entity database (GED), the GED including a plurality of registered entities identified to be secured against unauthorized disclosure;

15 receiving, from the GED, acknowledgement whether each of the one or more matching candidate entities matches against one of the plurality of registered entities included in the GED; and

performing, by the protect agent, a security action, when at least one of the one or more matching entities matches against one of the plurality of registered entities included in the GED.

21. The method of claim 20, wherein each of the plurality of registered entities included in the GED includes is generated by:

25 generating a hash-value for a first entity that an organization desires to protect from unauthorized disclosure; and

associating the hash-value with metadata related to the first entity.

22. The method of claim 21, wherein the metadata related to the first entity includes one or more of:

- 5           an entity type associated with the first entity;  
          a location of the first entity within a particular document; or  
          an origin information of the particular document.

23. The method of claim 21, wherein the LWED is a compressed version of the  
10 GED.

24. The method of claim 23, wherein the compressed version is generated by:  
          stripping metadata information associated with each of the hash-values  
          corresponding to the plurality of registered entities in the GED; and  
15           storing the stripped hash-values in a probabilistic data structure.

25. The method of claim 24, wherein the probabilistic data structure is a Bloom filter or a derivative thereof.

20 26. The method of claim 20, wherein the identification of a particular candidate entity from the plurality of words further comprises:

          utilizing an entity format matcher to identify a first word from the plurality of words that matches a particular word-pattern.

25 27. The method of claim 20, wherein the security action includes one or more of:

preventing the first text from being disclosed through the first egress point;  
logging the event as a security violation;  
requiring a password from a user to allow the first text to be disclosed;  
blocking the user's access to the first text;  
5 sending out a security alert; or  
integration of the first text with rights management information.

28. A system for preventing unauthorized disclosure of secure information, the system comprising:

10 a receiving module configured to receive digital information including a first text, the first text including a plurality of words;

a candidate ID module configured to identify a first candidate entity, the first candidate entity corresponding to a particular word of the plurality of words;

15 a comparison module configured to compare the first candidate entity against a plurality of registered entities stored in an entity database; and

a security action module configured to perform a security action when the first candidate entity matches against a particular registered entity of the plurality of registered entities.

20 29. The system of claim 28, wherein the candidate IP module includes:

an entity format matcher to identify a first word from the plurality of words that matches a particular word-pattern.

30. The system of claim 28, wherein the candidate ID module includes:

25 a heuristics engine to skip over one or more words from the plurality of words based on a heuristic rule.

31. The system of claim 30, wherein the heuristic rule includes one or more of:  
skipping over a first word from the plurality of words when the first word  
matches a first stop word of a plurality of stop words;

5 skipping over a second word from the plurality of words when the second  
word has a word-length that is shorter than a first word-length of a shortest  
registered entity of the plurality of registered entities; or

skipping over a third word from the plurality of words when the third word  
has a word-length that is longer than a second word-length of the longest  
10 registered entity of the plurality of registered entities.

32. The system of claim 28, wherein the identification of the first candidate  
entity from the plurality of words by the candidate ID module further comprises  
identifying a first entity type associated with the candidate entity.

15

33. The system of claim 32, wherein the plurality of registered entities stored in  
the entity database are categorized according an entity type associated with each  
of the plurality of registered entities.

20 34. The method of claim 33, wherein the comparison of the first candidate  
entity against the plurality of registered entities by the comparison module further  
comprises:

identifying a subplurality of registered entities that are categorized based on  
the first entity type;

25 comparing the first candidate entity against the subplurality of registered  
entities that are categorized based on the first entity type.

35. The system of claim 28, wherein the first candidate entity is converted to a canonical format prior to being compared against the plurality of registered entities, wherein the canonical format causes the protect agent to be impervious to differences in digital format and character encoding of the first candidate entity.

36. The system of claim 28, wherein the plurality of entities stored in the entity database correspond to entity words that are desired to be secured from unauthorized distribution.

37. The system of claim 36, wherein the entity database is a lightweight entity database (LWED) located at the site of the first egress point.

38. The system of claim 37, wherein the LWED includes a compressed database to hold the plurality of registered entities.

39. The system of claim 38, wherein the compressed database is generated by:

generating hash-values for each registered entity of the plurality of registered entities; and  
storing the generated hash-values in a probabilistic data structure.

40. The system of claim 39, wherein the probabilistic data structure is a Bloom filter or a derivative thereof.

41. The system of claim 36, wherein the entity database is a global entity database (GED), the GED including the plurality of registered entities.

42. The system of claim 41, wherein the GED is stored in association with a remote server, and wherein the protect agent at the first egress point communicates with the GED utilizing a network.

43. The system of claim 42, wherein each of the plurality of registered entities in the GED is represented using a corresponding hash-value.

10

44. The system of claim 43, wherein each of the plurality of registered entities in the GED includes metadata information, the metadata information for a given registered entity including one or more of:

an entity type associated with the given registered entity;  
15 a location of the given registered entity within a particular document; or  
an origin information of the particular document.

45. The system of claim 28, wherein the security action performed by the security module includes one or more of:

20 preventing the first text from being disclosed through the first egress point;  
logging the event as a security violation;  
requiring a password from a user to allow the first text to be disclosed;  
blocking the user's access to the first text;  
sending out a security alert; or  
25 integration of the first text with rights management information.

46. A system for preventing unauthorized disclosure of secure information, the system comprising:

a processor;

a network interface through which to communicate with one or more remote

5 servers over a network;

a memory storing code which, when executed by the processor, causes the network storage server system to perform a plurality of operations, including:

receiving, by a protect agent installed at a first egress point, digital information including a first text, the first text including a plurality of words;

10 identifying, by the protect agent, a plurality of candidate entities, each of the plurality of candidate entities corresponding to a particular word of the plurality of words;

identifying, by the protect agent, one or more matching candidate entities from the plurality of candidate entities that match against one of a plurality of lightweight entities stored in a lightweight entity database (LWED);

15 transmitting, by the protect agent, the one or more matching candidate entities to a global entity database (GED), the GED including a plurality of registered entities identified to be secured against unauthorized disclosure;

20 receiving, from the GED, acknowledgement whether each of the one or more matching candidate entities matches against one of the plurality of registered entities included in the GED; and

25 performing, by the protect agent, a security action, when at least one of the one or more matching entities matches against one or the plurality of registered entities included in the GED.

47. The system of claim 46, wherein each of the plurality of registered entities included in the GED includes is generated by:

- 5           generating a hash-value for a first entity that an organization desires to protect from unauthorized disclosure; and
- associating the hash-value with metadata related to the first entity.

48. The system of claim 47, wherein the metadata related to the first entity  
10 includes one or more of:

- an entity type associated with the first entity;
- a location of the first entity within a particular document; or
- an origin information of the particular document.

15 49. The system of claim 47, wherein the LWED is a compressed version of the GED.

50. The system of claim 49, wherein the compressed version is generated by:

              stripping metadata information associated with each of the hash-values  
20 corresponding to the plurality of registered entities in the GED; and

              storing the stripped hash-values in a probabilistic data structure.

51. The system of claim 50, wherein the probabilistic data structure is a Bloom filter or a derivative thereof.

25

52. The system of claim 46, wherein the identification of a particular candidate entity from the plurality of words further comprises:

utilizing an entity format matcher to identify a first word from the plurality of words that matches a particular word-pattern.

5

53. The system of claim 46, wherein the security action includes one or more of:

preventing the first text from being disclosed through the first egress point;

logging the event as a security violation;

10 requiring a password from a user to allow the first text to be disclosed;

blocking the user's access to the first text;

sending out a security alert; or

integration of the first text with rights management information.

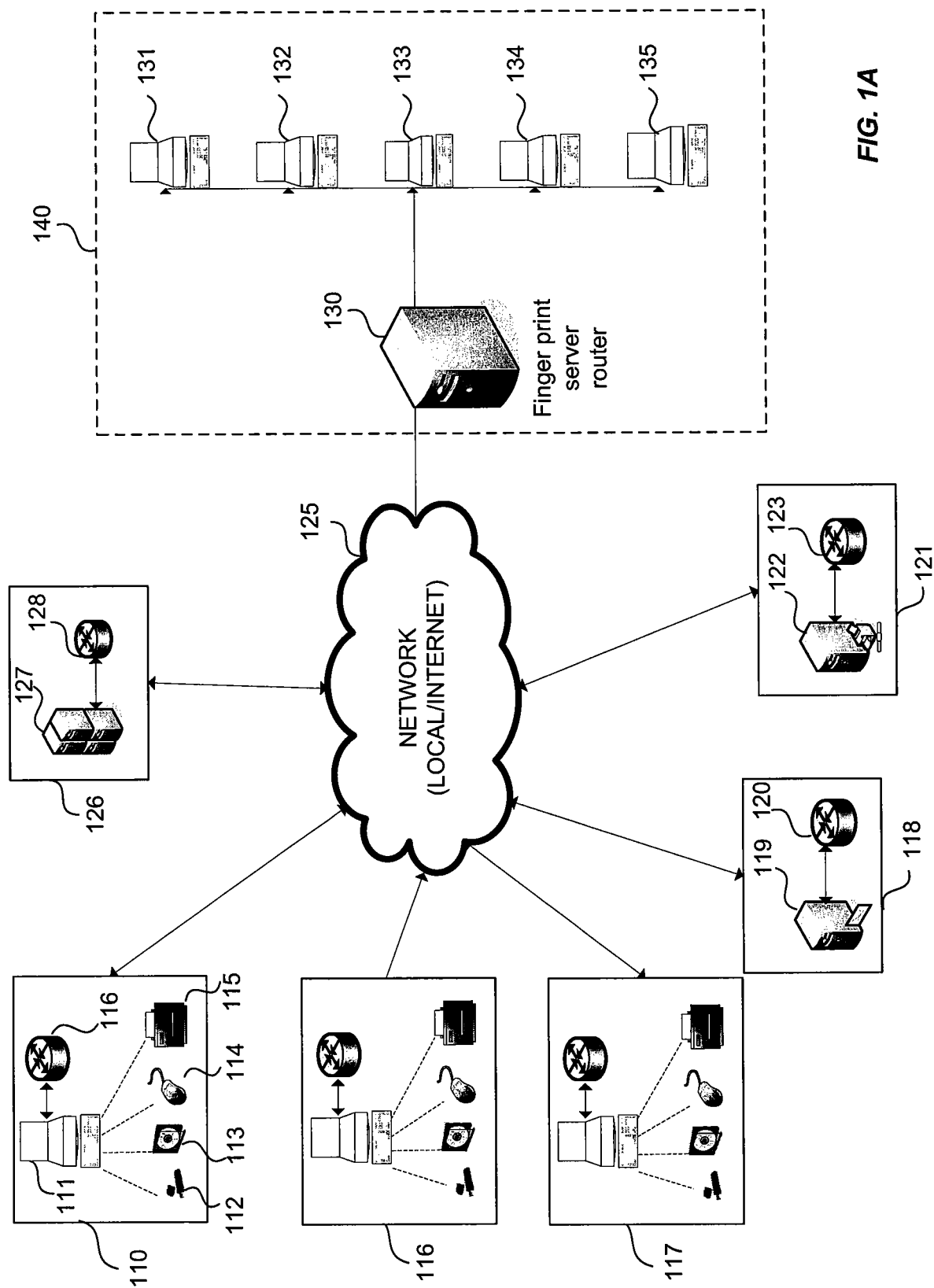


FIG. 1A

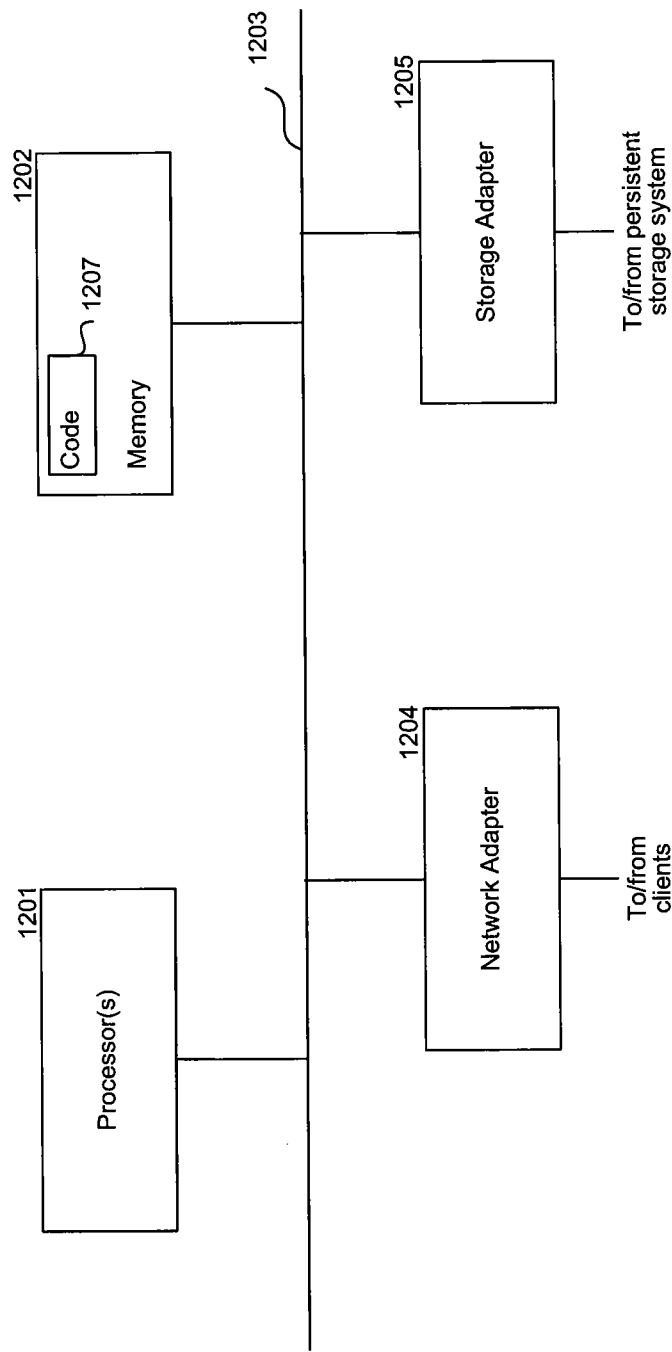
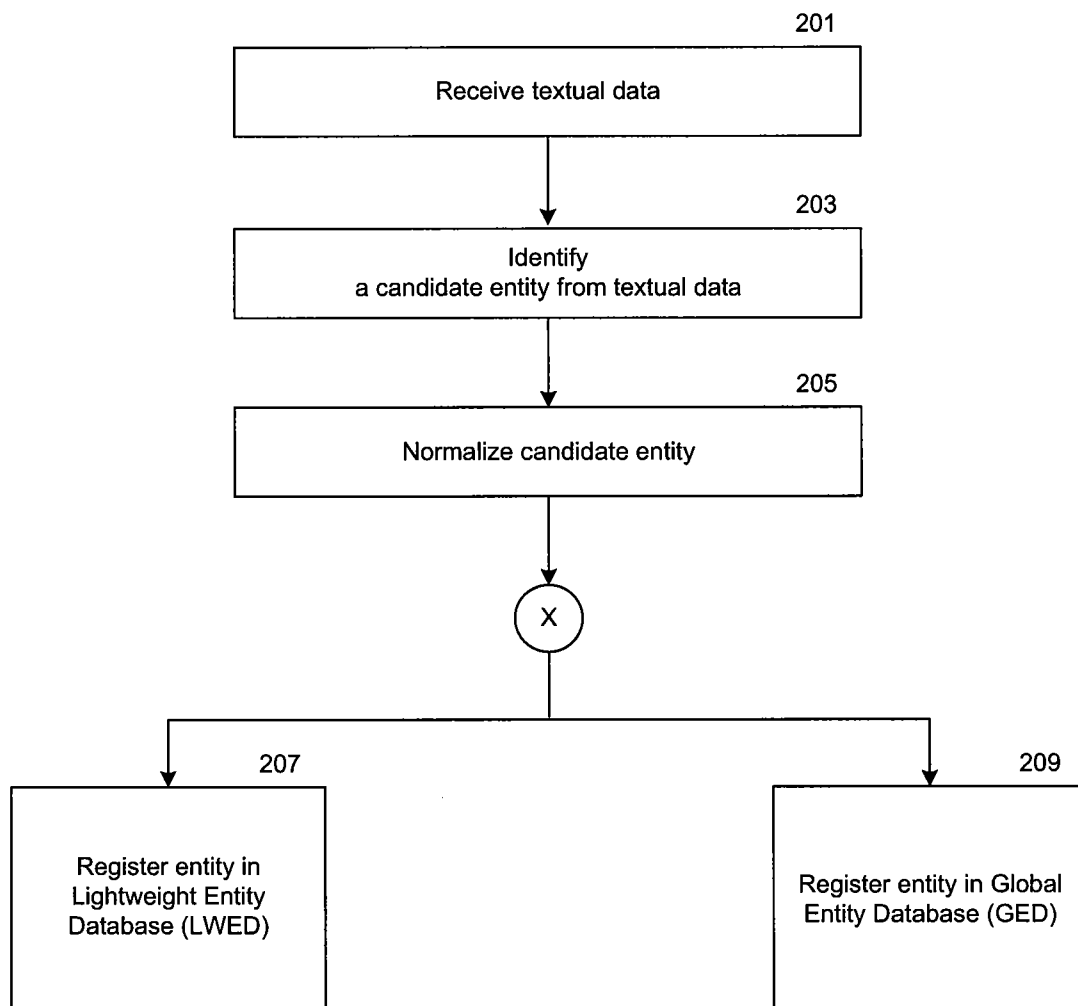
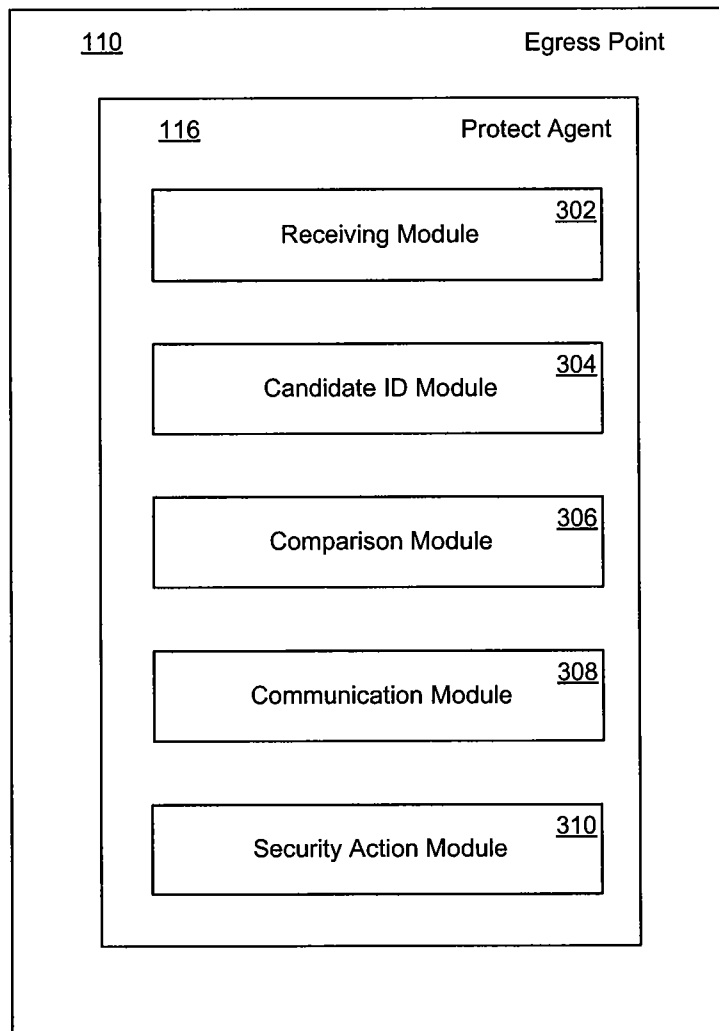
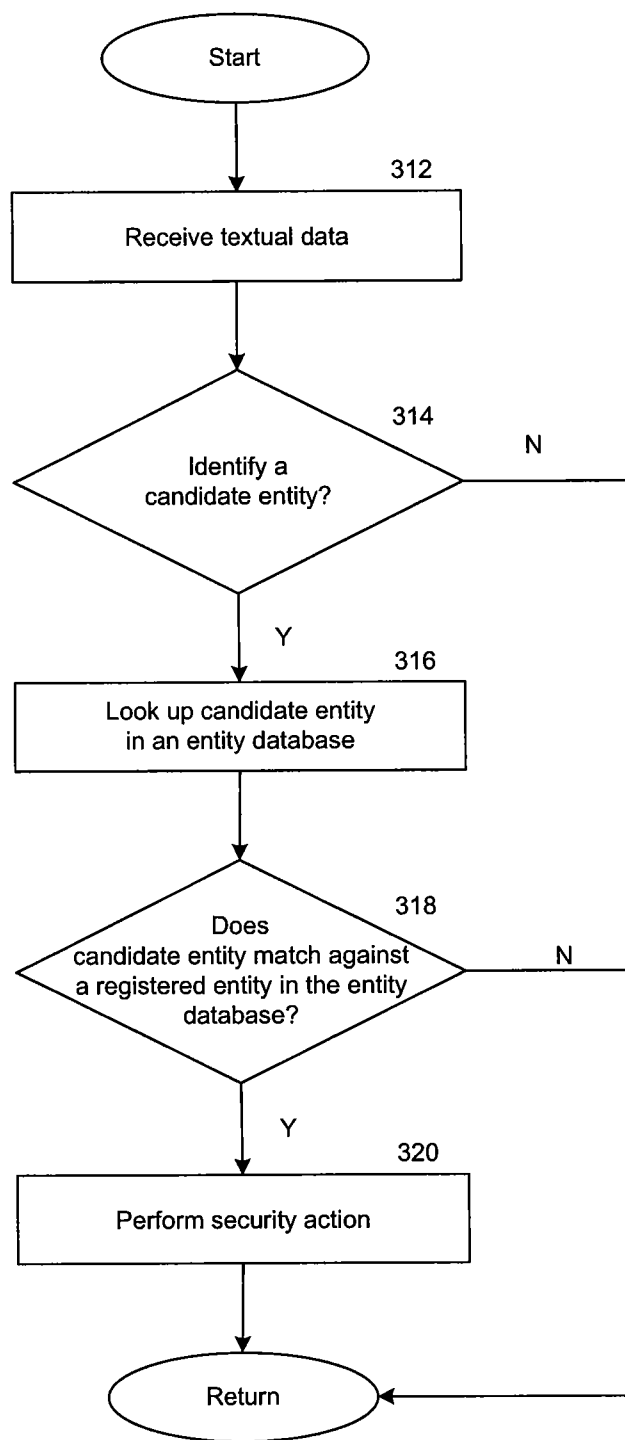
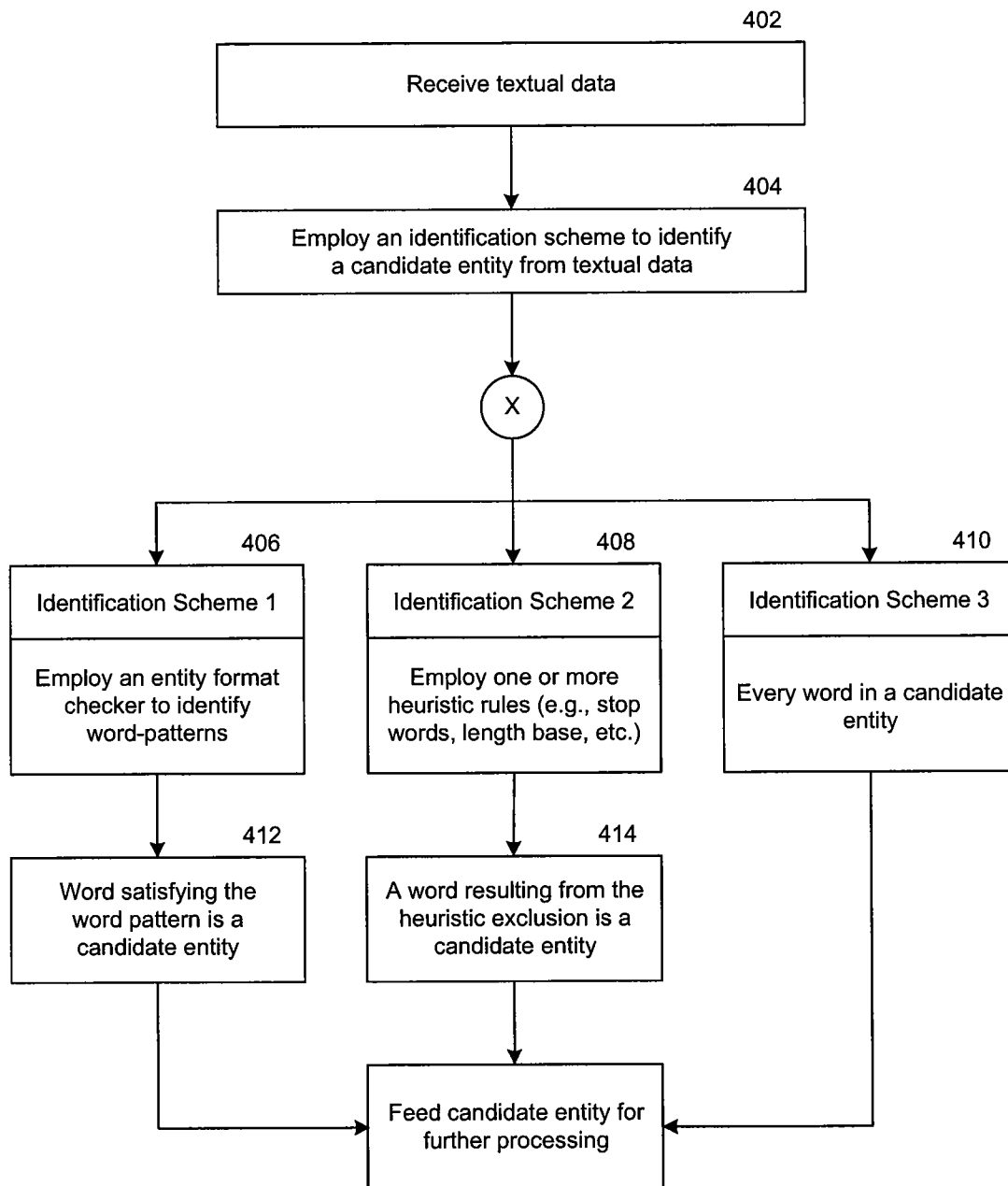


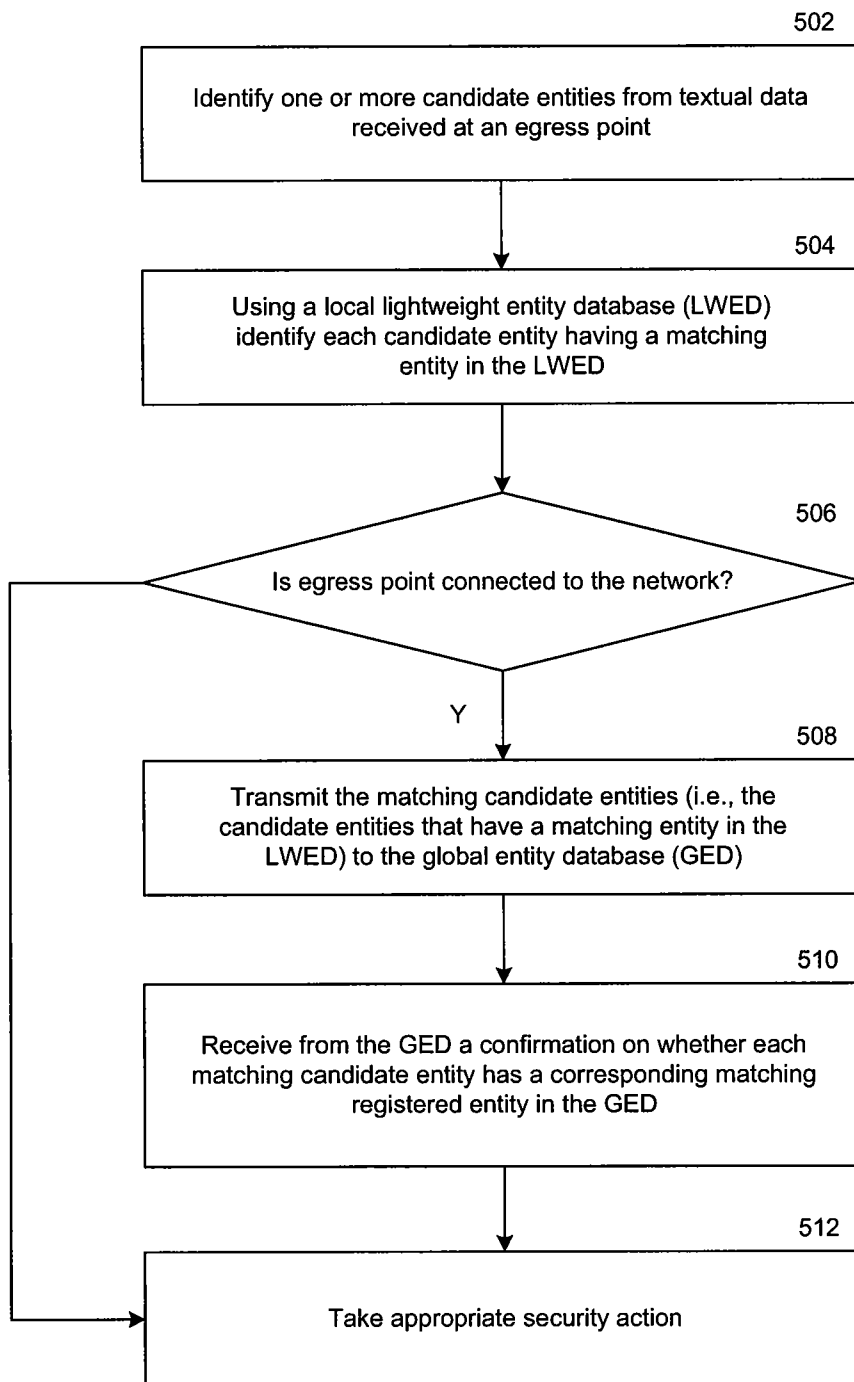
FIG. 1B

**FIG. 2**

***FIG. 3A***

**FIG. 3B**

**FIG. 4**

**FIG 5**