United States Patent	12395475
Kind Code	B2
Date of Patent	August 19, 2025
Inventor(s)	Mitchell; Stephen G.

Cryptographic systems and methods

Abstract

Systems and methods are described that use cryptographic techniques to improve the security of applications executing in a potentially untrusted environment associated with a software application. Embodiments of the disclosed systems and methods may, among other things, facilitate cryptographic operations within an execution environment associated with browser software of a client system while maintaining security of cryptographic keys imported into the environment. As the security of keys is maintained in an execution environment implementing embodiments of the disclosed systems and methods, users and/or systems may be more willing to consign their keys for use in connection with cryptographic operations performed in such environments.

Inventors:	**Mitchell; Stephen G. (Ben Lomond, CA)**
Applicant:	**Intertrust Technologies Corporation** (Berkeley, CA)
Family ID:	**1000008763276**
Assignee:	**Intertrust Technologies Corporation (Berkeley, CA)**
Appl. No.:	**18/530893**
Filed:	**December 06, 2023**

Prior Publication Data

Document Identifier	Publication Date
US 20240106809 A1	Mar. 28, 2024

Related U.S. Application Data

continuation parent-doc US 17530009 20211118 US 11848922 child-doc US 18530893
continuation parent-doc US 16245447 20190111 US 11196724 20211207 child-doc US 17530009
continuation parent-doc US 14991687 20160108 US 10205710 20190212 child-doc US 16245447
us-provisional-application US 62101293 20150108

Publication Classification

Int. Cl.: **H04L9/40** (20220101); **H04L9/08** (20060101); **H04L9/32** (20060101)

U.S. Cl.:

CPC   **H04L63/045** (20130101); **H04L9/0825** (20130101); **H04L9/0897** (20130101); **H04L9/3247** (20130101); **H04L63/061** (20130101); **H04L63/0823** (20130101); H04L2209/16 (20130101)

Field of Classification Search

CPC: H04L (63/045); H04L (9/0825); H04L (9/0897); H04L (9/3247); H04L (63/061); H04L (63/0823); H04L (2209/16); H04L (9/0822); H04L (9/3263); G06F (21/53); G06F (21/602); G06F (21/72); H04W (12/069)

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
5991399	12/1998	Graunke	713/168	G11B 20/00086
6735313	12/2003	Bleichenbacher	348/E7.056	H04N 21/4405
7051005	12/2005	Peinado	713/193	H04L 63/0442
7093129	12/2005	Gavagni	713/153	H04L 9/3263
7802111	12/2009	Tahan	713/193	G06F 21/57
8213620	12/2011	Sussland	380/278	H04L 9/0894
8234387	12/2011	Bradley et al.	N/A	N/A
8245270	12/2011	Cooperstein	713/168	G06F 21/53
8391494	12/2012	Serenyi	380/278	H04L 63/0428
8583911	12/2012	Miller	713/150	H04L 63/0428
8719590	12/2013	Faibish	713/190	G06F 21/602
8762712	12/2013	Kwan	713/165	H04L 9/0825
8776216	12/2013	Boccon-Gibod et al.	N/A	N/A
8838974	12/2013	Webb	713/168	G06F 21/51
8843750	12/2013	Sokolov	713/175	H04L 63/0823
8966581	12/2014	Gross	726/4	G06F 21/34
8984285	12/2014	Dorwin	713/151	H04N 21/24

9094206	12/2014	Di Crescenzo	N/A	H04L 9/3073
9141769	12/2014	Hitchcock	N/A	H04L 9/0877
9189609	12/2014	Antony	N/A	G06F 21/10
9225690	12/2014	Fitch	N/A	H04L 63/04
9246690	12/2015	Roth	N/A	G06F 21/53
9251334	12/2015	Molitor	N/A	H04L 9/3271
9367697	12/2015	Roth	N/A	H04L 9/0897
9515996	12/2015	Juels	N/A	H04L 63/0428
9584517	12/2016	Roth	N/A	G06F 21/602
9652631	12/2016	Novak	N/A	H04L 9/3247
9690941	12/2016	Thom	N/A	G06F 21/6209
9692741	12/2016	Vilhuber	N/A	H04L 63/0823
9754116	12/2016	Roth	N/A	G06F 21/602
9767271	12/2016	Ghose	N/A	G06F 21/52
9785489	12/2016	Chheda	N/A	G06F 9/45558
9887836	12/2017	Roth	N/A	H04L 9/088
9910997	12/2017	Brail	N/A	G06F 21/44
10079681	12/2017	Roth	N/A	H04L 47/70
10187363	12/2018	Smirnoff	N/A	H04L 9/0897
10554475	12/2019	Weinstein	N/A	G06F 21/53
10757082	12/2019	Buendgen	N/A	H04L 9/0897
11748492	12/2022	Campagna	713/150	H04L 9/0637
2002/0023212	12/2001	Proudler	714/E11.204	H04L 63/08
2002/0162104	12/2001	Raike	380/282	H04L 9/083
2002/0172367	12/2001	Mulder et al.	N/A	N/A
2003/0194085	12/2002	Dillaway	380/29	G06F 21/6209
2004/0025036	12/2003	Balard	713/176	G06F 21/575
2004/0255137	12/2003	Ying	713/193	H04L 63/0884
2005/0069138	12/2004	de Jong	380/278	G06F 9/3017
2005/0086479	12/2004	Ondet	713/172	G06F 21/10
2005/0114683	12/2004	Jin	713/187	G06F 21/14

2007/0040021	12/2006	Nakayma	235/382	G06F 12/1408
2007/0086586	12/2006	Jakubowski	380/28	H04L 9/002
2007/0160203	12/2006	Sudhakar	380/30	H04L 9/3263
2008/0056501	12/2007	McGough	380/282	H04L 63/08
2008/0063201	12/2007	Wormald	380/255	H04L 63/0823
2008/0065886	12/2007	McGough	713/168	H04L 63/061
2008/0080708	12/2007	McAlister	380/44	H04L 9/0844
2008/0152144	12/2007	Douguet	380/277	H04L 9/3236
2008/0181412	12/2007	Acar	380/279	G06F 21/79
2008/0192937	12/2007	Challener	380/278	H04L 9/14
2008/0229117	12/2007	Shin	713/190	G06F 21/72
2008/0232598	12/2007	Vennelakanti	380/279	G06F 21/62
2008/0301470	12/2007	Green	713/193	G06F 21/6218
2008/0313743	12/2007	Chan	726/28	G06F 21/105
2009/0097660	12/2008	Malaviarachchi	380/279	G06Q 20/3829
2009/0103726	12/2008	Ahmed	380/46	H04L 9/12
2009/0138708	12/2008	Miyazaki	713/168	H04L 63/04
2009/0150681	12/2008	Kaabouch	713/193	H04L 9/14
2009/0158035	12/2008	Stultz	713/160	H04L 63/0442
2009/0169013	12/2008	Fascenda	380/277	G06F 21/72
2009/0296941	12/2008	Devanand	380/281	H04L 63/0428
2010/0058047	12/2009	Medvinsky	713/189	H04L 9/321
2010/0082987	12/2009	Thom	713/171	G06F 21/34
2010/0146292	12/2009	Shi	713/189	H04L 9/3263
2010/0146293	12/2009	Shi	713/189	G06F 21/10
2010/0146501	12/2009	Wyatt	718/1	G06F 21/84
2010/0174919	12/2009	Ito	713/192	G06F 21/74
2010/0223456	12/2009	Schneider	380/277	H04L 63/0428
2010/0293049	12/2009	Maher et al.	N/A	N/A
2010/0318786	12/2009	Douceur	713/155	H04L 63/123
2011/0010547	12/2010	Noda	713/168	H04L 9/321
2011/0085667	12/2010	Berrios	709/203	H04L 9/3249
2011/0179268	12/2010	Strom	713/168	H04L 9/3247
2011/0261964	12/2010	Kahler	380/44	H04L 9/083

2011/0314271	12/2010	Boccon-Gibod et al.	N/A	N/A
2012/0042170	12/2011	Curtin	713/185	G06F 21/77
2012/0140923	12/2011	Lee	380/44	H04L 9/0894
2012/0159148	12/2011	Behren	713/150	G06Q 20/3823
2012/0159178	12/2011	Lin	713/176	H04L 63/065
2012/0166807	12/2011	Shear	713/176	G06F 21/51
2012/0173877	12/2011	Pendakur	713/169	H04N 21/43635
2012/0179902	12/2011	Noda	713/153	H04L 63/06
2012/0192283	12/2011	Gu	726/26	H04L 9/002
2012/0246487	12/2011	Gu	713/190	G06F 21/14
2012/0297200	12/2011	Thom	713/189	G06F 21/602
2012/0328106	12/2011	Dellow	380/283	G06F 21/73
2013/0091353	12/2012	Zhang	713/156	H04L 9/3268
2013/0124862	12/2012	Pestoni	713/168	G06F 21/121
2013/0151848	12/2012	Baumann	713/164	H04L 9/3263
2013/0152180	12/2012	Nair	726/26	G06F 21/575
2013/0166906	12/2012	Swaminathan	713/162	H04N 21/8456
2013/0166923	12/2012	Tsuhara	713/193	G06F 3/0605
2013/0318577	12/2012	Bulusu	726/5	H04W 12/08
2014/0003608	12/2013	MacMillan	380/279	H04L 9/3239
2014/0013109	12/2013	Yin	713/155	H04W 12/068
2014/0089658	12/2013	Raghuram	380/278	G06F 9/45533
2014/0095867	12/2013	Smith	713/164	H04L 63/0861
2014/0108805	12/2013	Smith	713/176	H04L 9/3247
2014/0129844	12/2013	Johnson	713/189	G06F 21/78
2014/0140504	12/2013	Karroumi	380/44	H04L 9/0861
2014/0140508	12/2013	Kamath	380/255	H04L 9/0822
2014/0164790	12/2013	Dodgson	713/193	G06Q 10/10
2014/0181518	12/2013	Kim	713/168	G06F 9/54

2014/0195690	12/2013	Harrison	709/227	H04L 67/141
2014/0195804	12/2013	Hursti	713/168	H04L 9/0863
2014/0195807	12/2013	Bar-El	713/168	H04L 9/0877
2014/0229729	12/2013	Roth	713/153	G06F 21/602
2014/0229732	12/2013	Roth	713/167	H04L 63/0428
2014/0229737	12/2013	Roth	713/176	H04L 9/088
2014/0230007	12/2013	Roth	726/1	G06F 21/602
2014/0237239	12/2013	Hursti	713/168	H04L 9/32
2014/0237252	12/2013	Hursti	713/176	H04L 9/30
2014/0237255	12/2013	Martin	713/182	G06F 21/10
2014/0281481	12/2013	Moroney	713/151	H04N 21/26613
2014/0281483	12/2013	Vigliaturo	713/153	H04L 63/0471
2014/0289535	12/2013	Gan	713/189	H04L 9/32
2014/0321646	12/2013	Ksontini	380/247	H04W 4/60
2014/0359288	12/2013	Jensen	713/168	H04W 12/04
2014/0365763	12/2013	Manohar	713/156	G06F 21/53
2015/0026483	12/2014	Jiang	713/190	G06F 21/10
2015/0052358	12/2014	Udupi	713/171	H04L 9/0822
2015/0058629	12/2014	Yarvis	713/171	H04L 9/0825
2015/0143117	12/2014	Freeman	713/168	H04L 67/1097
2015/0149772	12/2014	Leavy	713/168	H04L 63/06
2015/0150084	12/2014	Kiperberg	726/3	G06F 21/44
2015/0161382	12/2014	Boivie	726/28	G06K 19/06112
2015/0199528	12/2014	Bobinski	713/191	G06F 8/61
2015/0229471	12/2014	Nair	713/171	H04L 9/0891
2015/0248668	12/2014	Radu	705/71	G06Q 20/322
2015/0270951	12/2014	Michiels	380/28	H04L 63/1466
2015/0270956	12/2014	Basmov	713/189	G06F 21/74
2015/0278531	12/2014	Smith	713/165	H04L 63/0428
2015/0312223	12/2014	Michiels	713/168	G06F 21/51
2015/0332262	12/2014	Lingappa	705/71	G06Q 20/3823
2016/0006724	12/2015	Vlot	713/156	G06F 21/72

2016/0036826	12/2015	Pogorelik	726/1	G06F 21/10
2016/0057619	12/2015	Lopez	380/247	G06Q 20/3829
2016/0063219	12/2015	Vlot	713/168	H04L 63/0464
2016/0080144	12/2015	Choi	380/44	H04L 9/0822
2016/0092696	12/2015	Guglani	726/26	G06F 21/606
2016/0125180	12/2015	Smith	726/20	H04L 9/0822
2016/0127351	12/2015	Smith	726/10	G06F 21/316
2016/0134623	12/2015	Roth	713/156	H04L 63/0823
2016/0164849	12/2015	Smith	713/155	H04L 9/3263
2016/0180066	12/2015	Michiels	713/189	H04L 9/0631
2016/0182227	12/2015	Michiels	380/28	H04L 9/06
2016/0182472	12/2015	Michiels	713/171	H04L 63/0876
2016/0182499	12/2015	Sharaga	713/156	G06F 21/82
2016/0191236	12/2015	Smirnoff	713/171	H04L 9/0822
2016/0205074	12/2015	Mitchell	713/171	H04L 63/045
2016/0226661	12/2015	Keidar	N/A	H04L 9/3231
2016/0261592	12/2015	Hubert	N/A	G06F 21/53
2016/0315917	12/2015	Smith	N/A	H04L 63/10
2017/0048206	12/2016	Sui	N/A	H04L 63/0281
2017/0063811	12/2016	Hitchcock	N/A	H04L 9/3234
2017/0116438	12/2016	Ekberg	N/A	H04L 63/0876
2017/0187524	12/2016	Furukawa	N/A	H04L 9/0844
2017/0213027	12/2016	Gu	N/A	G06F 21/556
2017/0228525	12/2016	Wajs	N/A	H04L 9/3226
2017/0237551	12/2016	Van Foreest	713/189	G06F 21/10
2017/0243000	12/2016	Shraim	N/A	H04L 9/00
2017/0286141	12/2016	Adler	N/A	G06F 9/4856
2017/0288874	12/2016	Narendra Trivedi	N/A	G06F 9/45533
2017/0310647	12/2016	Hu	N/A	G06F 21/33

2017/0330402	12/2016	Menard	N/A	G07C 9/00857
2017/0373828	12/2016	Michiels	N/A	G06F 21/75
2018/0150646	12/2017	Roth	N/A	H04L 63/20
2019/0095617	12/2018	Stefan	N/A	G06F 8/75
2019/0312718	12/2018	Michiels	N/A	G06F 21/75
2020/0044837	12/2019	Bos	N/A	H04W 12/041
2021/0111884	12/2020	Nagao	N/A	H04L 9/0861
2021/0390189	12/2020	Nagao	N/A	H04L 9/3247
2022/0094536	12/2021	Nilsson	N/A	H04L 9/3263

FOREIGN PATENT DOCUMENTS

Patent No.	Application Date	Country	CPC
2341682	12/2010	EP	N/A
WO-2011068996	12/2010	WO	G06F 12/1408

OTHER PUBLICATIONS

Final Office Action mailed Apr. 17, 2018, received in U.S. Appl. No. 14/991,687. cited by applicant
International Search Report and Written Opinion dated May 3, 2016, received in PCT Application No. PCT/US2016/012726 (7 pages). cited by applicant
Notice of Allowance and Fee(s) Due mailed Oct. 11, 2018, received in U.S. Appl. No. 14/991,687. cited by applicant
Office Action mailed Aug. 11, 2017, received in U.S. Appl. No. 14/991,687. cited by applicant

---

Primary Examiner: Cervetti; David Garcia

Attorney, Agent or Firm: Thayne and Davis LLC

Background/Summary

RELATED APPLICATIONS (1) This application is a continuation of U.S. Nonprovisional application Ser. No. 17/530,009 filed Nov. 18, 2021, which is a continuation of U.S. Nonprovisional application Ser. No. 16/245,447, filed Jan. 11, 2019 (now U.S. Pat. No. 11,196,724), and entitled "CRYPTOGRAPHIC SYSTEMS AND METHODS," which is a is a continuation of U.S. Nonprovisional application Ser. No. 14/991,687, filed Jan. 8, 2016 (now U.S. Pat. No. 10,205,710), and entitled "CRYPTOGRAPHIC SYSTEMS AND METHODS," which claims the benefit of priority under 35 U.S.C. § 119(e) to U.S. Provisional Patent Application No. 62/101,293, filed Jan. 8, 2015, and entitled "CRYPTOGRAPHIC SYSTEMS AND METHODS," 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

BACKGROUND AND SUMMARY

(2) The present disclosure relates generally to systems and methods for performing cryptographic operations. More specifically, but not exclusively, the present disclosure relates to systems and methods that use cryptographic techniques to protect secure information shared with a potentially untrusted execution environment associated with a software application.

(3) Conventional cryptographic services implemented within browser software of a client system may be vulnerable to certain attacks. For example, a server provisioning a cryptographic implementation to browser software of a client system and/or a communication channel associated with the same may be compromised (e.g., via a man-in-the-middle attack or the like). In view of these potential vulnerabilities, a user of a client system may be unwilling to provide certain secure user keys or other sensitive data to cryptographic implementations and/or other data processing methods operating within browser software downloaded from an untrusted server. Similarly, a server may be unwilling to provide certain secure server keys to browser software of an untrusted client system.

(4) Certain embodiments of the systems and methods disclosed herein provide for secure implementation of cryptographic services including trusted credential and/or key management services operating within browser software executing on a client system. In some embodiments, a user may trust their secure keys to a cryptographic implementation operating within browser software based on the cryptographic service being signed and/or otherwise protected or authenticated by a trusted service (e.g., a trusted third party cryptographic service or the like). Similarly, a server may trust its secure keys to a cryptographic implementation operating within the browser software of a client system. In certain embodiments, the disclosed systems and methods may enable trusted credential and/or secure user key management within a sandboxed area of the client system associated with the browser software, thereby protecting the integrity of the trusted credentials and/or secure keys.

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 a trust configuration process consistent with embodiments disclosed herein.

(3) FIG. **2** illustrates a process of provisioning a client system with a protected client module consistent with embodiments disclosed herein.

(4) FIG. **3** illustrates an exemplary implementation of a cryptographic service included in browser software executing on a client system consistent with embodiments disclosed herein.

(5) FIG. **4** illustrates a flow chart of an exemplary cryptographic operation consistent with embodiments disclosed herein.

(6) FIG. **5** illustrates a system that may be used to implement certain embodiments of the systems and methods of the present disclosure.

DETAILED DESCRIPTION

(7) A detailed description of 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.

(8) Some embodiments of the disclosure may be understood by reference to the drawings, wherein like parts may be designated by like numerals. 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 certain illustrative embodiments is not intended to limit the scope of the disclosure, as claimed, but is merely representative of possible embodiments of the disclosure. In addition, the steps of any method 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.

(9) Embodiments of the systems and methods disclosed herein may employ trusted key management in connection with cryptographic services implemented within browser software executing on a client system. In certain embodiments, the systems and methods may use a trusted service operating as a root-of-trust. Credentials (e.g., certificates, keys, and/or the like) may be provisioned by the trusted service to client and/or server systems for use in connection with protecting the security of keys used in cryptographic operations performed by a protected client module executing within browser software of the client system (e.g., a JavaScript application and/or the like). In certain embodiments, the protected client module may be protected using a variety of cryptographic and/or white-box cryptographic techniques employed by the trusted service. Using such an architecture, the integrity of keys shared by a user of client system and/or the server system with the protected client module may be maintained. Although certain embodiments disclosed herein are discussed in connection with client modules executing within browser software of a client system, it will be appreciated that the disclosed embodiments may be further employed in connection with any other type of software and/or execution environments and in a variety of configurations and/or architectures.

(10) In certain embodiments, the systems and methods described herein can, for example, be used in connection with digital rights management ("DRM") technologies such as those described in commonly assigned, co-pending U.S. patent application Ser. No. 11/583,693 ("the '693 application"), service orchestration technologies such as those described in commonly assigned co-pending U.S. patent application Ser. No. 10/863,551 ("the '551 application"), and/or content delivery technologies such as those described in commonly assigned co-pending U.S. patent application Ser. No. 12/785,406 ("the '406 application") (the contents of '693 application, the '551 application, and the '406 application hereby being incorporated by reference in their entireties), as well as in other contexts. It will be appreciated that these systems and methods are novel, as are many of the components, systems, and methods employed therein.

(11) FIG. **1** illustrates a trust configuration process consistent with embodiments disclosed herein. In certain embodiments, a trusted service **100** may interact with a server system **102** and/or a client system **104** in connection with a trust configuration process. The trusted service **100**, the server system **102**, the client system **104**, and/or other services and/or systems (not shown) used in connection with the disclosed embodiments 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 trusted service **100**, the server system **102**, the client system **104**, and/or other systems 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 trusted service **100**, the server system **102**, the client system **104**, and/or other systems or services may further comprise a secure processing unit ("SPU") configured to perform sensitive operations such as trusted credential and/or key management, secure policy management, and/or other aspects of the systems and methods disclosed herein. The trusted service **100**, the server system **102**, and/or the client system **104**, and/or other services or systems may further comprise software and/or hardware configured to enable electronic communication of information between the devices and/or systems via one or

more associated network connections.

(12) The client system **104** may comprise a computing device executing one or more applications configured to implement certain embodiments of the systems and methods disclosed herein. In certain embodiments, the client system **104** may comprise a laptop computer system, a desktop computer system, a smartphone, a tablet computer, and/or any other computing system and/or device that may be utilized in connection with the disclosed systems and methods. In some embodiments, the client system **104** may comprise software and/or hardware configured to, among other things, implement cryptographic operations using software executing within browser software of the client system **104** and/or a sandboxed environment associated with the same. As discussed in more detail below, in some embodiments, such cryptographic functionality may be implemented using, at least in part, one or more protected applications (e.g., client modules) executing within a browser of the client system **104**.

(13) Communication between the client system **104**, server system **102**, trusted service **100**, and/or one or more other service providers may be facilitated by 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 utilize any suitable communication protocols and/or standards facilitating communication between the connected devices and systems. The network connections 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 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 utilize IEEE's 802.11 standards, Bluetooth®, ultra-wide band ("UWB"), Zigbee®, and or any other suitable communication protocol(s).

(14) As part of the trust configuration process, the trusted service **100** may generate and issue one or more trusted credentials to the server system **102** and/or the client system **104**. In certain embodiments, a trusted credential may comprise a certificate including one or more cryptographic components. For example, as illustrated, the trusted service **100** may generate an encryption certificate **108** that includes a cryptographic key using a certificate generation module **106**. In some embodiments, the cryptographic key included in the encryption certificate **108** may be a public cryptographic key. Although embodiments disclosed herein are discussed in connection with a trusted credential that comprises an encryption certificate **108** including a public key, it will be appreciated that any suitable trusted credential may be used in connection with the disclosed embodiments. As discussed in more detail below, the encryption certificate **108** may be used in connection with facilitating trust between a server system **102**, a client system **104**, and/or a module executing on the same (e.g., a script executing in browser software and/or the like)

(15) The encryption certificate **108** may be communicated to the server system **102** and/or the client system **104**. In certain embodiments, the encryption certificate **108** may be issued to the server system **102** from the trusted service **100**, which may in turn issue the encryption certificate **108** to the client system **104**. In other embodiments, the encryption certificate **108** may be issued to the client system **104** by the trusted service **100** directly and/or via one or more other intermediate services and/or systems. Similarly, the encryption certificate **108** may be issued to the server system **102** via one or more intermediate services and/or systems.

(16) Upon receiving the encryption certificate **108**, the server system **102** may use the encryption

certificate **108** in connection with a server trust configuration process **110**. In certain embodiments, the encryption certificate **108** may be loaded in a trusted certificate library maintained by the server system **102** as part of the server trust configuration process **110** for use in connection with future trusted and/or cryptographic operations. In certain embodiments, prior to committing secret information to the trusted service **100**, the server system **102** may verify that the trusted service **100** is trusted as part of the server trust configuration process **110**. Trust may be verified and/or otherwise established in a variety of suitable ways. For example, trust may be verified and/or otherwise established by determining that the trusted service **100** meets trust and/or security requirements articulated by the server system **102** and/or another service. After trust has been verified and/or established, the encryption certificate **108** may be used by the server system **102** to encrypt information in connection with the disclosed embodiments.

(17) The server system **102** may generate a key wrap module **112** as part of a key wrap module generation process **112**. In certain embodiments, the key wrap module **112** may use the encryption certificate **108** to perform a key wrapping operation. In some embodiments, the key wrap module **114** may be used to encapsulate cryptographic keys, thereby protecting wrapped cryptographic keys while in untrusted storage and/or during transmission over untrusted communication channels. In certain embodiments, the key wrap module **114** may comprise an asymmetric encryption algorithm that utilizes the public key included in the encryption certificate **108** to wrap and/or otherwise protect a cryptographic key. In some embodiments, the key wrap module **114** may be configured to perform a key wrapping operation using an RSA encryption algorithm, although other suitable key wrapping algorithms and/or processes may also be used in connection with the disclosed embodiments. Although in the illustrated embodiments the key wrap module **112** is shown as being generated and distributed by the server system **102**, it will be appreciated that the key wrap module **112** may be generated and distributed by the trusted service **100** and/or one or more other services.

(18) In some embodiments, the encryption certificate **108** may be separate from the key wrap module **114** and may be accessed by the key wrap module **114** in connection with key wrapping operations (e.g., from a certificate library and/or the like). In other embodiments, the encryption certificate **108** may be included as a component within the key wrap module **114**.

(19) The key wrap module **114** and/or the encryption certificate **108** may be transmitted to the client system **104** for use in connection with key wrapping and/or other cryptographic operations. In certain embodiments, upon receiving the encryption certificate **108**, the client system **104** may use the encryption certificate **108** in connection with a client trust configuration process **116**. In certain embodiments, prior to committing secret information to the trusted service **100** and/or modules and/or applications protected by the same, the client system **104** may verify that the trusted service **100** is trusted as part of the client trust configuration process **116**. Trust may be verified and/or otherwise established in a variety of suitable ways. For example, trust may be verified and/or otherwise established by determining that the trusted service **100** meets trust and/or security requirements articulated by the client system **104** and/or another service. After trust has been verified and/or established, the encryption certificate **108** may be used by the client system **104** to encrypt information in connection with the disclosed embodiments.

(20) In certain embodiments, the encryption certificate **108** may be loaded in a trusted certificate library maintained by the client system **104** as part of the user trust configuration process **116** for use in connection with future trusted and/or cryptographic operations. Similarly, the key wrap module **114** may be loaded into one or more cryptographic libraries of the client system **104** for use in connection with trusted and/or cryptographic operations performed by the client system **104**.

(21) Various trust configuration processes illustrated in connection with FIG. **1** may be performed at a variety of suitable times. For example, in some embodiments, the server system **102** and/or client system **104** may be provisioned with an encryption certificate **108** and/or key wrap module **114** at a time of manufacture. In other embodiments, various aspects of the trust configuration processes may be performed in connection with a system registration process (e.g., a server and/or

client registration process or the like). In yet further embodiments, aspects of the trust configuration processes may be performed when a protected client module is transmitted to a client system **104** from the server system **102** for execution within browser software of the client system **104**, as is discussed in more detail below.

(22) FIG. **2** illustrates a process of provisioning a client system **104** with a protected client module **204** consistent with embodiments disclosed herein. In certain embodiments, the server system **102** may transmit a client module **200** to the trusted service **100** for protection. In some embodiments, the client module **200** may be generated by the server system **102**. In further embodiments, a developer of the client module **200** may transmit the client module **200** to the server system **102** and/or the trusted service **100** for distribution to one or more client systems **104** and/or for protection by the trusted service **100**.

(23) The client module **200** may comprise a software application configured to be executed by the client system **104**. For example, in some embodiments, the client module **200** may be configured to be executed within browser software of the client system. In some embodiments, the client module **200** may comprise a JavaScript application downloaded to a client system **102** in connection with accessing a webpage using associated browser software. Consistent with embodiments disclosed herein, the client module **200** may implement certain cryptographic operations (e.g., operations using cryptographic keys and/or the like), which may be protected by the trusted service **100**.

(24) Upon receipt of the client module **200**, the trusted service **100** may engage in a protected module generation process **202**. In certain embodiments, the protected module generation process **202** may utilize any suitable software code protection method to yield a protected client module **204** including, for example, white-box cryptographic protection methods, fully homomorphic encryption ("FHE") methods, software obfuscation methods, functional encryption methods, and/or the like. For example, in certain embodiments, white-box protection methods may allow secret information associated with the client module **200** (e.g., secure keys such as a private key) to remain encrypted and/or otherwise protected, even during execution of code associated with the client module **200**. In some embodiments, such white-box cryptographic methods may protect software code and/or associated secret information from being exposed in clear text during execution of the module. Among other things, this may allow execution and/or storage of the client module **202** on and/or in a memory of a client system **104** having an open architecture.

(25) In certain embodiments, the protected client module **204** may comprise a key unwrap module **206**. The key unwrap module **206** may be configured to unwrap a cryptographic key using a private cryptographic key **208** included in the protected client module **204** to yield a protected cryptographic key (e.g., a white-box protected cryptographic key). For example, in some embodiments, the key unwrap module **206** may receive a cryptographic key wrapped using the public cryptographic key included in the encryption certificate **108** generated and/or provisioned by the trusted service **100** as part of a trust configuration process. The key unwrap module **206** may utilize the private cryptographic key **208** included in the protected client module **204** corresponding to the public key to unwrap the wrapped cryptographic key and yield a protected cryptographic key that may then be used in connection with certain cryptographic operations performed by the protected client module **204**. In some embodiments, the private cryptographic key **208** may comprise a white-box protected cryptographic key.

(26) The protected client module **204** may be transmitted to the server system **102** by the trusted service **100** for distribution to one or more client systems **104**. The server system **102** may distribute the protected client module **204** to the client system **104** under a variety of circumstances. For example, in some embodiments, the protected client module **204** may comprise a protected JavaScript application. When browser software executing on the client system **104** accesses an associated webpage, the protected JavaScript application may be downloaded by the client system **104** from the server system **102** for execution within an environment associated with the browser software. In some embodiments, the protected JavaScript application may not be permanently

installed on the client system **104**, but be cached for use during a single and/or over a limited number of sessions and/or period of time.

(27) In some embodiments, the protected client module **204** may perform cryptographic operations utilizing a variety of cryptographic keys. For example, the protected client module **204** may perform a secure cryptographically-enforced transaction between the server system **102** and the client system **104** using keys associated with the server system **102** and/or the client system **104**. Similarly, the protected client module **204** may use keys associated with the server system **102** and/or the client system **104** in connection with various web analytics methods, network authentication methods (e.g., Kerberos network authentication protocol methods), financial transaction methods including EMV payments, and/or the like. It will be appreciated that embodiments disclosed herein may also be utilized in connection with protecting secret information used in various non-cryptographic operations performed by the protected client module **204**.

(28) In certain embodiments, the server system **102** may not trust the integrity and/or security of the client system **104**, and thus may not wish to provide its unencrypted keys to the client system **104** for use in connection with various operations including, for example, cryptographic operations. Accordingly, the server system **102** may use the key wrap module **114** to wrap a server key **210** using the public key included in the encryption certificate **108**. After the server key **210** has been wrapped, the wrapped server key **212** may be transmitted to the client system **104** for use in connection with various cryptographic operations performed by the protected client module **204**.

(29) FIG. **3** illustrates an exemplary implementation of a cryptographic service included in browser software **300** executing on a client system **104** consistent with embodiments disclosed herein. In certain embodiments, certain software code executing on the client system **104** may be sandboxed and/or otherwise executed in some type of a limited virtualized environment. Software code executing in a sandbox may be restricted to accessing and/or using a controlled set of resources, functions, and/or services associated with the client system. For example, a sandbox may allow software code executing therein to only access a certain subset of storage disk space and/or memory associated with the client system. Access to certain input/output channels, networking resources, processing resources, and/or the like may be similarly controlled and/or restricted.

(30) In certain embodiments, browser software **300** executing on the client system **104** may be sandboxed, thereby protecting the client system **104** from nefarious code executing within the browser software **300**. In certain embodiments, the browser software **300** may have access to defined sandboxed resources **302** of the client system **104**. The sandboxed resources **302** may comprise certain portions and/or locations of storage on the client system **104** that may store, for example, cached and/or otherwise stored website information **304**, cookies **306**, and/or trusted credentials such as keys **312**, **310-314**. The sandboxed resources **302** may further comprise certain processing resources, functions, services, and/or interfaces of the client system **104** accessible by the browser software **300** and/or modules executing in an environment associated with the same. In other embodiments, the sandboxed resources **302** may comprise a secure clock, a secure random number generator, a secure user interface, and/or the like.

(31) A protected client module **204** may be loaded into the execution environment of the browser software **300** to facilitate implementation of certain cryptographic services within the browser software **300** consistent with embodiments disclosed herein. In certain embodiments, the interaction between the browser software **300** and/or modules executing in the sandboxed environment of the browser software **300** may be managed by one or more articulated policies. For example, in certain embodiments, the protected client module **304** may only be allowed access to sandboxed browser resources **302** associated with the module **304** (e.g., web storage **304**, cookies **306**, and/or keys **212**, **310-314**). Similarly, in some embodiments, a same-origin policy may be enforced by the browser software **300** to allow a specific browser application to access cookies and/or other information generated by the application while restricting access to cookies and/or other information generated by other applications. In this manner, the specific resources included in

the sandboxed browser resources **302** may vary between various browser-executed applications.

(32) A user of the client system **104** may wish to use a cryptographic key **308** in connection with a cryptographic operation performed by the protected client module **204** executing in the sandboxed environment of the browser software **300**. The user, however, may not necessarily trust applications executing within the browser software **300**, and thus may not wish to provide their unencrypted user key **308** to the execution environment of the browser software **300**. Accordingly, the client system **104** may use a key wrap module **114** provisioned as part of the trust configuration process detailed above to wrap the user key **308** using the public key included in the provisioned encryption certificate **108**. In certain embodiments, by wrapping the user key **308** with the public key included in the encryption certificate **108**, the client system **104** may limit use of the user key **308** to trusted applications protected by a trusted service.

(33) In certain embodiments, the encryption certificate **108** may also comprise descriptions of various functions and/or permissions relating to how a key wrapped with information included in the encryption certificate **108** (e.g., the public key) will be used by the protected client module **104**. In some embodiments, such descriptive information may allow a user of the client system **104** to better determine what operations they are consigning their secret user key **308** to prior to wrapping the key using information included in the encryption certificate **108** and importing the wrapped key **310** into the execution environment associated with the browser software **300**. In some embodiments, functions and/or permissions relating to how a key is used by the protected client module **204** and/or other related modules and/or systems may be enforced by a key authorization module **318** included in the protected client module **204** and/or the browser software **300**.

(34) After the user key **308** has been wrapped, the resulting wrapped user key **310** may be imported to the execution environment of the browser software **300** for use in connection with cryptographic operations implemented by the protected client module **204**. In certain embodiments, the key unwrap module **206** included in the protected client module **204** may be configured to unwrap the wrapped user key **310** using the private cryptographic key **208** provisioned to the protected client module **204** by the trusted service to yield a protected user key **314**. In certain embodiments, the protected user key **314** may comprise a white-box protected user key. Accordingly, the protected user key **314** may be exposed outside the sandboxed environment of the browser software **300** (e.g., to a remote server system and/or the like) without compromising the security of the user key. The protected user key **314** may be used in connection with cryptographic operations performed by the protected client module **204** and/or by a remote server system.

(35) Certain cryptographic operations performed by the protected client module **204** may further use a key provided by a server system. For example, certain secure transactions may utilize both a user key and a server key. Accordingly, a wrapped server key **212** sent to the execution environment of the browser software **300** by a server system may further be unwrapped using the key unwrap module **206** included in the protected client module **204** to generate a protected server key **312**. Like the protected user key **314**, the protected server key **312** may comprise a white-box protected user key. Accordingly, the protected server key **312** may be exposed outside the server system (e.g., to the execution environment of the browser software **300**) without compromising the security of the server key.

(36) Key unwrapping consistent with the disclosed embodiments may be performed at a variety of times. For example, in some embodiments, key unwrapping may be performed using the key unwrap module **206** when the wrapped server key **212** and/or the wrapped user key **310** are initially loaded into the execution environment of the browser software **300**. In further embodiments, key unwrapping may be performed when keys used in connection with a cryptographic operation are requested by the protected client module **204**. In certain embodiments, unwrapped protected keys **312**, **314** may be utilized in connection with cryptographic operations performed using a cryptographic library **316** of the protected client module **204**. In yet further embodiments, key unwrapping may be included as part of platform and/or service communication methods (e.g., SSL

communication methods and/or the like).

(37) Although certain embodiments disclosed herein are discussed in connection with the use of a key wrap module **114** to protect various secret information including user and server keys (e.g., user key **308**), it will be appreciated that other protection mechanisms may also be used in connection with the disclosed embodiments. For example, in some embodiments, keys may be communicated to the protected client module **204** via a secure and/or otherwise authenticated communications channel. In some embodiments, the secure and/or authenticated communication channel may protect the keys from being exposed in the clear.

(38) A variety of cryptographic operations may be performed using the protected keys **312**, **314**. For example, the protected client module **204** may use the protected keys **312**, **314** in connection with a secure cryptographically-enforced transaction between a server system and the client system **104**. Other cryptographic operations using embodiments of the disclosed systems and methods include, without limitation, data encryption/decryption operations, cryptographic signing operations, cryptographic signature verification operations, and/or the like. In yet further embodiments, the protected keys **312**, **314** may be used in a variety of other operations that may not necessarily be cryptographic operations.

(39) In certain embodiments, the protected client module **204** may implement a safe application program interface ("API"). In certain embodiments, the safe API may expose certain defined functions that may not be used to compromise the integrity of secret information imported to the protected client module **204** and/or the associated browser software **300** such as secure keys (e.g., keys **212**, **310-314**). In certain embodiments, such defined functions may be articulated in metadata and/or other information associated with the keys. For example, in certain embodiments, functions such as unwrapping a wrapped user key **310** and making it directly available may not be exposed by the safe API. In this manner, the integrity of the imported secure user key **310** may be maintained by the protected client module **204**. Similarly, specific permitted uses of secure keys **212**, **310-314** by the protected client module **204**, the browser software **300**, and/or other software and/or systems may be enforced by implementation of a safe API, thereby preventing the keys **212**, **310-314** from being used in arbitrary ways.

(40) In some embodiments, a safe API may further utilize certain code verification techniques to ensure that the integrity of secret information imported into a protected client module **204** and/or associated browser software is maintained. For example, the safe API may validate the integrity of its inputs and/or perform certain authorization checks. In certain embodiments, the safe API may perform and/or otherwise verify certain key authorizations using any suitable technique (e.g., DRM license checks and/or the like). In some embodiments, this functionality may, among other things, reduce the potential of repeated or rogue activations from detrimentally impacting an overall trust ecosystem and/or its various participants.

(41) In further embodiments, a user of the client system **104** may trust their secure user keys **308** to a trusted service. In certain embodiments, the trusted service may be trusted by the user through assurance and/or auditing of certain security implementations. The trusted service may protect the secret user keys **308** using white-box cryptographic protection methods and may distribute protected user keys **314** to the client system **104**. Similarly, a server may trust its keys to the trusted service, which may protect the server keys using white-box cryptographic protection methods and may similarly distribute the protected server keys **312** to the server system and/or the client system. The protected client module **204** executing within the browser software **300** of the client system **104** may then access the protected keys **312**, **314** in connection with various operations including, for example, cryptographic operations. In certain embodiments, such an implementation may reduce the burden on the client system **104** and/or a server system associated with performing key wrapping and/or unwrapping operations.

(42) In certain embodiments, the browser software **300** executing on the client system **104** may use Google® Native Client ("NaCl") and/or any other suitable sandboxing technology to facilitate safe

execution of native code within an environment associated with the browser software **300**. In some embodiments, native code that is written in a defined manner may be executed within the sandboxed environment. In certain embodiments, the defined manner in which the native code is written may be analyzable and verifiable as being secure by the browser software **300** executing the code. For example, native code may be analyzed to determine that the code may only call certain permissible functions. In some embodiments, such an implementation may allow for executing of web-based applications (e.g., a protected client module **204**) at near-native speeds on the client system **104**.

(43) Embodiments of the disclosed systems and methods may be utilized in connection with a variety of applications. For example, in some embodiments, a protected client module **204** may be utilized in connection with implementing a Kerberos Ticket Granting Service ("TGS"). In some embodiments, various secure resources used in connection with the Kerberos TGS (e.g., a secure random number generator, secure storage, etc.) may be provided by the client system **104** via a trusted SPU and/or the like. In further embodiments, a protected client module **204** may be used in connection with implementing a EMV payment service to, among other things, protect a user's account information (e.g., credit card number) and/or sign payment tokens using associated keys (e.g., symmetric keys).

(44) It will be appreciated that a number of variations can be made to the architecture and relationships presented in connection with FIGS. **1**-**3** within the scope of the inventive body of work. For example, without limitation, in some embodiments, some or all of the functions performed by the trusted service **100** may be performed by the server system **102**. Similarly, some or all of the functions performed by the client system **104** may be performed by the server system **102**. Furthermore, one or more other services and/or systems not necessarily illustrated may be utilized in connection with implementing various aspects of the embodiments of the disclosed systems and methods. Although certain embodiments are discussed in connection with protecting secure cryptographic keys, it will be appreciated that the disclosed embodiments may be further used in connection with protecting any suitable secret information including, without limitation, passwords, personal information, sensitive data, and/or the like. Thus it will be appreciated that FIGS. **1**-**3** are provided for purposes of illustration and explanation, and not limitation.

(45) FIG. **4** illustrates a flow chart of an exemplary cryptographic method **400** consistent with embodiments disclosed herein. The illustrated method **400** 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 **400** may be implemented by a client system, a server system, a trusted service, and/or any other related service or system as described above. Certain embodiments included in the illustrated method **400** may implement secure cryptographic operations within a sandboxed execution space of browser software executing on a client system. It will be appreciated, however, that embodiments of the disclosed systems and methods may be similarly implemented in a variety of other execution spaces. In addition, it will be appreciated that cryptographic operations consistent with the disclosed embodiments may incorporate all of the elements of the illustrated method **400** or a subset thereof, and may proceed in any suitable order.

(46) At **402**, a wrapped user key may be received via an import process in an execution environment associated with browser software. In certain embodiments, the wrapped user key may have been wrapped using a public key included in an encryption certificate provisioned to a client system executing the browser software from a trusted service. A wrapped server key may be similarly received in the execution environments of the browser software at **404**. Like the wrapped user key, the wrapped server key may have been wrapped using a public key included in an encryption certificate provisioned to the server system from a trusted service. The wrapped keys may be provided to the execution environment associated with the browser software and/or a protected application executing within the same through a platform API, a cookie, user interaction (e.g., user input via a browser widget or the like), etc.

(47) The wrapped user key may be unwrapped by a component of a protected application such as a key unwrapping module executing within the execution environment of the browser software at **406**. In certain embodiments, the wrapped user key may be unwrapped using, at least in part, a private key provisioned to the protected application from the trusted service corresponding to the public key included in the encryption certificate used to wrap the user key. Unwrapping the wrapped user key may generate a white-box protected user key.

(48) At **408**, the wrapped server key may be unwrapped by the key unwrapping module of the protected application executing within the execution environment of the browser software. Like the wrapped user key, the wrapped server key may be unwrapped using, at least in part, a private key provisioned to the protected application from the trusted service corresponding to the public key included in the encryption certificate used to wrap the server key. Unwrapping the wrapped server key may generate a white-box protected server key.

(49) A cryptographic operation may be performed by the protected client module using the generated protected server and/or user keys at **410**. In certain embodiments, the security of the user and/or server keys may be maintained, as the corresponding protected cryptographic keys used in connection with the cryptographic operation may remain white-box protected in the execution space of the browser application and/or elsewhere that they may be sent in connection with the operation. Accordingly, a user of the client system and/or the server system may be more willing to import wrapped representations of their secure keys into the otherwise potentially untrusted execution space of the browser software.

(50) FIG. **5** illustrates a system **500** that may be used to implement certain embodiments of the systems and methods of the present disclosure. The system **500** may comprise a trusted service, a server system, a client system, and/or any other system configured to implement certain aspects the systems and methods described herein. In certain embodiments, the system **500** may perform some or all of the disclosed functions associated with a trusted service, a server system, a client system, and/or any other related system and/or service as disclosed herein.

(51) As illustrated in FIG. **5**, the system **500** may include: a processor **502**; system memory **504**, which may include high speed RAM, non-volatile memory and/or one or more bulk non-volatile computer-readable storage mediums (e.g., a hard disk, flash memory, etc.) for storing programs and other data for use and execution by the processor **502**; an interface **516** (e.g., an input/output interface) that may include a display and/or one or more input devices such as, for example, a touchscreen, a keyboard, a mouse, a track pad, and the like; a port **506** for interfacing with removable memory **508** that may include one more diskettes, optical storage mediums, and/or other computer-readable storage mediums (e.g., flash memory, thumb drives, USB dongles, compact discs, DVDs, etc.); a network interface **510** for communicating with other systems via a network **530** using one or more communication technologies; and one or more buses **532** for communicatively coupling the aforementioned elements.

(52) In certain embodiments, the network **530** 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 or the like). In some embodiments, the network interface **510** and/or network **530** may be part of a wireless carrier system, such as a PCS, and/or any other suitable communication system incorporating any suitable communication standards and/or protocols. In further embodiments, the network interface **510** and/or the network **530** may be part of an analog mobile communications network and/or a digital mobile communications network utilizing, for example, CDMA, GSM, FDMA, and/or TDMA standards. In still further embodiments, the network interface **510** and/or the network **530** may incorporate one or more satellite communication links and/or use IEEE's 802.11 standards, near-field communication, Bluetooth®, UWB, Zigbee®, and or any other suitable standard or standards.

(53) In some embodiments, the system **500** may, alternatively or in addition, include a SPU **514** that is protected from tampering by a user of system **500** or other entities by utilizing secure

physical and/or virtual security techniques. An SPU **514** can help enhance and/or facilitate the security of sensitive operations such as private management of secret or otherwise secure information including cryptographic keys, and other aspects of the systems and methods disclosed herein. In certain embodiments, the SPU **514** may operate in a logically secure processing domain and be configured to protect and operate on secret information. In some embodiments, the SPU **514** may include internal memory storing executable instructions or programs configured to enable to the SPU **514** to perform secure operations. In further embodiments, the SPU **514** may facilitate certain secure platform resources including, without limitation, secure random number generators, secure clocks, secure user interfaces, secure storage, and/or the like.

(54) The operation of system **500** may be generally controlled by the processor **502** operating by executing software instructions and programs stored in the system memory **504** (and/or other computer-readable media, such as removable memory **508**). The system memory **504** may store a variety of executable programs or modules for controlling the operation of the system **500**. For example, the system memory **504** may include an operating system ("OS") **520** 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 **522** for implementing trust and privacy management functionality including protection and/or management of secret information. The system memory **504** may further include, without limitation, communication software **524** configured to enable in part communication with and by the system **500**, applications **526**, cryptographic keys **528** (e.g., wrapped keys, protected keys, etc.), browser software and/or protected applications and/or client modules **518** configured to execute within an execution space associated with the same, and/or any other information and/or applications configured to implement embodiments of the systems and methods disclosed herein.

(55) One of ordinary skill in the art will appreciate that the systems and methods described herein can be practiced with computing devices similar or identical to that illustrated in FIG. **5**, or with virtually any other suitable computing device, including computing devices that do not possess some of the components shown in FIG. **5** and/or computing devices that possess other components that are not shown. Thus it should be appreciated that FIG. **5** is provided for purposes of illustration and not limitation.

(56) The systems and methods disclosed herein are not inherently related to any particular computer, electronic control unit, 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, a magnetic tape, a magnetic disk, flash memory, integrated circuits, or any other non-transitory digital processing apparatus memory device.

(57) 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 performed by a system comprising a processor and a non-transitory computer-readable storage medium storing instructions that, when executed, cause the system to perform the method, the method comprising: receiving a protected cryptographic client module and a wrapped cryptographic key; executing the protected cryptographic client module within a sandboxed execution environment of the system; importing the wrapped cryptographic key into the protected cryptographic client module within the sandboxed execution environment; exposing an interface of the protected cryptographic client module, the interface providing a set of one or more defined functions, the set of one or more defined functions being based, at least in part, on the wrapped cryptographic key; receiving, via the interface of the protected cryptographic client module from a first application executing outside the sandboxed execution environment, a request to perform at least one function of the set of one or more defined functions using the wrapped cryptographic key; performing, by the protected cryptographic client module within the sandboxed execution environment, the at least one function using the wrapped cryptographic key; and sending, by the protected cryptographic client module, a result of the at least one function to the first application.

2. The method of claim 1, wherein the set of one or more defined functions are identified in information associated with the wrapped cryptographic key.

3. The method of claim 2, wherein the information associated with the wrapped cryptographic key comprises key metadata.

4. The method of claim 1, wherein the sandboxed execution environment is associated with a software application executing on the system.

5. The method of claim 4, wherein the software application comprises a browser application.

6. The method of claim 1, wherein the protected cryptographic client module comprises a script application.

7. The method of claim 1, wherein the wrapped cryptographic key comprises an encrypted cryptographic key.

8. The method of claim 7, wherein the protected cryptographic client module stores a protected private key.

9. The method of claim 8, wherein the encrypted cryptographic key is decryptable using the protected private key.

10. The method of claim 9, wherein the protected private key comprises a private key protected with white-box cryptography.

11. The method of claim 1, wherein the at least one function comprises a decryption operation.

12. The method of claim 1, wherein the at least one function comprises an encryption operation.

13. The method of claim 1, wherein the at least one function comprises a cryptographic signature operation.

14. The method of claim 1, wherein the at least one function comprises a cryptographic signature verification operation.

15. The method of claim 1, wherein the wrapped cryptographic key comprises a wrapped server key.

16. The method of claim 1, wherein the wrapped cryptographic key comprises a wrapped user key.

17. The method of claim 1, wherein the method further comprises performing, by the protected cryptographic client module, a code verification check operation to confirm that the protected cryptographic client module has not been modified.

18. The method of claim 1, wherein the method further comprises performing, by the protected

cryptographic client module prior to importing the wrapped cryptographic key, an authorization check to determine that the wrapped cryptographic key is permitted to be imported into the protected cryptographic client module.