

## SECURE MESSAGING FOR OUTAGE EVENTS

### CROSS-REFERENCE TO RELATED APPLICATIONS

5    **[0001]**    This application claims benefit of United States patent application serial number 17/124,428 filed December 16, 2020, which is hereby incorporated herein by reference.

### BACKGROUND

#### **Field of the Various Embodiments**

**[0001]**    The various embodiments relate generally to wireless network communications, and more specifically, to secure messaging for outage events.

#### 10    **Description of the Related Art**

**[0002]**    A wireless mesh network includes a plurality of nodes that are configured to communicate with and transmit data to one another using one or more communication protocols. In lieu of using a hierarchical topology, individual nodes within a wireless mesh network establish direct connections with other nodes within  
15   the network in order to efficiently route data to different locations in the network.

**[0003]**    A node in the network can experience any of a number of outage events. One example of an outage event is a power outage, in which the node loses electrical power. Other examples of outage events include software and/or hardware bugs in the node, and tampering of the node (e.g., removal of a battery from the node). When  
20   a node experiences and detects an outage event, the node can shut itself down to forestall further harm to itself and/or the network.

**[0004]**    A node that is shutting itself down can send one or more messages reporting its status and/or the outage event to a central location (e.g., to a control center or back office) before the node is completely shut down. The node can also  
25   receive and relay status reporting messages from neighboring nodes that are experiencing the same or a different outage event. A drawback of these conventional approaches is that when a node is shutting down, the resources available to the node to send and/or receive messages are very limited. When a node is shutting down, only a subset of the functionality of the node may be available. For example, certain  
30   components of the node may be inaccessible due to insufficient electrical power remaining in the node and/or to disablement by the node to prevent further damage

and/or corruption. The loss of components resulting from the outage event negatively affects the ability of the node to securely send and/or receive messages to report the outage by, for example, denying access to storage where encryption/decryption and/or authentication keys are stored.

- 5   **[0005]**   As the foregoing illustrates, what is needed in the art are more effective ways of transmitting and/or receiving secure messages in outage event.

### **SUMMARY**

- 10   **[0006]**   Various embodiments disclose a computer-implemented method for sending a message associated with an outage event. The method includes, at a first node in a network, receiving a key from a second node in the network, wherein the second node is adjacent to the first node; storing the key in a first memory, wherein the first memory is capable of operating in a low power mode; detecting an outage event; in response to detecting the outage event, operating a first processor in the low power mode; and via the first processor operating in the low power mode: generating  
15   a message, securing the message using the key, and sending the message to the second node.

- 20   **[0007]**   Other embodiments include, without limitation, one or more non-transitory computer-readable media storing instructions for performing one or more aspects of the disclosed techniques, as well as a system for performing one or more aspects of the disclosed techniques.

- 25   **[0008]**   At least one technical advantage and improvement of the disclosed techniques is that nodes in a network can securely send and/or receive messages in response to and during an outage event. Accordingly, nodes in the network can securely report and/or forward last-gasp messages reporting the outage event and associated node statuses to a location where administrators can review the reporting and take appropriate action. Another technical advantage and improvement of the disclosed techniques is that the nodes can securely send and/or receive last-gasp messages using less electrical power than in conventional techniques. These technical advantages provide one or more technological advancements over prior art  
30   approaches.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0009] So that the manner in which the above recited features of the various embodiments can be understood in detail, a more particular description of the inventive concepts, briefly summarized above, may be had by reference to various  
5 embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of the inventive concepts and are therefore not to be considered limiting of scope in any way, and that there are other equally effective embodiments.

[0010] FIG. 1 illustrates a network system configured to implement one or more  
10 aspects of the various embodiments;

[0011] FIG. 2 illustrates a node device configured to transmit and receive data within the network system of FIG. 1, according to various embodiments;

[0012] FIG. 3 is a diagram illustrating example last-gasp keys stored in a last-gasp memory, according to various embodiments;

15 [0013] FIG. 4 is a flow diagram of method steps for sending a secure message associated with an outage event, according to various embodiments; and

[0014] FIG. 5 is a flow diagram of method steps for receiving a secure message associated with an outage event, according to various embodiments.

**DETAILED DESCRIPTION**

20 [0015] In the following description, numerous specific details are set forth to provide a more thorough understanding of the various embodiments. However, it will be apparent to one of skilled in the art that the inventive concepts may be practiced without one or more of these specific details.

**System Overview**

25 [0016] FIG. 1 illustrates a network system configured to implement one or more aspects of the various embodiments. As shown, network system 100 includes field area network (FAN) 110, wide area network (WAN) backhaul 120, and control center 130. FAN 110 is coupled to control center 130 via WAN backhaul 120. Control center 130 is configured to coordinate the operation of FAN 110.

**[0017]** FAN 110 includes personal area network (PANs) A, B, and C. PANs A and B are organized according to a mesh network topology, while PAN C is organized according to a star network topology. Each of PANs A, B, and C includes at least one border router node device 112 and one or more mains-powered device (MPD) node devices 114. PANs B and C further include one or more battery-powered device (BPD) node devices 116.

**[0018]** MPD node devices 114 draw power from an external power source, such as mains electricity or a power grid. MPD node devices 114 typically operate on a continuous basis without powering down for extended periods of time. BPD node devices 116 draw power from an internal power source, such as a battery or other local source (e.g., solar cell, etc.). BPD node devices 116 typically operate intermittently and, in some embodiments, may power down for extended periods of time in order to conserve battery power. MPD node devices 114 and/or BPD node devices 116 are configured to gather sensor data, process the sensor data, and communicate data processing results and/or other information to control center 130. Border router node devices 112 operate as access points that provide MPD node devices 114 and BPD node devices 116 with access to control center 130.

**[0019]** Any of border router node devices 112, MPD node devices 114, and/or BPD node devices 116 are configured to communicate directly with one or more adjacent node devices (also referred to as neighbors or neighbor node devices) via bi-directional communication links. In various embodiments, a given communication link may be wired or wireless links, although in practice, adjacent node devices of a given PAN exchange data with one another by transmitting data packets via wireless radio frequency (RF) communications. The various node types are configured to perform a technique, known in the art as “channel hopping,” in order to periodically receive data packets on varying channels. As known in the art, a “channel” may correspond to a particular range of frequencies. In one embodiment, a node device may compute a current “receive” channel by evaluating a Jenkins hash function that is based on a total number of channels, the media access control (MAC) address of the node device, and/or other information associated with the node device.

**[0020]** In various embodiments, each node device within a given PAN may implement a discovery protocol to identify one or more adjacent node devices or “neighbors.” In such instances, a node device that has identified an adjacent,

neighboring node device may establish a bi-directional communication link with the neighboring node device. Each neighboring node device may update a respective neighbor table to include information concerning the other node device, including the MAC address of the other node device, as well as a received signal strength

5 indication (RSSI) of the communication link established with that node device. In various embodiments, the neighbor table may include information about one or more communication modes that the neighbor mode is capable of supporting, such as the operating parameters (e.g., data rates, modulation scheme, channel spacing, frequencies supported, channel schedule, etc.).

10 **[0021]** Node devices may compute the channel hopping sequences of adjacent node devices in order to facilitate successful transmission of data packets to such node devices. In embodiments where node devices implement the Jenkins hash function, a node device may compute a “current receive” channel of an adjacent node device using the total number of channels, the MAC address of the adjacent node  
15 device, and/or a time slot number assigned to a current time slot of the adjacent node device.

**[0022]** Any of the node devices discussed above may operate as a source node device, an intermediate node device, or a destination node device for the transmission of data packets. In some embodiments, a given source node device  
20 may generate a data packet and then transmit the data packet to a destination node device via any number of intermediate node devices (in mesh network topologies). In such instances, the data packet may indicate a destination for the packet and/or a particular sequence of intermediate node devices to traverse in order to reach the destination node device. In some embodiments, each intermediate node device may  
25 include a forwarding database indicating various network routes and cost metrics associated with each route.

**[0023]** Node devices 112, 114, 116 transmit data packets across a given PAN and across WAN backhaul 120 to control center 130. Similarly, control center 130 transmits data packets across WAN backhaul 120 and across any given PAN to a  
30 particular node device 112, 114, 116 included therein. As a general matter, numerous routes may exist which traverse any of PANs A, B, and C and include any number of intermediate node devices, thereby allowing any given node device or

other component within network system 100 to communicate with any other node device or component included therein.

**[0024]** Control center 130 includes one or more server machines (not shown) configured to operate as sources for, and/or destinations of, data packets that  
5 traverse within network system 100. In various embodiments, the server machines may query node devices within network system 100 to obtain various data, including raw and/or processed sensor data, power consumption data, node/network throughput data, status information, and so forth. The server machines may also  
10 transmit commands and/or program instructions to any node device 112, 114, 116 within network system 100 to cause those node devices to perform various operations. In one embodiment, each server machine is a computing device configured to execute, via a processor, a software application stored in a memory to perform various network management operations.

**[0025]** In various embodiments, node devices 112, 114, 116 can experience any of  
15 a number of outage events. Examples of outage events include, without limitation, a loss of electrical power from the power source (e.g., outage of or disconnection from the external or internal power source), detection of a software or hardware bug at the node device, or tampering of the node device (e.g., removal of the battery). When a node device experiences an outage event, the node device can shut down. A node  
20 device that is shutting down can send one or more messages (referred to herein as "last-gasp message(s)"), via one or more other node devices, to control center 130 before the shutdown is complete, in order to report the outage event and/or the status of the node device to control center 130. A node device can also receive and forward last-gasp messages sent from other node devices toward control center 130 that are  
25 shutting down due to the outage event. Last-gasp messages can be secured (e.g., encrypted and/or authenticated) messages. Capabilities of node devices and associated techniques for sending and receiving last-gasp messages are described in further detail below.

**[0026]** In various embodiments, node devices 112, 114, 116 may likewise include  
30 computing device hardware configured to perform processing operations and execute program code. Each node device may further include various analog-to-digital (A/D) converters, digital-to-analog (D/A) converters, digital signal processors (DSPs), harmonic oscillators, transceivers, and/or any other components generally associated

with RF-based communication hardware. FIG. 2 illustrates an exemplary node device that may operate within the network system 100.

**[0027]** FIG. 2 illustrates a node device configured to transmit and receive data within the network system 100 of FIG. 1, according to various embodiments. As shown, node device 210 is coupled to transceiver 250 and oscillator 260. Node device 210 includes processor 220, input/output devices 230, and memory 240. Memory 240 includes one or more applications (e.g., software application 242) that communicate with database 244. Node device 210 can further include a last-gasp (LG) processor 270 and a LG memory 280.

**[0028]** Node device 210 coordinates the operations of node device 210. Transceiver 250 is configured to transmit and/or receive data packets and/or other messages across network system 100 using a range of channels and power levels. Oscillator 260 provides one or more oscillation signals, according to which, in some embodiments, node device 210 may schedule the transmission and reception of data packets. In some embodiments, node device 210 may be used to implement any of border router node devices 112, MPD node devices 114, and/or BPD node devices 116 of FIG. 1.

**[0029]** Node device 210 includes a processor 220, input/output (I/O) devices 230, and memory 240, coupled together. In various embodiments, processor 220 may include any hardware configured to process data and execute software applications. Processor 220 may include a real-time clock (RTC) (not shown) according to which processor 220 maintains an estimate of the current time. The estimate of the current time may be expressed in Universal Coordinated Time (UTC), although any other standard of time measurement can also be used. I/O devices 230 include devices configured to receive input, devices configured to provide output, and devices configured to both receive input and provide output. Memory 240 may be implemented by any technically-feasible computer-readable storage medium.

**[0030]** Memory 240 includes one or more software applications 242 and database 244, coupled together. The one or more software applications includes program code that, when executed by processor 220, may performs any of the node-oriented computing functionality described herein. The one or more software applications 242 may also interface with transceiver 250 to coordinate the transmission and/or

reception of data packets and/or other messages across network system 100, where the transmission and/or reception is based on timing signals generated by oscillator 260. In various embodiments, memory 240 may be configured to store protocols used in communication modes, equations and/or algorithms for identifying metric values, constants, data rate information, and other data used in identifying metric values, etc. Memory 240 can also include a key store 246 where keys for encryption, decryption, and/or authentication of communications (e.g., messages) between node devices can be stored.

**[0031]** In operation, software application(s) 242 can implement various techniques to optimize communications with one or more linked node devices, such as a neighboring node device. In various embodiments, node device 210 may be configured to, using a plurality of different communication modes, transmit data messages to the linked node device and/or receive data messages from the linked node device by selecting a common communication mode that is supported by node device 210 and the linked node device. More generally, node device 210 can be configured for multi-mode communications. Node device 210 can communicate with a linked node or with control center 130 using any of a plurality of modes. The particular mode used for a given transmission depends on the particular circumstances of the transmission (e.g., the type of data message, the intended recipients of the data message, etc.). Examples of the modes include, without limitation, unicast, broadcast, and multi-cast.

**[0032]** In various embodiments, processor 220 can generate one or more encryption/decryption and/or authentication keys for communications with node devices adjacent to the node device 210. For each adjacent node device, processor 220 can generate a key for messages to be sent by the adjacent node device to node device 210. Node device 210 can send to the adjacent node device the key generated for the adjacent node device by processor 220, and can receive a key generated by the adjacent node device. Node device 210 can send encrypted and/or authenticated messages to the adjacent node device using the key received from the adjacent node device. The adjacent node device can send encrypted and/or authenticated messages to node device 210 using the key generated by processor 220, processor 220 can decrypt and/or authenticate the messages using that key or a corresponding asymmetric key. Processor 220 can generate keys in accordance with

instructions programmed in hardware (e.g., programmed into the processor 220) and/or software (e.g., in a software application 242). Node device 210 can store keys generated by processor 220 and keys received from adjacent node devices in memory 240 (e.g., in key store 246).

- 5   **[0033]**   In various embodiments, node device 210 can further include a last-gasp (LG) processor 270 and a LG memory 280 that are separate from processor 220 and memory 240, respectively. LG processor 270 and LG memory 280 are configured to operate in a low power mode (e.g., when node device 210 is shutting down and power in node device 210 is reduced) in conjunction with I/O devices 230, transceiver
- 10   250, and oscillator 260. LG processor 270 can be configured (e.g., programmed) with a limited instruction set for performing specific functions (e.g., functions associated with sending and/or receiving last-gasp messages, shut down processes, and/or the like). In some embodiments, LG memory 280 has a significantly smaller capacity than memory 240 and/or key store 246, which can facilitate operation of LG memory
- 15   280 in a low power mode. As a non-limiting example, memory 240 may have a size of 16 MB and LG memory 280 may have a size of 4 KB. A set of LG keys 282 for securing last-gasp messages can be stored in LG memory 280. LG keys 282, further described below, includes keys generated by node device 210 and keys received from one or more adjacent node devices. LG processor 270 may be implemented by
- 20   any technically feasible processor unit that is capable of operating in low power mode. Examples of LG processor 270 include, without limitation, a digital signal processor, a system-on-a-chip, a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), and/or the like. LG memory 280 may be implemented by any technically feasible computer-readable storage medium (e.g., volatile and/or non-
- 25   volatile memory) capable of operation in low power mode. Examples of LG memory 280 include, without limitation, dynamic random access memory (D-RAM). In some embodiments, in lieu of a separate LG processor 270, processor 220 can operate in a low power mode and includes instructions for performing functions associated with functions associated with sending and/or receiving last-gasp messages. In some
- 30   embodiments, in lieu of a separate LG memory 280, at least a portion of memory 240 can operate in a low power mode, and LG keys 282 can be stored in, and retrieved from, that portion of memory 240. For ease of understanding, processor 220 and memory 240 may be referred to herein as “main processor” 220 and “main memory” 240, respectively.

### Secure Messaging for Outage Events

**[0034]** In operation, prior to an outage event, a node device 210 can establish bi-directional communication links with one or more adjacent node devices in network system 100. In conjunction with establishing a communication link with a given adjacent node device, node device 210 can determine whether the adjacent node device has a first trust level or a second trust level. In some embodiments, the first trust level corresponds to a “manufacturer trust” level, where the adjacent node device does not yet have credentials associated with the owner of network system 100 (e.g., at first installation of the adjacent node device out-of-the-box), and the second trust level corresponds to a “customer trust” (as in a customer (e.g., owner of network system 100) that acquired, and now owns and deploys, the node device) level, where the adjacent node device has credentials associated with the network owner (e.g., after the adjacent node device has been installed and onboarded into network system 100). A “customer trust” level can also be referred to as an “owner trust” level.

**[0035]** For a given adjacent node device with whom a communication link has been established, node device 210 can generate a key (e.g., using a cryptographic random number generator). The key can be sent to the adjacent node device via the communication link; the adjacent node device can use that key to send secure messages to node device 210. Node device 210 can generate a key for the adjacent node based on the trust level. If the adjacent node device has the first trust level, node device 210 generates a key associated with the first trust level for the adjacent node device. If the adjacent node device has the second trust level, node device 210 generates a key associated with the second trust level for the adjacent node device. If the adjacent node device upgrades from the first trust level to the second trust level, node device 210 can generate a new key associated with the second trust level for the adjacent node device. Node device 210 can store the generated key in memory 240. In some embodiments, a node device 210 can send a key to an adjacent node device in a message that includes a timestamp associated with the key, a recipient (the adjacent node device) address, the key, and a key ID associated with the key. In some embodiments, the message for sending the key to the adjacent node device is formatted according to the ISO/IEC 11770-2 Key Establishment Mechanism 3 protocol.

**[0036]** Further, node device 210 can receive a key from the adjacent node device. The adjacent node device generates a key associated with the trust level of node device 210 and sends that key to node device 210. Node device 210 can store keys generated by node device 210 for an adjacent node device and keys received from adjacent node devices in memory 240 (e.g., inkey store 246). In various embodiments, node device 210 stores a set of generated key(s) and received key(s) for each adjacent node device.

**[0037]** In various embodiments, a key is associated with a key identifier (or “key ID”). Node device 210 can generate a key ID in conjunction with generating the corresponding key. When node device 210 secures a message with a key, the key ID is included along with the message (e.g., in a header or a payload of the message). When a node device 210 receives a message from an adjacent node device, node device 210 compares the key ID included with the message to the key ID of the generated key associated with the adjacent node device. If the key IDs do not match, then node device 210 can disregard the message as invalid. If the key IDs match, then node device 210 can proceed to decrypt and/or authenticate the message using the generated key identified by the key ID. In some embodiments, key IDs are unique with respect to the adjacent node device. In some other embodiments, key IDs are unique with respect to the network system 100.

**[0038]** In some embodiments, node device 210 can generate a “current” key and a “next” key for a given adjacent node device. The current key would be sent to the adjacent node device for current use. The next key would be sent to the adjacent node device when the current key needs replacing (e.g., when the adjacent node device disappeared from and reappeared in the network system 100, when a key update time interval has elapsed), and becomes the new current key. Node device 210 can then generate a new next key. Node device 210 can store the current and the next key in memory 240.

**[0039]** In various embodiments, node device 210 can select a subset of adjacent node devices as designated recipients for last-gasp messages. This last-gasp subset of adjacent node devices can include any number of adjacent node devices, up to the actual number of adjacent node devices. In some embodiments, the number of adjacent node devices in the last-gasp subset is relatively small (e.g., 1-3) relative to the actual number of adjacent node devices. In some embodiments, the number of

adjacent node devices to be selected for the last-gasp subset can be predefined. Node device 210 can select one or more adjacent node devices for the last-gasp subset based on one or more criteria, including but not limited to: that the adjacent node device has a path within network system 100 to control center 130 (as opposed to no path to control center 130), that the adjacent node device has the second trust level (customer trust level) (as opposed to having the first trust level), that the adjacent node device has a battery backup (as opposed to lacking a battery backup), and a lower link cost associated with transmitting messages to control center 130 via the adjacent node device (as opposed to a higher link cost associated with transmitting to control center 130 via another adjacent node). Similarly, adjacent node devices can select node device for their respective last-gasp subsets.

**[0040]** For the last-gasp subset of adjacent node devices, node device 210 stores keys generated for and/or received from those adjacent node devices as LG keys 282 in LG memory 280 (or in a portion of memory 240 capable of operating in low power mode). LG keys 282 can be a duplicate of one or more keys stored in key store 246 (e.g., keys are generated and/or received and stored in both memory 240 and LG memory 280). In some embodiments, LG keys 282 are a subset of the keys stored in key store 246. By storing LG keys 282 in LG memory 280 or in a low-power-capable portion of memory 240, LG keys 282 are made accessible for use in sending and/or receiving last-gasp messages in response to an outage event.

**[0041]** FIG. 3 is a diagram illustrating example last-gasp keys 282 stored in a last-gasp memory 280, according to various embodiments. LG keys 282 stored at a node device 210 can include keys generated for the adjacent node devices in the last-gasp subset of adjacent node devices, and can further include keys received from adjacent node devices in that last-gasp subset. As shown, LG keys 282 include keys generated for two adjacent node devices "Neighbor1" and "Neighbor2", one of which (Neighbor1) has the manufacturer trust level and the other (Neighbor2) has the customer trust level. LG keys 282 also includes keys received from adjacent node devices "Neighbor3" and "Neighbor4." It should be appreciated that LG keys 282 can have more or less sets of keys associated with more or less selected adjacent node devices, depending on the particulars of the last-gasp subset.

**[0042]** As shown, LG keys 282 includes a current MFG LG key 302 and next MFG LG key 304 generated for the adjacent node device "Neighbor1" that has the

manufacturer trust level. MFG LG keys 302 and 304 have respective key IDs 314 and respective timestamps 320. MFG LG keys 302 and 304 are both associated with the address (e.g., IP address, MAC address) 316 of Neighbor1. LG keys 282 also includes a current CUST LG key 306 and next CUST LG key 308 generated for the  
5 adjacent node device "Neighbor2" that has the customer trust level. CUST LG keys 306 and 308 have respective key IDs 314 and respective timestamps 320. CUST LG keys 306 and 308 are both associated with the address (e.g., IP address, MAC address) 318 of Neighbor2.

**[0043]** As shown, LG keys 282 further includes LG key 310 received from  
10 Neighbor3 and LG key 312 received from Neighbor4. LG keys 310 and 312 have respective key IDs 322 and respective timestamps 326. LG key 310 is associated with the address 328 of Neighbor3, and LG key 312 is associated with the address 330 of Neighbor4.

**[0044]** In operation, when node device 210 detects an outage event (e.g.,  
15 experiences a power outage, etc.), LG processor 270 can activate and execute instructions associated with sending and/or receiving last-gasp messages. For example, LG processor 270 can execute instructions to send a last-gasp message to Neighbor3 and/or Neighbor4. For a last-gasp message to be sent to Neighbor3, LG processor retrieves Neighbor3 LG key 310 from LG keys 282 and uses LG key 310 to  
20 encrypt and/or authenticate the last-gasp message to be sent to Neighbor3. For a last-gasp message to be sent to Neighbor4, LG processor retrieves Neighbor4 LG key 312 from LG keys 282 and uses LG key 312 to encrypt and/or authenticate the last-gasp message to be sent to Neighbor4.

**[0045]** When node device 210 receives a last-gasp message from adjacent node  
25 device Neighbor1, LG processor compares a key ID included in the last-gasp message to key ID 314 of current MFG LG key 302, associated with Neighbor1, in LG keys 282 or key store 246. If the read key ID matches the key ID 314 of LG key 302, then LG processor 270 can proceed with decrypting and/or authenticating the last-gasp message using LG key 302. If the key IDs do not match, the message can be  
30 disregarded. When node device 210 receives a last-gasp message from adjacent node device Neighbor2, LG processor compares a key ID included in the last-gasp message to key ID 314 of current CUST LG key 306, associated with Neighbor2, in LG keys 282 or key store 246. If the read key ID matches the key ID 314 of LG key

306, then LG processor 270 can proceed with decrypting and/or authenticating the last-gasp message using LG key 306. If the key IDs do not match, the message can be disregarded. If node device 210 is experiencing an outage event when the last-gasp message is received, node device 210 (e.g., LG processor 270) can retrieve LG  
5 key 302 or 306, and associated key IDs, from LG keys 282. If node device 210 is operating normally and not experiencing an outage event when the last-gasp message is received, node device 210 (e.g., processor 220) can retrieve LG key 302 or 306, and associated key IDs, from key store 246.

**[0046]** After the message is decrypted and/or authenticated, LG processor 270  
10 can forward (e.g., relay) the message by encrypting and/or authenticating the message using, depending on whether the adjacent node device Neighbor3 or Neighbor4 that is the forward recipient, LG key 310 or 312, and then sending the encrypted and/or authenticated message to the forward recipient. In some embodiments, LG processor 270 can forward the message to both Neighbor3 and  
15 Neighbor4, in which case the respective messages are encrypted and/or authenticated using LG keys 310 and 312, respectively. If node device 210 is experiencing an outage event when the last-gasp message is received, node device 210 (e.g., LG processor 270) can retrieve LG key 310 or 312, and associated key IDs, from LG keys 282. If node device 210 is operating normally and not experiencing an  
20 outage event when the last-gasp message is received, node device 210 (e.g., processor 220) can retrieve LG key 310 or 312, and associated key IDs, from key store 246.

**[0047]** It should be appreciated that, while as shown the generated LG keys 302, 304, 306, and 308, and received LG keys 310 and 312, are associated with different  
25 sets of adjacent node devices, in some embodiments the received keys are received from the same set of neighbors for which the generated keys are generated.

**[0048]** In some embodiments, a node device 210 can track a number of security-related attributes that can be included in last-gasp messages. For example, node device 210 can generate a reboot counter, which can be incremented with each  
30 outage event. When node device 210 restarts after an outage event, node device 210 can increment the reboot counter. Node device 210 can also generate a sequence number, which is unique for each unique reboot counter. Node device 210 can send the reboot counter and sequence number to adjacent node devices with

which node device 210 re-establish communication links following the reboot. When sending a last-gasp message, node device 210 includes the reboot counter and sequence number of node device 210. When node device 210 receives a last-gasp message from an adjacent node device, in addition to checking the key ID, node  
5 device 210 (e.g., LG processor 270) also checks the reboot counter and sequence number included in the last-gasp message. If the reboot counter is not the same as what was last received from the adjacent node device and/or the sequence number is less than what was last received from the adjacent node device, then the message can be considered invalid. Additionally and/or alternatively, the last-gasp message  
10 can include a timestamp that can also be checked to validate the last-gasp message. By checking the reboot counter, sequence number, and/or timestamp of the last-gasp message in addition to the key ID included in the last-gasp message, node device 210 can check whether the last-gasp message is valid. If the message is valid, node device 210 can forward the message. If the message is invalid, node device 210 can  
15 disregard it.

**[0049]** FIG. 4 is a flow diagram of method steps for sending a secure message associated with an outage event, according to various embodiments. Although the method steps are described with respect to the systems of FIGs. 1-3, persons skilled in the art will understand that any system configured to perform the method steps, in  
20 any order, falls within the scope of the various embodiments.

**[0050]** As shown, a method 400 begins at step 402, where node device 210 detects an outage event. For example, node device 210 can detect a loss of power due to a power outage, a hardware and/or software defect, and/or tampering of the node device 210.

25 **[0051]** At step 404, node device 210 activates LG processor 270. In response to detecting the outage event, LG processor 270 activates and begins operating, such as by executing instructions and/or the like.

**[0052]** At step 406, node device 210 generates a last-gasp message. LG processor 270 generates a last-gasp message, to be sent via one or more adjacent  
30 node devices, that reports the outage event and/or the status of node device 210 to, for example, control center 130. The last-gasp message includes a key ID of a LG key received from the adjacent node device that will receive the message (e.g., the

key ID of LG key 310 if adjacent node device Neighbor3 will be the recipient). If the message is to be sent to multiple adjacent node devices, then multiple messages, each including respective key IDs, can be generated. The generated message can be an original last-gasp message originating from node device 210 or a forward of a  
5 last-gasp message received from an adjacent node device.

**[0053]** At step 408, node device 210 secures the last-gasp message using a LG key received from an adjacent node device (for the last-gasp message) and stored in LG memory 280. LG processor 270 retrieves, from LG keys 282 stored in LG memory 280, a key received from the adjacent node device that will be the recipient  
10 of the last-gasp message (e.g., LG key 310 if adjacent node device Neighbor3 will be the recipient). LG processor 270 proceeds to secure (e.g., encrypt and/or authenticate) the message using that retrieved LG key.

**[0054]** At step 410, node device 210 sends the last-gasp message to the adjacent node device. LG processor 270, in conjunction with transceiver 250 and optionally  
15 oscillator 260, sends the last-gasp message to the adjacent node device. Then method 400 ends.

**[0055]** FIG. 5 is a flow diagram of method steps for receiving a secure message associated with an outage event, according to various embodiments. Although the method steps are described with respect to the systems of FIGs. 1-3, persons skilled  
20 in the art will understand that any system configured to perform the method steps, in any order, falls within the scope of the various embodiments.

**[0056]** As shown, a method 500 begins at step 502, where node device 210 receives a last-gasp message from an adjacent node device. The last-gasp message may originate from the adjacent node device (e.g., the adjacent node device  
25 experienced an outage event and is sending the last-gasp message in response to the outage event) or could be forwarded by the adjacent node device to node device 210.

**[0057]** At step 504, node device 210 detects whether an outage event is occurring at the node device 210. If node device 210 detects no outage event at node device  
30 210, then method 500 proceeds to step 506, where node device 210 performs the

succeeding steps in method 500, via main processor 220 and main memory 240. Method 500 then proceeds to step 510.

5     **[0058]**     If node device 210 detects an outage event at node device 210, then method 500 proceeds to step 508, where node device 210 performs the succeeding steps in method 500, via LG processor 270 and LG memory 280 (or via main processor 220 and main memory 240 in a low-power mode). Method 500 then proceeds to step 510.

10     **[0059]**     At step 510, node device 210 compares the key ID included in the received last-gasp message with a key ID of a current key associated with the adjacent node device and stored in main memory 240 or LG memory 280. If node device 210 is operating via main processor 220 and main memory 240, main processor 220 retrieves, from key store 246 in main memory 240, a key ID of a current key generated for the adjacent node device from which the last-gasp message is received. If node device 210 is operating via LG processor 270 and LG memory 280, 15 LG processor 270 retrieves, from LG keys 282 in LG memory 280, the key ID of the current key generated for the adjacent node device from which the last-gasp message is received. Main processor 220 or LG processor 270 then compares a key ID included in the last-gasp message with the key ID of the current key generated for the adjacent node device.

20     **[0060]**     At step 512, node device 210 (e.g., main processor 220 or LG processor 270) determines, based on the comparing, whether the key IDs match to determine whether the received last-gasp message is valid. In some embodiments, node device 210 performs additional validation checks on the last-gasp message, in addition to checking whether the key IDs match. The other validation checks can include, for 25 example, checking the timestamp, reboot counter, and/or sequence number of the last-gasp message. If node device 210 determines that the key IDs do not match, then method 500 proceeds to step 514, where node device 210 disregards the last-gasp message. If the key IDs do not match, then the last-gasp message is considered to be invalid, and accordingly node device 210 can disregard it.

30     **[0061]**     If node device 210 determines that the key IDs match, then method 500 proceeds to step 516, where the last-gasp message is decrypted and/or authenticated using the current key. If node device 210 is operating via main processor 220 and

main memory 240, main processor 220 retrieves the key having the key ID from key store 246 in main memory 240, and proceeds to decrypt and/or authenticate the last-gasp message using that key. If node device 210 is operating via LG processor 270 and LG memory 280, LG processor 270 retrieves the key having the key ID from LG keys 282 in LG memory 280, and proceeds to decrypt and/or authenticate the last-gasp message using that key.

**[0062]** At step 518, node device 210 forwards the message to a second adjacent node device using a key received from the second adjacent node device. In a manner similar to steps 406-410 of method 400 described above, main processor 220 or LG processor 270 sends (e.g., forwards) the message using a key associated with the second adjacent node device. The key was received from the second adjacent node device and stored in key store 246 and LG keys 282. If node device 210 is operating via main processor 220, main processor 220 retrieves the key from key store 246, generates the forward message, secures the message using the key, and sends the message to the second adjacent node device. If node device 210 is operating via LG processor 270, LG processor 270 retrieves the key from key store 246, generates the forward message, secures the message using the key, and sends the message to the second adjacent node device. Then method 500 ends.

**[0063]** In sum, a node in a network is configured to store encryption/decryption and/or authentication keys for use in outage events in a separate memory capable of low power operation, and to use those keys when sending and/or receiving messages associated with an outage event. A given node can select a number of neighboring nodes for sending last-gasp messages associated with outage events. The given node generates and exchanges keys with those neighbors, and stores those keys in a memory, separate from main memory, capable of low power operation. When an outage event occurs at the given node, the given node sends a secure message to one or more of the selected neighbors, using one or more of the keys received the selected neighbors and stored in the separate memory. When the given node receives a message associated with an outage event from a neighbor, the given node processes the message using a key generated by the given node and used by the neighbor to secure the message. The given node then can forward the message to a neighbor using a key received from the neighbor.

**[0064]** At least one technical advantage and improvement of the disclosed techniques is that nodes in a network can securely send and/or receive messages in response to and during an outage event. Accordingly, nodes in the network can securely report and/or forward last-gasp messages reporting the outage event and associated node statuses to a location where administrators can review the reporting and take appropriate action. Another technical advantage and improvement of the disclosed techniques is that the nodes can securely send and/or receive last-gasp messages using less electrical power than in conventional techniques. Accordingly, nodes in the network can securely send, receive, and/or forward last-gasp messages in the face of loss of electrical power or reduced electrical power. These technical advantages provide one or more technological advancements over prior art approaches.

**[0065]** 1. In some embodiments, a computer-implemented method for sending a message associated with an outage event comprises, at a first node in a network: receiving a key from a second node in the network, wherein the second node is adjacent to the first node; storing the key in a first memory, wherein the first memory is capable of operating in a low power mode; detecting an outage event; in response to detecting the outage event, operating a first processor in the low power mode; and via the first processor operating in the low power mode: generating a message; securing the message using the key; and sending the message to the second node.

**[0066]** 2. The method of clause 1, wherein the first node comprises the first processor and a second processor distinct from the first processor, and wherein the second processor is a main processor of the first node.

**[0067]** 3. The method of clauses 1 or 2, wherein the first processor is a main processor of the first node.

**[0068]** 4. The method of any of clauses 1-3, wherein the first node comprises the first memory and a second memory distinct from the first memory, wherein the second memory is a main memory of the first node.

**[0069]** 5. The method of any of clauses 1-4, wherein the first memory is a main memory of the first node; and the method further comprises, in response to detecting the outage event, operating the first memory in the low power mode.

**[0070]** 6. The method of any of clauses 1-5, wherein securing the message using the key comprises retrieving the key from the first memory via the first processor operating in the low power mode.

5 **[0071]** 7. The method of any of clauses 1-6, wherein the key is generated by the second node.

**[0072]** 8. The method of any of clauses 1-7, wherein the message comprises a key identifier of the key.

10 **[0073]** 9. The method of any of clauses 1-8, wherein the network comprises a plurality of nodes adjacent to the first node, wherein the plurality of nodes includes the second node, and wherein the method further comprises selecting a subset of the plurality of nodes adjacent to the first node as designated recipients of outage event messages sent by the first node, wherein the subset includes the second node.

15 **[0074]** 10. The method of any of clauses 1-9, further comprising receiving a second message from a third node in the network adjacent to the first node; validating the second message; and forwarding the second message to the second node, wherein the forwarded second message is secured using the key.

20 **[0075]** 11. The method of any of clauses 1-10, wherein the receiving the second message, validating, and forwarding steps are performed by the first processor operating in the low power mode in accordance with a determination that an outage event is detected.

**[0076]** 12. The method of any of clauses 1-11, wherein the receiving the second message, validating, and forwarding steps are performed by a main processor at the first node operating in a normal mode in accordance with a determination that an outage event is not detected.

25 **[0077]** 13. In some embodiments, one or more non-transitory computer-readable media at a first node in a network, the one or more computer-readable storage media store program instructions that, when executed by one or more processors at the first node, cause the one or more processors at the first node to perform the steps of receiving a key from a second node in the network, wherein the second node is  
30 adjacent to the first node; storing the key in a first memory, wherein the first memory

is capable of operating in a low power mode; detecting an outage event; in response to detecting the outage event, operating a first processor included in the one or more processors in the low power mode; and via the first processor operating in the low power mode: generating a message; securing the message using the key; and  
5 sending the message to the second node.

**[0078]** 14. The one or more non-transitory computer-readable storage media of clause 13, wherein the first node comprises the first processor and a second processor distinct from the first processor, and wherein the second processor is a main processor of the first node.

10 **[0079]** 15. The one or more non-transitory computer-readable storage media of clauses 13 or 14, wherein the first node comprises the first memory and a second memory distinct from the first memory, wherein the second memory is a main memory of the first node.

**[0080]** 16. The one or more non-transitory computer-readable storage media of  
15 any of clauses 13-15, wherein securing the message using the key comprises retrieving the key from the first memory via the first processor operating in the low power mode.

**[0081]** 17. The one or more non-transitory computer-readable storage media of any of clauses 13-16, wherein the steps further comprise receiving a second  
20 message from a third node in the network adjacent to the first node; validating the second message; and forwarding the second message to the second node, wherein the forwarded second message is secured using the key.

**[0082]** 18. The one or more non-transitory computer-readable storage media of any of clauses 13-17, wherein the network comprises a plurality of nodes adjacent to  
25 the first node, wherein the plurality of nodes includes the second node, and wherein the steps further comprise selecting a subset of the plurality of nodes adjacent to the first node as designated recipients of outage event messages sent by the first node, wherein the subset includes the second node.

**[0083]** 19. The one or more non-transitory computer-readable storage media of  
30 any of clauses 13-18, wherein the steps further comprise for each node in the

selected subset: generating a second key based on a trust level associated with the node; and sending the generated second key to the node.

**[0084]** 20. In some embodiments, a system in a wireless mesh network comprises a plurality of nodes included in a wireless mesh network; a first node included in the plurality of nodes; a second node included in the plurality of nodes and adjacent to the first node, the second node configured to perform the steps of receiving a key from the first node; storing the key in a first memory at the second node, wherein the first memory is capable of operating in a low power mode; detecting an outage event at the second node; in response to detecting the outage event, operating a first processor at the second node in the low power mode; and via the first processor operating in the low power mode: generating a message; retrieving the key from the first memory, wherein the first memory is operating in the low power mode; securing the message using the key; and sending the message to the first node.

**[0085]** Any and all combinations of any of the claim elements recited in any of the claims and/or any elements described in this application, in any fashion, fall within the contemplated scope of the present protection.

**[0086]** The descriptions of the various embodiments have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments.

**[0087]** Aspects of the present embodiments may be embodied as a system, method or computer program product. Accordingly, aspects of the present disclosure may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "module," a "system," or a "computer." In addition, any hardware and/or software technique, process, function, component, engine, module, or system described in the present disclosure may be implemented as a circuit or set of circuits. Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

**[0088]** Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

**[0089]** Aspects of the present disclosure are described above with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the disclosure. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine. The instructions, when executed via the processor of the computer or other programmable data processing apparatus, enable the implementation of the functions/acts specified in the flowchart and/or block diagram block or blocks. Such processors may be, without limitation, general purpose processors, special-purpose processors, application-specific processors, or field-programmable gate arrays.

**[0090]** The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present disclosure. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more

executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

**[0091]** While the preceding is directed to embodiments of the present disclosure, other and further embodiments of the disclosure may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

**WHAT IS CLAIMED IS:**

1. A computer-implemented method for sending a message associated with an outage event, comprising, at a first node in a network:
  - receiving a key from a second node in the network, wherein the second node is
  - 5 adjacent to the first node;
  - storing the key in a first memory, wherein the first memory is capable of
  - operating in a low power mode;
  - detecting an outage event;
  - in response to detecting the outage event, operating a first processor in the low
  - 10 power mode; and
  - via the first processor operating in the low power mode:
    - generating a message;
    - securing the message using the key; and
    - 15 sending the message to the second node.
2. The method of claim 1, wherein the first node comprises the first processor and a second processor distinct from the first processor, and wherein the second processor is a main processor of the first node.
- 20 3. The method of claim 1, wherein the first processor is a main processor of the first node.
4. The method of claim 1, wherein the first node comprises the first memory and a second memory distinct from the first memory, wherein the second memory is a
- 25 main memory of the first node.
5. The method of claim 1, wherein:
  - the first memory is a main memory of the first node; and
  - the method further comprises, in response to detecting the outage event,
  - 30 operating the first memory in the low power mode.
6. The method of claim 1, wherein securing the message using the key comprises retrieving the key from the first memory via the first processor operating in the low power mode.

7. The method of claim 1, wherein the key is generated by the second node.
8. The method of claim 1, wherein the message comprises a key identifier of the  
5 key.
9. The method of claim 1, wherein:  
the network comprises a plurality of nodes adjacent to the first node, wherein  
the plurality of nodes includes the second node, and  
10 wherein the method further comprises selecting a subset of the plurality of  
nodes adjacent to the first node as designated recipients of outage  
event messages sent by the first node, wherein the subset includes the  
second node.
- 15 10. The method of claim 1, further comprising:  
receiving a second message from a third node in the network adjacent to the  
first node;  
validating the second message; and  
forwarding the second message to the second node, wherein the forwarded  
20 second message is secured using the key.
11. The method of claim 10, wherein the receiving the second message, validating,  
and forwarding steps are performed by the first processor operating in the low power  
mode in accordance with a determination that an outage event is detected.  
25
12. The method of claim 10, wherein the receiving the second message, validating,  
and forwarding steps are performed by a main processor at the first node operating in  
a normal mode in accordance with a determination that an outage event is not  
detected.  
30
13. One or more non-transitory computer-readable media at a first node in a  
network, the one or more computer-readable storage media storing program  
instructions that, when executed by one or more processors at the first node, cause  
the one or more processors at the first node to perform the steps of:

receiving a key from a second node in the network, wherein the second node is adjacent to the first node;  
storing the key in a first memory, wherein the first memory is capable of operating in a low power mode;  
5 detecting an outage event;  
in response to detecting the outage event, operating a first processor included in the one or more processors in the low power mode; and  
via the first processor operating in the low power mode:  
generating a message;  
10 securing the message using the key; and  
sending the message to the second node.

14. The one or more non-transitory computer-readable storage media of claim 13, wherein the first node comprises the first processor and a second processor distinct  
15 from the first processor, and wherein the second processor is a main processor of the first node.

15. The one or more non-transitory computer-readable storage media of claim 13, wherein the first node comprises the first memory and a second memory distinct from  
20 the first memory, wherein the second memory is a main memory of the first node.

16. The one or more non-transitory computer-readable storage media of claim 13, wherein securing the message using the key comprises retrieving the key from the first memory via the first processor operating in the low power mode.  
25

17. The one or more non-transitory computer-readable storage media of claim 13, wherein the steps further comprise:

receiving a second message from a third node in the network adjacent to the first node;  
30 validating the second message; and  
forwarding the second message to the second node, wherein the forwarded second message is secured using the key.

18. The one or more non-transitory computer-readable storage media of claim 13, wherein:

the network comprises a plurality of nodes adjacent to the first node, wherein the plurality of nodes includes the second node, and

5 wherein the steps further comprise selecting a subset of the plurality of nodes adjacent to the first node as designated recipients of outage event messages sent by the first node, wherein the subset includes the second node.

10 19. The one or more non-transitory computer-readable storage media of claim 18, wherein the steps further comprise:

for each node in the selected subset:

generating a second key based on a trust level associated with the node; and

15 sending the generated second key to the node.

20. A system in a wireless mesh network, comprising:

a plurality of nodes included in a wireless mesh network;

a first node included in the plurality of nodes;

20 a second node included in the plurality of nodes and adjacent to the first node,

the second node configured to perform the steps of:

receiving a key from the first node;

storing the key in a first memory at the second node, wherein the first memory is capable of operating in a low power mode;

25 detecting an outage event at the second node;

in response to detecting the outage event, operating a first processor at the second node in the low power mode; and

via the first processor operating in the low power mode:

generating a message;

30 retrieving the key from the first memory, wherein the first memory is operating in the low power mode;

securing the message using the key; and

sending the message to the first node.