

CERTIFICATE CHAIN COMPRESSION TO EXTEND NODE OPERATIONAL LIFETIME

CROSS-REFERENCE TO RELATED APPLICATIONS

5 **[0001]** This application claims benefit of United States patent application serial number 16/515,953 filed July 18, 2019, which is hereby incorporated herein by reference.

BACKGROUND

Field of the Various Embodiments

10 **[0002]** Various embodiments relate generally to wireless networks and, more specifically, to certificate chain compression to extend node operational lifetime.

Description of the Related Art

15 **[0003]** A wireless network can include several different types of nodes that are coupled to one another and configured to wirelessly communicate with one another. Two types of nodes included in many wireless networks are mains-powered device (MPD) nodes and battery-powered device (BPD) nodes. MPD nodes are coupled to mains power, such as a power grid, and have continuous access to power when performing various node activities. BPD nodes are powered with battery cells, such as Lithium-ion battery cells, and have to perform node activities using only the limited amount of power available through those battery cells.

20 **[0004]** When a given BPD node joins a wireless network, the given BPD node typically identifies one or more neighboring nodes that already have network access and initiates a certificate exchange with the neighboring node(s). To initiate the certificate exchange with an identified neighboring node, the given BPD node transmits a certificate chain to the neighboring node. In response, the given BPD
25 node receives a certificate chain from the neighboring node. Based on the exchanged certificate chains, the given BPD node and the neighboring node establish identity and authentication between each other, and the neighboring node can then provide the given BPD node with network access. Subsequently, additional nodes can join the wireless network by performing a certificate exchange with the given BPD
30 node. Certificate chains usually include several certificates and can sometimes be quite large. In some instances, the certificate chains can be 900 bytes or greater in size.

[0005] One drawback of the above approach is that, when dealing with large certificate chains, a BPD node can expend a substantial amount of battery power transmitting a certificate chain to a given neighboring node and then receiving and processing a certificate chain transmitted back by that neighboring node. Expending the limited battery power of a BPD node in this fashion can reduce the operational life of the BPD node or can result in the battery cells of the BPD node needing premature replacement, which is time-consuming and expensive.

[0006] As the foregoing illustrates, what is needed in the art are more effective ways for BPD nodes to perform certificate exchanges in wireless networks.

SUMMARY

[0007] Some embodiments include a computer-implemented method for performing an authentication procedure when joining a wireless network, including compressing a certificate chain based on a first mapping to generate a compressed certificate chain, wherein the certificate chain includes a first data entry and the compressed certificate chain includes a first index in place of the first data entry, and transmitting the compressed certificate chain to a neighboring node to perform the authentication procedure with the neighboring node based on the first data entry.

[0008] At least one technological advantage of the disclosed techniques relative to the prior art is that, because a compressed certificate chain is smaller in size compared to a corresponding uncompressed certificate chain, a node consumes less power transmitting the compressed certificate chain to a neighboring node compared to transmitting the uncompressed certificate chain to the neighboring node.

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 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.

[00010] Figure 1 illustrates a network system configured to implement one or more aspects of the present embodiments;

[0010] Figure 2 is a more detailed illustration of one of the BPD nodes of Figure 1, according to various embodiments;

5 **[0011]** Figure 3 illustrates how an instance of the BPD node of Figure 2 initiates a certificate exchange procedure with another instance of the BPD node of Figure 2, according to various embodiments;

[0012] Figure 4 illustrates how the firmware application of Figure 2 compresses and decompresses certificate chains using a compression dictionary, according to
10 various embodiments;

[0013] Figure 5 illustrates how the firmware application of Figure 2 compresses and decompresses certificate chains using static and dynamic portions of a compression dictionary, according to various embodiments;

[0014] Figure 6 is a flow diagram of method steps for compressing an
15 uncompressed certificate chain, according to various embodiments; and

[0015] Figure 7 is a flow diagram of method steps for decompressing a compressed certificate chain, according to various embodiments.

DETAILED DESCRIPTION

[0016] In the following description, numerous specific details are set forth to
20 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.

[0017] As noted above, a wireless mesh network can include several different types of nodes, including mains-powered device (MPD) nodes and battery-powered
25 device (BPD) nodes. MPD nodes are coupled to mains power and therefore have access to a plentiful supply of power with which to perform various node activities, while BPD nodes include battery cells and therefore have access to a limited supply of battery power with which to perform various node activities. When a given BPD node joins the wireless mesh network, the given BPD node identifies one or more
30 neighboring nodes and transmits a certificate chain to the neighboring node(s) as part

of a certificate exchange. The certificate chain is used to authenticate the neighboring node(s) and facilitates encrypted communications between the given BPD node and the neighboring node(s). The certificate chain typically includes several certificates and can be quite large in size, in some instances 900 bytes or
5 more.

[0018] One drawback associated with the large size of certificate chains is that BPD nodes sometimes have to expend a significant percentage of limited battery power in order to transmit certificate chains to neighboring node(s) when joining wireless mesh networks. Another drawback associated with the large size of
10 certificate chains is that BPD nodes similarly have to expend a significant percentage of limited battery power in order to receive certificate chains during certificate exchanges. Because BPD nodes have a limited supply of battery power, expending power in order to transmit and receive large certificate chains can constraint the operational lifetime of BPD nodes or cause one or more batteries included in those
15 BPD nodes to need premature replacement, thereby incurring maintenance overhead.

[0019] To address these issues, various embodiments include a BPD node that performs a compression operation with certificate chains to generate compressed certificate chains. The BPD node includes a compression dictionary that indexes various data entries that occur across many certificate chains and/or repeat within a
20 particular certificate chain. The BPD node compresses a given certificate chain by replacing data entries within the given certificate chain with indices to corresponding data entries in the compression dictionary. The indices are smaller in size than the corresponding data entries. A neighboring BPD node also includes the compression dictionary and can decompress a compressed certificate chain by replacing indices
25 included in the compressed certificate chain with the indexed data entries stored in the compression dictionary. BPD nodes dynamically update respective compression dictionaries to include any missing data entries, thereby increasing compression efficiency.

[0020] At least one technological advantage of the disclosed techniques relative to
30 the prior art is that, because a compressed certificate chain is smaller in size compared to a corresponding uncompressed certificate chain, a BPD node consumes less power transmitting the compressed certificate chain to a neighboring node compared to transmitting the uncompressed certificate chain to the neighboring node.

Similarly, the BPD node consumes less power when receiving and processing a compressed certificate chain transmitted back by the neighboring node compared to receiving and processing an uncompressed certificate chain transmitted back by the neighboring node. Accordingly, with the disclosed techniques, BPD nodes can

5 conserve battery power during the certificate exchange process when joining a wireless network. These technological advantages represent one or more technological advancements over prior art approaches.

System Overview

[0021] Figure 1 illustrates a network system configured to implement one or more aspects of the present embodiments. As shown, network system 100 includes a field

10 area network (FAN) 110, a wide area network (WAN) backhaul 120, and a 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.

[0022] FAN 110 includes personal area network (PANs) A, B, and C. PANs A and

15 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 112 and one or more mains-powered device (MPD) nodes 114. PANs B and C further include one or more battery-powered device (BPD) nodes 116.

[0023] MPD nodes 114 draw power from an external power source, such as mains

20 electricity or a power grid. MPD nodes 114 typically operate on a continuous basis without powering down for extended periods of time. BPD nodes 116 draw power from an internal power source, such as a battery. BPD nodes 116 typically operate intermittently and power down, go to very low power mode, for extended periods of time in order to conserve battery power.

[0024] MPD nodes 114 and BPD nodes 116 are coupled to, or included within, a utility distribution infrastructure (not shown) that distributes a resource to consumers. MPD nodes 114 and BPD nodes 116 gather sensor data related to the distribution of the resource, process the sensor data, and communicate processing results and other information to control center 130. Border router nodes 112 operate as access

25 points to provide MPD nodes 114 and BPD nodes 116 with access to control center

30 130.

[0025] Any of border router nodes 112, MPD nodes 114, and BPD nodes 116 are configured to communicate directly with one or more adjacent nodes via bi-directional communication links. The communication links may be wired or wireless links, although in practice, adjacent nodes 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 may compute a current receive channel by evaluating a Jenkins hash function based on a total number of channels and the media access control (MAC) address of the node.

[0026] Each node within a given PAN may implement a discovery protocol to identify one or more adjacent nodes or “neighbors.” A node that has identified an adjacent, neighboring node may establish a bi-directional communication link with the neighboring node. Each neighboring node may update a respective neighbor table to include information concerning the other node, including the MAC address of the other node as well as a received signal strength indication (RSSI) of the communication link established with that node.

[0027] Nodes may compute the channel hopping sequences of adjacent nodes to facilitate the successful transmission of data packets to those nodes. In embodiments where nodes implement the Jenkins hash function, a node computes a current receive channel of an adjacent node using the total number of channels, the MAC address of the adjacent node, and a time slot number assigned to a current time slot of the adjacent node.

[0028] Any of the nodes discussed above may operate as a source node, an intermediate node, or a destination node for the transmission of data packets. A given source node may generate a data packet and then transmit the data packet to a destination node via any number of intermediate nodes (in mesh network topologies). The data packet may indicate a destination for the packet and/or a particular sequence of intermediate nodes to traverse in order to reach the destination node. In one embodiment, each intermediate node may include a forwarding database indicating various network routes and cost metrics associated with each route.

[0029] Nodes may transmit data packets across a given PAN and across WAN backhaul 120 to control center 130. Similarly, control center 130 may transmit data packets across WAN backhaul 120 and across any given PAN to a particular node included therein. As a general matter, numerous routes may exist which traverse any
5 of PANs A, B, and C and include any number of intermediate nodes, thereby allowing any given node or other component within network system 100 to communicate with any other node or component included therein.

[0030] Control center 120 includes one or more server machines (not shown) configured to operate as sources for, or destinations of, data packets that traverse
10 within network system 100. The server machines may query nodes within network system 100 to obtain various data, including raw or processed sensor data, power consumption data, node/network throughput data, status information, and so forth. The server machines may also transmit commands and/or program instructions to any node within network system 100 to cause those nodes to perform various
15 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.

[0031] Nodes may likewise include computing device hardware configured to perform processing operations and execute program code. Each node may further
20 include various analog-to-digital and digital-to-analog converters, digital signal processors (DSPs), harmonic oscillators, transceivers, and any other components generally associated with RF-based communication hardware. Figure 2 illustrates an exemplary BPD node 116 that may be included in network system 100.

[0032] Figure 2 is a more detailed illustration of one of the BPD nodes of Figure 1,
25 according to various embodiments. As shown, BPD node 116 includes a computing device 210 coupled to a transceiver 250 and an oscillator 260. Computing device 210 coordinates the operations of BPD node 116. Transceiver 250 is configured to transmit and receive data packets across network system 100 using a range of channels and power levels. Oscillator 260 provides one or more oscillation signals
30 according to which the transmission and reception of data packets can be scheduled.

[0033] Computing device 210 includes a processor 220, input/output (I/O) devices 230, and memory 240, coupled together. 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. I/O devices 230 include devices configured to receive input, devices configured to provide output, and devices configured to both
5 receive input and provide output. Memory 240 may be implemented by any technically feasible storage medium.

[0034] Memory 240 includes a firmware application 242, a compression dictionary 244, an uncompressed certificate chain 246, and a compressed certificate chain 248. Firmware application 242 includes program code that, when executed by processor
10 220, performs any of the node-oriented computing functionality described herein. During discovery, firmware application 242 interfaces with transceiver 250 to coordinate the exchange of data packets with other nodes across network system 100 based on timing signals generated by oscillator 260. Once the discovery process is complete, firmware application 242 causes the node to establish one or more
15 encrypted communication channels with one or more neighboring nodes.

[0035] Compression dictionary 244 is a mapping between relatively small index values and relatively larger data entries that can be found in uncompressed certificate chain 246. Compression dictionary 244 is pre-loaded on BPD node 116 but can be dynamically updated during operation of BPD node 116. Uncompressed certificate
20 chain 246 includes various data entries that can be used to verify the identity of a node within network system 100. Compressed certificate chain 248 is a compressed version of uncompressed certificate chain 246.

[0036] In one embodiment, uncompressed certificate chain 246 may include identifying information that BPD node 116 provides to a neighboring node in order to
25 establish an encrypted communication channel with the neighboring node. In this embodiment, firmware application 242 may perform a compression operation with uncompressed certificate chain 246 using compression dictionary 244 to generate compressed certificate chain 248. In so doing, firmware application 242 replaces data entries included in uncompressed certificate chain 246 with indices to corresponding
30 data entries in compression dictionary 244. The indices consume fewer bits than the data entries. Thus, compressed certificate chain 248 has a smaller size compared to uncompressed certificate chain 246.

- [0037]** In another embodiment, uncompressed certificate chain 246 may include identifying information that a neighboring node provides to BPD node 116 in order to establish an authenticated communication channel with BPD node 116. In this embodiment, firmware application 242 may receive compressed certificate chain 248
5 from the neighboring node and then perform a decompression operation with compressed certificate chain 248 using compression dictionary 244 to generate uncompressed certificate chain 246. In so doing, firmware application 242 replaces indices included in compressed certificate chain 248 with indexed data entries extracted from compression dictionary 244.
- 10 **[0038]** Firmware application 242 is configured to perform either or both of the above embodiments in order to compress certificate chains for transmission or to decompress received certificate chains. Various techniques for compressing and decompressing data are described in greater detail below in conjunction with Figures 4-5.
- 15 **[0039]** To establish an authenticated communication channel with a neighboring BPD node 116, firmware application 242 performs a certificate exchange procedure with the neighboring BPD node 116 to generate a shared secret key. Firmware application 242 can then encrypt data that is sent to the neighboring BPD node 116 using the shared secret key and decrypt data that is received from the neighboring
20 BPD node 116 using the shared secret key. Techniques for establishing shared secret keys via certificate exchange procedures are well known to those skilled in the art and not covered here in additional detail.
- [0040]** When initiating the certificate exchange, firmware application 242 performs the compression operation described above with uncompressed certificate chain 246
25 to generate compressed certificate chain 248. Firmware application 242 transmits compressed certificate chain 248 to the neighboring BPD node 116, and the neighboring BPD node 116 performs the decompression operation described above with compressed certificate chain 248 to reconstruct uncompressed certificate chain 246. These operations are described in greater detail below in conjunction with
30 Figure 3.
- [0041]** Figure 3 illustrates how an instance of the BPD node of Figure 2 initiates a certificate exchange procedure with another instance of the BPD node of Figure 2,

according to various embodiments. As shown, a portion 300 of FAN 110 of Figure 1 includes BPD nodes 116(0) and 116(1), or an MPD node (114), coupled together via a communication link 310. BPD node 116(0) includes firmware application 242(0), compression dictionary 244(0), uncompressed certificate chain 246(0), and
5 compressed certificate chain 248(0). BPD node 116(1) includes firmware application 242(1), compression dictionary 244(1), uncompressed certificate chain 246(1), and compressed certificate chain 248(1).

[0042] In operation, firmware applications 242(0) and 242(1) establish communications between BPD nodes 116(0) and 116(1) via the discovery process
10 described above in conjunction with Figure 1. Firmware application 242(0) then initiates a certificate exchange procedure with firmware application 242(1) in order to establish an authenticated communication channel between BPD nodes 116(0) and 116(1) or between BPD and MPD nodes.

[0043] In particular, firmware application 242(0) performs the above-mentioned
15 compression operation with uncompressed certificate chain 246(0) using compression dictionary 244(0) to generate compressed certificate chain 248(0). In one embodiment, uncompressed certificate chain 246(0) may include one or more X.509 certificates that include various fields and corresponding field values related to cryptographically securing communications between two or more entities. Firmware
20 application 242(0) encodes compressed certificate chain 248(0) for transmission via communication link 310 and transmits the encoded version of compressed certificate chain 248(0) to BPD node 116(0).

[0044] Firmware application 242(1) within BPD node 116(1) decodes the encoded version of compressed certificate chain 248(0) from communication link 310 to
25 generate compressed certificate chain 248(1). Under nominal conditions, compressed certificate chain 248(1) is a copy of compressed certificate chain 248(0).

[0045] Firmware application 242(1) then performs the above-mentioned decompression operation with compressed certificate chain 248(1) using compression dictionary 244(1) to generate uncompressed certificate chain 246(1). Under nominal
30 conditions, uncompressed certificate chain 246(1) is a copy of uncompressed certificate chain 246(0). Accordingly, in one embodiment, uncompressed certificate chain 246(1) may similarly include one or more X.509 certificates that include various

fields and corresponding field values related to cryptographically securing communications between two or more entities.

- [0046]** Using uncompressed certificate chain 246(1), firmware application 242(1) is configured to verify the identity of BPD node 116(0) and to establish an authenticated communication channel between BPD nodes 116(0) and 116(1) using techniques known to those skilled in the art. In one embodiment, BPD node 116(1) may transmit a compressed certificate to BPD node 116(0) that can be used to verify the identity of BPD node 116(1) and can be used to establish an authenticated and/or encrypted communication channel between BPD nodes 116(0) and 116(1).
- [0047]** Importantly, compressed certificate chain 248(0) can be significantly smaller than uncompressed certificate chain 246(0). For example, compressed certificate chain 248(0) could have a size that is 50% of the size of uncompressed certificate chain 246(0). Accordingly, BPD node 116(0) consumes less battery power when transmitting compressed certificate chain 248(0) than would otherwise be consumed in transmitting that certificate chain in uncompressed form. Similarly, BPD node 116(1) consumes less battery power when receiving and processing compressed certificate chain 248(1) than would otherwise be consumed in receiving and processing that certificate chain in uncompressed form. For these reasons, the disclosed techniques conserve the limited supply of battery power associated with BPD nodes 116(0) and 116(1) and therefore extend the operational lifetimes of BPD nodes 116(0) and 116(1).

- [0048]** Firmware application 242 of Figure 2 performs both the compression and decompression operations described above in conjunction with Figures 2-3. In so doing, firmware application 242 can operate using a static portion of compression dictionary 242, as described below in conjunction with Figure 4, and additionally using a dynamic portion of compression dictionary 242, as described below in conjunction with Figure 5.

Compressing and Decompressing via a Compression Dictionary

- [0049]** Figure 4 illustrates how the firmware application of Figure 2 compresses and decompresses certificate chains using a compression dictionary, according to various embodiments. As shown, compression dictionary 244 includes an exemplary mapping between indices 0-3 and respective data entries 0-3. Uncompressed

certificate chain 246 includes data entries 0-3, where data entry 2 repeats once.
Compressed certificate chain 248 includes indices 0-3, where index 2 repeats once.

- [0050]** In one embodiment, firmware application 242 may compress uncompressed certificate chain 246 for transmission to a neighboring node. In this
- 5 embodiment, firmware application 242 may perform a compression operation by replacing data entries included in uncompressed certificate chain 246 with corresponding indices from compression dictionary 244. The indices consume a fewer number of bits than the corresponding data entries, causing compressed certificate chain 248 to be smaller than uncompressed certificate chain 246.
- 10 **[0051]** In another embodiment, firmware application 242 may decompress compressed certificate chain 248 upon receipt from a neighboring node. In this embodiment, firmware application 242 may perform a decompression operation by replacing indices included in compressed certificate chain 248 with corresponding data entries from compression dictionary 244.
- 15 **[0052]** According to the above embodiments, firmware application 242 may compress uncompressed certificate chains for transmission to other nodes and decompress compressed certificate chains received from other nodes.
- [0053]** Compression dictionary 244 is pre-populated with mappings between indices and data entries that recur across certificate chains and is pre-loaded onto
- 20 each BPD node 116. Data entries that recur across certificate chains could include, for example, various data associated with BPD nodes 116, such as a manufacturer name associated with BPD nodes 116 or a product line name associated with BPD nodes 116, among others. In one embodiment, compression dictionary 244 may be at least partially populated at manufacture time by performing a frequency analysis
- 25 across all data entries included in any certificates with which BPD nodes 116 are configured to use for authentication and encryption purposes and then indexing those data entries within compression dictionary 244.
- [0054]** In some situations, certificate chains may include data entries that recur across certificate chains but are not included in the pre-populated portion of the
- 30 compression dictionary 244. Such data entries could include, for example, the name of a corporation that owns and/or operates BPD nodes 116 or other customer-specific

information. In these situations, firmware application 242 can dynamically update compression dictionary 244 with these recurring data entries, as described below in conjunction with Figure 5.

[0055] Figure 5 illustrates how the firmware application of Figure 2 compresses and decompresses certificate chains using a static portion and a dynamic portion of a compression dictionary, according to various embodiments. As shown, compression dictionary 244 includes a static portion 500 that includes the pre-populated exemplary mapping between indices 0-3 and respective data entries 0-3 that is shown in Figure 4. Compression dictionary 244 also includes a dynamic portion 510 that firmware application 242 populates with indexed data entries upon detecting a data entry that is not present in compression dictionary 244, as described in greater detail below. Uncompressed certificate chain 246 includes data entry 0, data entry 4, and data entry 3, and data entry 4 repeats once. Compressed certificate chain 248 includes index 0, data entry 4, index 3, and index 4.

[0056] In embodiments where firmware application 242 compresses uncompressed certificate chain 246 for transmission to a neighboring node, firmware application may replace data entries included in uncompressed certificate chain 246 with corresponding indices from compression dictionary 244, in like fashion as described above in conjunction with Figure 5. In addition, if firmware application 242 detects a data entry during compression that is not included in compression dictionary 244, firmware application 242 indexes that data entry in dynamic portion 510 of compression dictionary 244. In the example shown, during the compression operation, firmware application 242 could parse uncompressed certificate dictionary 246 and determine that data entry 4 is not indexed in compression dictionary 244. Firmware application 242 would then add data entry 4 to dynamic portion 510 of compression dictionary 244, indexed via index 4. When indexing a new data entry in this manner, firmware application 242 includes the first occurrence of the raw data entry in compressed certificate chain 248 to allow a receiving BPD node 116 to detect that data entry and then index the data entry accordingly. Subsequently, firmware application 242 includes the index for that data entry in compressed certificate chain 248 in the manner shown.

[0057] In other embodiments where firmware application 242 decompresses compressed certificate chain 248 upon receipt from a neighboring node, firmware

application 242 is required to replace indices included in compressed certificate chain 248 with corresponding data entries from compression dictionary 244. Those data entries will include any indexed data entries found in either static portion 500 or dynamic portion 510 of compression dictionary 244. In addition, if firmware

5 application 242 identifies a raw data entry within compressed certificate chain 248 during decompression, firmware application 242 indexes that data entry in dynamic portion 510 of compression dictionary 244. Indexing such data entries allows firmware application 242 to map subsequent indices to that data entry. In the example shown, during the decompression operation, firmware application 242 could

10 add data entry 4 to dynamic portion 510, indexed via index 4, and then subsequently map index 4 to data entry 4 when index 4 is subsequently detected. In some embodiments, firmware application 242 may flush dynamic portion 510 of compression dictionary 244 when the certificate exchange is complete.

[0058] Referring generally to Figures 4-5, firmware application 242 can be

15 configured to implement either or both of the above-described techniques in order to compress and decompress certificate chains, according to various embodiments. The disclosed techniques allow firmware application 242 to decrease the size of a certificate chain prior to transmission, thereby reducing the amount of battery power needed to transmit the certificate chain. The disclosed techniques also allow

20 firmware application 242 to receive and process compressed certificate chains, thereby decreasing the amount of battery power needed to implement a certificate exchange compared to prior art techniques.

[0059] In one embodiment, decreasing the size of certificate chains allows BPD nodes 116 to forego caching certificate chains. For example, a given BPD node 116

25 could receive a compressed certificate chain, determine that the certificate chain is compressed, and then prevent the compressed certificate chain from being cached. In another embodiment, a given BPD node 116 may store or cache a compressed portion of a certificate or certificate chain and/or other authentication data between security protocol state transitions. For example, the given BPD node 116 could cache

30 a MAC address portion of a certificate chain. In yet another embodiment, firmware application 242 may implement elliptic curve (EC) point compression on the public key when compressing certificate chains. Persons skilled in the art will understand how the techniques described herein can be implemented to improve the

effectiveness with which certificates can be transmitted in any number of technically feasible usage scenarios beyond those described above.

Compressing an Uncompressed Certificate Chain

[0060] Figure 6 is a flow diagram of method steps for compressing an

5 uncompressed certificate chain, according to various embodiments. Although the method steps are described in conjunction with the systems of Figures 1-5, persons skilled in the art will understand that any system configured to perform the method steps in any order falls within the scope of the present embodiments.

[0061] As shown, a method 600 begins at step 602, where firmware application

10 242 within a BPD node 116 obtains a certificate chain that is uncompressed. The certificate chain may be issued to the BPD node 116 at the time of manufacture, or the certificate chain may be issued to the BPD node 116 upon first joining FAN 110. The certificate chain generally includes information that the BPD node 116 can use for authentication, authorization, and encryption operations.

15 **[0062]** At step 604, firmware application 242 obtains a data entry from the certificate chain. The data entry can include any technically feasible form of data. For example, the data entry could be a key / value pair, a field name and corresponding field value, a line of comma-separated values, a data string, and so forth. In one embodiment, the data entry may be a field and/or corresponding field
20 value derived from an X.509 certificate.

[0063] At step 606, firmware application 242 determines whether compression dictionary 244 includes the data entry. Compression dictionary 244 includes a static portion 500 that is pre-populated at the time of manufacture of the BPD node 116 and configured to include data entries that recur across certificates and certificates chain
25 with which BPD node 116 is configured use during operation. Compression dictionary 244 also includes a dynamic portion 510 that firmware application 242 can update to include data entries dynamically determined to recur.

[0064] If at step 606 firmware application 242 determines that compression dictionary 244 includes the entry, then the method 600 proceeds to step 608. At step
30 608, firmware application 242 replaces the data entry with an index to the data entry to compress the certificate chain. The index to the data entry maps to the data entry

in compression dictionary 244. The index to the data entry generally consumes fewer bits than the data entry itself, thereby reducing the amount of bits needed to represent the certificate chain. The method 600 proceeds from step 608 to step 612, described below.

5 **[0065]** If at step 606 firmware application 242 determines that compression dictionary 244 does not include the data entry, then the method 600 proceeds to step 610. At step 610, firmware application 242 adds the data entry to dynamic portion 510 of compression dictionary 244 without replacing first occurrence of the data entry in the compressed certificate chain. Firmware application 242 does not replace the
10 first occurrence of the data entry in order to allow a receiving BPD node 116 to detect the data entry and perform a similar update to the dynamic portion of a compression dictionary associated with the receiving BPD node 116. The method 600 proceeds from step 610 to step 612, described below.

15 **[0066]** At step 612, firmware application 242 determines whether all data entries in the certificate chain been processed. If any data entries remain, then the method 600 returns to step 604 and proceeds as described above. Otherwise, the method 600 proceeds to step 614 where firmware application 242 transmits the compressed certificate chain to a neighboring BPD node 116. In this manner, the BPD node 116 can initiate a certificate exchange in order to authenticate and proceed to join a
20 wireless network while conserving battery power. Upon receipt of a compressed certificate chain, a BPD node 116 implements the techniques described below in conjunction with Figure 7 to decompress the compressed certificate chain.

Decompressing a Compressed Certificate Chain

25 **[0067]** Figure 7 is a flow diagram of method steps for decompressing a compressed certificate chain, according to various embodiments. Although the method steps are described in conjunction with the systems of Figures 1-5, persons skilled in the art will understand that any system configured to perform the method steps in any order falls within the scope of the present embodiments.

30 **[0068]** As shown, a method 700 begins at step 702, where firmware application 242 within a BPD node 116 obtains a certificate chain that has been compressed. The certificate chain is compressed using the technique described above in conjunction with Figure 6. Accordingly, the certificate chain can include indices into

compression dictionary 244 as well as data entries that should be added to dynamic portion 510 of compression dictionary 244.

[0069] At step 704, firmware application 242 parses a first portion of the certificate chain. As mentioned, the certificate chain can include both indices and data entries.

5 Accordingly, the first portion could be an index into compression dictionary 244 or a data entry. At step 706, firmware application 242 determines whether the first portion of the certificate chain is an index or a data entry.

[0070] If at step 706 firmware application 242 determines that the first portion of the certificate chain is an index, then the method proceeds to step 708. At step 708,
10 firmware application 242 replaces the index with a corresponding data entry derived from compression dictionary 244. The corresponding data entry could reside in static portion 500 of compression dictionary 244 or dynamic portion 510 of compression dictionary 244. Because the corresponding data entry has a larger size than the index, replacing the index with the corresponding data entry in the manner described
15 decompresses the certificate chain to a certain extent. The method 700 proceeds from step 708 to step 712, described below.

[0071] If at step 706 firmware application 242 determines that the first portion of the certificate chain is a data entry, then the method proceeds to step 710. At step
20 710, firmware application 242 allocates an index for the first occurrence of the data entry and associates the data entry with the newly allocated index. If the index subsequently appears in the certificate chain, firmware application 242 can map the index to the data entry. The method proceeds from step 710 to step 712, described below.

[0072] At step 712, firmware application 242 determines whether all portions of the
25 certificate chain been processed. If any portions remain, then the method 700 returns to step 704 and proceeds as described above. Otherwise, the method 700 proceeds to step 714 where firmware application 242 initiates a key exchange with a neighboring BPD node using the decompressed certificate chain.

[0073] Referring generally to Figures 6-7, firmware application 242 within a BPD
30 node 116 implements both the compression operations associated with the method 600 and the decompression methods associated with the method 700 in order to

facilitate power-efficient certificate exchanges. Moreover, because firmware application 242 implements certificate exchanges using compressed certificate chains, firmware application 242 can implement certificate exchanges faster than possible with conventional techniques. Accordingly, a given BPD node 116 can join or re-join FAN 110 faster when using the disclosed techniques, thereby increasing network uptime.

[0074] In sum, a BPD node performs a compression operation with certificate chains to generate compressed certificate chains. The BPD node includes a compression dictionary that indexes various data entries that occur across many certificate chains and/or repeat within a particular certificate chain. The BPD node compresses a given certificate chain by replacing data entries within the given certificate chain with indices to corresponding data entries in the compression dictionary. The indices are smaller in size than the corresponding data entries. A neighboring BPD node also includes the compression dictionary and can decompress a compressed certificate chain by replacing indices included in the compressed certificate chain with the indexed data entries stored in the compression dictionary. In some modes of operation, BPD nodes dynamically update respective compression dictionaries to include any missing data entries, thereby increasing compression efficiency.

[0075] At least one technological advantage of the disclosed techniques relative to the prior art is that, because a compressed certificate chain is smaller in size compared to a corresponding uncompressed certificate chain, a BPD node consumes less power transmitting the compressed certificate chain to a neighboring node compared to transmitting the uncompressed certificate chain to the neighboring node. Similarly, the BPD node consumes less power when receiving and processing a compressed certificate chain transmitted back by the neighboring node compared to receiving and processing an uncompressed certificate chain transmitted back by the neighboring node. Accordingly, with the disclosed techniques, BPD nodes can conserve battery power during the certificate exchange process when joining a wireless network. These technological advantages represent one or more technological advancements over prior art approaches.

[0076] 1. Some embodiments include a computer-implemented method for performing an authentication procedure when joining a wireless network, comprising

compressing a certificate chain based on a first mapping to generate a compressed certificate chain, wherein the certificate chain includes a first data entry and the compressed certificate chain includes a first index in place of the first data entry, and transmitting the compressed certificate chain to a neighboring node to perform the authentication procedure with the neighboring node based on the first data entry.

[0077] 2. The computer-implemented method of clause 1, wherein first data entry includes a greater number of bits than the size of the first index in bits, and the certificate chain includes a greater number of bits than the compressed certificate chain.

[0078] 3. The computer-implemented method of any of clauses 1-2, wherein the certificate chain comprises a chain of X.509 certificates, the first data entry comprises at least one of a field name and a field value included in the chain of X.509 certificates, and wherein the authentication procedure comprises a certificate exchange that, when completed, establishes an authenticated communication channel with the neighboring node.

[0079] 4. The computer-implemented method of any of clauses 1-3, wherein compressing the certificate chain comprises determining that the first mapping includes the first data entry, determining that the first data entry corresponds to the first index in the first mapping, and replacing the first data entry with the first index in the certificate chain.

[0080] 5. The computer-implemented method of any of clauses 1-4, wherein compressing the certificate chain comprises identifying a second data entry included in the certificate chain, determining that the first mapping does not include the second data entry, and incorporating the second data entry into the first mapping at a position that corresponds to a second index.

[0081] 6. The computer-implemented method of any of clauses 1-5, further comprising flushing the second data entry and the second index from the first mapping when the authentication procedure has completed.

[0082] 7. The computer-implemented method of any of clauses 1-6, wherein the first mapping is pre-populated with a first set of data entries prior to compressing the certificate chain, wherein compressing the certificate chain comprises dynamically

populating the first mapping with a second set of data entries, and further comprising removing the second set of data entries from the first mapping when the authentication procedure has completed.

[0083] 8. The computer-implemented method of any of clauses 1-7, further
5 comprising decompressing, at the neighboring node, the compressed certificate chain based on a second mapping to generate an uncompressed certificate chain, wherein the uncompressed certificate chain includes the first data entry in place of the first index.

[0084] 9. The computer-implemented method of any of clauses 1-8, further
10 comprising determining, at the neighboring node, that the second mapping does not include the first data entry, and incorporating the first data entry into the second mapping at a position that corresponds to the first index.

[0085] 10. Some embodiments include a computer-implemented method for
15 performing an authentication procedure when joining a wireless network, comprising receiving a compressed certificate chain from a neighboring node, decompressing the compressed certificate chain based on a first mapping to generate a decompressed certificate chain, wherein the compressed certificate chain includes a first index and the decompressed certificate chain includes a first data entry in place of the first index, and performing an authentication procedure with the neighboring node based
20 on the first data entry.

[0086] 11. The computer-implemented method of clause 10, wherein
decompressing the compressed certificate chain comprises determining that the first mapping includes the first index, determining that the first index corresponds to the first data entry in the first mapping, and replacing the first index with the first data
25 entry in the certificate chain.

[0087] 12. The computer-implemented method of any of clauses 10-11, further
comprising extracting a second data entry from the compressed certificate chain,
determining that the first mapping does not include the second data entry, and
incorporating the second data entry into the first mapping at a position that
30 corresponds to a second index.

[0088] 13. The computer-implemented method of any of clauses 10-12, further comprising determining that the authentication procedure has completed, and in response, flushing the second data entry and the second index from the first mapping.

5 **[0089]** 14. The computer-implemented method of any of clauses 10-13, further comprising determining that the compressed certificate chain is compressed, and in response, preventing the compressed certificate chain from being stored in a cache memory.

[0090] 15. The computer-implemented method of any of clauses 10-14, wherein the certificate chain comprises a chain of X.509 certificates, and the first data entry
10 comprises at least one of a field name and a field value included in the chain of X.509 certificates.

[0091] 16. The computer-implemented method of claim 10, wherein the authentication procedure comprises a certificate exchange that, when completed, establishes an authenticated communication channel with the neighboring node.

15 **[0092]** 16. The computer-implemented method of any of clauses 10-15, wherein the authentication procedure comprises a certificate exchange that, when completed, establishes an authenticated communication channel with the neighboring node.

[0093] 17. Some embodiments include a wireless network, comprising a first node that includes a first memory storing a first set of instructions, and a first processor that
20 executes the first set of instructions to perform the steps of compressing a certificate chain to generate a compressed certificate chain, and transmitting the compressed certificate chain across the wireless network, and a second node that includes a second memory storing a second set of instructions, and a second processor that executes the second set of instructions to perform the steps of receiving the
25 compressed certificate chain, decompressing the compressed certificate chain to generate a decompressed certificate chain, and performing an authentication procedure with the first node based on the decompressed certificate chain.

[0094] 18. The wireless network of clause 17, wherein the first processor compresses the certificate chain by determining that the certificate chain includes a
30 first data entry, determining that a first mapping also includes the first data entry,

determining that the first data entry corresponds to a first index in the first mapping, and replacing the first data entry with the first index in the certificate chain.

[0095] 19. The wireless network of any of clauses 17-18, wherein first data entry includes a greater number of bits than the size of the first index in bits, and the
5 certificate chain includes a greater number of bits than the compressed certificate chain.

[0096] 20. The wireless network of any of clauses 17-19, wherein the second processor decompresses the compressed certificate chain by determining that the compressed certificate chain includes a first index, determining that a second
10 mapping also includes the first index, determining that the first index corresponds to the first data entry in the second mapping, and replacing the first index with the first data entry in the certificate chain.

[0097] 21. The wireless network of any of clauses 17-20, wherein the second processor performs the additional steps of extracting a second data entry from the
15 compressed certificate chain, determining that a second mapping does not include the second data entry, and incorporating the second data entry into the second mapping at a position that corresponds to a second index.

[0098] 22. The wireless network of any of clauses 17-21, wherein the certificate chain comprises a chain of X.509 certificates, and a first data entry included in the
20 chain of X.509 certificates comprises at least one of a field name and a field value.

[0099] 23. The wireless network of any of clauses 17-22, wherein compressing the certificate chain comprises performing an elliptic curve compression operation on a public key portion of the certificate chain.

[0100] Any and all combinations of any of the claim elements recited in any of the
25 claims and/or any elements described in this application, in any fashion, fall within the contemplated scope of the present embodiments and protection.

[0101] 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.

[0102] Aspects of the present embodiments may be embodied as a system, method or computer program product. Accordingly, aspects of the present disclosure
5 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” or “system.” In addition, any hardware and/or software technique, process, function, component, engine, module, or system
10 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.

[0103] Any combination of one or more computer readable medium(s) may be
15 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
20 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
25 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.

[0104] Aspects of the present disclosure are described above with reference to
30 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
5 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.

[0105] The flowchart and block diagrams in the figures illustrate the architecture,
10 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
15 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
20 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.

[0106] While the preceding is directed to embodiments of the present disclosure,
25 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.