United States Patent 12395336
Kind Code B2
Date of Patent August 19, 2025
Inventor(s) Rao; Vanishree et al.

Content management systems and methods using proxy reencryption

Abstract

This disclosure relates to systems and methods for managing protected electronic content using proxy reencryption techniques. Rights management architectures are described that may, among other things, provide end-to-end protection of content keys from their point of origination at a content creator and/or content service to end user devices. Proxy reencryption techniques consistent with aspects of the disclosed embodiments may enable transformation of a ciphertext under one public key to a ciphertext containing the same plaintext under another public key. Consistent with embodiments disclosed herein, proxy reencryption processes may be implemented using indistinguishability obfuscation and puncturable public-key encryption schemes, functional encryption, and/or white box obfuscation techniques.

Inventors: **Rao; Vanishree (San Mateo, CA), Tarjan; Robert (Princeton, NJ), Maher; David P. (Philadelphia, PA)**

Applicant: **Intertrust Technologies Corporation** (Berkeley, CA)

Family ID: **1000008763493**

Assignee: **Intertrust Technologies Corporation (Berkeley, CA)**

Appl. No.: **18/664196**

Filed: **May 14, 2024**

Prior Publication Data

Document Identifier	Publication Date
US 20240313964 A1	Sep. 19, 2024

Related U.S. Application Data

continuation parent-doc US 17829241 20220531 US 12021984 child-doc US 18664196
continuation parent-doc US 16421002 20190523 US 11362824 child-doc US 17829241
us-provisional-application US 62676429 20180525

Publication Classification

Int. Cl.: **H04L9/30** (20060101); **H04L9/06** (20060101)

U.S. Cl.:

CPC **H04L9/30** (20130101); **H04L9/0618** (20130101); H04L2209/16 (20130101); H04L2209/603 (20130101); H04L2209/76 (20130101)

Field of Classification Search

CPC: H04L (9/30); H04L (9/0618); H04L (2209/16); H04L (2209/603); H04L (2209/76); H04L (9/14); H04L (9/0825)

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
1007615	12/1910	Tatum	N/A	N/A
6963974	12/2004	Skinner	N/A	N/A
7010808	12/2005	Leung	713/193	G06F 21/10
7170999	12/2006	Kessler	N/A	N/A
7500269	12/2008	Huotari	N/A	N/A
7792300	12/2009	Caronni	380/263	H04L 9/0825
7822207	12/2009	Douguet	713/192	H04L 9/004
8266448	12/2011	Shi	N/A	N/A
8296583	12/2011	Sparks	N/A	N/A
8479018	12/2012	Futa	N/A	N/A
8556247	12/2012	Nagel	N/A	N/A
8751800	12/2013	Dorwin	713/160	H04L 65/613
8806187	12/2013	Vemula	713/150	H04L 67/306
8855317	12/2013	Rong et al.	N/A	N/A
8954740	12/2014	Moscaritolo	380/278	H04L 63/065
9094191	12/2014	Avanzi et al.	N/A	N/A
9374373	12/2015	Chan	N/A	N/A
9806887	12/2016	Campagna	N/A	H04L 9/0822
9819487	12/2016	Fujii	N/A	H04L 9/3073
10277563	12/2018	Rao et al.	N/A	N/A
10460774	12/2018	Lee	N/A	N/A
10461943	12/2018	Norum	N/A	N/A
10778657	12/2019	Ding et al.	N/A	N/A
RE48313	12/2019	Sparks et al.	N/A	N/A
10826685	12/2019	Campagna	N/A	N/A
11062042	12/2020	McKervey et al.	N/A	N/A
11095620	12/2020	Sirota et al.	N/A	N/A
2002/0099663	12/2001	Yoshino	705/65	G06Q 30/06
2002/0133396	12/2001	Barnhart	N/A	N/A
2002/0184154	12/2001	Hori	N/A	N/A
2003/0110130	12/2002	Pelletier	N/A	N/A
2003/0120611	12/2002	Yoshino et al.	N/A	N/A
2003/0123670	12/2002	Shimada	N/A	N/A
2003/0126430	12/2002	Shimada	713/155	H04L 63/0464
2003/0126457	12/2002	Kohiyama	713/193	G06F 21/10
2003/0135464	12/2002	Mourad et al.	N/A	N/A
2004/0001594	12/2003	Krishnaswamy	N/A	N/A
2004/0111631	12/2003	Kocher et al.	N/A	N/A
2004/0133794	12/2003	Kocher	713/193	G06F 21/62
2005/0069138	12/2004	de Jong	380/278	G06F 9/3017
2005/0071280	12/2004	Irwin	N/A	N/A
2005/0235361	12/2004	Alkove	N/A	N/A
2006/0004662	12/2005	Nadalin	705/50	H04L 63/0823
2006/0004803	12/2005	Aschen	N/A	N/A

2006/0080732	12/2005	Ohkubo	726/9	G07F 7/1008
2006/0085352	12/2005	Hug	N/A	N/A
2006/0089912	12/2005	Spagna et al.	N/A	N/A
2007/0100768	12/2006	Boccon-Gibod	N/A	N/A
2007/0140479	12/2006	Wang	380/30	H04L 9/30
2007/0185814	12/2006	Boccon-Gibod	N/A	N/A
2007/0294170	12/2006	Vantalon	N/A	N/A
2008/0005024	12/2007	Kirkwood	N/A	N/A
2008/0092239	12/2007	Sitrick et al.	N/A	N/A
2008/0092240	12/2007	Sitrick et al.	N/A	N/A
2008/0148067	12/2007	Sitrick et al.	N/A	N/A
2008/0170701	12/2007	Matsuo	380/45	H04L 9/083
2009/0013177	12/2008	Lee	N/A	N/A
2009/0016537	12/2008	Ju et al.	N/A	N/A
2009/0199287	12/2008	Vantalon	N/A	N/A
2009/0210697	12/2008	Chen	N/A	N/A
2009/0252327	12/2008	Ciet	380/277	H04L 9/002
2010/0058485	12/2009	Gonzalez	N/A	N/A
2010/0138671	12/2009	Kim	N/A	N/A
2011/0047371	12/2010	Timby	713/168	G06F 21/33
2011/0067012	12/2010	Eisen	713/189	G06F 8/51
2011/0110525	12/2010	Gentry	N/A	N/A
2011/0145562	12/2010	Mangalore	N/A	N/A
2011/0150213	12/2010	Michiels	380/28	H04L 9/30
2012/0201380	12/2011	Kohiyama	N/A	N/A
2012/0239942	12/2011	Yan	713/189	H04L 63/0421
2012/0275597	12/2011	Knox	380/210	H04N 21/6334
2012/0278608	12/2011	Kohiyama	N/A	N/A
2012/0284522	12/2011	Lewis	N/A	N/A
2012/0284804	12/2011	Lindquist	726/29	G06F 21/10
2012/0290843	12/2011	Belenky	713/168	G06F 21/10
2012/0331283	12/2011	Chandran	713/150	H04L 9/088
2013/0086393	12/2012	Pogmore	N/A	N/A
2013/0156188	12/2012	Xu	380/255	H04L 9/3073
2013/0212388	12/2012	D'Souza	N/A	N/A
2013/0283392	12/2012	Mirashrafi	N/A	N/A
2013/0318347	12/2012	Moffat	N/A	N/A
2014/0040622	12/2013	Kendall	713/171	H04W 12/041
2014/0050318	12/2013	Hayashi	380/46	H04L 9/3073
2014/0089202	12/2013	Bond	713/166	H04L 9/14
2014/0095890	12/2013	Mangalore et al.	N/A	N/A
2014/0098890	12/2013	Mangalore	N/A	N/A
2014/0108786	12/2013	Kreft	713/194	G06Q 20/3825
2014/0140504	12/2013	Karroumi et al.	N/A	N/A
2014/0164776	12/2013	Hook	713/171	G06F 21/6218
2014/0208097	12/2013	Brandwine	713/164	G06F 21/33
2014/0208100	12/2013	Kendall	N/A	N/A
2014/0237614	12/2013	Irvine	N/A	N/A
2014/0281545	12/2013	Erofeev	713/171	G06F 11/1402
2014/0314233	12/2013	Evans	N/A	N/A
2014/0348323	12/2013	Chevallier-Mames	380/28	H04L 9/0631
2015/0026452	12/2014	Roelse	N/A	N/A
2015/0033020	12/2014	Madden	N/A	N/A
2015/0043735	12/2014	Fujii et al.	N/A	N/A
2015/0180661	12/2014	Fujii	380/46	H04L 9/30
2015/0195258	12/2014	Kohlyama	N/A	N/A
2015/0200917	12/2014	Fujii	713/171	G06F 21/602

2015/0229471	12/2014	Nair	N/A	N/A
2015/0235011	12/2014	Swaminathan	N/A	N/A
2015/0270964	12/2014	Yasuda	713/171	H04L 9/0825
2016/0063219	12/2015	Vlot	713/168	H04L 63/0464
2016/0092871	12/2015	Gordon	705/44	H04W 4/06
2016/0119292	12/2015	Kaseda	713/165	H04L 63/045
2016/0241389	12/2015	Le Saint et al.	N/A	N/A
2016/0277367	12/2015	Fischer	N/A	H04N 21/6582
2016/0330022	12/2015	Ito	N/A	H04L 9/0825
2016/0352711	12/2015	Kohiyama	N/A	H04L 63/0457
2016/0380767	12/2015	Hayashi	380/45	H04L 9/14
2017/0006025	12/2016	Liu	N/A	N/A
2017/0116393	12/2016	Choi	N/A	G09C 1/00
2017/0155628	12/2016	Rohloff	N/A	H04L 63/02
2017/0163429	12/2016	Stuntebeck	N/A	H04L 63/0823
2017/0228525	12/2016	Wajs	N/A	H04L 9/3226
2017/0236123	12/2016	Ali et al.	N/A	N/A
2017/0237551	12/2016	Van Foreest	713/189	G06F 21/10
2017/0323114	12/2016	Egorov	N/A	N/A
2017/0373828	12/2016	Michiels	N/A	N/A
2018/0219678	12/2017	Medvinsky	N/A	N/A
2018/0314827	12/2017	Wells	N/A	N/A
2018/0349577	12/2017	Goldwasser et al.	N/A	N/A
2019/0014094	12/2018	Le Saint	N/A	H04L 63/06
2019/0044940	12/2018	Khalil	N/A	N/A
2019/0188703	12/2018	Murray	N/A	N/A
2019/0205558	12/2018	Gonzales, Jr.	N/A	N/A
2019/0222878	12/2018	Cocchi	N/A	N/A
2019/0258778	12/2018	Park	N/A	H04L 9/0819
2019/0297063	12/2018	De Gaspari	N/A	N/A
2019/0334708	12/2018	Carpov et al.	N/A	N/A
2019/0363883	12/2018	Rao et al.	N/A	N/A
2020/0242039	12/2019	Shani et al.	N/A	N/A
2020/0382328	12/2019	Bhattacharya	N/A	N/A
2021/0089676	12/2020	Ford et al.	N/A	N/A
2021/0226785	12/2020	Notani	N/A	N/A
2022/0222590	12/2021	Wang	N/A	G06Q 30/0206
2023/0306089	12/2022	Park	N/A	N/A

FOREIGN PATENT DOCUMENTS

Patent No.	Application Date	Country	CPC
2016161134	12/2015	WO	N/A

OTHER PUBLICATIONS

Susan Hohenberger, Guy N. Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely Obfuscating Re-encryption. In Salil P. Vadhan, editor, Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, Feb. 21-24, 2007, Proceedings, vol. 4392 of Lecture Notes in Computer Science, pp. 233-252. Springer, 2007. (20 pgs). cited by applicant

PKCS1: RSA cryptography standard, Version 2.0. RSA Laboratories, 1998. (43 pgs). cited by applicant

Shashank Agrawal and David J. Wu. Functional encryption: Deterministic to randomized functions from simple assumptions. Jean-Sebastien Coron and Jesper Buus Nielsen, editors, Advances in Cryptology—EUROCRYPT 2017—36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, Apr. 30-May 4, 2017, Proceedings, Part II, vol. 10211 of Lecture Notes in Computer Science, 2017. (45 pgs). cited by applicant

Non-Final Office Action issued in U.S. Appl. No. 16/421,002. Mar. 18, 2021. (6 pgs). cited by applicant

Final Office Action issued in U.S. Appl. No. 16/421,002. Sep. 15, 2021. (6 pgs). cited by applicant

Notice of Allowance issued in U.S. Appl. No. 16/421,002. Feb. 15, 2022. (7 pgs). cited by applicant

Non-Final Office Action dated Oct. 25, 2023, issued in U.S. Appl. No. 17/829,241. cited by applicant
Notice of Allowance dated Feb. 14, 2024, issued in U.S. Appl. No. 17/829,241. cited by applicant

Primary Examiner: Cervetti; David Garcia

Attorney, Agent or Firm: Thayne and Davis LLC

Background/Summary

RELATED APPLICATION (1) This application is a continuation of U.S. patent application Ser. No. 17/829,241, filed May 31, 2022, and entitled “CONTENT MANAGEMENT SYSTEMS AND METHODS USING PROXY REENCRYPTION,” which is a continuation of U.S. patent application Ser. No. 16/421,002, filed May 23, 2019, and entitled “CONTENT MANAGEMENT SYSTEMS AND METHODS USING PROXY REENCRPYTION,” which claims the benefit of priority under 35 U.S.C. § 119(e) to U.S. Provisional Application No. 62/676,429, filed May 25, 2018, and entitled “SYSTEMS AND METHODS FOR MANAGING ELECTRONIC CONTENT USING PROXY RE-ENCRYPTION,” all of which are hereby incorporated by reference in their entirety.

COPYRIGHT AUTHORIZATION
(1) Portions of the disclosure of this patent document may contain material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the U.S. Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.
SUMMARY
(2) The present disclosure relates generally to systems and methods for managing electronic content. More specifically, the present disclosure relates to systems and methods for managing protected electronic content using proxy reencryption techniques.
(3) In certain digital rights management (“DRM”) protocols, content keys may be revealed in the clear to a DRM service. This may be undesirable, as it introduces a potential attack surface. Moreover, content creators, content owners, and/or content service providers may be relatively protective of the distribution of their content keys, and therefore may be less willing to provide these keys to other parties and/or services, including DRM services.
(4) Consistent with embodiments disclosed herein, DRM protocols are described that, in some implementations, may provide end-to-end protection of content keys from their point of origination (e.g., a content creator and/or content service provider) to user devices. In some embodiments, content key ciphertexts communicated to devices may remain encrypted (e.g., encrypted in the RSA v1.5 and/or RSA-OAEP format). Certain embodiments may further provide for message protocols where fewer messages are sent in connection with a DRM license request process, thereby reducing latency associated with such processes.
(5) Various embodiments of the disclosed systems and methods may use a cryptographic functionality that may be referred to in certain instances herein as proxy reencryption (“PRE”). In certain embodiments, PRE may enable transformation of a ciphertext under one public key to a ciphertext containing the same plaintext under another public key. Embodiments of the disclosed PRE implementations may use receiver ciphertext in the RSA-OAEP encryption format, although other suitable encryption formats are also contemplated. Consistent with embodiments disclosed herein, PRE may be implemented using indistinguishability obfuscation (“iO”) and puncturable public-key encryption schemes, functional encryption (“FE”), and/or white box obfuscation techniques.
(6) In some embodiments, a simulation-based security model may be used. In further embodiments, functionalities of the various underlying methods may be randomized. In certain embodiments, the disclosed methods may receive as input a content key and output a randomized RSA-OAEP encryption of the content key under a device's public key. In some embodiments, the FE scheme may not necessarily to hide all and/or some of the functionalities of the underlying cryptographic methods.

Description

BRIEF DESCRIPTION OF THE DRAWINGS
(1) The inventive body of work will be readily understood by referring to the following detailed description in conjunction with the accompanying drawings, in which:

(2) FIG. **1** illustrates an example of an interaction between a content service, a DRM service, and a user device consistent with certain embodiments of the present disclosure.
(3) FIG. **2** illustrates an example of an interaction between a content service, a DRM service, and a user device employing a reencryption process consistent with certain embodiments of the present disclosure.
(4) FIG. **3** illustrates an example of a program that may decrypt a content service's ciphertext and perform an encryption operation under the public key of a user device consistent with certain embodiments of the present disclosure.
(5) FIG. **4** illustrates an example of a reencryption program consistent with certain embodiments of the present disclosure.
(6) FIG. **5** illustrates an example of an obfuscated reencryption program consistent with certain embodiments of the present disclosure.
(7) FIG. **6** illustrates a diagram of a representation of an example of a commutative encryption scheme consistent with certain embodiments of the present disclosure.
(8) FIG. **7** illustrates an example of a method for managing protected content consistent with certain embodiments of the present disclosure.
(9) FIG. **8** illustrates an example of a system that may be used to implement certain embodiments of the systems and methods of the present disclosure.

DETAILED DESCRIPTION

(10) A detailed description of the systems and methods consistent with embodiments of the present disclosure is provided below. While several embodiments are described, it should be understood that the disclosure is not limited to any one embodiment, but instead encompasses numerous alternatives, modifications, and equivalents. In addition, while numerous specific details are set forth in the following description in order to provide a thorough understanding of the embodiments disclosed herein, some embodiments can be practiced without some or all of these details. Moreover, for the purpose of clarity, certain technical material that is known in the related art has not been described in detail in order to avoid unnecessarily obscuring the disclosure.
(11) The embodiments of the disclosure may be understood by reference to the drawings, wherein in certain instances, but not necessarily all instances, like parts may be designated by like numerals or descriptions. The components of the disclosed embodiments, as generally described and illustrated in the figures herein, could be arranged and designed in a wide variety of different configurations. Thus, the following detailed description of the embodiments of the systems and methods of the disclosure is not intended to limit the scope of the disclosure but is merely representative of possible embodiments of the disclosure. In addition, the steps of any method and/or process disclosed herein do not necessarily need to be executed in any specific order, or even sequentially, nor need the steps be executed only once, unless otherwise specified.
(12) Consistent with certain embodiments of the disclosed systems and methods, PRE techniques may be used to, among other things, provide for end-to-end protection of content keys from their point of origination (e.g., a content creator and/or content service provider) to user devices. As detailed below, in certain embodiments, PRE methods consistent with aspects of the disclosed systems and methods may enable transformation of a ciphertext under one public key to a ciphertext containing the same plaintext under another public key.
(13) Content License Provisioning Processes
(14) FIG. **1** illustrates an example of an interaction between a content service **100**, a DRM service **102**, and a user device **104** consistent with certain embodiments of the present disclosure. In certain embodiments, a DRM license request and/or provisioning process may involve interactions between the content service **100**, the DRM service **102**, and/or the user device **104**. Although embodiments disclosed herein are discussed in connection with interactions between a content service **100**, a DRM service **102**, and/or a user device **104**, it will be appreciated that embodiments of the disclosed systems and processes may be implemented using a variety of other devices, systems, and/or services, and/or involve interactions between other devices, systems, and/or services, including intermediate devices, systems, and/or services. In addition, although illustrated as single systems and/or services, it will be appreciated that, in certain embodiments, the DRM service **102** and/or the content service **100** may be implemented using a plurality of systems operating to deliver content license provisioning services.
(15) The content service **100**, DRM service **102**, user device **104**, and/or one or more other systems and/or services (not shown) may comprise any suitable computing system or combination of systems configured to implement embodiments of the systems and methods disclosed herein. In certain embodiments, the content service **100**, DRM service **102**, user device **104** and/or one or more other systems and/or services may comprise at least one processor system configured to execute instructions stored on an associated non-transitory computer-readable storage medium. As discussed in more detail below, the content service **100**, DRM service **102**, user device **104**, and/or one or more other systems and/or services and/or other service providers may further

comprise a secure processing unit ("SPU") configured to perform sensitive operations such as trusted credential, license, and/or key management, secure policy management, cryptographic operations, and/or other aspects of the systems and methods disclosed herein. The content service **100**, DRM service **102**, user device **104**, and/or one or more other systems and/or services may further comprise software and/or hardware configured to enable electronic communication of information between the devices and/or services via one or more associated network connections.

(16) The content service **100**, DRM service **102**, user device **104**, and/or one or more other systems and/or services may comprise a computing device executing one or more applications configured to implement embodiments of the systems and methods disclosed herein. In certain embodiments, the user device **104** may comprise at least one of a smartphone, a smartwatch, a laptop computer system, a desktop computer system, a display, a gaming system, an entertainment system, a streaming media system, a wearable health monitoring device, a tablet computer, a smart home device, a digital assistant device, a connected appliance, and/or any other computing system and/or device that may be used in connection with the disclosed systems and methods. In certain embodiments, the user device **104** may comprise software and/or hardware configured to request and receive content and/or content licenses from a content service **100**, DRM service **102**, and/or another system or service, and/or to use received content licenses to decrypt and/or otherwise enable access to, rendering of, and/or use of protected content. The content service **100** may comprise a service and/or system associated with a content creator and/or generator, a content distributor, and/or any other content-related system and/or service.

(17) The content service **100**, DRM service **102**, user device **104**, and/or one or more other systems and/or services may communicate using a network comprising any suitable number of networks and/or network connections. The network connections may comprise a variety of network communication devices and/or channels and may use any suitable communication protocols and/or standards facilitating communication between the connected devices and systems. For example, in some embodiments, the network may comprise the Internet, a local area network, a virtual private network, and/or any other communication network utilizing one or more electronic communication technologies and/or standards (e.g., Ethernet and/or the like). In some embodiments, the network connections may comprise a wireless carrier system such as a personal communications system ("PCS"), and/or any other suitable communication system incorporating any suitable communication standards and/or protocols. In further embodiments, the network connections may comprise an analog mobile communications network and/or a digital mobile communications network utilizing, for example, code division multiple access ("CDMA"), Global System for Mobile Communications or Groupe Special Mobile ("GSM"), frequency division multiple access ("FDMA"), and/or time divisional multiple access ("TDMA") standards. In certain embodiments, the network connections may incorporate one or more satellite communication links. In yet further embodiments, the network connections may use IEEE's 802.11 standards, Bluetooth®, ultra-wide band ("UWB"), Zigbee®, and or any other suitable communication protocol(s).

(18) A license and/or content key request and/or provisioning process consistent with aspects of the disclosed embodiments may result in a DRM license and/or a content key, ck, being communicated to an authorized device that has requested and/or otherwise wishes to access, use, and/or render content. In certain embodiments, the content key may be included in the DRM license and communicated with the license and/or may be communicated separately from the DRM license.

(19) As illustrated, the user device **106** may provide the DRM service **102** with a content request message **106**. The content request message **106** may comprise information identifying protected content that the user device **104** would like to access, use, and/or otherwise render, information relating to a type requested content access, and/or the like. In some embodiments, the content request message **106** may be associated with content that the user device **104** has downloaded and/or otherwise stored and would like to access, use, and/or render. In further embodiments, the content request message **106** may be associated with content that the user device **104** would like to download and/or otherwise store for later use. In yet further embodiments, the content request message **106** may be associated with content that the user would like to stream from a service (e.g., the content service **100** and/or another associated service provider).

(20) The content service **100** may provide a DRM service **102** with a content key, ck, associated with the content identified in the content request message **106** and/or associated constraints **108**. In some embodiments, the content key and/or associated constraints **108** may be protected during transmission between the content service **100** and/or the DRM service **102** (e.g., using suitable cryptographic encryption and/or other secure communication techniques). In certain embodiments, the constraints may articulate one or more requirements and/or parameters that the DRM service **102** may use in connection with license generation processes.

(21) The DRM service **102** may generate a token **110** based, at least in part, on the content key and the constraints **108** received from the content service **100**. In some embodiments, the token **110** may comprise the content key provided by the content service **100**, ck, encrypted with a symmetric key of the DRM service,

k.sub.e **120**. The token **110** may be communicated from the DRM service **102** to the content service **100**.
(22) In response to the content request message **106**, the content service **100** may return to the user device **104** a message **112** that includes the token provided by the DRM service **102** and/or information that identifies (e.g., uniquely identifies) the associated content.
(23) The user device **104** may be associated with a public key secret-key pair, **114** pk.sub.d, sk.sub.d. To obtain a license and/or an associated content key, ck, from the DRM service **102**, the user device **104** may communicate a license request message **116** to the DRM service **102**. The license request message **116** may comprise the information identifying the associated content, the token included in the message **112** received from the content service **100**, and/or the public key, pk.sub.d, of the user device **104**.
(24) The DRM service **102** may determine whether the user device **104** is authorized to access, use, and/or otherwise render the protected content associated with the license request message **116**. If the user device **104** is authorized, the DRM service **102** may issue a license **118** to the user device **104**. In certain embodiments, the license **118** may comprise an encrypted copy of the content key, ck. For example, the license **118** may comprise the content key, ck, encrypted with the public key, pk.sub.d, of the user device **104**. The license **118** may further comprise various license terms and/or other associated license information that may be enforced by the user device **104** in connection with the access, use, and/or rendering of the protected content. In certain embodiments, the encrypted content key may be communicated separately from other license information included in the license **118**.
(25) In the interaction illustrated in FIG. **1**, the DRM service **102** may have access to content keys in the process of relating the content keys from the content service **100** to the user device **104**. Accordingly, the content keys in the illustrated interaction may not necessarily be end-to-end protected. In addition, the number of messages exchanged from the time a user device **104** issues a playback request (e.g., by transmitting a content request **106** to the content service **100**) to the time when the license **118** and/or content key is received by the user device **104** may introduce higher latency.
(26) In some embodiments, devices may expect ciphertexts in the licenses in a certain format. For example, devices may expect ciphertexts to be encrypted under the RSA-OAEP encryption scheme-a randomized public-key encryptions scheme combining the RSA algorithm with the Optimal Asymmetric Encryption Padding ("OAEP") method. In various embodiments, it may be easier to modify protocols at the DRM service side compared to end user devices. Accordingly, certain embodiments of the disclosed protocol may maintain that ciphertexts received by devices are in an RSA-OAEP format.
(27) Proxy Reencryption Overview
(28) Consistent with certain embodiments of the disclosed systems and methods, a reencryption scheme is described that may allow for conversion of a ciphertext under one public key to a ciphertext (e.g., a ciphertext of the same plaintext) under a different public key. In some embodiments, reencryption may proceed without exposing and/or decrypting the ciphertext outside protected processes. In various embodiments, a reencryption scheme may generate and/or use a special key, which may be referred to in certain instances herein as a reencryption key.
(29) In certain embodiments, the reencryption key may be generated based on a function of a function of a "senders" decryption key and a "receivers" encryption key that converts ciphertexts under the sender's public key to ciphertexts under the receiver's public key. In instances herein, a reencryption key may be denoted as rk.sub.c.fwdarw.d with the sender's public key pk.sub.c and the receivers public key pk.sub.d. In various embodiments, the reencryption key may comprise and/or be included in a protected reencryption program configured to perform reencryption operations consistent with embodiments disclosed herein.
(30) Content License Protocol Using Proxy Reencryption
(31) FIG. **2** illustrates an example of an interaction between a content service **100**, a DRM service **102**, and a user device **104** employing a reencryption process consistent with certain embodiments of the present disclosure. The content service **100** may be associated with a public key secret-key pair **200** pk.sub.c, sk.sub.c. The content service **100** and/or another associated service and/or system may maintain a database **200** of information relating to one or more registered devices including, for example, the user device **104**. In certain embodiments, the device information database **200** may be stored and/or otherwise maintained and/or managed directly by the content service **100**. In other embodiments, the device information database **200** may be stored, maintained, and/or managed by a different system and/or service and accessed by the content service **100**.
(32) The device information database **200** may include a variety of information relating to registered devices including, for example, public keys associated with registered devices. For example, the device information database **200** may include the public key pk.sub.d of user device **104**.
(33) The content service **100** may generate a corresponding reencryption key rk.sub.c.fwdarw.d for the user device **104**. In some embodiments, the content service **100** may generate and/or store reencryption keys for

multiple registered devices (e.g., devices having associated information included in the device information database **200**). In certain embodiments, computed reencryption keys may be stored, managed, and/or otherwise maintained in the device information database **200**.
(34) In some embodiments, the generated reencryption key may comprise a reencryption program. Consistent with various embodiments disclosed herein, the generated reencryption key and/or reencryption program may be used to transform an encryption of a content key under the public key pk.sub.c of the content service **100** to a randomized encryption (e.g., RSA-OAEP encryption) of the content key under the public key of the pk.sub.d user device.
(35) The content service **100** may generate a ciphertext ct.sub.c of the content key ck associated with a content item by encrypting the content key using its public key pk.sub.c:ct.sub.c=Enc(pk.sub.c, ck). Applicable registered device information, a generated reencryption key, ciphertext of the encrypted content key ct.sub.c, and/or content identification associated with the corresponding content (i.e., a content ID) may be communicated from the content service **100** to the DRM service **102** via message **202**.
(36) Information communicated from the content service **100** to the DRM service **102** may be maintained in a database **204**. For example, as illustrated, reencryption keys and/or programs and/or ciphertext of encrypted content keys may be stored, managed, and/or otherwise maintained by the DRM service **102** in a database **204**. In certain embodiments, the database **204** may be stored and/or otherwise maintained and/or managed directly by the DRM service **102**. In other embodiments, the database **204** may be stored, maintained, and/or managed by a different system and/or service and accessed by the DRM service **102**.
(37) As discussed in more detail below, when a user device **104** with the public key pk.sub.d makes an authorized request for content, the DRM service **102** may reencrypt the ciphertext containing the corresponding content key and may provide the resulting ciphertext (e.g., RSA ciphertext) and/or the rest of the license to the user device **104**. For example, when the user device **104** requests playback for content, the user device **104** may send a content request message α **206** to the content service **100** that may include various parameters associated with the user device **104** and/or the associated content request. For example, the content request message α **206** may comprise an identification of a requested content item (i.e., a content ID) and/or the public key of the user device pk.sub.d. In some embodiments, the content request message **206** may comprise information identifying protected content that the user device **104** would like to access, use, and/or otherwise render, information relating to a type requested content access, and/or the like. For example, the content request message **206** may comprise information identifying protected content that the user device **104** has downloaded and/or intends to download and/or stream and render on the user device **104**.
(38) The content service **100** may determine whether the user device **104** that sent the content request message α **206** is authorized to access the requested content item. In some embodiments, determining whether the user device **104** is authorized to access the requested content item may be based on information included in the content request message α **206** (e.g., device identification information, the device's public key, and/or the like). If the request is authorized, the content service may sign the content request message α **206** and return a response message σ **208** to the user device **104**. In some embodiments, the signature may be generated using a private key sk.sub.c associated with the content service **100**, although other suitable signature keys may also be used.
(39) If the request is not authorized, the content service **100** may abort and/or otherwise terminate the protocol. In some embodiments, a message may be communicated to the user device **104** by the content service **100** explicitly denying the content request. In further embodiments, the content service **100** may simply not respond to the content request message α **206** if the content request is denied.
(40) Upon receipt of the response message σ **208** from the content service **100**, the user device **104** may communicate a license request message **210** to the DRM service **102**. In some embodiments, the license request message **210** may comprise an identification of the requested content item (e.g., a content ID), the public key pk.sub.d of the user device **104**, and/or the signed response message σ **208**.
(41) The DRM service **102** may verify the signature of the signed response message σ **208** included in the license request message **210** to confirm it was signed by the content service **100**. If the signature is not verified, the DRM service **102** may abort and/or otherwise terminate the protocol. For example, in some embodiments, a message may be communicated to the user device **104** by the DRM service **102** explicitly denying the license request. In further embodiments, the DRM service **102** may simply not respond to the license request message **210** if the license request is denied.
(42) If the signature is verified, the DRM service **102** may use a secure reencryption program **214** executing thereon that may comprise the reencryption key rk.sub.c.fwdarw.d to reencrypt the ciphertext ct.sub.c=Enc(pk.sub.c, ck) under the public key pk.sub.d of the user device **104** and generate reencrypted ciphertext ct.sub.d: ct.sub.d ← ReEnc(rk.sub.c.fwdarw.d, ct.sub.c). In certain embodiments, the reencrypted ciphertext ct.sub.d may comprise an RSA-OAEP ciphertext including the content key ck associated with the

content ID identified in the license request message **210**: RSA(pk.sub.d, ck). In certain embodiments, the reencryption program **214** and/or key may allow the DRM service **102** to only reencrypt the content key. The content key ck may not be revealed in the clear to the DRM service **102** during this process, thus achieving end-to-end protection of the content key. For example, in some embodiments, the reencryption program **214** and/or its operations during a reencryption process may be obfuscated and/or otherwise employ the use of secure software execution methods such that the plaintext of the content key is not revealed to the DRM service **102** and/or revealed outside the secure execution environment of the reencryption program **214**.
(43) The DRM service **102** may generate a license **212** that includes the reencrypted ciphertext ct.sub.d. The license **212** may further comprise various license terms and/or other associated license information that may be enforced by the user device **104** in connection with the accessing, use, and/or rendering of the content item. In certain embodiments, the reencrypted ciphertext ct.sub.d may be communicated separately from other license information included in the license **212**.
(44) The user device **104** may decrypt the reencrypted ciphertext ct.sub.d received in the license **212** using its corresponding secret key, sk.sub.d, and may allow access, use, and/or rendering of the content in accordance with any applicable terms included in the license **212**.
(45) Instantiation of a Proxy Reencryption Scheme
(46) Consistent with embodiments disclosed herein, a PRE scheme may be constructed with receiver ciphertexts in the RSA-OAEP format. In some embodiments, iO and FE schemes may be used to protect the integrity of secret information during the reencryption process. An FE scheme may, in certain embodiments, comprise an encryption scheme where each secret key is associated with a function and decryption with that secret key provides a function of the plaintext (as compared to the plaintext itself like in a usual encryption scheme). In some embodiments, whitebox cryptographic obfuscation and/or other suitable software obfuscation techniques may be used to protect the integrity of secret information (e.g., plaintext content keys) during a reencryption process).
(47) Proxy Reencryption Instantiation Using Indistinguishability Obfuscation
(48) In some embodiments, a PRE scheme with RSA-OAEP receiver ciphertext format may be instantiated by obfuscating with iO a program that first decrypts the sender's ciphertexts and then encrypts the resulting plaintext with the receiver's public key. FIG. **3** illustrates an example of a program **300** that may decrypt a content service's ciphertext and perform an encryption operation under the public key of a user device pk.sub.d consistent with certain embodiments of the present disclosure. As illustrated, the program may use as constants a sender's secret key, the receiver's public key, and/or a pseudorandom seed value. These constants and/or various aspects of the program **300** and/or its operation during execution may be obfuscated and/or be protected (e.g., using iO and/or other suitable obfuscation and/or protection techniques) such that secret information used by and/or operated on by the program **300** may not be readily revealed to a system executing the program **300** (e.g., a DRM system). The program **300** may further receive as an input ciphertext ct.sub.c received from the sender.
(49) As illustrated, the program **300** may decrypt the ciphertext ct.sub.c using the senders secret key sk.sub.c to obtain plaintext m. A pseudorandom string r may be generated using the pseudorandom seed. Ciphertext ct.sub.d may be generated as an RSA-OAEP encryption of m using the receiver public key pk.sub.d and the generated pseudorandom string r. The program **300** may output the ciphertext ct.sub.d encrypted under the receiver public key pk.sub.d.
(50) FIG. **4** illustrates another example of a reencryption program **400** consistent with certain embodiments of the present disclosure. In some embodiments, ζ=(KeyGen, Enc, Dec, Puncture.sub.ζ, PDec) may be a puncturable public-key encryption scheme, RSA=(RSAKeyGen, RSAEnc, RSADec) may be the RSA-OAEP encryption scheme, and PRF may be a puncturable pseudorandom function. Various aspects of the disclosed embodiments may include one or more of: Key Generation: The sender's keys may be generated using ζ's KeyGen. The receiver's keys may be generated using RSA key generation algorithm. Encryption: The sender's encryption may be performed using the encryption algorithm of ζ and that of the receiver using RSA encryption. Decryption: The sender's decryption may be performed using the decryption algorithm of ζ and that of the receiver using RSA decryption. Reencryption Key Generation: For a receiver's public key pk.sub.d and sender's secret key sk.sub.c, the reencryption key may iO(Prog.sub.pkd), where Prog.sub.pkd is illustrated in FIG. **4** as program **400** (an example of a program the decrypts content service ciphertexts and reencrypt under pk.sub.d) and K←Key.sub.F.
(51) FIG. **5** illustrates an example of an obfuscated reencryption program **500** consistent with certain embodiments of the present disclosure. The obfuscated reencryption program **500** may, in certain instances herein, be referred to as Prog.sub.pk.sub.d.sup.(s′) where C*←Enc(pk.sub.c, ck*) and ck* comprises a random content key, sk.sub.c[C*]←Puncture.sub.ζ(sk.sub.c, C*), K[C*]=Puncture.sub.F(K, C*), r*←$, and {tilde over (C)}*←RSAEnc(pk.sub.d, ck*; r*).

(52) Proxy Reencryption Instantiation Using Functional Encryption
(53) In certain embodiments, a PRE scheme may be instantiated with an FE scheme where the functions associated with secret keys may receive a plaintext and output an RSA-OAEP encryption of the plaintext under the receiver's public key. In certain embodiments, rFE may be a functional encryption scheme for a randomized function family F={F.sub.λ}.sub.λεN defined as follows: The input space may be the content key space; the output space may be the ciphertext space of RSA encryption with the content key space as the plaintext space. Considering $f \in$F.sub.λ, f may be associated with an RSA public key pk corresponding to security parameter λ. On input ck, f may compute RSAEnc(pk, ck) as the output.
(54) Various aspects of the disclosed embodiments may include one or more of: Key Generation: The sender's keys may be generated using FE's KeyGen. The receiver's keys may be generated using RSA key generation algorithm. Encryption: The sender's encryption may be performed using the encryption algorithm of FE and that of the receiver using RSA encryption. Decryption: The sender's decryption may be performed using the decryption algorithm of FE and that of the receiver using RSA decryption. Reencryption Key Generation: For a receiver's public key pk.sub.d, sk.sub.fd ← KeyGen(msk, f.sub.d) may be computed, where f.sub.d corresponds to the public key pk.sub.d. Reencryption: Computed as {tilde over (C)} ← Dec(sk.sub.fd, C).
Proxy Reencryption Notations
(55) In certain instances herein, λ may denote a security parameter. If two distributions D.sub.1, D.sub.2 are statistically relatively close, then this may be denoted by D.sub.1≡D.sub.2. s ← S may denote randomly sampling an element s from a set S. A bit string s may be sampled uniformly at random, where the length may be implicit, by s ← $. In certain instances herein, by default, algorithms may receive the security parameter 1.sup.λ as an input, although in some instances this may not be explicitly specified. Probabilistically polynomial time may be denoted as “PPT”. For n∈N, [n] may be written to denote the set of integers {1, . . . , n}. An interactive Turning Machine may be denoted as A with n rounds by A.sub.1, . . . , A.sub.n which share states. R may be a randomized function; to distinguish between its inputs and randomness, an invocation may be denoted as R(x.sub.1, x.sub.2, . . . ; r), where x.sub.1, x.sub.2, . . . are the inputs and r is the randomness.
(56) Negligible Function Definitions
(57) In various embodiments, a function negl may be negligible if ∀.sub.c∈N, ∃n.sub.0∈N, such that ∀.sub.n≥n.sub.0, negl(n)<n.sup.−c. A negligible function may be denoted by negl. A reencryption scheme consistent with embodiments disclosed herein may allow conversion of a ciphertext under one public key to a ciphertext (of the same plaintext) under a different public key. As discussed above, a reencryption scheme may provide a special key, that may be referred to herein as a reencryption key, that may be a function of ‘sender's’ decryption key and ‘receiver's’ encryption key, that may convert ciphertexts under the sender's public to ciphertexts under the receiver's public key.
(58) Proxy Reencryption Definitions
(59) In some embodiments, a proxy reencryption scheme may comprise a tuple of (that may be probabilistic) polynomial time algorithms (KeyGen, Enc, Dec, RKeyGen, ReEnc), where the components may be defined as follows: (KeyGen, Enc, Dec) may be the standard key generation, encryption, and decryption algorithms for the underlying cryptosystem. On input the security parameter 1.sup.λ, KeyGen may output a key pair (pk, sk). On input pk and message m, the output of Enc may be a ciphertext ct. On input sk and ciphertext c, the output of Dec may be the message m. On input sk.sub.b, pk.sub.a, the reencryption key generation algorithm, RKeyGen, may output the reencryption key rk.sub.a.fwdarw.b for the proxy. On input rk.sub.a.fwdarw.b and ciphertext ct.sub.a, the reencryption function, ReEnc, may output ct.sub.b.
Simulation-Based Security of Proxy Reencryption Definitions
(60) For PPT adversaries A, there may exist a simulator S=(S.sub.1, S.sub.2, S.sub.3, S.sub.4, S.sub.5) such that the following holds. S/may generate the simulated sender's public key; S.sub.2, S.sub.4 generate the simulated reencryption keys before and after the adversary receives the set of ciphertexts on the plaintexts of its choice encrypted under the sender's public key; and S.sub.3 may generate the simulated reencryption values.
(61) Indistinguishability Obfuscation Definitions
(62) Consistent with various aspects of the disclosed embodiments, an indistinguishability obfuscator may transform any two programs that compute the same functionality into indistinguishable programs that also compute the same functionality. The security provided by an indistinguishability obfuscator may be that, for any two circuits that have the same input-output functionality, their obfuscations are computationally indistinguishable. For example, if f is a functionality and C.sub.0 and C.sub.1 are circuits corresponding to JAVA and Python implementations respectfully of f, then the obfuscations of these circuits may be indistinguishable.
(63) In some embodiments, a uniform PPT machine iO may be called an indistinguishability obfuscator for circuits if the following conditions are satisfied: For security parameters λ∈N, for circuits C, for inputs x, for

every C′←iO(C): C′(x)=C(x). For any (not necessarily uniform) PPT adversaries Samp, D, there may exist a negligible function a such that the following holds: if Pr[|C.sub.0|=|C.sub.1| and ∀x, C.sub.0(x)=C.sub.1(x): (C.sub.0, C.sub.1, σ)←Samp]>1-α(λ), then:

(64)

.Math. Pr[$D(\ ,\mathrm{iO}(C_0))=1:(C_0,C_1,\ )\leftarrow$ Samp] - Pr[$D(\ ,\mathrm{iO}(C_1))=1:(C_0,C_1,\ )\leftarrow$ Samp] .Math. ≤ ( )

Puncturable Pseudorandom Function ("PRF") Definitions

(65) In certain embodiments, a puncturable family of PRFs F may be given by a triple of Turing Machines Key.sub.F, Puncture.sub.F, and Eval.sub.F, and a pair of computable functions n(.Math.) and m(.Math.) that may satisfy the following conditions: Functionality preserved under puncturing: For PPT adversaries A such that A outputs a PRF input x*∈{0, 1}.sup.n(λ), ∀K∈Key.sub.F, K[x*]←Puncture.sub.F (K, x*): Eval.sub.F (K, x)=Eval.sub.F (K[x*], x). Pseudorandom at punctured points: For PPT adversaries (A.sub.1, A.sub.2) such that A.sub.1 outputs a set S.Math.{0, 1}.sup.n(λ) and state σ, an experiment may be constructed where K←Key.sub.F and K[S]=Puncture.sub.F (K, S), then:

(66)

.Math. Pr[$A_2(\ ,K[S],S,\mathrm{Eval}_F(K,S))=1$] - Pr[$A_2(\ ,K[S],S,U_{m(\ )}$ .Math. .Math. S .Math. $)=1$] .Math. = negl( )

where Eval.sub.F (K, S) denotes the concatenation of Eval.sub.F (K, x.sub.1), . . . , Eval.sub.F (K, x.sub.k), S={x.sub.1, . . . , x.sub.k} is the enumeration of the elements of S in lexicographic order and U.sub.l denotes a uniform distribution over l bits.

Puncturable Public Key Encryption Scheme Definitions

(67) In some embodiments, a puncturable public key encryption scheme may be given by a tuple of Turing Machines ζ=(KeyGen, Enc, Dec, Puncture.sub.ζ, PDec) with the following syntax examples: KeyGen.fwdarw.(pk,sk): The key-generation algorithm may output a public-key secret-key pair. Enc(pk, m).fwdarw.C: On input a public key pk, a message m, the encryption algorithm may output a ciphertext C. Dec(sk, C).fwdarw.m: On input a secret key sk and a ciphertext C, the decryption algorithm may output the plaintext m. Puncture.sub.ζ(sk, C).fwdarw.sk[C]: On input a secret key sk and a ciphertext C, the puncturing algorithm may output a punctured secret key sk[C]. PDec(sk[C*], C).fwdarw.m: On input a punctured secret key sk[C*] and a ciphertext C, the "punctured" decapsulation algorithm may output the plaintext if C≠C*; otherwise, it may output ⊥.

(68) In certain embodiments, the scheme may satisfy the property of extended chosen plaintext attached ("CPA") security. This may property specify that, for PPT adversaries A=(A.sub.1, A.sub.2), there may exist a negligible function negl(.Math.) such that Adv.sub.ζ,A.sup.eCPA defined below may be negl(λ):

(69) Experiment eCPA:

(70) (pk,sk)←KeyGen (m.sub.0, m.sub.1,st)←A.sub.1(pk) b*←{0, 1} C*←Enc(pk, m.sub.b*) sk[C*]←Puncture.sub.ζ (sk, C*) b′←A.sub.2(st, C*, sk[C]) Output 1 if b′=b* and 0 otherwise.

(71) $\mathrm{Adv}^{\mathrm{eCPA}}_{,A}$: = .Math. Pr[ExperimenteCPA .fwdarw. 1] - $\frac{1}{2}$ .Math.

(72) In various embodiments, a FE scheme may use a setup algorithm that first generates a master public key—master secret key pair (mpk, msk). A plaintext may be encrypted using mpk. A secret key sk.sub.f may be generated for a functionality f by using a key-generation algorithm. Using such a secret key, a ciphertext may be decrypted to generate f(m), where m is the plaintext encrypted in the ciphertext.

(73) Functional Encryption for Randomized Functionalities Definitions

(74) In some embodiments, a functional encryption scheme rFE for a randomized function family F={F.sub.λ}.sub.λ over a message space X={X.sub.λ}.sub.λ, a randomness space R={R.sub.λ}.sub.λ and the output space Y={Y.sub.80}.sub.λ comprising a tuple of Turing Machines (Setup, Enc, KeyGen, Dec) may have the following example syntax: Setup.fwdarw.(mpk, msk): The setup algorithm outputs a master public key mpk and a master secret key msk. Enc(mpk, m).fwdarw.ct: On input a master public key mpk and a message m, the encryption algorithm outputs a ciphertext ct. KeyGen(msk, f).fwdarw.skf: On input a master secret key msk and a function f∈F.sub.λ, the key generation algorithm outputs a secret key sk.sub.f. Dec(mpk,skf, ct).fwdarw.y/⊥: On input a master public key mpk, a secret key sk.sub.f corresponding to some function f and a ciphertext ct, the decryption algorithm either outputs a string y∈Y.sub.λ or a special symbol ⊥. In some embodiments, this algorithm may be deterministic.

(75) In certain embodiments, this scheme may satisfy the following correctness property: For every polynomial n=n(λ), every f∈F.sub.λ.sub.n and every x∈X.sub.λ.sub.n, the following two distributions may be computationally indistinguishable: 1. Real: {Dec(mpk,sk.sub.fi), ct.sub.j}.sub.i,j∈[n], where: (mpk, msk)←Setup; sk.sub.fi←KeyGen (msk, f.sub.i) for i∈[n]; ct.sub.j←Enc(mpk, x.sub.j). 2. Ideal: {f.sub.i(x.sub.j; r.sub.ij)}.sub.i,j∈[n], where r.sub.ij←R.sub.λ.

Example-Simulation-Security for rFE Definitions

(76) In certain instances herein, F={F.sub.λ}.sub.λ∈N may be a randomized function family over a domain X=

{X.sub.λ}.sub.λ∈N and randomness space R={R.sub.λ}.sub.λ∈N. rFE=(Setup, Enc,KeyGen, Dec) may be a randomized FE scheme for F with ciphertext space T. rFE may be simulation-secure against malicious encrypters if there exists an efficient simulator S=(S.sub.1, S.sub.2, S.sub.3, S.sub.4, S.sub.5) such that for efficient adversaries A=(A.sub.1, A.sub.2) where A.sub.1 makes at most q.sub.1 key-generation queries and A.sub.2 makes at most q.sub.2 key-generation queries, the outputs of the following experiments may be computationally indistinguishable:
(77) TABLE-US-00001 Experiment Real.sub.A.sup.rFE: Experiment Real.sub.A.sup.rFE: (mpk, msk) ← Setup (mpk, st′) ← S.sub.1 (x, st) ← custom character (mpk), xϵX.sub.λ.sup.q.sup.c (x, st) ← custom character (mpk) where ct.sub.i*←Enc(mpk, x.sub.i) for iϵ[q.sub.c] xϵX.sub.λ.sup.q.sup.c α ← custom character (mpk, {ct.sub.i*}, st) Let f′.sub.1, ... , f′.sub.q1 be A.sub.1's oracle queries to custom character (st′;) Output (x, {f}, {g}, {y}, α) Pick r.sub.ij ← R.sub.λ, let y.sub.ij =f.sub.j(x.sub.i;r.sub.ij), ∀i ϵ [q.sub.c], j ϵ [q.sub.1] ({ct.sub.i*}, st′) ← S.sub.3(st′, {y.sub.ij}) α ← custom character (mpk, {ct.sub.i*}, st) Output (x, {f′}, {g′}, {y′}, α) where the oracles O.sub.1(msk,.Math.), O.sub.1′(st′,.Math.), O.sub.2(msk,.Math.), O.sub.2′(st′,.Math.) are the analogs of the key generation oracles: Real Experiment: Oracles O.sub.1(msk,.Math.) and O.sub.2(msk,.Math.) implement KeyGen(msk,.Math.), and {f} is the (ordered) set of key queries made to oracles O.sub.1(msk,.Math.) and O.sub.2(msk,.Math.). Ideal Experiment: Oracles O.sub.1′(st′,.Math.) and O.sub.2′(st′,.Math.) are the simulator algorithms S.sub.2(st′,.Math.) and S.sub.4(st′,.Math.), respectively. The simulator S.sub.4 may be given oracle access to KeyIdeal(x,.Math.) which on input a function f'∈F.sub.λ, output f'(x.sub.i; r.sub.i) for every x.sub.i∈x and r.sub.i←R.sub.λ. The set {f}, which may be ordered, may include key queries made to O.sub.1′(st′,.Math.) and the queries S.sub.4 makes to KeyIdeal.
(78) Oracles O.sub.3(msk,.Math.,.Math.) and O.sub.3(st′,.Math.,.Math.) are the decryption oracles that take inputs of the form (g,C) where g∈F.sub.λ and C={ct.sub.i}.sub.i∈[m] is a collection of m ciphertexts, where m is polynomial in λ. For queries made in a post-challenge phase, ct.sub.i*∈C for all i∈[q.sub.c]. Real Experiment: On input (g, C), O.sub.3 computes sk.sub.f←KeyGen(msk, f). For i∈[m], it may set y.sub.i=Dec(sk.sub.f, ct.sub.i) and reply with the ordered set {y.sub.i}.sub.i∈[m]. The ordered set {g} may denote the functions that appears in the decryption queries of A.sub.2 and {y} may denote the set of responses of O.sub.3. Ideal Experiment: On input (g, C), O.sub.3′ may do the following: 1. For each ct.sub.i′∈C′, invoke the simulator algorithm S.sub.5(st′, ct.sub.i′) to obtain a value x.sub.i∈X.sub.λ∪{⊥}. 2. For each i∈{m}, if x.sub.i=⊥, then the oracle may set y.sub.i′=⊥. Otherwise, the oracle may chose r.sub.i←R.sub.λ and set y.sub.i′=g′(x.sub.i; r.sub.i). 3. Output the ordered set of responses {y.sub.i′}.sub.i∈[m
(79) The set {g′}, which may be ordered, may denote the functions that appear in the decryption queries of A.sub.2 and {y′} denotes the outputs of O.sub.3′.
(80) Proxy Reencryption Using Commutative Encryption Pair
(81) In various embodiments, proxy reencryption may be implemented using a commutative encryption scheme pair. Consistent with embodiments disclosed herein, a commutative encryption scheme pair may describe a process where ciphertexts under a delegator's public key can be converted into ciphertexts of the same plaintexts under the delegatee's public key in the following manner: Consider an encryption of the delegator's secret key under the delegatee's public key. The resulting ‘special’ ciphertext may be treated as the secret key to ‘decrypt’ ciphertexts under the delegator's public key. This may result in ciphertexts of the same plaintexts under the delegatee's public key. In some embodiments, the delegatee's encryption/decryption algorithms may be homomorphic with respect to the delegator's encryption/decryption algorithms.
(82) In certain embodiments, there may be a direct mapping between the algorithms of a commutative encryption scheme pair to a PRE scheme. The special ciphertext, that is used to convert ciphertexts from under the delegator's public key to under the delegatee's public key, may comprise the reencryption key.
(83) Proxy Reencryption Using Commutative Encryption Notations
(84) In certain instances herein, λ may denote the security parameter. If two distributions D.sub.1, D.sub.2 are statistically close, then this may be denoted by D.sub.1˜D.sub.2. s←S may denote randomly sampling an element s from a set S. A bit string s may be denoted as being sampled uniformly at random, where the length is implicit, by s←$. The ← symbol may be overloaded in y←A(x) to denote that, upon execution of an algorithm A with x as the input, the output is y. In certain instances herein, every algorithm may receive the security parameter 1.sup.λ, even if it is not explicitly specified.
(85) As discussed above, a function negl may be negligible if ∀.sub.c∈N, ∃n.sub.0∈N, such that ∀.sub.n≥n.sub.0, negl(n)<n.sup.−c. A negligible function may be denoted by negl. A negligible function may grow slower than other polynomials.
(86) Proxy Reencryption Scheme
(87) In various embodiments, a PRE scheme may allow a secret key holder to create a reencryption key. A semi-

trusted proxy, such as a DRM service, can use this key to translate a message m encrypted under the delegator's public key into an encryption of the same message under a delegatee's public key, as specified by the delegator. This may be done without allowing the proxy the ability to perform tasks outside of these proxy delegations. For example, in some embodiments, the proxy can neither recover the delegator's secret key nor decrypt the delegator's ciphertext.

(88) In some embodiments, a PRE scheme may comprise the following algorithms, where a, b may be two special symbols corresponding to delegator and delegate, respectively: (pk.sub.δ, sk.sub.δ) ← KeyGen(δ): The key generation algorithm may take δ∈{a,b} and generate a public/secret key pair (pk.sub.δ, sk.sub.δ). ct ← Enc(δ, pk.sub.δ, m): The encryption algorithm may take as input δ∈{a,b}, a public key pk.sub.δ, and a message m. It may output a ciphertext ct under pk.sub.δ. m ← Dec(δ, sk.sub.δ, ct): The decryption algorithm may take as input δ∈{a,b}, a secret key sk.sub.δ and a ciphertext ct under pk.sub.δ. It may output a message m. rk ← ReKeyGen(sk.sub.a, pk.sub.b): The reencryption key generation algorithm may take as input a private key sk.sub.a and another public key pk.sub.b. It may output a reencryption key rk. ct.sub.b ← ReEnc(rk, ct.sub.a): The reencryption algorithm may take as input a reencryption key rk and a ciphertext ct.sub.a under public key pk.sub.a. It may output a ciphertext ct.sub.b under public key pk.sub.b.

(89) In certain embodiments, a PRE scheme may satisfy the following correctness property: Informally, a party holding a secret key sk.sub.b may be able to decrypt ciphertexts encrypted under pk.sub.b and also ciphertexts generated as ReEnc(rk, ct.sub.a). Formally, correctness may be satisfied as follows: 1. The following may correspond to the correctness of (KeyGen, Enc, Dec) being an encryption scheme. ∀(pk,sk) ← KeyGen(δ), ∀m, ∀ct ← Enc(δ, pk, m): Dec(δ,sk, ct)=m. 2. The following may correspond to a requirement that reencrypted ciphertexts are decrypted correctly by the delegatee. ∀(pk.sub.a,sk.sub.a) ← KeyGen(a), ∀(pk.sub.b,sk.sub.b) ← KeyGen(b), ∀m, ∀ct.sub.a ← Enc(a, pk.sub.a, m), ∀rk ← ReKeyGen(sk.sub.a, pk.sub.b), ∀ct.sub.b ← ReEnc(rk, ct.sub.a) Dec(b, sk.sub.b, ct.sub.b)=m

Obfuscation Security of Proxy Reencryption

(90) In certain embodiments, average-case obfuscation security for PRE may view having access to a reencryption key as equivalent to having a black-box (and/or oracle) access to the corresponding reencryption functionality. In other words, an efficient adversary may not learn either the delegator's secret key and/or the plaintexts.

(91) A PRE scheme ζ=(KeyGen, Enc, Dec, ReKeyGen, ReEnc) may be said to be average-case obfuscation secure if the following holds: For any efficient adversary A, there may exist an efficient simulator S and a negligible function negl(.Math.), such that:

(92)

$$\Pr\left[\begin{array}{l}(pk_a, \mathrm{sk}_a) \leftarrow \mathrm{KeyGen}(a),\\ (pk_b, \mathrm{sk}_b) \leftarrow \mathrm{KeyGen}(b),\\ \mathrm{rk} \leftarrow \mathrm{ReKeyGen}(sk_a, pk_b)\end{array} : {}^{[\mathrm{sk}_b, \mathrm{pk}_b](.\mathrm{Math.})}(pk_a, pk_b, \mathrm{rk})\ .\mathrm{fwdarw.}\ 1\right]$$

.Math. .Math. ≤ negl(

$$\Pr\left[\begin{array}{l}(pk_a, \mathrm{sk}_a) \leftarrow \mathrm{KeyGen}(a),\\ (pk_b, \mathrm{sk}_b) \leftarrow \mathrm{KeyGen}(b),\end{array} : {}^{[\mathrm{sk}_b, \mathrm{pk}_b](.\mathrm{Math.})}(pk_a, pk_b, {}^{[\mathrm{sk}_b, \mathrm{pk}_b](.\mathrm{Math.})})\ .\mathrm{fwdarw.}\ 1\right]$$

where O[sk.sub.a, pk.sub.b](.Math.) is an oracle that may take as an input a ciphertext ct.sub.a under pk.sub.a and output Enc(b, pk.sub.b, m), where m ← Dec(a, sk.sub.a, pk.sub.b). In other words, the oracle may perform the reencryption function by first decrypting the input ciphertext with sk.sub.a, then encrypting the resulting plaintext with pk.sub.b, and outputting the resulting ciphertext.

(93) Average-case obfuscation security may capture CPA security for the delegatee. A proxy reencryption scheme consistent with various aspects of the disclosed embodiments may further satisfy CPA security for the delegator.

(94) Commutative Encryption Schemes

(95) Consistent with various embodiments, a PRE key and/or associated reencryption program may essentially perform a decryption followed by an encryption, without revealing the details of the decryption (i.e., the decryption key or the decrypted plaintext). The PRE key and/or associated reencryption program may “contain” the delegator's secret key, but not in the clear. Hence, the secret key may be encoded/encrypted in a suitable manner.

(96) In some embodiments, the pair of schemes is such that, if the delegator's secret key is encrypted under the delegatee's public key, then the resulting ciphertext can be used to perform decryptions of ciphertexts under delegator's public key ‘under the hood’. That is, the decrypted value may still be under the delegatee's public key. In certain embodiments, this may be described as a limited form of homomorphic property of the delegatee's

public-key encryption ("PKE") scheme with respect to the delegator's PKE scheme. In some embodiments, the PRE key could contain an encryption under the delegatee's public key of the delegator's secret key. With this, ciphertexts under delegator's public key can be transformed into ciphertexts of the same plaintexts under the delegatee's public key. FIG. **6** illustrates a diagram of a representation **600** of an example of a commutative encryption scheme consistent with certain embodiments of the present disclosure.

(97) Commutative Encryption Scheme Definitions

(98) In some embodiments, an ordered pair of chosen-ciphertext attack ("CPA") secure encryption schemes (Σ.sub.a, Σ.sub.b) may be said to be commutative if the following holds: Let Σ.sub.a=(KeyGen.sub.a, Enc.sub.a, Dec.sub.a) and Σ.sub.b=(KeyGen.sub.b, Enc.sub.b, Dec.sub.b). The pair may be said to be commutative if the following holds:

(99)
$$\forall(\mathrm{pk}_a, \mathrm{sk}_a) \leftarrow \mathrm{KeyGen}_a(), :$$
$$\forall(\mathrm{pk}_b, \mathrm{sk}_b) \leftarrow \mathrm{KeyGen}_b(), \quad \mathrm{ct}_b \leftarrow \mathrm{Dec}_a(\tilde{ct}, \mathrm{ct})$$
$$\forall m, \quad m \leftarrow \mathrm{Dec}_b(\mathrm{sk}_b, \mathrm{ct}_b)$$
$$\forall \tilde{ct} \leftarrow \mathrm{Enc}_b(\mathrm{pk}_b\, \mathrm{sk}_a),$$
$$\forall \mathrm{ct}_a \leftarrow \mathrm{Enc}_a(\mathrm{pk}_a, m)$$

(100) In certain embodiments, the secret-key space of Σ.sub.a may be a subset of the plaintext space of Σ.sub.b and the ciphertext space of Σ.sub.b may be a subset of the secret-key space of Σ.sub.a.

(101) Proxy Reencryption from a Commutative Encryption Scheme Pair

(102) Consistent with various disclosed embodiments, a PRE scheme may be constructed from a commutative encryption scheme pair. Let Σ.sub.a=(KeyGen.sub.a, Enc.sub.a, Dec.sub.a) and Σ.sub.b=(KeyGen.sub.b, Enc.sub.b, Dec.sub.b) be a commutative scheme pair. In some embodiments, a PRE scheme may be constructed ζ=(KeyGen, Enc, Dec, ReKeyGen, ReEnc) as follows: KeyGen(δ): Run KeyGen.sub.δ. Output the resulting public key/secret key pair. Enc(δ, pk.sub.δ, m): Run Enc.sub.δ(pk.sub.δ, m). Output the resulting ciphertext. Dec(δ, sk.sub.δ, ct): Run Dec.sub.δ(sk.sub.δ, ct). Output the resulting plaintext. ReKeyGen(sk.sub.a, pk.sub.b): Compute rk←Enc.sub.b(pk.sub.b, sk.sub.a). Output rk. ReEnc(rk, ct.sub.a): Compute ct.sub.b←Dec.sub.a(rk, ct.sub.a). Output ct.sub.b.

(103) Correctness: Correctness of the PRE scheme ζ may be verified. In some embodiments, the correctness property of the scheme may follow from the correctness properties of the underlying encryption schemes. Consider any: (pk.sub.a, sk.sub.a)←KeyGen(a)←KeyGen.sub.a( ), (pk.sub.b, sk.sub.b)←KeyGen(b)←KeyGen.sub.b( ), m, ct.sub.b+Enc(a, pk.sub.a, m)=Enc.sub.a(pk.sub.a, m) rk←ReKeyGen(sk.sub.a, pk.sub.b)=Enc.sub.b(pk.sub.b, sk.sub.a) ct.sub.b←ReEnc(rk, ct.sub.a)=Dec.sub.a(rk, ct.sub.a)
Since the encryption scheme pair Σ.sub.a, Σ.sub.b may be commutative: m←Dec.sub.b(sk.sub.b, ct.sub.b)

(104) FIG. **7** illustrates an example of a method for managing protected content consistent with certain embodiments of the present disclosure. The illustrated method **700** may be implemented in a variety of ways, including using software, firmware, hardware, and/or any combination thereof. In certain embodiments, various aspects of the method **700** and/or its constituent steps may be performed by a user device, a content service, a DRM service, and/or any other suitable system and/or services or combination of systems and/or services.

(105) At **702**, a protected reencryption program may be received from a content service system at a rights management system. In various embodiments, the protected reencryption program may comprise an obfuscated program that may be obfuscated using, for example, iO, FE, whitebox cryptographic obfuscation, and/or any other suitable software obfuscation and/or protection technique and/or combination of techniques, including any of the techniques disclosed herein. In some embodiments, the protected reencryption program may include a protected private decryption key of the content service system. The rights management system may further receive from the content service system an encrypted content key encrypted using a public encryption key of the content service system and an identifier of a piece of content associated with the encrypted content key.

(106) A license request message may be received from a user device at **704**. In certain embodiments, the license request message may include an identifier associated with the piece of content and/or a public encryption key of the user device. In some embodiments, the license request message may further comprise a signed message. For example, the license request message may comprise a content request message issued by the user device to the content service signed by the content service.

(107) At **706**, a reencrypted content key may be generated based on the encrypted content key and the public encryption key of the user device using the protected reencryption program. In some embodiments, the reencrypted content key may be encrypted using the public encryption key of the user device. Consistent with

various aspects of the disclosed embodiments, the reencrypted content key may be generated without exposing plaintext of the content key and/or operations using the same to the rights management system during execution of the protected reencryption program.
(108) In some embodiments, generating the reencrypted content key may include decrypting, in a protected execution process, the encrypted content key using the protected private decryption key of the content service system to generate a decrypted content key and, encrypting the decrypted content key using the public encryption key of the user device to generate the reencrypted content key.
(109) A content license associated with the piece of content that includes the reencrypted content key and/or any associate licenses terms may be generated at **708**. At **710**, the generated content license may be communicated to the user device.
(110) FIG. **8** illustrates an exemplary system **800** that may be used to implement embodiments of the systems and methods of the present disclosure. Certain elements associated with the illustrated exemplary system may be included in a user device, a content service, a DRM service, and/or any other system and/or service configured to implement embodiments of the disclosed systems and methods.
(111) As illustrated in FIG. **8**, the system **800** may include: a processing unit **802**; system memory **804**, which may include high speed random access memory ("RAM"), non-volatile memory ("ROM"), and/or one or more bulk non-volatile non-transitory computer-readable storage mediums (e.g., a hard disk, flash memory, etc.) for storing programs and other data for use and execution by the processing unit **802**; a port **806** for interfacing with removable memory **808** that may include one or more diskettes, optical storage mediums, and/or other non-transitory computer-readable storage mediums (e.g., flash memory, thumb drives, USB dongles, compact discs, DVDs, etc.); a network interface **810** for communicating with other systems via one or more network connections **812** using one or more communication technologies; a user interface **814** that may include a display and/or one or more input/output devices such as, for example, a touchscreen, a keyboard, a mouse, a track pad, and the like; and one or more busses **816** for communicatively coupling the elements of the system.
(112) In some embodiments, the system **800** may, alternatively or in addition, include an SPU **818** that is protected from tampering by a user of the system or other entities by utilizing secure physical and/or virtual security techniques. An SPU **818** can help enhance the security of sensitive operations such as personal information management, trusted credential and/or key management, license, privacy, and policy management, and other aspects of the systems and methods disclosed herein. In certain embodiments, the SPU **818** may operate in a logically secure processing domain and be configured to protect and operate on secret information, including cryptographic keys, as described herein. In some embodiments, the SPU **818** may include internal memory storing executable instructions or programs configured to enable the SPU **818** to perform secure operations.
(113) The operation of the system **800** may be generally controlled by a processing unit **802** and/or an SPU **818** operating by executing software instructions and programs stored in the system memory **804** (and/or other computer-readable media, such as removable memory **808**). The system memory **804** may store a variety of executable programs or modules for controlling the operation of the system. For example, the system memory may include an operating system ("OS") **820** that may manage and coordinate, at least in part, system hardware resources and provide for common services for execution of various applications and a trust and privacy management system **822** for implementing trust and privacy management functionality including protection and/or management of secure data and/or keys through management and/or enforcement of associated policies. The system memory **804** may further include, without limitation, communication software **824** configured to enable in part communication with and by the system **800**; one or more applications; a secure communication and/or processing module **826** configured to perform various aspects of the disclosed embodiments (e.g., message generation, cryptographic operations, etc.), a DRM module **828** configured to perform various aspects of the disclosed embodiments (e.g., license request and/or response generation and/or processing, cryptographic operations including protected reencryption operations, and/or the like), and/or any other information and/or applications configured to implement embodiments of the systems and methods disclosed herein.
(114) The systems and methods disclosed herein are not inherently related to any particular computer, device, service, or other apparatus and may be implemented by a suitable combination of hardware, software, and/or firmware. Software implementations may include one or more computer programs comprising executable code/instructions that, when executed by a processor, may cause the processor to perform a method defined at least in part by the executable instructions. The computer program can be written in any form of programming language, including compiled or interpreted languages, and can be deployed in any form, including as a standalone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. Further, a computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

Software embodiments may be implemented as a computer program product that comprises a non-transitory storage medium configured to store computer programs and instructions, that when executed by a processor, are configured to cause the processor to perform a method according to the instructions. In certain embodiments, the non-transitory storage medium may take any form capable of storing processor-readable instructions on a non-transitory storage medium. A non-transitory storage medium may be embodied by a compact disk, digital-video disk, an optical storage medium, flash memory, integrated circuits, or any other non-transitory digital processing apparatus memory device.

(115) Although the foregoing has been described in some detail for purposes of clarity, it will be apparent that certain changes and modifications may be made without departing from the principles thereof. It should be noted that there are many alternative ways of implementing both the systems and methods described herein. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.

Claims

1. A method for managing data performed by an electronic data access management system comprising a processor and a non-transitory computer-readable medium storing instructions that, when executed by the processor, cause the electronic data access management system to perform the method, the method comprising: receiving, from an electronic data service system, a protected reencryption program, an encrypted data access key encrypted using a public encryption key of the electronic data service system, and an identifier of electronic data associated with the encrypted data access key, the protected reencryption program comprising a protected private decryption key of the electronic data service system; receiving, from a user device, a data access request message, the data access request message comprising the identifier of the electronic data and a public encryption key of the user device; generating a reencrypted data access key using the protected reencryption program based on the encrypted data access key and the public encryption key of the user device, wherein generating the reencrypted data access key comprises: decrypting the encrypted data access key by the protected reencryption program to generate a data access key, and encrypting the data access key using the public encryption key of the user device to generate the reencrypted data access key, wherein decrypting the encrypted data access key to generate the data access key and encrypting the data access key to generate the reencrypted data access key are performed without exposing plaintext of the data access key to the electronic data access management system outside the protected reencryption program during execution of the protected reencryption program; generating a data access response associated with the electronic data, the data access response comprising the reencrypted data access key; and transmitting the data access response to the user device.

2. The method of claim 1, wherein the data access request message further comprises a signed message.

3. The method of claim 2, wherein the method further comprises: prior to generating the reencrypted data access key and the data access response, verifying a signature of the signed message.

4. The method of claim 3, wherein the signed message comprises a data access request message issued by the user device to the electronic data service system signed by the electronic data service system.

5. The method of claim 3, wherein verifying the signature of the signed message comprises verifying that the signed message has been signed by the electronic data service system.

6. The method of claim 1, wherein generating the reencrypted data access key comprises generating the reencrypted data access key without exposing the protected private decryption key of the electronic data service system to the electronic data access management system during execution of the protected reencryption program.

7. The method of claim 1, wherein the protected reencryption program comprises an obfuscated program.

8. The method of claim 7, wherein the protected reencryption program is obfuscated using indistinguishability obfuscation.

9. The method of claim 7, wherein the protected reencryption program is obfuscated using whitebox cryptographic obfuscation.

10. The method of claim 1, wherein the data access response further comprises one or more data access terms relating to use of the electronic data.

11. The method of claim 1, wherein the electronic data comprises protected content.

12. The method of claim 11, wherein the data access key comprises a content key.

13. The method of claim 1, wherein the protected reencryption program is associated with a system identifier associated with the user device.

14. The method of claim 13, wherein the method further comprises receiving the system identifier from the user

device.

15. The method of claim 1, wherein the public encryption key of the user device is issued to the user device as part of a registration process.