Data Authentication for Software Updates redistributed by Wireless Multi-hop Nodes

Taketsugu Yao **Jun Nakashima**
Kiyoshi Fukui

920MHz wireless multi-hop networks are networks positioned at endpoints of the highly anticipated M2M (Machine-to-Machine) systems. These networks have been proposed for use in such applications as home/office energy management, structural health monitoring and environmental monitoring. A 920 MHz wireless multi-hop network is autonomously formed from several 920MHz wireless devices (nodes) that are geographically distributed. Therefore, remote management of the network and the nodes becomes important.

One capability of the remote network/node management is the software update function. For example, maintenance of systems that operate for long periods, such as the structural health monitoring, requires software updating from a remote location using wireless communication to avoid having the workers retrieve each node. The problem is authenticating the software update sent via wireless. If a node accepts fraudulent data, the stability of the overall system cannot be guaranteed. A method to confirm the validity of multicast software updates is the digital signature technology, which can authenticate the data integrity and distribution source. However, digital signature verification will require algorithms including public key cryptography, hash function and multiple length arithmetic program to be implemented in the nodes. On the other hand, there have been proposals in which the 920MHz wireless devices use only symmetric key cryptography, used for communication data encryption and authentication, to authenticate data integrity and distribution source of multicast data[1), 2)].

This article proposes a data authentication method based on symmetric key cryptography for authenticating software updates distributed over a 920MHz wireless multi-hop network and describes the technique's superiority over the digital signature technology.

Update Distribution and Security Requirements

In order to perform safe updates of remote software, the nodes must authenticate the received updates as valid data sent from the management server.

If the management server distributes updates to individual nodes via unicast, only a pre-distributed pairwise key needs to be shared individually between the management server and each node for authenticating the validity of the updates. However, unicast distribution to the large number of nodes forming a multi-hop network will increase communication traffic. For efficient distribution of updates, it is preferable to generate/distribute data that can be authenticated by all nodes in an update node group. As an example, there have been proposals for each relay node in a multi-hop network to redistribute updates to relay destinations on behalf of the management server[3), 4)]. Nevertheless, since the use of symmetric key cryptography requires the same authentication key to be shared between the management server and the update node group, it is possible for an attacker in the group to impersonate the management server and introduce fraudulent data.

Therefore, OKI has decided to meet the following security requirements for remote software updates in a multi-hop network.

- Each node in an update node group shall authenticate the distribution source of multicast update uniquely based on symmetric key cryptography.
- If a node has already obtained an update and proceeds to redistribute that update on behalf of the management server to other nodes in the node group, each node shall authenticate the original distribution source of the redistributed update uniquely.
- Updates shall be concealed from those outside the node group.

Existing Methods and Issues

Securely distributing update keys to individual nodes in a group is one way of concealing updates from those outside the node group. In this method, the management server and individual nodes share a pre-distributed pairwise key. Using the pairwise key, the management server encrypts the update key and notifies it to the

target node. When it is time to distribute an update, the management server uses the update key to generate an authentication code for the update and encrypts the update. Each node will use the update key it received previously to obtain the update securely. However, in this approach, the attacker notified of the update key will be able to impersonate the management server and spoof the other nodes. **Figure 1** shows an example of such an impersonation attack. In **Figure 1**, the attacker tricks the other nodes in the update group to authenticate fraudulent data as a correct update from the management server.

Additionally, there have been proposals for methods that only use symmetric key cryptography to authenticate multicast data yet are resistive to management server impersonation attacks[1), 2)]. Authentication of the update's distribution source with these methods required synchronous authentication of the entire group and the previously mentioned security requirement to "authenticate the original distribution source of the redistributed update uniquely" could not be fulfilled.

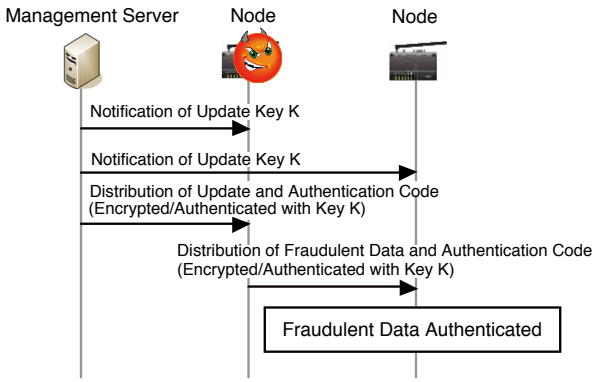

Figure 1. Management Server Impersonation Attack

Authentication of Redistributed Data

A data authentication method that only uses symmetric key cryptography, but allows redistribution from other nodes in an update group while preventing impersonation attacks is proposed.

In this proposed method, the management server securely notifies the multicast update key and multicast update authentication value (for example, authentication code or hash value for the multicast update) using a secure unicast communication channel based on the pairwise key with the node. An overview of the method operation is shown in **Figure 2**. The method consists of the following two steps.

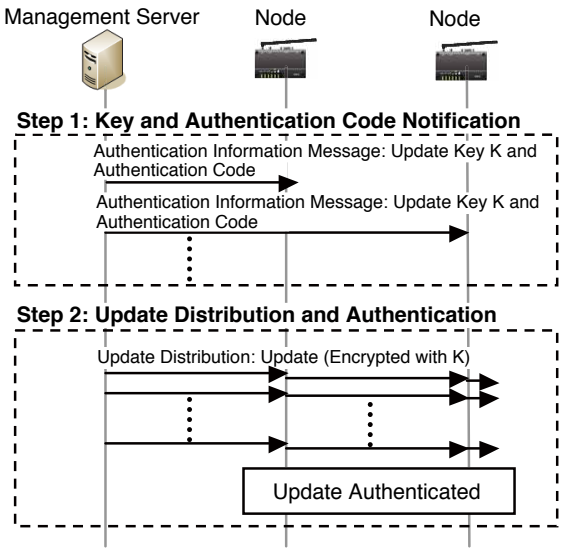

Figure 2. Multicast Update Authentication using the Proposed Method

- **Step 1: Key and Authentication Code Notification**

When delivering multicast update, the management server generates an update key K then uses K to generate an authentication code for the update. Next, the management server creates an authentication information message consisting of the update key K and update authentication code. To prove the management server is indeed the message source, the message is encoded using the pairwise key shared individually with each node. Then an authentication code for the message is appended before the message is distributed.

After receiving the message, each node will decrypt and authenticate the message using the pairwise key shared with the management server. If successful, each node will securely have in its possession the update key K and update authentication code.

- **Step 2: Update Distribution and Authentication**

The management server creates an update distribution message. Since the update size is expected to range from a few kB to several hundred kB, update distribution message will be fragmented into multiple update distribution packets before being sent. The management server encrypts the update with the key K to conceal the update from those outside the group and multicasts each update distribution packet to the target node group.

Each node in the node group restores the data from the update distribution packets and uses the key K obtained in Step 1 to decrypt and obtain the update. Then each node will use the key K to generate an authentication code for the retrieved update and verifies the generated

code matches the one obtained in Step 1. If the codes match, the node will authenticate the restored data as a valid update distributed by the management server.

In addition, the proposed method allows relay nodes that have already authenticated and obtained an update to redistribute the update in place of the management server. **Figure 3** shows an example of relay node A acting as a proxy for the management server to deliver the update. In the **Figure 3**, the relay node A, which has authenticated and obtained the update, notifies the identification information of the update it has in its possession ((i) Update Notice). When node B receives identification information for an update it has not obtained, it requests a key from the management server to authenticate the update ((ii) Authentication Information Request) and receives the key K and authentication code ((iii) Authentication Information Message). Thereafter, node B requests delivery of the update from node A ((iv) Update Request), and node A complies by sending the update to node B ((v) Update Distribution).

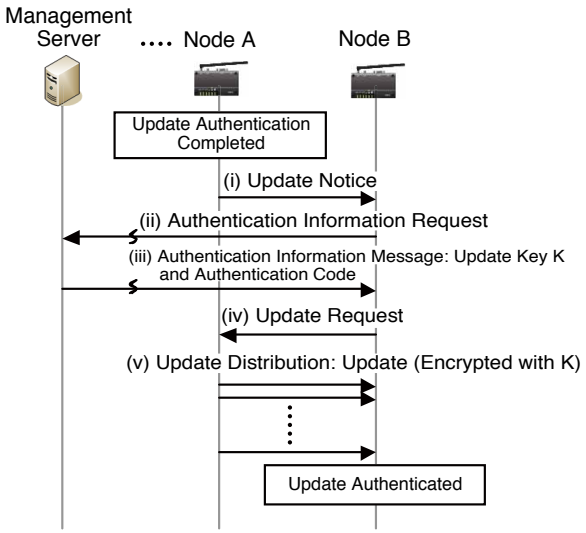

Figure 3. Update Redistribution in the Proposed Method

Security Consideration

In this section, the proposed method's resistance to spoofing attacks is discussed. As shown in **Figure 2**, the proposed method notifies to each node both the key K and the authentication code for the multicast update. When each node possesses the correct authentication code along with the correct key K, it is difficult even for an attacker with the correct key K to make the other nodes authenticate fraudulent data. This is because of the computational difficulty at finding fraudulent data that will output the same authentication code as the true update.

Feature Comparison with Digital Signature

When digital signature is used to authenticate the distribution source of a multicast update, the management server generates a signature using its private key and attaches the signature to the update. The nodes use the management server's public key to verify the attached signature.

Comparison of the proposed method's features with the digital signature technology is discussed below from the perspectives of safety when a node is compromised, memory usage and communication volume.

(1) Safety when a node is compromised

In digital signature technology, the server's private key required to generate a valid signature is not revealed even if a node is compromised. Thus, it is difficult for an attacker to introduce fraudulent data into a node. On the other hand, in the proposed method, when a node is compromised and the pairwise key is revealed, an attacker can plant a set of false authentication key and code into the compromised node. In the proposed method, it is necessary to protect the pairwise key from unauthorized disclosure and under this prerequisite, the same level of safety as the digital signature technology can be ensured.

(2) Memory Usage

Digital signature technology requires memory space to implement hash function and public key cryptography used in signature verification. Memory usage for example on an EFM32-GG390[5] to perform a 256-bit ECDSA (Elliptic Curve Digital Signature Algorithm) signature verification was estimated to require about 24kB of ROM. However, the encryption function is implemented based on OpenSSL, and the memory usage includes elliptic curve cryptography (secp256r1) required for ECDSA signature verification, hash function (SHA-256) and multiple length arithmetic program. On the other hand, the authentication code used in the proposed method can be generated using only symmetric key cryptography. The use of symmetric key cryptography is common for encrypting and authenticating wireless communication data. A 128-bit AES encryption can be supported with about 3kB of ROM on a 32-bit microcomputer, and in recent years, many microcomputers equipped with hardware AES have appeared[5]. Hence, one can consider it is possible to realize the proposed method with only the encryption normally provided in the 920MHz wireless devices.

(3) Communication Volume

In digital signature technology, a digital signature is attached to the distributed update, which each node uses in the verification of the update. The proposed method, in addition to distributing the update, must individually notify each node of the authentication information. Thus, the communication volume is greater compared with the digital signature technology. However, since the proposed method distributes a common key to each node in a group, it simultaneously provides update authentication and concealment of the update through encryption. This function cannot be realized with digital signature technology, which only has functionality to authenticate an update. For digital signature technology to conceal the update from those outside the node group, it would require the pre-distribution of an encryption key to each node in the group similar to the proposed method.

When the overall procedure of an update protocol is considered in general, it is necessary for the management server to notify each node of the coming update, the update size and other information in advance. The notification of the key and authentication code required for the proposed method can be included in such advance notices, thus notification of authentication information necessary for the proposed method in not method-specific traffic.

Although the proposed method requires protection of the pairwise key from unauthorized disclosure, under the assumption the protection is incorporated in the update procedure, the method has an advantage over the digital signature technology in terms of memory usage.

Conclusion

A multicast update authentication method with redistribution capability has been proposed for remotely managing a 920MHz wireless multi-hop network. Although it is a multicast update distribution method based solely on symmetric key cryptography, the proposed method is able to conceal an update from those outside the group, allows redistribution from other nodes within the group and resistive to source impersonation attacks. The plan now is to integrate the proposed function into the update procedure of existing 920MHz wireless devices and evaluate the performance for practical use. ◆◆

References

1) A. Perrig, et al., “SPINS: Security Protocols for Sensor Networks,” Wireless Networks Journal, vol. 8, no. 5, 2002, pp. 521-534
2) T. Yao, S. Fukunaga, and T. Nakai, “Reliable broadcast message authentication in wireless sensor networks,” LNCS, vol. 4097, 2006, pp. 271-280
3) J. W. Hui and D. Culler, “The dynamic behavior of a data dissemination protocol the dynamic behavior of a data dissemination protocol for network programming at scale,” ACM SenSys, Baltimore, Maryland, USA, November 2004
4) S. S. Kulkarni, et al., “MNP: Multihop network reprogramming service for sensor networks,” in IEEE ICDCS, Columbus, Ohio, USA, June 2005
5) Energy Micro, “EFM32GG390 DATASHEET (2011-02-04 d0040_Rev0.90)”

Authors

Taketsugu Yao, Smart Solution Business Innovation Department, Corporate Research & Development Center

Jun Nakashima, Smart Solution Business Innovation Department, Corporate Research & Development Center

Kiyoshi Fukui, Smart Solution Business Innovation Department, Corporate Research & Development Center