(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 October 2020 (22.10.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/214157 A1

(51) International Patent Classification:
H04L 29/06 (2006.01) *G06Q 30/02* (2012.01)
H04L 29/08 (2006.01)

(21) International Application Number:
PCT/US2019/027737

(22) International Filing Date:
16 April 2019 (16.04.2019)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **GOOGLE LLC** [US/US]; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(72) Inventors: **WANG, Gang**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **GAO, Yian**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **YUNG, Marcel**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(74) Agent: **LANZA, John D.** et al.; Foley & Lardner LLP, 3000 K Street N.W., Suite 600, Washington, District of Columbia 20007 (US).

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

WO 2020/214157 A1

(54) Title: SELF-AUTHENTICATING DOMAIN SPECIFIC BROWSER IDENTIFIERS

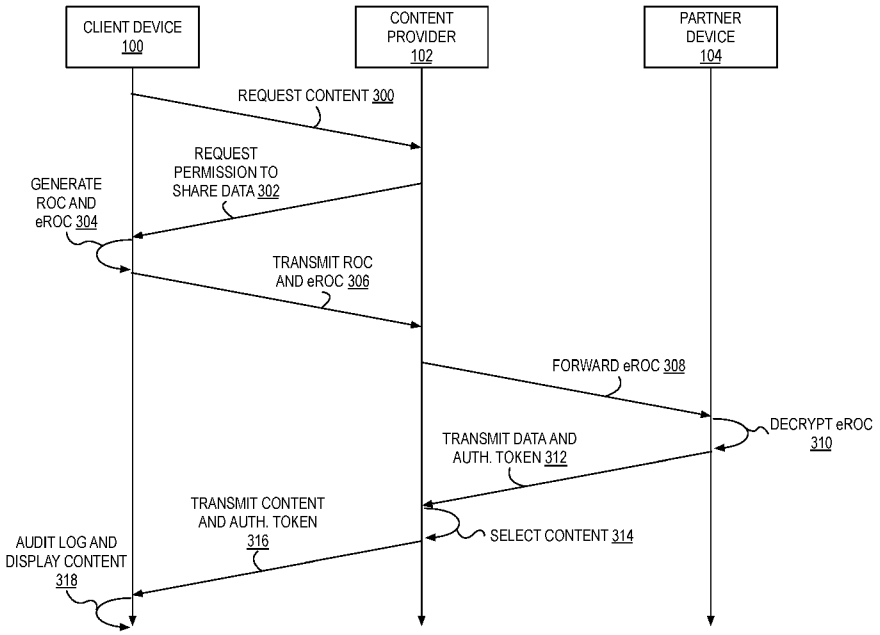


FIG. 3

(57) Abstract: The present disclosure provides a secure, user-transparent, and highly efficient content provider-specific identifier read-only cookie ("ROC"). These ROCs include the content provider-specific identifier of the client device and are generated by the client device (100) and encrypted with a public key of the content provider (102) or partner device (104) to create an encrypted ROC, eROC, preventing third parties from indirectly identifying matches. The ROCs can include authentication token to enable the client device to determine whether the ROC was cached or otherwise misused by a content provider or other third-party. Content from a partner device is displayed (318) only in case the authentication token is verified. The partner device comprises content that for example personalises a web page retrieved from the content provider.

[Continued on next page]

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

Published:

— *with international search report (Art. 21(3))*

SELF-AUTHENTICATING DOMAIN SPECIFIC BROWSER IDENTIFIERS

BACKGROUND

[0001] Browser cookies, or short device-, user-, or session-specific identifiers, are used for uniquely (or semi-uniquely) identifying a device, user, or session to a content provider. These cookies are typically generated by the content provider, e.g., during a first session or login process, and provided in subsequent requests from the browser or other application of the client device. The publisher can then associate each request with the same identifier (and implicitly, the same device, user, or session), allowing customization of content selection.

[0002] However, typical browser cookies have some critical drawbacks. Third-party content providers that provide embedded content within a website can provide cookies to track a browser, without the user necessarily being aware that they are being identified or tracked by these third parties (e.g., because the address shown in the browser address bar can only identify the source of the main page and not necessarily the embedded content). Thus, it can be difficult for users to identify potential privacy issues and control privacy of their personal information.

SUMMARY

[0003] According to at least one aspect of the disclosure, a system to control data exchange can include an encryptor. The encryptor can, for each partner domain in a list of partner domains, generate a partner domain-specific identifier and encrypt the partner domain-specific identifier and a first authentication token using an encryption key of the corresponding partner domain. The system can include a network interface to transmit to a content provider the encrypted partner domain-specific identifiers for each partner domain in the list of partner domains. The network interface can receive, from the content provider, a content item selected by the content provider based on data from each partner domain in the list of partner domains. The data from each partner domain in the list of partner domains is selected based on receiving the encrypted partner domain-specific identifier. The network interface can receive, by the client device from the content provider, a second authentication token for each partner domain in the list of partner domains. The system can include an audit engine to determine whether to display the content item based on a comparison of the first authentication token for each partner domain in the list of

partner domains and the second authentication token for each partner domain in the list of partner domains.

[0004] According to at least one aspect of the disclosure, a method to control data exchange can include receiving, by a client device from a content provider, a list of partner domains. The method can include, for each partner domain in the list of partner domains, generating, by the client device, a partner domain-specific identifier and encrypting, by the client device, the partner domain-specific identifier and a first authentication token using an encryption key of the corresponding partner domain. The method can include transmitting, by the client device to the content provider, the encrypted partner domain-specific identifiers for each partner domain in the list of partner domains. The method can include receiving, by the client device from the content provider, a content item selected by the content provider based on data from each partner domain in the list of partner domains. The data from each partner domain in the list of partner domains is selected based on receiving the encrypted partner domain-specific identifier. The method can include receiving, by the client device from the content provider, a second authentication token for each partner domain in the list of partner domains. The method can include displaying, by the client device, the content item based on a comparison of the first authentication token for each partner domain in the list of partner domains and the second authentication token for each partner domain in the list of partner domains.

[0005] These implementations are mentioned not to limit or define the scope of the disclosure, but to aid in understanding it. Particular implementations can be developed to realize one or more of the following advantages. By replacing traditional browser cookies with encrypted content provider-specific identifiers, the user of the client device has greater control over private data, including data exchanges between partner content providers, with a transparent audit log that identifies any such exchanges. The encrypted content provider-specific identifiers additionally allow the client device to automatically control modification of content, and in particular to control whether particular third parties are permitted to modify content for display. Furthermore, by leveraging cookie generation and encryption on the client device, resource consumption from large cookie matching tables at content providers is avoided, increasing efficiency. The elimination of tracking pixels and dual-round trip communications for cookie

matching also increases efficiency and better protects user privacy, allowing the client browser or application to render pages faster, with less battery and processor utilization.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The details of one or more implementations are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the disclosure will become apparent from the description, the drawings, and the claims, in which:

[0007] FIG. 1 illustrates a block diagram of a network and device environment for data exchange, according to some implementations;

[0008] FIG. 2 illustrates a block diagram of implementations of computing devices for use in the network and device environment illustrated in FIG. 1, according to some implementations;

[0009] FIG. 3 illustrates a flow diagram for a client device to filter content requests, according to some implementations; and

[0010] FIGS. 4 and 5 illustrate flow charts of a method for data exchange with a self-enforcing permissions; according to some implementations.

[0011] Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0012] Browser cookies, or short device-, user-, or session-specific identifiers, are used for uniquely (or semi-uniquely) identifying a device, user, or session to a content provider, such as a webpage publisher, advertiser, or other such entity. These cookies are typically generated by the content provider or other devices during a first session or login process. The cookies can be provided in subsequent requests from the browser or other application of the client device. The publisher can then associate each request with the same identifier (and implicitly, the same device, user, or session), allowing customization of content selection.

[0013] However, typical browser cookies have some critical drawbacks. Third-party content providers that provide embedded content within a website can provide cookies to track a browser, without the user necessarily being aware that they are being identified or tracked by these third parties (e.g., because the address shown in the browser address bar can only identify the source of the main page and not necessarily the embedded content). Thus, it can be difficult for users to identify potential privacy issues and control privacy of their personal information. Privacy control can be technically difficult because it is technically challenging to enforce how data is controlled once provided to a second party. For example, the second party can share the data with a third party without consent from the original provider of the data.

[0014] The systems and methods described herein can increase user security and privacy while enabling customized content selection. Additionally, the current systems and methods can reduce the number of transmissions required to select or customize content to save bandwidth and other computational resources. The customization of content can be performed based on data from a plurality of different domains. Each of the domains can provide the client device with cookies that enable preferences of the client device to be stored and later retrieved. The preferences from a plurality of different domains can be used in the selection and customization of content. The systems and methods described herein can filter, limit, or prevent content requests. Filtering selected content requests to third parties enables a user to control customization of content by third parties. However, limiting the additional content requests saves transmission bandwidth and also increases user privacy.

[0015] The systems and methods discussed herein provide a replacement for traditional browser cookies with a secure, user-transparent, and highly efficient content provider-specific identifier, which can be referred to as a "read-only cookie" (ROC). The ROCs can be generated by the client device and encrypted with a public key of a third-party domain (e.g., a content provider). The encrypted ROCs can be referred to as eROCs. The encryption of the ROC prevents unintended third-party domains from reading the ROC.

[0016] The generation of the ROCs can be controlled by user policies, such that identifiers are only created for third parties with compliant terms of service (ToS) that are retrievable from a predetermined address within the third party's domain; third parties that are on a whitelist (e.g.,

for which the user has explicitly provided consent); and/or third parties that are not on a blacklist (e.g., for which the user has explicitly refused consent).

[0017] Once the ROC is generated, violations in ToS agreements can be difficult to detect. For example, a third party may have a valid ToS agreement; however, the third party may cache a second third party's ROC or eROC in violation of the ToS. In another example, a white-listed third party could share the ROC with a second third party that is listed on a blacklist. The present systems and methods can identify privacy violations and prevent subsequent ROC generation of violating third-party domains. Additionally, the systems and methods can prevent ROC generation for third parties that do not add to the selection of customized content, increasing the user's privacy.

[0018] For example, when selecting content for a client device, a content provider may interact with one or more partner domains. The partner domains can provide analytics or other data for the selection or customization of content. To request the data from the partner, under a ToS agreement, the content provider can be required to request permission to share data about the browser with the partner domains. The systems and methods described herein can provide a self-enforcing ROC that can provide an engineering-based enforcement of the ToS. The self-enforcing ROC can prevent, for example, content providers from caching the ROC for later use. The systems and methods described herein can generate eROCs that include one-time use authentication tokens. The authentication tokens can temporarily authorize data sharing between a content provider and partner domain.

[0019] When a content item is returned in response to a content request, the browser can audit the authentication tokens returned by the content provider with the content item. The browser can determine whether the content item was selected using cached ROCs, cached eROCs, or in violation of a ToS. If the content item fails the audit process, the browser can decide to not display the returned content.

[0020] FIG. 1 illustrates a block diagram of a network and device environment 50 for data exchange, according to some implementations. As illustrated, a client device 100 can communicate via the networks 106, 106' with one or more content providers 102 and one or

more partner devices 104(1)-104(*N*), which can generally be referred to as partner devices 104. The networks 106, 106' can collectively be referred to as the networks 106.

[0021] The network and device environment 50 can include one or more client devices 100. The client device 100 can include any type and form of computing device, including a desktop computer, laptop computer, portable computer, tablet computer, wearable computer, embedded computer, smart television, console, Internet of Things (IoT) device or smart appliance, or any other type and form of computing device. The client device 100 can execute an application to request content, such as a web browser, social media application, video game, or other such application. The client device 100 can request content and can provide a device identifier, cookie, or other such identifier such that a content provider 102 or partner device 104 can use to select customized content for the corresponding device or a user of the client device 100.

[0022] The network and device environment 50 can include one or more networks 106, 106'. The networks 106, 106' can include any type and form of network, including local area networks (LANs), wide area networks (WANs) such as the Internet, satellite networks, cable networks, broadband networks, fiber optic networks, microwave networks, cellular networks, wireless networks, or any combination of these or other such networks. The networks 106, 106' can be the same type of network or different types. The networks 106, 106' can be different portions of the same network. The networks 106, 106' can include a plurality of additional devices, including gateways, modems, firewalls, routers, switches, etc. The networks 106, 106' can also include any number of computing devices (e.g., computer, servers, routers, network switches, etc.) that are configured to receive and/or transmit data within the networks 106, 106'. The networks 106, 106' can include any number of hardwired and/or wireless connections. A client device 100 can communicate wirelessly (e.g., via Wi-Fi, cellular, radio, etc.) with a transceiver that is hardwired (e.g., via a fiber optic cable, a CAT5 cable, etc.) to other computing devices in the networks 106, 106'. In some implementations, the networks 106, 106' may be a virtual network, such as a virtual network between a plurality of virtual machines executed by a single physical machine, or an abstract network such as an offline transfer of data via physically movable media (e.g., a Sneakernet, transferring data via tape media, CD-ROM, flash media, external hard drives, floppy disks, etc.).

[0023] The network and device environment 50 can include one or more content providers 102. The content provider 102 can select content. The content provider 102 can select or receive the content from a pool of partner devices 104. For example, the content provider 102 can receive a request for content from the client device 100 and can select from among the partner devices 104 a partner device 104 to provide the requested content. The selection can be via load balancing algorithms, auction algorithms (e.g., with providers bidding for opportunities to provide content), etc. The content provider 102 can thus be referred to as an exchange server, a load balancer, an auction provider, or by any other such term. Although shown between networks 106, 106', the content provider 102 can be deployed on the same network segment as a partner device 104.

[0024] The network and device environment 50 can include one or more partner devices 104. The partner devices 104 can select content to provide to the client device 100, via the networks 106, 106', based on an identifier of the client device 100. The identifier can be one of the ROCs described herein. The partner devices 104 can also support the selection of content by the content provider 102 or other partner devices 104. For example, the partner devices 104 can provide analytics, preferences, content, and other information to the content provider 102 and other partner devices 104 that can enable the selection of customized or tailored content for the client device 100. The analytics, preferences, content, and other information that a partner device 104 provides to the content provider 102 or other partner devices 104 can generally be referred to generally as data.

[0025] The content provider 102 and the partner device 104 can utilize cookies, such as ROCs, to provide customized content to the client device 100. For example, when issuing a request for content, a client device 100 can provide a device identifier or cookie to the content provider 102. To maintain client privacy, the client device 100 can generate a private cookie, such as a ROC, for each of the content provider 102 and partner devices 104. To prevent caching of the cookie by the content provider 102, the partner devices' cookies can include a unique token that can only be extracted from the cookie by a respective partner device 104 or content provider 102. The unique token can be referred to as an authentication token. The authentication token can include random string of text or an encoded string. The encoded string can include a combination of a time stamp, public key, unique identifier, nonce, salt, domain identifier, or combination

thereof. The partner device 104 can return the authentication token when returning data or content to the content provider 102. The authentication token is then forwarded to the client device 100 as proof that the data from the partner device 104 was selected in response to the present request and not generated in response to a cached cookie.

[0026] The content provider 102 and the partner devices 104 can include any type and form of computing device, including desktop computers, servers, workstations, laptop computers, portable computers, embedded computers, or any other type and form of computing device. The content provider 102 and the partner devices 104 can include virtual machines executed by one or more physical computing devices and can be configured as a server farm, cluster, or cloud of devices.

[0027] FIG. 2 illustrates a block diagram of implementations of computing devices for use in the network and device environment 50. As described above, the network and device environment 50 can include one or more client device 100 that can receive content via the content provider 102. The content provider 102 can receive the content or other data from one or more partner devices 104.

[0028] As discussed above, the client devices 100 can be referred to variously as a client, device, client device, computing device, user device, or any other such term and can be a desktop computer, laptop computer, tablet computer, smartphone, video game console, smart television or set top box, server, workstation, or any other type and form of computing device capable of communicating over a network 106. In some implementations, a client device 100 can execute an application, service, server, daemon, routine, or other executable logic for communicating over a network 106, such as a web browser, mail client, video player, music player, video game, or any other such application. Such applications can include a command line interface, graphical user interface, or any combination of these or other interfaces. In implementations in which a client device is a smart television or set top box, the client device can receive content via a first interface, such as a terrestrial, satellite, or cable broadcast, and can communicate with an audience measurement server via a second interface via network 106, such as an Ethernet or Wi-Fi interface. In other implementations, client device 100 can receive content via network 106 and can transmit identifications of interactions via network 106.

[0029] The client device 100 can be any number of different types of user electronic devices configured to communicate via network 106, including, without limitation, a laptop computer, a desktop computer, a tablet computer, a smartphone, a digital video recorder, a set-top box for a television, a video game console, or any other type and form of computing device or combinations of devices. In some implementations, the type of client device 100 can be categorized as a mobile device, a desktop device or a device intended to remain stationary or configured to primarily access network 106 via a local area network, or another category of electronic devices, such as a media consumption device.

[0030] The client device 100 can include a processor 200 and a memory 206. The memory 206 can store machine instructions that, when executed by the processor 200, cause the processor 200 to perform one or more of the operations described herein. The processor 200 can include a microprocessor, ASIC, FPGA, etc., or combinations thereof. The processor 200 can be a multi-core processor or an array of processors. The memory 206 can include, but is not limited to, electronic, optical, magnetic, or any other storage devices capable of providing the processor 200 with program instructions. The memory 206 can include a floppy disk, CD-ROM, DVD, magnetic disk, memory chip, ROM, RAM, EEPROM, EPROM, flash memory, optical media, or any other suitable memory from which the processor 200 can read instructions. The instructions can include code from any suitable computer programming language, such as, but not limited to, C, C++, C#, Java, JavaScript, Perl, HTML, XML, Python, and Visual Basic.

[0031] The client device 100 can include one or more network interfaces 202. A network interface 202 can include any type and form of interface, including Ethernet, including 10BASE-T, 100BASE-T, or 1000BASE-T ("Gigabit"); any of the varieties of 802.11 wireless, such as 802.11a, 802.11b, 802.11g, 802.11n, or 802.11ac; cellular, including CDMA, LTE, 3G, or 4G cellular; Bluetooth or other short range wireless connections; or any combination of these or other interfaces for communicating with a network 106. The client device 100 can include a plurality of network interfaces 202 of different types, allowing for connections to a variety of networks 106 or a network 106 such as the Internet via different sub-networks.

[0032] The client device 100 can include one or more user I/O interfaces 204. The I/O interface 204 can be or can be connected to any electronic device that conveys data to a user by generating

sensory information (e.g., a visualization on a display, one or more sounds, tactile feedback, etc.) and/or converts received sensory information from a user into electronic signals (e.g., a keyboard, a mouse, a pointing device, a touch screen display, a microphone, etc.) The one or more user interface devices can be internal to the housing of client device 100, such as a built-in display, touch screen, microphone, etc., or external to the housing of the client device 100, such as a monitor connected to the client device 100, a speaker connected to the client device 100, etc., according to various implementations.

[0033] The client device 100 can include, in memory 206, an application 208. The client device 100 can execute the application 208 with a processor 200. The application 208 can be an application, applet, script, service, daemon, routine, or other executable logic for receiving content and displaying or otherwise outputting content via the I/O interface 204 (e.g., display, speaker) of the client device. The application 208 can be a web browser. The application 208 can include functionality for displaying content received via network interface 202 and/or generated locally by processor 200. In some implementations, the application 208 can be a media player or include an embedded media player, such as a plug-in or native media player within a web browser.

[0034] The client 100 can include or be identified with a device identifier 210. The device identifier 210 can be an alphanumeric string, data string, serial number, media access control (MAC) address, internet protocol (IP) address, username or account name, globally unique identifier (GUID), cookie, random or pseudorandom number, or any other type and form of identifier, including combinations of these or other identifiers. In some implementations, the device identifier 210 can be fixed to the device or preconfigured in the device, such as a manufacturer serial number or MAC address. The device identifier 210 can be dynamically set by a content provider, streaming server, application 208, user, or other entity. For example, the device identifier can be set and stored in a cookie or username.

[0035] In some implementations, a unique or new device identifier 210 can be set for each communication with a content provider 102 or partner device 104. In some implementations, the device identifier 210 can be static, time-based (e.g., changed periodically—hourly, daily, weekly), or interval-based (e.g., changed on restart of the client device or login to a service).

[0036] In some implementations, a device identifier 210 can be associated with one or more other device identifiers 210 (e.g., a device identifier for a mobile device, a device identifier for a home computer, etc.) The device identifier 210 can be provided by the content provider 102 or the partner device 104. In some implementations, the client 100 can request a device identifier 210 from the content provider 102 or partner device 104 and can transmit the device identifier 210 to the content provider 102 or partner device 104 in association with requests for content or measurement data.

[0037] The client device 100 can include an encryptor 214. The encryptor 214 can be any application, applet, script, service, daemon, routine, or other executable logic to generate ROCs and eROCs. The encryptor 214 can be used to calculate hash results based on inputs. The hash results can be used in (or as) the ROCs and eROCs. The encryptor 214 can include hardware, software, or a combination of hardware and software. In some implementations, the encryptor 214 can include an encryption module, a trusted platform module (TPM), an ASIC, or any other type and form of circuitry for performing encryption and cryptographic hash calculations.

[0038] As described above, the client device 100 can replace a traditional browser cookie with a secure, user-transparent ROC. The encryptor 214 can generate the ROC. The encryptor 214 can then encrypt the ROC with a public key of the content provider 102, partner device 104, or other third party to form an encrypted ROC (eROC). The use of the eROC can prevent unintended parties using the identifier contained within the eROC. For example, only the owner of the private key associated with the public key can decrypt the eROC to access the identifier. Encrypting the ROC can enable the client device 100 to transmit eROCs destined for the partner devices 104 to the content provider 102, which can forward the eROCs to the respective partner devices 104. As the eROCs are not encrypted with the content provider's public key (but rather the public key of the partner devices 104), the content provider 102 cannot use or access the identifier contained within the eROCs destined for the partner devices 104.

[0039] The generation of content provider-specific identifiers may be controlled by user policies, such that identifiers are only created for content providers with compliant terms of service (ToS); from content providers that are on a whitelist (e.g., for which the user has explicitly provided consent); and/or from content providers that are not on a blacklist (e.g., for

which the user has explicitly refused consent). In some implementations, a ToS can be compliant if retrievable from a predetermined address within a domain and the content of retrieved ToS is unchanged from the original ToS.

[0040] The encryptor 214 can generate the ROC and eROC using a cryptographic hash. The cryptographic hash can include any suitable hashing algorithm, such as MD5 and SHA256. The encryptor 214 can generate the ROC by hashing inputs that can include a browser-, application-, device-, or session-specific private identifier (e.g., an identifier specific to an instance of a browser for a particular user, separate from other users of the device) and a domain name or address of the recipient (e.g., the content provider 102 or partner device 104).

[0041] In some implementations, to provide additional salt, the hash input may also include a creation time or expiration time for the ROC. Other sources of entropy may be utilized to similarly increase the complexity of the hash inputs, such as device types or serial numbers, clock skew times, hardware identifiers, or any other such data. In some implementations, the "expiration time" may be a granular value, such as a number of weeks, months, or years before the content provider-specific identifier expires, allowing automatic regeneration by the client.

[0042] The encryptor 214 can generate an eROC by encrypting the ROC via a public key of the rightful owner or intended recipient (e.g., content provider 102 or partner device 104). The encryptor 214 can generate an eROC by first concatenating the ROC with a creation timestamp, a nonce, and encrypt the concatenation result with the public key of the recipient prior such that the resulting eROCs, even for a single domain, are highly varied.

[0043] The client device 100 can use the nonce as an authentication token. The encryptor 214 can generate a new authentication token each time the encryptor 214 generates an eROC. The encryptor 214 can generate the authentication token by concatenating the requesting domain, the recipient or partner domain, a timestamp, and a private identifier. The private identifier can be a domain-specific identifier. The encryptor 214 can encrypt the concatenated string to form the nonce. The encryptor 214 can encrypt the concatenated string with symmetric key encryption to form the nonce. The symmetric key encryption can include AES encryption or AES-based GCM encryption, among others. In some implementations, the intended recipient (e.g., the content

provider 102 or the partner device 104) can decrypt and extract the private identifier and the authentication token. The content provider 102 or the partner device 104 can use the private identifier to select or customize content.

[0044] The client device 100 can include an audit log 216. The audit log 216 can be a database stored in the memory 206. As the encryptor 214 generates authentication tokens, the encryptor 214 can store each authentication token in a lookup table as a value of a key-value pair. The key for the key-value pair can be a request identifier. For example, in response to a content request, the client device 100 can receive a request from the content provider 102 to initiate data sharing with a plurality of partner devices 104. The encryptor 214 can generate an eROC for each of the partner devices 104 and store the authentication tokens in the audit log 216 as values keyed to the request identifier. As described in relation to FIGS. 3 and 4, among others, the audit engine 212 can audit the content request before displaying the returned content by determining whether each of the partner devices 104 returned the valid authentication token, as indicated by the audit engine 212 finding a match to the returned authentication token in the audit log 216. The audit log 216 can also include another lookup table that provides an indication of the partner devices' relative contribution in selecting the returned content. The audit engine 212 can save the indication of the contribution in association with an identifier of the partner device 104.

[0045] The client device 100 can execute an audit engine 212. The audit engine 212 can include an application, server, service, daemon, routine, or other executable logic for auditing the responses to returned content requests. The audit engine 212 can audit the audit log 216 to determine whether the content provider 102 or the partner device 104 are violating a ToS by, for example, caching the eROCs or ROCs.

[0046] As described above, responsive to sending a content request to the content provider 102, the content provider 102 can request permission to share information or data with one or more partner devices 104 for the selection or customization of the content. The client device 100 can approve the request by generating a respective eROC for each of the partner devices 104. Each eROC can include an authentication token, which the encryptor 214 can store in the audit log 216. The audit log 216 can include a plurality of rows. Each of the rows can include a randomly generated key, which can used as the authentication token. The data saved to each of

the respective rows can be the data that is encoded with or in the authentication token. When the content provider 102 responds to the content request with a content item, the content provider 102 can forward the authentication token to the client device 100 from the partner device 104. As the eROC is encoded with the partner device's public key, only the partner device 104 (and not the content provider 102) can extract the authentication token from the eROC. The audit engine 212 can determine whether the authentication tokens are valid by accessing the audit log 216 and determining whether the authentication tokens correspond to the most recent content request. In some implementations, as described above, a timestamp can be included or embedded in the authentication token. The audit engine 212 can set a predetermined time out period for authentication tokens. If the timestamp of the returned authentication token is outside of the time out period, the audit engine 212 can determine that the authentication token is not valid. If the authentication token is not valid, indicating, for example, that the eROC was cached, the audit engine 212 can instruct the application 208 to not display the returned content. In some implementations, the authentication token can be "self-contained." For example, the authentication token can include the data required for the client device 100 (or other device) to authenticate the authentication token. For example, an indication of the request, partner domain, timestamp, or other data can be encrypted with a private key of the client device 100 and included in the authentication token. Upon receipt of the authentication token, the client device 100 can decrypt the authentication token to retrieve the encrypted data and use the decrypted data to authenticate the authentication token.

[0047] With the content, the content provider 102 can also transmit a data structure that includes the contribution of each partner device 104 in the selection of the content. The data structure can include a key-value pair indicating the contribution (value) of each partner device 104 (key). The contribution of a given partner device 104 to the selection of a content item can be negotiated between the partner device 104 and the content provider 102. After the partner device 104 is involved in a predetermined number of content requests, the audit engine 212 can calculate the average contribution of a partner device 104 to selecting or customizing content. Responsive to a future content request, the encryptor 214 may not generate an eROC for the partner device 104 if the partner device's contribution is below the predetermined threshold.

[0048] The network and device environment 50 can include a content provider 102. As with client devices 100, the content provider 102 can include one or more processors 200, memories or storage devices 206, network interfaces 202, and interfaces 204. In some implementations, the content provider 102 can be a headless server that does not include a user interface but can communicate with clients 100 or partner devices 104 via the network 106. In some implementations, memory 206 can store one or more applications for execution by processor 200 of the server, including FTP servers, web servers, mail servers, file sharing servers, peer to peer servers, or other such applications for delivering content or redirection commands to allow clients to access content at a content provider.

[0049] The content provider 102 can directly or indirectly select content response to a content request from the client device 100. For example, the content provider 102 can directly respond to the content request by selecting a content item from the content database 234 of the content provider 102. The content provider 102 can indirectly respond to the content request by forwarding the request to one or more partner devices 104 that can select the content or by receiving analytics and other data that enable the content provider 102 to select customized content.

[0050] The content provider 102 can select partner devices 104 from a list of partner devices. The content provider 102 can select the partner devices 104 via a round robin or other load balancing system or via an auction-based system. When one or more partner devices 104 are selected, the content provider 102 can transmit a request for permission to include the partner devices 104 in the selection of the requested content. In some implementations, the request can be included in webpage or other application where the content will be rendered. For example, the content slot where the content will be rendered can include a plurality of tags that identify partner devices 104 that will be used in the fulfilment of the content item. An indication of the partner devices 104 can be included in the header of the webpage. In response to the request for permission, the content provider 102 can receive, from the client device 100, an eROC for each partner device 104 that the client device 100 permits to be involved in the selection of the content item.

[0051] The content provider 102 can include a response table 220. The response table 220 can include a database, flat file, indexed file, or other type and form of data structure for storing authentication tokens received from the partner device 104. As described herein, the eROC can include an authentication token. When a partner device provides content or other data to the content provider 102, the partner device 104 can provide an instance of the authentication token to the content provider 102. The content provider 102 can store the authentication token into the response table 220. In association with the authentication token, the content provider 102 can also store an indication of the partner device's contribution to the selection of the content that the content provider 102 provides to the client device 100. When the content provider 102 provides the content to the client device 100, the content provider 102 can also provide an instance of the response table 220. The instance of the response table 220 transmitted to the client device 100 can include a key-value pair with the authentication token as the key and the partner device's contribution stored as the value in the response table 220.

[0052] Also as illustrated in FIG. 2, the network and device environment 50 can include one or more partner devices 104. A partner device 104 can include one or more computing devices connected to network 106 and configured to select or enable the selection of content in response to a content request. The partner devices 104 can provide the content item to the client device 100 directly or via the content provider 102. The partner device 104 can be a content provider, server, web server, data server, publisher, service provider, analytics system, or other such device. The partner device 104 can include a plurality of computing devices configured as a server farm or cloud and can include routers, load balancers, network address translators, firewalls, or other such devices. The partner devices 104 can be computer servers (e.g., FTP servers, file sharing servers, web servers, etc.) or combinations of servers (e.g., data centers, cloud computing platforms, etc.) The partner devices 104 can provide any type and form of content, including text, images, video, audio, multimedia, or other data, or any combination of these. Content can include live media content, prerecorded media content, rendered content, movies, television shows, podcasts, video blogs, video games or other interactive content, advertising in any format, social media, or any other type and form of content.

[0053] The partner devices 104 can include one or more processors 200, network interfaces 202, I/O interfaces 204, and/or memory devices 206. The partner devices 104 can include a plurality of computing devices, including virtual machines executed by physical devices. For example, a content provider 104 can comprise a plurality of virtual machines executed by one or more physical computing devices, each such virtual machine executing a partner device (e.g., web server, file server, streaming media server, etc.) and in communication with one or more storage devices, such as network storage devices, RAID storage devices, or other such devices.

[0054] The partner device 104 and the content provider 102 can include a content selector 230. Content selector 230 can include an application, service, server, daemon, routine, or other executable logic for selecting content from content storage 234 for delivery to a client device 100. The content selector 230 can select content for delivery to the client device 100 based on information about the client stored in an identifier database 232. For example, the database 232 can include information about the client device's capabilities (e.g., screen resolution or orientation, color depth, bandwidth, etc.) or any other such information for selection of customized content. The content stored in the database 232 can be index by ROC or a private identifier that is contained within, derived from, or otherwise associated with the ROC.

[0055] For example, upon receipt of an eROC from the client device 100, the content provider 102 and partner device 104 can decrypt the eROC to recover the ROC. The content selector 230 can identify the information about the client device 100 in database 232 via the ROC and select corresponding content from the content storage 234. The content selector 230 can push or stream the content to the client device 100, provide a unique resource identifier (URI) or address of content to the client device 100 such that the client application 208 can subsequently request the content for delivery and rendering (e.g., from a web server of content provider 104), or provide the URI or address to content provider 102 for forwarding to the client device.

[0056] The content storage 234 can be provided by one or more additional storage devices in communication with the content provider 102 or partner device 104. The content storage 234 can include any type and form of data, including audio, video, animations, text, multimedia, still or animated graphics, executable scripts, or any other type and form of content.

[0057] The identifier database 232 can include a database, flat file, indexed file, or other type and form of data structure for associating information about a client device, account, or session with a ROC or private identifier. The identifier database 232 can be stored in memory 206. The identifier database 232 can be stored separately and provided to content selector 230 (e.g., by a separate database server).

[0058] The content provider 102 and the partner device 104 can each include a decryptor 236. The decryptor 236 can be any application, applet, script, service, daemon, routine, or other executable logic to decrypt an eROC. The decryptor 236 can include software, hardware, or a combination of hardware and software for decrypting an eROC via a private key of the recipient of the eROC (e.g., a content provider 102 or partner device 104). The decryptor 236 can perform any suitable type of decryption algorithm and can include decryption circuitry such as an ASIC, a FPGA, a trusted platform module (TPM), or other such elements. The decryptor 236 can decrypt the eROC and extract the authentication token and other data from the decrypted ROC, such as the ROC.

[0059] FIG. 3 illustrates a flow diagram for a client device to self-enforce terms of service and filter content request. At step 300, a client device 100 can request an item of content. In some implementations, the request can be generated during rendering of a webpage, e.g., a request for embedded content in the page, such as a banner or other image, a media file such as a pre-roll, post-roll, or interstitial media file, or any other such content. In some implementations, the request can be for an unknown item of content, such that the content provider 102 or a partner device 104 can select an appropriate item of content. The request may, in some implementations, include a device identifier or other identifier of the client device, including a user name, account name, address or hardware identifier, or any other such information.

[0060] At step 302, the content provider 102 can request permission to share data with one or more selected partner devices 104. For example, responsive to receiving the request, the content provider 102 can select one or more partner devices 104 that can provide the content to the client device 100 or assist in providing the content to the client device 100. The selection of the partner devices 104 can be based on load balancing requirements, auction entrants, or any other such method. For example, the content provider 102 can communicate with partner devices 104 to

request bids or other offers for an opportunity to provide the content to client device 100. In some implementations, the partner devices 104 can provide additional information or data that can enable or improve the selection of content for the client device 100. For example, the partner devices 104 can provide preferences or indications of interests on which the content can be selected or customized. In some implementations, employing the partner devices 104 in the selection of content for the client device 100 can require that the content provider 102 share information about the client device 100 with the partner devices 104. Before sharing information with the partner devices 104, the content provider 102 can request permission from the client device 100 to share the information with the partner devices 104. In some implementations, the request for permission can be embedded in the application or webpage into which the content will be rendered. For example, the request for permission can be in the header of a webpage or the content slot of a webpage as a plurality of tags that identify each of the partner devices 104 that are involved in providing the content. In these implementations, the content provider 102 may not send an explicit transmission to the client device 100 requesting permission (e.g., step 302) and the client device's approval of data sharing request embedded in the webpage can be included in the request for content (e.g., step 300). For example, the request for content at step 300 can also include the below-described transmission of the ROC and eROC (e.g., step 306).

[0061] The request for permission can include a domain, address, or other identifier of each selected partner device 104. In some implementations, the identifiers can have a predetermined length, such as 32 or 64 bytes, with shorter domains or addresses padded and longer domains truncated. The identifiers can then be concatenated directly, as the client device 100 can extract each identifier according to the predetermined length. In other implementations, the identifiers can be separated by delimiters (e.g., predetermined characters or strings).

[0062] At step 304, if the client device 100 approves the request to share data, the client device 100 can generate an encrypted eROC for each of the approved partner devices 104. For example, as described herein, the client device 100 can generate an eROC for each of the approved partner devices 104. The eROCs can be generated based on an identifier associated with each of the partner devices 104 and can include an authentication token. The eROC can include a private identifier (e.g., the ROC) that identifies the client device 100 to the partner device 104. The

client device 100 can store, in the audit log, each authentication token in association with the partner device identification.

[0063] At step 306, the client device 100 can transmit the eROCs to the content provider 102. The client device 100 can transmit the eROCs to the content provider 102 in an HTTPs request. At step 308, the content provider 102 can forward a respective eROC to each of the respective partner devices 104. When forwarding the eROC to the partner devices 104, the transmission can include a request for content or a request for data to enable the selection of content.

[0064] At step 310, each partner device 104, upon receipt of the request and the eROC corresponding to the partner device 104, can decrypt the eROC using their private key to recover the ROC. Once the eROC is decrypted, the partner device 104 can extract the client device's private identifier and the authentication token. Based on the private identifier, the partner device 104 can select content for delivery to client device 100 or data that can be used in the selection of content. At step 312, the partner device 104 can transmit the data or content to the content provider 102.

[0065] At step 314, the content provider 102 can select content. The content provider 102 can select a content item from the plurality of content items provided by each of the different partner devices 104. The content provider 102 can select a content item based on preferences or other data provided by the partner devices 104. For example, a partner device 104 can provide an indication of the client device's computational capabilities, preferences of a user of the client device 100, or other data. In this example, the content provider 102 can select content that meets the user's preferences and can be rendered in accordance with the client device's computational capabilities.

[0066] As the content provider 102 receives content or data from the partner device 104, the content provider 102 can collect the authentication tokens that the partner devices 104 provide and the data transmitted from the partner devices 104. The content provider 102 can store the authentication tokens in a table in association with a content request identifier that corresponds to the current content request.

[0067] At step 316, the content provider 102 can transmit the content and collected authentication tokens to the client device 100. At step 318, the client device 100 can audit the returned authentication tokens and display the received content. The client device 100 can audit the authentication tokens prior to displaying or rendering the content. If the authentication tokens do not pass the audit, the client device 100 can refuse or decline to render the content. For example, the client device 100 can audit the authentication tokens to determine whether each of the returned authentication tokens are valid. A valid authentication token can indicate that the content provider 102 received permission to share data with a partner device 104. An invalid authentication token can indicate that the content provider 102 did not receive permission to share data with the partner device 104 and attempted, for example, to cache and reuse one or more eROCs or authentication tokens.

[0068] FIGS. 4 and 5 illustrate flow charts of a method for data exchange with self-enforcing permissions. Each flow chart is divided in part to show steps or procedures performed by the client device 100, content provider 102, and partner devices 104.

[0069] At step 400, the client device 100 can transmit a request for content to a content provider 102. An application on the client device 100, such as a web browser or media application, can generate the request for content. The application can generate the request during the rendering of a webpage or encountering a timed event during playback of another item of content. The request can be transmitted via any suitable transmission protocol (e.g., HTTPS request, RESTful request, etc.) and can include performing a handshaking procedure. The request can include a device identifier, account identifier, user identifier, session identifier, or other such identifier or a cookie generated by the client device (e.g., ROC) for the content provider 102 to uniquely identify the client device 100. For example, in some implementations, content provider 102 can have previously provided a cookie to client device 100 (e.g., during an authentication procedure or other such event), and client device 100 can provide the same cookie in the request thereafter.

[0070] At step 402, the content provider 102 can identify one or more partner devices 104 to potentially provide content to the client device 100 or to provide data for the selection of the content provided to the client device 100. The content provider 102 can select the partner devices

104 via any method, such as an auction system or round robin algorithm or other load balancing system. In some implementations, the partner devices 104 can be identified to the client device 100 in the webpage or application for which the content request was generated. For example, the list of partner devices 104 can be included in a plurality of tag parameters in a webpage or via a publicly-accessible partner file identified in the webpage.

[0071] At step 404, the content provider 102 can request permission to share data with the one or more partner devices 104 that were identified at step 402. The sharing of data with the partner devices 104 can include the content provider 102 sharing, for example, the above-described client identifier the content provider 102 received from the client device 100. In some implementations, each of the partner devices 104 may have previously exchanged a device identifier, account identifier, user identifier, session identifier, or other such identifier or a cookie with the client device 100. The previously shared identifier can be a ROC (or other identifier) that is unique between the client device 100 and respective partner device 104. For example, the client device 100 can store a different ROC for each of the partner devices 104. In some implementations, the request for permission to share data can include a request for the client device 100 to transmit the previously stored identifier to the partner device 104.

[0072] The permission request can include an identifier, such as a domain, that identifies the partner device 104 to the client device 100. For example, the content provider 102 can add the selected partner devices 104 to a list. Adding the partner device 104 can include adding a domain name or address of the partner device 104 to the list, concatenating the domain name or address to a set of names or addresses, or otherwise adding an identification of the partner device 104 to the list.

[0073] At step 406, the client device 100 can extract a first partner device 104 domain or address from the list that was received with the request to share data. As discussed above, the list can include fields of a predetermined length or variable length fields separated by delimiters. The client device 100 can extract a domain or address according to the length or delimiters, accordingly.

[0074] The client device 100 can determine whether permission should be given for data to be shared (by the content provider 102 or directly from the client device 100) with the partner device 104. As described above, the client device 100 can maintain an audit log that can include a list of whitelisted or blacklisted partner devices 104. The client device 100 can search the whitelisted or blacklisted partner devices 104 to determine whether data should be shared with the partner device 104. For example, if partner device's domain is included in a blacklist, the client device 100 can determine that data should not be shared with the partner device 104. The client device 100 can determine not to generate an eROC for the partner device 104 by returning to step 406 and identifying the next partner device 104 in the list provided by the content provider 102.

[0075] As described above, the audit log can include contribution information for each of the partner devices 104. The contribution information can indicate a partner device's contribution to the selection of previously provided content. For example, as described further in relation to step 428, when fulfilling a content request, the content provider 102 can provide an indication of the amount of contribution that each of the partner devices 104 provided in the selection of the content to fulfill the request. The level of contribution can be expressed, for each partner device 104, as a percentage. The summed percentage contribution of each of the partner devices 104 for fulfilling a content request can equal 100%. The level of contribution that a partner device 104 provides in fulfilling a content request can be determined based on an agreement between the content provider 102 and respective partner devices 104. For example, in exchange for data from a partner device 104 that enables the content provider 102 to customize the content provided to the client device 100, the content provider 102 can give the partner device 104 a 10% contribution. In some implementations, if the content provider 102 is provided a payment for providing the content, the payment can be split between each of the content provider 102 and partner devices 104 based on the level of contribution each played in selecting the content.

[0076] To determine the past contribution of the partner device 104, the client device 100 can calculate a contribution value for the partner device 104. The contribution value can be an average of the level of contribution the partner device 104 provided in selecting past content items. In some implementations, the contribution value can be calculated based on a time-limited

or specific number of content fulfillments. For example, the contribution value may be based only on the content fulfillments in the last day, week, month, year, or other time frame. In another example, the contribution value may be based on the past *N* number of content fulfillments. The contribution value can be a weighted average. For example, the level of contribution from the most recent content fulfillments can be weighted higher than the contribution level from relatively older content fulfillments.

[0077] At step 410, the client device 100 can determine whether the calculated contribution value is above a predetermined threshold. If the contribution value is above the threshold, the client device 100 can proceed to step 412. If the contribution value is below the threshold, the client device 100 can return to step 406 and not generate an eROC for the currently identified partner device 104. The client device 100 can compare the calculated contribution value to the threshold to determine what effect the partner device 104 has on the selection and customization of the content. To enable the partner device 104 to provide input in the selection of content for the client device 100 some information is provided from (or about) the client device 100 to the partner device 104. To increase the privacy of the client device 100, the client device 100 can decide to not authorize data sharing with partner devices 104 that do not substantially affect the selection of content (e.g., has a contribution value less than the threshold). By not sharing data with the partner device 104 when the partner device 104 has a low contribution value, the client device 100 can protect the privacy of the client device 100 while not affecting the quality of the content that is selected for the client device 100. In some implementations, if the partner device 104 has not participated in the fulfilment of a content request prior to the request transmitted at step 400, the client device 100 can pass the partner domain to the generation of a ROC (e.g., step 412) even though the partner domain's contribution value is below the predetermined threshold. The client device 100 can provide a new partner domain with a grace period where the partner domain is authorized to participate in data sharing even when the partner domain's contribution value is below the predetermined threshold. The grace period can last a predetermined length of time (e.g., a day, week, or month) or a predetermined interval (e.g., for 10, 50, or 100 content requests).

[0078] If the client device 100 determines the contribution value is above the threshold, the client device 100 can calculate a ROC for the partner device 104. In some implementations, in place of or in addition to comparing the contribution value to the threshold, the client device 100 can determine whether the domain of the partner device 104 is on a whitelist or blacklist that indicates data sharing authorizations. For example, the client device 100 can include a whitelist of partner devices 104 with which the client device 100 has given authorization to share data irrespective of the partner device's contribution value. In another example, the client device 100 can include a domain of a partner device 104 on a blacklist, and permission may never be provided to share data with the partner device 104 irrespective of the partner device's contribution. In another example, the client device 100 may consider only whether an identifier associated with the partner device 104 is on a whitelist or blacklist and may not calculate the relative contribution of the partner device 104 when determining whether to permit data sharing with the partner device 104.

[0079] At step 412, having approved the sharing of data with the partner device 104, the client device 100 can generate a ROC for the partner device 104. To generate the ROC, the client device 100 can calculate a one-way secure crypto hash of a domain or address or other identifier of the partner device 104 and a private key or identifier of the client device 100. The identifier of the client device 100 can be a private identifier that between the client device 100 and the partner device 104 can identify the client device 100 to the partner device 104. For example, the private identifier can be a cookie or other identifier that was previously provided by the partner device 104 to the client device 100, such as, for example, when the client device 100 accessed a webpage on the domain of the partner device 104. Any suitable crypto hashing algorithm can be used. In some implementations, the client device 100 can store the ROC in the audit log. For example, the client device 100 can store or cache the ROC for a predetermined length of time to reduce computing resources needed for the cryptographic hashing. When the ROC is locally cached at the client device 100, the client device 100 can search the cache prior to generating a new ROC at step 412.

[0080] At step 414, the client device 100 can generate an eROC from the ROC generated at step 412. The client device 100 can generate the eROC by retrieving a public key of the partner

device 104. The client device 100 can retrieve the key from a predetermined address (e.g., "[domain]/key.html") or from a key server. In some implementations, once the client device 100 receives the public key, the client device 100 can cache the key for a predetermined length of time for use in generating future eROCs for the partner device 104.

[0081] The client device 100 can generate the eROC by encrypting the ROC generated at step 412 with the partner device's public key. In some implementations, the client device 100 can concatenate the ROC with other data and encrypt the resulting character string to generate the eROC. For example, the client device 100 can concatenate the ROC, a timestamp, an authentication token, a public key, or any combination thereof to generate a character string.

[0082] The client device 100 can store the authentication token in an audit log. The authentication token can be stored in the audit log in association with the domain. In some implementations, the authentication token may be only valid for a predetermined length of time, and a timestamp or expiration time can be saved to the audit log in association with the authentication token to enable the client device 100 to determine whether the authentication token is valid when returned by a content provider 102 or partner device 104.

[0083] At step 416, the client device can determine if the list includes additional providers and, if so, can repeat steps 406–414.

[0084] After repeating steps 406–416 for each of the partner devices 104 in the list received from the content provider 102, the client device 100 exits the loop of steps 406–416 having generated an eROC for each of the partner devices 104 that the client device 100 approves of data being shared with. The client device 100 has also updated the audit log to include an indication of the authentication token generated for each of the partner devices 104.

[0085] At step 418, the client device 100 can generate a ROC for the content provider 102. The client device 100 can generate the ROC as described above in relation to the generation of a ROC for each of the partner devices 104. In some implementations, the client device 100 may not encrypt the ROC for the content provider 102, because the client device 100 will directly transmit the content provider's ROC to the content provider 102 in a secure communication

channel. While, as illustrated in FIG. 3, the partner devices 104 can receive their ROCs via the content provider 102 as the content provider 102 receives the ROCs (as eROCs) from the client device 100 and forwards ROCs (which can be encrypted as eROCS by the client device 100) to each of the respective partner devices 104. As the content provider's ROC will be directly transmitted to the content provider 102, the client device 100 can save computational resources by not encrypting the content provider's ROC. The client device 100 can transmit the ROC to the content provider 102 (and the eROC to the content provider 102) over a secure channel, such as via an HTTPS request. In some implementations, after step 418, the client device 100 can encrypt the content provider's ROC to generate an eROC that is transmitted to the content provider 102.

[0086] As step 420, the client device 100 can transmit the set of eROCs to the content provider 102. At step 422, the content provider 102 can forward the eROCs to each of the respective partner devices 104.

[0087] At step 500, as illustrated in FIG. 5, the partner device 104 can decrypt the eROC. The partner device 104 can decrypt the eROC using the partner device's private key. Decrypting the eROC generates the character string used to generate the eROC at step 414. Each of the values (e.g., the ROC, timestamp, authentication token, and public key) can be of a known, fixed length or separated by a delimiter. The partner device 104 can extract each of the ROC, timestamp, authentication token, and public key based on the fixed length of the values or the delimiter.

[0088] At step 502, the partner device 104 can select data based on the ROC. As described above, the ROC can be or can include a private identifier between the client device 100 and the partner device 104 that identifies the client device 100 to the partner device 104. The partner device 104 can select an item of content for delivery to the client device 100 based on the ROC. The content item can be selected via any suitable means and can be personalized or customized for the client device 100. The content can be personalized based on information associated with the ROC in the local database of the partner device. For example, the information associated with the ROC can include information previously received from the client device 100 (e.g., preferences), web browsing or search histories associated with the client device 100, previous content requests or interactions, or any other type and form of information. In some

implementations, in addition to or in place of selecting a content item, the partner device 104 can select information previously received from the client device 100 (e.g., preferences), web browsing or search histories associated with the client device 100, previous content requests or interactions, or any other type and form of information that is stored in association with the ROC in the internal database of the partner device 104. The partner device 104 can transmit the data to the content provider 102 or another partner device 104, which can select content based on the data.

[0089] At step 504, the partner device 104 can determine the authentication token. As described above, the authentication token can be included in the character string encrypted within the eROC. Once the eROC is decrypted to reveal the character string, the partner device 104 can extract the authentication token from the character string based on fixed lengths of the values within the character string and/or delimiters within the character string.

[0090] At step 506, the partner device 104 can transmit the authentication token and data, which can include a content item, to the content provider 102. In some implementations, when the data includes a content item, the data transmitted to the content provider 102 can include a bid for displaying or providing the content item to the client device 100.

[0091] Returning to FIG. 4, at step 424, the content provider 102 can collect the authentication tokens and relative contributions from each of the partner devices 104 in selecting the content item. For example, the content provider 102 can include an internal database of content items from which the content provider 102 selects the content for responding to the client device's content request. The content provider 102 may have received preferences of the client device 100 from each of four different partner devices 104. With the preference information, the content provider 102 can receive the authentication token from each of the partner devices 104 providing the preference information. The content provider 102 can collect each of the authentication tokens and store the authentication tokens in a table in association with the relative contribution of each of the partner devices 104. For example, the table for the four partner devices 104 can be an array, such as [(token1, 5), (token2, 5), (token3, 7), (token4, 2)], indicating that the partner device 104 associated with token1 contributed 5%, the partner device 104 associated with token2 contributed 5%, the partner device 104 associated with token3 contributed 7%, and the partner

device 104 associated with token4 contributed 2% to the selection of the content item. In this example, the content provider 102 may have been 81% (100%−5%−5%−7%−2%) responsible for the selection of the content item. The total contribution can be 100% such that the content provider 102 is responsible for the contribution not provided for by one of the partner devices 104. In some implementations, the content provider 102 can determine the contribution of each partner device 104 by searching a contribution table stored at the content provider 102. In some implementations, prior to a partner device 104 providing data to the content provider 102, the content provider 102 and the partner device 104 can agree on a contribution score the partner device 104 will receive when providing the content provider 102 data (or content) in the furtherance of content selection. The agreed upon contribution score can be stored in the contribution table.

[0092] At step 428, the content provider 102 can transmit the content item, authentication tokens, and relative contributions to the client device 100.

[0093] At step 430, the client device 100 can render the received content. In some implementations, the client device 100 can audit the returned authentication tokens before rendering the content. The client device 100 can use each of the returned authentication tokens as a key to look up the authentication token in the audit log. If the authentication token is expired, associated with a different content request, or not located in the audit log, the client device 100 can determine that the authentication token is invalid. For example, in violation of a ToS, the content provider 102 may have cached the eROC or authentication token of a partner device 104 such that the content provider 102 could share data with the partner device 104 without first asking permission of the client device 100. In this example, when the content provider 102 cached the authentication token, the client device 100 could determine that the authentication token has expired and is invalid. In some implementations, if one or more of the authentication tokens returned to the client device 100 at step 428 are invalid, the client device 100 can decide to not render the content.

[0094] At step 432, the client device 100 can update the audit log. Updating the audit log can include adding an entry for the just completed content request. The entry can include updating the overall or running contribution score for each of the partner devices 104 based on the

contribution score transmitted to the client device 100 at step 428. In some implementations, updating the audit log can include adding to a blacklist an identifier of the content provider 102 or partner device 104 if any of the respective content provider's or partner devices' authentication tokens were determined to be invalid.

[0095] According to at least one aspect of the disclosure, a system to control data exchange can include an encryptor. The encryptor can, for each partner domain in a list of partner domains, generate a partner domain-specific identifier and encrypt the partner domain-specific identifier and a first authentication token using an encryption key of the corresponding partner domain. The partner domain-specific identifier can be, or can include, an indication of the partner domain's domain (e.g.,. example.com) or a hash of thereof. The partner domain-specific identifier can be assigned to the partner domain by the system as a randomly generated, sequentially generated, or other type of identifier. The system can include a network interface to transmit to a content provider the encrypted partner domain-specific identifiers for each partner domain in the list of partner domains. The network interface can receive, from the content provider, a content item selected by the content provider based on data from each partner domain in the list of partner domains. The data from each partner domain in the list of partner domains is selected based on receiving the encrypted partner domain-specific identifier. For example, the partner domain can decrypt the encrypted partner domain-specific identifier and use the partner domain-specific identifier as a key in a lookup table to identify data associated with the partner domain-specific identifier. The network interface can receive, by the client device from the content provider, a second authentication token for each partner domain in the list of partner domains. The system can include an audit engine to decide to display the content item based on a comparison of the first authentication token for each partner domain in the list of partner domains and the second authentication token for each partner domain in the list of partner domains. For example, the audit engine can display the content item if the first and the second authentication token match.

[0096] In some implementations, the audit engine is configured to decide to display the content item based on the comparison which further comprises matching each of the first authentication token for each partner domain in the list of partner domains with one of the second

authentication tokens for each partner domain in the list of partner domains. The audit engine can also be configured to determine a contribution score for the second partner domain. The contribution score can indicate a level of contribution in selecting a second content item. The audit engine can decide to not generate a partner domain-specific identifier for the second partner domain based on the contribution score being less than a predetermined threshold. The contribution score can indicate a level of contribution in selecting one or more prior content items.

[0097] In some implementations, the encryptor can generate a first authentication token for each domain partner in a second list of partner domains. The network interface can receive, from the content provider, a second content item selected by the content provider based on data from each partner domain in the second list of partner domains. The network interface can also receive, from the content provider, a second authentication token for each domain partner in the second list of partner domains. The audit engine can decide to not display the second content item based on not matching the first authentication token for each domain partner in the second list of partner domains with a respective one of the second authentication token for each domain partner in the second list of partner domains.

[0098] The encryptor can be configured to generate the partner-specific identifier by calculating a cryptographic hash of a client device identifier and a domain name of the corresponding partner domain. The client device identifier can be unique for each of the partner domains in the list of partner domains. To encrypt the partner domain-specific identifier, the encryptor is configured to generate a character string by concatenating at least the partner domain-specific identifier, the first authentication token, and a timestamp. The encryptor can then encrypt the character string.

[0099] In some implementations, the encryptor can be configured to generate the partner domain-specific identifier based on the partner domain not being listed in a blacklist. The encryptor can be configured to generate the partner domain-specific identifier based on a contribution score of the partner domain being above a predetermined threshold.

[0100] According to at least one aspect of the disclosure, a method to control data exchange can include receiving, by a client device from a content provider, a list of partner domains. The method can include, for each partner domain in the list of partner domains, generating, by the client device, a partner domain-specific identifier and encrypting, by the client device, the partner domain-specific identifier and a first authentication token using an encryption key of the corresponding partner domain. The method can include transmitting, by the client device to the content provider, the encrypted partner domain-specific identifiers for each partner domain in the list of partner domains. The method can include receiving, by the client device from the content provider, a content item selected by the content provider based on data from each partner domain in the list of partner domains. The data from each partner domain in the list of partner domains is selected based on receiving the encrypted partner domain-specific identifier. The method can include receiving, by the client device from the content provider, a second authentication token for each partner domain in the list of partner domains. The method can include displaying, by the client device, the content item based on a comparison of the first authentication token for each partner domain in the list of partner domains and the second authentication token for each partner domain in the list of partner domains.

[0101] In some implementations, the method can include displaying the content item based on the comparison and further comprises matching each of the first authentication token for each partner domain in the list of partner domains with one of the second authentication tokens for each partner domain in the list of partner domains.

[0102] The method can include receiving, by the client device, a second partner domain. The method can include determining, by the client device, a contribution score for the second partner domain. The contribution score indicates a level of contribution in selecting a second content item. The method can include deciding, by the client device, to not generate a partner domain-specific identifier for the second partner domain based on the contribution score being less than a predetermined threshold. The contribution score can indicate a level of contribution in selecting one or more prior content items.

[0103] In some implementations, the method can include receiving, by the client device, a second list of partner domains. The method can include generating, by the client device, a first

authentication token for each domain partner in the second list of partner domains. The method can include receiving, by the client device from the content provider, a second content item selected by the content provider based on data from each partner domain in the second list of partner domains. The method can include receiving, by the client device from the content provider, a second authentication token for each domain partner in the second list of partner domains. The method can include deciding, by the client device, to not display the second content item based on not matching the first authentication token for each domain partner in the second list of partner domains with a respective one of the second authentication token for each domain partner in the second list of partner domains.

[0104] In some implementations, generating the partner-specific identifier can include calculating a cryptographic hash of a client device identifier and a domain name of the corresponding partner domain. The client device identifier can be unique for each of the partner domains in the list of partner domains. Encrypting the partner domain-specific identifier can include generating a character string by concatenating at least the partner domain-specific identifier, the first authentication token, and a timestamp. The method can include encrypting the character string. The method can include generating the partner domain-specific identifier based on the partner domain not being listed in a blacklist. The method can include generating the partner domain-specific identifier based on a contribution score of the partner domain being above a predetermined threshold.

[0105] The operations described in this specification can be implemented as operations performed by a data processing apparatus on data stored on one or more computer-readable storage devices or received from other sources.

[0106] The term "client" or "server" includes all kinds of apparatus, devices, and machines for processing data, such as a programmable processor, a computer, a system on a chip or multiple chips, or combinations of the foregoing. The apparatus can include special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). The apparatus can also include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, a

cross-platform runtime environment, a virtual machine, or a combination of one or more of them. The apparatus and execution environment can realize various different computing model infrastructures, such as web services, distributed computing, and grid computing infrastructures.

[0107] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages and declarative or procedural languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, object, or other unit suitable for use in a computing environment. A computer program may, but need not, correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub-programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0108] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform actions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatuses can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

[0109] Processors suitable for the execution of a computer program include both general and special purpose microprocessors and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for performing actions in accordance with instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic disks, magneto-optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a Global Positioning

System (GPS) receiver, or a portable storage device (e.g., a universal serial bus (USB) flash drive), to name just a few. Devices suitable for storing computer program instructions and data include all forms of non-volatile memory, media, and memory devices, including semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0110] To provide for interaction with a user, implementations of the subject matter described in this specification can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube), LCD (liquid crystal display), OLED (organic light emitting diode), TFT (thin-film transistor), plasma display, other flexible configuration, or any other monitor for displaying information to the user and a keyboard, a pointing device, e.g., a mouse, trackball, etc., or a touch screen, touch pad, etc., by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user or by sending webpages to a web browser on a user's client device in response to requests received from the web browser.

[0111] Implementations of the subject matter described in this specification can be implemented in a computing system that includes a back-end component, e.g., a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Communication networks can

include a local area network ("LAN"), a wide area network ("WAN"), an inter-network (e.g., the Internet), and peer-to-peer networks (e.g., ad hoc peer-to-peer networks).

[0112] For situations in which the systems discussed herein collect personal information about users, or can make use of personal information, the users can be provided with an opportunity to control whether programs or features can collect personal information (e.g., information about a user's social network, social actions, or activities, a user's preferences, or a user's location) or to control whether or how to receive content from a partner device or other data processing system that can be more relevant to the user. In addition, certain data can be anonymized in one or more ways before it is stored or used, so that personally identifiable information is removed when generating parameters. For example, a user's identity can be anonymized so that no personally identifiable information can be determined for the user, or a user's geographic location can be generalized where location information is obtained (such as to a city, postal code, or state level), so that a particular location of a user cannot be determined. Thus, the user can have control over how information is collected about him or her and used by the partner device.

[0113] While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any inventions or of what can be claimed, but rather as descriptions of features specific to particular implementations of particular inventions. Certain features that are described in this specification in the context of separate implementations can also be implemented in combination in a single implementation. Conversely, various features that are described in the context of a single implementation can also be implemented in multiple implementations separately or in any suitable subcombination. Moreover, although features can be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination can be directed to a subcombination or variation of a subcombination.

[0114] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing can be advantageous.

Moreover, the separation of various system components in the implementations described above should not be understood as requiring such separation in all implementations, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

[0115] Thus, particular implementations of the subject matter have been described. Other implementations are within the scope of the following claims. In some cases, the actions recited in the claims can be performed in a different order and still achieve desirable results. In addition, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking or parallel processing can be utilized.

CLAIMS

What is claimed:

1. A system to control data exchange, comprising:

an encryptor to, for each partner domain in a list of partner domains:

generate a partner domain-specific identifier; and

encrypt the partner domain-specific identifier and a first authentication token using an encryption key of the corresponding partner domain;

a network interface to:

transmit to a content provider the encrypted partner domain-specific identifiers for each partner domain in the list of partner domains;

receive, from the content provider, a content item selected by the content provider based on data from each partner domain in the list of partner domains, wherein the data from each partner domain in the list of partner domains is selected based on receiving the encrypted partner domain-specific identifier;

receive, from the content provider, a second authentication token for each partner domain in the list of partner domains; and

an audit engine to determine to display the content item based on a comparison of the first authentication token for each partner domain in the list of partner domains and the second authentication token for each partner domain in the list of partner domains.

2. The system of claim 1, wherein the audit engine determining to display the content item based on the comparison further comprises matching each of the first authentication token for each partner domain in the list of partner domains with one of the second authentication tokens for each partner domain in the list of partner domains.

3. The system of claim 1 or 2, further comprising the audit engine to:

determine a contribution score for a second partner domain, wherein the contribution score indicates a level of contribution in selecting a second content item; and

determine to not generate a partner domain-specific identifier for the second partner domain based on the contribution score being less than a predetermined threshold.

4. The system of claim 3, wherein the contribution score indicates a level of contribution in selecting one or more prior content items.

5. The system of any proceeding claim, further comprising:

the encryptor to generate a first authentication token for each domain partner in a second list of partner domains;

the network interface to:

receive, from the content provider, a second content item selected by the content provider based on data from each partner domain in the second list of partner domains;

receive, from the content provider, a second authentication token for each domain partner in the second list of partner domains; and

the audit engine to determine to not display the second content item based on not matching the first authentication token for each domain partner in the second list of partner domains with a respective one of the second authentication token for each domain partner in the second list of partner domains.

6. The system of any proceeding claim, wherein the encryptor is configured to generate the partner-specific identifier by calculating a cryptographic hash of a client device identifier and a domain name of the corresponding partner domain.

7. The system of claim 6, wherein the client device identifier is unique for each of the partner domains in the list of partner domains.

8. The system of any proceeding claim, wherein to encrypt the partner domain-specific identifier, the encryptor is configured to:

generate a character string by concatenating at least the partner domain-specific identifier, the first authentication token, and a timestamp; and

encrypt the character string.

9. The system of any proceeding claim, wherein the encryptor is configured to generate the partner domain-specific identifier based on the partner domain not being listed in a blacklist.

10. The system of any proceeding claim, wherein the encryptor is configured to generate the partner domain-specific identifier based on a contribution score of the partner domain being above a predetermined threshold.

11. A method to control data exchange, comprising:

receiving, by a client device from a content provider, a list of partner domains;

for each partner domain in the list of partner domains:

generating, by the client device, a partner domain-specific identifier; and

encrypting, by the client device, the partner domain-specific identifier; and a first authentication token using an encryption key of the corresponding partner domain;

transmitting, by the client device to the content provider, the encrypted partner domain-specific identifiers for each partner domain in the list of partner domains;

receiving, by the client device from the content provider, a content item selected by the content provider based on data from each partner domain in the list of partner domains, wherein the data from each partner domain in the list of partner domains is selected based on receiving the encrypted partner domain-specific identifier;

receiving, by the client device from the content provider, a second authentication token for each partner domain in the list of partner domains; and

displaying, by the client device, the content item based an authentication of the second authentication token for each partner domain in the list of partner domains.

12. The method of claim 11, wherein displaying the content item based on the comparison further comprises matching each of the first authentication token for each partner domain in the list of partner domains with one of the second authentication tokens for each partner domain in the list of partner domains.

13. The method of claim 11 or 12, further comprising:

receiving, by the client device, a second partner domain;

determining, by the client device, a contribution score for the second partner domain, wherein the contribution score indicates a level of contribution in selecting a second content item; and

determining, by the client device, to not generate a partner domain-specific identifier for the second partner domain based on the contribution score being less than a predetermined threshold.

14. The method of claim 13, wherein the contribution score indicates a level of contribution in selecting one or more prior content items.

15. The method of any one of claims 11 to 14, further comprising:

receiving, by the client device, a second list of partner domains;

generating, by the client device, a first authentication token for each domain partner in the second list of partner domains;

receiving, by the client device from the content provider, a second content item selected by the content provider based on data from each partner domain in the second list of partner domains;

receiving, by the client device from the content provider, a second authentication token for each domain partner in the second list of partner domains; and

determining, by the client device, to not display the second content item based on not matching the first authentication token for each domain partner in the second list of partner domains with a respective one of the second authentication token for each domain partner in the second list of partner domains.

16. The method of any one of claims 11 to 15, wherein generating the partner-specific identifier comprises calculating a cryptographic hash of a client device identifier and a domain name of the corresponding partner domain.

17. The method of claim 16, wherein the client device identifier is unique for each of the partner domains in the list of partner domains.

18. The method of any one of claims 11 to 17, wherein encrypting the partner domain-specific identifier comprises:

generating a character string by concatenating at least the partner domain-specific identifier, the first authentication token, and a timestamp; and

encrypting the character string.

19. The method of any one of claims 11 to 18, further comprising generating the partner domain-specific identifier based on the partner domain not being listed in a blacklist.

20. The method of any one of claims 11 to 19, further comprising generating the partner domain-specific identifier based on a contribution score of the partner domain being above a predetermined threshold.

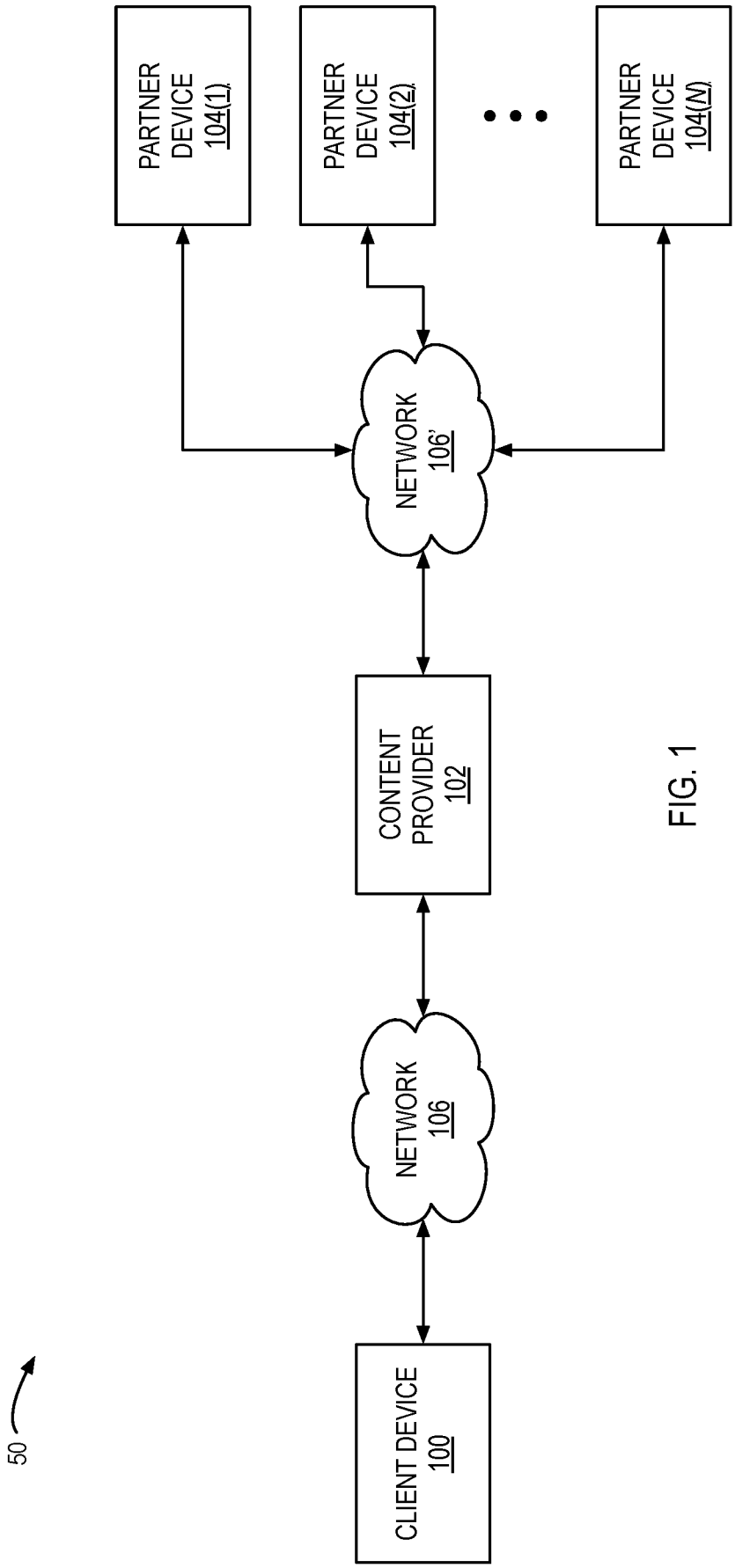

PARTNER DEVICE 104(1)
PARTNER DEVICE 104(2)
PARTNER DEVICE 104(N)
NETWORK 106'
CONTENT PROVIDER 102
NETWORK 106
CLIENT DEVICE 100
50

FIG. 1

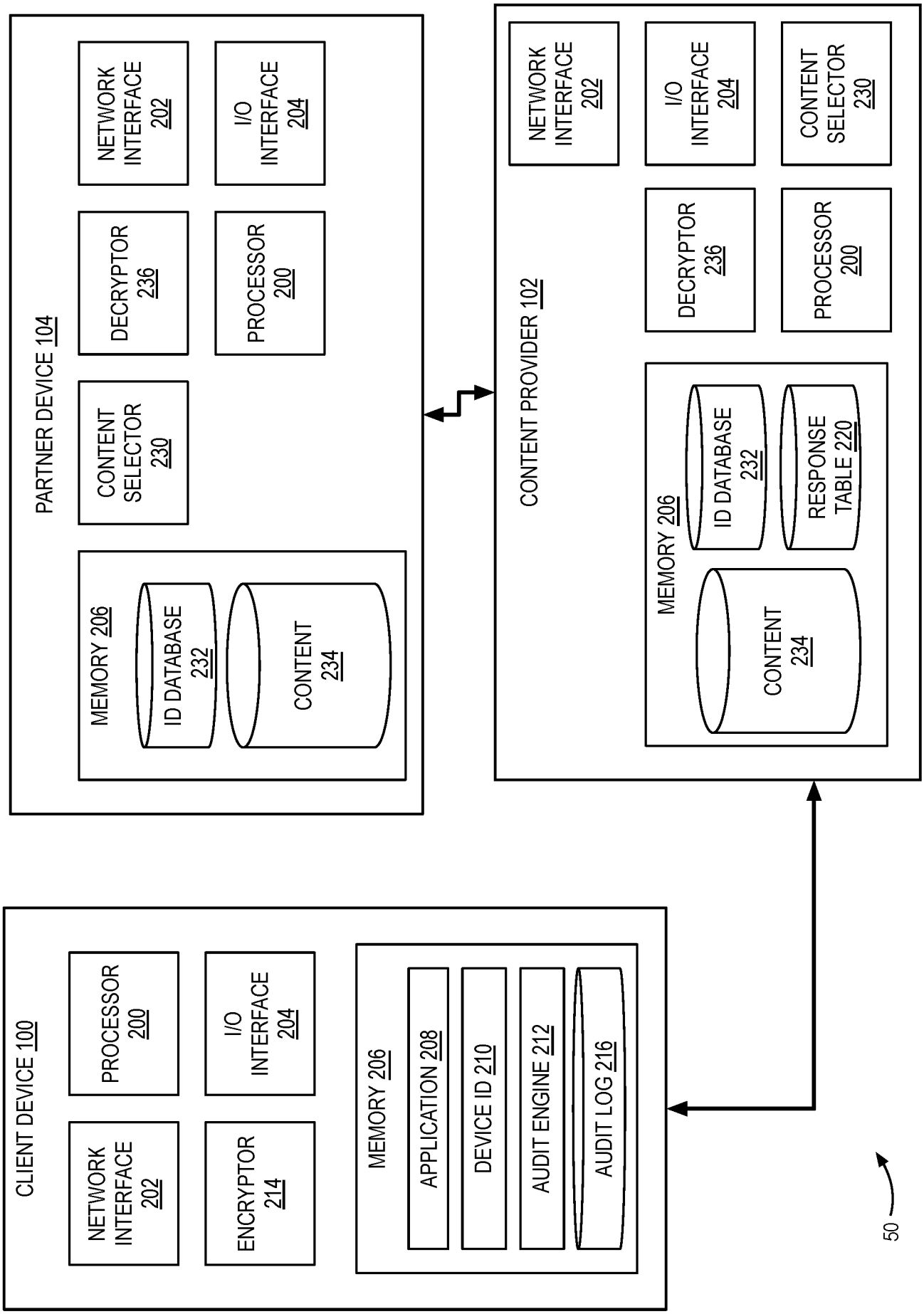

CLIENT DEVICE 100
NETWORK INTERFACE 202
PROCESSOR 200
ENCRYPTOR 214
I/O INTERFACE 204
MEMORY 206
APPLICATION 208
DEVICE ID 210
AUDIT ENGINE 212
AUDIT LOG 216
50
PARTNER DEVICE 104
NETWORK INTERFACE 202
I/O INTERFACE 204
DECRYPTOR 236
PROCESSOR 200
CONTENT SELECTOR 230
MEMORY 206
ID DATABASE 232
CONTENT 234
CONTENT PROVIDER 102
NETWORK INTERFACE 202
I/O INTERFACE 204
CONTENT SELECTOR 230
DECRYPTOR 236
PROCESSOR 200
MEMORY 206
ID DATABASE 232
RESPONSE TABLE 220
CONTENT 234

FIG. 2

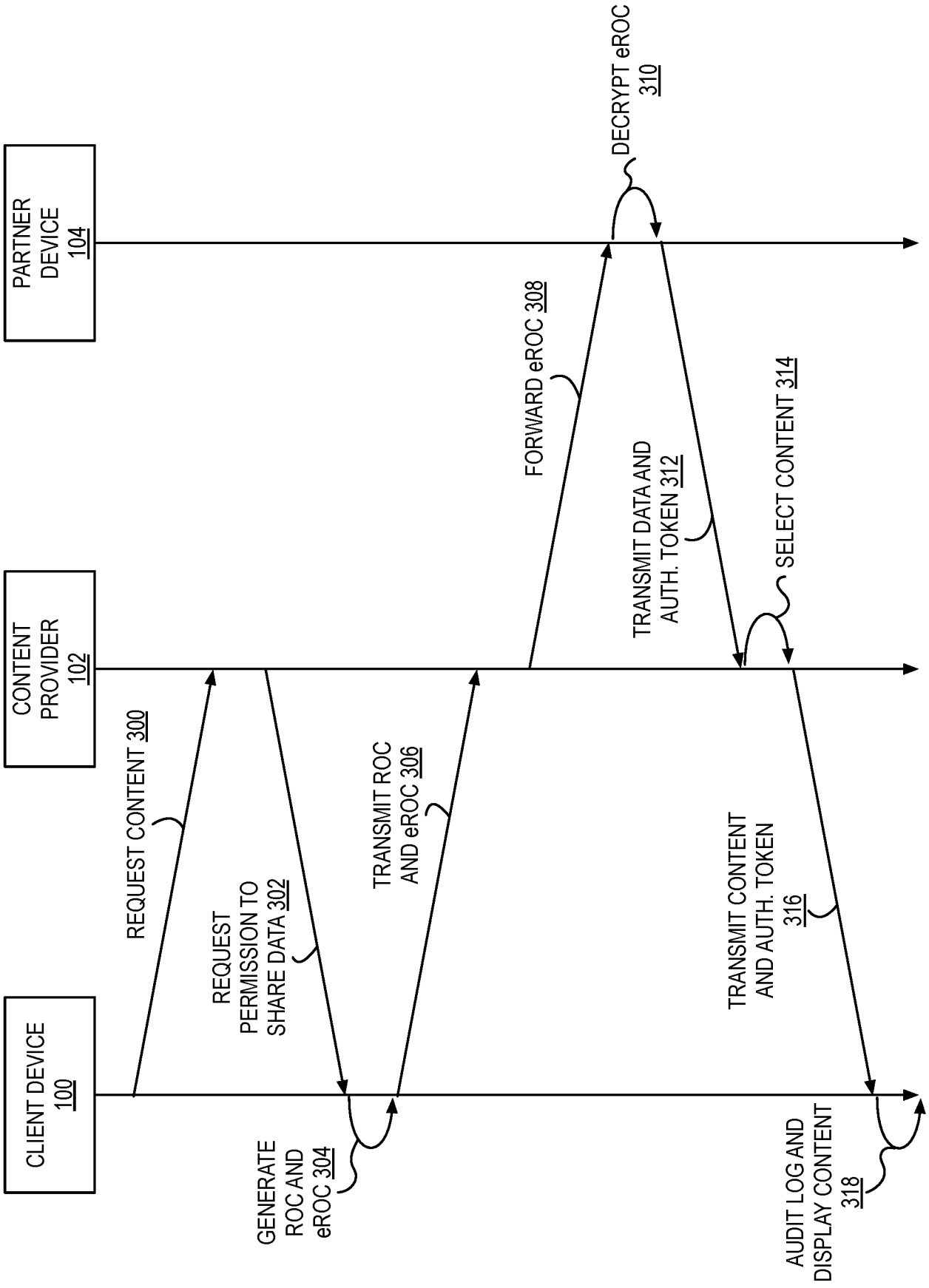

CLIENT DEVICE 100
CONTENT PROVIDER 102
PARTNER DEVICE 104
REQUEST CONTENT 300
REQUEST PERMISSION TO SHARE DATA 302
GENERATE ROC AND eROC 304
TRANSMIT ROC AND eROC 306
FORWARD eROC 308
DECRYPT eROC 310
TRANSMIT DATA AND AUTH. TOKEN 312
SELECT CONTENT 314
TRANSMIT CONTENT AND AUTH. TOKEN 316
AUDIT LOG AND DISPLAY CONTENT 318

FIG. 3

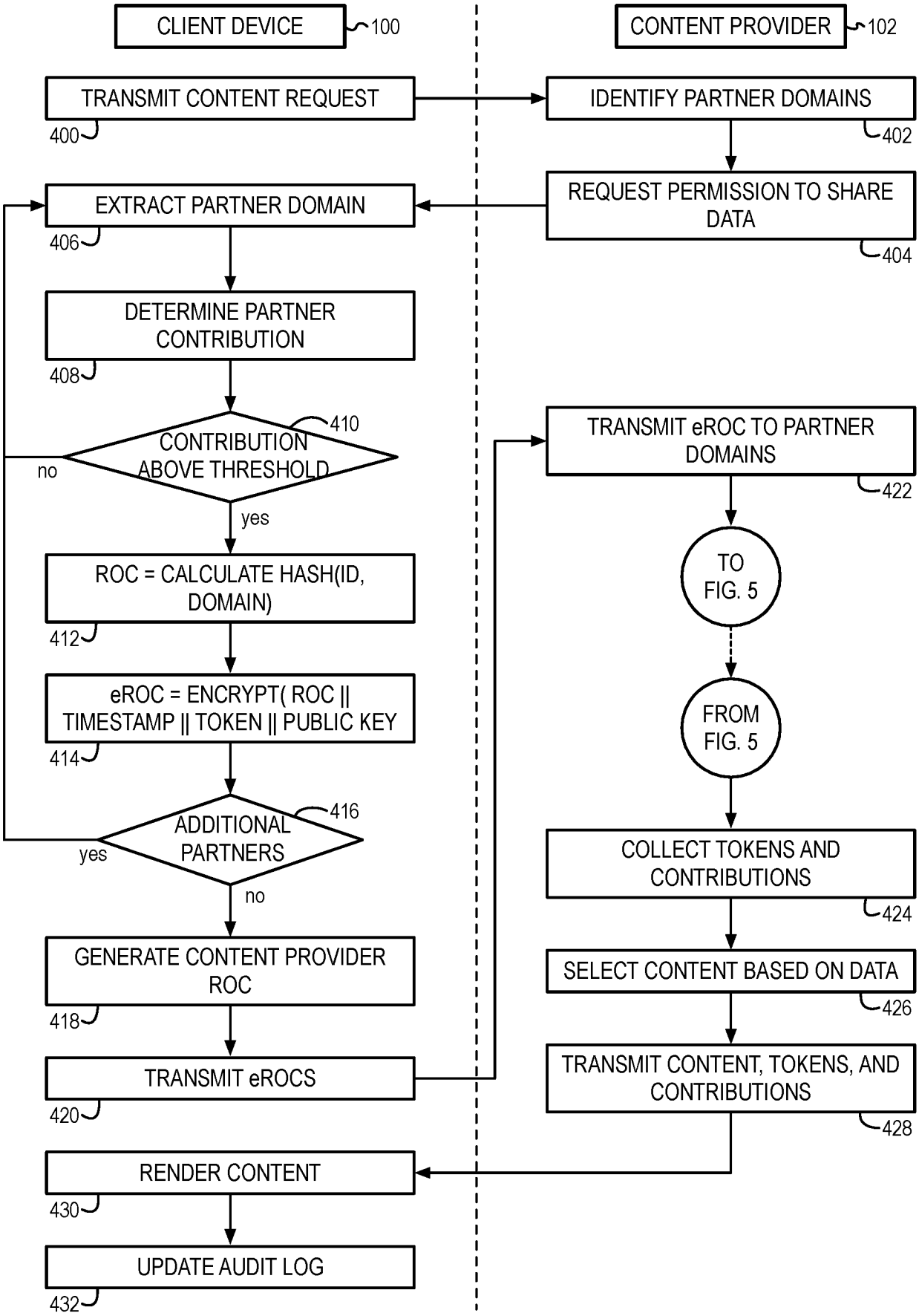

CLIENT DEVICE
100
CONTENT PROVIDER
102
TRANSMIT CONTENT REQUEST
400
IDENTIFY PARTNER DOMAINS
402
REQUEST PERMISSION TO SHARE DATA
404
EXTRACT PARTNER DOMAIN
406
DETERMINE PARTNER CONTRIBUTION
408
CONTRIBUTION ABOVE THRESHOLD
410
no
yes
ROC = CALCULATE HASH(ID, DOMAIN)
412
eROC = ENCRYPT(ROC || TIMESTAMP || TOKEN || PUBLIC KEY
414
ADDITIONAL PARTNERS
416
yes
no
GENERATE CONTENT PROVIDER ROC
418
TRANSMIT eROCS
420
RENDER CONTENT
430
UPDATE AUDIT LOG
432
TRANSMIT eROC TO PARTNER DOMAINS
422
TO FIG. 5
FROM FIG. 5
COLLECT TOKENS AND CONTRIBUTIONS
424
SELECT CONTENT BASED ON DATA
426
TRANSMIT CONTENT, TOKENS, AND CONTRIBUTIONS
428

FIG. 4

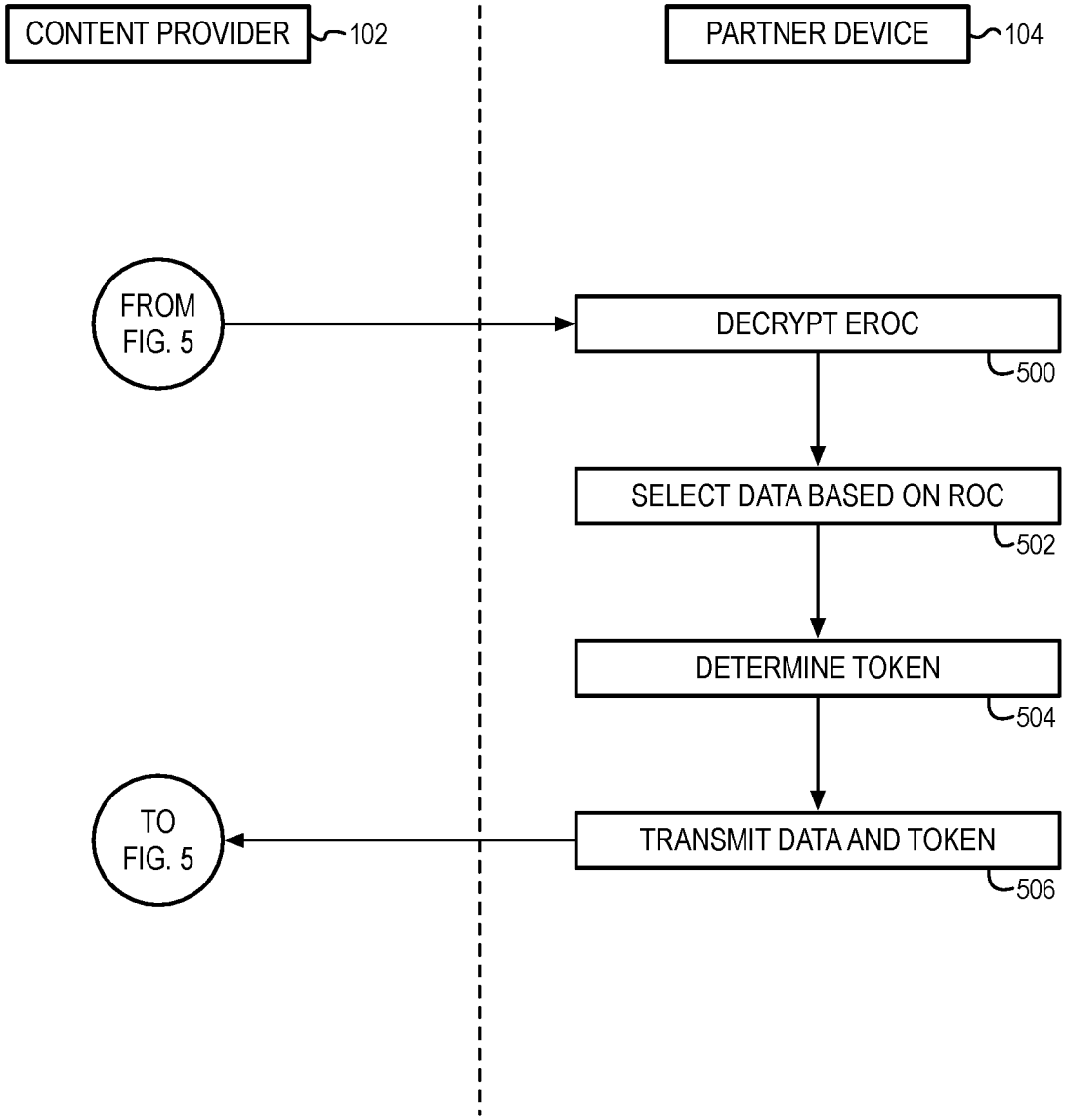

CONTENT PROVIDER
102
PARTNER DEVICE
104
FROM FIG. 5
DECRYPT EROC
500
SELECT DATA BASED ON ROC
502
DETERMINE TOKEN
504
TO FIG. 5
TRANSMIT DATA AND TOKEN
506

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/027737

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06
ADD. H04L29/08 G06Q30/02

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L G06Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2015/339734 A1 (WILSON J T) 26 November 2015 (2015-11-26) paragraph [0007] - paragraph [0012] paragraph [0028] - paragraph [0040] figures 1,2 -----	1-20
A	US 2011/055552 A1 (FRANCIS PAUL [DE] ET AL) 3 March 2011 (2011-03-03) paragraph [0002] - paragraph [0009] -----	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
7 January 2020	24/01/2020

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Ströbeck, Anders

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2019/027737

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015339734 A1	26-11-2015	US 2015339734 A1	26-11-2015
		US 2018322541 A1	08-11-2018
		WO 2015179424 A1	26-11-2015
US 2011055552 A1	03-03-2011	EP 2474124 A2	11-07-2012
		JP 2013504123 A	04-02-2013
		US 2011055552 A1	03-03-2011
		WO 2011028669 A2	10-03-2011