United States Patent	12395353
Kind Code	B2
Date of Patent	August 19, 2025
Inventor(s)	Keith, Jr.; Robert O. et al.

Authentication process with an exposed and unregistered public certificate

Abstract

Digital signatures using the Diophantine system of equations are implemented. A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. A digital signature scheme typically includes three algorithms: a key generation algorithm, a signing algorithm, and a signature verifying algorithm. The key generation algorithm selects a private key uniformly at random from a set of possible private keys. The key generation algorithm outputs the private key and a corresponding public key. The signing algorithm produces a signature given a message and a private key. The signature verifying algorithm either accepts or rejects a message's claim to authenticity based at least in part on the message, the public key, and the signature.

Inventors: **Keith, Jr.; Robert O. (San Jose, CA), Islamov; Rustam (South Lake Tahoe, CA), Akhiarov; Roustem (Menlo Park, CA), Silaev; Maxim (Windermere Park, AU)**

Applicant: **WINKK, INC** (Menlo Park, CA)

Family ID: **1000008764821**

Assignee: **Winkk, Inc. (Menlo Park, CA)**

Appl. No.: **18/371242**

Filed: **September 21, 2023**

Prior Publication Data

Document Identifier	Publication Date
US 20240113892 A1	Apr. 04, 2024

Related U.S. Application Data

us-provisional-application US 63408543 20220921

Publication Classification

Int. Cl.: **H04L9/32** (20060101); **H04L9/08** (20060101); **H04L9/30** (20060101)

U.S. Cl.:

CPC **H04L9/3247** (20130101); **H04L9/0825** (20130101); **H04L9/0861** (20130101); **H04L9/30** (20130101); **H04L9/3213** (20130101); **H04L9/3236** (20130101);

Field of Classification Search

CPC: H04L (9/3247); H04L (9/0825); H04L (9/0861); H04L (9/30); H04L (9/3213); H04L (9/3236); H04L (9/3093)

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
5581615	12/1995	Stern	N/A	N/A
5850444	12/1997	Rune	N/A	N/A
5966444	12/1998	Yuan	380/283	H04L 9/0838
5987130	12/1998	Chang	N/A	N/A
6446207	12/2001	Vanstone	713/180	H04L 9/3066
6895506	12/2004	Abu-Husein	N/A	N/A
6947943	12/2004	DeAnna	N/A	N/A
7100051	12/2005	Kipnis	713/168	H04L 9/3093
7167565	12/2006	Rajasekaran	N/A	N/A
7468927	12/2007	Battista	N/A	N/A
7571320	12/2008	Davis	N/A	N/A
D607009	12/2008	McEnaney	N/A	N/A
7683773	12/2009	Goodall	N/A	N/A
D614192	12/2009	Takani	N/A	N/A
7885635	12/2010	Laursen	N/A	N/A
7925013	12/2010	Washington	N/A	N/A
7992102	12/2010	De Angelo	N/A	N/A
7992190	12/2010	Mevissen	N/A	N/A
8139581	12/2011	Mraz	N/A	N/A
8161463	12/2011	Johnson	N/A	N/A
8218762	12/2011	Itoh	N/A	N/A
8363259	12/2012	Gillboa	N/A	N/A
8417642	12/2012	Oren	N/A	N/A
8417643	12/2012	Mardikar	N/A	N/A
8543834	12/2012	Barra	N/A	N/A
8543884	12/2012	Mansour	N/A	N/A
8621209	12/2012	Johansson	N/A	N/A
8639785	12/2013	Kiley	N/A	N/A
8892871	12/2013	Cho	N/A	N/A
D719176	12/2013	Cohen	N/A	N/A
D719177	12/2013	Cohen	N/A	N/A
D723050	12/2014	Minsung et al.	N/A	N/A
8959579	12/2014	Barton	N/A	N/A
9112835	12/2014	Izozaki	N/A	N/A
9210156	12/2014	Little	N/A	N/A
9219732	12/2014	Baghdassaryan	N/A	N/A
9225695	12/2014	Riera	N/A	N/A
9350539	12/2015	Veugen	N/A	N/A
9392460	12/2015	Blake	N/A	N/A
9419951	12/2015	Felsher et al.	N/A	N/A
D765669	12/2015	Shaw	N/A	N/A
9485237	12/2015	Johansson	N/A	N/A
9615066	12/2016	Tran	N/A	N/A
9665169	12/2016	Dai	N/A	N/A
9706406	12/2016	Adams	N/A	N/A
9721080	12/2016	Moran	N/A	N/A

D800764	12/2016	Thoreson	N/A	N/A
9854218	12/2016	Mardikar	N/A	N/A
D813884	12/2017	Penker	N/A	N/A
9959694	12/2017	Lindsay	N/A	N/A
9961547	12/2017	Molina-Markham	N/A	N/A
10019561	12/2017	Shelton	N/A	N/A
10200364	12/2018	Ketharaju et al.	N/A	N/A
10257229	12/2018	Kuo	N/A	N/A
D847857	12/2018	Elatta	N/A	N/A
10374800	12/2018	Sharfi	N/A	N/A
10380333	12/2018	Moran	N/A	N/A
10402800	12/2018	Lucas	N/A	N/A
10404458	12/2018	Yamada	N/A	N/A
10430789	12/2018	Herald, Jr.	N/A	N/A
10432605	12/2018	Lester	N/A	N/A
10437975	12/2018	Shelton	N/A	N/A
10521223	12/2018	Bogushefsky, III	N/A	N/A
10559307	12/2019	Khalegi	N/A	N/A
10630467	12/2019	Gilbert	N/A	N/A
10674446	12/2019	Trent	N/A	N/A
10762406	12/2019	Cash	N/A	N/A
10769633	12/2019	Dua	N/A	N/A
10810290	12/2019	Minter et al.	N/A	N/A
10867021	12/2019	Shelton	N/A	N/A
10887307	12/2020	Newstadt	N/A	N/A
10911425	12/2020	Hitchcock	N/A	N/A
10936744	12/2020	Trepetin	N/A	N/A
10958424	12/2020	Chhabra	N/A	N/A
D916890	12/2020	Nagpal	N/A	N/A
10970607	12/2020	Xue	N/A	N/A
11005839	12/2020	Shahidzadeh	N/A	N/A
11030618	12/2020	Budko	N/A	N/A
11038694	12/2020	Kleinman	N/A	N/A
D925602	12/2020	Xu	N/A	N/A
D928803	12/2020	Faller	N/A	N/A
D928820	12/2020	Bodduluri	N/A	N/A
11121878	12/2020	McCarty	N/A	N/A
D942469	12/2021	Abdullah et al.	N/A	N/A
11283835	12/2021	Gordon	N/A	N/A
11328042	12/2021	Keith, Jr.	N/A	N/A
11510172	12/2021	Feng	N/A	N/A
11553337	12/2022	Keith, Jr	N/A	N/A
11563582	12/2022	Keith, Jr	N/A	N/A
11574045	12/2022	Keith, Jr.	N/A	N/A
11588794	12/2022	Keith, Jr	N/A	N/A
11637694	12/2022	Islamov	N/A	N/A
11652815	12/2022	Keith, Jr.	N/A	N/A
11657140	12/2022	Keith, Jr.	N/A	N/A
2002/0099955	12/2001	Peled et al.	N/A	N/A
2002/0114454	12/2001	Hamilton	N/A	N/A
2002/0131592	12/2001	Hinnant	N/A	N/A
2002/0169871	12/2001	Cravo de Almeida	N/A	N/A
2002/0186688	12/2001	Inoue	N/A	N/A
2003/0014750	12/2002	Kamen	N/A	N/A
2003/0016844	12/2002	Numaoka	N/A	N/A
2003/0021416	12/2002	Brown	N/A	N/A
2003/0147267	12/2002	Huttunen	N/A	N/A

2003/0174067	12/2002	Soliman	N/A	N/A
2003/0221030	12/2002	Pontius	N/A	N/A
2004/0151309	12/2003	Gentry	713/176	H04L 9/3093
2004/0198392	12/2003	Harvey	N/A	N/A
2004/0223616	12/2003	Kocarev	N/A	N/A
2005/0084114	12/2004	Jung	N/A	N/A
2005/0135609	12/2004	Lee	N/A	N/A
2005/0147240	12/2004	Agrawal	N/A	N/A
2005/0210260	12/2004	Venkatesan	713/180	H04L 9/0643
2006/0031301	12/2005	Herz et al.	N/A	N/A
2006/0075060	12/2005	Clark	N/A	N/A
2006/0196950	12/2005	Killccote	N/A	N/A
2006/0210067	12/2005	Vedula	380/28	H04L 9/3271
2006/0236408	12/2005	Yan	N/A	N/A
2006/0285544	12/2005	Taylor	N/A	N/A
2007/0086653	12/2006	Javidi	N/A	N/A
2007/0185718	12/2006	DiMambro	N/A	N/A
2008/0022141	12/2007	Hammarlund	N/A	N/A
2008/0031460	12/2007	Brookner	N/A	N/A
2008/0045218	12/2007	Okochi	N/A	N/A
2008/0084836	12/2007	Baird	N/A	N/A
2008/0165937	12/2007	Moore	N/A	N/A
2008/0301057	12/2007	Oren	N/A	N/A
2009/0006796	12/2008	Chang	N/A	N/A
2009/0090577	12/2008	Takahashi	N/A	N/A
2009/0161873	12/2008	Simard	N/A	N/A
2009/0194592	12/2008	Ming et al.	N/A	N/A
2009/0279693	12/2008	Billet	N/A	N/A
2009/0315671	12/2008	Gocho	N/A	N/A
2009/0327746	12/2008	Greco	N/A	N/A
2010/0079591	12/2009	Lee	N/A	N/A
2010/0100716	12/2009	Scott et al.	N/A	N/A
2010/0122274	12/2009	Gillies	N/A	N/A
2010/0329232	12/2009	Tubb	N/A	N/A
2011/0072142	12/2010	Herz et al.	N/A	N/A
2011/0106935	12/2010	Srinivasan	N/A	N/A
2011/0167255	12/2010	Matzkel	N/A	N/A
2011/0167273	12/2010	Maas	N/A	N/A
2011/0187642	12/2010	Faith	N/A	N/A
2011/0194694	12/2010	Struik	N/A	N/A
2011/0231673	12/2010	Alekseev	N/A	N/A
2011/0233284	12/2010	Howard	N/A	N/A
2011/0276952	12/2010	Tyloch	N/A	N/A
2011/0302405	12/2010	Marlow	N/A	N/A
2011/0321052	12/2010	Long	N/A	N/A
2012/0047563	12/2011	Wheeler	N/A	N/A
2012/0098750	12/2011	Allen	N/A	N/A
2012/0185910	12/2011	Miettinen	N/A	N/A
2012/0214442	12/2011	Crawford	N/A	N/A
2012/0221859	12/2011	Marien	N/A	N/A
2012/0272058	12/2011	Wang et al.	N/A	N/A
2012/0281885	12/2011	Syrdal	N/A	N/A
2013/0086625	12/2012	Driscoll	N/A	N/A
2013/0111208	12/2012	Sabin et al.	N/A	N/A
2013/0170363	12/2012	Millington	N/A	N/A
2013/0177151	12/2012	Sella	N/A	N/A
2013/0185779	12/2012	Tamai	N/A	N/A

2013/0202104	12/2012	Ghouti	N/A	N/A
2013/0205410	12/2012	Sambamurthy	N/A	N/A
2013/0239191	12/2012	Bostick	N/A	N/A
2013/0243187	12/2012	Hortsmeyer	N/A	N/A
2013/0304676	12/2012	Gupta	N/A	N/A
2013/0305324	12/2012	Alford, Jr.	N/A	N/A
2013/0326224	12/2012	Yavuz	713/176	H04L 9/3247
2013/0346023	12/2012	Novo	N/A	N/A
2014/0002481	12/2013	Broughton	N/A	N/A
2014/0007048	12/2013	Qureshi	N/A	N/A
2014/0013422	12/2013	Janus	N/A	N/A
2014/0038583	12/2013	Berg	N/A	N/A
2014/0039892	12/2013	Mills	N/A	N/A
2014/0040628	12/2013	Fort et al.	N/A	N/A
2014/0053261	12/2013	Gupta	N/A	N/A
2014/0064166	12/2013	HomChadhuri	N/A	N/A
2014/0098723	12/2013	Battista	N/A	N/A
2014/0108803	12/2013	Probert	N/A	N/A
2014/0201531	12/2013	Toy	N/A	N/A
2014/0215222	12/2013	Sakumoto	N/A	N/A
2014/0244514	12/2013	Rodriguez	N/A	N/A
2014/0244515	12/2013	Garfinkle	N/A	N/A
2014/0250496	12/2013	Amidon	N/A	N/A
2014/0278077	12/2013	Levin	N/A	N/A
2014/0304371	12/2013	Mraz	N/A	N/A
2014/0344455	12/2013	Cheng	N/A	N/A
2014/0351618	12/2013	Connell	N/A	N/A
2014/0368601	12/2013	deCharms	N/A	N/A
2015/0089568	12/2014	Sprague	N/A	N/A
2015/0095352	12/2014	Lacey	N/A	N/A
2015/0095580	12/2014	Liu	N/A	N/A
2015/0095648	12/2014	Nix	N/A	N/A
2015/0095986	12/2014	Karpey	N/A	N/A
2015/0103136	12/2014	Anderson	N/A	N/A
2015/0121524	12/2014	Fawaz	N/A	N/A
2015/0134963	12/2014	Izu	N/A	N/A
2015/0142666	12/2014	Landrok	N/A	N/A
2015/0223731	12/2014	Sahin	N/A	N/A
2015/0242601	12/2014	Griffiths	N/A	N/A
2015/0242605	12/2014	Du	N/A	N/A
2015/0258892	12/2014	Wu	N/A	N/A
2015/0262067	12/2014	Sridhara	N/A	N/A
2015/0271679	12/2014	Park	N/A	N/A
2015/0278805	12/2014	Spencer, III	N/A	N/A
2015/0280911	12/2014	Andoni	N/A	N/A
2015/0294092	12/2014	Balasubramanian	N/A	N/A
2015/0347734	12/2014	Beigi	N/A	N/A
2015/0350201	12/2014	Cornell	N/A	N/A
2015/0356289	12/2014	Brown	N/A	N/A
2015/0356462	12/2014	Fawaz	N/A	N/A
2015/0365229	12/2014	Patey	N/A	N/A
2015/0365235	12/2014	Hostyn	N/A	N/A
2015/0370826	12/2014	Mraz	N/A	N/A
2015/0373007	12/2014	Sheller	N/A	N/A
2015/0379238	12/2014	Connor	N/A	N/A
2016/0007288	12/2015	Samardzija	N/A	N/A
2016/0011224	12/2015	Pollack	N/A	N/A

2016/0055327	12/2015	Moran	N/A	N/A
2016/0057623	12/2015	Dutt	N/A	N/A
2016/0063492	12/2015	Kobres	N/A	N/A
2016/0065558	12/2015	Suresh	N/A	N/A
2016/0065570	12/2015	Spencer	N/A	N/A
2016/0098334	12/2015	Harihharakrishnan	N/A	N/A
2016/0103996	12/2015	Salajegheh	N/A	N/A
2016/0110528	12/2015	Gupta	N/A	N/A
2016/0117673	12/2015	Landrock	N/A	N/A
2016/0135107	12/2015	Hampel	N/A	N/A
2016/0148222	12/2015	Davar	N/A	N/A
2016/0180078	12/2015	Chhabra	N/A	N/A
2016/0182503	12/2015	Cheng	N/A	N/A
2016/0191499	12/2015	Momchillov	N/A	N/A
2016/0227411	12/2015	Lundblade	N/A	N/A
2016/0239649	12/2015	Zhao	N/A	N/A
2016/0239657	12/2015	Loughlin-McHugh et al.	N/A	N/A
2016/0253498	12/2015	Valencia	N/A	N/A
2016/0283406	12/2015	Linga	N/A	N/A
2016/0300049	12/2015	Guedalia	N/A	N/A
2016/0320831	12/2015	McCubbin	N/A	N/A
2016/0342873	12/2015	Winkk et al.	N/A	N/A
2016/0352696	12/2015	Essigmann	N/A	N/A
2017/0005995	12/2016	Yang	N/A	N/A
2017/0013453	12/2016	Lee	N/A	N/A
2017/0024660	12/2016	Chen	N/A	N/A
2017/0041309	12/2016	Ekambaram et al.	N/A	N/A
2017/0048062	12/2016	Polak	N/A	N/A
2017/0055878	12/2016	Chon	N/A	N/A
2017/0063528	12/2016	Seo	N/A	N/A
2017/0068994	12/2016	Slomkowski	N/A	N/A
2017/0070340	12/2016	Hibshoosh	N/A	N/A
2017/0070890	12/2016	Luff	N/A	N/A
2017/0085382	12/2016	Kamakari	N/A	N/A
2017/0104590	12/2016	Wang	N/A	H03M 13/05
2017/0124385	12/2016	Ganong	N/A	N/A
2017/0134372	12/2016	Dube	N/A	N/A
2017/0147345	12/2016	Clevenger	N/A	N/A
2017/0193211	12/2016	Blake	N/A	N/A
2017/0214529	12/2016	Oliveira	N/A	N/A
2017/0220407	12/2016	Estrada	N/A	N/A
2017/0230172	12/2016	Andersson	N/A	N/A
2017/0230344	12/2016	Dhar	N/A	N/A
2017/0264597	12/2016	Pizot	N/A	N/A
2017/0272419	12/2016	Kumar	N/A	N/A
2017/0287490	12/2016	Biswal	N/A	N/A
2017/0289168	12/2016	Bar	N/A	N/A
2017/0295010	12/2016	Shibutani	N/A	N/A
2017/0310479	12/2016	Sato	N/A	N/A
2017/0311250	12/2016	Rico Alvarino	N/A	N/A
2017/0317823	12/2016	Gandhi	N/A	N/A
2017/0339118	12/2016	Hwang	N/A	N/A
2017/0366514	12/2016	Malka	N/A	N/A
2018/0005239	12/2017	Schlesinger	N/A	N/A
2018/0005465	12/2017	Truong	N/A	N/A
2018/0007530	12/2017	Tanaka	N/A	N/A
2018/0012003	12/2017	Asulin	N/A	N/A

2018/0025135	12/2017	Odom	N/A	N/A
2018/0027411	12/2017	Taneja	N/A	N/A
2018/0029560	12/2017	Mohaupt	N/A	N/A
2018/0039990	12/2017	Lindermann	N/A	N/A
2018/0046803	12/2017	Li	N/A	N/A
2018/0063784	12/2017	Abraham	N/A	N/A
2018/0109696	12/2017	Thanigasalam	N/A	N/A
2018/0114221	12/2017	Karantzis	N/A	N/A
2018/0135815	12/2017	Rowles	N/A	N/A
2018/0144615	12/2017	Kinney	N/A	N/A
2018/0150622	12/2017	Zaitsev	N/A	N/A
2018/0167816	12/2017	Kusens et al.	N/A	N/A
2018/0176015	12/2017	Wang	N/A	H04L 9/0825
2018/0189160	12/2017	Yasin	N/A	N/A
2018/0189161	12/2017	Yasin	N/A	N/A
2018/0212770	12/2017	Costa	N/A	N/A
2018/0248865	12/2017	Johansson	N/A	N/A
2018/0285879	12/2017	Gadnis	N/A	N/A
2018/0302416	12/2017	Einberg	N/A	N/A
2018/0322266	12/2017	Kwok	N/A	N/A
2018/0329857	12/2017	Ko	N/A	N/A
2018/0375848	12/2017	Tunnell	N/A	N/A
2019/0021001	12/2018	Park	N/A	N/A
2019/0103957	12/2018	Isobe	N/A	N/A
2019/0122024	12/2018	Schwartz	N/A	N/A
2019/0133537	12/2018	Ghose	N/A	N/A
2019/0149333	12/2018	Harnik	N/A	N/A
2019/0188111	12/2018	Ozog	N/A	N/A
2019/0207918	12/2018	Kurian	N/A	N/A
2019/0220583	12/2018	Douglas	N/A	N/A
2019/0245704	12/2018	Pala	N/A	N/A
2019/0268774	12/2018	Kusens et al.	N/A	N/A
2019/0271349	12/2018	Madru	N/A	N/A
2019/0271578	12/2018	Moeller	N/A	N/A
2019/0272495	12/2018	Moeller	N/A	N/A
2019/0278895	12/2018	Streit	N/A	N/A
2019/0279204	12/2018	Norton	N/A	N/A
2019/0280868	12/2018	Streit	N/A	N/A
2019/0281025	12/2018	Harriman	N/A	N/A
2019/0281036	12/2018	Eisen	N/A	N/A
2019/0287427	12/2018	Schepers	N/A	N/A
2019/0289017	12/2018	Agarwal	N/A	N/A
2019/0318122	12/2018	Hockey	N/A	N/A
2019/0334708	12/2018	Carpor	N/A	N/A
2019/0342092	12/2018	Handschuh	N/A	N/A
2019/0354660	12/2018	Fong	N/A	N/A
2019/0354787	12/2018	Fong	N/A	N/A
2019/0370445	12/2018	Fong	N/A	N/A
2019/0386814	12/2018	Ahmed	N/A	N/A
2019/0387098	12/2018	McEnroe	N/A	N/A
2019/0391895	12/2018	Della Corte	N/A	N/A
2020/0014541	12/2019	Streit	N/A	N/A
2020/0029214	12/2019	Aylward	N/A	N/A
2020/0042723	12/2019	Krishnamoorthy	N/A	N/A
2020/0044852	12/2019	Streit	N/A	N/A
2020/0050745	12/2019	Kim	N/A	N/A
2020/0053096	12/2019	Bendersky	N/A	N/A

2020/0066071	12/2019	Budman	N/A	N/A
2020/0092111	12/2019	Anshel	N/A	H04L 63/0435
2020/0097643	12/2019	Uzun	N/A	N/A
2020/0099675	12/2019	Mardkis	N/A	N/A
2020/0100115	12/2019	Skaaksrud	N/A	N/A
2020/0120071	12/2019	Wimmer	N/A	N/A
2020/0125704	12/2019	Chavez	N/A	N/A
2020/0127974	12/2019	Moralndo	N/A	N/A
2020/0133373	12/2019	Huang	N/A	N/A
2020/0134145	12/2019	Bapst	N/A	N/A
2020/0152206	12/2019	Shen	N/A	N/A
2020/0162435	12/2019	Kubo	N/A	N/A
2020/0175157	12/2019	Wilding	N/A	N/A
2020/0193051	12/2019	Van Antwerp	N/A	N/A
2020/0242417	12/2019	Sagi	N/A	N/A
2020/0358611	12/2019	Hoang	N/A	N/A
2020/0358787	12/2019	Barker	N/A	N/A
2020/0387696	12/2019	Kushwah	N/A	N/A
2020/0403787	12/2019	Islam	N/A	N/A
2020/0403992	12/2019	Huffman	N/A	N/A
2021/0005224	12/2020	Rothschild	N/A	N/A
2021/0014314	12/2020	Yamada	N/A	N/A
2021/0049032	12/2020	White	N/A	N/A
2021/0051015	12/2020	Widmann	N/A	N/A
2021/0051177	12/2020	White	N/A	N/A
2021/0096826	12/2020	Duggal	N/A	N/A
2021/0123835	12/2020	Glennon	N/A	N/A
2021/0152417	12/2020	Baird	N/A	N/A
2021/0152554	12/2020	Taft	N/A	N/A
2021/0157291	12/2020	Uchizawa	N/A	N/A
2021/0167946	12/2020	Bitan	N/A	N/A
2021/0173906	12/2020	Keith, Jr	N/A	N/A
2021/0173907	12/2020	Keith, Jr.	N/A	N/A
2021/0173914	12/2020	Keith, Jr.	N/A	N/A
2021/0173915	12/2020	Keith, Jr.	N/A	N/A
2021/0173949	12/2020	Keith, Jr.	N/A	N/A
2021/0174333	12/2020	Keith, Jr.	N/A	N/A
2021/0176064	12/2020	Keith, Jr.	N/A	N/A
2021/0176066	12/2020	Keith, Jr.	N/A	N/A
2021/0176218	12/2020	Keith, Jr.	N/A	N/A
2021/0176223	12/2020	Falk	N/A	N/A
2021/0176235	12/2020	Keith, Jr.	N/A	N/A
2021/0176633	12/2020	Keith, Jr.	N/A	N/A
2021/0194608	12/2020	Yao	N/A	N/A
2021/0200852	12/2020	Gupta	N/A	N/A
2021/0248928	12/2020	Akiyama	N/A	N/A
2021/0250759	12/2020	Ziv	N/A	N/A
2021/0297258	12/2020	Keith, Jr.	N/A	N/A
2021/0297448	12/2020	Keith, Jr.	N/A	N/A
2021/0297455	12/2020	Keith, Jr.	N/A	N/A
2021/0350918	12/2020	Paul	N/A	N/A
2021/0362750	12/2020	Yang	N/A	N/A
2021/0390537	12/2020	Budko et al.	N/A	N/A
2022/0027439	12/2021	Greenberger	N/A	N/A
2022/0027447	12/2021	Keith, Jr.	N/A	N/A
2022/0028200	12/2021	Keith, Jr.	N/A	N/A
2022/0030022	12/2021	Keith, Jr.	N/A	N/A

2022/0036905	12/2021	Keith, Jr.	N/A	N/A
2022/0038895	12/2021	Keith, Jr	N/A	N/A
2022/0038897	12/2021	Liu	N/A	N/A
2022/0043913	12/2021	Keith, Jr.	N/A	N/A
2022/0045841	12/2021	Keith, Jr.	N/A	N/A
2022/0045865	12/2021	Mukherjee	N/A	H04L 9/3247
2022/0092161	12/2021	Keith, Jr.	N/A	N/A
2022/0092162	12/2021	Keith, Jr.	N/A	N/A
2022/0092163	12/2021	Keith, Jr.	N/A	N/A
2022/0092164	12/2021	Keith, Jr.	N/A	N/A
2022/0092165	12/2021	Keith, Jr.	N/A	N/A
2022/0093256	12/2021	Keith, Jr	N/A	N/A
2022/0094545	12/2021	Islamov et al.	N/A	N/A
2022/0094550	12/2021	Keith, Jr.	N/A	N/A
2022/0108026	12/2021	Ortiz et al.	N/A	N/A
2022/0130501	12/2021	Keith, Jr	N/A	N/A
2022/0138300	12/2021	Manjunath et al.	N/A	N/A
2022/0139546	12/2021	Manjunath et al.	N/A	N/A
2022/0164424	12/2021	Keith, Jr.	N/A	N/A
2022/0197985	12/2021	Keith, Jr.	N/A	N/A
2022/0200971	12/2021	Vigneswaran	N/A	N/A
2022/0229888	12/2021	Keith, Jr.	N/A	N/A
2022/0286966	12/2021	Zhao	N/A	N/A
2022/0337425	12/2021	Kim	N/A	H04L 9/3247
2022/0382844	12/2021	Keith, Jr.	N/A	N/A
2022/0385458	12/2021	Keith, Jr.	N/A	N/A
2022/0394023	12/2021	Keith, Jr.	N/A	N/A
2022/0394464	12/2021	Keith, Jr.	N/A	N/A
2022/0394465	12/2021	Keith, Jr.	N/A	N/A
2023/0096233	12/2022	Islamov et al.	N/A	N/A
2023/0106024	12/2022	Keith, Jr.	N/A	N/A
2023/0107624	12/2022	Keith, Jr.	N/A	N/A
2023/0114650	12/2022	Keith, Jr.	N/A	N/A
2023/0116527	12/2022	Keith, Jr.	N/A	N/A
2023/0185896	12/2022	Keith, Jr.	N/A	N/A
2023/0198766	12/2022	Keith, Jr.	N/A	N/A
2023/0198962	12/2022	Keith, Jr.	N/A	N/A
2023/0254120	12/2022	Islamov	N/A	N/A
2023/0254121	12/2022	Islamov	N/A	N/A
2023/0254122	12/2022	Islamov	N/A	N/A
2023/0267454	12/2022	Budko	N/A	N/A
2023/0283602	12/2022	Keith, Jr.	N/A	N/A
2023/0289431	12/2022	Keith, Jr.	N/A	N/A
2023/0291573	12/2022	Cheon	N/A	H04L 9/008

FOREIGN PATENT DOCUMENTS

Patent No.	**Application Date**	**Country**	**CPC**
107918790	12/2017	CN	N/A
107924475	12/2017	CN	N/A
106413128	12/2019	CN	N/A
3276561	12/2017	EP	N/A
3457344	12/2018	EP	N/A
WO2009060004	12/2008	WO	N/A
WO2009066004	12/2008	WO	N/A
WO-2014188336	12/2013	WO	H04L 9/3247
2016179433	12/2015	WO	N/A
2020065132	12/2019	WO	N/A

2020092542	12/2019	WO	N/A
2021119187	12/2020	WO	N/A
WO-2022172040	12/2021	WO	N/A

OTHER PUBLICATIONS

Maxrizal, M. “Public Key Cryptosystem Based on Singular Matrix”, 2022, Trends in Sciences. Nakhon Si Thammarat, Thailand, 19(3), p. 2147. (Year: 2022). cited by examiner

Tao et al., “Simple Matrix—A Multivariate Public Key Cryptosystem (MPKC) for Encryption” from Finite Field and Their Applications vol. 35, Sep. 2015, pp. 352-368 (Year 2015). cited by applicant

Erdem Alkim et al., “Post-Quantum key exchange—a new hope”, International Association For Cryptologic Research, vol. 20161116:063839, Nov. 16, 2016, pp. 1-22. cited by applicant

Joppe W. Bos et al., “Post-quantum key exchange for the TLS protocol from the ring learning with errors problem”, International Association for Cryptologic Research, vol. 20150316:235249, Mar. 17, 2015, pp. 1-28. cited by applicant

International Search Report mailed Aug. 11, 2016, for PCT Application No. PCT/US2016/031055, filed May 5, 2016, five pages. cited by applicant

International Search Report mailed Oct. 9, 2019, for PCT Application No. PCT/US2019/041871, filed Jul. 15, 2019, four pages. cited by applicant

Li et al., “Addressable Metasurfaces for Dynamic Holography and Optical Information Encryption”, Jun. 15, 2018, http://advances.sciencemag.org/content/advances/4/6/eaar6768.full.pdf. cited by applicant

The International Search Report and Written Report for the International Application No. PCT/US2020/064099 dated Mar. 16, 2021. cited by applicant

Bywater Films, “Winkk: Emotion to Action.” Vimeo, published Oct. 7, 2015 (Retrieved from the Internet Mar. 22, 2021). Internet URL: <https://vimeo.com/141695923> (Year: 2015). cited by applicant

Schiff, Eli, “Unofficial Apple Icon Design Awards.” Eli Schiff Blog, published Jan. 5, 2016 (Retrieved from the Internet Mar. 22, 2021), Internet URL: <www.elischiff.com/blog/2016/1/5/apple-icon-design-awards> (Year: 2016). cited by applicant

International Report on Patentability from International Application No. PCT/US2020/064099, mailed on Jun. 23, 2022, 7 pages. cited by applicant

Magoon, Owais, “iOS app.” Behance published Sep. 7, 2015 (Retrieved from the Internet Mar. 22, 2021). Internet URL: <https://www.behance.net/gallery/27383661/iOS-app> (Year: 2015). cited by applicant

---

Primary Examiner: Schwartz; Darren B

Attorney, Agent or Firm: Haverstock & Owens, A Law Corporation

---

Background/Summary

CROSS-REFERENCE TO RELATED APPLICATION(S) (1) This application claims priority under 35 U.S.C. § 119(e) of the U.S. Provisional Patent Application Ser. No. 63/408,543, filed Sep. 21, 2022 and titled, “DIOPHANTINE SYSTEM FOR DIGITAL SIGNATURES,” which is hereby incorporated by reference in its entirety for all purposes.

FIELD OF THE INVENTION

(1) The present invention relates to digital signatures. More specifically, the present invention relates to digital signatures using Diophantine systems.

BACKGROUND OF THE INVENTION

(2) A digital signature is a mathematical scheme for demonstrating the authenticity of digital messages or documents. The digital signature is a mathematical code that authenticates the document from the sender and ensures the document remains unaltered on reaching the recipient. Digital signatures employ asymmetric cryptography. Digital signatures can also provide non-repudiation, meaning that the signer cannot successfully claim that the signer did not sign a message, while also claiming their private key remains secret.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

(1) FIG. **1** illustrates an example networked environment for using digital signatures according to various examples described herein.

(2) FIG. **2** depicts a model of a digital signature scheme according to one or more embodiments.

(3) FIG. **3** illustrates an example message transfer process with authentication transaction identifications according to one or more embodiments.

(4) FIG. **4** illustrates a scenario for sending an authentication transaction value to a server and validating the delivery according to one or more embodiments.

(5) FIG. **5** illustrates a flowchart of a method of generating a digital signature for data according to some embodiments.

(6) FIG. **6** illustrates a flowchart of a method of verifying a digital signature for data according to some embodiments.

(7) FIG. **7** illustrates a flowchart of a method of implementing an authentication process with an exposed and unregistered public certificate according to some embodiments.

(8) FIG. **8** illustrates a block diagram of an exemplary computing device configured to implement the digital signature method according to some embodiments.

(9) FIG. **9** illustrates a diagram of an architecture of a system to secure endpoints across various network LAN and WAN infrastructures according to some embodiments.

(10) FIG. **10** illustrates a diagram of an architecture of a system to perform a 3-way key exchange according to some embodiments.

(11) FIG. **11** illustrates a diagram of an authentication server and endpoints according to some embodiments.

(12) FIG. **12** illustrates a diagram of an accepted connection and rejected connection according to some embodiments.

DETAILED DESCRIPTION

(13) The present disclosure relates to implementing digital signatures using the Diophantine system of equations. A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. A digital signature scheme typically includes three algorithms: a key generation algorithm, a signing algorithm, and a signature verifying algorithm. The key generation algorithm selects a private key uniformly at random from a set of possible private keys. The key generation algorithm outputs the private key and a corresponding public key. The signing algorithm produces a signature given a message and a private key. The signature verifying algorithm either accepts or rejects a message's claim to authenticity based at least in part on the message, the public key, and the signature.

(14) Digital signature schemes have two primary properties. First, the authenticity of a signature generated from a fixed message and fixed private key can be verified by using the corresponding public key. Second, it should be computationally infeasible to generate a valid signature for a party without knowing that party's private key.

(15) However, the fundamentals of digital signatures merit revision because of the computational properties of quantum computers. A digital signature should not have an algorithm for calculating a private key on an open key except for brute force. The algorithm for calculating the private key using a public key should be protected from parallel or sequential computations. The algorithm should be able to subsequently increase the complexity of calculations, for example, with increasing the length of the key.

(16) Various embodiments of the present disclosure introduce a digital signature scheme developed to be secure against standard and quantum computing. This digital signature scheme is based on the Diophantine system of equations (e.g., a polynomial equation with integer coefficients and a finite number of unknowns) and uniform distribution of random variables. The resistance against standard and quantum computing follows from the Hilbert's tenth problem: for any given Diophantine equation, the general algorithm (e.g., whether the equation has a solution with all unknowns taking integer values) does not exist. Also, the system of equation is cycled in respect to the parameters, thereby providing protection against parallel computation.

(17) Turning to the drawings, FIG. **1** illustrates an example networked environment **10** for using digital signatures according to various examples described herein. The networked environment **10** includes an authentication system **100**, a network **150**, and a number of computing devices **160**-**164** communicatively coupled to each other (and to the authentication system **100**) over the network **150**. The networked environment **10** is provided as a representative example of a system in which computing devices are capable of communicating data among each other. As described below, the authentication system **100** and the computing devices **160**-**164** can securely communicate data between each other to implement the digital signature scheme described herein. However, the concepts described herein can be applied to other networked computing environments, systems, and devices.

(18) The authentication system **100** can be embodied as one or more computing environments, computer systems, computing devices, or processing systems or devices. The authentication system **100** can include one or more computing devices arranged, for example, in one or more server or computer banks. The computing device or devices can be located at a single installation site or distributed among different geographical locations. The authentication system **100** can include a plurality of computing devices that together embody a hosted computing resource, a grid computing resource, or other distributed computing arrangement. In some cases, the authentication system **100** can be embodied as an elastic computing resource where an allotted capacity of processing, network, storage, or other computing-related resources varies over time. As further described below, the authentication system **100** can also be embodied, in part, as certain functional or logical (e.g., computer-readable instruction) elements or modules. Those elements can be executed to direct the authentication system **100** to act as an authentication or identity-verification system in the networked environment **10**, as described in further detail below.
(19) As also shown in FIG. **1**, the authentication system **100** includes a data store **120** and an application **130**. The data store **120** can be embodied as a memory, of any suitable type, and can be used to store data and data files, including sensitive or secret data, executable code, and other information. The application **130** is an example of one application program executable on the authentication system **100**. The authentication system **100** can host and execute any number of applications concurrently, as would be understood in the field of computing. As shown in FIG. **1**, the application **130** includes an authentication engine **132**. The operation of the authentication system **100**, including the application **130** and the authentication engine **132**, is described in greater detail below.
(20) The network **150** can include the Internet, intranets, extranets, wide area networks (WANs), local area networks (LANs), wired networks, wireless networks, cable networks, satellite networks, other suitable networks, or any combinations thereof. As one example, the authentication system **100** and the computing devices **160**-**164** can be respectively coupled to one or more public or private LANs or WANs and, in turn, to the Internet for communication of data among each other. Although not shown in FIG. **1**, the network **150** can also include communicative connections to any number and type of network hosts or devices, such as website servers, file servers, cloud computing resources, databases, data stores, or any other network or computing architectures.
(21) In the networked environment **10**, the authentication system **100** and the computing devices **160**-**164** can communicate data among each other using one or more network transfer protocols or interconnect frameworks, such as hypertext transfer protocol (HTTP), simple object access protocol (SOAP), representational state transfer (REST), real-time transport protocol (RTP), real time streaming protocol (RTSP), real time messaging protocol (RTMP), user datagram protocol (UDP), internet protocol (IP), transmission control protocol (TCP), other protocols and interconnect frameworks, and combinations thereof.
(22) As noted above, the authentication system **100** and the computing devices **160**-**164** can communicate data between each other over the network **150**. The concepts and processes described herein can be relied upon to exchange and verify messages between and among the authentication system **100** and the computing devices **160**-**164** over the network **150**.
(23) The computing devices **160**-**164** are representative of various types of computing devices, processing devices, and/or processor-based device or systems, including those in the form of a server computer, desktop computer, a laptop computer, a tablet computer, a personal digital assistant, a cellular telephone, a wearable computing device, a set-top box, and other example computing devices and systems. Each of the computing devices **160**-**164** can include one or more processors or processing devices, cryptographic trusted platform modules (TPMs), memory devices, local interfaces, various peripheral devices, and other components. The peripheral devices can include input or communications devices or modules, such as keyboards, keypads, touch pads, touch screens, microphones, cameras, network communications interfaces, wireless network communications modules (e.g., infra-red, WI-FI®, or BLUETOOTH®), buttons, switches, sensors, etc. The peripheral devices can also include a display, indicator lights, speakers, global positioning system (GPS) circuitry, accelerometers, gyroscopes, and other peripheral devices.
(24) As shown in FIG. **1**, the computing device **160** includes a data store **170** and an application **180**. The data store **170** can be embodied as any suitable type of memory and can be used to store data and data files, including data to be signed in plaintext or ciphertext forms, random numbers, executable code, and other information. In some cases, the data store **170** includes, at least in part, the memory of a TPM.
(25) The application **180** is an example of one application program executable on the computing device **160**. The computing device **160** can host and execute any number of applications concurrently, as would be understood in the field of computing. As one example, the application **180** can be embodied as a hypertext-based network browser, such as the Internet Explorer®, Firefox®, Chrome®, Safari®, or Silk® browsers, among other types of browsers. Additionally or alternatively, the application **180** can be embodied as an e-mail client, messaging

client, or other application(s) for other purpose(s). In any case, when executed on the computing device **160**, the application **180** can receive user input and data, process data, interpret and render various interfaces on display devices, and conduct other processes and tasks. As shown in FIG. **1**, the application **180** includes a cryptography engine **182** (also, "first engine **182**"), among other application submodules.

(26) The computing device **161** includes a data store **175** and an application **190**. The data store **175** can be embodied as any suitable type of memory and can be used to store data and data files, including sensitive or secret data, executable code, and other information. The application **190** is an example of one application program executable on the computing device **161**. The computing device **161** can host and execute any number of applications concurrently, as would be understood in the field of computing. As one example, the application **190** can be embodied as a hypertext-based network browser, such as the Internet Explorer®, Firefox®, Chrome®, Safari®, or Silk® browsers, among other types of browsers. Additionally or alternatively, the application **190** can be embodied as an e-mail client, messaging client, or other application(s) for other purpose(s). In any case, when executed on the computing device **161**, the application **190** can receive user input and data, process data, interpret and render various interfaces on display devices, and conduct other processes and tasks. As shown in FIG. **1**, the application **190** includes a cryptography engine **192** (also, "second engine **192**"), among other application submodules.

(27) FIG. **2** depicts a model of a digital signature scheme according to one or more embodiments. The digital signature scheme is based on public key cryptography. Each user adopting this scheme has a public-private key pair. Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key S and the public key P as the verification key.

(28) The signer feeds data to the hash function and generates a hash H of data. The hash value H, and the signer's random data R and signature key S are then fed to the signature algorithm, which produces the digital signature on the given hash. A signature is appended to the data and then both are sent to the verifier.

(29) The verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as its output. The verifier also runs the same hash function on the received data to generate a hash value. For verification, the hash value and the output of verification algorithm are compared. Based on the comparison result, the verifier determines whether the digital signature is valid. Since the digital signature is created by "private" key of signer and no one else can have this key, the signer cannot repudiate having signed the data in future.

(30) It should be noticed that instead of signing data directly by signing algorithm, a hash of data is created. Since the hash of data is a unique representation of data, it is sufficient to sign the hash in place of the original data. A reason for using the hash instead of the data directly for signing is efficiency of the scheme. Signing a large data object through modular exponentiation is computationally expensive and time consuming. The hash of the data is a relatively small digest of the data. Therefore, signing a hash is more efficient than signing the entire data.

(31) Consider the message M as a set of n.sub.m bytes, each including one of the American Standard Code for Information Interchange (ASCII) codes from 0 to 255, as follows:

M={m.sub.1,*m*.sub.2, . . . ,*m*.sub.n.sub.m},0≤*m*.sub.i≤255 (1)

(32) The private (secret) key integer number array S and random integer number array R are used to sign the message (digital signature).

S={s.sub.1,*s*.sub.2, . . . ,*s*.sub.n.sub.s},1≤*s*.sub.i≤2.sup.L (2)

R={r.sub.1,*r*.sub.2, . . . ,*r*.sub.n.sub.r},1≤*r*.sub.i≤2.sup.L (3)

(33) The public key integer number array P is used to check the digital signature.

P={p.sub.1,*p*.sub.2, . . . ,*p*.sub.n.sub.p},1<*p*.sub.i<2.sup.L (4)

(34) The initial message M is transferred into the hash H using the hash function F.

H=F(M) (5)

H={h.sub.1,*h*.sub.2, . . . ,*h*.sub.n.sub.h},0≤*h*.sub.i≤2.sup.L (6)

(35) For instance, in the case of using the hash function algorithm SHA512, n.sub.h=N=64, L=16.

(36) In various embodiments, the digital signature scheme uses the following matrix form for second order Diophantine equations:

(37) $X_1 \mathrm{AX}_2 = B$ (7) where X_1 = .Math. $\begin{matrix} x_{11} & x_{12} & \text{.Math.} & x_{1n} \\ x_{21} & x_{22} & \text{.Math.} & x_{2n} \\ \text{.Math.} & \text{.Math.} & \ddots & \text{.Math.} \\ x_{m1} & x_{m2} & \text{.Math.} & x_{\mathrm{mn}} \end{matrix}$ .Math. ,

X_2 = .Math. $\begin{matrix} x_{m+1,n+1} & x_{m+1,n+2} & \text{.Math.} & x_{m+1,2n} \\ x_{m+2,n+1} & x_{m+2,n+2} & \text{.Math.} & x_{m+2,2n} \\ \text{.Math.} & \text{.Math.} & \ddots & \text{.Math.} \\ \text{x2}_{m,n+1} & x_{2m,n+2} & \text{.Math.} & x_{2m,2n} \end{matrix}$ .Math. $x_{kl} \in Z, k = 1,2m, l = 1,2n$

A = .Math. $\begin{matrix} a_{11} & a_{12} & \text{.Math.} & a_{1n} \\ a_{21} & a_{22} & \text{.Math.} & a_{2n} \\ \text{.Math.} & \text{.Math.} & \ddots & \text{.Math.} \\ a_{m1} & a_{m2} & \text{.Math.} & a_{mn} \end{matrix}$ .Math. , B = .Math. $\begin{matrix} b_{11} & b_{12} & \text{.Math.} & b_{1n} \\ b_{21} & b_{22} & \text{.Math.} & b_{2n} \\ \text{.Math.} & \text{.Math.} & \ddots & \text{.Math.} \\ b_{m1} & b_{m2} & \text{.Math.} & b_{mn} \end{matrix}$ .Math.

$a_{ij}, b_{ij} \in Z, i = 1, m, j = 1, n$

(38) The singular matrices X.sup.(1), X.sup.(2) and regular matrices A.sup.(1), A.sup.(2) are the signer's private key.

(39) The singular matrices X and Y are used as a public key from the following relations:

X=X.sub.(1)X.sup.(2) (8)

Y.sup.(1)=A.sup.(1)X.sup.(1)

Y.sup.(2)=X.sup.(2)A.sup.(2)

Y=Y.sup.(1)Y.sup.(2)=A.sup.(1)XA.sup.(2) (9)

(40) Assume that the singular matrices H.sup.(1), H.sup.(2) represent the hash function of data. The regular matrices R.sup.(1), R.sup.(2) are randomly generated by the signer and form the matrices W.sup.(1), W.sup.(2) as follows:

W.sup.(1)=A.sup.(1)+R.sup.(1)

W.sup.(2)=A.sup.(2)+R.sup.(2) (10)

(41) The singular matrices Z.sup.(1), Z.sup.(2) are calculated as follows:

Z.sup.(1)=H.sup.(1)W.sup.(1)

Z.sup.(2)=W.sup.(2)H.sup.(2) (11)

(42) The signer uses the matrices Z.sup.(1), Z.sup.(2) and Y.sub.R as the signer's signature in the following relation:

H.sup.(1)(Y+Y.sub.R)H.sup.(2)=Z.sup.(1)XZ.sup.(2) (12)

where

Y.sub.R=R.sup.(1)XA.sup.(2)+A.sup.(1)XR.sup.(2)+R.sup.(1)XR.sup.(2)

(43) The verifier checks the equation (11) using public key X,Y and the matrices Z.sup.(1), Z.sup.(2) and Y.sub.R as the signer's signature.

(44) The resistance of the proposed algorithm is based on the Hilbert's tenth problem: for any given Diophantine equation, the general algorithm (whether the equation has a solution with all unknowns taking integer values) does not exist. Breaking the algorithm would involve solving one of the following tasks:

(45) First, to find unknown variables in matrices A.sup.(1), A.sup.(2) from Diophantine second order equation (9):

Y=A.sup.(1)XA.sup.(2)

(46) Second, to find unknown variables in matrices Z.sup.(1), Z.sup.(2) and Y.sub.R from Diophantine second order equation (12):

H.sup.(1)(Y+Y.sub.R)H.sup.(2)=.sup.Z(1)XZ.sup.(2)

(47) In accordance with the Hilbert's Tenth Problem, there is no algorithm except for the brute force for both of the two cases. The brute force algorithm is O(N) time complexity for a regular computer and O (√{square root over (N)}) for a quantum computer.

(48) The process of signing is next described. The secret key S, random array R, hash H and public key P are used in a matrix form in the signing algorithm. First, the hash H is prepared. The hash H can be presented in the form of matrices H.sub.n.sup.(2) and H.sub.n.sup.(1) of integer numbers 0≤h.sub.i.sup.(1,2)≤2.sup.2L as follows:

(49) $H_n^{(2)}$ = .Math. $\begin{matrix} h_{4n-3}^{(2)} & h_{4n-2}^{(2)} \\ h_{4n-1}^{(2)} & h_{4n}^{(2)} \end{matrix}$ .Math. (13)

$h_{4n-3}^{(2)} = h_n h_{n+2} h_{4n-2}^{(2)} = h_{n+1} h_{n+2} h_{4n-1}^{(2)} = h_n h_{n+3} h_{4n}^{(2)} = h_{n+1} h_{n+3}$ (14)

$H_n^{(1)}$ = .Math. $\begin{matrix} h_{4n-3}^{(1)} & h_{4n-2}^{(1)} \\ h_{4n-1}^{(1)} - & h_{4n}^{(1)} \end{matrix}$ .Math. (15) $h_k^{(1)} = h_{4n-3}^{(2)} h_{k+1}^{(1)} = h_{4n-3}^{(2)} h_{k+2}^{(1)} = h_{4n-2}^{(2)} h_{k+3}^{(1)} = h_{4n}^{(2)}$ (16)

$k = (4n + 1) \bmod 4N, n = 1, N$ (17)

(50) Next, the secret key is prepared. The array D is used for forming array X so that so that the matrices X.sub.n.sup.(1) and X.sub.n.sup.(2) are singular.

(51) $D = \{d_n^{(1)}, d_n^{(2)}\}_{n=1}^{4N}, 1 < d_i^{(1)}, d_i^{(2)} < 2^L$ (18) $X = \{x_n^{(1)}, x_n^{(2)}\}_{n=1}^{4N}, 1 < x_i^{(1)}, x_i^{(2)} < 2^L$ (19)

$X_n^{(i)} =$ .Math. $\begin{matrix} x_{4n-3}^{(1)} & x_{4n-2}^{(1)} \\ x_{4n-1}^{(1)} & x_{4n}^{(1)} \end{matrix}$ .Math. , $i = 1, 2, n = 1, N$ (20)

$x_{4n-3}^{(i)} = d_{4n-3}^{(i)} d_{4n-1}^{(i)} x_{4n-2}^{(i)} = d_{4n-2}^{(i)} d_{4n-1}^{(i)} x_{4n-1}^{(i)} = d_{4n-3}^{(i)} d_{4n}^{(i)} x_{4n}^{(i)} = d_{4n-2}^{(i)} d_{4n}^{(i)}$ (21)

(52) The condition of singularity is fulfilled automatically due to the following equalities:

(53) $x_{4n}^{(i)} = \frac{x_{4n-2}^{(i)} x_{4n-1}^{(i)}}{x_{4n-3}^{(i)}}, n = 1, N, i = 1, 2$ (22)

(54) The secret key S with the length n p=7N comprises the areas of X and A.

S={*s*.sub.n}.sub.n=1.sup.7N (23)

A={*a*.sub.n}.sub.n=1.sup.3N (24)

(55) Accordingly, it can be seen that:

s.sub.n=*a*.sub.n,*n*=1,3*N*

s.sub.n=*x*.sub.n.sup.(1),*n*=3*N*+1,7*N*

s.sub.i=*x*.sub.i.sup.(2),*i*=3*N*+1,7*N* (25)

(56) Next, a public key is prepared. The matrix X.sub.n is calculated as follows:

(57) $X_n = X_n^{(1)} X_n^{(2)}, n = 1, N$ (26) $X_n =$ .Math. $\begin{matrix} x_{4n-3} & x_{4n-2} \\ x_{4n-1} & x_{4n} \end{matrix}$ .Math. , $n = 1, N$ (27)

(58) The array of random numbers R is used for forming the array W and corresponding matrices W.sub.n.sup.(1) and W.sub.n.sup.(2) (n=1, N).

(59) $R = \{r_n\}_{n=1}^{3N}, 0 < r_i < 2^L$ (28) $W = \{w_1^{(1)}, w_2^{(2)}\}_{n=1}^{4N}, 0 < w_i^{(1)}, w_i^{(2)} < 2^L$ (29)

$W_n^{(i)} =$ .Math. $\begin{matrix} w_{4n-3}^{(i)} & w_{4n-2}^{(i)} \\ w_{4n-1}^{(i)} & w_{4n}^{(i)} \end{matrix}$ .Math. , $i = 1, 2$ (30)

.Math. $\begin{matrix} w_{4n-3}^{(1)} & w_{4n-2}^{(1)} \\ w_{4n-1}^{(1)} & w_{4n}^{(1)} \end{matrix}$ .Math. = .Math. $\begin{matrix} a_{3n-2} & a_{3n-1} \\ r_{3n-2} & r_{3n-1} \end{matrix}$ .Math. , $n = 1, N$ (31)

.Math. $\begin{matrix} w_{4n-3}^{(2)} & w_{4n-2}^{(2)} \\ w_{4n-1}^{(2)} & w_{4n}^{(2)} \end{matrix}$ .Math. = .Math. $\begin{matrix} a_{3n} & r_{3n} \\ a_{3n+1} & r_{3n+1} \end{matrix}$ .Math. , $n = 1, N - 1$ (32)

.Math. $\begin{matrix} w_{4N-3}^{(2)} & w_{4N-2}^{(2)} \\ w_{4N-1}^{(2)} & w_{4N}^{(2)} \end{matrix}$ .Math. = .Math. $\begin{matrix} a_{3N} & r_{3N} \\ a_1 & r_1 \end{matrix}$ .Math. (33)

(60) The matrices Y.sub.n.sup.(1) and Y.sub.n.sup.(2) (n=1, N) as given below are the result of the calculations in equations (35) through (38).

(61) $Y_n^{(i)} =$ .Math. $\begin{matrix} y_{4n-3}^{(i)} & y_{4n-2}^{(i)} \\ y_{4n-1}^{(i)} & y_{4n}^{(i)} \end{matrix}$ .Math. , $i = 1, 2$ (34) $Y_n^{(1)} = W_n^{(1)} X_n^{(1)}$ (35)

.Math. $\begin{matrix} y_{4n-3}^{(i)} & y_{4n-2}^{(i)} \\ y_{4n-1}^{(i)} & y_{4n}^{(i)} \end{matrix}$ .Math. = .Math. $\begin{matrix} a_{3n-2} & a_{3n-1} \\ r_{3n-2} & r_{3n-1} \end{matrix}$ .Math. .Math. $\begin{matrix} x_{4n-3}^{(i)} & x_{4n-2}^{(i)} \\ x_{4n-1}^{(i)} & x_{4n}^{(i)} \end{matrix}$ .Math. (36)

$Y_n^{(2)} = X_n^{(2)} W_n^{(2)}$ (37)

.Math. $\begin{matrix} y_{4n-3}^{(2)} & y_{4n-2}^{(2)} \\ y_{4n-1}^{(2)} & y_{4n}^{(2)} \end{matrix}$ .Math. = .Math. $\begin{matrix} x_{4n-3}^{(2)} & x_{4n-2}^{(2)} \\ x_{4n-1}^{(2)} & x_{4n}^{(2)} \end{matrix}$ .Math. .Math. $\begin{matrix} a_{3n} & r_{3n} \\ a_{3n+1} & r_{3n+1} \end{matrix}$ .Math. , $n = 1, N - 1$ (38)

.Math. $\begin{matrix} y_{4N-3}^{(2)} & y_{4N-2}^{(2)} \\ y_{4N-1}^{(2)} & y_{4N}^{(2)} \end{matrix}$ .Math. = .Math. $\begin{matrix} x_{4N-3}^{(2)} & x_{4N-2}^{(2)} \\ x_{4N-1}^{(2)} & x_{4N}^{(2)} \end{matrix}$ .Math. .Math. $\begin{matrix} a_{3N} & r_{3N} \\ a_1 & r_1 \end{matrix}$ .Math.

(62) The matrix Y n is obtained as follows:

(63) $Y_n = Y_n^{(1)} Y_n^{(2)}, n = 1, N$ (39) $Y_n =$ .Math. $\begin{matrix} y_{4n-3} & y_{4n-2} \\ y_{4n-1} & y_{4n} \end{matrix}$ .Math. , $n = 1, N$ (40)

(64) The public key P with the length n.sub.s=5N comprises the calculation result using arrays X and A. For n=1, N−1:

$$\begin{aligned} p_n = y_{4n-3} &= (a_{3n-2} x_{4n-3}^{(1)} + a_{3n-1} x_{4n-1}^{(1)} (a_{3n} x_{4n-3}^{(2)} \\ &\quad + a_{3n} x_{4n-2}^{(2)}) \\ &\quad + (a_{3n-2} x_{4n-2}^{(1)} + a_{3n-1} x_{4n}^{(1)} (a_{3n} x_{4n-1}^{(2)} \\ &\quad + a_{3n+1} x_{4n}^{(2)}) \end{aligned} \quad (65) \quad (41)$$

$$\begin{aligned} p_N = y_{4N-3} &= (a_{3N-2} x_{4N-3}^{(1)} + a_{3N-1} x_{4N-1}^{(1)} (a_{3N} x_{4N-3}^{(2)} \\ &\quad + a_1 x_{4N-2}^{(2)}) \\ &\quad + (a_{3N-2} x_{4N-2}^{(1)} + a_{3N-1} x_{4N}^{(1)} (a_{3N} x_{4N-1}^{(2)} \\ &\quad + a_1 x_{4N}^{(2)}) \end{aligned}$$

(66) For n=1, N:

p.sub.4n+N−3=*x*.sub.4n−3=*x*.sub.4n−3.sup.(1)*x*.sub.4n−3.sup.(2)+*x*.sub.4n−2.sup.(1)*x*.sub.4n−1.sup.(2)
p.sub.4n+N−2=*x*.sub.4n−2=*x*.sub.4n−3.sup.(1)*x*.sub.4n−2.sup.(2)+*x*.sub.4n−2.sup.(1)*x*.sub.4n.sup.(2)
p.sub.4n+N−1=*x*.sub.4n−1=*x*.sub.4n−1.sup.(1)*x*.sub.4n−3.sup.(2)+*x*.sub.4n.sup.(1)*x*.sub.4n−1.sup.(2)
p.sub.4n+N=*x*.sub.4n=*x*.sub.4n−1.sup.(1)*x*.sub.4n−2.sup.(2)+*x*.sub.4n.sup.(1)*x*.sub.4n.sup.(2) (42)

(67) Note that y.sub.4n−2, y.sub.4n−2, y.sub.4n−2 are not included into the public key P because they include random parameters r.sub.i.

(68) The signing process is next described. The matrices Z.sub.n.sup.(1) and Z.sub.n.sup.(2) (n=1, N) as shown in equations (43) and (44) are result of the calculations in equations (45) and (46).

(69) 0 $Z_n^{(1)} =$ .Math. $\begin{matrix} z_{4n-3}^{(1)} & z_{4n-2}^{(1)} \\ z_{4n-1}^{(1)} & z_{4n}^{(1)} \end{matrix}$ .Math. (43) $Z_n^{(2)} =$ .Math. $\begin{matrix} z_{4n-3}^{(2)} & z_{4n-2}^{(2)} \\ z_{4n-1}^{(2)} & z_{4n}^{(2)} \end{matrix}$ .Math. (44)

$Z_n^{(1)} = H_n^{(1)} W_n^{(1)}$ (45) $Z_n^{(2)} = W_n^{(2)} H_n^{(2)}$ (46)

(70) Then the signer sends the array Z given in equation (47), the array extracted from Y.sub.n given in equation (48), and the message M to a verifier.

Z={*z*.sub.n.sup.(1),*z*.sub.n.sup.(2)}.sub.n=1.sup.4N,**1**<*z*.sub.i.sup.(1),*z*.sub.i.sup.(2)<2.sup.L (47)

{*y*.sub.4n−2,*y*.sub.4n-1,*y*.sub.4n}.sub.n=1.sup.N (48)

(71) The process of verification is next described. The public key P is used to verify that the array Z is correctly follows from the message M. The hash H is calculated as described above, and the verifier obtains the matrices H.sub.n.sup.(1) and H.sub.n.sup.(2). The verification result is successful if equation (49) is fulfilled.

H.sub.n.sup.(1)*Y*.sub.n*H*.sub.n.sup.(2)=*Z*.sub.n.sup.(1)*X*.sub.n*Z*.sub.n.sup.(2) (49)

(72) Consider an example where L=16. Multiplying the equations (41) and (42) yields equation (50):

H.sub.n.sup.(1)*W*.sub.n.sup.(1)*W*.sub.n.sup.(2)*H*.sub.n.sup.(2)=*Z*.sub.n.sup.(1)*Z*.sub.n.sup.(2) (50)

(73) The message exchange method described above is not resistant against the man-in-the-middle attack (MITM) because neither Alice nor Bob have any authentication information about each other. In order to set the message exchange process resistant against MITM, an authentication transaction process is added to the message exchange method.

(74) Assume that Alice wants to pass the secret message M.sub.1 to Bob using Ed for the authentication procedure. Similarly, Bob wants to pass the secret message M.sub.4 to Alice using Ed.

(75) From Alice's part, the process includes generating the following numbers: secret key S.sub.A, public key P.sub.A, random A.sub.s and A.sub.r. S.sub.A, P.sub.A, A.sub.s, A.sub.r∈N.

(76) From Bob's parts, the process includes generating the following numbers: secret key S.sub.B, public key P.sub.B, random B.sub.s and B.sub.r, S.sub.B, P.sub.B, B.sub.s, B.sub.r∈N.

(77) From Ed's part, for the authentication transaction between Alice and Bob, the process includes the number

C.sub.A and C.sub.B as a result of equation (51):
$B_{r} = A_{s} \oplus C_{A}$
$A_{r} = B_{s} \oplus C_{B}$ (51)
(78) FIG. **3** illustrates the message transfer process with authentication transaction identifications on the “Ed” side.
(79) Message exchange forming will next be discussed. Alice wants to send a secret message X.sub.A to Bob. The process includes the generation of random variable G.sub.A on Alice's part. The messages M.sub.1 and M.sub.2 are a result of the calculations of equation (52).
$M_{1} = X_{A} \oplus G_{A}$
$M_{2} = A_{s} \oplus G_{A}$ (52)
(80) The message transfer process with authentication transaction identifications on the “Ed” side is shown in equation (53).
$M_{3} = M_{2} \oplus C_{A} = A_{s} \oplus G_{A} \oplus C_{A}$ (53)
(81) Bob receives messages M.sub.2 and M.sub.3 from Alice and Ed respectively. Bob applies the value B r to obtain G.sub.A as follows in equation (54).
$G_{A} = M_{3} \oplus B_{r} = A_{s} \oplus G_{A} \oplus C_{A} \oplus B_{r}$ (54)
(82) The message M.sub.1 are signed by Alice using secret key S.sub.A. Bob can check Alice's signature using public key P.sub.A. A similar process is used if Bob wants to send a secret message X.sub.B to Alice. The process includes the generation of random variable G.sub.B on Bob's part. The messages M.sub.4 and M.sub.5 are a result of the following calculations in equation (55).
$M_{4} = X_{B} \oplus G_{B}$
$M_{5} = B_{s} \oplus G_{B}$ (55)
(83) The message transfer process with authentication transaction identifications on the “Ed” side is modeled in equation (56).
$M_{6} = M_{5} \oplus C_{B} = B_{s} \oplus G_{B} \oplus C_{B}$ (56)
(84) Alice receives messages M.sub.5 and M.sub.6 from Bob and Ed respectively. The value A.sub.r is applied to obtain G B as follows in equation (57):
$G_{B} = M_{6} \oplus A_{r} = B_{s} \oplus G_{B} \oplus C_{B} \oplus A_{r}$ (57)
(85) The message M.sub.4 is signed by Bob using secret key S.sub.B. Alice can verify Bob's signature using public key P.sub.B.
(86) The user registration process is next described. The process below defines the registration procedure of Alice and Bob (or any other user in the system). The registration process is performed before participating in any data exchange. The registration process comprises generation of an authentication transaction value Ca on Alice's (user's) side and delivering an authentication transaction value to server (“Ed” side).
(87) An authentication transaction value is generated. An authentication transaction value Ca is taken as a random value (a byte sequence), which is generated and stored on user's side.
(88) The authentication transaction value is then delivered to a server (“Ed” side). This step includes delivery of an authentication transaction value Ca and storing this value in server records.
(89) FIG. **4** illustrates a scenario for sending an authentication transaction value to a server and validating the delivery. To deliver the Ca value and verify that the server received it, Alice knows a server public key Ps. A server public key Ps may be included into Alice's source code (making it initially known to Alice and immutable). A corresponding server key is Ss, which is known to the server only. To deliver the Ca value to server (“Ed”), Alice performs the next steps: (1) sending the Ca value to server (“Ed”), and (2) confirming Ca delivery by verifying the server signature and result.
(90) V is calculated on a server side as follows:
$H = \text{hash}(Ca)$ (58)
$V = \text{SIGN}(H, Ss)$ (59)
(91) Where: hash is a cryptographic hash function, Ca is an authentication transaction value, and SIGN is a cryptographic signature function.
(92) Alice calculates and verifies Ha=H as follows:
$Ha = \text{hash}(Ca)$ (60)
$H = Ha?$ (61)
(93) If the value of H does not equal value of Ha the verification procedure terminates with error.
(94) Next, Alice verifies the signature of H using the embedded Ps key. If the signature verification was unsuccessful, the verification procedure terminates with error. Otherwise, the procedure continues.
(95) The Ca value is stored locally (on Alice's side). Once Alice verified Ca delivery, it is safe to store the Ca

value locally.
(96) Then, Ed stores the received Ca value locally on its side.
(97) The inverse matrix X.sup.−1 of matrix
(98) $X =$ .Math. $\begin{matrix} x_1 & x_2 \\ x_3 & x_4 \end{matrix}$ .Math.
is defined as follows
(99) $X^{-1} = \frac{\text{.Math. } \begin{matrix} x_4 & -x_2 \\ -x_3 & x_1 \end{matrix} \text{ .Math.}}{x_1 x_4 - x_2 x_3}$ and $XX^{-1} = X^{-1}X = I$
(100) where I is the identity matrix,
(101) $I =$ .Math. $\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$ .Math. .
(102) In linear algebra and matrix mathematics, a centrosymmetric matrix is a matrix which is symmetric about its center. A centrosymmetric matrix A has the following form:
(103) $A =$ .Math. $\begin{matrix} a_1 & a_2 \\ a_2 & a_1 \end{matrix}$ .Math.
(104) Centrosymmetric matrices A and B satisfy the following conditions: AB=BC.
(105) A square matrix is singular if and only if its determinant is 0.
(106) The matrix
(107) $X =$ .Math. $\begin{matrix} x_1 & x_2 \\ x_3 & x_4 \end{matrix}$ .Math.
is singular if the determinant of the matrix X, det(X)=0 (e.g., x.sub.1x.sub.4−x.sub.2x.sub.3=0). If the matrix X is singular, the matrix B=AX is also singular.
(108) Consider a singular matrix S and an invertible, nondegenerate, or non-singular matrix V. The matrix W is also singular as a result of SV=W. The singular matrix S can be obtained if the matrices V and W are known, because S=WV.sup.−1, but the non-singular matrix V=S.sup.−1W is not obtained even if matrices S and W are known because the inverse matrix S.sup.−1 does not exist (division by zero). In this sense, there is no unique solution of the equation (ambiguity).
(109) The authentication system **100** can include at least one processing circuit. Such a processing circuit can include, for example, one or more processors and one or more storage devices that are coupled to a local interface. The local interface can include, for example, a data bus with an accompanying address/control bus or any other suitable bus structure. Similarly, each of the computing devices **160**-**164** can include at least one processing circuit. Such a processing circuit can include, for example, one or more processors and one or more storage devices that are coupled to a local interface.
(110) The storage devices for a processing circuit can store data or components that are executable by the processors of the processing circuit. For example, the authentication engine **132**, the cryptography engine **182**, the cryptography engine **192**, and/or other components can be stored in one or more storage devices and be executable by one or more processors in the authentication system **100**, the computing device **160**, and the computing device **161**.
(111) The authentication engine **132**, the cryptography engine **182**, the cryptography engine **192**, and/or other components described herein can be embodied in the form of hardware, as software components that are executable by hardware, or as a combination of software and hardware. If embodied as hardware, the components described herein can be implemented as a circuit or state machine that employs any suitable hardware technology. The hardware technology can include, for example, one or more microprocessors, discrete logic circuits having logic gates for implementing various logic functions upon an application of one or more data signals, application specific integrated circuits (ASICs) having appropriate logic gates, programmable logic devices (e.g., field-programmable gate array (FPGAs), and complex programmable logic devices (CPLDs)).
(112) Also, one or more or more of the components described herein that include software or program instructions can be embodied in any non-transitory computer-readable medium memory device for use by or in connection with an instruction execution system such as, a processor in a computer system or other system. The computer-readable medium can contain, store, and/or maintain the software or program instructions for use by or in connection with the instruction execution system.
(113) A computer-readable medium can include a physical media, such as, magnetic, optical, semiconductor, and/or other suitable media. Examples of a suitable computer-readable media include, but are not limited to, solid-state drives, magnetic drives, or flash memory. Further, any logic or component described herein can be implemented and structured in a variety of ways. For example, one or more components described can be implemented as modules or components of a single application. Further, one or more components described herein can be executed in one computing device or by using multiple computing devices.

(114) Further, any logic or applications described herein, including the authentication engine **132**, the cryptography engine **182**, the cryptography engine **192**, and/or other components can be implemented and structured in a variety of ways. For example, one or more applications described can be implemented as modules or components of a single application. Further, one or more applications described herein can be executed in shared or separate computing devices or a combination thereof. For example, a plurality of the applications described herein can execute in the same computing device, or in multiple computing devices. Additionally, terms such as “application,” “service,” “system,” “engine,” “module,” and so on can be used interchangeably and are not intended to be limiting.

(115) FIG. **5** illustrates a flowchart of a method of generating a digital signature for data according to some embodiments. In the step **500**, a hash of data is generated. The hash is able to include a first set of singular matrices. In the step **502**, a digital signature is generated based at least in part on the hash, random data and a private key. The random data is able to be a first set of regular matrices. The private key is able to be a second set of singular matrices. The digital signature is able to correspond to a set of matrices. The digital signature is able to correspond to a Diophantine second order equation. In some embodiments, fewer or additional steps are implemented. For example, the digital signature is able to be verified based at least in part on a public key of singular matrices. In another example, a signer user is able to be registered with an authentication server by generating an authentication transaction value and sending the authentication transaction value to the authentication server. In another example, a server digital signature and a hash of the authentication transaction value are received from the authentication server, and the server digital signature and the hash of the authentication transaction value are verified. In some embodiments, the order of the steps is modified.

(116) FIG. **6** illustrates a flowchart of a method of verifying a digital signature for data according to some embodiments. In the step **600**, data, a digital signature and a public key are received. The public key is able to include singular matrices. In the step **602**, a hash of the data is generated. The hash is able to include singular matrices. In the step **604**, the digital signature is verified in response to determining that a relation is fulfilled (e.g., H.sub.n.sup.(1)Y.sub.nH.sub.n.sup.(2)=Z.sub.n.sup.(1)X.sub.nZ.sub.n.sup.(2). In some embodiments, fewer or additional steps are implemented. For example, an authentication message is received for the data from an authentication server. In some embodiments, the order of the steps is modified.

(117) FIG. **7** illustrates a flowchart of a method of implementing an authentication process with an exposed and unregistered public certificate according to some embodiments. In the step **700**, a first device generates a hash of data. In the step **702**, the first device generates a digital signature based at least in part on the hash, random data, and a private key. The public key includes a first set of singular matrices. The digital signature corresponds to a set of matrices. The random data includes a first set of regular matrices, and the private key includes a second set of singular matrices and a second set of regular matrices. The hash includes a third set of singular matrices. In the step **704**, a second device receives the data, the digital signature, and a public key. In the step **706**, the second device verifies the digital signature in response to determining that a relation is fulfilled. The relation is H.sub.n.sup.(1)Y.sub.nH.sub.n.sup.(2)=Z.sub.n.sup.(1)X.sub.nZ.sub.n.sup.(2). The digital signature corresponds to a relation which corresponds to a Diophantine second order equation. In some embodiments, fewer or additional steps are implemented. For example, an authentication message for the data is received from an authentication server. In another example, the device for receiving the data is registered. Registering the device includes generating an authentication transaction value. In another example, a signer user is registered with an authentication server by generating an authentication transaction value and sending the authentication transaction value to the authentication server. In another example, a server digital signature and a hash of the authentication transaction value from the authentication server are received, and the server digital signature and the hash of the authentication transaction value are verified. In some embodiments, the order of the steps is modified.

(118) FIG. **8** illustrates a block diagram of an exemplary computing device configured to implement the authentication system according to some embodiments. The computing device **800** is able to be used to acquire, store, compute, process, communicate and/or display information such as images and videos including 3D content. The computing device **800** is able to implement any of the encoding/decoding aspects. In general, a hardware structure suitable for implementing the computing device **800** includes a network interface **802**, a memory **804**, a processor **806**, I/O device(s) **808**, a bus **810** and a storage device **812**. The choice of processor is not critical as long as a suitable processor with sufficient speed is chosen. The memory **804** is able to be any conventional computer memory known in the art. The storage device **812** is able to include a hard drive, CDROM, CDRW, DVD, DVDRW, High Definition disc/drive, ultra-HD drive, flash memory card or any other storage device. The computing device **800** is able to include one or more network interfaces **802**. An example of a network interface includes a network card connected to an Ethernet or other type of LAN. The I/O device(s) **808** are able to include one or more of the following: keyboard, mouse, monitor, screen, printer, modem, touchscreen, button interface and other devices. Authentication application(s) **830** used to implement the

authentication system are likely to be stored in the storage device **812** and memory **804** and processed as applications are typically processed. More or fewer components shown in FIG. **8** are able to be included in the computing device **800**. In some embodiments, authentication hardware **820** is included. Although the computing device **800** in FIG. **8** includes applications **830** and hardware **820** for the authentication system, the authentication method is able to be implemented on a computing device in hardware, firmware, software or any combination thereof. For example, in some embodiments, the authentication applications **830** are programmed in a memory and executed using a processor. In another example, in some embodiments, the authentication hardware **820** is programmed hardware logic including gates specifically designed to implement the authentication system.

(119) In some embodiments, the authentication application(s) **830** include several applications and/or modules. In some embodiments, modules include one or more submodules as well. In some embodiments, fewer or additional modules are able to be included.

(120) Examples of suitable computing devices include a personal computer, a laptop computer, a computer workstation, a server, a mainframe computer, a handheld computer, a personal digital assistant, a cellular/mobile telephone, a smart appliance, a gaming console, a digital camera, a digital camcorder, a camera phone, a smart phone, a portable music player, a tablet computer, a mobile device, a video player, a video disc writer/player (e.g., DVD writer/player, high definition disc writer/player, ultra high definition disc writer/player), a television, a home entertainment system, an augmented reality device, a virtual reality device, smart jewelry (e.g., smart watch), a vehicle (e.g., a self-driving vehicle) or any other suitable computing device.

(121) The architecture described herein includes the concept of network endpoints and devices being registered (onboarded) onto the authentication server system. Client software embedded onto endpoint systems work in conjunction with the authentication server and additional security technologies including digital signatures, key agreement and encapsulation and encryption techniques to generate a completely comprehensive and secure network.

(122) The system is composed of several major processes:

(123) Registration and authentication of endpoint systems. Registration and authentication manage the identification and security of endpoint systems for network access.

(124) Point-to-Point connection authentication. Incoming network connection requests from endpoints which are not registered to the authentication server and are therefore are rejected. This includes several layer 2-7 protocols including low-level management network functions such as ICMP, DHCP and ARP.

(125) Endpoint-to-endpoint network traffic is optionally protected using any of several techniques using shared key agreements. This supports payload encryption, packet tampering protection, eavesdropping and network tapping prevention, system masquerading and spoofing, man-in-the-middle attacks and guaranteed payload integrity.

(126) In some embodiments, authentication is required on only incoming connections. This allows support for network broadcast and multicast traffic.

(127) FIG. **9** illustrates a diagram of an architecture of a system to secure endpoints across various network LAN and WAN infrastructures according to some embodiments.

(128) The system secures endpoints **900** across various network LAN and WAN infrastructures.

(129) The endpoints **900** may be almost any network connected devices from large cloud systems and local servers, to desktops, printers, smartphones and even tiny IoT devices distributed over wireless network links. The endpoints **900** include any network-connected target computer systems from tiny IoT devices to large computers. An authentication server **902** is a centralized system which is populated with endpoint systems. A public name is the clear text name which is used to uniquely identify the endpoint system. This will often be a network address or network node name. A secret name is a unique endpoint system name which is encrypted before being exposed. Since network names such as addresses can be easily spoofed, the secret name is kept secret and encrypted when sent across the network. Only the endpoints and the authentication server have the decryption secret key to decrypt and view this secret endpoint name. A secret key is a unique encryption key which is generated and shared between each endpoint system and the authentication server. The secret key can be shared across insecure networks without ever exposing the key contents. The system prevents MAC spoofing, Man in the Middle/Replay attacks, impersonation and trespassing.

(130) Registration

(131) Each software or device endpoint is registered to the authentication server when onboarding to private networks. Registered devices can be authorized to communicate with each other, and bad actor systems **904** will not be able to engage or interoperate with other authorized endpoints.

(132) Onboarding systems into networks protected by authentication server domains is performed by an authorized administrator or protected embedded system. Examples of embedded systems include IoT gateway

and manager systems.
(133) The new endpoint and the authentication server perform a secret key exchange. This will implement a shared key structure to support communications using symmetric encryption operations. The secret key exchange is performed on both the new system and on the authentication server(s). The secret key is randomized and designed to prevent duplicate collisions. The key size is at least 256 bits to guarantee quantum resistance in subsequent operations. The secret key is refreshed, replacing the keys on both ends. The secret keys are stored inside system HSM hardware. Each endpoint system being authorized is assigned a unique name. The unique name is designed to be unique on the network. Secrecy of the unique name is not required. The unique name is stored in the Authentication Server and used to identify each endpoint.
(134) Authentication
(135) For any endpoint-to-endpoint network communications, each incoming network connection is authorized. Non-authorized endpoint systems attempting to perform a network connection are rejected.
(136) Software installed on each endpoint systems contain a cached list of other authorized endpoints. Incoming network connections are compared with the local authorized systems managed by the authentication server.
(137) If the incoming connection is from an endpoint not in the local endpoint cached list, this node queries the authentication server to authorize the incoming connection.
(138) If the incoming node name exists, the authentication server returns an acknowledgement record. The acknowledgment record is then added to the endpoint local authorized nodes list and the connection is completed, and network traffic is allowed.
(139) If the incoming node is not registered in the authentication server, the network connection is rejected by dropping the packet. Dropping the packet and not responding makes the endpoint “dark” on the network and not vulnerable to low-level network scans.
(140) The authentication server logs the failed connection attempt. This can be directed to log management systems to detect suspicious network incidents.
(141) The local endpoint authorized records list contains a Time-to-Live (TTL) value and periodically expires the record. Connections from this node involve re-authentication. The authorization refresh can be configured and increases the security level in possible cases where endpoint systems are possibly compromised.
(142) Authentication Server
(143) Server administration for the authentication server is managed directly by human or automated system administrators. In some cases, endpoint registration may be ingested from other network management systems.
(144) In some use case scenarios, the authentication server may be an embedded system. For example, the authentication server is able to be embedded into the dashboard computer electronics controlling autonomous vehicles and IoT devices.
(145) For authentication servers residing on public WAN networks, the server should be protected by standard TLS PKI systems using public/private certificates. This protects the authentication server from external security attacks including man-in-the-middle and other impersonation attacks.
(146) Secret Key Exchange
(147) FIG. **10** illustrates a diagram of an architecture of a system to perform a 3-way key exchange according to some embodiments.
(148) The libraries contain a technology (both high-performance proprietary and other NIST standards) which perform quantum resistant secret key exchanges over unsecure public networks. A key agreement is a 3-way key exchange protocol **1000**, where two endpoint systems (**1002**, **1006**) share secret data over a 3-way data exchange protocol **1000**.
(149) At the end of the key exchange, both systems have identical secret keys (**1004**, **1008**). During the exchange, the key is never exposed.
(150) The keys on both systems are not exposed, so they are stored securely. This is similar to private certificate management in standard PKI practices. This is typically done using HSM hardware which is hardware backed secure storage. Examples of this technology include: FIPS hardware, TPM embedded systems, and others.
(151) Hardware Security Modules (HSMs) are hardened, tamper-resistant hardware devices that strengthen encryption practices by generating keys, encrypting and decrypting data, and generating and verifying digital signatures.
(152) Encrypted Communications Protocol
(153) The communication between endpoints and other endpoints or the authentication server does not expose sensitive data elements. Security sensitive data, such as the endpoint database keys are encrypted using the shared secret keys (which are never exposed). Endpoints can communicate with other endpoints or with the authentication server securely, even over public networks.
(154) For some high-security environments, the data communications between systems can be further encrypted

using symmetrical encryption technologies, such as AES or other encryption technologies. These encryption methods can use the existing shared secret keys established.
(155) The technology has been optimized for high-speed LAN network segments using XOR encryption techniques. For WAN environments including the Internet, AES is not a noticeable performance issue.
(156) Periodic Refresh
(157) The shared keys are periodically refreshed with new keys. The refresh is performed to increase security by prevention of data from being harvested and attacked by external systems.
(158) Packet Payload Encryption
(159) Using endpoint to endpoint shared keys, the packet payload data can be encrypted with quantum resistant methods. This prevents eavesdropping of network traffic.
(160) The technique can be performed at the datalink layers (e.g., Ethernet or Bluetooth), or at higher network layers such as IP packets.
(161) Packet CRC/Hash Encryption
(162) Another level of security is available to guarantee data integrity is using a technique where only interpacket CRC or other hashes only are encrypted.
(163) This will guarantee data has not been tampered or altered. The advantages of this method are that it generates minimal performance overhead, but prevents packet data from being tampered with and from man-in-the-middle type attacks.
(164) System Registration Protocol
(165) Before endpoint systems can be allowed to communicate with other registered endpoints, each endpoint must be securely registered to the authentication server.
(166) Each endpoint system has a public name, a secret name, and Time-To-Live (TTL) data elements.
(167) The authentication server also stores the symmetric secret key for each endpoint.
(168) Each endpoint system must be explicitly registered by an authorized system administrator with the authentication server. The process includes:
(169) Performing a Secret Key exchange between the endpoint system and the Authentication Server. The secret key is stored securely in a key storage methodology on both endpoints. The new endpoint then provides the following data elements: a public name, a secret name and TTL. For Ethernet, a public name may be the MAC address. The secret name is known only by the endpoint and the authentication server. The TTL is provided as a policy by the system administrator. The secret name is a 32 (or more) byte random string. The value can be auto-generated. The secret name is encrypted using an XOR operation with the secret key. This is what will be transmitted to the authentication server. Key to transmit: Secret Name XOR Secret Key. The elements are transferred to the authentication server. The authentication server does not decrypt the secret name but stores the encrypted name.
(170) FIG. **11** illustrates a diagram of an authentication server and endpoints according to some embodiments. An authentication server **1100** stores public names, TTL and a secret key for each endpoint **1102**. Each endpoint **1102** stores the public name of the other endpoints and TTL.
(171) Endpoint Authentication Protocol
(172) Endpoint authentication is the process which authenticates endpoints from incoming network packets, and either authorizes data communications or rejects connection attempts.
(173) When endpoints receive incoming connection requests, the endpoint performs several steps. The source endpoint includes in the connection packet the public endpoint name and a token encrypted with the key shared with the authentication server during onboarding. The target endpoint checks if the source endpoint system is stored on the target endpoint system. This will reside in the target endpoint connection cache data records. If the record exists, the source endpoint will have been authenticated prior against the authentication server, and the target endpoint allows communications to continue. If the record does not exist, the endpoint queries the authentication server to validate if the incoming endpoint is authorized for communications. The public endpoint name and the encrypted token are sent to the authentication server. If the authentication server does not have a record of the incoming endpoint, the authentication server responds with a rejection message. The endpoint then rejects all incoming requests from this unregistered endpoint for a designated time defined by network policies. This prevents Denial of Service-type attacks.
(174) FIG. **12** illustrates a diagram of an accepted connection and rejected connection according to some embodiments. Based on the steps performed by the endpoint, a connection is able to be accepted **1200** or rejected **1202**.
(175) When the authentication server receives the authentication request from endpoints, the authentication server receives: the public name of the endpoint to authenticate and the secret encrypted token of the endpoint to authenticate. Examples of public names may be IP Addresses, Ethernet MAC addresses, Bluetooth addresses, or

others, or these names may be user defined for the endpoint. The token is encrypted using the shared encryption key which was generated when the endpoint was onboarded onto the authentication server.
(176) The authentication server then performs a database lookup for the stored public name. If the database record is found, the authentication server: uses the secret symmetric key stored for this endpoint, performs an XOR operation which will decrypt the secret token into a readable form (or potentially other encryption methods, but XOR is extremely compute efficient), and the decrypted token is compared to the token on the authentication server DB associated with the public endpoint name. If the decrypted token is the same as the stored token, the endpoint is considered valid and registered.
(177) XOR Encryption and Decryption
(178) XOR is a common technique to perform secure cryptographic functions with very low resource requirements. Doing an XOR operation includes: Secret Name: abcde in binary is: 01100001 01100010 01100011 01100100 01100101; Using a Secret Key of: 01010101 01010101 01010101 01010101 01010101; The encrypted result is: 11001011 11001000 11001001 11001110 11001111;
(179) When performing the XOR with the secret key again, the resultant is: 01100001 01100010 01100011 01100100 01100101, which is “abcde” in ASCII.
(180) Using large enough keys (e.g., >32 bytes), the technique is extremely difficult to hack using brute force. The technique uses encryption keys that are the same length of the source data, which is appropriate for smaller sized data elements.
(181) Denial of Service Prevention
(182) When an endpoint incoming request is the endpoint is denied by the authentication server, the endpoint system will continue to reject all subsequent connection attempts for a specified time. This is to prevent Denial of Service type of attacks.
(183) If a bad actor endpoint flooded another endpoint(s), and each connection attempt resulted in an authentication request to the authentication server, this would effectively overwhelm the authentication server and cause subsequent network disruptions. The authentication server is an important component of this network architecture and is protected from these kinds of threats.
(184) Network Use Cases
(185) The authentication system is able to use network names such as IP addresses, Ethernet MAC addresses, and various other network infrastructures.
(186) Private networks environments such as LAN segments are able to implement the authentication system.
(187) Public IP networks which can be routed by IP address are able to implement the authentication system. Many Internet networks use technologies which mask internal addresses, and such are not uniquely addressable.
(188) The authentication system is able to be used with Bluetooth, WiFi, IoT, Autonomous Cars, Smart Cities, Metaverse, and other implementations.
(189) Broadcast and Multicast
(190) Since only incoming network connection requests are authenticated, broadcast and multicast traffic are authenticated using the same method. Incoming traffic is from a single endpoint and is processed and authenticated similarly.
(191) Isolated Security Zones
(192) This authentication system can be used to identify scopes of endpoints which are allowed to communicate. These zones can be identified as individual groups, and any groups not allowed to communicate within this group are isolated.
(193) This is a valuable technique to generate secure zones of endpoints which can intercommunicate securely.
(194) The method described herein is able to be implemented in a LAN environment. A LAN is able to include switches, routers, gateways, and other networking equipment. When a device attempts to communicate with another device in the LAN, several steps are performed. Initially, registration or pre-registration occurs. All of the nodes of the LAN are registered. Secret key exchange (using HSM, TPN, other hardware) is implemented as described herein. If a secret key exchange fails, then the communication is blocked (e.g., using XOR verification). Once a node is verified (pre-authorized), the process of verification is not performed again, and a node is able to communicate with another node (e.g., by using a pre-authenticated token).
(195) To utilize the authentication system, a device sends or receives data using a Diophantine system. The authentication system is able to be implemented with user assistance or automatically without user involvement.
(196) In operation, the authentication system enables the secure transmission of data from one device to another. For example, a user is able to communicate with another user without worry that a third party is going to intercept and view the contents of the communication. In another example, a device is able to communicate information (e.g., a password, banking information, private information) to another device in a secure manner.
(197) The authentication system includes a network security methodology for preventing unauthorized access,

packet tampering, network tapping and eavesdropping, man-in-the-middle and other security attacks. The network infrastructures are agnostic supporting a form of WAN and LAN network architectures.
(198) The design objectives for the authentication system are: 1. High performance, high-scale and high-throughput performance with limited overhead and network delays. 2. Control of hardware or software endpoints that can connect and participate with other network resources. This is achieved by identifying unauthorized network guests and excluding them from communicating within the network. 3. Guarantee endpoint authenticity. Preventing authorized network endpoints from being impersonated or spoofed by malicious systems. 4. Prevention of Man-in-the-Middle and Replay-type attacks. 5. Network architecture agnostic—functional on a broad variety of network protocols and architectures including Ethernet, Internet Protocol, Bluetooth, RF Wireless, 5G cellular, and others. 6. Does not rely on specific hardware types or features, such as network switch or router security extensions. Any network hardware transports are supported including fiber cabling, radio and carrier networks, and all layers of the OSI network stack protocols. 7. Designed for high throughput low latency performance with insignificant hardware overhead. 8. Able to be deployed in very small capacity devices like smartphones, IoT hardware, Bluetooth attach devices as simple as mice and keyboards. 9. Endpoint systems equipped with this technology can be completely isolated from the rest of the network. Incoming connection requests are dropped and not detectable by other network systems. 10. Supports broadcast and multicast network communications. 11. Endpoint to endpoint security (optional advanced secure mode). Prevents network packet tampering where payload data is modified fraudulently. Fully encrypts network packet payload data. 12. Provides fully quantum resistant security solutions.
(199) The above-described examples of the present disclosure are merely possible examples of implementations set forth for a clear understanding of the principles of the disclosure. Many variations and modifications can be made without departing substantially from the spirit and principles of the disclosure. All such modifications and variations are intended to be included herein within the scope of this disclosure and protected by the following claims.
(200) Disjunctive language, such as the phrase “at least one of X, Y, or Z,” unless specifically stated otherwise, is to be understood with the context as used in general to present that an item, term, etc., can be either X, Y, or Z, or any combination thereof (e.g., X, Y, and/or Z). Thus, such disjunctive language is not generally intended to, and should not, imply that certain embodiments require at least one of X, at least one of Y, or at least one of Z to be each present.
(201) It should be emphasized that the above-described embodiments of the present disclosure are merely possible examples of implementations set forth for a clear understanding of the principles of the disclosure. Many variations and modifications can be made to the above-described embodiment(s) without departing substantially from the spirit and principles of the disclosure. All such modifications and variations are intended to be included herein within the scope of this disclosure and protected by the following claims.

Claims

1. A method comprising: generating, with a first device, a hash of data; generating, with the first device, a digital signature based at least in part on the hash, random data, and a private key; receiving, at a second device, the data, the digital signature, and a public key; and verifying, with the second device, the digital signature in response to determining that a relation is fulfilled, wherein the first device and the second device communicate over a network, wherein the public key comprises a first set of singular matrices, wherein the digital signature corresponds to a set of matrices, and wherein the random data comprises a first set of non-singular matrices, and the private key comprises a second set of singular matrices and a second set of non-singular matrices.
2. The method of claim 1 wherein the hash comprises a third set of singular matrices.
3. The method of claim 1 wherein the relation comprises: H.sub.n.sup.(1)Y.sub.nH.sub.n.sup.(2)=Z.sub.n.sup.(1)X.sub.nZ.sub.n.sup.(2), wherein H.sub.n.sup.(1) is a first singular matrix, Y.sub.n is a matrix, H.sub.n.sup.(2) is a second singular matrix, Z.sub.n.sup.(1) is a third singular matrix, X.sub.n is a second matrix, and Z.sub.n.sup.(2) is fourth singular matrix.
4. The method of claim 1 further comprising receiving an authentication message for the data from an authentication server.
5. The method of claim 1 wherein the digital signature corresponds to a relation which corresponds to a Diophantine second order equation.
6. The method of claim 1 further comprising registering the device for receiving the data.
7. The method of claim 6 wherein registering the device includes generating an authentication transaction value.
8. The method of claim 6 wherein registering the device is performed by a protected embedded system.

9. The method of claim 1 further comprising registering a signer user with an authentication server by generating an authentication transaction value and sending the authentication transaction value to the authentication server.
10. The method of claim 9, further comprising: receiving a server digital signature and a hash of the authentication transaction value from the authentication server, and verifying the server digital signature and the hash of the authentication transaction value.
11. An apparatus comprising: a non-transitory memory configured for storing an application, the application configured for: generating a hash of data; generating a digital signature based at least in part on the hash, random data, and a private key; and sending the data, the digital signature, and a public key to a second apparatus, wherein the digital signature is verified by the second apparatus in response to determining that a relation is fulfilled, wherein the apparatus and the second apparatus communicate over a network, wherein the public key comprises a first set of singular matrices, wherein the digital signature corresponds to a set of matrices, and wherein the random data comprises a first set of non-singular matrices, and the private key comprises a second set of singular matrices and a second set of non-singular matrices; and a processor configured for processing the application.
12. The apparatus of claim 11 wherein the hash comprises a third set of singular matrices.
13. The apparatus of claim 11 wherein the relation comprises: H.sub.n.sup.(1)Y.sub.nH.sub.n.sup.(2)=Z.sub.n.sup.(1)X.sub.nZ.sub.n.sup.(2), wherein H.sub.n.sup.(1) is a first singular matrix, Y.sub.n is a matrix, H.sub.n.sup.(2) is a second singular matrix, Z.sub.n.sup.(1) is a third singular matrix, X.sub.n is a second matrix, and Z.sub.n.sup.(2) is fourth singular matrix.
14. The apparatus of claim 11 wherein the digital signature corresponds to a relation which corresponds to a Diophantine second order equation.
15. A method comprising: generating, with a first device, a hash of data; generating, with the first device, a digital signature based at least in part on the hash, random data, and a private key, wherein the digital signature corresponds to a set of matrices, wherein the random data comprises a first set of non-singular matrices, and the private key comprises a first set of singular matrices and a second set of non-singular matrices, wherein the hash comprises a third set of singular matrices; receiving, at a second device, the data, the digital signature, and a public key, wherein the public key comprises a second set of singular matrices; and verifying, with the second device, the digital signature in response to determining that a relation is fulfilled, wherein the first device and the second device communicate over a network, wherein the relation comprises: H.sub.n.sup.(1)Y.sub.nH.sub.n.sup.(2)=Z.sub.n.sup.(1)X.sub.nZ.sub.n.sup.(2), wherein H.sub.n.sup.(1) is a first singular matrix, Y.sub.n is a matrix, H.sub.n.sup.(2) is a second singular matrix, Z.sub.n.sup.(1) is a third singular matrix, X.sub.n is a second matrix, and Z.sub.n.sup.(2) is fourth singular matrix.
16. The method of claim 15 further comprising receiving an authentication message for the data from an authentication server.
17. The method of claim 15 further comprising registering a user device for receiving the data.
18. The method of claim 17 wherein registering the user device includes generating an authentication transaction value.
19. The method of claim 17 wherein registering the device is performed by a protected embedded system.
20. A method comprising: generating, with a first device, a hash of data; generating, with the first device, a digital signature based at least in part on the hash, random data, and a private key; receiving, at a second device, the data, the digital signature, and a public key; and verifying, with the second device, the digital signature in response to determining that a relation is fulfilled, wherein the first device and the second device communicate over a network, wherein the relation comprises: H.sub.n.sup.(1)Y.sub.nH.sub.n.sup.(2)=Z.sub.n.sup.(1)X.sub.nZ.sub.n.sup.(2), wherein H.sub.n.sup.(1) is a first singular matrix, Y.sub.n is a matrix, H.sub.n.sup.(2) is a second singular matrix, Z.sub.n.sup.(1) is a third singular matrix, X.sub.n is a second matrix, and Z.sub.n.sup.(2) is fourth singular matrix.
21. An apparatus comprising: a non-transitory memory configured for storing an application, the application configured for: generating a hash of data; generating a digital signature based at least in part on the hash, random data, and a private key; and sending the data, the digital signature, and a public key to a second apparatus, wherein the digital signature is verified by the second apparatus in response to determining that a relation is fulfilled, wherein the apparatus and the second apparatus communicate over a network, wherein the relation comprises: H.sub.n.sup.(1)Y.sub.nH.sub.n.sup.(2)=Z.sub.n.sup.(1)X.sub.nZ.sub.n.sup.(2), wherein H.sub.n.sup.(1) is a first singular matrix, Y.sub.n is a matrix, H.sub.n.sup.(2) is a second singular matrix, Z.sub.n.sup.(1) is a third singular matrix, X.sub.n is a second matrix, and Z.sub.n.sup.(2) is fourth singular matrix; and a processor configured for processing the application.