(12) **United States Patent**
Ghafourifar

(10) **Patent No.: US 9,990,513 B2**
(45) **Date of Patent: *Jun. 5, 2018**

(54) **SYSTEM AND METHOD OF APPLYING ADAPTIVE PRIVACY CONTROLS TO LOSSY FILE TYPES**

(71) Applicant: **Entefy Inc.**, Palo Alto, CA (US)

(72) Inventor: **Alston Ghafourifar**, Los Altos Hills, CA (US)

(73) Assignee: **Entefy Inc.**, Palo Alto, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days. days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/986,072**

(22) Filed: **Dec. 31, 2015**

(65) **Prior Publication Data**

US 2016/0188893 A1 Jun. 30, 2016

Related U.S. Application Data

(63) Continuation-in-part of application No. 14/584,329, filed on Dec. 29, 2014.

(51) **Int. Cl.**
G06F 21/62 (2013.01)
G09C 5/00 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 21/6227*** (2013.01); ***G09C 5/00*** (2013.01); *G06F 2221/2111* (2013.01); *G06F 2221/2137* (2013.01)

(58) **Field of Classification Search**
CPC G06F 21/602; H04L 9/0816; H04L 22/24; H04L 9/00; G09C 5/00
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,067,399	A	5/2000	Berger	
8,744,143	B2	6/2014	Chen	
8,799,022	B1	8/2014	O'Brien	
9,264,581	B2 *	2/2016	Lerios	G06T 3/00
9,350,914	B1	5/2016	Kaur	

(Continued)

OTHER PUBLICATIONS

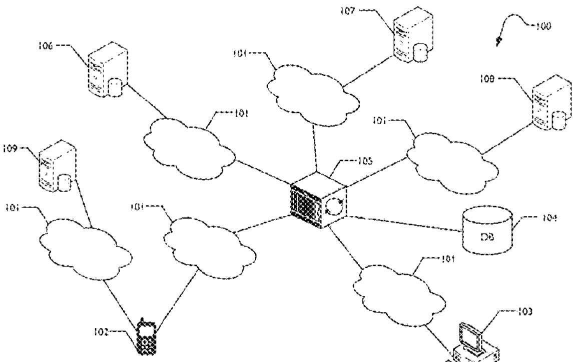
Goyal, et al., "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," '06, Oct. 30-Nov. 3, 2006, Alexandria, Virginia, USA.

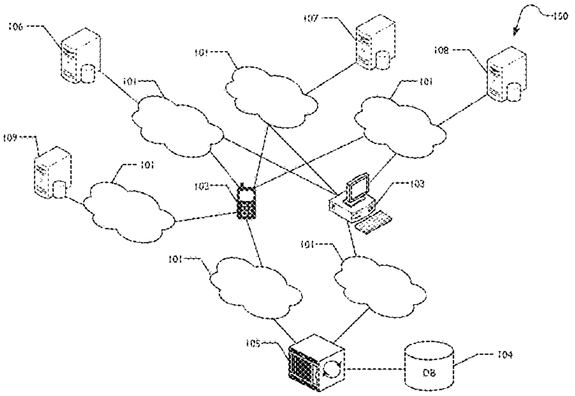
Primary Examiner — Wasika Nipa
(74) *Attorney, Agent, or Firm* — Blank Rome LLP

(57) **ABSTRACT**

The proliferation of personal computing devices in recent years, especially mobile personal computing devices, combined with a growth in the number of widely-used communications formats has led to increased concerns regarding the safety and security of documents and messages that are sent over networks. Users desire a system that provides for the setting of custom access permissions at a file-level or sub-file-level. Such a system may allow the user to apply customized privacy settings (and, optionally, encryption keys) differently to particular portions of a document—even if the document is of a 'lossy' file type, e.g., a JPEG image. According to some embodiments, the custom access permission settings may be implemented by obfuscating portions of the original file and then embedding "secret," e.g., hidden and/or encrypted, versions of the obfuscated portions in parts of the data structure of the original lossy file before transmitting the file to the desired recipients.

16 Claims, 10 Drawing Sheets





(56) References Cited

U.S. PATENT DOCUMENTS

9,426,387	B2	8/2016	Jung	
9,571,785	B2	2/2017	Farrell	
9,590,949	B2	3/2017	Murphy	
9,646,398	B2	5/2017	Yuan	
9,749,321	B2	8/2017	Farnsworth	
2002/0078361	A1 *	6/2002	Giroux	H04L 63/0428 713/183
2003/0105719	A1 *	6/2003	Berger	H04L 63/0428 705/51
2003/0108240	A1	6/2003	Gutta	
2006/0017747	A1 *	1/2006	Dawson	G06Q 20/085 345/629
2008/0140578	A1	6/2008	Felt	
2008/0193018	A1	8/2008	Masuda	
2008/0267403	A1	10/2008	Boult	
2009/0207269	A1	8/2009	Yoda	
2009/0244364	A1	10/2009	Nonogaki	
2010/0103193	A1	4/2010	Abe	
2010/0246890	A1	9/2010	Ofek	
2013/0011068	A1	1/2013	Albouyeh	
2013/0024901	A1	1/2013	Sharif-Ahmadi	
2013/0093829	A1	4/2013	Rosenblatt	
2013/0156263	A1	6/2013	Yamashita	
2014/0112534	A1	4/2014	Sako	
2015/0006390	A1 *	1/2015	Aissi	G06Q 20/40 705/44
2015/0016602	A1 *	1/2015	de los Reyes	G09C 5/00 380/28
2015/0033362	A1	1/2015	Mau	
2015/0113661	A1	4/2015	Mishra	
2015/0371049	A1	12/2015	Xavier	
2015/0371613	A1	12/2015	Patel	
2016/0217300	A1	7/2016	Kim	
2016/0241627	A1 *	8/2016	Ortega	H04L 65/80
2016/0292494	A1	10/2016	Ganong	
2016/0294781	A1 *	10/2016	Ninan	H04L 63/0407
2016/0316219	A1	10/2016	Yuan	
2017/0061155	A1	3/2017	Gordon	
2017/0220816	A1	8/2017	Matusek	

* cited by examiner

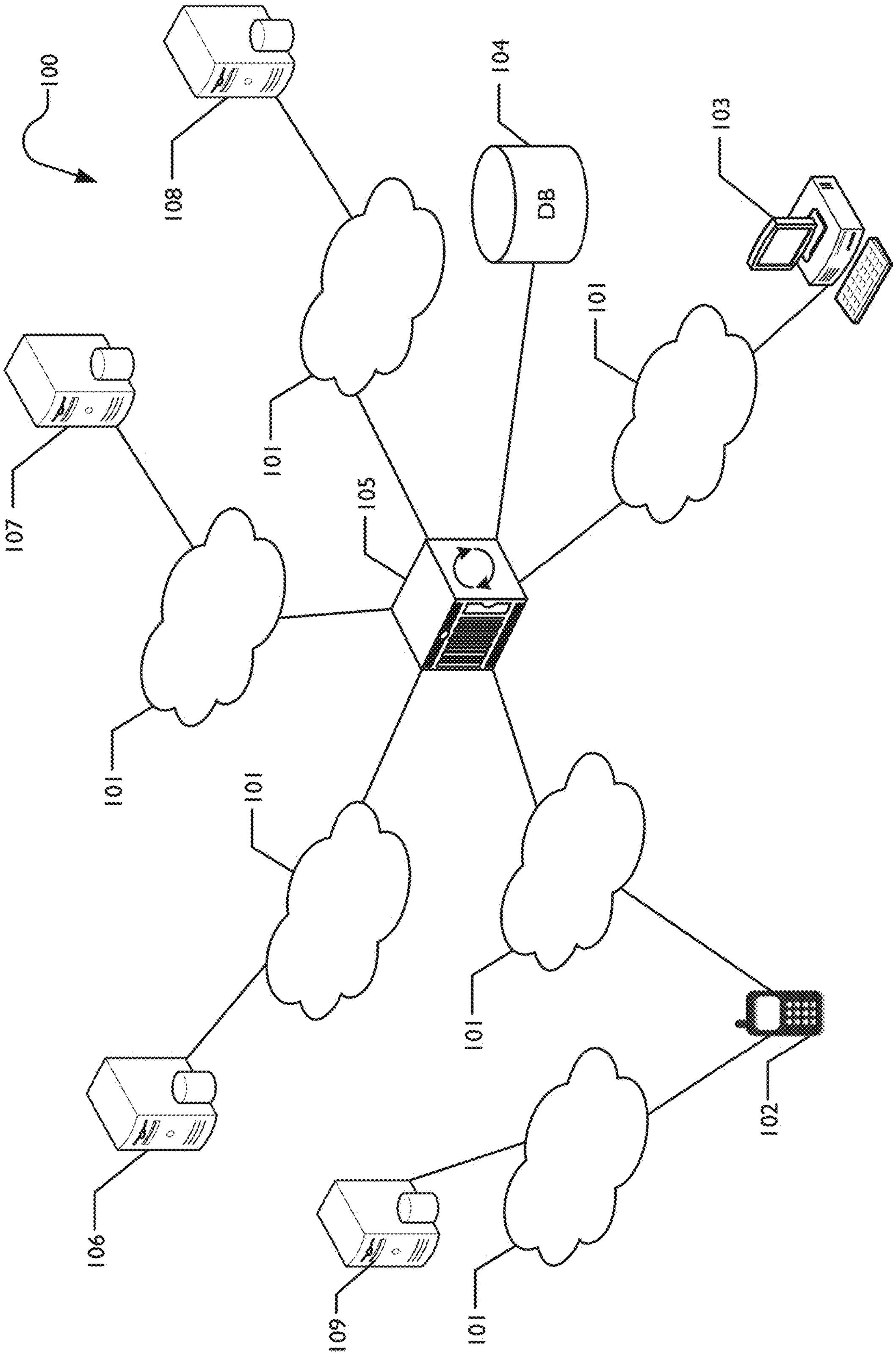


FIG. 1A

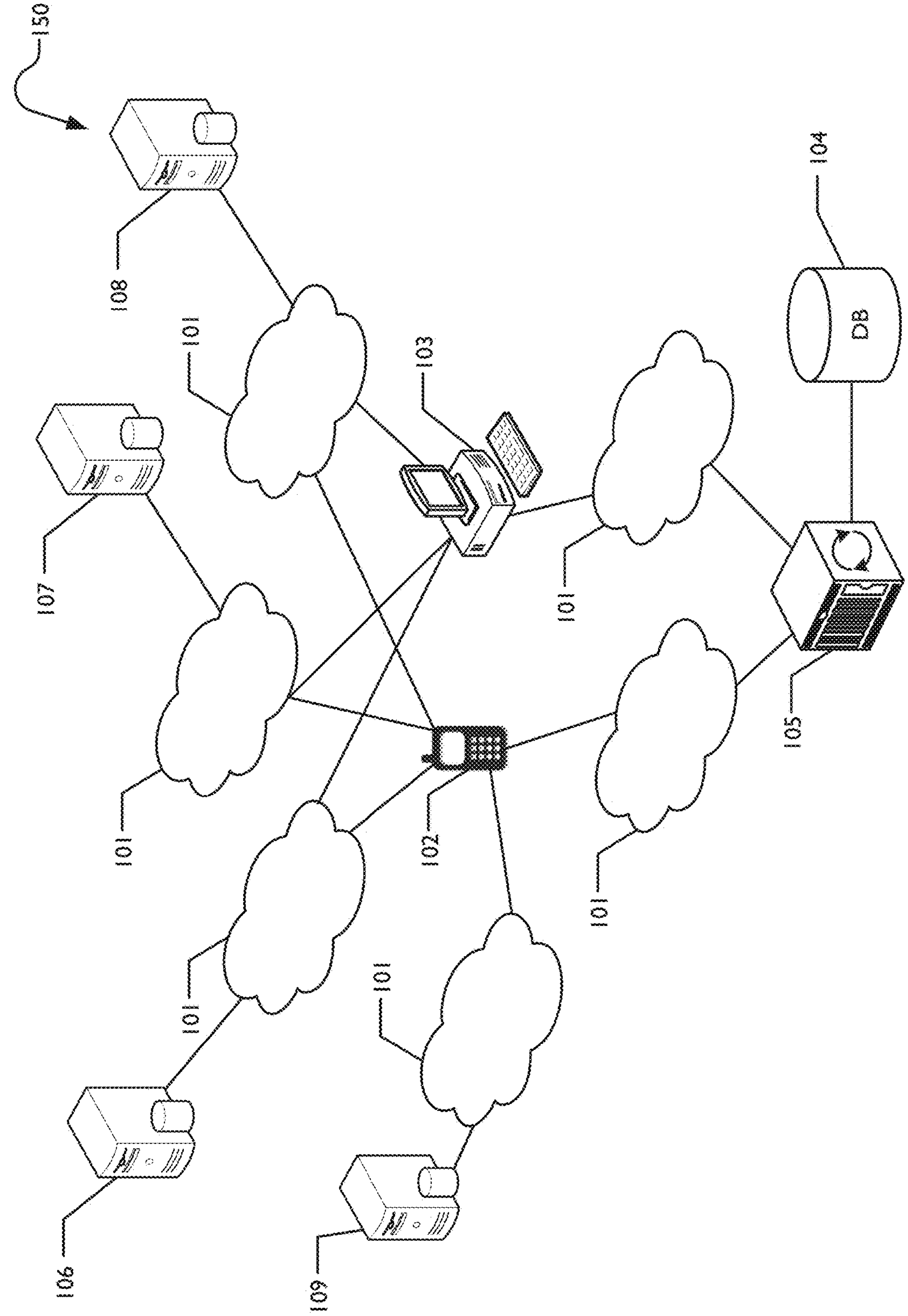


FIG. 1B

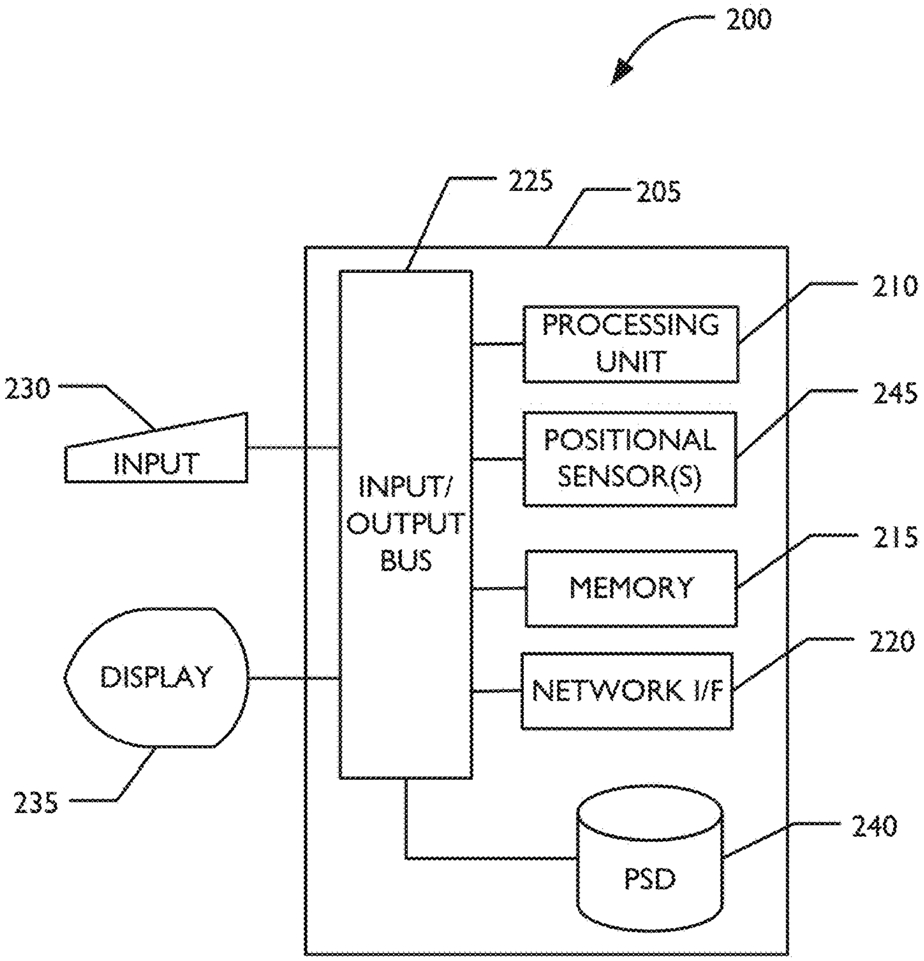

200
225
205
210
PROCESSING UNIT
245
POSITIONAL SENSOR(S)
230
INPUT
INPUT/ OUTPUT BUS
215
MEMORY
220
NETWORK I/F
DISPLAY
235
240
PSD

FIG. 2A

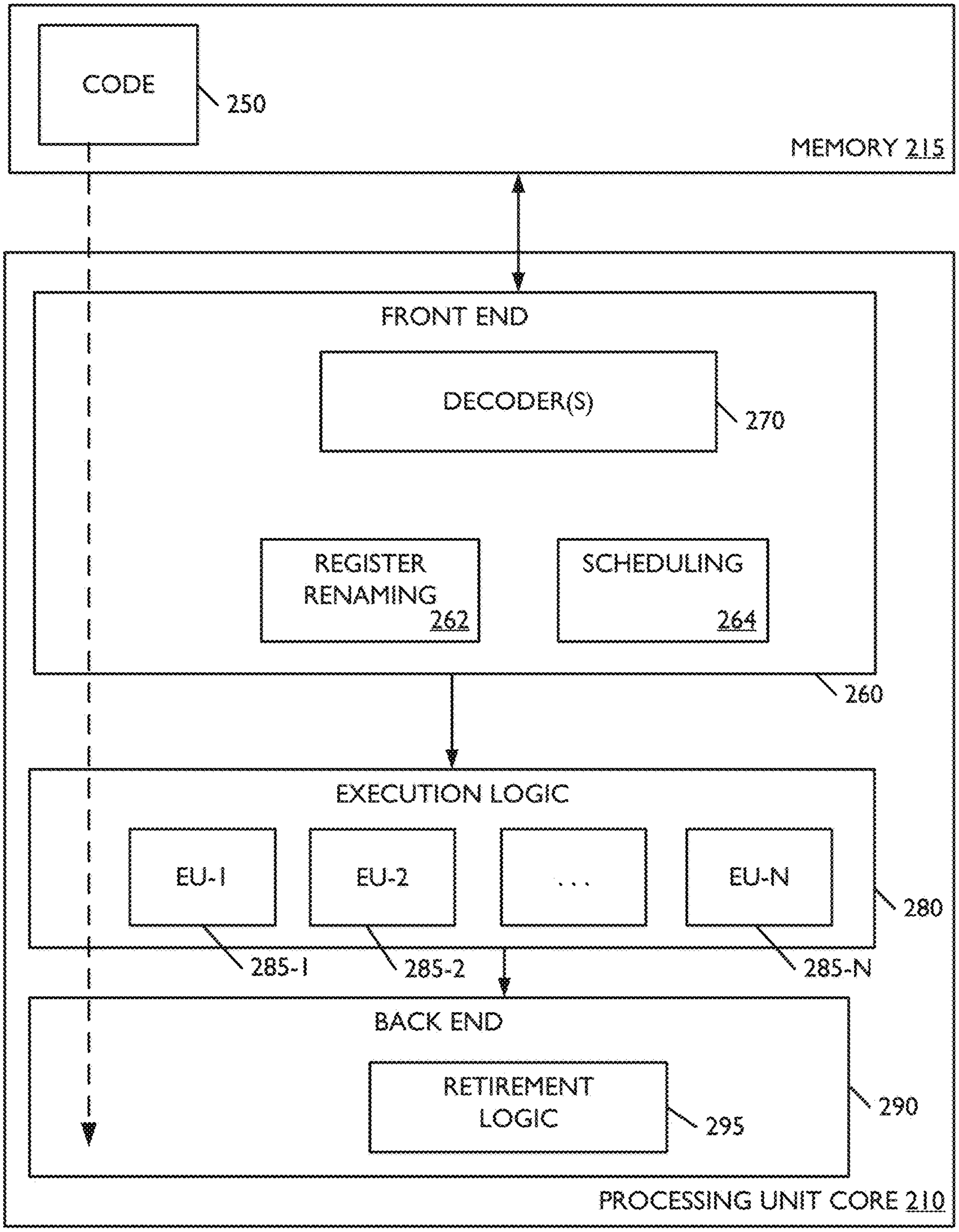

CODE
250
MEMORY 215
FRONT END
DECODER(S)
270
REGISTER RENAMING 262
SCHEDULING 264
260
EXECUTION LOGIC
EU-1
EU-2
. . .
EU-N
280
285-1
285-2
285-N
BACK END
RETIREMENT LOGIC
295
290
PROCESSING UNIT CORE 210

FIG. 2B

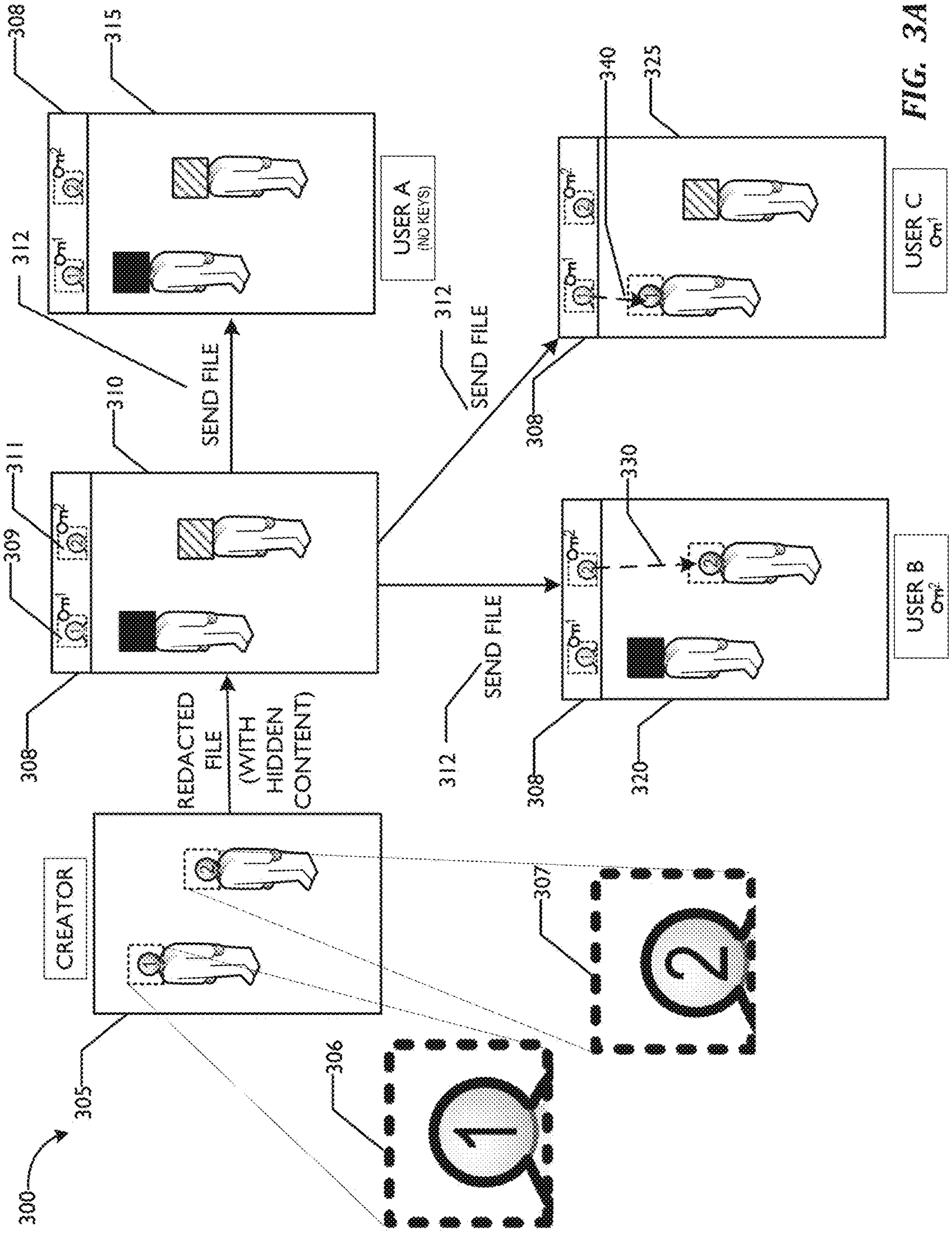

300
305
CREATOR
306
1
307
2
REDACTED FILE (WITH HIDDEN CONTENT)
308
309
311
310
SEND FILE
312
315
USER A (NO KEYS)
320
330
USER B
325
340
USER C

FIG. 3A

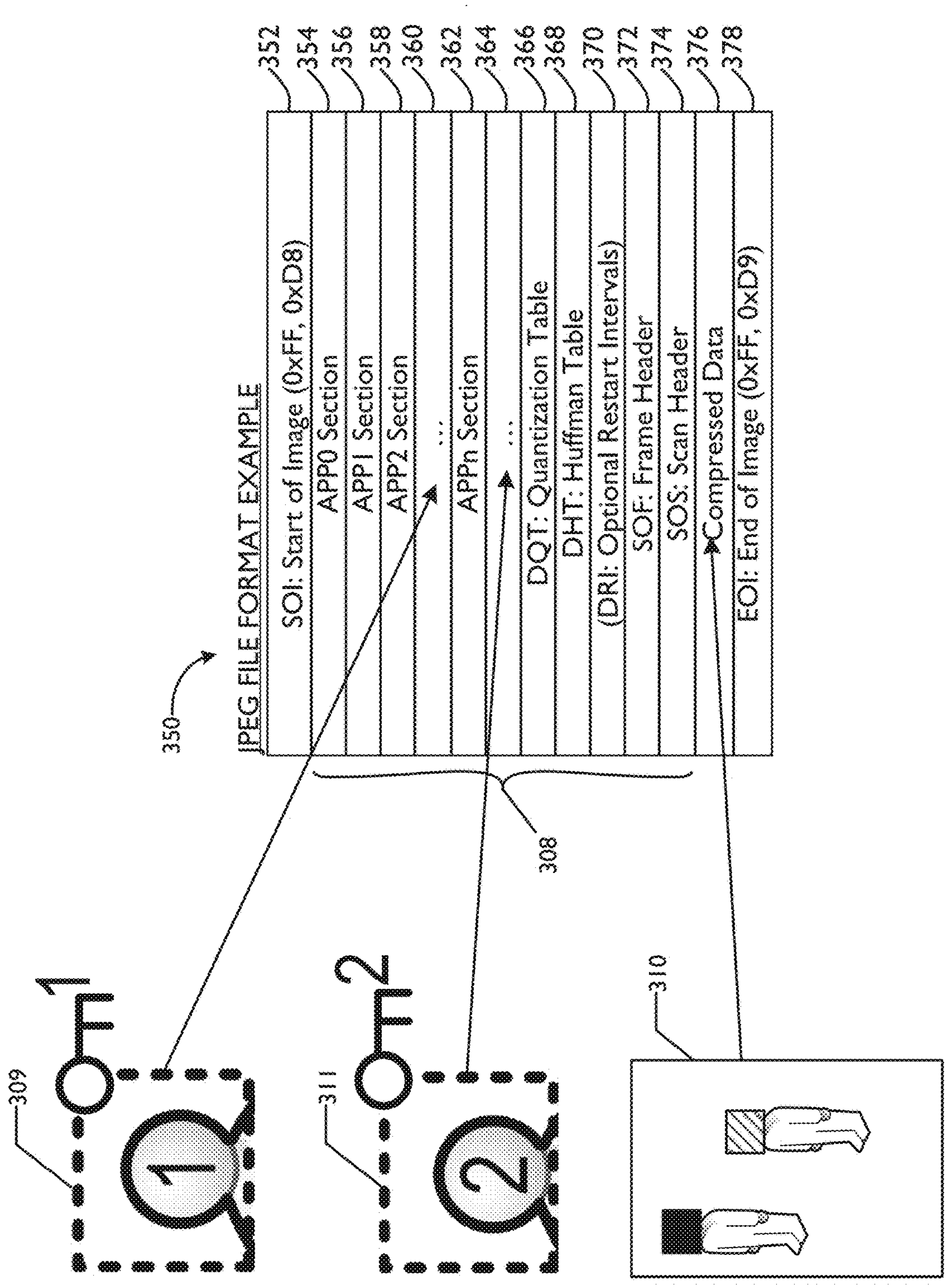

350
JPEG FILE FORMAT EXAMPLE
SOI: Start of Image (0xFF, 0xD8)
APP0 Section
APP1 Section
APP2 Section
...
APPn Section
...
DQT: Quantization Table
DHT: Huffman Table
(DRI: Optional Restart Intervals)
SOF: Frame Header
SOS: Scan Header
Compressed Data
EOI: End of Image (0xFF, 0xD9)
352
354
356
358
360
362
364
366
368
370
372
374
376
378
308
309
1
311
2
310

FIG. 3B

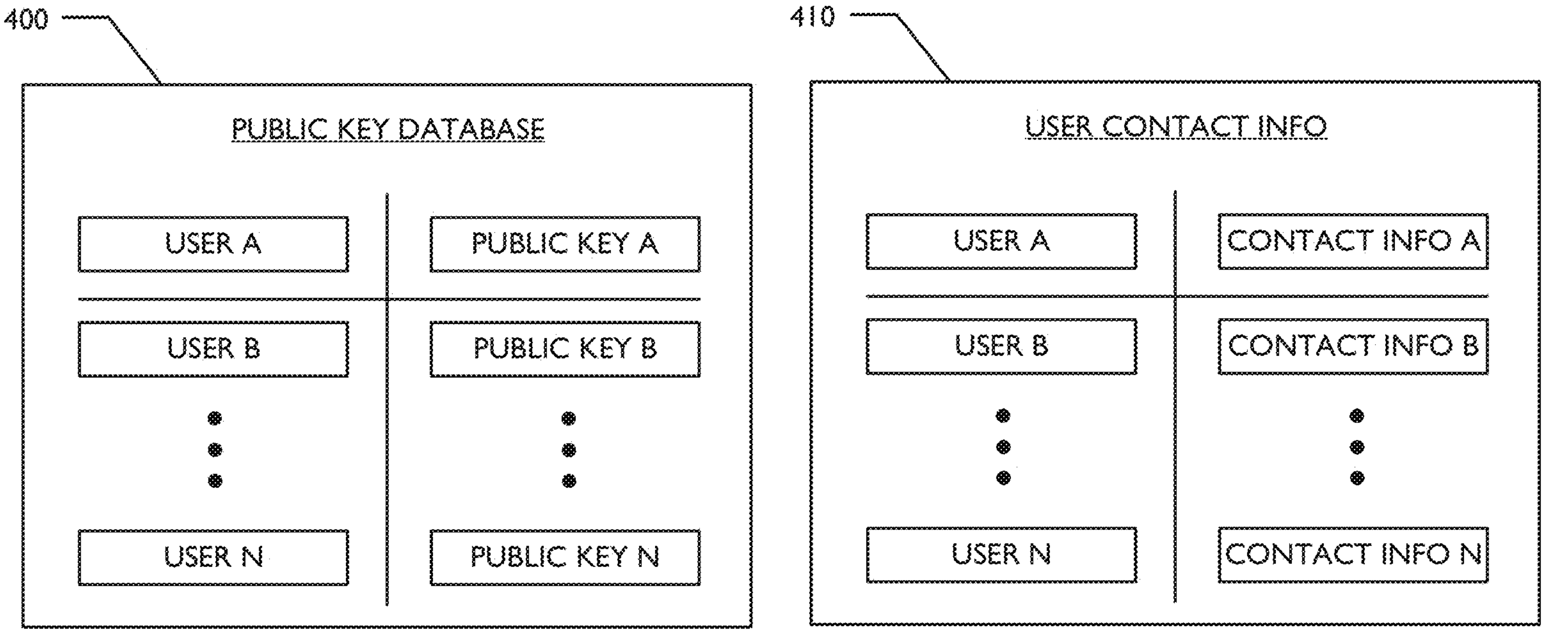

400
PUBLIC KEY DATABASE
USER A
PUBLIC KEY A
USER B
PUBLIC KEY B
USER N
PUBLIC KEY N
410
USER CONTACT INFO
USER A
CONTACT INFO A
USER B
CONTACT INFO B
USER N
CONTACT INFO N

FIG. 4

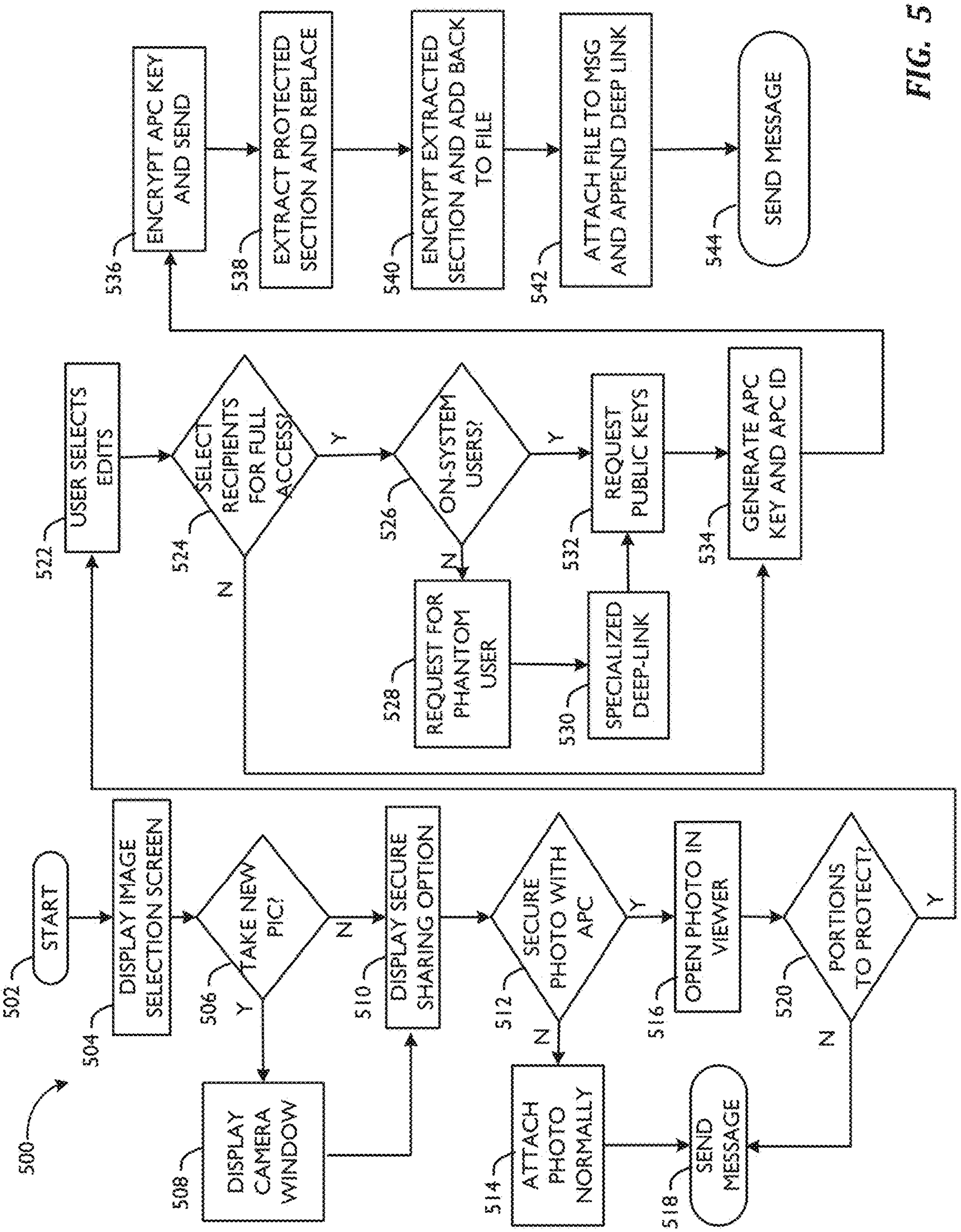

500
502
START
504
DISPLAY IMAGE SELECTION SCREEN
506
TAKE NEW PIC?
Y
N
508
DISPLAY CAMERA WINDOW
510
DISPLAY SECURE SHARING OPTION
512
SECURE PHOTO WITH APC
Y
N
514
ATTACH PHOTO NORMALLY
518
SEND MESSAGE
516
OPEN PHOTO IN VIEWER
520
PORTIONS TO PROTECT?
Y
N
522
USER SELECTS EDITS
524
SELECT RECIPIENTS FOR FULL ACCESS?
Y
N
526
ON-SYSTEM USERS?
Y
N
528
REQUEST FOR PHANTOM USER
530
SPECIALIZED DEEP-LINK
532
REQUEST PUBLIC KEYS
534
GENERATE APC KEY AND APC ID
536
ENCRYPT APC KEY AND SEND
538
EXTRACT PROTECTED SECTION AND REPLACE
540
ENCRYPT EXTRACTED SECTION AND ADD BACK TO FILE
542
ATTACH FILE TO MSG AND APPEND DEEP LINK
544
SEND MESSAGE

FIG. 5

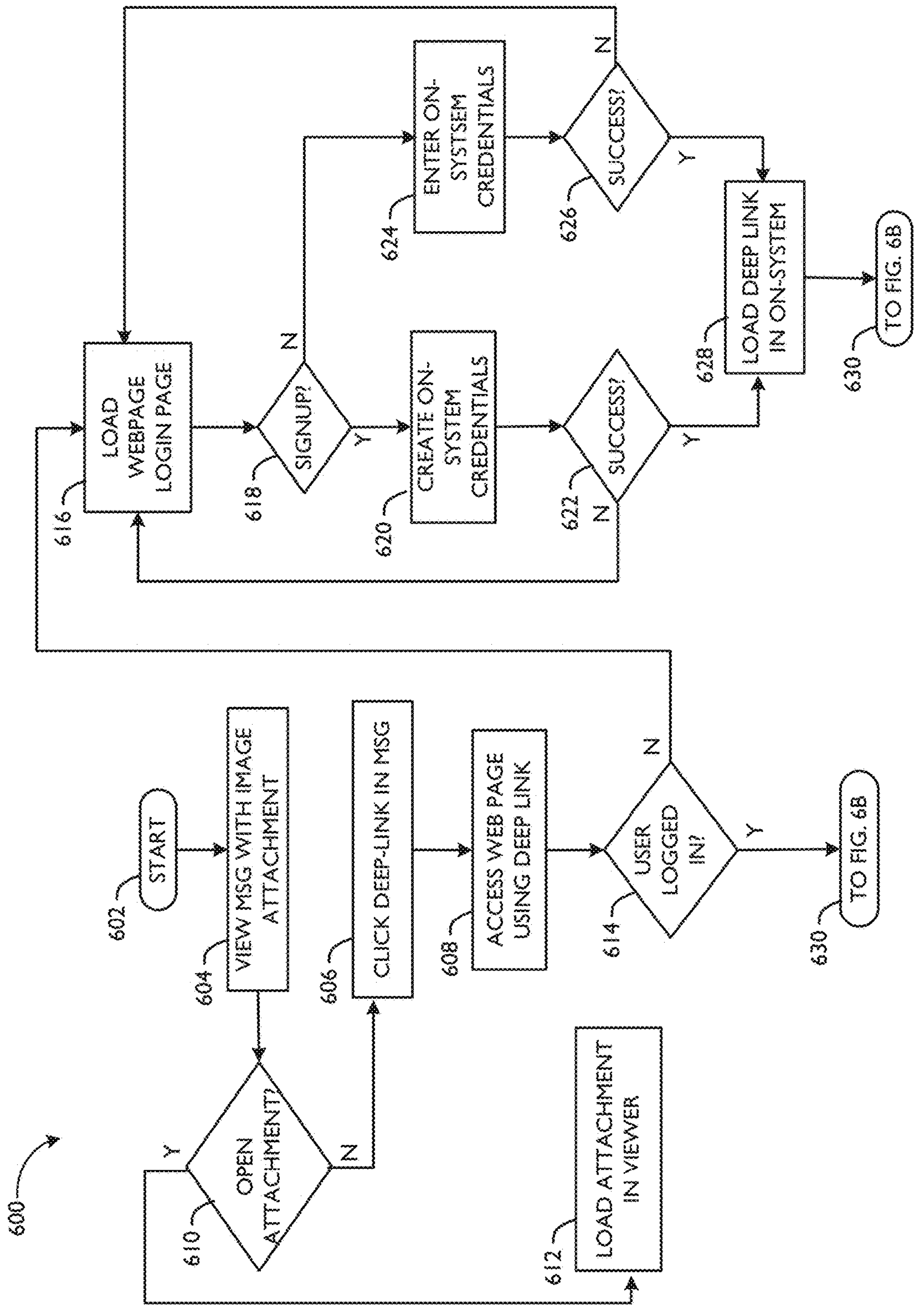

600
602
START
604
VIEW MSG WITH IMAGE ATTACHMENT
610
OPEN ATTACHMENT?
Y
N
612
LOAD ATTACHMENT IN VIEWER
606
CLICK DEEP-LINK IN MSG
608
ACCESS WEB PAGE USING DEEP LINK
614
USER LOGGED IN?
Y
N
630
TO FIG. 6B
616
LOAD WEBPAGE LOGIN PAGE
618
SIGNUP?
Y
N
620
CREATE ON-SYSTEM CREDENTIALS
622
SUCCESS?
Y
N
624
ENTER ON-SYSTSEM CREDENTIALS
626
SUCCESS?
Y
N
628
LOAD DEEP LINK IN ON-SYSTEM
630
TO FIG. 6B

FIG. 6A

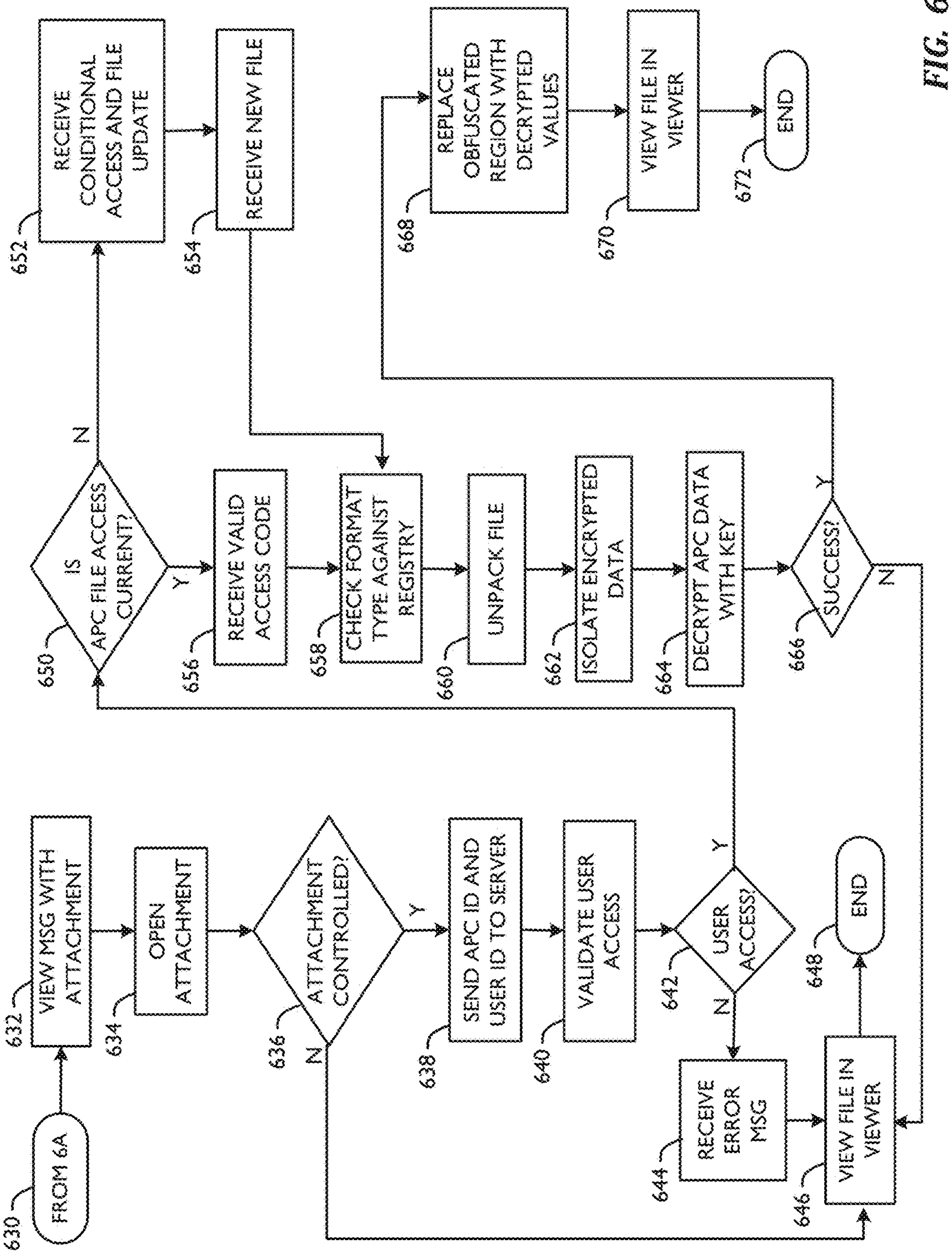

630
FROM 6A
632
VIEW MSG WITH ATTACHMENT
634
OPEN ATTACHMENT
636
ATTACHMENT CONTROLLED?
Y
N
638
SEND APC ID AND USER ID TO SERVER
640
VALIDATE USER ACCESS
642
USER ACCESS?
Y
N
644
RECEIVE ERROR MSG
646
VIEW FILE IN VIEWER
648
END
650
IS APC FILE ACCESS CURRENT?
Y
N
652
RECEIVE CONDITIONAL ACCESS AND FILE UPDATE
654
RECEIVE NEW FILE
656
RECEIVE VALID ACCESS CODE
658
CHECK FORMAT TYPE AGAINST REGISTRY
660
UNPACK FILE
662
ISOLATE ENCRYPTED DATA
664
DECRYPT APC DATA WITH KEY
666
SUCCESS?
Y
N
668
REPLACE OBFUSCATED REGION WITH DECRYPTED VALUES
670
VIEW FILE IN VIEWER
672
END

FIG. 6B

SYSTEM AND METHOD OF APPLYING ADAPTIVE PRIVACY CONTROLS TO LOSSY FILE TYPES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to, and is a continuation-in-part of, U.S. patent application Ser. No. 14/584,329, filed Dec. 29, 2014, entitled "System And Method of Determining User-Defined Permissions Through A Network" ("the '329 application"). The '329 application is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

This disclosure relates generally to systems, methods, and computer readable media for applying user-defined access permission settings to files in lossy file formats, those files may then be disseminated over a network. More particularly, the access permission settings may be implemented by embedding "secret," e.g., hidden and/or encrypted, information in such lossy file formats.

BACKGROUND

The proliferation of personal computing devices in recent years, especially mobile personal computing devices, combined with a growth in the number of widely-used communications formats (e.g., text, voice, video, image) and protocols (e.g., SMTP, IMAP/POP, SMS/MMS, XMPP, etc.) has led to increased concerns regarding the safety and security of documents and messages that are sent over networks. Users desire a system that provides for the setting of custom, e.g., user-defined access permissions for a lossy file or part of the lossy file that comprises less than the entire file through a communications network. A 'lossy' file, as used herein, refers to a file (or file format) that is compressed using inexact approximation methods (e.g., partial data discarding methods). As such, lossy compression techniques may be used to reduce data size for storage, handling, and transmitting content. However, because lossy compression reduces a file by permanently discarding certain information (e.g., redundant information), when the file is decompressed, it is not decompressed to 100% of the original. Lossy compression is generally used for multimedia files, e.g., images files, such as JPEG files or PNG files, video files, and/or sound files—where a certain amount of information loss will not be detected by most users and can result in significant gains in file size reduction or performance.

Lossy files may contain header properties. These header properties may be used to store alternate contents such as metadata, random information, or even full encoding of other files or portions of files, such as in the embodiments described herein. Such a system would allow customized privacy settings to be specified for different recipients, e.g., recipients at various levels of social distance from the user sending the document or message (e.g., public, private, followers, groups, Level-1 contacts, Level-2 contacts, Level-3 contacts, etc.). Such a system may also allow the user to apply customized privacy settings and encryption keys differently to particular parts of a lossy file, e.g., making one or more parts of the lossy file available only to a first class of users, or by making other parts of the lossy file available to the first class of users and a second class of users, all while preventing access to parts of lossy file by users who do not have the requisite access privileges.

Thus, a system for providing access permission setting through Adaptive Privacy Controls (APC) is described herein. APC, as used herein, will refer to a user-controllable or system-generated, intelligent privacy system that can limit viewing, editing, and re-sharing privileges for lossy files, for example, image files and other multimedia files that include a lossy compression, wherein changes made to the content of such 'lossy' files may not be reliably reversed or dynamically changed—as would be necessary according to prior art techniques attempting to implement the kinds of fine-grained access permission setting methods disclosed herein. Other embodiments of APC systems will, of course, be able to handle the setting of access permissions for recipients of lossless file formats, as well. In summary, APC systems, as used herein, allow users to share whatever information they want with whomever they want, while keeping others from accessing the same information, e.g., via hiding and/or encryption processes that can be initiated by user command or via system intelligence, even on lossy file types. APC access permission settings may be applied to individuals, pre-defined groups, and/or ad-hoc groups. Customized encryption keys may further be applied to particular parties or groups of parties to enhance the security of the permission settings.

APC may be used to apply privacy settings to only particular portions of a lossy file, for example, a particular portion of a JPEG image. For example, User A may be a family member who may be authorized to see an entire JPEG image, but User B and other users may be mere acquaintances, who are only authorized to see a redacted portion or portions of the JPEG image. For example, the entire JPEG image file would be viewable to User A, but only a redacted portion or portions of the JPEG image (e.g., everything but the face of the subject(s) in the image) would be available to the User B and other users when viewing the JPEG image file in an authorized viewing application.

According to some embodiments disclosed herein, a standard, i.e., "unauthorized," JPEG viewing application would also be able to open the redacted version of the JPEG image file; it simply would not "know" where to look within the JPEG image file for the "true" content from the redacted portion or portions of the JPEG image file. According to still other embodiments, even if an unauthorized JPEG viewing application were able to find the "true" content from the redacted portion or portions "hidden" within the file structure of the JPEG image, the redacted portion or portions may be encrypted, and the unauthorized JPEG viewing application would not possess the necessary decryption keys to decrypt the encrypted redacted portion or portions. Moreover, the unauthorized JPEG viewing application also would not know where to "place" the encrypted portion or portions back within the image to reconstruct the original, i.e., unredacted, JPEG image in a seamless fashion.

Thus, according to some embodiments, the network-based, user-defined, APC controls for lossy file types or files (e.g., JPEG images) may include access permission systems, methods, and computer readable media that provide a seamless, intuitive user interface (e.g., using touch gestures or mouse input) allowing a user to: "block out" particular portions or areas of interest in a lossy file; hide (and optionally encrypt) such "blocked out" portions within parts the lossy file's data structure; and then send the lossy file to particular recipients or groups of recipients with customized access permission settings, which settings may be specified on a per-recipient basis and that either allow or do not allow a given recipient to locate the hidden portions and (if necessary) decrypt such hidden portions so that the original lossy file may be reconstructed by the recipient.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A is a block diagram illustrating a server-entry point network architecture infrastructure, according to one or more disclosed embodiments.

FIG. 1B is a block diagram illustrating a client-entry point network architecture infrastructure, according to one or more disclosed embodiments.

FIG. 2A is a block diagram illustrating a computer that could be used to execute the cloud-based user defined APC approaches described herein according to one or more of disclosed embodiments.

FIG. 2B is a block diagram illustrating a processor core, which may reside on a computer according to one or more of disclosed embodiments.

FIG. 3A shows an example of sub-document-level access permission setting scheme with custom recipient-based privacy settings, according to one or more disclosed embodiments.

FIG. 3B shows an example of a lossy file type used to store hidden (and/or encrypted) content.

FIG. 4 shows an example of customized privacy setting using encryption keys, according to one or more disclosed embodiments.

FIG. 5 is a flowchart showing a method for utilizing an APC process for lossy files from a sender's perspective, according to one or more disclosed embodiments.

FIGS. 6A-6B show flowcharts that depict a method for utilizing an APC process for lossy files from a receiver's perspective, according to one or more disclosed embodiments.

DETAILED DESCRIPTION

Disclosed are systems, methods and computer readable media for creating user-defined custom access permission settings for files stored in lossy file types, e.g., JPEG images, which settings may serve to limit the viewing and/or sharing privileges for the files (or portions of the files) on a per-recipient basis. More particularly, but not by way of limitation, this disclosure relates to systems, methods, and computer readable media to permit users of the access permission setting system to redact certain content which corresponds to particular portions of the lossy file and then "hide" (and optionally encrypt) the redacted content within one or more parts of the data structure of the lossy file type. The recipient receiving the lossy file may then, if an authorized recipient and using an authorized viewing application, locate the hidden content within the file, decrypt the hidden content (if necessary), and then seamlessly reconstruct the content of the lossy file in its original form. This process may thus allow for the reconstruction of the original content in a seamless and secure fashion that enforces the sender's original recipient-specific privacy intentions for the various portions of the lossy file, while still allowing other unauthorized-recipients to view the redacted version of the file in standard viewing applications for the particular lossy file type.

Referring now to FIG. 1A, a server-entry point network architecture infrastructure 100 is shown schematically. Infrastructure 100 contains computer networks 101. Computer networks 101 include many different types of computer networks available today, such as the Internet, a corporate network, or a Local Area Network (LAN). Each of these networks can contain wired or wireless devices and operate using any number of network protocols (e.g., TCP/IP). Networks 101 may be connected to various gateways and routers, connecting various machines to one another, represented, e.g., by sync server 105, end user computers 103, mobile phones 102, and computer servers 106-109. In some embodiments, end user computers 103 may not be capable of receiving SMS text messages, whereas mobile phones 102 are capable of receiving SMS text messages. Also shown in infrastructure 100 is a cellular network 101 for use with mobile communication devices. As is known in the art, mobile cellular networks support mobile phones and many other types of devices (e.g., tablet computers not shown). Mobile devices in the infrastructure 100 are illustrated as mobile phone 102. Sync server 105, in connection with database(s) 104, may serve as the central "brains" and data repository, respectively, for the adaptive privacy control system to be described herein. In the server-entry point network architecture infrastructure 100 of FIG. 1A, centralized sync server 105 may be responsible for querying and obtaining all the messages from the various communication sources for individual users of the system, communicating public keys, applying adaptive privacy controls to lossy files or objects, which may be communicated to one or more users of the system synchronized with the data on the various third party communication servers that the system is in communication with. Database(s) 104 may be used to store local copies of messages sent and received by users of the system, as well as individual documents associated with a particular user, which may or may not also be associated with particular communications of the users. As such, the database portion allotted to a particular user will contain a record of all communications in any form to and from the user.

Server 106 in the server-entry point network architecture infrastructure 100 of FIG. 1A represents a third party email server (e.g., a GOOGLE® or YAHOO!® email server). (GOOGLE is a registered service mark of Google Inc. YAHOO! is a registered service mark of Yahoo! Inc.) Third party email server 106 may be periodically pinged by sync server 105 to determine whether particular users of a multi-protocol, multi-format communication composition and inbox feed system described herein have received any new email messages via the particular third-party email services. Server 107 represents a third party instant message server (e.g., a YAHOO!® Messenger or AOL® Instant Messaging server). (AOL is a registered service mark of AOL Inc.) Third party instant messaging server 107 may also be periodically pinged by sync server 105 to determine whether particular users of the multi-protocol, multi-format communication composition and inbox feed system described herein have received any new instant messages via the particular third-party instant messaging services. Similarly, server 108 represents a third party social network server (e.g., a FACEBOOK® or TWITTER® server). (FACEBOOK is a registered trademark of Facebook, Inc. TWITTER is a registered service mark of Twitter, Inc.) Third party social network server 108 may also be periodically pinged by sync server 105 to determine whether particular users of the multi-protocol, multi-format communication composition and inbox feed system described herein have received any new social network messages via the particular third-party social network services. It is to be understood that, in a "push-based" system, third party servers may push notifications to sync server 105 directly, thus eliminating the need for sync server 105 to periodically ping the third party servers. Finally, server 109 represents a cellular service provider's server. Such servers may be used to manage the sending and receiving of messages (e.g., email or SMS text messages) to users of mobile devices on the provider's cellular network. Cellular service provider servers may also be used: 1) to provide geo-fencing for location and movement determination; 2) for data transference; and/or 3) for live telephony (i.e., actually answering and making phone calls with a user's client device). In situations where two 'on-network' or 'on-system' users are communicating with one another via the multi-protocol communication system itself, such communications may occur entirely via sync server **105**, and third party servers **106-109** may not need to be contacted. An 'on-network' user may include a user that has set up a user profile on sync server **105** specifying preferred communications formats and/or protocols for a given communication session/message (e.g., if the recipient is in an area with a poor service signal, lower bit-rate communication formats, such as text, may be favored over higher bit-rate communications formats, such as video or voice), and/or economic considerations of format/protocol choice to the recipient and/or sender (e.g., if SMS messages would charge the recipient an additional fee from his or her provider, other protocols, such as email, may be chosen instead).

Referring now to FIG. **1**B, a client-entry point network architecture infrastructure **150** is shown schematically. Similar to infrastructure **100** shown in FIG. **1**A, infrastructure **150** contains computer networks **101**. Computer networks **101** may again include many different types of computer networks available today, such as the Internet, a corporate network, or a Local Area Network (LAN). However, unlike the server-centric infrastructure **100** shown in FIG. **1**A, infrastructure **150** is a client-centric architecture. Thus, individual client devices, such as end user computers **103** and mobile phones **102** may be used to query the various third party computer servers **106-109** to retrieve the various third party email, IM, social network, and other messages for the user of the client device. Such a system has the benefit that there may be less delay in receiving messages than in a system where a central server is responsible for authorizing and pulling communications for many users simultaneously. Also, a client-entry point system may place less storage and processing responsibilities on the central multi-protocol, multi-format communication composition and inbox feed system's server computers since the various tasks may be distributed over a large number of client devices. Further, a client-entry point system may lend itself well to a true, "zero knowledge" privacy enforcement scheme. In infrastructure **150**, the client devices may also be connected via the network to the central sync server **105** and database **104**. For example, central sync server **105** and database **104** may be used by the client devices to reduce the amount of storage space needed on-board the client devices to store communications-related content and/or to keep all of a user's devices synchronized with the latest communication-related information and content related to the user. It is to be understood that, in a "push-based" system, third party servers may push notifications to end user computers **102** and mobile phones **103** directly, thus eliminating the need for these devices to periodically ping the third party servers.

Referring now to FIG. **2**A, an example processing device **200** for use in the communication systems described herein according to one embodiment is illustrated in block diagram form. Processing device **200** may serve in, e.g., a mobile phone **102**, end user computer **103**, sync server **105**, or a server computer **106-109**. Example processing device **200** comprises a system unit **205** which may be optionally connected to an input device **230** (e.g., keyboard, mouse, touch screen, etc.) and display **235**. A program storage device (PSD) **240** (sometimes referred to as a hard disk, flash memory, or non-transitory computer readable medium) is included with the system unit **205**. Also included with system unit **205** may be a network interface **220** for communication via a network (either cellular or computer) with other mobile and/or embedded devices (not shown). Network interface **220** may be included within system unit **205** or be external to system unit **205**. In either case, system unit **205** will be communicatively coupled to network interface **220**. Program storage device **240** represents any form of non-volatile storage including, but not limited to, all forms of optical and magnetic memory, including solid-state storage elements, including removable media, and may be included within system unit **205** or be external to system unit **205**. Program storage device **240** may be used for storage of software to control system unit **205**, data for use by the processing device **200**, or both.

System unit **205** may be programmed to perform methods in accordance with this disclosure. System unit **205** comprises one or more processing units, input-output (I/O) bus **225** and memory **215**. Access to memory **215** can be accomplished using the communication bus **225**. Processing unit **210** may include any programmable controller device including, for example, a mainframe processor, a mobile phone processor, or, as examples, one or more members of the INTEL® ATOM™, INTEL® XEON™, and INTEL® CORE™ processor families from Intel Corporation and the Cortex and ARM processor families from ARM. (INTEL, INTEL ATOM, XEON, and CORE are trademarks of the Intel Corporation. CORTEX is a registered trademark of the ARM Limited Corporation. ARM is a registered trademark of the ARM Limited Company). Memory **215** may include one or more memory modules and comprise random access memory (RAM), read only memory (ROM), programmable read only memory (PROM), programmable read-write memory, and solid-state memory. As also shown in FIG. **2**A, system unit **205** may also include one or more positional sensors **245**, which may comprise an accelerometer, gyrometer, global positioning system (GPS) device, or the like and, which, may be used to track the movement of user client devices.

Referring now to FIG. **2**B, a processing unit core **210** is illustrated in further detail, according to one embodiment. Processing unit core **210** may be the core for any type of processor, such as a micro-processor, an embedded processor, a digital signal processor (DSP), a network processor, or other device to execute code. Although only one processing unit core **210** is illustrated in FIG. **2**B, a processing element may alternatively include more than one of the processing unit core **210** illustrated in FIG. **2**B. Processing unit core **210** may be a single-threaded core or, for at least one embodiment, the processing unit core **210** may be multithreaded, in that, it may include more than one hardware thread context (or "logical processor") per core.

FIG. **2**B also illustrates a memory **215** coupled to the processing unit core **210**. The memory **215** may be any of a wide variety of memories (including various layers of memory hierarchy), as are known or otherwise available to those of skill in the art. The memory **215** may include one or more code instruction(s) **250** to be executed by the processing unit core **210**. The processing unit core **210** follows a program sequence of instructions indicated by the code **250**. Each instruction enters a front end portion **260** and is processed by one or more decoders **270**. The decoder may generate as its output a micro operation such as a fixed width micro operation in a predefined format, or may generate other instructions, microinstructions, or control signals, which reflect the original code instruction. The front end **260** may also include register renaming logic **262** and scheduling logic **264**, which generally allocate resources and queue the operation corresponding to the convert instruction for execution.

The processing unit core **210** is shown including execution logic **280** having a set of execution units **285-1** through **285**-N. Some embodiments may include a number of execution units dedicated to specific functions or sets of functions. Other embodiments may include only one execution unit or one execution unit that can perform a particular function. The execution logic **280** performs the operations specified by code instructions.

After completion of execution of the operations specified by the code instructions, back end logic **290** retires the instructions of the code **250**. In one embodiment, the processing unit core **210** allows out of order execution but requires in order retirement of instructions. Retirement logic **295** may take a variety of forms as known to those of skill in the art (e.g., re-order buffers or the like). In this manner, the processing unit core **210** is transformed during execution of the code **250**, at least in terms of the output generated by the decoder, the hardware registers and tables utilized by the register renaming logic **262**, and any registers (not shown) modified by the execution logic **280**.

Although not illustrated in FIG. **2**B, a processing element may include other elements on chip with the processing unit core **210**. For example, a processing element may include memory control logic along with the processing unit core **210**. The processing element may include I/O control logic and/or may include I/O control logic integrated with memory control logic. The processing element may also include one or more caches.

File- and Sub-File-Level Access Permission Setting Scheme with Custom, Recipient-Based Privacy Settings

According to some embodiments of a system for providing Adaptive Privacy Controls (APC), file-level access permission setting may be implemented. For example, in one scenario, a user may wish to share a file of a lossy file type (e.g., a JPEG image that incorporates lossy compression) with a first colleague, but not allow that information to be visible to other colleagues who may receive the lossy file from the first colleague. The first colleague may be an 'on-system' recipient or an 'off-system' recipient. In such a scenario, User A may use the access permission setting system to send an obfuscated lossy file (e.g., by attaching the file to a MIME format email and sending using SMTP) to the first colleague, User B, while selecting the appropriate encryption attributes in the original lossy file to limit the visibility of User B (and other users who may view the container file) to only specific portions of the file's content. In one embodiment, User A may create an edited copy of the original file, referred to herein as an "obfuscated" lossy file. Obfuscation may include any of a number of techniques to "mask" the true contents of the file, e.g.: blurs, color-out, scratch-out, or the like at particular coordinate locations in the JPEG image that the sender wishes to obfuscate or protect portions of the image file. The client application or server (depending on the system architecture) may then "hide" (and optionally encrypt) the "true" copy of the obfuscated content within a part of the data structure of the lossy file. If encryption is desired, any compatible encryption process may be used, e.g., a public/private key process, with the specific public key being provided by the device, the recipient user, or another central authority to create an encrypted lossy file. User B can then receive a typical message with the lossy file attached, which includes the hidden (and optionally encrypted) true copy of the obfuscated portions of the file. In some embodiments, a part of the data structure of the lossy file may also include a deep-link for validating the receiving user's credentials, as well information for creating a so-called "phantom user identifier," e.g., a temporary authorized identifier that may be used by an 'off-system' user to authenticate himself or herself for the purposes of viewing a particular piece(s) of protected content. The deep-link may be used to validate user credentials, as well as to view the hidden (and/or encrypted) obfuscated contents of the file in a compatible authorized viewer application.

User B may be an 'on-network' or 'on-system' recipient or an 'off-system recipient'. If User B is an 'on-network' recipient, and the hidden "true" contents of the obfuscated portions of the file are also encrypted, the system may use one (or more) of a number of encryption schemes to ensure that only authorized recipients are able to view the true contents of the file. For example, the bits of the hidden "true" content of the obfuscated portions may be encrypted with different keys for different people/groups. Alternately, the bits may be encrypted once with a single key. The single key may then be encrypted many times with different per-user or per-device keys and stored within the same JPEG container thereby saving space. Alternately, the keys may be stored on the server and recalled dynamically, or they may be sent via public/private key exchange. Finally, the bits may be encrypted using Key-Policy Attribute Based Encryption (KP-ABE), regular public/private key, AES keys, or the like.

As mentioned above, another exemplary situation wherein sub-document-level access permission setting may be employed in the sharing of files of lossy file types, e.g., pictures, video, or other media content compressed using lossy compression, is the situation whereby specific portions of the media content require selective censorship, redaction, or other protection for certain recipients, so as to maintain desired privacy or security levels on a per-recipient level. In one scenario, User A, the sharer, may want to share a humorous JPEG picture with his wife (User B) and young son (User C). Knowing that the picture contains certain explicit words or imagery—but is still funny even without the explicit sub-portions of the content. User A may attach the photo to a message in an authorized 'on-system' application and use the application's selection capabilities to "block-out," or redact, the explicit sub-portions of the image. Prior to sending the lossy JPEG file, User A could instruct the system to allow User B to view the full uncensored image after receipt and decryption in an authorized viewing application, while only allowing User C to view the censored portions of the image. Embodiments of a system for providing Adaptive Privacy Controls as discussed above with respect to User A and User B are discussed in further detail below with respect to FIGS. **3**A, **3**B, **5**, **6**A, and **6**B.

For the exemplary file-level privacy control scenarios described above, the application can present a view of the lossy file in question (e.g., via a compatible authorized file viewer or image thumbnail, etc.) to the sender of the lossy file. The sender can then use any desired form of selection input (e.g., touch gestures, mouse gestures, etc.) to indicate which content should be protected and/or access-controlled, e.g., via hiding and/or encryption. Those selections will be recorded and either processed locally or sent to a central server (depending on client capabilities), whereby the system will process the object's original source code (e.g. in XML format, JPEG format, etc.), corresponding to the section or sections matching the user selection, in order to enforce the user's selection of protected and/or access-controlled portions.

The section(s) in question may then be isolated (maintaining suitable markup) at the code level and, if desired, encrypted (e.g., using any one of standard encryption practices, such as asymmetric public/private key, or more advanced Key-Policy Attribute Based Encryption, i.e., "KP-ABE," etc.). The marked-up sections are identified at the code level, with pixel coordinates being replaced at the code level to generate representative pixels that are black, blurred, scratched, or similarly obfuscated when viewed by an unauthorized viewer in a compatible application for the particular file type of the lossy file. Certain embodiments may replace selected bit array regions with other content to be read by an authorized viewer application to perform custom operations and prompt the application to contact a server to retrieve access codes for the obfuscated file (if encrypted) in an attempt to decrypt with the private key stored in the authorized application. Unsuccessful retrieval or decryption will result in the recipient only viewing the obfuscated lossy file that depicts "part" of the original file. Because this service requires knowledge of the markup structure of any compatible file type, all APC changes may be made with data at the code level to create a "flat" file, while keeping the protected sections encrypted in complementary file portion(s), such that the application may insert the protected sections if user privileges are verified to view the protected sections. In some embodiments, the complementary file portion(s) may be inserted as hidden data within the data structure of the original file, resulting in a single, larger file, rather than being transmitted along with the original file as a distinct file(s).

FIG. **3**A shows an example of an access permission setting scheme **300** with custom recipient-based privacy settings for JPEG files, according to one embodiment. As demonstrated in the exemplary access permission scheme **300**, a creator or sender ("Creator") may create or edit an image, such as a JPEG file **305**, with custom permission settings applied to it. For example, Creator may edit a copy of a JPEG file to obfuscate information in one or more portions of the JPEG file **305**. Specifically, the Creator may identify particular redacted portions in the JPEG file **305** to block out from view of certain recipients of the JPEG file **305**. In this example, JPEG file **305** comprises two human subjects, whose heads are labeled '1' and '2' for illustrative purposes. The first portion selected for obfuscation by the Creator is portion **306**, which covers the head of person '1,' as is shown in zoomed-in form in FIG. **3**A for illustrative purposes. The second portion selected for obfuscation by the Creator is portion **307**, which covers the head of person '2,' as is shown in zoomed-in form in FIG. **3**A for illustrative purposes.

The obfuscated portions **306** and **307** of JPEG file **305** are represented by the black and diagonally-striped squares, respectively, over the corresponding portions of the redacted copy of the JPEG file shown in element **310** in FIG. **3**A. For example, pixels at particular coordinates in the redacted copy of the JPEG file **310** may be overwritten at the code level to obfuscate the sub-portions of the image **305** corresponding to portions **306** and **307**. The protected portions **306** and **307** from the original JPEG file **305** may be hidden (and optionally encrypted), e.g., within one or more applicable header parts **308** of the redacted copy of the JPEG file **310**, in order to protect the information prior to transmission to recipient users. As shown in FIG. **3**A, the hidden (and optionally encrypted) data corresponding to portion **306** from the original JPEG image file **305** (i.e., the head of person '1') is labeled **309**, and is shown overlaid with a key icon and a small number '1.' The key icon **309** with the number '1' next to it indicates that only recipients authorized to see the head of person '1' (e.g., either through pre-set access permissions and/or by virtue of holding the appropriate decryption key)—when viewing the JPEG file **310** in authorized viewing application-will actually be able to see the "true" content of portion **306**, i.e., the head of person '1.' Any recipient that is not an authorized recipient of portion **306**—or who is viewing image **310** outside of an authorized viewing application—will simply see the black box over the head of person '1.' Likewise, The key icon **311** with the number '2' next to it indicates that only recipients authorized to see the head of person '2' (e.g., either through pre-set access permissions and/or by virtue of holding the appropriate decryption key)—when viewing the JPEG file **310** in authorized viewing application-will actually be able to see the "true" content of portion **307**, i.e., the head of person '2.' Any recipient that is not an authorized recipient of portion **307**—or who is viewing image **310** outside of an authorized viewing application-will simply see the diagonally-striped box over the head of person '2.'

The obfuscated JPEG file **310** that has been augmented with the hidden (and optionally encrypted) redacted content **309** and **311** may then be selectively shared with one of three separate users in this example: User A, User B, and User C. In this example, each of User A, User B, and User C may receive the same redacted JPEG image file **310**. In this embodiment, the access permissions for the redacted JPEG image file **310** may be defined differently for each of User A, User B, and User C, such that each User may be able to view one or more different portions of the redacted information as it was presented in the original JPEG file.

Finally, the Creator may choose to send the JPEG file **312** to three separate users (either simultaneously or at different times), with the same portions obfuscated out for each recipient (i.e., portions **306** and **307** in this case). However, each user may be able to locate (and optionally decrypt) the hidden portions **309** and **311** (corresponding, in this case, to the heads of persons '1' and '2,' respectively) from the header **308** of the JPEG file **310**, in order to view non-redacted portions of the JPEG image file, in accordance with, e.g., their identity, access permissions, status as a member of a particular group, or their status as a follower of the Creator, etc.

For example, as is shown in FIG. **3**A, the view **315** of the obfuscated file **310** sent to "User A" **315** may still comprise both of the blocked out portions **306** and **307** form the original image **305**, e.g., owing to the fact that User A lacks either the appropriate access permissions and/or decryption keys to view the "true" content of portions **306** and **307** from the original image **305**. By contrast, the view **320** of the obfuscated file **310** sent to "User B" has only portion **306** (i.e., the portion corresponding to the head of person '1') blocked out from the view, e.g., owing to the fact that User B lacks either the appropriate access permissions and/or decryption keys to view the "true" content of portion **306** from the original image **305** but does have the appropriate access permissions and/or decryption keys to view the "true" content of portion **307** from the original image **305** (i.e., the portion corresponding to the head of person '2'). This is indicated by arrow **330** showing the protected portion corresponding to the head of person '2' being located, decrypted (if necessary), and seamlessly displayed at the correct coordinates within the image. It is also indicated by the presence of the key icon with the number 2 below the

"USER B" label in FIG. **3**A, denoting the fact that USER B has the necessary permissions and/or decryption keys to locate and decrypt hidden portion **311**. Finally, the view **325** of the obfuscated file **310** sent to "User C" has only portion **307** (i.e., the portion corresponding to the head of person '2') blocked out from the view, e.g., owing to the fact that User C lacks either the appropriate access permissions and/or decryption keys to view the "true" content of portion **307** from the original image **305** but does have the appropriate access permissions and/or decryption keys to view the "true" content of portion **306** from the original image **305** (i.e., the portion corresponding to the head of person '1'). This is indicated by arrow **340** showing the protected portion corresponding to the head of person '1' being located, decrypted (if necessary), and seamlessly displayed at the correct coordinates within the image. It is also indicated by the presence of the key icon with the number 1 below the "USER C" label in FIG. **3**A, denoting the fact that USER C has the necessary permissions and/or decryption keys to locate and decrypt hidden portion **309**.

Such a system allows a single version of the lossy file type, for example, JPEG file **305**, to be sent to multiple recipients, while access permission settings associated with each recipient allow each recipient to be able to use an authorized viewing application to seamlessly view only the particular sub-portion(s) of the file that they are authorized to see, while still maintaining the integrity of the lossy file type, such that it could be viewed in a standard, i.e., unauthorized, viewing application without any of the redacted portions of the file being visible.

Turning now to FIG. **3**B, an example of a lossy file type used to store hidden (and/or encrypted) content is shown in greater detail. In this example, the lossy file type is shown via exemplary JPEG object **350**, which shows one example of a JPEG file format structure. JPEG object **350** is comprises of a plurality of fields **352**-**378**. Data structures for lossy file formats are typically defined over time by the applicable standards settings bodies for each respective file format, and thus are not something that a particular user or system may be able to modify if they wish to have their files be readable/writeable/executable by industry standard viewers for the particular file format. For example, exemplary JPEG object **350** may begin with Start of the Image marker (SOI) property **352**. This property may then be followed by a plurality of Application Marker Sections (APPn) labeled **354**/**356**/**358**/**362**. Various other properties may also be present in a typical JPEG object file structure, e.g., DQT: Quantization Table (**366**); DHT: Huffman Table (**368**); DRI: Optional Restart Intervals (**370**); SOF: Frame Header (**372**); and SOS: Scan Header (**374**). The JPEG object file structure may also include a "Compressed Data" element **376**, where the actual compressed JPEG image data displayed to a user, e.g., JPEG image **310** from the example of FIG. **3**A, may be stored. Finally, exemplary JPEG object **350** may conclude with End of the Image marker (EOI) property **378**.

Among the various header elements **308** of JPEG object **350** may also be one or more fields **360**/**364** where the aforementioned "hidden" (and/or encrypted) redacted content from the image may be stored. In this example, the (optionally encrypted) "true" content from image portion **309** (i.e., the portion corresponding to the head of person '1') is stored in exemplary element **360**, and the (optionally encrypted) "true" content from image portion **311** (i.e., the portion corresponding to the head of person '2') is stored in exemplary element **364**. As may now be understood, one or more versions of each redacted portion of the lossy file type may be stored at one or more portions of the lossy file's data structure. Each such portion may be encrypted in such a fashion that only the desired recipient(s) are able to decrypt the respective portions. In some embodiments, only an authorized viewer application may know: 1.) where to look in the lossy file's data structure for "hidden" content; 2.) how to decrypt (if necessary) such hidden content intended for a particular recipient, and 3.) how and where (e.g., at what coordinates or at what time stamp) to "re-insert" the located and decrypted information into the original lossy file, so as to seamlessly present a view of the lossy file to the desired recipient that shows only those portions of the file that the sender intended the desired recipient to be able to see.

APC System Access Permission Settings Options

Several examples of potential APC system permissioning settings that may be applied to particular files of known lossy file types are shown and described below:

Public: Visible to the world. Searchable by search engines. Auto-broadcasted to the creator's "Followers." The "followers" of a particular user may be established by the followers that have been created within the APC file access permission setting system itself (if the recipients are users of such a system), or may be pulled in from third-party services, such as Facebook, Twitter, LinkedIn, etc.

Followers: Notifies and is visible to all followers of the creator.

Just Me: Private setting. Viewable only by user that creates the lossy file type.

My Contacts: All contacts available on user's contact list. The "contacts" of a particular user may be established by the contacts that have been created within the APC file access permission setting system itself, or may be pulled in from third-party services or applications, such as Gmail, Yahoo! Mail, Outlook, etc.

Level 1 Contacts: All registered-user contacts who have directly connected with the creator via the APC file access permission setting system itself, e.g., by accepting an invitation from the creator to become a contact. This access permission setting may be thought of as being bi-directional, e.g.: 1.) User A invites User B, and User B accepts; 2.) User B invites User A, and User A accepts. In some embodiments, all "Level 1" contacts of a user may be automatically added to that user's "My Contacts" list.

Level 2 Contacts: Direct contacts of the user's Level 1 contacts.

Level 3 Contacts: Direct contacts of user's Level 2 contacts.

Groups: Users may create one or multiple custom groups for use with the APC access permission setting system.

Custom: Users may manually add contacts, e.g., using an email address or name. The APC file access permission setting system may then auto-suggest users based on name entry (if the name is present in the user's "My Contacts" list). Lossy file types that have a custom access permission setting system associated with them will then only be viewable by the particular users whose information may be added to the custom authorization list for the lossy file type.

As will be understood, the settings levels described above are merely exemplary, and other ways of specifying access permission setting schemes may be used in particular implementations of an APC file access permission setting system.

Customized Privacy and Access Permission Setting Using Encryption Keys

FIG. 4 shows an example of a customized APC system that defines access permission setting for one or more users using encryption keys, according to one or more disclosed embodiments. In an embodiment, any encryption methodology may be used such as, for example, AES encryption, or Key-Policy Attribute-Based Encryption (KP-ABE), but other similar encryption methodologies are also contemplated within the scope of the embodiments. Public key database **400** comprises an association of user profiles and public keys associated with those users. User A in public key database **400** may refer to the sender in the scenario described above with reference to FIG. **3**, whereas Users B-N may refer to potential desired recipients in the scenario described above with reference to FIG. **3**. User contact info database **410** comprises an association of user profiles and contact information associated with those users. Again, user A in contact info database **410** may refer to the sender in the scenario described above with reference to FIG. **3**, whereas Users B-N may refer to potential desired recipients in the scenario described above with reference to FIG. **3**.

According to some embodiments of the customized privacy settings system described herein, a user may define the recipients for a particular lossy file type, for example, recipients that can view one or more portions of the JPEG file based on user settings. The user may set privacy setting for a particular user, public (e.g., universally viewable) or to a particular group of recipients.

According to one embodiment of a method of utilizing user-defined privacy settings for file sharing, first, the user, e.g., User A as shown in FIG. **4**, may create a JPEG file or, alternatively, may select a JPEG file that user A desires to send. Next, the user may choose the user or users that are user A's desired recipients for the selected JPEG file, e.g., User B. Next, the user A contact information, e.g., "Contact Info B" in the contact info database **410** of FIG. **4**, is matched to the user or users that are the desired recipients of the document. Next, each desired recipient user's information may be found in the public encryption key database, e.g., "Public Key B" in public key database **400** of FIG. **4**. Finally, the located public key, e.g., "Public Key B," that may include a set of attributes associated with each user is used to encrypt protected portions in an original JPEG file to create encrypted ciphertext. Further, an edited or obfuscated JPEG file is created that includes the one or more protected portions with coordinates of the JPEG file that may be replaced with black, blur, or the like to create an obfuscated JPEG file. Information that defines whether the obfuscated file is current and/or access code to access a current version of the obfuscated file may also be transmitted as metadata within the obfuscated JPEG file. The encrypted protected portions of the JPEG file and the obfuscated JPEG file may be sent to each of the desired recipients (either separately, or with the protected portions embedded or "hidden" within the obfuscated JPEG file's data structure, as discussed above), who may then use their private keys based on the user attributes to locate and decrypt the encrypted portions of the JPEG file and selectively replace the obfuscated portions with protected information from the decrypted JPEG file based on the user attributes. In some embodiments, attributes can include time of day (viewing at a specific time), location (viewing within a specific location or distance to a GPS location), or the like.

FIG. **5** is a flowchart that depicts a method or process flow **500** for utilizing the APC process from a sender's perspective, according to one or more embodiments. Particularly, flow **500** may be used by a user to attach an obfuscated file, created through user edits, and send the obfuscated file together with optionally encrypted portions of the original file to user recipients (or groups of recipients) associated with on-system or off-system client devices. Flow **500** begins in **502** and, next, sender may be prompted to select a JPEG image from an image selection screen (step **504**). The image selection screen may display one or more JPEG images that may be received from, for example, sync server **105** or from a storage location on the on-system client associated with the sender. Sender may determine whether to take a new picture, e.g., using his or her device's camera application (step **506**). For example, the sender may either capture a new image (i.e., step **506**="Y") at step **508** or simply select an existing image from his or her device (and or accessible third party storage) (i.e., step **506**="N") before proceeding to step **510**, where the on-system client may display an APC access permission settings dialog box.

Next, system may determine whether to apply APC access permission settings to the JPEG image (step **512**). Access permission settings may be manually received by the sender or may be automatically set based on predefined permissions for the sender that are defined for one or more users in contacts. If APC access permission settings are not applied to the JPEG image (i.e., step **512**="N"), then the sender may save the JPEG image as a "normal" JPEG image file (step **514**) and may use the sync server **105** to transmit the JPEG image file to one or more recipients (step **518**).

However, if APC access permission settings are to be applied to the JPEG image (i.e., step **512**="Y"), then the sender may open the JPEG image in a viewer, e.g., using an on-screen client (step **516**).

Next, the sender may determine whether to protect one or more portions of the JPEG image (step **520**). If the JPEG image is not to be protected (i.e., step **520**="N"), then, step **520** proceeds to step **518** where the sender may send the unprotected JPEG file as a "normal" JPEG image file.

However, if one or more portions of the JPEG image are to be protected (i.e., step **520**="Y"), then the sender may select one or more protected portions of the JPEG image to protect and replace with obfuscation in a copy of the JPEG image file (step **522**). Obfuscation can include selection of edits to the pixels in the image that can include blur, color-out, scratch-out, or the like at the coordinate locations within the code that defines the data for the coordinate locations in the JPEG image.

Next, sender may determine whether there are one or more recipients that may receive access to the full content of the JPEG image (step **524**). The recipients for the entire contents of the full JPEG image file can include recipients in the sender's contacts, direct contacts of the user contacts through level 1, 2 and 3, or custom contacts identified by the sender or system, as discussed above. If one or more recipients are selected (i.e., step **524**="Y"), then the system may determine if any recipients are on-system users (step **526**). If the recipients are on-system users (i.e., step **526**="Y"), then, step **526** proceeds to step **532** where the system may request public keys for a particular encryption methodology for each selected recipient (if so desired).

However, if any recipients are off-system users (i.e., step **526**="N"), then step **526** proceeds to step **528** where system **528** may request that a phantom user identifier be created for contacts associated with the off-system recipients. Next, sync server **105** may generate a specialized deep-link associated with the phantom user identifier (step **530**). The deep-link may include a hypertext link to a page on a web site that includes, in some embodiments, information for logging-into the system for accessing information associated with the JPEG image file having hidden (and optionally encrypted) content and entering user credentials associated with the user recipient. Step **530** proceeds to step **532** where system requests public keys for the identified recipients.

However, if there is at least one recipient that receives restricted access to content in the JPEG file (i.e., step **524**="N"), then the system may generate, in an embodiment, APC encryption keys for encrypting distinct protected portions of the JPEG file and APC identifiers associated with the respective keys (step **534**). APC identifiers may include access codes that identify the current protection settings for the JPEG file. The APC keys may be generated according to a Key Policy Attribute Based Encryption (KP-ABE) methodology. In KP-ABE, APC encryption keys and ciphertext (i.e., encrypted protected portions) are each labeled with descriptive attributes that controls which ciphertexts a user recipient is able to decrypt. Attributes that match may provide a user with the requisite access to the protected sections of the lossy file. It is to be understood that, although KP-ABE encryption is discussed here, any suitable form of asymmetric encryption may be utilized to encrypt the image file and/or portions of the image file. Further, any suitable key size, e.g., 128, 192, or 256 bits, may be used, based on a particular implementation of the APC system.

Next, each APC key and recipient's public key may be encrypted and sent to the desired recipients in a message, text, or the like (step **536**). The selected coordinates that define protected portions of the JPEG image that were identified in step **522** are extracted and replaced in the original JPEG file with obfuscated portions (step **538**). The original JPEG file can include one or more protected portions, including the entire JPEG image. As discussed above, obfuscation can include selection of edits to the pixels in the image that can include blur, color-out, scratch-out, or the like at the coordinate locations within the code that defines the data for the coordinate locations in the JPEG image. The obfuscated JPEG information is saved as an obfuscated JPEG file.

Next, the protected portions from the original JPEG file with the extracted information may be encrypted with the APC key and added back into the file structure of the original JPEG file to create an encrypted protected JPEG file, e.g., in an unused header element, as discussed with reference to FIG. 3B above (step **540**).

The image file with the hidden and optionally encrypted protected portions may then be attached to a message and appended (if so desired) with the specialized deep-link generated in step **530** (step **542**). Process **500** ends when message is sent to the desired recipient (step **544**).

FIGS. **6A-6B** depict flowcharts that show a process **600** from a receiver's or recipient's perspective, according to one or more embodiments. Process **600** begins in step **602**, and in step **604**, a receiver may view (e.g., in a third-party email client or a web browser) a message with an image attachment that is received, e.g., in a known lossy file format. In **610**, the receiver may determine whether to download and open the attachment in an off-system device, e.g., via a plug-in. If the receiver chooses to open and/or download the attachment (i.e., step **610**="Y"), then the receiver may load the file in an appropriate 'off-system' image viewer on the receiver client device (step **612**). However, if the receiver chooses not to download or open the attachment in an appropriate 'off-system' viewer, (step **610**="N"), then the receiver may click the deep-link in the message to receive access to the JPEG image (step **606**). Next, clicking the deep-link redirects the receiver to a web page associated with the deep-link (step **608**), e.g., an 'on-system' web page. For example, the redirected web page may be a landing page for an account that is associated with the on-system network of the receiver or a login page for an on-system network to establish an account in the system. Next, the system determines whether the receiver is logged into the system (Step **614**). If the receiver is logged into the system (i.e., step **614**="Y"), process **614** proceeds to step **630** (which follows on to FIG. **6**B).

However, if the receiver is not logged into the system (i.e., step **614**="N"), then the system may load a login web page that prompts the receiver to input the receiver's credentials for authentication (step **616**). Next, the receiver may sign-on to create on-system credentials or may enter on-system credentials for an existing account in the system (step **618**). If the receiver signups to be an on-system user (i.e., step **618**="Y"), then, the receiver enters credentials in the system to create an account as an on-system user (step **620**). If the credentials are successfully entered into the system (i.e., step **622**="Y"), the webpage associated with the deep-link is loaded in a viewer for the on-system receiver (step **628**). However, if the credentials are not successful (i.e., step **622**="N"), the webpage may be redirected to a login webpage for reentry of user credentials and/or an error message may be provided (step **616**). Step **628** proceeds to step **630**.

If the receiver logs into the system as an on-system user (i.e., step **618**="N"), then the receiver may enter on-system credentials to be authenticated in the system (step **624**). If the credentials are accepted (i.e., step **626**="Y"), then the deep-link hyperlink may be loaded in an on-system viewer associated with the on-system receiver (step **628**). Step **628** proceeds to step **630** (which follows on to FIG. **6**B).

Referring now to FIG. **6**B, step **630** proceeds to step **632** where a receiver may view the message having the attachment in a compatible viewer on a receiver's authorized client device. The attachment can be an unprotected JPEG file or a obfuscated JPEG file with the associated extracted (and optionally encrypted) protected portions of the original JPEG file "hidden" in one or more parts of the JPEG file's data structure. Next, the client device may open the attachment when the user selects the attachment (step **634**). Next, the system determines whether the attachment includes APC access permission settings (step **636**). If APC access permission settings are not applied to the attachment (i.e., step **636**="N"), then the original JPEG file may be displayed on a compatible viewer on the receiver's client device (step **646**). Step **646** proceeds to step **648** where process ends.

However, if APC access permission settings have been applied to the attachment (i.e., step **636**="Y"), then the APC identifier from the attachment, as well as the user identifier, may be transmitted to the sync server (step **638**). Next, the system may validate the receiver by comparing the user identifier and APC identifier associated with the attachment with information that is stored on the server (step **640**). If the user does not have access to view the attachment (i.e., step **642**="N"), then the receiver may receive an error message that the client device for the receiver has an invalid APC identifier (step **644**). Next, the attachment may be opened as an obfuscated JPEG file in a compatible viewer on the receiver's client device (step **646**).

However, if user access is granted (i.e., step **642**="Y"), the system may determine if file access for the JPEG file in the attachment is still available (step **650**). File access may be determined using the APC identifier. If APC file access is not the most current (i.e., step **650**="N"), then the server may transmit a conditional valid access code and file update link to the receiver client. The conditional access code may be used to validate whether the permissions in the container file are current (step **652**). Next, a new file may be received by the receiver's client device upon selecting file update link (step **654**).

However, if the APC file access is current (i.e., step **650**="Y"), the server responds with a valid access code that is transmitted to the viewer (step **656**). Next, the receiver's client device may check the format of one or more JPEG files that are received from the server against the registry on the receiver's client device (step **658**). The Registry can include settings for applications that may be used to access the information in the JPEG file. Next, the container file may be unpacked by the receiver's client device (step **660**). After unpacking the container file, the APC data in the encrypted protected portion(s) may be isolated from the rest of the file (step **662**). Next, the receiver's client device decrypts the encrypted protected portion(s) with the valid APC key (step **664**). Particularly, the receiver's client device may decrypt the encrypted APC data with the receiver's client device private key to retrieve the user public keys (APC key). Once the APC key is decrypted, the APC keys may be used to decrypt the encrypted protected portion(s) if the attributes in the ciphertext match the attributes in the APC keys. If the APC data is not decrypted (i.e., step **666**="N"), then the obfuscated JPEG file may be displayed in a compatible viewer on the receiver client device (step **646**).

However, if the APC data is successfully decrypted (i.e., step **666**="Y"), then the receiver's client device may replace one or more pixel coordinates in an obfuscated region for the obfuscated JPEG file with the decrypted APC data that represent information at the same pixel coordinates that were obfuscated (step **668**). Next, the JPEG file that is created with one or more replaced coordinates may be displayed as a regenerated JPEG object or image in an appropriate viewer on the receiver's authorized client device (step **670**). Step **670** ends after proceeding to step **672**.

EXAMPLES

The following examples pertain to further embodiments.

Example 1 is a non-transitory computer readable medium comprising computer executable instructions stored thereon that when executed cause one or more processing units to: receive an indication of a first protected portion and a second protected portion of a file of a lossy file type; receive first and second respective access permission settings for each of the first and second protected portions; receive an indication of a first recipient for the first protected portion; receive an indication of a second recipient for the second protected portion, generate an edited copy of the file based on the indication of the first and the second protected portions to create an edited lossy file; add the first and second protected portions, associated with their respective access permission settings, back into the file structure of the edited lossy file; and transmit the edited lossy file to the first and the second recipients.

Example 2 includes the subject matter of Example 1, wherein the instructions further include instructions to cause the one or more processing units to: generate the edited lossy file by obfuscating at least one of the first protected portion and the second protected portion.

Example 3 includes the subject matter of Example 1, wherein at least one of the first protected portion and the second protected portion comprises a sub-portion of the file.

Example 4 includes the subject matter of Example 2, wherein the instructions to obfuscate at least one of the first protected portion and the second protected portion further

comprise instructions to mask the content of the file at the first protected portion and the second protected portion, respectively.

Example 5 includes the subject matter of Example 1, wherein the instructions further include instructions to cause the one or more processing units to: encrypt at least one of the first and second protected portions based, at least in part, on the first and the second access permission settings, respectively.

Example 6 includes the subject matter of Example 5, wherein the instructions to encrypt at least one of the first and second protected portions further comprise instructions to cause the one or more processing units to: encrypt at least one of the first and second protected portions based, at least in part, on the first and the second recipients, respectively.

Example 7 includes the subject matter of Example 1, wherein the lossy file type is an image file type.

Example 8 includes the subject matter of Example 7, wherein the instructions to add the first and second protected portions, associated with their respective access permission settings, back into the file structure of the edited lossy file further comprise instructions to: insert the first and second protected portions, associated with their respective access permission settings, into one or more header elements of the image file type's data structure.

Example 9 includes the subject matter of Example 1, wherein the edited lossy file is configured such that the first and second protected portions are only viewable by a recipient in an authorized viewing application.

Example 10 includes the subject matter of Example 1, wherein the first access permission setting comprises an indication that the first recipient may access the first protected portion but not the second protected portion.

Example 11 includes the subject matter of Example 1, wherein the second access permission setting comprises an indication that the second recipient may access the second protected portion but not the first protected portion.

Example 12 is a system, comprising: a memory, and one or more processing units, communicatively coupled to the memory, wherein the memory stores instructions to configure the one or more processing units to: receive an indication of a first protected portion of a file of a lossy file type; receive a first access permission setting for the first protected portion; receive an indication of a first recipient for the first protected portion; generate an edited copy of the file based on the indication of the first protected portion to create an edited lossy file; add the first protected portion, associated with its access permission settings, back into the file structure of the edited lossy file; and transmit the edited lossy file to the first recipient.

Example 13 includes the subject matter of Example 12, wherein the instructions are further configured to cause the one or more processing units to: generate the edited lossy file by obfuscating the first protected portion.

Example 14 includes the subject matter of Example 12, wherein the first protected portion comprises a sub-portion of the file.

Example 15 includes the subject matter of Example 13, wherein the instructions to obfuscate the first protected portion are further configured to cause the one or more processing units to mask the content of the file at the first protected portion.

Example 16 includes the subject matter of Example 12, wherein the instructions are further configured to cause the one or more processing units to: encrypt the first protected portion based, at least in part, on the first access permission setting.

Example 17 includes the subject matter of Example 16, wherein the instructions are further configured to cause the one or more processing units to: encrypt the first protected portion based, at least in part, on the first recipient.

Example 18 includes the subject matter of Example 12, wherein the lossy file type is an image file type.

Example 19 includes the subject matter of Example 18, wherein the instructions to add the first protected portion, associated with its access permission settings, back into the file structure of the edited lossy file further comprise instructions to: insert the first protected portion, associated with its respective access permission settings, into one or more header elements of the image file type's data structure.

Example 20 includes the subject matter of Example 12, wherein the edited lossy file is configured such that the first and second protected portions are only viewable by a recipient in an authorized viewing application.

Example 21 includes the subject matter of Example 12, wherein the first access permission setting comprises an indication that the first recipient may access the first protected portion.

Example 22 includes the subject matter of Example 12, wherein the first access permission setting comprises an indication that the first recipient may not access the first protected portion.

Example 23 is a computer-implemented method, comprising: receiving an indication of a first protected portion of a file of a lossy file type; receiving a first access permission setting for the first protected portion; receiving an indication of a first recipient for the first protected portion; generating an edited copy of the file based on the indication of the first protected portion to create an edited lossy file; encrypting the first protected portion based, at least in part, on the first access permission setting and the first recipient; adding the first encrypted protected portion, associated with its access permission settings, back into the file structure of the edited lossy file; and transmitting the edited lossy file to the first recipient.

Example 24 includes the subject matter of Example 23, wherein the act of generating the edited lossy file further comprises obfuscating the first protected portion.

Example 25 includes the subject matter of Example 23, wherein adding the first encrypted protected portion, associated with its access permission settings, back into the file structure of the edited lossy file further comprises inserting the first encrypted protected portion, associated with its respective access permission settings, into one or more header elements of the lossy file type's data structure.

In the foregoing description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the disclosed embodiments. It will be apparent, however, to one skilled in the art that the disclosed embodiments may be practiced without these specific details. In other instances, structure and devices are shown in block diagram form in order to avoid obscuring the disclosed embodiments. References to numbers without subscripts or suffixes are understood to reference all instance of subscripts and suffixes corresponding to the referenced number. Moreover, the language used in this disclosure has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter, resort to the claims being necessary to determine such inventive subject matter. Reference in the specification to "one embodiment" or to "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least one disclosed embodiment, and multiple references to "one embodiment" or "an embodiment" should not be understood as necessarily all referring to the same embodiment.

It is also to be understood that the above description is intended to be illustrative, and not restrictive. For example, above-described embodiments may be used in combination with each other and illustrative process steps may be performed in an order different than shown. Many other embodiments will be apparent to those of skill in the art upon reviewing the above description. The scope of the invention therefore should be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled. In the appended claims, terms "including" and "in which" are used as plain-English equivalents of the respective terms "comprising" and "wherein."

What is claimed is:

1. A non-transitory computer readable medium comprising computer executable instructions stored thereon that when executed cause one or more processing units to:

receive an indication of a first protected portion of a file of a lossy file type, wherein the first protected portion comprises a sub-portion of the file;

generate first and second versions of the first protected portion of the file, wherein at least a part of the first version and the second version are identical, and wherein at least a part of the first version and the second version differ from each other;

receive a first access permission setting for the first version of the first protected portion of the file;

receive a second access permission setting for the second version of the first protected portion of the file;

receive an indication of a first recipient for the first version of the first protected portion;

receive an indication of a second recipient for the second version of the first protected portion;

generate an edited copy of the file that is obfuscated at a code level by generating modified pixel values corresponding to a location of the first protected portion to create an edited lossy file, wherein the obfuscation further comprises masking an original content of the file at the location of the first protected portion;

encrypt the first version of the first protected portion based, at least in part, on the first access permission settings and the first recipient;

encrypt the second version of the first protected portion based, at least in part, on the second access permission settings and the second recipient;

add the encrypted first and second versions of the first protected portion, associated with their respective access permission settings, back into the edited lossy file as hidden data within a data structure of the edited lossy file, wherein at least a part of the first version also masks the original content of the file at at least part of the location of the first protected portion; and

transmit the edited lossy file to the first and the second recipients.

2. The non-transitory computer readable medium of claim **1**, wherein the lossy file type is an image file type.

3. The non-transitory computer readable medium of claim **2**, wherein the instructions to add the first and second versions of the first protected portion, associated with their respective access permission settings, back into the edited lossy file as hidden data within the data structure of the edited lossy file further comprise instructions to: insert the first and second versions of the first protected portion, associated with their respective access permission settings, into one or more header elements of the image file type's data structure.

4. The non-transitory computer readable medium of claim **1**, wherein the edited lossy file is configured such that the first protected portion is only viewable by a recipient in an authorized viewing application.

5. The non-transitory computer readable medium of claim **1**, wherein the first access permission setting comprises an indication that the first recipient may access the first version of the first protected portion but not the second version of the first protected portion.

6. The non-transitory computer readable medium of claim **1**, wherein the second access permission setting comprises an indication that the second recipient may access the second version of the first protected portion but not the first version of the first protected portion.

7. A system, comprising:

a memory; and

one or more processing units, communicatively coupled to the memory, wherein the memory stores instructions to configure the one or more processing units to:

receive an indication of a first protected portion of a file of a lossy file type, wherein the first protected portion comprises a sub-portion of the file;

generate first and second versions of the first protected portion of the file, wherein at least a part of the first version and the second version are identical, and wherein at least a part of the first version and the second version differ from each other;

receive a first access permission setting for the first version of the first protected portion;

receive an indication of a first recipient for the first version of the protected portion;

receive a second access permission setting for the second version of the first protected portion;

receive an indication of a second recipient for the second version of the protected portion;

generate an edited copy of the file that is obfuscated at a code level by generating modified pixel values corresponding to a location of the first protected portion to create an edited lossy file, wherein the obfuscation further comprises masking an original content of the file at the location of the first protected portion;

encrypt the first version of the first protected portion based, at least in part, on the first access permission settings and the first recipient;

encrypt the second version of the first protected portion based, at least in part, on the second access permission settings and the second recipient;

add the encrypted first and second versions of the first protected portion, associated with their respective access permission settings, back into the edited lossy file as hidden data within a data structure of the edited lossy file, wherein at least a part of the first version also masks the original content of the file at at least part of the location of the first protected portion; and

transmit the edited lossy file to the first recipient and the second recipients.

8. The system of claim **7**, wherein the instructions are further configured to cause the one or more processing units to: encrypt the first version of the first protected portion based, at least in part, on the first access permission setting.

9. The system of claim **8**, wherein the instructions are further configured to cause the one or more processing units to: encrypt the first version of the first protected portion based, at least in part, on the first recipient.

10. The system of claim **7**, wherein the lossy file type is an image file type.

11. The system of claim **10**, wherein the instructions to add the first version of the first protected portion, associated with its respective access permission settings, back into the edited lossy file as hidden data within the data structure of the edited lossy file further comprise instructions to: insert the first version of the first protected portion, associated with its respective access permission settings, into one or more header elements of the image file type's data structure.

12. The system of claim **7**, wherein the edited lossy file is configured such that the first protected portion is only viewable by a recipient in an authorized viewing application.

13. The system of claim **7**, wherein the first access permission setting comprises an indication that the first recipient may access the first version of the first protected portion.

14. The system of claim **7**, wherein the first access permission setting comprises an indication that the first recipient may not access the second version of the first protected portion.

15. A computer-implemented method, comprising:

receiving, by one or more hardware processor, an indication of a first protected portion of a file of a lossy file type, wherein the first protected portion comprises a sub-portion of the file;

generate, by the one or more hardware processor, first and second versions of the first protected portion of the file, wherein at least a part of the first version and the second version are identical, and wherein at least a part of the first version and the second version differ from each other;

receiving, by the one or more hardware processor, a first access permission setting for the first version of the first protected portion;

receiving, by the one or more hardware processor, an indication of a first recipient for the first version of the first protected portion;

receiving, by the one or more hardware processor, a second access permission setting for the second version of the first protected portion;

receiving, by the one or more hardware processor, an indication of a second recipient for the second version of the first protected portion;

generating, by the one or more hardware processor, an edited copy of the file that is obfuscated at a code level by generating modified pixel values corresponding to a location of the first protected portion to create an edited lossy file, wherein the obfuscation further comprises masking an original content of the file at the location of the first protected portion;

encrypting, by the one or more hardware processor, the first version of the first protected portion based, at least in part, on the first access permission setting and the first recipient;

encrypting, by the one or more hardware processor, the second version of the first protected portion based, at least in part, on the second access permission setting and the second recipient;

adding, by the one or more hardware processor, the encrypted first and second versions of the first protected portion, associated with their respective access permission settings, back into the edited lossy file as hidden data within a data structure of the edited lossy file, wherein at least a part of the first version also masks the original content of the file at at least part of the location of the first protected portion; and

transmitting, by the one or more hardware processor, the edited lossy file to the first recipient and the second recipient.

16. The method of claim **15**, wherein adding the encrypted first version of the first protected portion, associated with its respective access permission settings, back into the edited lossy file as hidden data within the data structure of the edited lossy file further comprises inserting, by the one or more hardware processor, the encrypted first version of the first protected portion, associated with its respective access permission settings, into one or more header elements of the lossy file type's data structure.

* * * * *