US012395495B2

(12) **United States Patent**
Ezrielev et al.

(10) **Patent No.: US 12,395,495 B2**
(45) **Date of Patent: Aug. 19, 2025**

(54) **SYSTEM AND METHOD FOR RESECURING DISTRIBUTED SYSTEM RESPONSIVE TO COMPROMISE EVENT**

(71) Applicant: **Dell Products L.P.**, Round Rock, TX (US)

(72) Inventors: **Ofir Ezrielev**, Be'er Sheva (IL); **Yehiel Zohar**, Sderot (IL); **Lee Serfaty**, Be'er Sheva (IL)

(73) Assignee: **Dell Products L.P.**, Round Rock, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 267 days.

(21) Appl. No.: **18/308,241**

(22) Filed: **Apr. 27, 2023**

(65) **Prior Publication Data**

US 2024/0364698 A1 Oct. 31, 2024

(51) **Int. Cl.**
H04L 9/40 (2022.01)

(52) **U.S. Cl.**
CPC ***H04L 63/105*** (2013.01); ***H04L 63/0823*** (2013.01)

(58) **Field of Classification Search**
USPC 726/4
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,298,704 B2 11/2007 Kodialam
8,336,100 B1 12/2012 Glick
9,740,859 B2 8/2017 Harris
10,735,203 B2 8/2020 Reddy
11,438,369 B2 9/2022 Schwartau
11,886,557 B1 1/2024 Thanh Tran
11,973,878 B2 4/2024 Young
2009/0086977 A1 4/2009 Berggren
2013/0347094 A1 12/2013 Bettini
2014/0359281 A1 12/2014 Saboori
(Continued)

FOREIGN PATENT DOCUMENTS

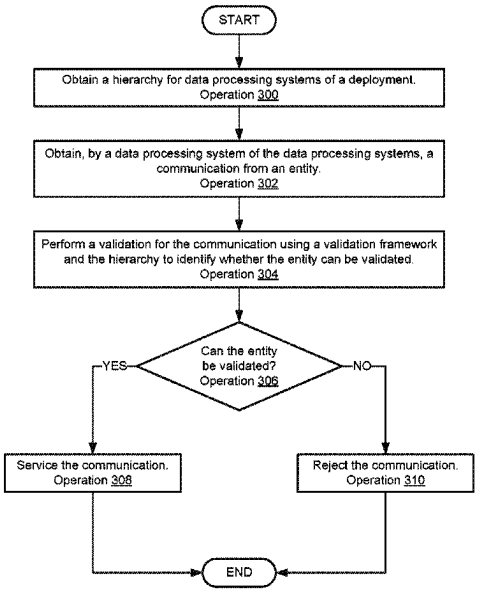
CN 112861106 A 5/2021
CN 117082515 A 11/2023
(Continued)

Primary Examiner — Sanchit K Sarker
(74) *Attorney, Agent, or Firm* — WOMBLE BOND DICKINSON (US) LLP

(57) **ABSTRACT**

Methods and systems for securing distributed systems are disclosed. The distributed systems may include data processing systems subject to compromise by malicious entities. If compromised, the data processing systems may impair the services provided by the distributed system. To secure the distributed systems, the data processing systems may implement a security framework. The security framework may utilize a hierarchy that defines authority for validating trusted entities. The hierarchy may vest authority across the distributed system, and may be based on a reputation (e.g., weighted reputation) of each of the data processing systems within the distributed system. The hierarchy may be dynamically updated over time as new information regarding data processing systems is discovered. If the reputation of a data processing system meets criteria, that data processing system may be treated as being compromised. Consequently, the impact of compromise of the data processing system may be limited by the distributed authority.

20 Claims, 16 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

2016/0191253	A1	6/2016	Pyle	
2016/0350531	A1*	12/2016	Harris	H04L 63/20
2017/0005805	A1	1/2017	Wang	
2017/0331575	A1	11/2017	Ruffini	
2018/0069849	A1	3/2018	Kraemer	
2019/0109717	A1*	4/2019	Reddy	H04L 63/1433
2019/0312906	A1	10/2019	Schwartau	
2020/0107201	A1	4/2020	Sinha	
2020/0136838	A1*	4/2020	Kucharski	H04L 9/3265
2023/0179422	A1	6/2023	Young	
2023/0336581	A1	10/2023	Dunn	
2024/0137358	A1*	4/2024	Nambannor Kunnath	H04L 63/0823
2024/0146756	A1*	5/2024	Debenedetti	H04L 63/1433
2024/0333730	A1	10/2024	Wang	
2024/0364676	A1	10/2024	Ezrielev	
2024/0364698	A1	10/2024	Ezrielev	
2024/0364752	A1	10/2024	Ezrielev	

FOREIGN PATENT DOCUMENTS

CN	117354023	A	1/2024
JP	2004179724	A *	6/2004

* cited by examiner

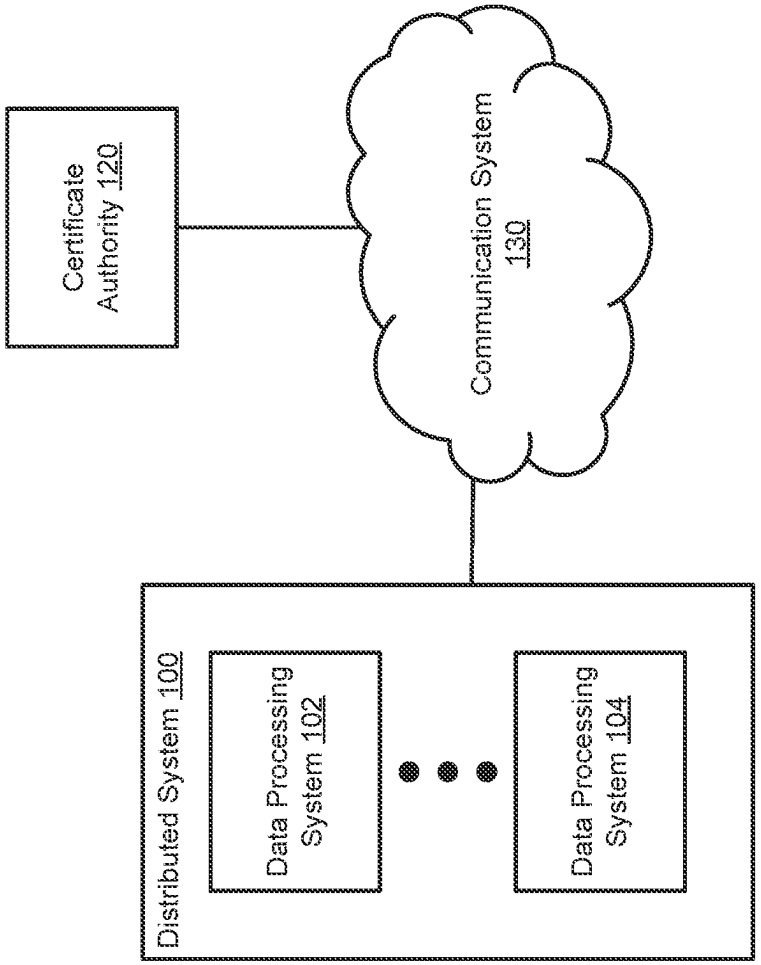

Certificate Authority 120
Communication System 130
Distributed System 100
Data Processing System 102
Data Processing System 104

FIG. 1

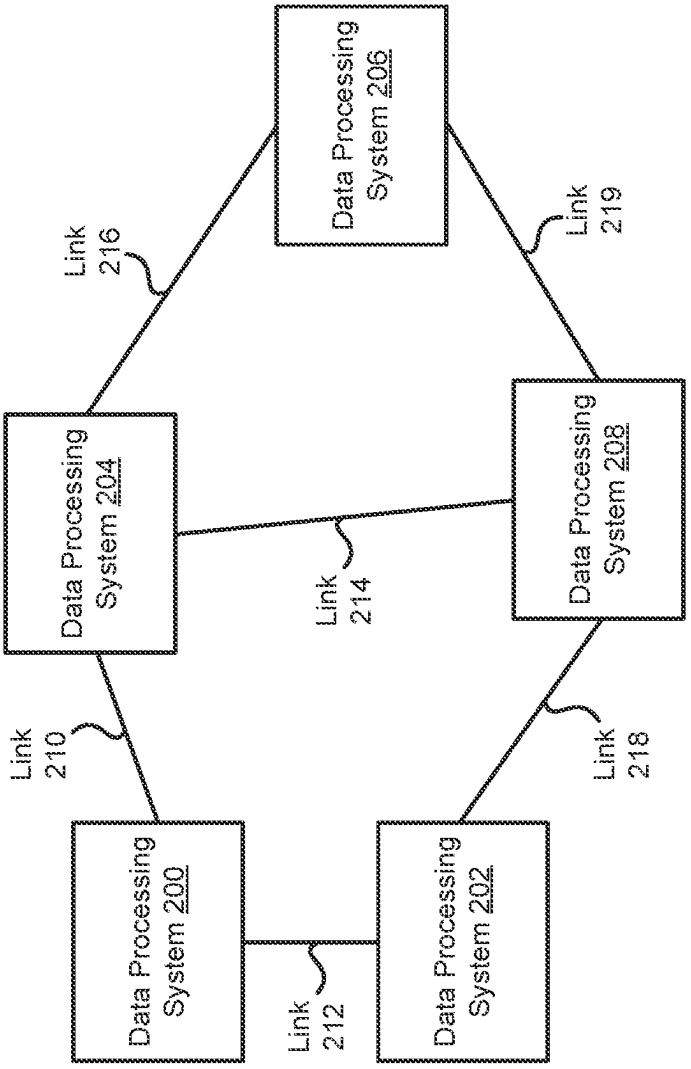

Data Processing System 206
Link 216
Link 219
Data Processing System 204
Data Processing System 208
Link 214
Link 210
Link 218
Data Processing System 200
Data Processing System 202
Link 212

FIG. 2A

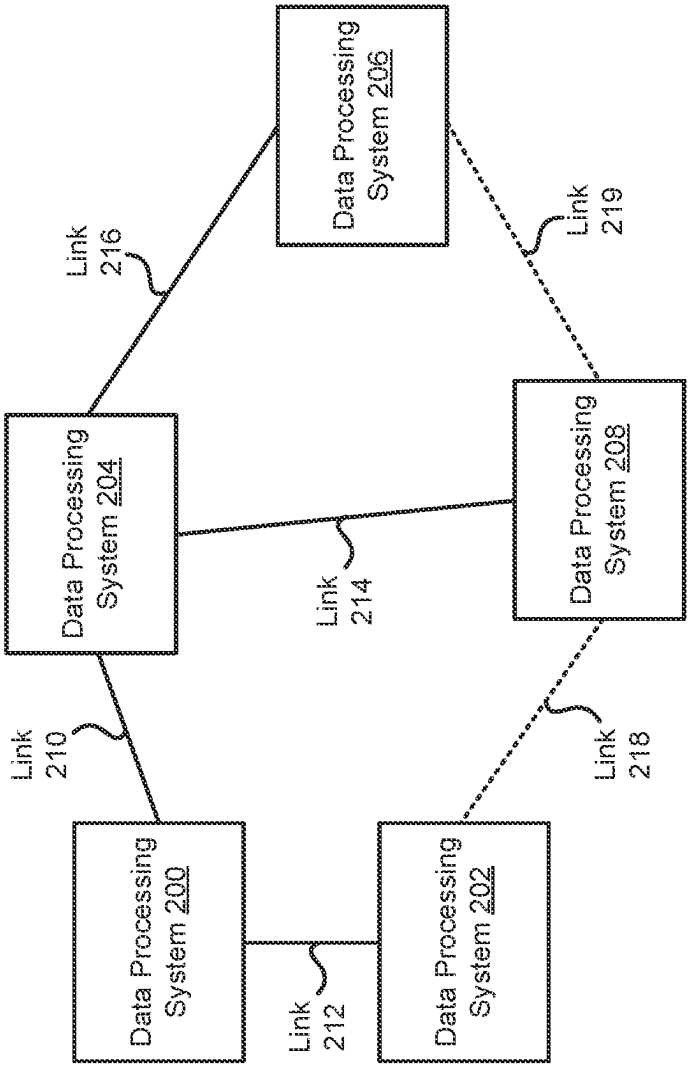

Data Processing System 206
Link 216
Link 219
Data Processing System 204
Data Processing System 208
Link 214
Link 210
Link 218
Data Processing System 200
Data Processing System 202
Link 212

FIG. 2B

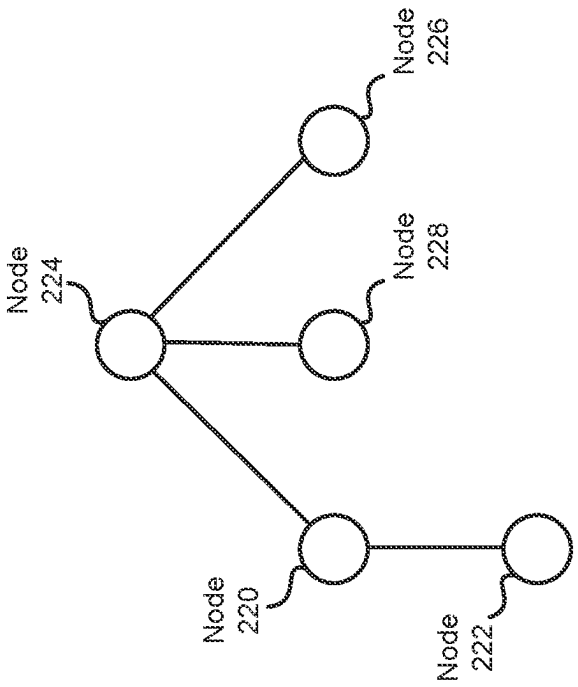


FIG. 2C

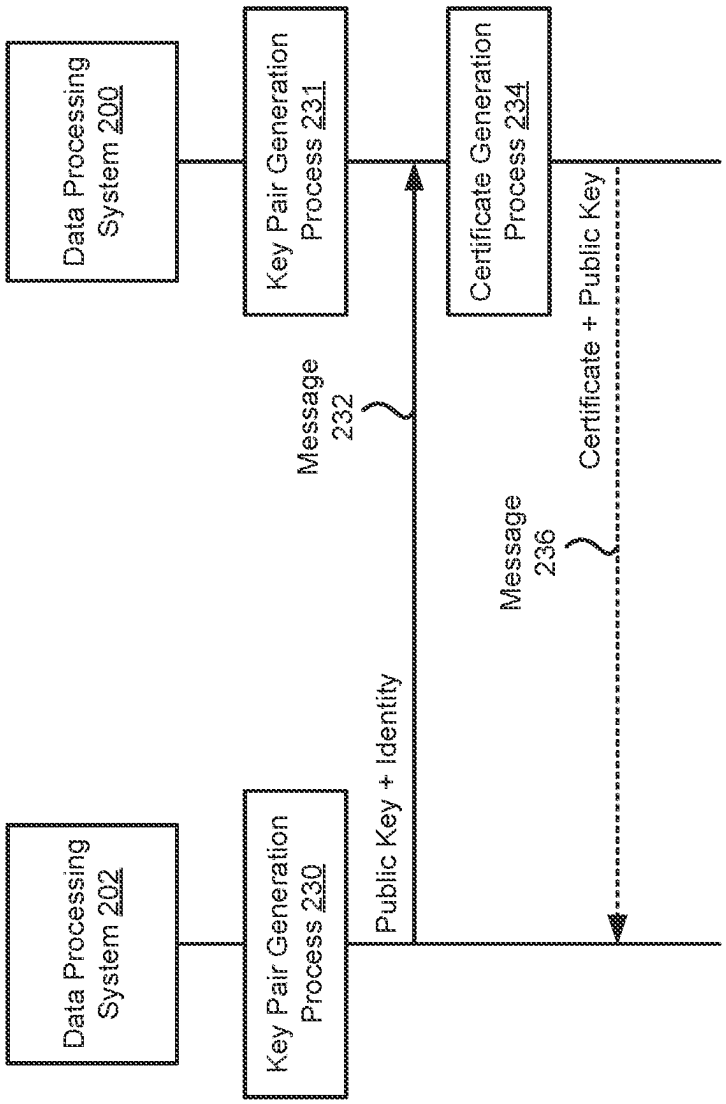

Data Processing System 202
Key Pair Generation Process 230
Public Key + Identity
Message 232
Data Processing System 200
Key Pair Generation Process 231
Certificate Generation Process 234
Message 236
Certificate + Public Key

FIG. 2D

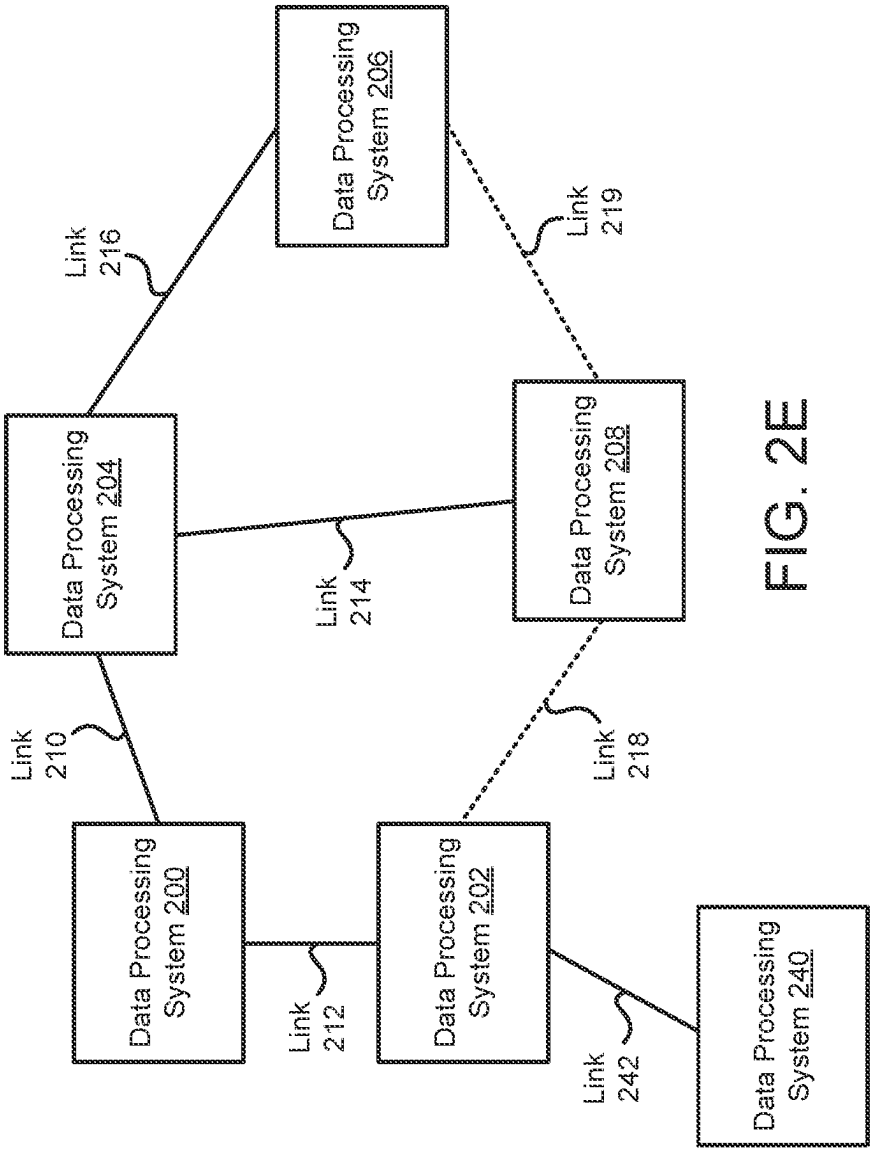

Data Processing System 206
Link 216
Link 219
Data Processing System 204
Data Processing System 208
Link 214
Link 210
Link 218
Data Processing System 200
Data Processing System 202
Link 212
Link 242
Data Processing System 240

FIG. 2E

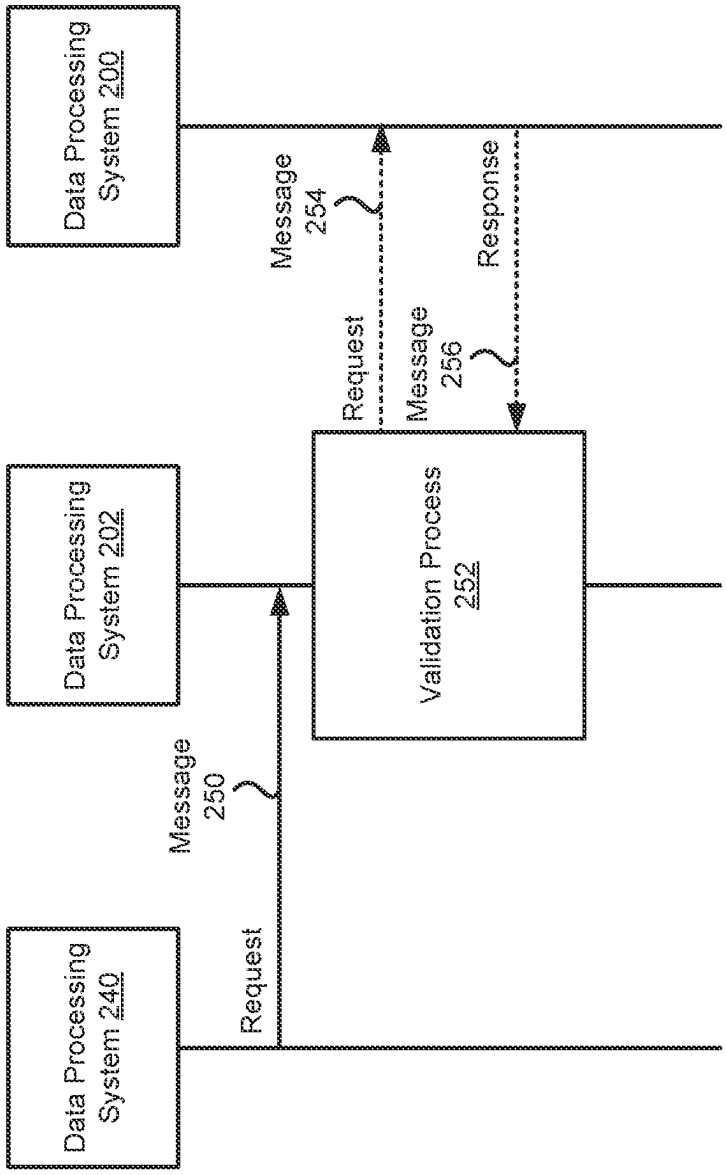

Data Processing System 200
Message 254
Response
Request Message 256
Data Processing System 202
Validation Process 252
Message 250
Request
Data Processing System 240

FIG. 2F

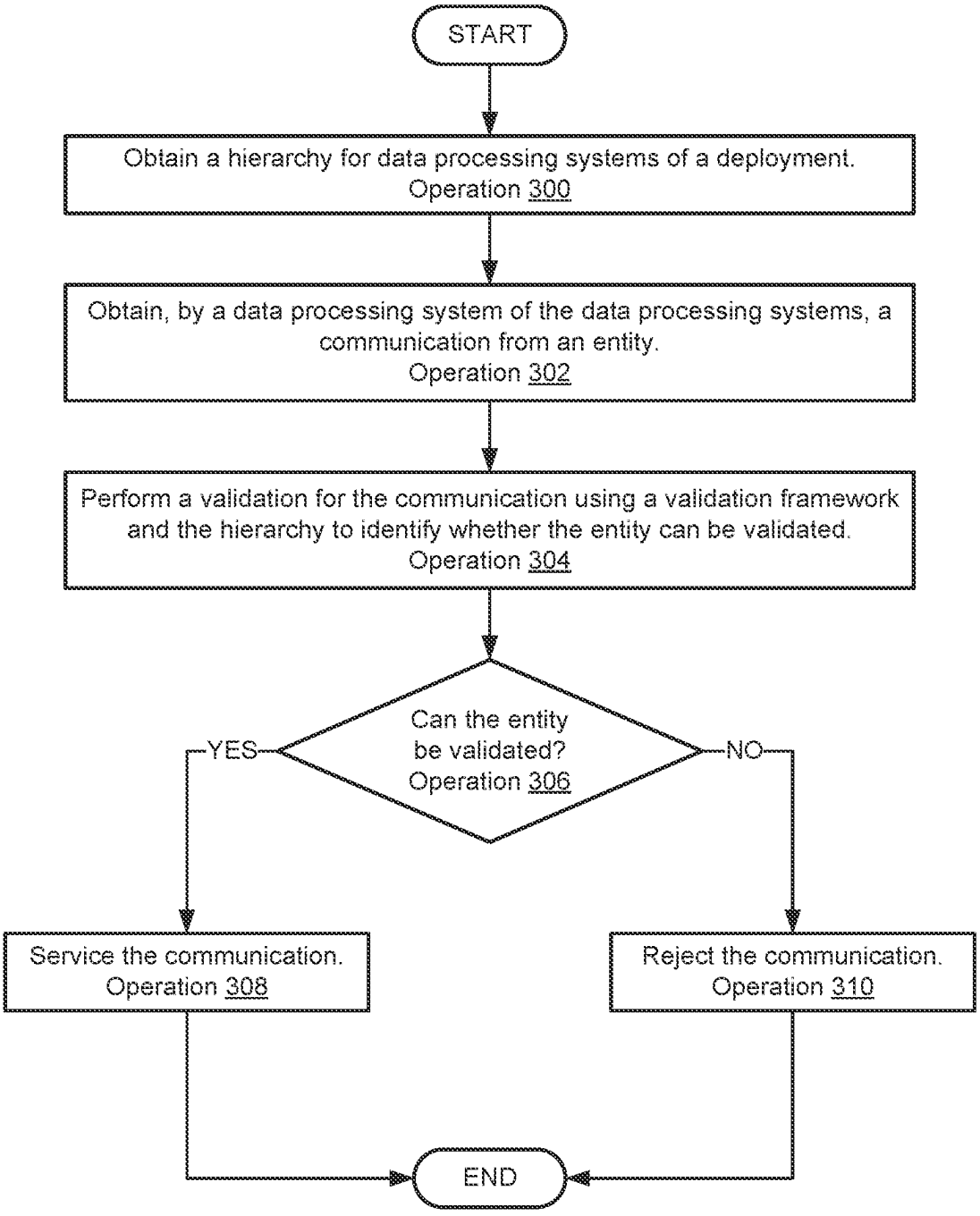

START
Obtain a hierarchy for data processing systems of a deployment.
Operation 300
Obtain, by a data processing system of the data processing systems, a communication from an entity.
Operation 302
Perform a validation for the communication using a validation framework and the hierarchy to identify whether the entity can be validated.
Operation 304
Can the entity be validated?
Operation 306
YES
NO
Service the communication.
Operation 308
Reject the communication.
Operation 310
END

FIG. 3

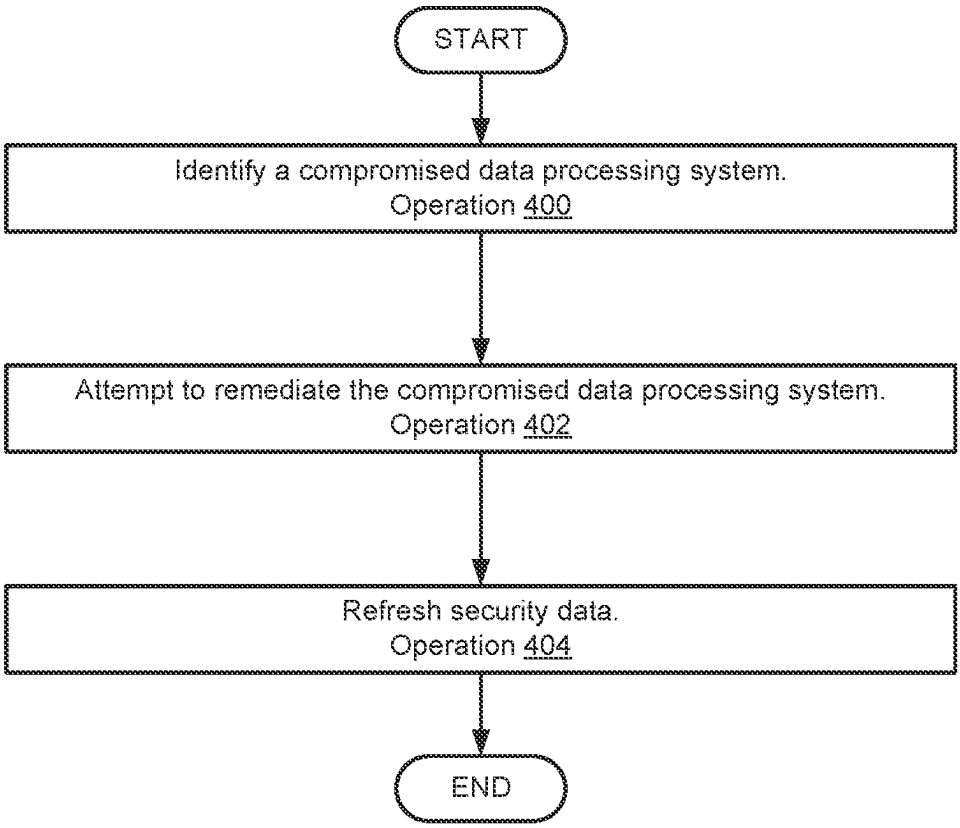

START
Identify a compromised data processing system.
Operation 400
Attempt to remediate the compromised data processing system.
Operation 402
Refresh security data.
Operation 404
END

FIG. 4

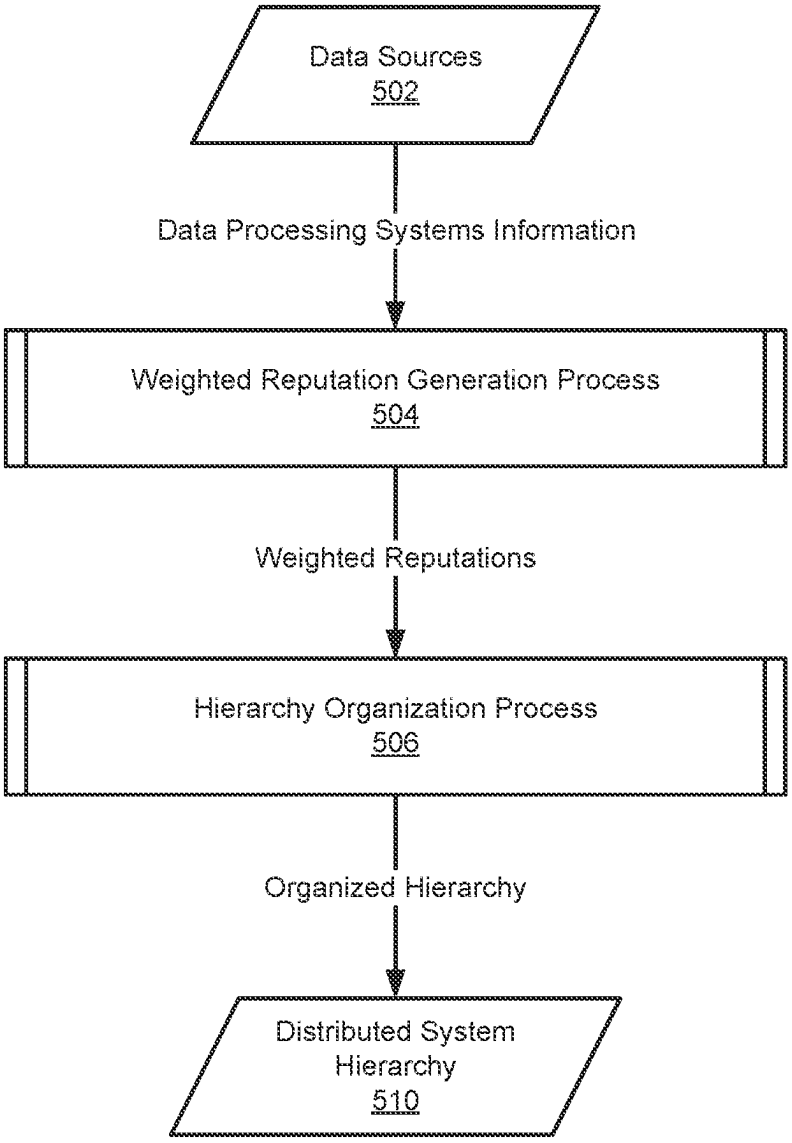

Data Sources
502
Data Processing Systems Information
Weighted Reputation Generation Process
504
Weighted Reputations
Hierarchy Organization Process
506
Organized Hierarchy
Distributed System
Hierarchy
510

FIG. 5

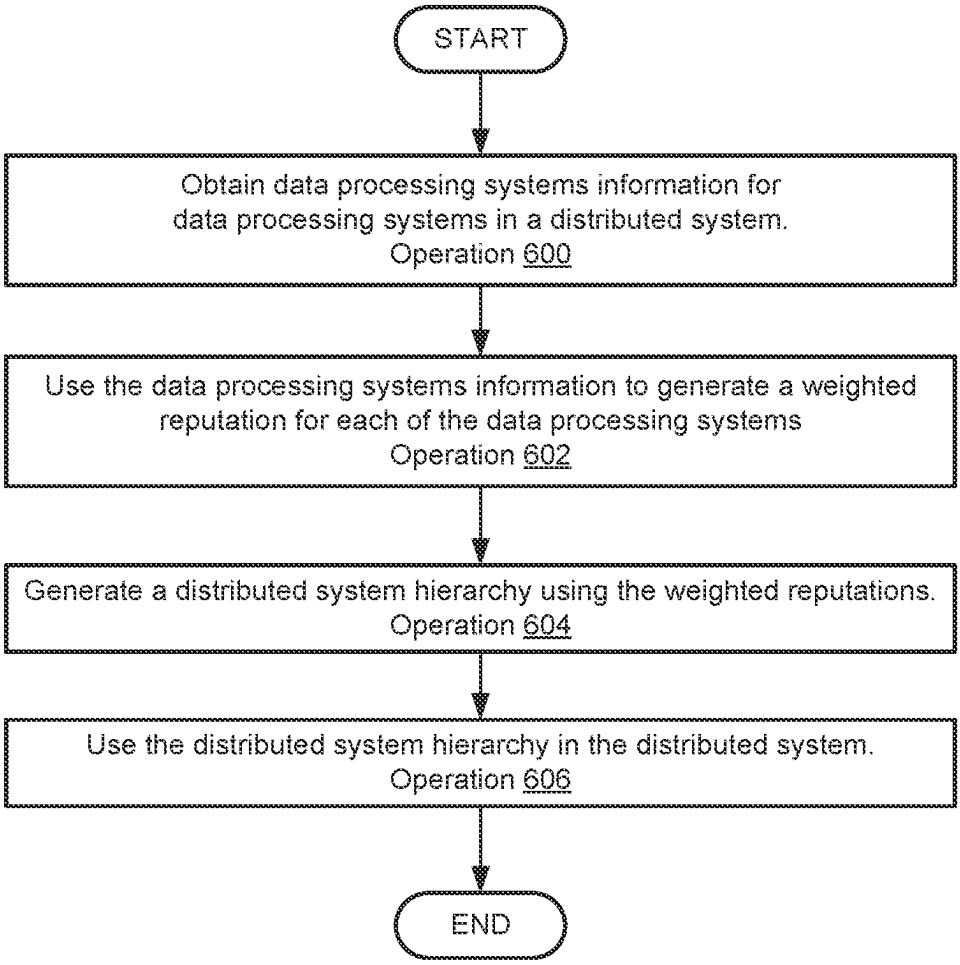

START
Obtain data processing systems information for data processing systems in a distributed system. Operation 600
Use the data processing systems information to generate a weighted reputation for each of the data processing systems Operation 602
Generate a distributed system hierarchy using the weighted reputations. Operation 604
Use the distributed system hierarchy in the distributed system. Operation 606
END

FIG. 6

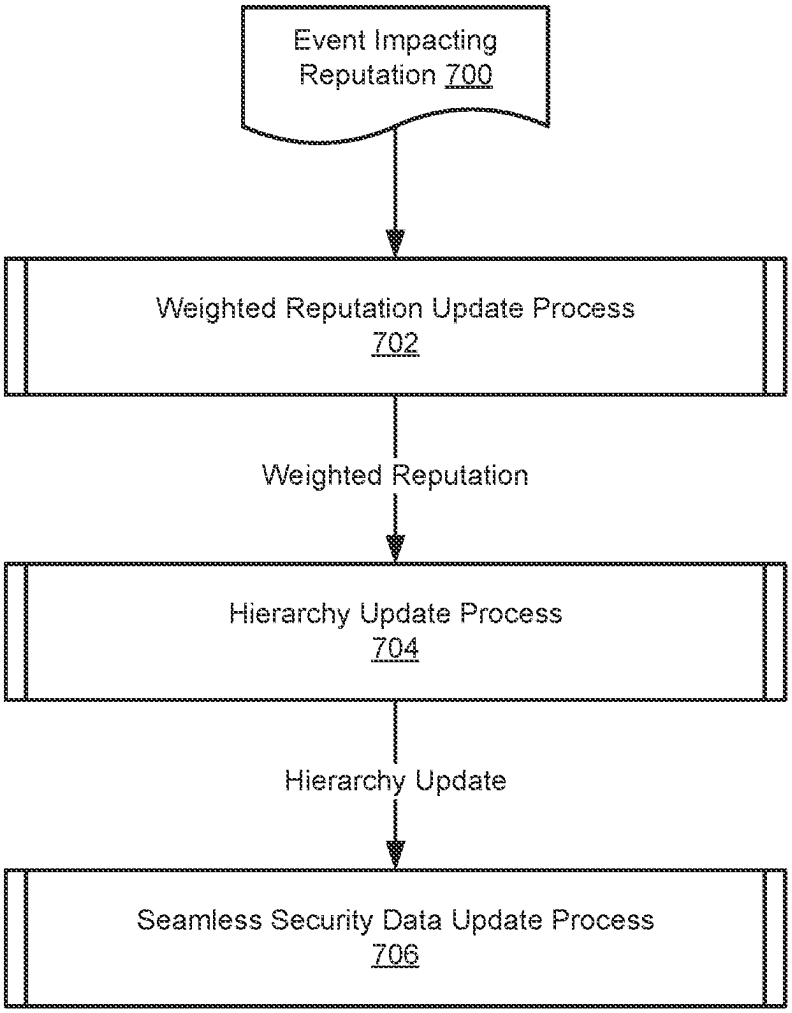

Event Impacting Reputation 700
Weighted Reputation Update Process 702
Weighted Reputation
Hierarchy Update Process 704
Hierarchy Update
Seamless Security Data Update Process 706

FIG. 7

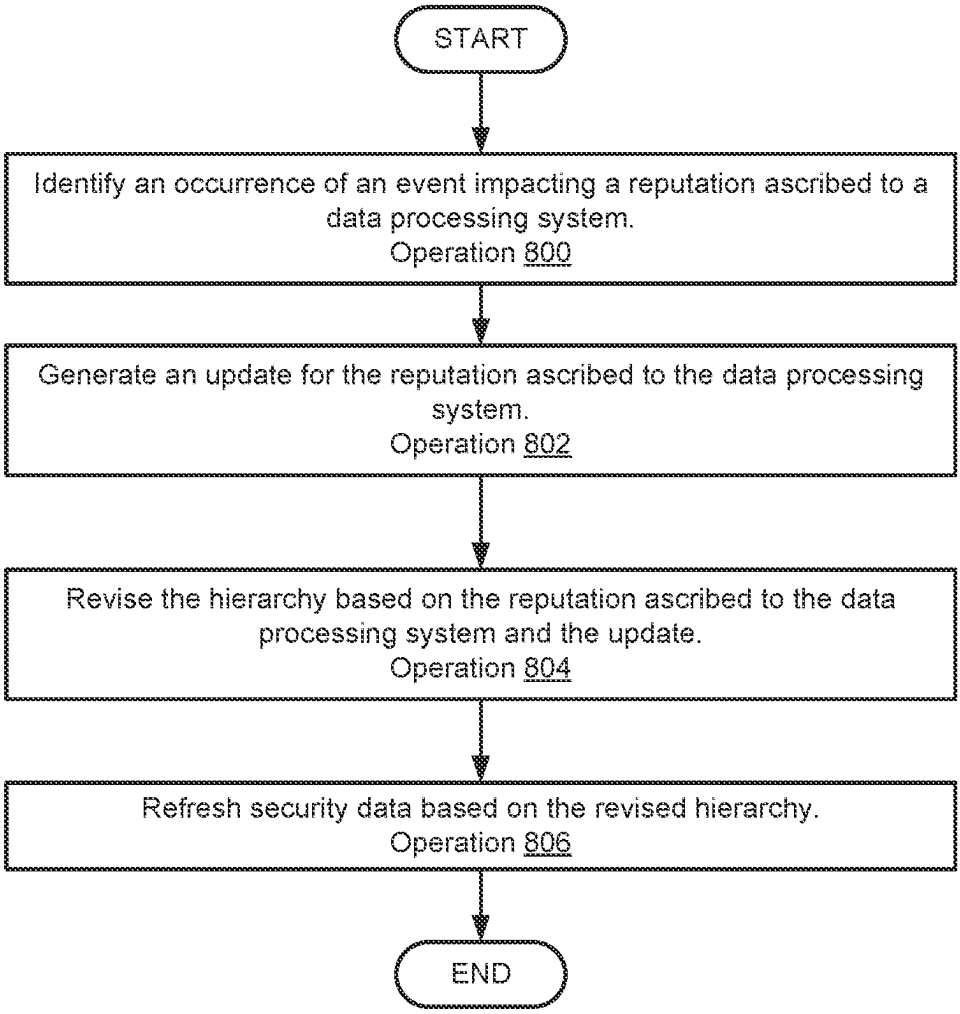

START
Identify an occurrence of an event impacting a reputation ascribed to a data processing system.
Operation 800
Generate an update for the reputation ascribed to the data processing system.
Operation 802
Revise the hierarchy based on the reputation ascribed to the data processing system and the update.
Operation 804
Refresh security data based on the revised hierarchy.
Operation 806
END

FIG. 8

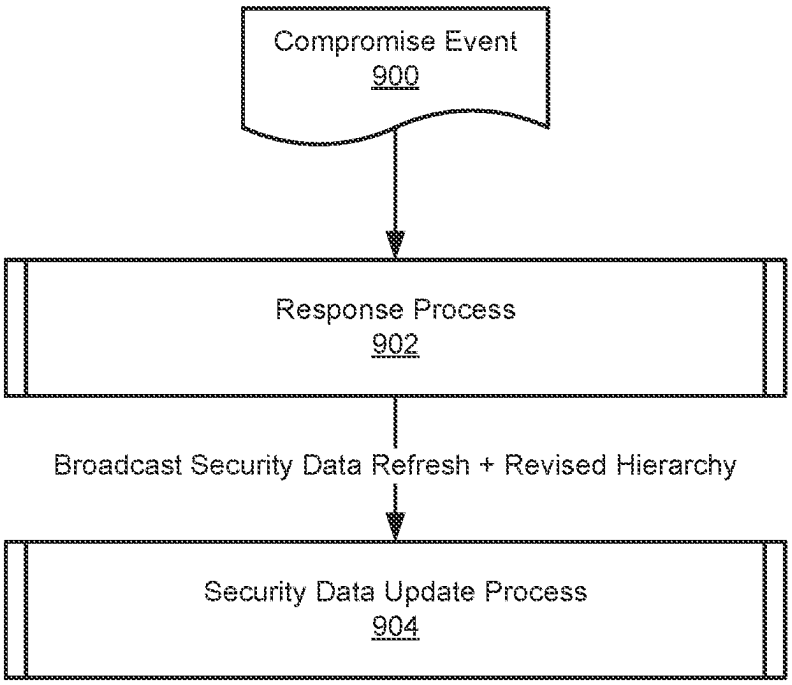

Compromise Event
900
Response Process
902
Broadcast Security Data Refresh + Revised Hierarchy
Security Data Update Process
904

FIG. 9

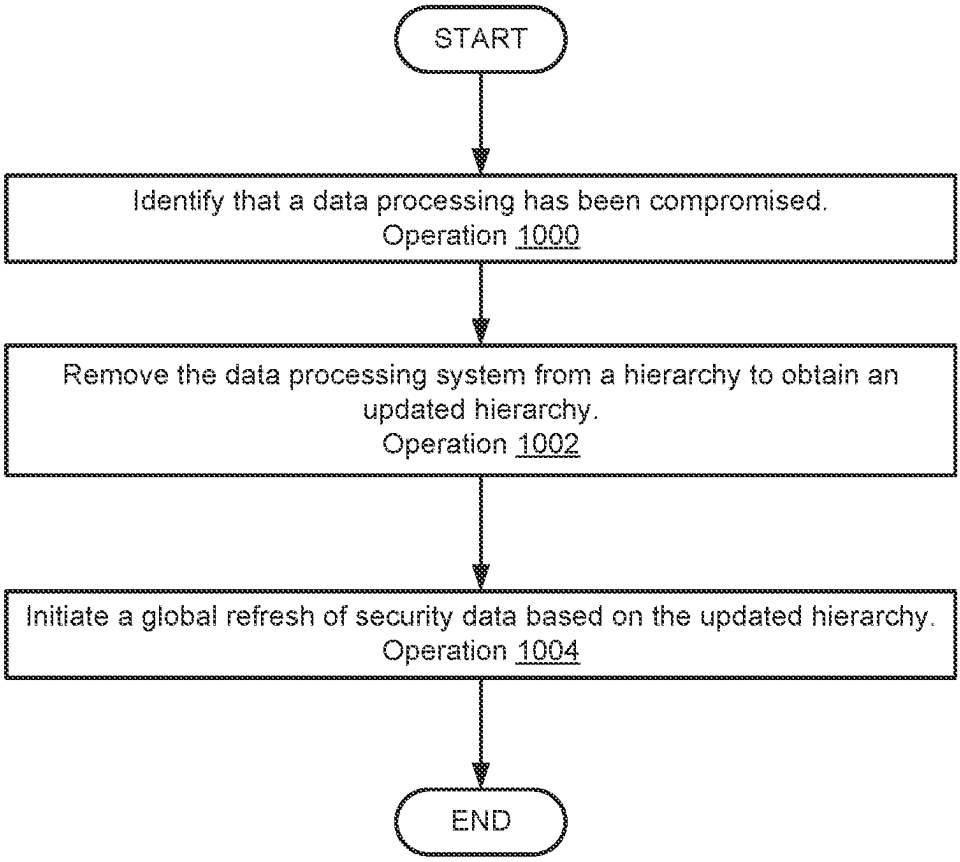

START
Identify that a data processing has been compromised.
Operation 1000
Remove the data processing system from a hierarchy to obtain an updated hierarchy.
Operation 1002
Initiate a global refresh of security data based on the updated hierarchy.
Operation 1004
END

FIG. 10

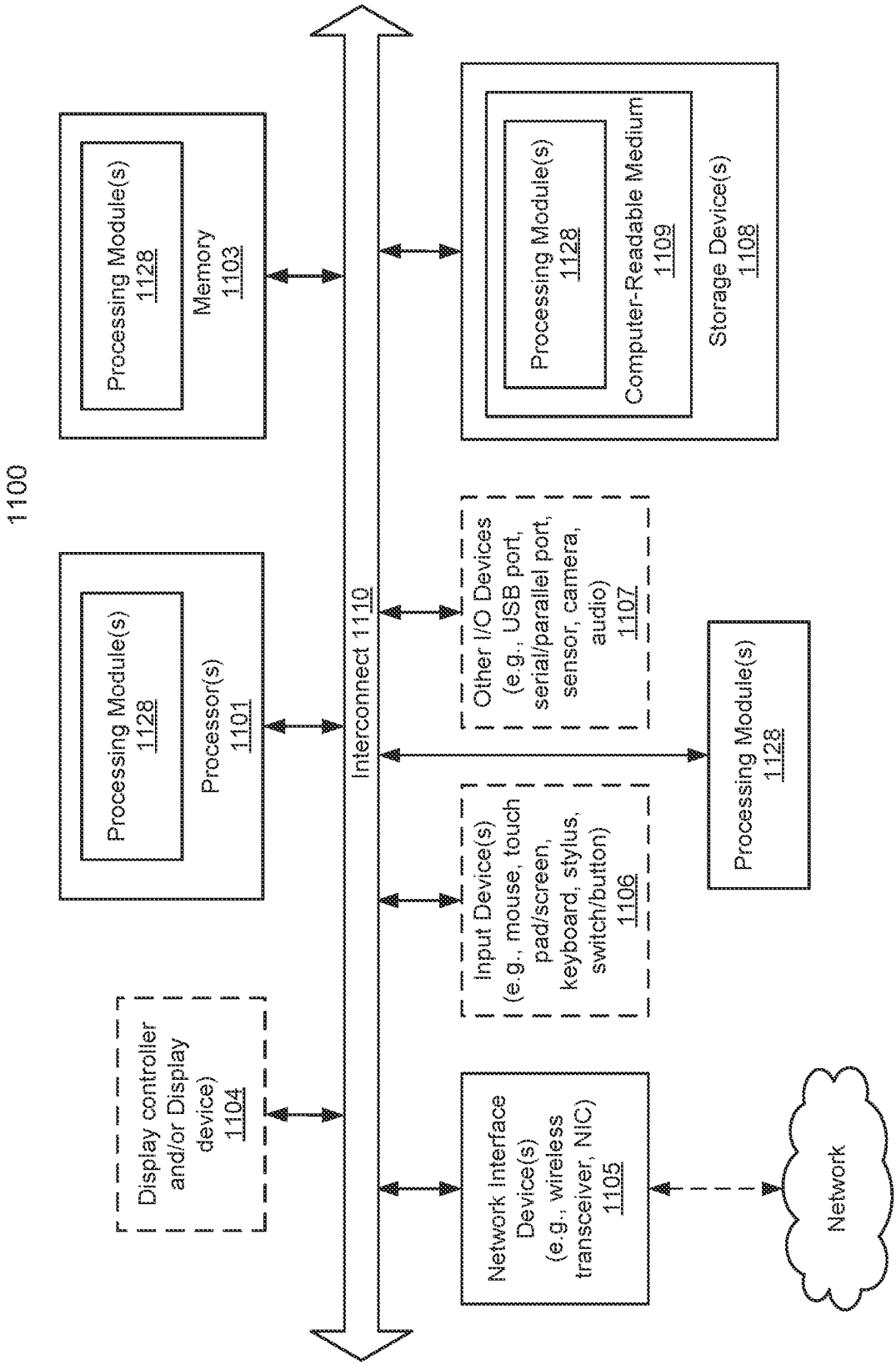

1100
Processing Module(s) 1128
Memory 1103
Processing Module(s) 1128
Processor(s) 1101
Interconnect 1110
Processing Module(s) 1128
Computer-Readable Medium 1109
Storage Device(s) 1108
Other I/O Devices (e.g., USB port, serial/parallel port, sensor, camera, audio) 1107
Input Device(s) (e.g., mouse, touch pad/screen, keyboard, stylus, switch/button) 1106
Display controller and/or Display device) 1104
Network Interface Device(s) (e.g., wireless transceiver, NIC) 1105
Network

FIG. 11

SYSTEM AND METHOD FOR RESECURING DISTRIBUTED SYSTEM RESPONSIVE TO COMPROMISE EVENT

FIELD

Embodiments disclosed herein relate generally to security. More particularly, embodiments disclosed herein relate to systems and methods to secure distributed systems.

BACKGROUND

Computing devices may provide computer-implemented services. The computer-implemented services may be used by users of the computing devices and/or devices operably connected to the computing devices. The computer-implemented services may be performed with hardware components such as processors, memory modules, storage devices, and communication devices. The operation of these components and the components of other devices may impact the performance of the computer-implemented services.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments disclosed herein are illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements.

FIG. **1** shows a block diagram illustrating a system in accordance with an embodiment.

FIG. **2**A shows a first diagram of a distributed system in accordance with an embodiment.

FIG. **2**B shows a second diagram of a distributed system in accordance with an embodiment.

FIG. **2**C shows a second diagram of a spanning tree in accordance with an embodiment.

FIG. **2**D shows a first data flow diagram illustrating data flows in accordance with an embodiment.

FIG. **2**E shows a third diagram of a distributed system in accordance with an embodiment.

FIG. **2**F shows a second data flow diagram illustrating data flows in accordance with an embodiment.

FIGS. **3-4** show flow diagrams illustrating methods in accordance with an embodiment.

FIG. **5** shows a data flow diagram in accordance with an embodiment.

FIG. **6** shows a flow diagram illustrating a method in accordance with an embodiment.

FIG. **7** shows a data flow diagram in accordance with an embodiment.

FIG. **8** shows a flow diagram illustrating a method in accordance with an embodiment.

FIG. **9** shows a data flow diagram in accordance with an embodiment.

FIG. **10** shows a flow diagram illustrating a method in accordance with an embodiment.

FIG. **11** shows a block diagram illustrating a data processing system in accordance with an embodiment.

DETAILED DESCRIPTION

Various embodiments will be described with reference to details discussed below, and the accompanying drawings will illustrate the various embodiments. The following description and drawings are illustrative and are not to be construed as limiting. Numerous specific details are described to provide a thorough understanding of various embodiments. However, in certain instances, well-known or conventional details are not described in order to provide a concise discussion of embodiments disclosed herein.

Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in conjunction with the embodiment can be included in at least one embodiment. The appearances of the phrases "in one embodiment" and "an embodiment" in various places in the specification do not necessarily all refer to the same embodiment.

References to an "operable connection" or "operably connected" means that a particular device is able to communicate with one or more other devices. The devices themselves may be directly connected to one another or may be indirectly connected to one another through any number of intermediary devices, such as in a network topology.

In general, embodiments disclosed herein relate to methods and systems for managing distributed system. The distributed systems may include data processing systems that may provide any number and type of computer implemented services. Malicious entities may attempt to compromise the data processing systems. While described below with respect to malicious entities, it will be appreciated that the operation of data processing systems may be compromised due to other reasons (e.g., errors in operation) without departing from embodiments disclosed herein.

To manage risks of compromise, the data processing systems of the distributed system may implement a security framework. The security framework may distribute authority for validating entities to a variety of different data processing systems. The security framework may utilize a hierarchy of the data processing systems to define which data processing system has authority with respect to other data processing systems. Data processing systems higher in the hierarchy may have authority over data processing systems that are lower in the hierarchy.

When data processing systems of the distributed system are compromised, the hierarchy may be used to identify other data processing systems that may be impacted by the compromised data processing systems. The identified data processing systems may be remediated. If the remediations are unsuccessful, the confirmed compromised data processing systems may be excluded from the hierarchy. Exclusion from the hierarchy may render the data processing systems as untrusted to the remaining data processing systems. Consequently, the confirmed compromised data processing systems may be excluded from continuing influence on the distributed system.

The hierarchy of the distributed system (also referred to herein as "distributed system hierarchy") may be established using, at least, a weighted reputation score (also referred to herein simply as a "weighted reputation") of each data processing system making up the distributed system. The weighted reputation of a data processing system may be affected by (e.g., calculated using) factors such as, but not limited to: seniority, adherence to rules, stability of traffic, etc. Data processing systems with higher (e.g., larger) weighted reputations are deemed more reliable and thus arranged higher within the hierarchy (e.g., may be used as a root data processing system (e.g., a root node) and/or may be located closer to the root node). More reliable data processing systems may also be used to sign (e.g., authenticate/certify) more of the other data processing systems within the distributed system while less reliable data processing systems may be configured to sign less amounts of other data processing systems. Such a hierarchy that prioritizes less risky data processing systems (e.g., data processing systems

with higher weighted reputations that are less likely to fail an authentication by another data processing system) advantageously improves the stability of the above-discussed data processing system authentication process.

Overtime, various events may occur that impact security postures of data processing systems. Occurrences of these events may trigger revision of the hierarchy. Security data used to validate authority of data processing systems may be updated based on the revised hierarchies in a manner that allows for authority to continue to be validated while the security data is refreshed.

If the posture of a data processing system falls below a certain standard, the data processing system may be treated as being compromised. When compromised data processing systems are identified, they may be excluded from the hierarchy and a global security data refresh may be performed to limit influence of the compromised data processing system on operation of the distributed system (e.g., after being excluded).

Thus, embodiments disclosed herein may address, among others, the technical problem of resource cost for security in distributed systems. By implementing the framework, the resource cost for securing the distributed systems may be reduced. By distributing authority for validation of trusted entities across the distributed system, the impact of compromises on the system may be limited. Thus, the resource cost for subsequent remediations of the distributed system may also be reduced. Accordingly, a system in accordance with embodiments disclosed herein may more efficiently marshal limited computing resources of system through reduce resource expenditures for securing the distributed system.

In an embodiment, a method for managing security of a distributed system is provided. The method may include identifying that a data processing system of the distributed system is compromised; removing the data processing system from a hierarchy of data processing systems of the distributed system to obtain a revised hierarchy, the hierarchy being based on security postures of the data processing systems; initiating a global refresh of security data based on the revised hierarchy, the global refresh revoking certificates through which authority of the data processing system may be validated thereby removing authority of the data processing system within the distributed system; using the refreshed security data to validate authority of other data processing systems of the data processing systems; and providing computer implemented services based on the validated authority of the other data processing systems.

Identifying that the data processing system may include revising a reputation for the data processing system based on an occurrence of an event impacting a security posture of the data processing system; and identifying that the data processing has been compromised based on the revised reputation.

Identifying that the data processing has been compromised based on the revised reputation may include comparing the revised reputation to a reputation threshold.

The reputation threshold may be a static value. The reputation threshold may be based on an average reputation of data processing systems of the distributed system. The reputation threshold may be an acceptable level of deviation from the average reputation.

The global refresh of the security data may revoke all certificates usable to validate authority of any of the data processing systems, and may generate new certificates for the other data processing systems usable to validate the authority of the other data processing systems.

Revising the hierarchy may retain all relationships of a previous hierarchy except for a portion of the relationships related to the data processing system.

In an embodiment, a non-transitory media is provided. The non-transitory media may include instructions that when executed by a processor cause the method to be performed.

In an embodiment, a data processing system is provided. The data processing system may include the non-transitory media and a processor, and may initiate performance of the method when the computer instructions are executed by the processor.

Turning to FIG. **1**, a block diagram illustrating a system in accordance with an embodiment is shown. The system shown in FIG. **1** may provide computer-implemented services. The computer implemented services may include any type and quantity of computer implemented services. For example, the computer implemented services may include data storage services, data acquisition services, and/or any other type of service that may be implemented with a computing device.

To provide the computer implemented services, the system of FIG. **1** may include distributed system **100**. Distributed system **100** may include any number of data processing systems **102-104** that provide the computer implemented services. The data processing systems may cooperatively provide the computer implemented services.

The data processing systems of distributed system **100** may be geographically distributed and communicate with one another via wired and/or wireless networks. The geographic distribution and communications may present risks to the computer implemented services. For example, malicious entities (not shown) may attempt to interfere with the operation of any of the data processing systems. To do so, the malicious entities may communicate with the data processing systems. The communications may request, for example, that various operations be performed, that various information be provided, and/or may otherwise ask data processing systems **102-104** to perform one or more operations that may compromise the provided the computer implemented services.

To manage such risks, the data processing systems of distributed system **100** may implement a security framework. The security framework may require that the data processing systems validate that they are communicating with and/or otherwise interacting with trusted entities.

To enable the data processing systems to perform validations as part of the security framework, the data processing systems may obtain information from certificate authority **120**. Certificate authority perform validation processes for trusted entities and distribute security data (e.g., certificates, which may be signed by the certificate authority and verifiable using a publicly available key for the certificate authority) to data processing systems usable to validate trusted entities. When a data processing system interacts with another entity, the data processing system may attempt to validate the entity using the security data. If validated, the data processing system may continue to interact with the entity. Otherwise, the data processing system may discontinue interaction with the entity.

However, this approach to validating entities relies on a single certificate authority. If the certificate authority is unavailable (e.g., to manage security data and update it over time as new information regarding entities becomes available) temporarily or never available, then the security data distributed by certificate authority **120** may not be reliable for validation purposes.

For example, after an entity that was previously validated by certificate authority **120** becomes compromised, if the previously distributed security data is not timely updated (e.g., revoked and/or replaced) by certificate authority **120** then the compromised entity may be believe a trusted entity by the data processing systems of distributed system **100**. Further, if certificate authority **120** itself becomes compromised, then all of data processing systems **102-104** may be subject to compromise. For example, the certificate authority may (i) distribute security data that indicates that malicious entities are to be trusted, and/or (ii) may use its authority over data processing systems **102-104** to induce performance of actions that may directly compromise the computer implemented services provided by distributed system **100**.

In general, embodiments disclosed herein may provide methods, systems, and/or devices for managing security of distributed systems. The disclosed systems may manage security using security framework that distributes authority throughout the distributed system. By distributing the authority throughout the distributed system, the impact of compromise of portions of the distributed may be limited (e.g., rather than expansive in the case of a compromised certificate authority).

The authority may be distributed throughout the system based on a likelihood of members of the distributed system being compromised. Members less likely to be compromised may be given higher levels of authority (e.g., may occupy higher levels in a hierarchy that defines the distribution of authority). The levels of authority may be verified over time to make adjustment to the distribution of the authority.

In the event that a member of the distributed system becomes compromised (e.g., based on a sufficiently low reputation), the member may be ejected from the distributed system. During the ejection process, security data throughout the distributed system may be updated to reduce the likelihood of the compromise of the compromised member impacting security of the updated distributed system.

Thus, embodiments disclosed herein may address, among others, the technical problem of security in distributed systems. The disclosed embodiments may address security in distributed systems through distributed authority for determining whether entities within the system are trusted.

To provide the above noted functionality, data processing systems **102-104** may implement a security framework that manages authority (e.g., for determining trust) based on a hierarchy. The hierarchy may be established when the distributed system is initially setup. As part of the setup process, a spanning tree or other type of structure for the data processing systems may be established. The spanning tree may be established via any process.

For example, the spanning tree may be established based on connectivity between the data processing systems. Upon startup, the data processing systems may establish a mesh network or other communication topology between the data processing systems. The mesh network may use the spanning tree protocol or other methodology for defining which links between the data processing systems should be active.

The hierarchy of the distributed system may also be established using, at least, a weighted reputation of each data processing system making up the distributed system. The weighted reputation of a data processing system may be affected by (e.g., calculated using) factors such as, but not limited to: seniority, adherence to rules, stability of traffic, etc. Data processing systems with higher (e.g., larger) weighted reputations are deemed more reliable and thus arranged higher within the hierarchy (e.g., may be used as a root data processing system (e.g., a root node) and/or may be located closer to the root node). More reliable data processing systems may also be used to sign (e.g., authenticate/certify) more of the other data processing systems within the distributed system while less reliable data processing systems may be configured to sign less amounts of other data processing systems. Such a hierarchy that prioritizes less risky data processing systems (e.g., data processing systems with higher weighted reputations that are less likely to fail an authentication by another data processing system) advantageously improves the stability of the above-discussed data processing system authentication process.

The hierarchy may be established using both weighted reputation and connectivity of the data processing systems. For example, the weighted reputations may be used to modify an initial hierarchy established through use of the spanning tree protocol. Various exchanges and/or other operations may be performed. The corresponding connectivity of the data processing systems may also be modified, or may not be modified. Refer to FIGS. **2A-2C** and **5** for additional details regarding setup of distributed systems and establishing hierarchy.

Overtime, the hierarchy may be adjusted to address changes in the security posture of the data processing system. As new information regarding the likelihood of compromise of data processing systems becomes available, the hierarchy may be updated. Refer to FIG. **7** for additional details regarding updating of hierarchies.

If a reputation of a data processing system is sufficiently low, the member may be ejected from the distributed system and the hierarchy may be revised to manage an impact of the compromise. Additionally, security data used to validate authority may be updated throughout the distributed system to ensure that influence of the compromised data processing system does not persist. Refer to FIG. **9** for additional details regarding managing impacts of compromised data processing system.

Once established, the hierarchy may be used to obtain security data for the data processing systems. To obtain the security data, each data processing system may generate a key pair. Data processing systems more highly rated than other data processing systems may establish certificates for lower ranked data processing systems. Refer to FIG. **2D** for additional details regarding security data.

As the hierarchy is modified, security data for the data processing systems may be refreshed. During refreshes of the security data, different portions may be updated in orders such that the authority of the data processing systems may still be validated. Refer to FIG. **7** for additional details regarding refreshing security data.

The certificates and key pairs may be used in the security framework to validate entities. For example, when a data processing system obtains a communication from an entity, the communication may be validated by ascertaining (e.g., using the certificates) whether a data processing system higher in the hierarchy has validated that the entities is to be trusted. The determination may be made by attempting to use public keys in certificates signed by the data processing system higher in the hierarchy to validate a signature included in the communication. If the signature is validated, then the entity may be identified as being trusted. Otherwise, the entity may be treated as not being trusted. Refer to FIGS. **2E-2F** for additional details regarding validating entities.

Over time, some of the data processing systems may be compromised. For example, malicious code may be

executed by a data processing system which may modify operation of the data processing system in an undesired manner.

Compromised data processing systems may be identified via any method (e.g., code checksums, challenge-response, etc.). As data processing systems become compromised, some of the security data used in the framework may be updated and/or the hierarchy may be updated (e.g., to exclude the compromised devices). To identify the security data to refresh, the hierarchy may be used to identify data processing systems likely to be impacted by the compromised data processing system. The security data for only those data processing systems may be updated.

If any data processing system cannot be remediated to return to a nominal operating state, then the hierarchy may be updated to exclude the data processing system from the hierarchy. Consequently, the authority vested in these data processing systems may be divested.

When providing their functionality, any of data processing systems **102-104** may perform all, or a portion, of the methods illustrated in FIGS. **3-4**, **6**, **8**, and **10**.

Any of data processing systems **102-104** may be implemented using a computing device (also referred to as a data processing system) such as a host or a server, a personal computer (e.g., desktops, laptops, and tablets), a "thin" client, a personal digital assistant (PDA), a Web enabled appliance, a mobile phone (e.g., Smartphone), an embedded system, local controllers, an edge node, and/or any other type of data processing device or system. For additional details regarding computing devices, refer to FIG. **11**.

Any of the components illustrated in FIG. **1** may be operably connected to each other (and/or components not illustrated) with communication system **130**. In an embodiment, communication system **130** includes one or more networks that facilitate communication between any number of components. The networks may include wired networks and/or wireless networks (e.g., and/or the Internet). The networks may operate in accordance with any number and types of communication protocols (e.g., such as the internet protocol).

While illustrated in FIG. **1** as including a limited number of specific components, a system in accordance with an embodiment may include fewer, additional, and/or different components than those illustrated therein. For example, a system may include any number of instances of distributed system **100** and/or other components not shown in FIG. **1**. Any of the instances may perform similar and/or different functions performed by other instances.

Turning to FIG. **2**A, a first diagram of an example topology of a distributed system in accordance with an embodiment is shown. The distributed system may provide any type and quantity of computer implemented services.

To provide the services, the distributed system may include data processing systems **200-208**. The data processing systems may be implemented using, for example, internet of things devices. The data processing systems may include wired and/or wireless communication hardware through which any number of links (e.g., communication channels) may be established. In the example distributed system shown in FIG. **2**A, only a limited number of links **210-219** may be established between the data processing systems. Links **210-219** may form the basis for a mesh or other type of network.

In the state shown in FIG. **2**A, the data processing systems **200-208** may have been migrated from a first environment (e.g., factory) in which trust between the data processing systems was established to a new environment in which devices of malicious parties may be present. Consequently, the data processing systems may need to establish a security framework mitigate the threats presented by the malicious parties.

Turning to FIG. **2**B, a second diagram of an example topology of a distributed system in accordance with an embodiment is shown. Continuing with the previous example discussed with respect to FIG. **2**A, once deployed the data processing systems may perform the spanning tree protocol or other network analysis algorithms to establish a loop-free logical topology for communication between the data processing systems. In this example, the result of performing the spanning tree protocol may indicate that link **218** and link **219** are to be disabled (drawn in dashing to indicate being disabled). In this example, data processing system **204** may be a gateway to other networks or devices. Thus, data processing system **204** may serve as the root with the links being pruned to accomplish various goals (e.g., minimize link contention).

As discussed above, a hierarchy for the data processing systems may be established based at least in part on a spanning tree for the data processing systems. The result of the spanning tree protocol for network configuration may be a spanning tree usable for other purposes. In this example, the same spanning tree generated for network configuration may be used to establish the hierarchy, or at least an initial iteration of the hierarchy.

Turning to FIG. **2**C, a diagram of a spanning tree in accordance with an embodiment is shown. As discussed above, the spanning tree may be established using the spanning tree protocol. However, it will be appreciated that other algorithms may be used to obtain a spanning tree (or other type of structure) for the distributed system.

The spanning tree may include nodes **220-228**. The nodes may correspond to different data processing systems. Node **224** (e.g., a root node) may correspond to data processing system **204**. Node **220** may correspond to data processing system **200**. Node **222** may correspond to data processing system **202**. Node **226** may correspond to data processing system **206**. Node **228** may correspond to data processing system **208**.

While used to configure the links, the spanning tree may also be used to establish authority for validating entities in the system. Rather than having a central authority for validating entities, each data processing system may have authority for validating entities below them (i.e., the data processing system corresponding to child nodes) in the hierarchy.

For example, node **220** may be responsible for validating for node **222**. In other words, node **222** may treat node **220** as the entity that must attest to the validity of any entity.

To facilitate validation, each of the nodes may generate security data usable for validation purposes. Refer to FIG. **2**D for additional details regarding generation of security data.

While the spanning tree illustrated in FIG. **2**C may be used as a hierarchy, it will be appreciated that the hierarchy may be modified based on weighted reputations of the data processing systems corresponding to the nodes, and/or may be based entirely on weighted reputations of the data processing systems. If based on a combination of each, an initial spanning tree may be establishing using the spanning tree protocol. The weighted reputations for the data processing systems may then be established and used to, for example, modify locations of some nodes in the spanning tree, divide branches, consolidate branches, and/or perform other types of modifications. The modifications may be performed to

move nodes corresponding data processing systems with larger weighted reputations towards the root and other nodes corresponding to other data processing systems with smaller weighted reputations towards the leaves (e.g., **222**, **228**, **226**).

Turning to FIG. **2**D, a first data flow diagram in accordance with an embodiment is shown. In the diagram, processes performed by entities are diagrammed as boxes positioned over the line descending from the box representing each entity. For example, the box marked as **230** in FIG. **2**D may be a process performed by data processing system **202** through which data is obtained. Additionally, in the diagram, data flows between devices are shown with lines terminating in arrows between the lines descending from each of the boxes representing the entities. For example, the line terminating in an arrow and marked as **232** may indicate a data flow from data processing system **202** to data processing system **200**. Further, the activities indicated by the elements below the boxes representing each entity may be temporally ordered such that activities shown higher on the page may be performed activities shown lower on the page. Other data flows diagrams discussed below may follow similar conventions.

Continuing with the discussion from FIG. **2**D, security data may be generated usable to perform validations. A portion of the security data may be generated by each of the data processing systems through key pair generation processes (e.g., **230**, **231**). The data processing systems may include hardware capable of generating secure key pairs including private and public keys.

Once generated, each data processing system may identify their respective place in the hierarchy. Once identified, the data processing system above them (if any) in the hierarchy may be identified. In this example, data processing system **200** is above data processing system **202** in the hierarchy.

Once identified, data processing system **202** may send message **232** to data processing system **200**. The message may request that data processing system **200** sign for data processing system (e.g., create an attestation usable to validate data processing system **202**). Message **232** may include the public key of the key pair and an identity of data processing system **202**. Data processing system **200** perform certificate generation process **234** to sign as requested. During certification generation process **234**, data processing system **200** may sign the public key and identity (e.g., may be an identifier for data processing system **202**) using a private key generated by data processing system **200** during key pair generation process **231** thereby generating additional security data (e.g., a certificate). The additional security data may allow any entity that views data processing system **202** as authoritative (e.g., based on the hierarchy) to validate whether a communication is from data processing system **202**.

For example, data processing system **202** may sign communications using a private key of the key pair generated in key pair generation process **230**. An entity that receives the communication may use the public key from the certificate to identify that the entity that generated the communication has access to the private key of the key pair. And the entity may use the public key of data processing system **200** to determine that the certificate should be trusted.

The certificate and public key of data processing system **200** may be published by sending them to data processing system **202** and/or other entities (e.g., via message **236**). Once distributed, the data processing systems of the distributed system may have sufficient security data to quickly ascertain whether a communication is from a trusted source and/or whether and entity that sent the communication should be trusted. Any quantity of security data for any number of data processing systems of a distributed system may be established in this manner.

Turning to FIG. **2**E, a third diagram of an example topology of a distributed system in accordance with an embodiment is shown. Continuing with the example discussed with respect to FIG. **2**D, now consider an example scenario where data processing system **240** establishes link **242** with data processing system **202** and attempt to communicate with data processing system **202**. Prior to acting on the communications, data processing system **202** may initiate performance of a validation for data processing system **240**.

Turning to FIG. **2**F, a second data flow diagram in accordance with an embodiment is shown. Continuing with the example from FIG. **2**E, data processing system **240** may initiate communication with data processing system **202** by sending message **250** in which a request is made to data processing system **202**. The request may indicate that a particular action is to be performed.

Prior to acting on the request, data processing system **202** may perform validation process **252**. During validation process **252**, data processing system **202** may evaluate the security data (e.g., signatures) included in message **250**.

If no signature is included, then message **250** may be treated as being from an untrusted entity.

If a signature is included, then security data may be used to ascertain whether the entity that generated message **250** should be trusted. To do so, data processing system **202** may initiate a verification for the signature. During the verification, data processing system **202** may use security data generated by data processing system **200** to attempt to verify the signature. For example, public keys and/or identifiers from certificates signed by data processing system **200** may be used to attempt to (i) validate the signatures and (ii) verify that the signatures are associated with the identifiers. If the signatures are both valid and associated with the entity alleged in message **250** to have generated message **250**, then the entity and/or message **250** may be treated as a trusted entity (e.g., the message **250** may be processed rather than discarded). If the signatures cannot be validated or are not associated with the entity alleged in message **250** to have generated message **250**, then the entity and/or message **250** may be treated as not trustworthy.

To verify the signatures, data processing system **202** may use local copies of certificates, or data processing system **202** may forward the request via message **254** for data processing system **200** for analysis. If forwarded to data processing system **200**, then a response may be returned to data processing system **202** via message **256**.

Once the trustworthiness of the message and/or sender identified, then data processing system **202** may take appropriate action.

As discussed above, the components of FIG. **1** may perform various methods to manage security of distributed systems. FIGS. **3**-**4** illustrate methods that may be performed by the components of the system of FIG. **1**. In the diagrams discussed below and shown in FIGS. **3**-**4**, any of the operations may be repeated, performed in different orders, and/or performed in parallel with or in a partially overlapping in time manner with other operations.

Turning to FIG. **3**, a flow diagram illustrating a method for performing validations in accordance with an embodiment is shown. The method may be performed by any of

data processing systems **102-104** and/or other components of the system shown in FIG. **1**.

At operation **300**, a hierarchy for data processing systems of a distributed system is obtained. The hierarchy may be obtained by performing an analysis of the data processing systems to obtain a spanning tree. The hierarchy may be defined by the spanning tree.

The analysis performed to obtain the spanning tree may be the spanning tree protocol, or other algorithm. The data processing systems may cooperatively perform the algorithm used to obtain the spanning tree.

The hierarchy may be used to obtain security data, as described with respect to FIGS. **2**D-**2**F.

At operation **302**, a communication from an entity is obtained by one of the data processing systems. The communication may be obtained by receiving it from another device.

At operation **304**, a validation for the communication is performed using a validation framework and the hierarchy to identify whether the entity can be validated. The validation may be performed by (i) using the hierarchy to identify an authoritative data processing system (e.g., higher in the hierarchy and has authority to identify valid entities) with respect to the data processing system, and (ii) using security data associated with the authoritative data processing system to determine whether the entity can be validated.

For example, signatures in the communication may attempt to be validated, as discussed with respect to FIG. **2**F. If the signatures can be validated and are associated with the entity that sent the communication, then it may be determined that the entity can be validated. If the communication is not signed, the signature cannot be validated, or the signature is not associated with the entity that send the communication, then the entity may be determined to not be valid.

If the entity can be validated, then the method may proceed to operation **308**. If the entity cannot be validated, then the method may proceed to operation **310**.

At operation **308**, the communication is serviced. The communication may be serviced by, for example, processing a request included in the message. Processing the request may cause the data processing system to perform any number and type of actions.

The method may end following operation **308**.

Returning to operation **306**, the method may proceed to operation **310** if the entity cannot be validated.

At operation **310**, the communication is rejected. The communication may be rejected by, for example, discarding the communication without processing the content. The communication may also be rejected by, for example, sending alerts or notification to other data processing systems indicating presence of a malicious entity, attempting to validate the entity again, etc.

The method may end following operation **310**.

Using the method shown in FIG. **3**, distributed systems may be secured. However, if data processing systems of a distributed system becomes compromised, then the data processing system and other data processing systems for which the data processing services an authoritative data processing system may be subject to compromise. For example, the compromised data processing system may validate communication that are, in face, malicious in nature.

Turning to FIG. **4**, a flow diagram illustrating a method for responding to compromised data processing systems in accordance with an embodiment is shown. The method may be performed by any of data processing systems **102-104** and/or other components of the system shown in FIG. **1**.

At operation **400**, a compromised data processing system may be identified. The compromised data processing system may be identified by (i) analyzing operation of the compromised data processing system, (ii) obtaining an indication from another device that the compromised data processing system is compromised, and/or via other methods.

At operation **404**, an attempt to remediate the compromised data processing system may be performed. The attempt may be made by performing a remediation process for the data processing system.

The remediation process may include, for example, (i) performing processes to attempt to remove malware or other entities from the compromised data processing system, and (ii) performing security scans or other types of confirmatory operations to confirm whether the compromised data processing systems is no longer compromised.

At operation **406**, the security data is refreshed. The security data may be refreshed similar to how the security data was generated (e.g., generation of new key pairs, certificates, etc.).

The method may end following operation **404**.

By remediating compromised data processing systems using the method shown in FIG. **4**, embodiments disclosed herein may resecure distributed systems following compromises more efficiently.

However, the extent of remediations that may need to be performed may depend on the location in which the node for the compromised data processing system resides in the hierarchy (e.g., nodes closer to the root that become compromised may result in larger portions). To reduce the likelihood of higher impact data processing systems be compromised due to their location within the hierarchy, the hierarchy may be established, at least in part, based on the likelihood of each of the data processing systems being compromised.

Turning to FIG. **5**, a data flow diagram in accordance with an embodiment is shown. The data flow diagram illustrates a process of obtaining a hierarchy for the distributed system (e.g., a hierarchy of the data processing systems making up the distributed system). The flow in FIG. **5** may be used to establish a hierarchy that places data processing systems that are less likely to be compromised closer to the root of the hierarchy. As discussed above, the hierarchy may be based on a spanning tree for the data processing systems. The hierarchy may also be based on (e.g., in addition to being based on the spanning tree) a reliability of each data processing system in the distributed system, as discussed in more detail below.

As shown in FIG. **5**, data processing systems information of the data processing systems may be gathered (e.g., retrieved from) from data sources **502**. Data sources **502** may include: any of the data processing systems of the distributed system; a provider (e.g., manufacturer, vendor, distributer, etc.) of the data processing systems; any number of computing devices and/or storage devices external to the distributed system; and/or any source from which data processing systems information may be stored and/or retrieved.

The data processing systems information from the data sources **502** may include, but is not limited to, statistics and parameters (e.g., characteristics) for each of the data processing system including: operational data (e.g., from the system/data logs of each data processing system); specification data including default factory specifications; and/or any relevant data that could be used to characterize (e.g.,

describe) a state of each of the data processing systems. Operational data of each of the data processing systems may include, but is not limited to: a total time of operation (e.g., run time) within the distributed system; an installation date of the data processing system within the distributed system; a total downtime of the data processing system; the number of errors/crashes experienced by the data processing system; adherence of the data processing systems to one or more rules (e.g., communication rules, connection rules, data transfer rules, etc.) set for the distributed system (also referred to herein as an "operating rule adherence rate"); traffic rate and/or traffic amount flowing through the data processing system; traffic latency; and/or any other relevant data that can be used to describe an operating state of the data processing system.

Weighted reputation generation process **504** may obtain (e.g., receive, retrieve, collect, etc.) the data processing systems information from the data sources **502** to generate (e.g., calculate) a weighted reputation score (also referred to herein simply as a "weighted reputation") for each of the data processing systems in the distributed system. The weighted reputation score may be calculated using any formulas, algorithms, and/or models that are able to covert the data processing systems information into a weighted score (e.g., weighted value). The weighted reputation score may indicate a reliability of each data processing system within the distributed system where a higher weighted reputation score represents higher reliability while a lower weighted reputation score represents lower reliability. For example, a first data processing system that is more senior (e.g., been operating longer), adheres better to rules, has zero down time within the distributed system, and has stable traffic will have a higher weighted reputation score (e.g., be more reliable) than a second data processing system that is younger (e.g., recently installed) and has more downtime (e.g., experienced more errors and crashes) within the distributed system. Similarly, a data processing system that hosts an agent that screens for malicious activity may be less likely to be compromised by a malicious entity than another data processing system that does not screen for malicious activity.

Hierarchy organization process **506** may obtain the weighted reputations (e.g., the weighted reputation scores generated by weighted reputation generation process **504**) and generate a hierarchy for the distributed system (e.g., distributed system hierarchy **510** generated by the organized hierarchy produced by the hierarchy organization process **506**). Data processing systems with higher weighted reputations may be: (i) placed higher on the hierarchy than data processing systems with lower weighted reputations; and/or (2) configured to be given responsibility to sign (e.g., authenticate/certify) more of the other data processing systems (e.g., data processing systems with a lower weighted reputation than the weighted reputation of the signing data processing system) within the distributed system. A data processing system with the highest weighted reputation may be selected as a root node for the distributed system. Alternatively, the root node may be selected irrespective of (e.g., without taking into consideration of/independent of) the weighed reputations while the hierarchy of the remaining data processing systems to be connected to the root node may be selected based on (e.g., dependent of) the weighted reputations.

In some embodiments, the hierarchy for the distributed system may first be established based on the spanning tree before the weighted reputations are taken into consideration by the hierarchy organization process, or vice versa. In some embodiments, the weighted reputations may be continuously updated throughout the operation lifetime of the distributed system. Each time the weighted reputations are updated, the hierarchy may also be updated (e.g., by hierarchy organization process **506**) to reflect (e.g., take into account/consider) the updated weighed reputations.

For example, referring back to FIG. **2**C showing a diagram of an example spanning tree. Here, node **224** (e.g., the root node) would have a highest weighted reputation score among all of the existing nodes (e.g., nodes **220**, **222**, **224**, **226**, and **228**) shown in this spanning tree. As a result, node **224** is more trusted and given the responsibility of signing (e.g., authenticating/certifying) more nodes than the other nodes. Furthermore, each of nodes **220**, **228**, and **226** that are directly connected to the root node **224** would have a higher weighted reputation than node **222** (and a lower weighted reputation than node **224**). Lastly, node **220** which signed node **222** would have a higher weighted reputation than nodes **228** and **226**, both of which have no further connected neighboring nodes. In this same example, the root node **224** need not have the highest weighted reputation.

As yet another example and still referring to the spanning tree of FIG. **2**C, assume, for example, that node **222** has a higher weighted reputation than node **220**. As a result, instead of node **220** being connected to node **224** (e.g., the root node) and authenticating/certifying node **222**, node **222** would now be directly connected to node **224** while node **220** would be connected to (and authenticated/certified by) node **222**.

Turning now to FIG. **6**, FIG. **6** shows a flow diagram illustrating a method of managing operation of a data processing system in accordance with an embodiment. The method may be performed by the system of FIG. **1**. Any of the operations described with respect to FIG. **6** may be performed in different orders, skipped, repeated, and/or be performed in a parallel or partially overlapping in time manner.

At operation **600**, data processing systems information may be obtained for each of the data processing systems configured within a distributed system. The data processing systems information may be obtained from one or more data sources (e.g., data source **500** as described above in connection with FIG. **5**). The data processing systems information may be obtained by: being received via a transmission from any of the data sources, directly retrieved from the data sources, etc.

At operation **602**, as discussed above in reference to FIG. **5**, the data processing system information may be used to generate (e.g., using weighted reputation generation process **504**) a weighted reputation score for each of the data processing systems in th distributed systems.

At operation **604**, as discussed above in reference to FIG. **5**, the weighted reputations (e.g., ones generated in operation **602**) may be used to generate (e.g., using hierarchy organization process **506** of FIG. **5**) a hierarchy (also referred to as a "distributed system hierarchy") for the data processing systems of the distributed system. In some embodiments, the hierarchy for the distributed system may first be established based on the spanning tree before the weighted reputations are taken into consideration by the hierarchy organization process, or vice versa. In some embodiments, the weighted reputations may be continuously updated throughout the operation lifetime of the distributed system. Each time the weighted reputations are updated, the hierarchy may also be updated (e.g., by hierarchy organization process **506**) to reflect (e.g., take into account/consider) the updated weighed reputations.

At operation **606**, the hierarchy (e.g., the hierarchy generated in operation **604**) may be used in the distributed system. In particular, by having data processing systems with higher weighted reputations (e.g., by prioritizing data processing systems that are less risky) located closer to a root node of the distributed system, the stability of the above-discussed authentication hierarchy (e.g., in FIGS. **3** and **4**) may be improved as reliable nodes posing less risk and potential impact on a larger number of nodes signed by the reliable nodes are placed closer to the roots (e.g., directly to connected to or within a predetermined connection threshold from the root).

The method may end following operation **606**.

Once obtained, the hierarchy may be used to validate authority of various entities, and received requests. The validated authority may be used to provide computer implemented services. For example, requests by entities that lack validated authority may be rejected and requests for performance of various actions from entities that have validated authority may be implemented. Consequently, the resulting computer implemented services that are provided may include performance of certain requested actions and not performed based on other requested actions.

However, over time various events may occur. The events may impact the security posture of any of the data processing systems. Consequently, any of the data processing systems may become more or less likely to be compromised depending on the impact of the events. Consequently, the hierarchy may become stale because processing systems may no longer be ordered by the hierarchy on the basis of security. Accordingly, the ordering of the hierarchy may allow less secure data processing systems to occupy more impactful locations within the hierarchy.

Turning to FIG. **7**, a data flow diagram in accordance with an embodiment is shown. The data flow diagram illustrates a process of updating a hierarchy for the distributed system and security data used to validate authority within the data processing system based on the updated hierarchy. The flow in FIG. **7** may be used to adapt the operation of the distributed system as the security posture of data processing systems of the distributed system changes over time.

To identify when a hierarchy may have become outdated, the data processing systems may monitor for events that indicate a change in the security posture of any of the data processing system has occurred. The event may be, for example, receiving a message indicating a change in security of a data processing system.

When event impacting reputation **700** is identified, weighted reputation update process **702** may be performed. During weighted reputation update process **702**, the change in security posture of a data processing system based on event impacting reputation **700** may be used to obtain a new weighted reputation for the data processing system. The new weighted reputation may be obtained similarly as described with respect to FIG. **5** (e.g., process **504**).

Once obtained, the updated weighted reputation for the data processing system may be used in hierarchy update process **704**. During hierarchy update process **704**, the new weighted reputation may be used to update an existing hierarchy. For example, a new hierarchy may be obtained as described with respect to process **506** shown in FIG. **5**. The new hierarchy may then be compared to the existing hierarchy to identify any changes in the positions of data processing systems. The changes may be the basis for a hierarchy update.

The hierarchy update may indicate changes to the positions of data processing systems from the existing hierarchy. The hierarchy update may be used during seamless security data update process **706** to update security data used by data processing systems of the distributed system.

During seamless security data update process **706**, new certificates may be generated for data processing systems that have been repositioned in the new hierarchy. Once generated, existing certificates for the now-moved data processing systems may be invalidated.

For example, new certificates may be generated as described with respect to FIG. **2**D. The existing certificates may continue to be used while the new certificates are generated.

Once the new certificates are generated, the existing certificates (that are no longer valid based on the new hierarchy) may be discarded and/or information regarding the invalidity of the existing certificates may be distributed (e.g., so that all entities know not to trust any copies of the existing certificates).

In the event that a data processing system is demoted, a data processing system higher in the hierarchy may maintain a certificate for the data processing system until the data processing system immediately above the demoted data processing system is able to establish a new certificate.

In the event that the data processing system is elevated, a data processing system higher in the hierarchy may generate a certificate for the data processing system prior to data processing system immediately above the elevated data processing prior to elevation of the elevated data processing system invalidates a certificate for the elevated data processing system.

Turning to FIG. **8**, a flow diagram illustrating a method of managing responses to events impacting reputations of data processing systems in accordance with an embodiment is shown. The method may be performed by the system of FIG. **1**. Any of the operations described with respect to FIG. **8** may be performed in different orders, skipped, repeated, and/or be performed in a parallel or partially overlapping in time manner.

At operation **800**, an occurrence of an event impacting a reputation ascribed to a data processing system is identified. The occurrence may be identified by (i) another data processing system observing the data processing system, (ii) obtaining information regarding the occurrence, and/or other methods.

The event may indicate that the security posture of the data processing system has changed.

At operation **802**, an update for the reputation ascribed to the data processing system may be updated. The updated reputation may be obtained similarly to as described with respect to process **504** of FIG. **5**. For example, a function that takes into account a broad variety of information regarding the data processing may be used to obtain a quantification for the reputation of the data processing system. The function may also ingest information regarding the event.

For example, the function may use a scoring system regarding characteristics of the data processing system reflecting its security posture. The scoring system may award points based on these characteristics. The function may then normalize a score for the data processing to a predetermined range such that direct comparisons between different scores may be used to order data processing systems with respect to one another based on the normalized scores.

The scoring system may award points based on (i) a duration of time that the data processing system has been a member of the distributed system (may be positive or

negative, depending on implementation), (ii) security components hosted by the data processing system, (iii) extent of exposure of the data processing system to other entities, (iv) ratio of uptime to downtime of the data processing system, (v) time stability of network traffic to the data processing system, (vi) rate at which errors in operation of the data processing system occur (and/or other characterizations of the errors in operation of the data processing system), (vii) duration of operation of the data processing system (preference being given to longer duration of operation or shorting, depending on implementation), (viii) stability of components hosted by the data processing system (preference given to fewer changes in components), and/or (ix) other indicators of stability of the data processing system.

The update may reflect a change in the quantification and/or new quantification of the reputation for the data processing system.

At operation **804**, the hierarchy may be revised based on the reputation ascribed to the data processing system and the update. The hierarchy may be revised by exchange the location of the data processing in the hierarchy with other data processing systems immediately above or below it in the hierarchy, or retaining the position of the data processing system in the hierarchy.

The determination may be made by comparing the new reputation of the data processing system to the reputations of the data processing systems immediately above and below the data processing system in the hierarchy.

If the new reputation exceeds the reputation of the data processing system above it in the hierarchy, then the position of the data processing system may be exchanged with the position of the data processing system immediately above it in the hierarchy. This process may be repeated until the new reputation is no longer higher than the reputation of data processing systems above the data processing system in the hierarchy.

A similar process may be performed for data processing systems lower in the hierarchy if the new reputation is lower than the previous reputation of the data processing system.

If the new reputation is neither higher or lower than the reputations of the data processing systems immediately above or below it in the hierarchy, then the position of the data processing system may be retained in the revised hierarchy.

At operation **806**, the security data for the distributed system is revised based on the revised hierarchy. The security may be revised by refreshing the security data, as described with respect to operation **406** of FIG. **4**, and with respect to FIG. **2**D.

However, the security data may be refreshed in an order of operations that allow authority to continue to be checked during the refresh. New certificates may be generated prior to existing certificates that are stale being revoked (e.g., invalidated).

For example, if a data processing system is demoted in the hierarchy, a parent data processing system immediately above the data processing system in the revised hierarchy may establish a certificate for the demoted data processing system prior to the previous parent data processing system (now no longer immediately above the demoted data processing system in the hierarchy) revoking an existing certificate and/or other certificates for the demoted data processing system. Consequently, authority of the demoted data processing system may continue to be validated throughout this process.

The method may end following operation **806**.

Once obtained, the refreshed security data may be used to validate authority of the data processing system thereby continuing to allow computer implemented services to be provided while maintaining security of the distributed system.

However, in some cases, a reduced reputation of a data processing system may indicate that it has been compromised rather than just being more likely to be compromised. When a data processing system is identified as having been compromised, a response may be performed to reduce an impact of the compromise on operation of a distributed system.

Turning to FIG. **9**, a data flow diagram in accordance with an embodiment is shown. The data flow diagram illustrates a process of responding to a compromise event. The flow in FIG. **9** may be used to adapt the operation of the distributed system as the security posture of data processing systems of the distributed system changes over time.

As discussed above, when a reputation of a data processing system changes, it may be compared to criteria that define when a data processing system is compromised. The criteria may include, for example, a static or dynamic threshold. The reputation for a data processing system meeting the criteria may be treated as a compromise event (e.g., **900**). A compromise event may be an occurrence of an event that indicates that a data processing system is compromised.

When a compromise event occurs, the distributed system of FIG. **1** may automatically respond to manage an impact of the compromised data processing system on operation of the system. To do so, response process **902** may be performed. During response process **902**, a hierarchy for the distributed system may be revised by removing the node corresponding to the data processing system that is considered to be compromised from the hierarchy. Edges leading to the node may be joined to close the gap in the hierarchy.

For example, referring to FIG. **2**C, if the data processing system corresponding to node **220** is considered compromised, node **220** may be removed and the edges between nodes **220**, **222**, **224** may be joined such that node **222** is considered to be directly connected to node **224** via an edge. To facilitate these types of determinations, any of the data processing systems may maintain information regarding the hierarchy. The information maintained by each data processing system may reflect data processing systems within a predetermined distance in the hierarchy, or within a certain number of communication hops (e.g., forwarding of network data units, such as packets).

Returning to the discussion of FIG. **9**, once the revised hierarchy is obtained, a broadcast out to the remaining data processing systems of the distributed system may be made (e.g., excluding the compromised data processing system). The broadcast may include a request to refresh security data and the revised hierarchy (or portions thereof such that each remaining member of the distributed system may identify their place in the hierarchy).

Upon receipt, the data processing systems may initiate performance of security data update process **904**. During security data update process **904**, existing certificates hosted by the data processing certificates may be revoked, and new certificates may be generated as described with respect to FIG. **2**D. Other security data (e.g., key pairs) may also be discarded/revoked.

Consequently, the authority of the compromised may be globally revoked by the distributed system.

Turning to FIG. **10**, a flow diagram illustrating a method of managing the impact of compromised data processing

systems in accordance with an embodiment is shown. The method may be performed by the system of FIG. **1**. Any of the operations described with respect to FIG. **10** may be performed in different orders, skipped, repeated, and/or be performed in a parallel or partially overlapping in time manner.

At operation **1000**, a data processing system that has been compromised may be identified. The compromised data processing system may be identified by (i) comparing the reputation of the data processing to criteria, (ii) analyzing operation of the compromised data processing system, (iii) obtaining an indication from another device that the compromised data processing system is compromised, and/or via other methods.

The criteria may include a static threshold (e.g., minimum reputation threshold) or a dynamic threshold. The dynamic threshold may be based on the average reputation of the members of the distributed system. For example, the dynamic threshold may be an acceptable level of deviation from the average reputation of the members of the distributed system. The criteria may include other metrics for identify whether a data processing system is compromised based on reputation without departing from embodiments disclosed herein.

At operation **1002**, the data processing system is removed from a hierarchy to obtain an updated hierarchy. The data processing system may be removed from the hierarchy by removing a node corresponding to the data processing system, and joining edges to the now-removed node.

At operation **1004**, a global refresh of security data based on the updated hierarchy is initiated. The global refresh may be updated by broadcasting a communication to remaining members of the distributed system.

Upon receipt, each of the remaining members may (i) revoke existing certificates, public-private key pairs, and/or other cryptographic data structures, and (ii) replace the revoked cryptographic data structures. The cryptographic data structures may be revoked similarly to as discussed with respect to FIG. **2**D.

The method may end following operation **1004**.

Once obtained, the refreshed security data may be used to validate authority of the data processing system thereby continuing to allow computer implemented services to be provided while maintaining security of the distributed system (e.g., by excluding the compromised data processing system).

Any of the components illustrated in FIGS. **1**-**2**F, **5**, **7**, and **9** may be implemented with one or more computing devices. Turning to FIG. **11**, a block diagram illustrating an example of a data processing system (e.g., a computing device) in accordance with an embodiment is shown. For example, system **1100** may represent any of data processing systems described above performing any of the processes or methods described above. System **1100** can include many different components. These components can be implemented as integrated circuits (ICs), portions thereof, discrete electronic devices, or other modules adapted to a circuit board such as a motherboard or add-in card of the computer system, or as components otherwise incorporated within a chassis of the computer system. Note also that system **1100** is intended to show a high level view of many components of the computer system. However, it is to be understood that additional components may be present in certain implementations and furthermore, different arrangement of the components shown may occur in other implementations. System **1100** may represent a desktop, a laptop, a tablet, a server, a mobile phone, a media player, a personal digital assistant (PDA), a personal communicator, a gaming device, a network router or hub, a wireless access point (AP) or repeater, a set-top box, or a combination thereof. Further, while only a single machine or system is illustrated, the term "machine" or "system" shall also be taken to include any collection of machines or systems that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

In one embodiment, system **1100** includes processor **1101**, memory **1103**, and devices **1105**-**1107** via a bus or an interconnect **1110**. Processor **1101** may represent a single processor or multiple processors with a single processor core or multiple processor cores included therein. Processor **1101** may represent one or more general-purpose processors such as a microprocessor, a central processing unit (CPU), or the like. More particularly, processor **1101** may be a complex instruction set computing (CISC) microprocessor, reduced instruction set computing (RISC) microprocessor, very long instruction word (VLIW) microprocessor, or processor implementing other instruction sets, or processors implementing a combination of instruction sets. Processor **1101** may also be one or more special-purpose processors such as an application specific integrated circuit (ASIC), a cellular or baseband processor, a field programmable gate array (FPGA), a digital signal processor (DSP), a network processor, a graphics processor, a network processor, a communications processor, a cryptographic processor, a co-processor, an embedded processor, or any other type of logic capable of processing instructions.

Processor **1101**, which may be a low power multi-core processor socket such as an ultra-low voltage processor, may act as a main processing unit and central hub for communication with the various components of the system. Such processor can be implemented as a system on chip (SoC). Processor **1101** is configured to execute instructions for performing the operations discussed herein. System **1100** may further include a graphics interface that communicates with optional graphics subsystem **1104**, which may include a display controller, a graphics processor, and/or a display device.

Processor **1101** may communicate with memory **1103**, which in one embodiment can be implemented via multiple memory devices to provide for a given amount of system memory. Memory **1103** may include one or more volatile storage (or memory) devices such as random access memory (RAM), dynamic RAM (DRAM), synchronous DRAM (SDRAM), static RAM (SRAM), or other types of storage devices. Memory **1103** may store information including sequences of instructions that are executed by processor **1101**, or any other device. For example, executable code and/or data of a variety of operating systems, device drivers, firmware (e.g., input output basic system or BIOS), and/or applications can be loaded in memory **1103** and executed by processor **1101**. An operating system can be any kind of operating systems, such as, for example, Windows® operating system from Microsoft®, Mac OS®/iOS® from Apple, Android® from Google®, Linux®, Unix®, or other real-time or embedded operating systems such as VxWorks.

System **1100** may further include IO devices such as devices (e.g., **1105**, **1106**, **1107**, **1108**) including network interface device(s) **1105**, optional input device(s) **1106**, and other optional IO device(s) **1107**. Network interface device(s) **1105** may include a wireless transceiver and/or a network interface card (NIC). The wireless transceiver may be a WiFi transceiver, an infrared transceiver, a Bluetooth transceiver, a WiMax transceiver, a wireless cellular telephony transceiver, a satellite transceiver (e.g., a global

positioning system (GPS) transceiver), or other radio frequency (RF) transceivers, or a combination thereof. The NIC may be an Ethernet card.

Input device(s) **1106** may include a mouse, a touch pad, a touch sensitive screen (which may be integrated with a display device of optional graphics subsystem **1104**), a pointer device such as a stylus, and/or a keyboard (e.g., physical keyboard or a virtual keyboard displayed as part of a touch sensitive screen). For example, input device(s) **1106** may include a touch screen controller coupled to a touch screen. The touch screen and touch screen controller can, for example, detect contact and movement or break thereof using any of a plurality of touch sensitivity technologies, including but not limited to capacitive, resistive, infrared, and surface acoustic wave technologies, as well as other proximity sensor arrays or other elements for determining one or more points of contact with the touch screen.

IO devices **1107** may include an audio device. An audio device may include a speaker and/or a microphone to facilitate voice-enabled functions, such as voice recognition, voice replication, digital recording, and/or telephony functions. Other IO devices **1107** may further include universal serial bus (USB) port(s), parallel port(s), serial port(s), a printer, a network interface, a bus bridge (e.g., a PCI-PCI bridge), sensor(s) (e.g., a motion sensor such as an accelerometer, gyroscope, a magnetometer, a light sensor, compass, a proximity sensor, etc.), or a combination thereof. IO device(s) **1107** may further include an imaging processing subsystem (e.g., a camera), which may include an optical sensor, such as a charged coupled device (CCD) or a complementary metal-oxide semiconductor (CMOS) optical sensor, utilized to facilitate camera functions, such as recording photographs and video clips. Certain sensors may be coupled to interconnect **1110** via a sensor hub (not shown), while other devices such as a keyboard or thermal sensor may be controlled by an embedded controller (not shown), dependent upon the specific configuration or design of system **1100**.

To provide for persistent storage of information such as data, applications, one or more operating systems and so forth, a mass storage (not shown) may also couple to processor **1101**. In various embodiments, to enable a thinner and lighter system design as well as to improve system responsiveness, this mass storage may be implemented via a solid state device (SSD). However, in other embodiments, the mass storage may primarily be implemented using a hard disk drive (HDD) with a smaller amount of SSD storage to act as a SSD cache to enable non-volatile storage of context state and other such information during power down events so that a fast power up can occur on re-initiation of system activities. Also a flash device may be coupled to processor **1101**, e.g., via a serial peripheral interface (SPI). This flash device may provide for non-volatile storage of system software, including a basic input/output software (BIOS) as well as other firmware of the system.

Storage device **1108** may include computer-readable storage medium **1109** (also known as a machine-readable storage medium or a computer-readable medium) on which is stored one or more sets of instructions or software (e.g., processing module, unit, and/or processing module/unit/logic **1128**) embodying any one or more of the methodologies or functions described herein. Processing module/unit/logic **1128** may represent any of the components described above. Processing module/unit/logic **1128** may also reside, completely or at least partially, within memory **1103** and/or within processor **1101** during execution thereof by system **1100**, memory **1103** and processor **1101** also constituting machine-accessible storage media. Processing module/unit/logic **1128** may further be transmitted or received over a network via network interface device(s) **1105**.

Computer-readable storage medium **1109** may also be used to store some software functionalities described above persistently. While computer-readable storage medium **1109** is shown in an exemplary embodiment to be a single medium, the term "computer-readable storage medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, and/or associated caches and servers) that store the one or more sets of instructions. The terms "computer-readable storage medium" shall also be taken to include any medium that is capable of storing or encoding a set of instructions for execution by the machine and that cause the machine to perform any one or more of the methodologies of embodiments disclosed herein. The term "computer-readable storage medium" shall accordingly be taken to include, but not be limited to, solid-state memories, and optical and magnetic media, or any other non-transitory machine-readable medium.

Processing module/unit/logic **1128**, components and other features described herein can be implemented as discrete hardware components or integrated in the functionality of hardware components such as ASICS, FPGAs, DSPs or similar devices. In addition, processing module/unit/logic **1128** can be implemented as firmware or functional circuitry within hardware devices. Further, processing module/unit/logic **1128** can be implemented in any combination hardware devices and software components.

Note that while system **1100** is illustrated with various components of a data processing system, it is not intended to represent any particular architecture or manner of interconnecting the components; as such details are not germane to embodiments disclosed herein. It will also be appreciated that network computers, handheld computers, mobile phones, servers, and/or other data processing systems which have fewer components or perhaps more components may also be used with embodiments disclosed herein.

Some portions of the preceding detailed descriptions have been presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the ways used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of operations leading to a desired result. The operations are those requiring physical manipulations of physical quantities.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description, discussions utilizing terms such as those set forth in the claims below, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

Embodiments disclosed herein also relate to an apparatus for performing the operations herein. Such a computer program is stored in a non-transitory computer readable medium. A non-transitory machine-readable medium

includes any mechanism for storing information in a form readable by a machine (e.g., a computer). For example, a machine-readable (e.g., computer-readable) medium includes a machine (e.g., a computer) readable storage medium (e.g., read only memory ("ROM"), random access memory ("RAM"), magnetic disk storage media, optical storage media, flash memory devices).

The processes or methods depicted in the preceding figures may be performed by processing logic that comprises hardware (e.g. circuitry, dedicated logic, etc.), software (e.g., embodied on a non-transitory computer readable medium), or a combination of both. Although the processes or methods are described above in terms of some sequential operations, it should be appreciated that some of the operations described may be performed in a different order. Moreover, some operations may be performed in parallel rather than sequentially.

Embodiments disclosed herein are not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of embodiments disclosed herein.

In the foregoing specification, embodiments have been described with reference to specific exemplary embodiments thereof. It will be evident that various modifications may be made thereto without departing from the broader spirit and scope of the embodiments disclosed herein as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

What is claimed is:

1. A method for managing security of a distributed system, the method comprising:

identifying, by a first data processing system of data processing systems of the distributed system, that a second data processing system of the data processing systems of the distributed system is compromised by at least:

revising a reputation for the second data processing system based on an occurrence of an event impacting a security posture of the second data processing system; and

identifying that the second data processing has been compromised based on the revised reputation;

removing, by the data processing systems of the distributed system other than the second data processing system, the second data processing system from a hierarchy of the data processing systems of the distributed system to obtain a revised hierarchy, the hierarchy being based on security postures of the data processing systems;

initiating a global refresh of security data based on the revised hierarchy, the global refresh revoking certificates through which authority of the second data processing system is validated thereby removing authority of the second data processing system within the distributed system;

using the refreshed security data to validate authority of other data processing systems of the data processing systems beside the second data processing system; and

providing computer implemented services based on the validated authority of the other data processing systems.

2. The method of claim **1**, wherein identifying that the second data processing has been compromised based on the revised reputation comprises:

comparing the revised reputation to a reputation threshold.

3. The method of claim **2**, wherein the reputation threshold is static value.

4. The method of claim **2**, wherein the reputation threshold is based on an average reputation of the data processing systems of the distributed system, the reputation threshold being a level of deviation from the average reputation.

5. The method of claim **1**, wherein the global refresh of the security data revokes all certificates usable to validate authority of any of the data processing systems, and generates new certificates for the other data processing systems usable to validate the authority of the other data processing systems.

6. The method of claim **5**, wherein revising the hierarchy retains all relationships of a previous hierarchy except for a portion of the all relationships related to the second data processing system.

7. The method of claim **1**, wherein the hierarchy is also based on connectivity between the data processing systems.

8. A non-transitory machine-readable medium having instructions stored therein, which when executed by a processor, cause the processor to perform operations for managing security of a distributed system, the operations comprising:

identifying, by a first data processing system of data processing systems of the distributed system, that a second data processing system of the data processing systems of the distributed system is compromised by at least:

revising a reputation for the second data processing system based on an occurrence of an event impacting a security posture of the second data processing system; and

identifying that the second data processing has been compromised based on the revised reputation;

removing, by the data processing systems of the distributed system other than the second data processing system, the second data processing system from a hierarchy of the data processing systems of the distributed system to obtain a revised hierarchy, the hierarchy being based on security postures of the data processing systems;

initiating a global refresh of security data based on the revised hierarchy, the global refresh revoking certificates through which authority of the second data processing system is validated thereby removing authority of the second data processing system within the distributed system;

using the refreshed security data to validate authority of other data processing systems of the data processing systems beside the second data processing system; and

providing computer implemented services based on the validated authority of the other data processing systems.

9. The non-transitory machine-readable medium of claim **8**, wherein identifying that the second data processing has been compromised based on the revised reputation comprises:

comparing the revised reputation to a reputation threshold.

10. The non-transitory machine-readable medium of claim **9**, wherein the reputation threshold is a static value.

11. The non-transitory machine-readable medium of claim **9**, wherein the reputation threshold is based on an average reputation of the data processing systems of the

distributed system, the reputation threshold being an acceptable level of deviation from the average reputation.

12. The non-transitory machine-readable medium of claim **8**, wherein the global refresh of the security data revokes all certificates usable to validate authority of any of the data processing systems, and generates new certificates for the other data processing systems usable to validate the authority of the other data processing systems.

13. The non-transitory machine-readable medium of claim **12**, wherein revising the hierarchy retains all relationships of a previous hierarchy except for a portion of the all relationships related to the second data processing system.

14. A data processing system, comprising:

a processor; and

a memory coupled to the processor to store instructions, which when executed by the processor, cause the processor to perform operations for managing security of a distributed system, the operations comprising:

identifying, by a first data processing system of data processing systems of the distributed system, that a second data processing system of the data processing systems of the distributed system is compromised by at least:

revising a reputation for the second data processing system based on an occurrence of an event impacting a security posture of the second data processing system; and

identifying that the second data processing has been compromised based on the revised reputation;

removing, by the data processing systems of the distributed system other than the second data processing system, the second data processing system from a hierarchy of the data processing systems of the distributed system to obtain a revised hierarchy, the hierarchy being based on security postures of the data processing systems;

initiating a global refresh of security data based on the revised hierarchy, the global refresh revoking certificates through which authority of the second data processing system is validated thereby removing authority of the second data processing system within the distributed system;

using the refreshed security data to validate authority of other data processing systems of the data processing systems beside the second data processing system; and

providing computer implemented services based on the validated authority of the other data processing systems.

15. The data processing system of claim **14**, wherein identifying that the second data processing has been compromised based on the revised reputation comprises:

comparing the revised reputation to a reputation threshold.

16. The data processing system of claim **15**, wherein the reputation threshold is a static value.

17. The data processing system of claim **15**, wherein the reputation threshold is based on an average reputation of data processing systems of the distributed system, the reputation threshold being an acceptable level of deviation from the average reputation.

18. The data processing system of claim **14**, wherein the global refresh of the security data revokes all certificates usable to validate authority of any of the data processing systems, and generates new certificates for the other data processing systems usable to validate the authority of the other data processing systems.

19. The data processing system of claim **18**, wherein revising the hierarchy retains all relationships of a previous hierarchy except for a portion of the all relationships related to the second data processing system.

20. The data processing system of claim **14**, wherein the hierarchy is also based on connectivity between the data processing systems.

* * * * *