US012388840B2

(12) **United States Patent**
Kaliski, Jr. et al.

(10) **Patent No.: US 12,388,840 B2**
(45) **Date of Patent: Aug. 12, 2025**

(54) **STRENGTHENING INTEGRITY ASSURANCES FOR DNS DATA**

(71) Applicant: **VeriSign, Inc.**, Reston, VA (US)

(72) Inventors: **Burton S. Kaliski, Jr.**, McLean, VA (US); **Eric Osterweil**, Fairfax, VA (US)

(73) Assignee: **VeriSign, Inc.**, Reston, VA (US)

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

(21) Appl. No.: **18/126,188**

(22) Filed: **Mar. 24, 2023**

(65) **Prior Publication Data**

US 2023/0231856 A1 Jul. 20, 2023

Related U.S. Application Data

(63) Continuation of application No. 16/938,345, filed on Jul. 24, 2020, now Pat. No. 11,616,788, which is a continuation of application No. 16/143,232, filed on Sep. 26, 2018, now Pat. No. 11,005,856, which is a continuation of application No. 15/221,867, filed on Jul. 28, 2016, now Pat. No. 10,110,614.

(51) **Int. Cl.**

H04L 29/06	(2006.01)
H04L 9/40	(2022.01)
H04L 61/4511	(2022.01)
H04L 101/604	(2022.01)

(52) **U.S. Cl.**

CPC ***H04L 63/126*** (2013.01); ***H04L 61/4511*** (2022.05); ***H04L 63/123*** (2013.01); *H04L 63/1483* (2013.01); *H04L 2101/604* (2022.05)

(58) **Field of Classification Search**

CPC . H04L 63/126; H04L 63/123; H04L 61/1511; H04L 63/1483; H04L 61/6004

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,480,508	B1	11/2002	Mwikalo et al.	
8,224,994	B1 *	7/2012	Schneider	G06Q 30/02 709/217
8,626,950	B1	1/2014	MacCarthaigh et al.	
8,874,790	B2	10/2014	McPherson	
9,473,516	B1	10/2016	Jezorek et al.	
10,110,614	B2	10/2018	Kalinski et al.	
10,530,734	B2	1/2020	Kalinski et al.	
10,791,085	B2	9/2020	Thakar	

(Continued)

FOREIGN PATENT DOCUMENTS

EP	2779591 A2	9/2014
EP	3035650 A2	6/2016

(Continued)

OTHER PUBLICATIONS

Arends et al., Resource Records for the DNS Security Extensions. Network Working Group. 29 pages, Mar. 2005.

(Continued)

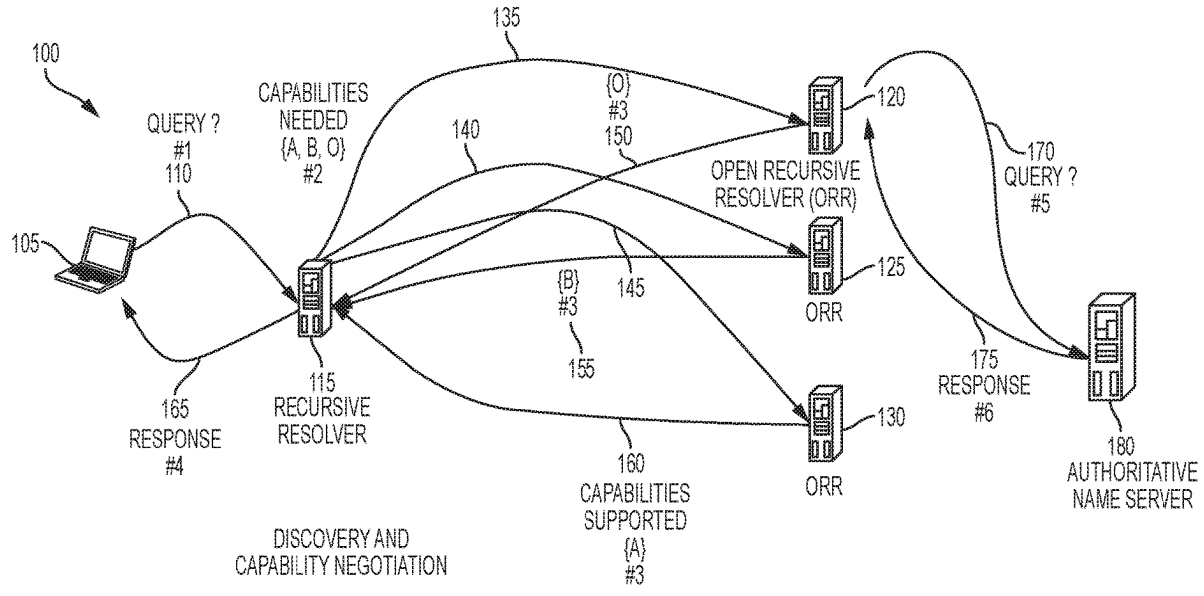
Primary Examiner — Mahfuzur Rahman

(74) *Attorney, Agent, or Firm* — McCarter & English, LLP; Michael A. Sartori

(57) **ABSTRACT**

One or more DNS services are provided that are configured to not only tolerate some commonly observed DNSSEC misconfigurations (while still providing DNSSEC's security guarantees), but also provide a more intelligent DNS resolution process informed by DNSSEC.

20 Claims, 6 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

11,005,856	B2	5/2021	Kaliski, Jr. et al.	
11,082,392	B1	8/2021	Kaliski, Jr. et al.	
11,616,788	B2	3/2023	Kaliski, Jr. et al.	
12,020,178	B2 *	6/2024	Donoho	G06Q 40/02
2002/0161745	A1	10/2002	Call	
2003/0182447	A1	9/2003	Schilling	
2004/0194102	A1	9/2004	Neerdaels	
2005/0259645	A1	11/2005	Chen et al.	
2006/0056371	A1	3/2006	Sakuda et al.	
2006/0088039	A1	4/2006	Kakivaya et al.	
2006/0129665	A1	6/2006	Toebes et al.	
2007/0160200	A1	7/2007	Ishikawa et al.	
2007/0250189	A1	10/2007	Rourke et al.	
2007/0294419	A1	12/2007	Ulevitch	
2008/0071616	A1	3/2008	Hovnanian et al.	
2009/0031006	A1 *	1/2009	Johnson	H04M 3/42348 709/218
2009/0049164	A1	2/2009	Mizuno	
2009/0157889	A1	6/2009	Treuhaft	
2010/0121981	A1	5/2010	Drako	
2012/0054497	A1	3/2012	Korhonen	
2013/0173825	A1	7/2013	McPherson	
2013/0254423	A1	9/2013	George, IV	
2013/0290563	A1 *	10/2013	Fleischman	H04L 61/4511 709/245
2014/0155018	A1	6/2014	Fan et al.	
2014/0215628	A1	7/2014	Yan	
2014/0244998	A1	8/2014	Amenedo et al.	
2014/0304412	A1	10/2014	Prakash et al.	
2015/0058931	A1 *	2/2015	Miu	G06Q 20/02 707/784
2015/0074221	A1	3/2015	Kuparinen et al.	
2015/0180892	A1	6/2015	Balderas	
2015/0271031	A1 *	9/2015	Beevers	H04L 43/08 709/224
2015/0295882	A1	10/2015	Kaliski, Jr.	
2015/0304199	A1	10/2015	Leask et al.	
2016/0021055	A1 *	1/2016	Krzywonos	H04L 61/4511 709/245
2016/0036848	A1	2/2016	Reddy et al.	
2016/0057237	A1 *	2/2016	Yang	H04L 63/123 709/224
2016/0170814	A1 *	6/2016	Li	G06F 9/542 719/318
2016/0173439	A1	6/2016	Kaliski, Jr. et al.	
2016/0182473	A1 *	6/2016	Cignetti	G06F 21/44 713/171
2016/0197898	A1	7/2016	Hozza et al.	
2017/0048186	A1	2/2017	Blinn	
2018/0034827	A1	2/2018	Kaliski, Jr. et al.	
2019/0044955	A1	2/2019	Kaliski, Jr. et al.	
2019/0097965	A1 *	3/2019	Linari	H04L 63/168

FOREIGN PATENT DOCUMENTS

EP	3276921	A1	1/2018
EP	3576381	A1	12/2019
KR	10-2007-0113600	A	11/2007
WO	WO-2009/005433	A1	1/2009

OTHER PUBLICATIONS

Bortzmeyer et al., DNS query name minimisation to improve privacy, draft-ietf-dnsop-qname-minimisation-00. Network Working Group. 7 pages, Oct. 22, 2014.

Busch, How to Set Up OpenDNS on Your Home Network [Router Configuration]. gPost. Retrieved online at: http://www.groovypost.com/howtosetup/-opendns-home-network-router/. 10 pages, Feb. 25, 2013.

Cox et al., Serving DNS using a Peer-to-Peer Lookup Service. International Workshop on Peer-to-Peer Systems. 7 pages, (2002).

Daley et al., Use of DNS SRV records for host selection draft-daley-dnext-host-srv-00.txt. Network Working Group. 13 pages, Dec. 31, 2009.

Damas et al., Extension Mechanisms for DNS (EDNS (0). Internet Engineering Task Force (IETF). 16 pages, Apr. 2013.

Datatracker, Managed Incident Lightweight Exchange (mile). Retrieved online at: https://datatracker.ietf.org/wg/mile/documents/. 2 pages, retrieved Nov. 24, 2020.

DNS-OARC, Domain Name System Operations Analysis and Research Center, Introduction to DNS-OARC. Retrieved online at: https://www.dns-oarc.net. 3 pages, Jul. 3, 2008.

Edmonds et al., Signaling DNS Capabilities. Networking Group. 7 pages, Jul. 2, 2017.

Elz et al., Clarifications to the DNS Specification. Network Working Group. 15 pages, Jul. 1997.

Getdns, Download the 1.6.0 beta release. Retrieve online at: https://tetdnsapi.net. 6 pages, retrieved Dec. 10, 2014.

Google, Set DNS. Retrieved online at: http://web.archive.org/web/20151003145236 htts://play.google.com/store/apps/details?id=uk.com/mytechie.setDNS&hl+en. 2 pages, Oct. 3, 2015.

Herzberg et al., Cipher-Suite Negotiation for DNSSEC Hop-by-Hop or End-to-End? IEEE Internet Computing. 2015 19(1):80-84.

Herzberg et al., Less is More: Cipher-Suite Negotiations for DNSSEC. Proceedings of the 30th Annual Computer Security Applications Conference, ACM. 10 pages, Dec. 8-12, 2014.

Herzberg et al., Negotiating DNSSEC Algorithms Over Legacy Proxies. International Conference on Cryptology and Network Security, CANS. pp. 111-126, (2014).

Hu et al., Specification for DNS over Transport Layer Security (TLS). Internet Engineering Task Force (IETF). 18 pages, May 2016.

Hu et al., Starting TLS over DNS, draft-hzhwm-start-tls-for-dns-01. Network Working Group. 12 pages, Jul. 4, 2014.

Huque et al., Algorithm Negotiation in DNSSEC draft-huque-dnssec-alg-nego-01. Internet Engineering Task Force. 7 pages, Jul. 20, 2017.

Iana, Internet Assigned Numbers Authority, Domain Name System (DNS) Parameters. Retrieved online at: https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml. 4 pages, Aug. 30, 2017.

Icann, Draft Contract for Services between ccTLD Managers and ICANN. ccTLD Constituency Draft Contract for Services. 8th Draft. Retrieved online at: https://archive.icann.org/en/cctlds/cctldconst-8th-draft-contract-14nov00.htm. 4 pages, Nov. 14, 2000.

ICANN, Identifier Technology Innovation Panel—Draft Report. The Internet Corporation for Assigned Names and Numbers. Retrieved online at: http://www.icann.org/en/about/planning/strategic-engagement/identifier-technology/report-21feb14-en.pdf. 28 pages, Feb. 21, 2014.

ICANN, RSSAC002, Rssac Advisory on Measurements of the Root Server System. Retrieved online at: https://www.icann.org/en/system/files/files/rssac-002-measurements-root-20nov14-en.pdf. 15 pages, Nov. 20, 2014.

ICANN, Service Expectations of Root Servers. RSSAC-001, 10 pages, May 2, 2013.

ITFT, Information-Centric Networking Research Group, ICNRG. Retrieved online at: https://irtf.org/icnrg. 4 pages, Apr. 7, 2019.

Kumari et al., Decreasing Access Time to Root Servers by Running One on Loopback, draft-wkumari-dnsop-root-loopback-00. Network Working Group. 5 pages, Oct. 25, 2014.

Lee et al., How to scale the DNS root system? draft-lee-dnsop-scalingroot-00.txt. DNSOP Working Group. 11 pages, Jul. 3, 2014.

Lewis et al., DNS Zone Transfer Protocol (AXFR). Internet Engineering Task Force (IETF). 28 pages, Jun. 2010.

Microsoft Docs, Assign a Conditional Forwarder for a Domain Name. Retrieved online at: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc794735(v=ws.10)?redirectedfrom=MSDN. 2 pages, Jul. 2, 2012.

Mozilla Foundation, Public Suffix List. Retrieved online at: https://publicsuffix.org. 1 page, (2007).

Nagele, Analysis of Increased Query Load on Root Name Servers. Retrieved online at: https://labs.ripe.net/author/wnagele/increased-query-load-on-root-name-servers/. 6 pages, Jul. 11, 2011.

Namecoin, Decentralize all the things! Retrieved online at: http://namecoin.info/. 5 pages, Jul. 20, 2016.

(56) **References Cited**

OTHER PUBLICATIONS

Osterweil et al., Deploying and Monitoring DNS Security (DNSSEC). Computer Security Applications Conference, ACSAC'09. pp. 429-438, (2009).

Osterweil et al., Opportunistic Encryption with DANE Semantics and IPsec: IPSECA draft-osterweil-dane-ipsec-03. DANE, 19 pages, Jul. 6, 2015.

Osterweil et al., Verifying Keys through Publicity and Communities of Trust: Quantifying Off-Axis Corroboration. IEEE Transactions on Parrallel and Distributed Systems. 2014;25:9 pages.

Osterweil, Measurable Security: a New Substrate for DNSSEC. A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Computer Science. University of California, Los Angeles. 177 pages, (2010).

Park et al., CoDNS: Improving DNS Performance and Reliability via Cooperative Lookups. USENIX Association, OSDI '04: 6th Symposium on Operating Systems Design and Implementation. pp. 199-214, (2004).

Ramasubramanian et al., The Design and Implementation of a Next Generation Name Service for the Internet. ACM SIGCOMM Computer Communication Review. 2004;34(4):331-342.

Savolainen et al., Improved Recursive DNS Server Selection for Multi-Interfaced Nodes. Internet Engineering Task Force (IETF). 29 pages, Dec. 2012.

Sevilla et al., iDNS: Enabling Information Centric Networking Through The DNS. 2014 IEEE Infocom Workshops on Name-Oriented Mobility. pp. 476-481, (2014).

Shaikh et al., On the Effectiveness of DNS-based Server Selection. IEEE INFOCOM. Apr. 22, 2001;3:1801-1810.

Shimokawa et al., Flexible Server Selection in Widely Distributed Environments. Research Reports on Information Services and Electrical Engineering of Kyushu University. Mar. 5, 2000(1):7-12.

Taxii, Trusted Automated exchange of Indicator Information (TAXII™) 1.x Archive Website. Go to the TAXII 2.0 website (https://oasis-open.github.io/documentation/). Retrieved online at: taxiiproject.github.lo. 2 pages, (2019).

Ulevitch, Introducing FamilyShield Parental Controls. Retrieved online at: https://umbrella.cisco.com/blog/introducing-familyshield-parental-controls. 3 pages, Jun. 23, 2010.

VeriSign, Comments on Identifier Technology Innovation Panel Draft Report. Press release, http://mm.icann.org/pipermail/itipanel/attachments/20140430/361e539a/verisign-comments-panel-2014-04-30-0001.pdf. 4 pages, Apr. 30, 2014.

VeriSign, New gTLD Security and Stability Considerations. Verisign Labs Technical Report #1130007 version 2.2. 10 pages, Mar. 2013.

VerSign, New gTLD Security, Stability, Resiliency Update: Exploratory Consumer Impact Analysis. Verisign Labs Technical Report #1130008 Version 1.1. 28 pages, Aug. 22, 2013.

Vixie, Passive DNS and ISC SIE. DNS-OARC Workshop, retrieved online at: https://indico.dns-oarc.net/contributionDisplay.py?contribid=4&confid=8. 10 pages, Nov. 2-3, 2007.

Weimer, Passive DNS Replication. FIRST, retrieved online at: http://www.enyo.de/fw/sofrware/dnslogger/first2005-paper.pdf. 13 pages, Apr. 2005.

Zhang et al., Certificate Transparency for Domain Name System Security Extensions draft-zhang-trans-ct-dnssec-03. Networking Group. 13 pages, Jul. 5, 2015.

European Office Action for Application No. 1520405.7, dated May 30, 2016, 10 pages.

European Office Action for Application No. 17183682.8, dated Dec. 4, 2017, 7 pages.

European Office Action for Application No. 19173127.2, dated Jul. 16, 2020, 12 pages.

European Office Action for Application No. 19173127.2, dated Oct. 7, 2019, 15 pages.

International Search Report and Written Opinion for Application No. PCT/US2016/061679, dated Feb. 28, 2017, 13 pages.

US Final Office Action for U.S. Appl. No. 16/143,232, dated Apr. 2, 2020, 15 pages.

US Non-Final Office Action for U.S. Appl. No. 15/221,867, dated Dec. 20, 2017, 14 pages.

US Notice of Allowance for U.S. Appl. No. 15/221,867, dated Jun. 6, 2018, 9 pages.

US Notice of Allowance for U.S. Appl. No. 15/221,867, dated Sep. 6, 2018, 5 pages.

US Notice of Allowance for U.S. Appl. No. 16/143,232, dated Jan. 13, 2021, 11 pages.

US Notice of Allowance for U.S. Appl. No. 16/143,232, dated Aug. 26, 2020, 9 pages.

US Office Action for U.S. Appl. No. 16/143,232, dated Sep. 30, 2019, 12 pages.

* cited by examiner

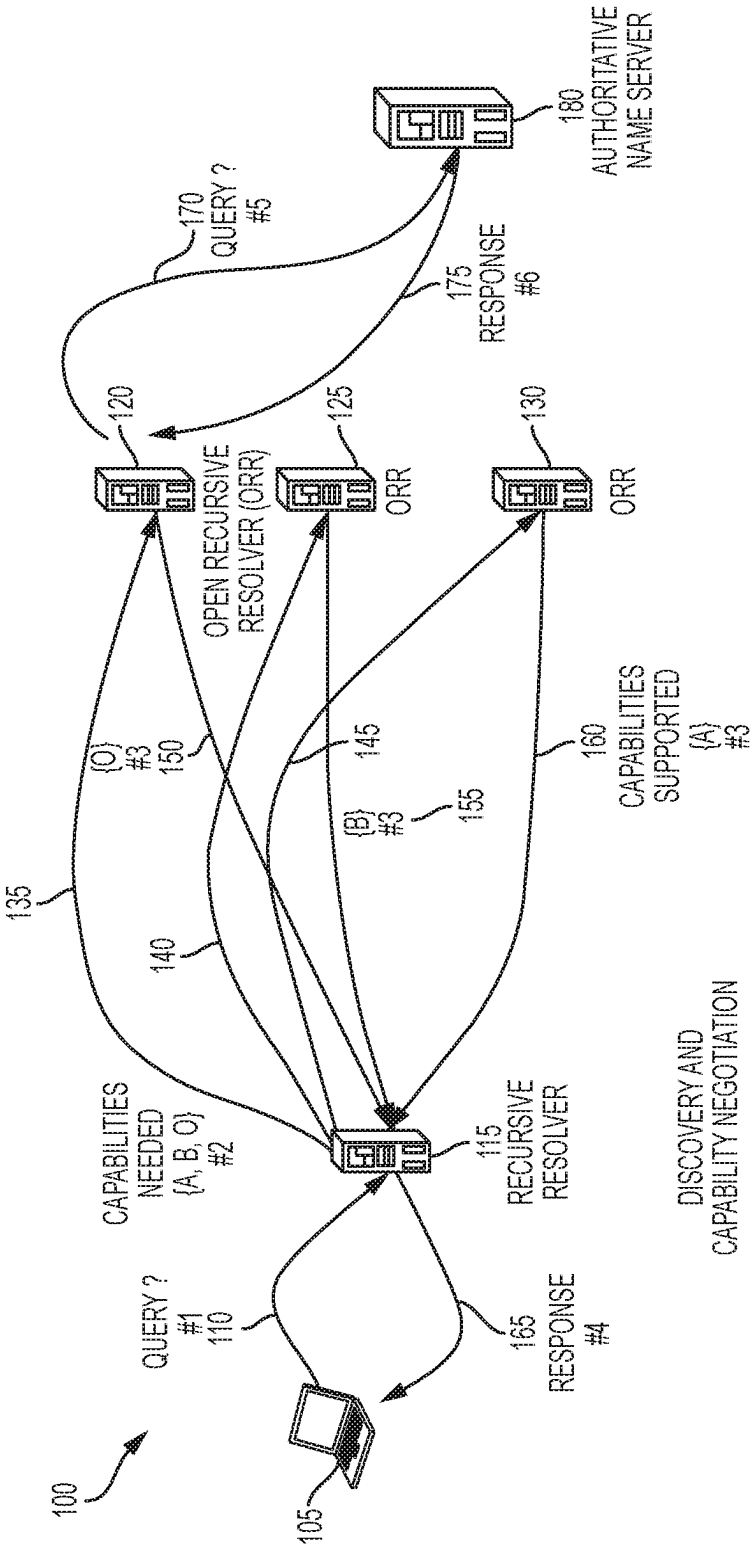

100
105
QUERY ?
#1
110
CAPABILITIES
NEEDED
{A, B, O}
#2
115
RECURSIVE
RESOLVER
DISCOVERY AND
CAPABILITY NEGOTIATION
120
OPEN RECURSIVE
RESOLVER (ORR)
125
ORR
130
ORR
135
140
{O}
#3
150
{B}
#3
155
145
160
CAPABILITIES
SUPPORTED
{A}
#3
165
RESPONSE
#4
170
QUERY ?
#5
175
RESPONSE
#6
180
AUTHORITATIVE
NAME SERVER

FIG. 1

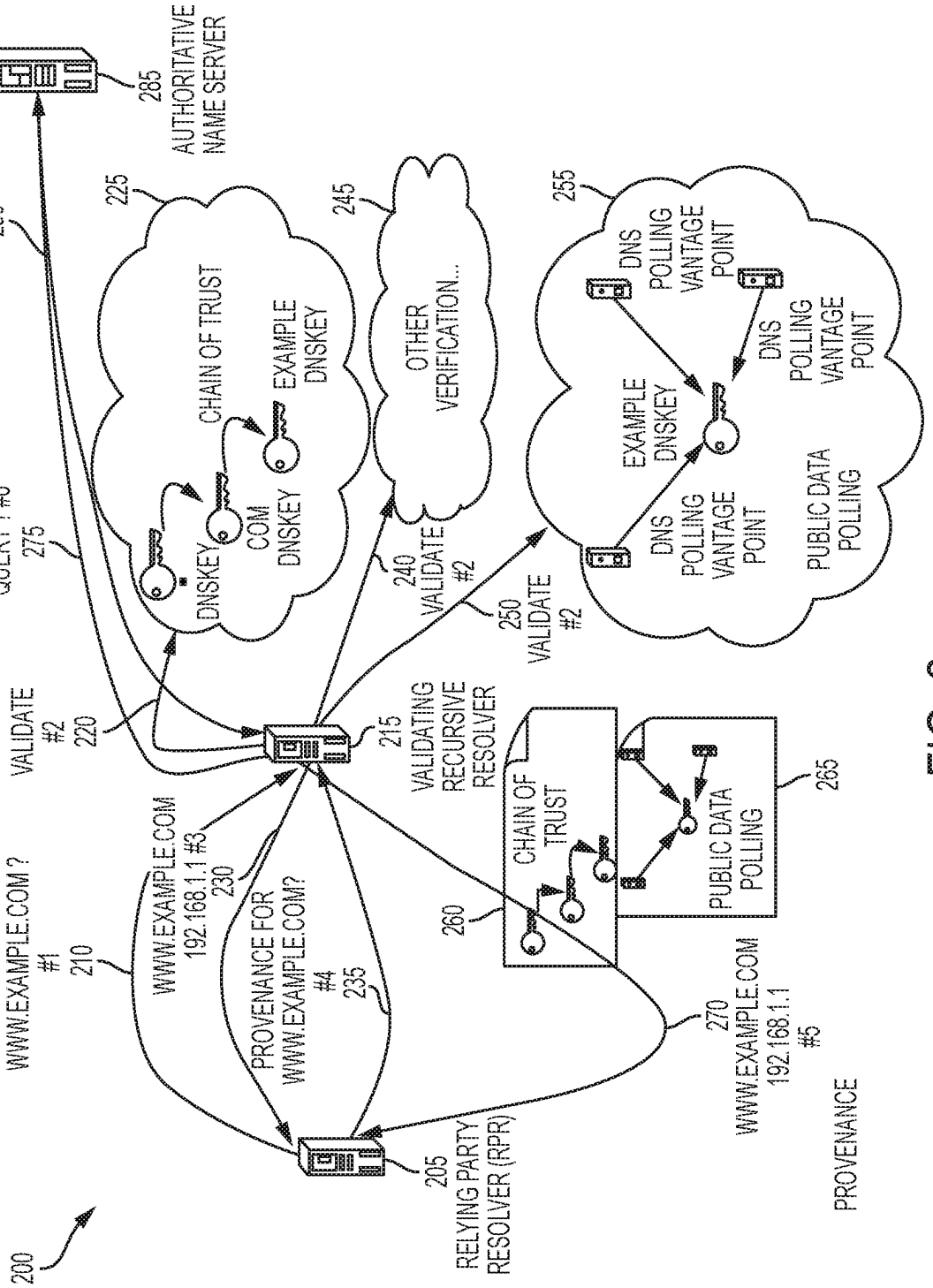

200
RELYING PARTY RESOLVER (RPR)
205
WWW.EXAMPLE.COM ?
#1
210
VALIDATE
#2
220
WWW.EXAMPLE.COM
192.168.1.1 #3
230
PROVENANCE FOR
WWW.EXAMPLE.COM?
#4
235
VALIDATING RECURSIVE RESOLVER
215
CHAIN OF TRUST
260
PUBLIC DATA POLLING
265
WWW.EXAMPLE.COM
192.168.1.1
#5
270
PROVENANCE
QUERY ? #6
275
RESPONSE #7
280
AUTHORITATIVE NAME SERVER
285
CHAIN OF TRUST
DNSKEY
COM DNSKEY
EXAMPLE DNSKEY
225
OTHER VERIFICATION...
245
240 VALIDATE #2
250 VALIDATE #2
DNS POLLING VANTAGE POINT
DNS POLLING VANTAGE POINT
EXAMPLE DNSKEY
DNS POLLING VANTAGE POINT
PUBLIC DATA POLLING
255

FIG. 2

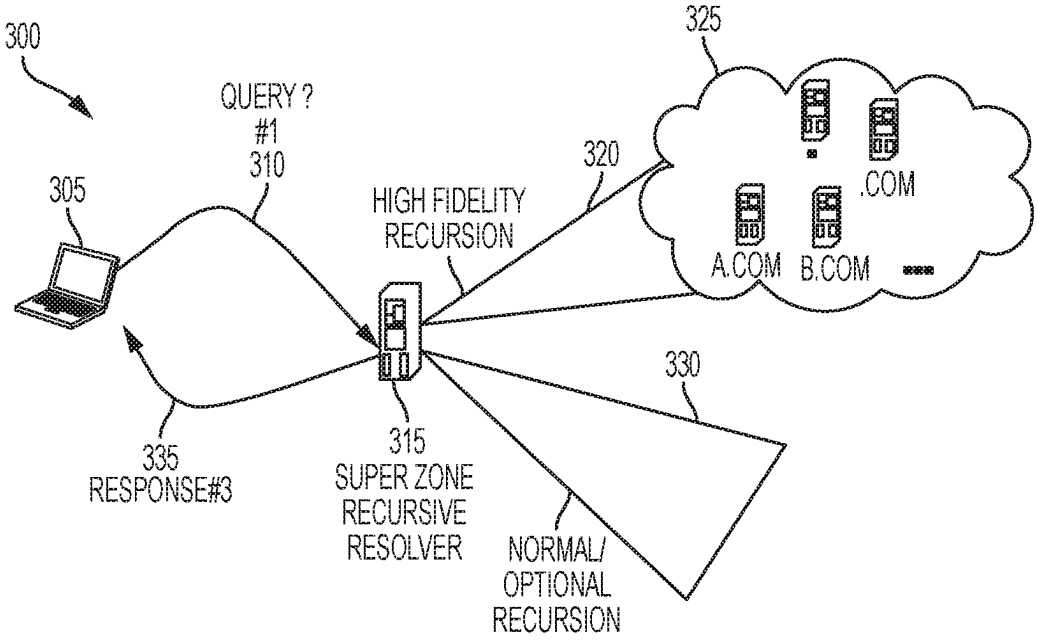

300
QUERY ?
#1
310
305
HIGH FIDELITY
RECURSION
320
325
.COM
A.COM
B.COM
330
335
RESPONSE#3
315
SUPER ZONE
RECURSIVE
RESOLVER
NORMAL/
OPTIONAL
RECURSION

FIG. 3

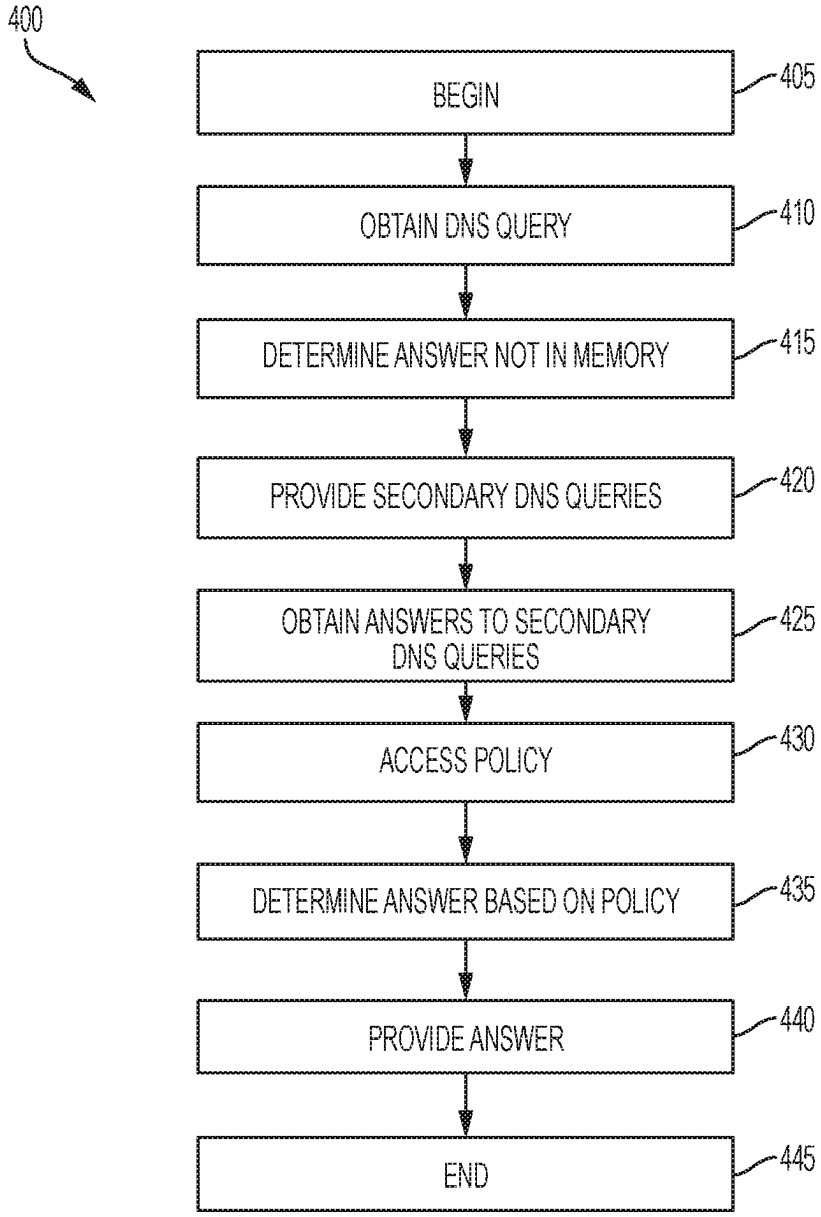

400
BEGIN
405
OBTAIN DNS QUERY
410
DETERMINE ANSWER NOT IN MEMORY
415
PROVIDE SECONDARY DNS QUERIES
420
OBTAIN ANSWERS TO SECONDARY DNS QUERIES
425
ACCESS POLICY
430
DETERMINE ANSWER BASED ON POLICY
435
PROVIDE ANSWER
440
END
445

FIG. 4

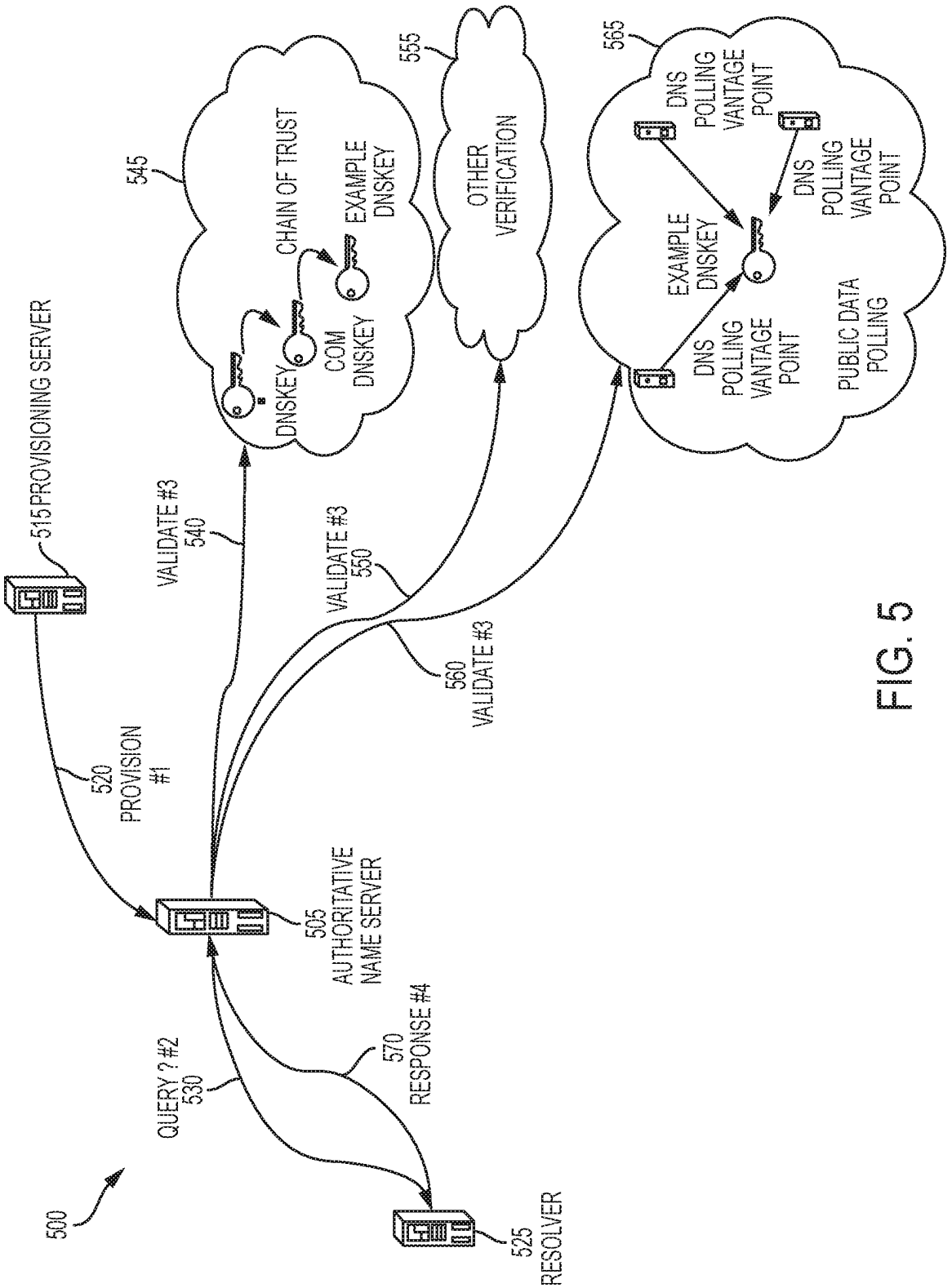

500
515 PROVISIONING SERVER
520 PROVISION #1
505 AUTHORITATIVE NAME SERVER
525 RESOLVER
530 QUERY ? #2
570 RESPONSE #4
540 VALIDATE #3
550 VALIDATE #3
560 VALIDATE #3
545
DNSKEY
COM DNSKEY
EXAMPLE DNSKEY
CHAIN OF TRUST
555 OTHER VERIFICATION
565
DNS POLLING VANTAGE POINT
DNS POLLING VANTAGE POINT
DNS POLLING VANTAGE POINT
EXAMPLE DNSKEY
PUBLIC DATA POLLING

FIG. 5

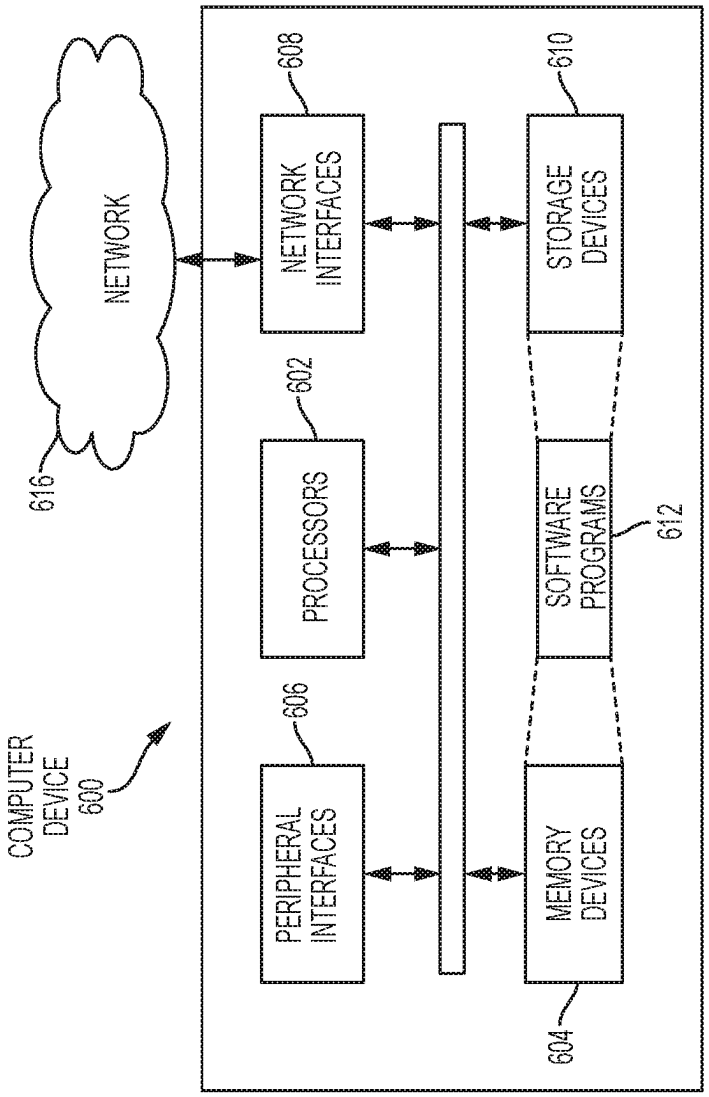

COMPUTER DEVICE 600
NETWORK 616
NETWORK INTERFACES 608
PROCESSORS 602
PERIPHERAL INTERFACES 606
STORAGE DEVICES 610
SOFTWARE PROGRAMS 612
MEMORY DEVICES 604

FIG. 6

STRENGTHENING INTEGRITY ASSURANCES FOR DNS DATA

CROSS REFERENCE TO RELATED APPLICATIONS

The present application is a continuation of U.S. patent application Ser. No. 16/938,345 filed on Jul. 24, 2020, which is a continuation of U.S. patent application Ser. No. 16/143,232 filed on Sep. 26, 2018, which in turn is a continuation of U.S. patent application Ser. No. 15/221,867 filed on Jul. 28, 2016, each of which are hereby incorporated by reference in their entireties.

FIELD

The present disclosure relates generally to domain name system ("DNS") security extensions ("DNSSEC").

BACKGROUND

DNSSEC is a suite of Internet Engineering Task Force (IETF) specifications for securing certain kinds of information provided by the DNS as used on Internet Protocol (IP) networks. It is a set of extensions to DNS, which provide to DNS users origin authentication of DNS data, authenticated denial of existence, and data integrity. All answers in DNSSEC are digitally signed. By checking the digital signature, a DNS user is able to check if the information is identical (correct and complete) to the information published at the authoritative DNS server.

DNSSEC was standardized in 2005 and uses a straightforward hierarchical verification architecture to learn keys and verify data. DNSSEC has become one component of naming and resolution services provided by DNS registry services. However, it has become apparent that DNSSEC's verification model does not adequately support the flexibility and robustness needed by Internet systems. DNSSEC's design verifies DNS data when deployment operates without benign misconfigurations. This is in contrast to DNS, which offers robustness to many types of misconfigurations. In a sense, DNSSEC presumes near-perfect operational deployments.

By way of a simplified example of a top-level domain (TLD) implementing DNSSEC, the DNS records in the TLD zone file are digitally signed using a private key. The corresponding public key is published as a DNSKEY record in the TLD zone file, and is given to the root name server's provisioning system, which digitally signs a DNS record containing the fingerprint of the public key (a Delegation Signer ("DS") record) with the root zone's private key. The root zone's public key may be retrieved directly by a relying party from a local trust list by a client application. A lookup request queries the trusted root zone for authoritative name server information for the TLD and for the associated public key fingerprint. The public key fingerprint is then used to verify the TLD's public key. This process keeps the chain of trust intact. Because a lookup request begins with a trusted node (the root server), each subsequent step in the chain of lookups maintains the trust by using the public/key private key infrastructure. Thus, once the TLD's public key is verified using the public key fingerprint from its "parent", the root zone, the TLD name server returns the public key fingerprint for the next authoritative name server, which is digitally signed with the TLD's private key. The next authoritative name server has also digitally signed its DNS records with a private key. The chain continues indefinitely

until the last node is reached and the ultimate DNS record, e.g., a record containing a web server's IP address, is determined. (Note that in practice, the DNSSEC trust chain typically is slightly more complex, with two levels of keys per zone. A key-signing key signs DNSKEY records, and a zone-signing key signs other records, including the DS record containing the fingerprint of the next zone's key-signing key.)

If a failure occurs during at any stage of the DNSSEC chain of trust verification process, the requestor typically has no other mechanism to validate the requested DNS record. The requestor may be provided the DNS record and may have to make a determination as to whether the record is trustworthy. Alternatively, the requestor may not be provided the DNS record. In either case, the results are not optimal for the requestor. Thus, there is a need for a mechanism to validate DNS records when DNSSEC is not functioning properly, i.e., when DNSSEC is "imperfect."

SUMMARY

According to examples of the present disclosure, a method of resolving a Domain Name System (DNS) query is provided. The method comprises enabling a capability offered by a resolver to be determined by a relying party, wherein the capability relates to a predetermined set of domains; obtaining the DNS query at the resolver; determining, by the resolver, whether the DNS query is for a domain within the predetermined set of domains; and resolving the DNS query using a first recursion process when the DNS query is for the domain within the predetermined set of domains.

According to examples of the present disclosure, a method of resolving a Domain Name System (DNS) query is provided. The method comprises determining, by a relying party, a capability offered by a resolver, wherein the capability relates to an association between the resolver and a predetermined set of domains; determining, by a hardware processor of the relying party, whether the DNS query is for a domain within the predetermined set of domains; and sending the DNS query to the resolver when the DNS query is for the domain within the predetermined set of domains, wherein the DNS query is resolved using a first recursion process.

According to examples of the present disclosure, a method of resolving a Domain Name System (DNS) query is provided. The method comprises determining, by a relying party, a first capability offered by a first resolver, wherein the first capability relates to a first association between the first resolver and a first predetermined set of domains; determining, by the relying party, a second capability offered by a second resolver, wherein the second capability relates to a second association between the second resolver and a second predetermined set of domains; determining, by a hardware processor of the relying party, whether the DNS query is for a domain within the first predetermined set of domains; sending the DNS query to the first resolver when the DNS query is for the domain within the first predetermined set of domains, wherein the first resolver resolves the DNS query using a first recursion process of the first resolver; determining, by the hardware processor of the relying party, whether the DNS query is for a domain within the second predetermined set of domains, when the DNS query is not for the domain within the first predetermined set of domains; and sending the DNS query to the second resolver when the DNS query is for the domain within the second predetermined set of domains, wherein the second

resolver resolves the DNS query using a first recursion process of the second resolver.

It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the implementations, as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a discovery and capability negotiation process, according to examples of the present disclosure.

FIG. **2** shows a provenance discovery mechanism, according to examples of the present disclosure.

FIG. **3** shows an enhanced resolution process, according to examples of the present disclosure.

FIG. **4** shows a method for evidence-based DNS resolution according to examples of the present disclosure.

FIG. **5** shows a DNS resolution process for an authoritative name server according to examples of the present disclosure.

FIG. **6** is an example computer system for performing the disclosed implementations, consistent with the present disclosure.

DETAILED DESCRIPTION

Reference will now be made in detail to example implementations, which are illustrated in the accompanying drawings. When appropriate, the same reference numbers are used throughout the drawings to refer to the same or like parts.

For simplicity and illustrative purposes, the principles of the present disclosure are described by referring mainly to exemplary implementations thereof. However, one of ordinary skill in the art would readily recognize that the same principles are equally applicable to, and can be implemented in, all types of information and systems, and that any such variations do not depart from the true spirit and scope of the present disclosure. Moreover, in the following detailed description, references are made to the accompanying figures, which illustrate specific exemplary implementations. Electrical, mechanical, logical and structural changes may be made to the exemplary implementations without departing from the spirit and scope of the present disclosure. The following detailed description is, therefore, not to be taken in a limiting sense and the scope of the present disclosure is defined by the appended claims and their equivalents.

Generally speaking, examples of the present disclosure provide DNS services that are configured to not only tolerate some commonly observed DNSSEC misconfigurations (while still providing DNSSEC's security guarantees), but also provide a more intelligent DNS resolution process informed by DNSSEC. Examples of the present disclosure enhance DNSSEC's robustness by adding one or more additional contributions. First, DNSSEC's simplistic architecture for verifying DNSKEY records and DNS content (via the delegation chain of trust) is augmented by creating an orthogonal concept for validating data. DNSSEC zones and content can be validated during verification failures of the DNSSEC chain of trust through other evidence (rather than just using the delegation chain of trust). Evidence-based validation can include, but is not limited to, using quantifiably diverse observations of DNSKEY consistency, using the chain of trust itself, incorporating reputation aspects of one or more witnesses, i.e., upstream recursive resolver(s), authoritative name server(s), etc., and drawing from integrity assurances provided when responses are delivered by a recursive resolver and/or authoritative name server over a secure communications channel (e.g., protected with DNS-over-TLS or another security protocol that provides integrity protection). To do this, one or more witnesses, such as the upstream recursive resolvers, can be diversely spread across one or more networks, thereby creating topologically diverse evidence for DNSSEC-secured answers and/or for DNS answers more generally. The one or more witnesses can include witnesses of the same type platforms, such as all the witnesses being open recursive resolvers ("ORRs") or can include mixed platforms, where one can be an ORR, another can be a private resolver, another can be a non-resolver that provides evidence for DNSSEC processing or other assurance that a DNS record is correct, and another can be an authoritative name server with a secure communications channel. Resolvers may vary in trustworthiness, e.g., some may be managed by well-known operators, and others' oversight may be uncertain or unknown. In general, evidence from more trusted resolvers may be preferred, although evidence from a set of resolvers that are less trusted, but are unlikely to collude with one another, may also be acceptable. Examples of non-resolvers that may provide assurance that a DNS record is correct include services that provide access to one or more of the following: zone files containing the DNS record or related DNSSEC keys; zone modification request logs containing the command sequence that resulted in the DNS record or related DNSSEC keys (including Extensible Provisioning Protocol (EPP) transaction logs, DNS operator command logs, and email request/response logs); error logs and "trouble tickets" indicating possible errors in the DNS record or related DNSSEC keys that may have since been corrected; "lookaside" validation zones providing lists of trusted DNSSEC keys; passive DNS databases and DNSSEC transparency logs offering relying parties' views of the DNS record or related DNSSEC keys; public ledgers, e.g., block chains, providing support for the DNS record or related DNSSEC keys; and alternate DNS data distribution networks. In addition, if there is more than one authoritative name server serving a DNS record and/or related DNSSEC key, each of the authoritative name servers may be considered as an additional witness for the record and/or key. Although in principle, each name server that is authoritative for a given DNS record or key should respond with the same copy of the record, in practice, responses may vary due to instability. Accordingly, it may be valuable for a resolver or other relying party to consult with more than one authoritative name server for information about a given DNS record or key similarly to the reliance on other non-resolver witnesses in cases of uncertainty. A query sent to a witness may be associated with a single DNS query, or may support multiple potential DNS queries (e.g., a request for a zone file). The query sent to the witness may be sent before, or after, the DNS query(ies) with which it is associated and/or that it supports. The query sent to a witness may or may not be a DNS query. The evidence exchanged among parties may be exchanged directly, e.g., the actual observations, records, logs, etc., and/or indirectly, e.g., identifiers of such information, pointers to the location(s) where such information may be obtained, hashes or digital signatures of such information, etc.

The evidence-based validation can include information obtained at a local DNS recursive resolver on behalf of a requestor based, at least in part, from one or more witnesses, i.e., ORRs, etc. The one or more witnesses can be discovered in an in-line process (during DNS resolution processes for a requestor) or in an out-of-band process (witnesses discov-

ered separate from the DNS resolution process). The local DNS recursive resolver can be configured to negotiate capabilities needed by a requestor and offered by the one or more witnesses. The one or more witnesses can be configured to expose/export the provenance needed for evidence-based decisions. Other evidence-based validation information can include, but is not limited to, the reputation of the DNS provider for the authoritative name server, and WHOIS/registration data associated with a domain name. The WHOIS/registration data can be used to determine, among other things, changes of registrant and/or registrar that might affect the stability of the associated zone file and/or DNS records. Depending on the relying party's policy, DNS responses that do not have perfect DNSSEC validation, but are received over a secure connection with a resolver that includes the evidence-based validation information can be considered as providing the same assurance as DNS responses that do have perfect DNSSEC validation. Similarly, DNS responses that are received over a secure connection with an authoritative name server, but do not have perfect DNSSEC validation, can be considered as providing the same level of assurance.

Second, examples of the present disclosure provide for the arbitration of a single response to a requestor by evaluating responses from one or more witnesses, i.e., one or more upstream recursive resolvers—this is more general than the first example and is not limited to mitigating DNSSEC failures. Third, examples of the present disclosure provide for the determination that different recursive resolvers may be better/more trustworthy/more optimized/etc. for different zones and different features, i.e., Transport Layer Security Association (TLSA) records, privacy, etc. A relying party can select among one or more of multiple resolvers on the basis of the zone and/or feature of interest.

Examples of the present disclosure provide benefits including, but are not limited to, facilitating more operationally feasible transfers of zones between registrars and hosting providers (allowing zones to not go insecure during handoffs), protecting zones against validation failures if predecessor zones (higher in the delegation hierarchy) misconfigure DNSSEC (which would normally cause verification failures for all descendant zones), more secure responses, more heterogeneity of query response capabilities, faster resolution, increased privacy, and more nuanced privacy.

Rather than relying solely on DNSSEC's chain of trust secure delegation model, evidence-based validation with DNSSEC in accordance with examples of the present disclosure allows other evidence to be used to validate a DNSKEY record. For example, the other evidence can come from polling one or more of witnesses throughout the internet to see if the one or more witnesses have a record for the same DNSKEY record for the same DNS zone. A relying party's resolver can incorporate this evidence into its decision whether to trust a given DNSKEY record, e.g., based on a majority, or all, of the witnesses returning the same answer, or applying different weights to different witnesses based on reputation or other data. Then, the relying party's resolver can have some assurance that that DNSKEY record can be safely used to verify the digital signature on another DNS record that was requested, i.e., an A record, a MX record, etc. More generally, evidence-based validation can complement DNSSEC by providing alternate forms of assurance that a DNS record is correct, e.g., if the DNS record was obtained from a reputable source and/or over a secure communications channel. Reputation and integrity assurances for witnesses may be based in part on authenticating a witness's identity and/or responses via a certificate and/or key. The certificate may further attest to certain properties of the witness, e.g., its compliance with privacy and/or security policies of potential interest to relying parties. The certificate, key, or associated information may be published via a DANE record. A relying party may decide whether to connect to a witness and/or rely on its responses based on these properties. In addition, although evidence-based validation is described here in the context of DNS records and DNSSEC, the approaches can also be applied in similar systems where records are authenticated with digital signatures and digital signatures are validated following a (potentially imperfect) chain of trust, e.g., Information-Centric Networking.

FIG. **1** shows a discovery and capability negotiation process **100**, according to examples of the present disclosure. Computer (requestor) **105** composes DNS query **110** using, for example, a local stub resolver (not shown) that forms DNS query **110** according to a DNS protocol. Computer **105** provides DNS query **110** to local recursive resolver **115** over a network (not shown). Computer **105** is located in a domain, e.g., network, that is serviced by local recursive resolver **115**. Local recursive resolver **115** maintains a list of witnesses that have been pre-discovered in a separate discovery process. The witnesses can be geographically dispersed in the network and/or across networks providing topologically diverse network information to local recursive resolver **115** and computer **105**. The geographically dispersed witnesses can provide quantifiably diverse observations of DNSKEY consistency. Witnesses can vary based on suitability for different zones and features where a relying party can select among one or more resolvers on the basis of the zones and/or feature of interest. As shown in FIG. **1**, witnesses are shown as three open recursive resolver ("ORR") **120**, **125**, **130**; however, this is just one example of the witnesses. Witnesses can be of the same type or can include different mixture of types. Other examples of witnesses include private resolvers configured to service queries from at least local recursive resolver **115**, and non-resolvers that provide evidence for DNSSEC processing. ORRs are resolvers that are configured, for example based on a local policy, to answer queries from any requestor, whereas local recursive resolver **115** is configured, based on a local policy, to answer queries from devices having a predetermined range of network addresses. In some examples, local recursive resolver **115** may be implemented directly on computer **105** without network communication between computer **105** and local recursive resolver **115** and/or may be integrated with a local stub resolver on computer **105** as a single process. Local recursive resolver **115** provides requests **135**, **140**, **145** to ORRs **120**, **125**, **130**, respectively, to negotiate capabilities needed for computer **105** and/or for local recursive resolver **115**. ORRs **120**, **125**, **130** provide answers **150**, **155**, **160**, respectively, to local recursive resolver **115** and local recursive resolver **115** analyzes answers **150**, **155**, **160** to complete the capability negotiation. The capability negotiation may be performed in conjunction with the processing of DNS query **110** or separately as a configuration or maintenance operation. Local recursive resolver **115** may also send DNS queries (not shown) to and obtain DNS responses (not shown) from ORRs **120**, **125**, **130** as part of processing DNS query **110**. Local recursive resolver **115** then provides response **165** to computer **105** according to the DNS protocol over a network (not shown). If the witness is not a private resolver then the operations may be substantially the same as just described. If the witness is a non-resolver then the capability negotia-

tion may be substantially the same but local recursive resolver **115** may send non-DNS queries and obtain non-DNS responses to the witness as part of processing DNS query **110**. Note that local recursive resolver **115** and ORRs **120**, **125**, **130** may also send DNS queries **170** to and obtain DNS responses **180** from one or more authoritative name servers **175** as part of processing DNS query **110** (interaction shown only from ORR **120** in figure).

FIG. **2** shows a provenance discovery mechanism **200**, according to examples of the present disclosure. Relying party resolver ("RPR") **205** provides DNS query **210** to validating recursive resolver ("VRR") **215** for a DNS resource record ("RR") A-type, e.g., the IP address, for www.example.com. In some examples, RPR **205** may be local recursive resolver **115**, VRR **215** may be one of the ORRs **120**, **125**, **130**, and the relying party (not shown) that communicates with RPR **205** may be computer **105**, as described in FIG. **1**. In some examples, VRR **215** may be configured to provide at least some of the security functionality of RPR **205** and may not be initially fully trustworthy to RPR **205**. In some examples, RPR **205** may be implemented directly on computer **105** without network communication between computer **105** and RPR **205**.

Query **210** from RPR **205** may be for other types of RRs, such as but not limited to, DNSSEC-specific RRs including resource record signature ("RRSIG"), DNSKEY, DS, as well as other DNS RRs. VRR **215** can provide one or more validations for the answer to query **210**. The first validation **220** can be for a chain of trust validation using DNSSEC **225**. RPR **205** can set a "DO" flag bit in DNS query **210**. RPR **205** receives an answer via the normal DNS lookup process and RPR **205** then checks to make sure that the answer is correct. RPR **205** starts with verifying the DS and DNSKEY records at the DNS root. Then RPR **205** uses the DS records for the "corn" top level domain found at the root to verify the DNSKEY records in the "corn" zone. From there, RPR **205** checks for a DS record for the "example-.com" subdomain in the "corn" zone, and if there were, RPR **205** uses the DS record to verify a DNSKEY record found in the "example.com" zone. Finally, RPR **205** verifies the RRSIG record found in the answer for the A records for "www.example.com". If the chain of trust is verified using the above process, RPR **205** sets an "AD" flag bit in the answer **230**, i.e., the IP address for www.example.com, provided to the relying party. If, on the other hand, the chain of trust is not verified, the "AD" flag bit is not set. Depending on a policy of the requestor, the unverified answer or no answer is returned.

RPR **205** can request **235** that VRR **215** provide provenance to prove that the answer **230** is correct in either case where the chain of trust is verified or not. VRR **215** can provide provenance in the form of a second validation **240** that can include other verification **245** including, but are not limited to, the reputation of the DNS provider for the authoritative name server, and WHOIS/registration data associated with a domain name (to determine, e.g., changes of registrant and/or registrar that might affect the stability of the associated zone file), whether responses are delivered over a secure communications channel as described above, zone files, zone modification request logs, error logs, and public ledgers, etc. VRR **215** can provide provenance in the form of a third validation **250** that can include public data polling information **255** and in the form of a fourth validation including both a chain of trust **260** and public data polling **265**. The various provenance may thus assist RPR in its processing of a DNS query. RPR **205** may interact with multiple witnesses, e.g., multiple VRRs, and request and obtain provenance in multiple forms from these witnesses. Note that RPR **205** and VRR **215** may also send DNS queries **275** to and obtain DNS responses **280** from one or more authoritative name servers **285** as part of processing a DNS query (interaction shown only from VRR **215** in figure).

FIG. **3** shows an enhanced resolution process, according to examples of the present disclosure. Computer (requestor) **305** composes DNS query **310** using, for example, a local stub resolver (not shown) that forms DNS query **310** according to a DNS protocol. Computer **305** provides DNS query **310** to super zone recursive resolver **315** over a network (not shown). A super zone is a set of related zones above and below a given domain in the DNS delegation hierarchy. For example, super zone **325** includes the root zone ("."), the ".com" zone, child zones delegated from the ".com" zone such as "a.com" and "b.com", and possibly further descendants of these child zones. Thus, super zone **325** includes a set of related zones above and below the ".com" domain. (Note that the use of the ".com" TLD is illustrative only and the process can also be applied to other TLDs and domains. Note also that super zone **325** may be configured to include only a subset of such zones and domains.) Computer **305** is located in a domain, e.g., network, that is serviced by super zone recursive resolver **315**. In some examples, super zone recursive resolver **315** may be local recursive resolver **115** or RRR **205**. Super zone recursive resolver **315** may provide high-fidelity recursion for DNS records in super zone **325** by interactions described in FIG. **1** and FIG. **2**. For example, super zone recursive resolver **315** may interact with ORRs **120**, **125**, **130**, VRR **215**, and/or other witnesses that have been pre-discovered in a separate discovery process and that can provide high-fidelity recursion **320** for DNS records in super zone **325**. High-fidelity recursion may include witness discovery, capability negotiation, evidence-based validation, and provenance discovery, in addition to normal recursive services, as described in FIG. **1** and FIG. **2**. Such high-fidelity recursion may be specialized based on particular characteristics of domains, DNS records, and/or services in super zone **325**, e.g., the resolution process may be enhanced based on provenance, reputation, WHOIS data, communications channel security, etc. specific to super zone **325**. As previously, witnesses can be geographically dispersed in the network and/or across networks providing topologically diverse network information to super zone recursive resolver **315** and computer **305**. The geographically dispersed witnesses can provide quantifiably diverse observations of DNSKEY consistency. Super zone recursive resolver **315** may also provide normal recursive services **330** for computer **305** according to the DNS protocol, in which case super zone recursive resolver **315** process DNS query **310** in the normal way for domain names not in super zone **325**, and in the enhanced way described herein for domain names in super zone **325**. Super zone recursive resolver **315** may determine response **335** itself and/or by interacting with other resolvers. Alternatively, if DNS query **310** specifies a domain name that is not in super zone **325**, super zone recursive resolver **315** may provide response **335** indicating that it is not configured to respond to queries not in super zone **325**. Super zone recursive resolver **315** may maintain a policy (not shown) that determines the type of services to provide to requestors, such as whether to provide high-fidelity recursion **320** or normal recursive services **330**. Super zone recursive resolver **315** analyzes answers received from either high-fidelity recursion **320** or the normal recursive service **330** and provides response **335** to computer **305** according to the DNS protocol over a network

(not shown). In addition to interacting directly with computer **305**, super zone recursive resolver **315** may also interact with local recursive resolver **115** or RPR **205** as a witness or upstream recursive resolver, e.g., as ORRs **120**, **125**, **130** and/or VRR **215**. In other words, super zone recursive resolver **315** can be a witness suitable for a particular zone and/or feature and a relying party (e.g., local recursive resolver **115**, RPR **205**, and/or computer **305**) can select super zone recursive resolver **315** based on a zone and/or feature interest. For example, a relying party can interact with super zone recursive resolver **315** specifically for support in resolving DNS records in the super zone **325**. A relying party may interact with multiple such super zone recursive resolvers **315**, for the same and/or for different super zones **325**. Super zone recursive resolver **315** may advertise its capabilities and/or negotiate capabilities with a relying party according to methods described in U.S. patent application Ser. No. 14/627,506, "Balancing Visibility in the Domain Name System." Note that the specialization to super zone **325** is convenient for enhanced DNSSEC processing because the DNSSEC verification architecture follows the DNS delegation hierarchy, i.e., the zone structure. Witnesses including ORRs **120**, **125**, **130**, VRR **215**, and/or super zone recursive resolver **315** can vary based on suitability for different zones and features where a relying party can select among one or more resolvers on the basis of the zones and/or feature of interest. Witnesses may also be constructed that specialize in other features, such as TLSA records (where the recursive resolver processes evidence related to such records), or privacy (where the recursive resolver provides additional privacy protection for DNS transactions and data). The enhanced resolution process thus provides the benefit of specialization: a witness can be optimized for specific purposes, and a relying party can gain this advantage for multiple purposes by employing multiple specialized recursive resolvers.

FIG. **4** shows a method **400** of resolving DNS queries, according to examples of the present disclosure. The method begins at **405**. At **410**, a first DNS recursive resolver, such as resolver **115**, **205**, or **315**, obtains a first DNS query, such as query **110**, **310** from a requestor, such as computer **105**, **305**. For example, computer **105**, **305** issues a DNS query for a domain, such as www.example.com, with the DO bit set to 1 (asking for the answer using DNSSEC). At **415**, at least one hardware processor of the first DNS recursive resolver determines that the first DNS recursive resolver does not contain an answer to the first query stored in a memory. If the answer is not in the memory of the recursive resolver, the recursive resolver queries one or more authoritative name servers for the answer. The recursive resolver then receives an answer with a DNSSEC key.

If, for example, the DNSSEC key for example.com (the hosting zone) does not match the DNS record for that DNSSEC key (meaning the chain of trust for that zone is broken), the recursive resolver can poll one or more witnesses that have been previously vetted and their capabilities previously negotiated to discover the DNSSEC key that they have on record for that zone. At **420**, the first DNS recursive resolver provides one or more second queries to a respective one or more witnesses, i.e., second DNS recursive resolver(s), authoritative name server(s), etc. At **425**, the first DNS recursive resolver obtains an answer from the one or more witnesses. At **430**, the first DNS recursive resolver access from a memory a policy, wherein the policy specifies a type of associated evidence of correctness the requestor is willing to accept.

At **435**, at least one hardware processor of the first DNS recursive resolver determines an answer to the first DNS query based on the policy, the one or more answers from the one or more witnesses, and evidence of correctness associated with at least one of: the one or more witnesses and the one or more answers. The associated evidence of correctness includes one or more metrics comprising a reputation score associated with a second DNS recursive resolver, a comparison of a DNSKEY record associated with the domain name record with other DNSKEY records from other open or private DNS recursive resolvers, a chain of trust associated with the domain name record, WHOIS and/or registration data associated with the domain name, communications channel security indicators, a zone file, a zone modification request log, an error log, or a public ledger. At **440**, the first DNS recursive resolver provides the answer to the requestor. The answer can be also set with an AD bit set to 1, which would indicate that the policy has been met. At **445**, the method ends.

FIG. **5** shows a method **500** of processing queries at an authoritative name server **505**. A provisioning server **515** provisions **520** one or more DNS records to be served by the authoritative name server in response to DNS queries. The DNS records may include one or more DNSKEY records as part of a DNS chain of trust, and may be provided by the provisioning server **515** in the form of zone file updates to the authoritative name server **505**. Authoritative name server **505** is thereby equipped to respond to DNS queries, e.g., resolver **525** can send DNS query **530** and authoritative name server **505** can return response **570**, according to the DNS records/zone file provided to authoritative name server **505** by provisioning server **515**. Authoritative name server **505** can obtain and optionally provide one or more validations for the answer **570** to query **530**, and/or for its own assurance of the accuracy of the responses **570** it provides. The first validation **540** can be for a chain of trust validation using DNSSEC **545**.

As a remediation to the possibility of instability in the provisioning system for zone file updates and/or to the possibility of inaccuracies at other authoritative name servers provisioned by provisioning server **515**, and/or according to its local policy, authoritative name server **505** may also employ evidence-based validation as described herein, and/or evidence-based DNS resolution more generally, to increase its assurance of the correctness and appropriateness of the DNS records it is serving in response to queries. For example, provisioning server **515** may have had an outage and/or may have been compromised, and may have not provided DNS records including associated DNSSEC records accurately, in a timely manner, or otherwise in compliance with an external process, and/or consistent with obligations of the operator of the authoritative server to serve DNS records to its community of requesters (which may be local, regional, or global). In particular, authoritative name server **505** may itself consult one or more witnesses and/or obtain provenance, as supplementary evidence, e.g., authoritative name server **505** may obtain, validate and optionally provide provenance in the form of a second validation **550** that can include other verification **555** including, but are not limited to, WHOIS/registration data associated with a domain name (to determine, e.g., changes of registrant and/or registrar that might affect the stability of the associated zone file); whether responses are delivered over a secure communications channel, zone files, zone modification request logs, error logs, and public ledgers, etc. Authoritative name server **505** can obtain, validate and optionally provide provenance in the form of a third vali-

dation **560** that can include public data polling information **565**. The various provenance, which may be considered in combination, may thus assist authoritative name server **505** and possibly resolver **525** in its processing of a DNS query. Authoritative name server **505** may thereby offer higher assurance than other authoritative name servers provisioned for the same zone, and/or complementary assurance based on the diversity of its sources of evidence.

In some examples, prior to the first DNS recursive resolver providing the plurality of second queries, the first DNS resolver can provide a third query to an authoritative name server, where the first DNS resolver can obtain an answer from the authoritative name server and determine that the answer is not secured using DNSSEC.

In some examples, the first DNS recursive resolver can rank a plurality of second DNS recursive resolvers based on reputation and select the one or more second DNS recursive resolver to use to answer the first DNS query based on the ranking.

In some examples, the first DNS recursive resolver can provide a request to a trusted third party for a list of second DNS recursive resolvers, obtain the list from the trusted third party; and select the one or more second DNS recursive resolvers from the list. In some examples, the first DNS recursive resolver can determine that a particular second DNS recursive resolver from the one or more of DNS recursive resolvers is better at a particular task and/or particular zone than others of the plurality of second DNS recursive resolvers and select the one or more second DNS recursive resolvers to use to answer the first DNS query based on the determining.

In some examples, the first DNS recursive resolver can provide the evidence of correctness associated with the one or more second DNS recursive resolvers and/or the one or more answers to the requestor. For example, each witness can provide to the first DNS recursive resolver a DNSSEC key query, the DNS response, and a timestamp for each. The first DNS recursive resolver can provide this information from one or more of the witnesses along with the source IP address based on the queries to the witnesses to the requestor as the evidence and proof of the correctness of the answer. The policy associated with the requestor at the first DNS recursive resolver can set a particular threshold of witnesses needed for an answer to be determined to be correct. For example, if ⅔ of the witnesses return the same DNSSEC key, then the answer will be assumed to be correct. This threshold can be changed based on the degree of tolerance that the requestor is willing to accept, and the degree of trust it places in the selected witnesses. If the requestor is not willing to accept the possibility that an answer is being spoofed, then the policy can set so that all the polled witnesses have to agree on the DNSSEC key.

FIG. **6** illustrates an example of a hardware configuration for a computer device **600** that can be used as mobile device or server, which can be used to perform one or more of the processes described above. While FIG. **6** illustrates various components contained in the computer device **600**, FIG. **6** illustrates one example of a computer device and additional components can be added and existing components can be removed.

The computer device **600** can be any type of computer devices, such as desktops, laptops, servers, DNS server, etc., or mobile devices, such as smart telephones, tablet computers, cellular telephones, personal digital assistants, etc. As illustrated in FIG. **6**, the computer device **600** can include one or more processors **602** of varying core configurations and clock frequencies. The computer device **600** can also include one or more memory devices **504** that serve as a main memory during the operation of the computer device **600**. For example, during operation, a copy of the software that supports the DNS operations can be stored in the one or more memory devices **604**. The computer device **600** can also include one or more peripheral interfaces **606**, such as keyboards, mice, touchpads, computer screens, touchscreens, etc., for enabling human interaction with and manipulation of the computer device **600**.

The computer device **600** can also include one or more network interfaces **608** for communicating via one or more networks, such as Ethernet adapters, wireless transceivers, or serial network components, for communicating over wired or wireless media using protocols. The computer device **600** can also include one or more storage device **610** of varying physical dimensions and storage capacities, such as flash drives, hard drives, random access memory, etc., for storing data, such as images, files, and program instructions for execution by the one or more processors **602**.

Additionally, the computer device **600** can include one or more software programs **612** that enable the functionality described above. The one or more software programs **612** can include instructions that cause the one or more processors **602** to perform the processes described herein. Copies of the one or more software programs **612** can be stored in the one or more memory devices **604** and/or on in the one or more storage devices **610**. Likewise, the data, for example, the super zone data, utilized by one or more software programs **612** can be stored in the one or more memory devices **604** and/or on in the one or more storage devices **610**.

In implementations, the computer device **600** can communicate with other devices via a network **616**. The other devices can be any types of devices as described above. The network **616** can be any type of electronic network, such as a local area network, a wide-area network, a virtual private network, the Internet, an intranet, an extranet, a public switched telephone network, an infrared network, a wireless network, and any combination thereof. The network **616** can support communications using any of a variety of commercially-available protocols, such as TCP/IP, UDP, OSI, FTP, UPnP, NFS, CIFS, AppleTalk, and the like. The network **616** can be, for example, a local area network, a wide-area network, a virtual private network, the Internet, an intranet, an extranet, a public switched telephone network, an infrared network, a wireless network, and any combination thereof.

The computer device **600** can include a variety of data stores and other memory and storage media as discussed above. These can reside in a variety of locations, such as on a storage medium local to (and/or resident in) one or more of the computers or remote from any or all of the computers across the network. In some implementations, information can reside in a storage-area network ("SAN") familiar to those skilled in the art. Similarly, any necessary files for performing the functions attributed to the computers, servers, or other network devices may be stored locally and/or remotely, as appropriate.

In implementations, the components of the computer device **600** as described above need not be enclosed within a single enclosure or even located in close proximity to one another. Those skilled in the art will appreciate that the above-described componentry are examples only, as the computer device **600** can include any type of hardware componentry, including any necessary accompanying firmware or software, for performing the disclosed implementations. The computer device **600** can also be implemented

in part or in whole by electronic circuit components or processors, such as application-specific integrated circuits (ASICs) or field-programmable gate arrays (FPGAs).

If implemented in software, the functions can be stored on or transmitted over a computer-readable medium as one or more instructions or code. Computer-readable media includes both tangible, non-transitory computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A storage media can be any available tangible, non-transitory media that can be accessed by a computer. By way of example, and not limitation, such tangible, non-transitory computer-readable media can comprise RAM, ROM, flash memory, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code in the form of instructions or data structures and that can be accessed by a computer. Disk and disc, as used herein, includes CD, laser disc, optical disc, DVD, floppy disk and Blu-ray disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Also, any connection is properly termed a computer-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Combinations of the above should also be included within the scope of computer-readable media.

The foregoing description is illustrative, and variations in configuration and implementation can occur to persons skilled in the art. For instance, the various illustrative logics, logical blocks, modules, and circuits described in connection with the embodiments disclosed herein can be implemented or performed with a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor can be a microprocessor, but, in the alternative, the processor can be any conventional processor, controller, microcontroller, or state machine. A processor can also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

In one or more exemplary embodiments, the functions described can be implemented in hardware, software, firmware, or any combination thereof. For a software implementation, the techniques described herein can be implemented with modules (e.g., procedures, functions, subprograms, programs, routines, subroutines, modules, software packages, classes, and so on) that perform the functions described herein. A module can be coupled to another module or a hardware circuit by passing and/or receiving information, data, arguments, parameters, or memory contents. Information, arguments, parameters, data, or the like can be passed, forwarded, or transmitted using any suitable means including memory sharing, message passing, token passing, network transmission, and the like. The software codes can be stored in memory units and executed by processors. The memory unit can be implemented within the processor or external to the processor, in which case it can be communicatively coupled to the processor via various means as is known in the art.

While the teachings have been described with reference to examples of the implementations thereof, those skilled in the art will be able to make various modifications to the described implementations without departing from the true spirit and scope. The terms and descriptions used herein are set forth by way of illustration only and are not meant as limitations. In particular, although the processes have been described by examples, the stages of the processes can be performed in a different order than illustrated or simultaneously. Furthermore, to the extent that the terms "including", "includes", "having", "has", "with", or variants thereof are used in the detailed description, such terms are intended to be inclusive in a manner similar to the term "comprising." As used herein, the terms "one or more of" and "at least one of" with respect to a listing of items such as, for example, A and B, means A alone, B alone, or A and B. Further, unless specified otherwise, the term "set" should be interpreted as "one or more." Also, the term "couple" or "couples" is intended to mean either an indirect or direct connection. Thus, if a first device couples to a second device, that connection can be through a direct connection, or through an indirect connection via other devices, components, and connections.

Those skilled in the art will be able to make various modifications to the described embodiments without departing from the true spirit and scope. The terms and descriptions used herein are set forth by way of illustration only and are not meant as limitations. In particular, although the method has been described by examples, the steps of the method can be performed in a different order than illustrated or simultaneously. Those skilled in the art will recognize that these and other variations are possible within the spirit and scope as defined in the following claims and their equivalents.

The foregoing description of the disclosure, along with its associated embodiments, has been presented for purposes of illustration only. It is not exhaustive and does not limit the disclosure to the precise form disclosed. Those skilled in the art will appreciate from the foregoing description that modifications and variations are possible in light of the above teachings or may be acquired from practicing the disclosure. For example, the steps described need not be performed in the same sequence discussed or with the same degree of separation. Likewise various steps may be omitted, repeated, or combined, as necessary, to achieve the same or similar objectives. Similarly, the systems described need not necessarily include all parts described in the embodiments, and may also include other parts not describe in the embodiments.

Accordingly, the disclosure is not limited to the above-described embodiments, but instead is defined by the appended claims in light of their full scope of equivalents.

What is claimed is:

1. A computer-implemented method of resolving one or more queries for a domain name record, the method comprising:

receiving, from a requestor, a first query for the domain name record;

determining, by a hardware processor of a computer, to obtain evidence of correctness for a first answer to the first query;

obtaining, by the computer, a first evidence of correctness from a first witness, wherein the first answer is validated based on the first evidence of correctness; and

sending, to the requestor, a second answer to the first query, wherein the second answer is based on the first answer and the first evidence of correctness,

wherein the computer comprises: a DNS recursive resolver, a non-recursive resolver, a non-resolver witness, a stub resolver, or an authoritative name server.

2. The computer-implemented method of claim **1**, wherein determining to obtain evidence of correctness for the first answer comprises: determining, by the computer, that a DNSSEC key does not match a record for the DNSSEC key or the first answer is not secured using DNSSEC.

3. The computer-implemented method of claim **2**, wherein validation of the first answer based on the first evidence of correctness provides the same assurance as a DNSSEC key that matches the record for the DNSSEC key.

4. The computer-implemented method of claim **1**, further comprising:

determining, by the hardware processor of the computer, a plurality of witnesses having one or more validation features; and

maintaining a list of the plurality of witnesses.

5. The computer-implemented method of claim **4**, further comprising:

selecting, by the computer and based on the first query, the first witness from the plurality of witnesses, wherein the first witness is selected based on one or more validation features of the first witness.

6. The computer-implemented method of claim **5**, wherein the one or more validation features of the first witness comprise at least one of: one or more metrics comprising a reputation score associated with a resolver; a comparison of a DNSKEY record associated with the domain name record with other DNSKEY records from a resolver; a chain of trust associated with the domain name record; a WHOIS and/or registration data associated with the domain name record; a communications channel security indicator; a zone file; a zone modification request log; an error log; or a public ledger.

7. The computer-implemented method of claim **1**, wherein obtaining the first evidence of correctness from the first witness comprises: determining the first answer was obtained over a secure communications channel.

8. The computer-implemented method of claim **1**, further comprising: wherein determining to obtain evidence of correctness for the first answer comprises: receiving, by the computer, a request to provide DNSSEC validation or provenance.

9. The computer-implemented method of claim **1**, wherein the first evidence of correctness is stored at a location where the first evidence of correctness can be obtained.

10. The computer-implemented method of claim **1**, wherein the second answer comprises the first answer and the first evidence of correctness.

11. A computer system for resolving one or more queries for a domain name record, the system comprising:

one or more processors; and

a memory system comprising one or more non-transitory computer-readable media storing instructions that, when executed by the at least one of the one or more processors, cause the one or more processors to perform operations comprising:

receiving, from a requestor, a first query for the domain name record;

determining, by the computer system, to obtain evidence of correctness for a first answer to the first query;

obtaining, by the computer system, a first evidence of correctness from a first witness, wherein the first answer is validated based on the first evidence of correctness; and

sending, to the requestor, a second answer to the first query, wherein the second answer is based on the first answer and the first evidence of correctness,

wherein the computer system comprises: a DNS recursive resolver, a non-recursive resolver, a non-resolver witness, a stub resolver, or an authoritative name server.

12. The computer system of claim **11**, wherein determining to obtain evidence of correctness for the first answer comprises: determining, by the computer system, that a DNSSEC key does not match a record for the DNSSEC key or the first answer is not secured using DNSSEC.

13. The computer system of claim **12**, wherein validation of the first answer based on the first evidence of correctness provides the same assurance as a DNSSEC key that matches the record for the DNSSEC key.

14. The computer system of claim **11**, further comprising:

determining, by the computer system, a plurality of witnesses having one or more validation features; and

maintaining a list of the plurality of witnesses.

15. The computer system of claim **14**, further comprising:

selecting, by the computer system and based on the first query, the first witness from the plurality of witnesses, wherein the first witness is selected based on one or more validation features of the first witness.

16. The computer system of claim **15**, wherein the one or more validation features of the first witness comprise at least one of: one or more metrics comprising a reputation score associated with a resolver; a comparison of a DNSKEY record associated with the domain name record with other DNSKEY records from a resolver; a chain of trust associated with the domain name record; a WHOIS and/or registration data associated with the domain name record; a communications channel security indicator; a zone file; a zone modification request log; an error log; or a public ledger.

17. The computer system of claim **11**, wherein obtaining the first evidence of correctness from the first witness comprises: determining the first answer was obtained over a secure communications channel.

18. The computer system of claim **11**, further comprising: wherein determining to obtain evidence of correctness for the first answer comprises: receiving, by the computer system, a request to provide DNSSEC validation or provenance.

19. The computer system of claim **11**, wherein the first evidence of correctness is stored at a location where the first evidence of correctness can be obtained.

20. The computer system of claim **11**, wherein the second answer comprises the first answer and the first evidence of correctness.

* * * * *