Responding to Security Issues in WiMAX Networks

Chin-Tser Huang, *University of South Carolina*
J. Morris Chang, *Iowa State University*

WiMAX technology has attracted significant attention and interest because of its long transmission range, high transmission rate, and mobility support. But to make WiMAX networks usable and reliable, several security issues must be addressed in both the standard and its protocols.

With its promise of a target transmission range of up to 30 miles and a target transmission rate of more than 100 Mbps, WiMAX—the commercialization of the evolving IEEE 802.16 standard—is the latest hot technology. A recent amendment boosted the protocol's mobility support, making it even more attractive to roaming users. The standard specifies both a common physical layer and a media access control (MAC) layer for WiMAX networks, but to make such networks more usable and reliable, several security issues must be addressed first. In this article, we introduce the security protocols used in WiMAX networks and discuss the problems they confront.

Security Sublayer

In a WiMAX network, the base station (BS) and the subscriber station (SS) face most of the same security threats and attacks as their wired counterparts, including eavesdropping, masquerading, session hijacking, message modification, message replay, denial of service, and so on. Moreover, wireless networks—regardless of whether they use WiMAX or Wi-Fi LANs as defined in the IEEE 802.11 standard—are inherently less secure than wired networks because they lack a physical infrastructure. To address security problems, the 802.16 standard specifies a security sublayer at the bottom of the MAC layer to provide the SS with authentication and privacy and to protect

 Published by the IEEE Computer Society 1520-9202/08/$25.00 © 2008 IEEE

Table 1. Comparison of Wi-Fi and WiMAX security mechanisms.

	Authentication	Key management	Encryption	Secure multicast
802.11i (Wi-Fi)	Uses 802.1x and the Extensible Authentication Protocol (EAP) to forward messages; establishes a pairwise master key (PMK) through four-way handshake between wireless station (WS)	Derives four temporal keys from the PMK to encrypt and verify the integrity of Extensible Authentication Protocol over LAN (EAPOL) handshake and user data every time the WS associates with an AP	Its use of Wired Equivalent Privacy (WEP) with RC4 as cipher has problems; defines Temporal Key Integrity protocol (TKIP) to address WEP problems and transition to Counter Mode CBC-MAC (AES-CCM)	Creates a group master key (GMK) and then derives group transient key (GTK) from GMK and distributes to each station in secure pairwise connection
802.16e (WiMAX)	Initializes an authorization state machine at the subscriber station (SS) to execute the authorization protocol, which authenticates the SS to the base station (BS) and establishes an authorization key (AK) and one or more security associations (SAs)	For each SA, the SS initializes a traffic encryption key (TEK) state machine that uses the key management protocol and an AK to manage the secure exchange and periodic update of TEKs	Specifies the use of two cipher suites: Data Encryption Standard-Cipher Block Chaining (DES-CBC) and Advanced Encryption Standard-Counter Mode CBC-MAC (AES-CCM)	Uses a multicast and broadcast rekeying algorithm to provide rekeying, but has problems with scalability

the BS from unauthorized network access and service hijacking. This security sublayer has two protocols: a privacy and key management (PKM) protocol helps the BS enforce access control to the SSs and securely distribute keying material, and an encapsulation protocol encrypts packet payloads across fixed broadband wireless access systems.

PKM uses a two-tier key system:

- at startup, SS initializes an authorization state machine that runs the authorization protocol to authenticate the SS to the BS and establish a shared secret (the authorization key [AK]) and one or more security associations (SAs);
- for each SA, the SS then initializes a traffic encryption key (TEK) state machine that uses the key management protocol and the AK to manage the secure exchange and update the TEKs.

To illustrate the new security schemes incorporated in the 802.16 standard, Table 1 compares mechanisms in both 802.16 and 802.11i (Wi-Fi); we discuss them in more detail in the following sections.

Authorization Protocol

The BS uses the authorization protocol to authenticate and authorize network access to an SS. As Figure 1 shows, the protocol consists of three messages:

- The first is an optional authentication information message, in which an SS sends its manufacturer's X.509 certificate to the BS.
- The second is an authorization request (Auth-REQ), in which the SS sends its certificate and information about its capabilities to the BS.
- The third message is an authorization response (Auth-RSP), in which the BS validates the requesting SS's identity, determines the encryption algorithms and protocols to share with the SS, generates an AK, and sends it to the SS.

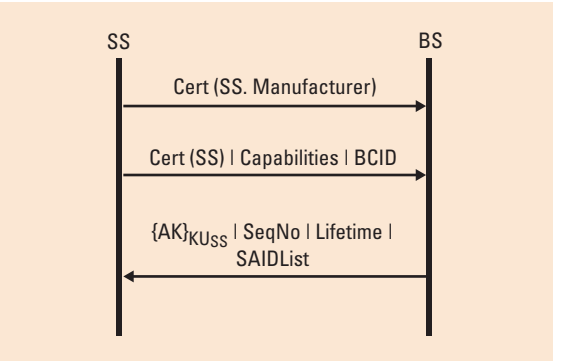

Figure 1. The authorization protocol in IEEE 802.16 standard. The first message is an optional authentication information message from SS to BS, the second message is an authorization request message from SS to BS, and the third message is an authorization response message from BS to SS.

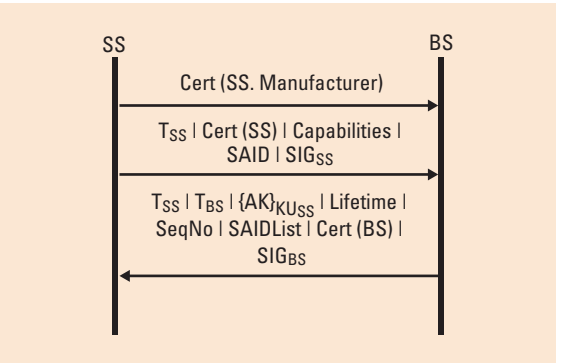

Figure 2. Proposed revision on the authorization protocol. Compared to Figure 1, the revised protocol adds a timestamp and a signature to the second and third messages, and requires BS to send its certificate in the third message.

In the figure, $\{AK\}_{KUSS}$ stands for the AK encrypted by the SS's public key; BCID is the SS's basic connection ID, which is also its primary security association ID (SAID); SeqNo is a 4-bit sequence number for the AK; lifetime represents the number of seconds before the AK expires; and SAIDList contains SA identities and properties for which the SS is authorized to obtain keying information.

The original authentication protocol is quite naïve and vulnerable to *replay attacks*, in which the attacker replays an instance of the second message a legitimate SS sent earlier. If the BS sets a timeout value that makes itself reject duplicate Auth-REQs from the same SS within a certain period, it might also ignore legitimate requests by the victim SS, resulting in a denial of service. An attacker could also replay an instance of the BS's third message to disrupt a legitimate SS's communication. Even worse, the attacker could make its own Auth-RPS message with the AK generated by the attacker itself, thus gaining control of the victim SS's communications. This is a typical *man-in-the-middle attack*.

To counter replay attacks, we could add timestamps to the second and third messages, along with the SS's signature. To counter man-in-the-middle attacks, we could incorporate mutual authentication by requiring the BS to send its certificate in the third message, so that the SS could also authenticate the BS. Figure 2 shows the revised protocol with these modifications.

The 802.16e amendment specifies PKMv2, which revises the original authorization protocol to provide mutual authentication and adds an additional message to provide SS acknowledgment and achieve X.509 three-way authentication. However, this enhanced version is still vulnerable to an *interleaving attack*, in which an attacker first impersonates a legitimate SS to run a first PKMv2 protocol instance and exchange the first two messages of PKMv2 with the BS, then impersonates a legitimate BS to run a second PKMv2 protocol instance with the impersonated SS, and finally uses the third message sent by the impersonated SS in the second protocol instance to reply to the BS as the third message in the first protocol instance. The result is that the attacker gets authenticated as the legitimate SS.

Key Management Protocol

Figure 3 shows the procedure in which the SS begins to request keying materials after completing authentication. It starts by the SS sending a Key-Request message to the BS periodically, corresponding to one of its legitimate SAIDs. The BS responds with a Key-Reply message, containing the BS's active keying material for the specific SAID.

In this protocol, the first message is optional and is sent only if the BS deems it necessary to rekey before the SS requests it. The BS will choose a SAID from the SAIDList, which the SS is allowed to access. SeqNo is the sequence number of the AK that the BS provides to the SS in the authorization protocol. Upon receiving the first message, the SS will reply with a Key-Request. If the SS doesn't receive the first message from the BS before the current key expires, the SS will send the normal Key-Request message when the key is just about to expire (typically, when the SS chooses the SAID from the SAIDList). In the third message, the BS responds with a Key-Reply that includes keying materials. At all times, the BS maintains two active sets of keying materials per SAID: the OldTEK for the currently used TEK and the NewTEK for when it expires. The keying materials include the TEK encrypted by the key encryption key (KEK), which is derived from the AK, the CBC initialization vector used by the encryption algorithm, and the remaining lifetimes of keying materials. Each message in the key management protocol contains a keyed

message authentication code (HMAC) for checking message integrity. The SeqNo lets the BS and the SS determine which HMAC key (also derived from the AK) to use for computing the HMAC.

The SS in the key management protocol is secure from replay attacks because the OldTEK in the recently received Key-Reply message should be the NewTEK in the previous message. However, the BS is still vulnerable to replay attacks in the second message. If an adversary replays the Key-Request message to the BS, the latter can't determine whether it's a fresh request from the SS, so it'll send a Key-Reply message with new keying materials. This can result in frequent exchange of keying materials (thus exhausting the BS's capabilities) or confusion about the TEK's use. As in the authorization protocol, a timestamp is a suitable freshness identifier to counter replay attacks, but the signatures in the authorization protocol messages are unnecessary here because the HMAC already provides message authentication.

Encryption

The 802.16 standard also allows encryption of packets in a WiMAX network to provide confidentiality; Figure 4 illustrates the encryption mechanism in the 802.16 standard. Typically, the payload of a MAC Protocol Data Unit (MPDU) is encrypted, whereas the generic MAC header (GMH) is sent in clear. The BS and the SS decide which cryptographic suite to apply via the SAID and which TEK to use according to the two encryption key sequence bits in the GMH. The 802.16 standard specifies two main cryptographic suites for encryption: Data Encryption Standard-Cipher Block Chaining (DES-CBC) and Advanced Encryption Standard in Counter with CBC-MAC (AES-CCM). The latter is considered state of the art because of the former's insufficient key length and other known vulnerabilities.

Data Content Distribution

The Multicast and Broadcast Service (MBS) in IEEE 802.16e is a mechanism for efficiently distributing data content, especially multimedia traffic, across multiple BSs. The MBS aims to provide subscribers with strong protection from service theft by encrypting broadcast connections between SSs and BSs. Specifically, the MBS uses the multicast and broadcast rekeying

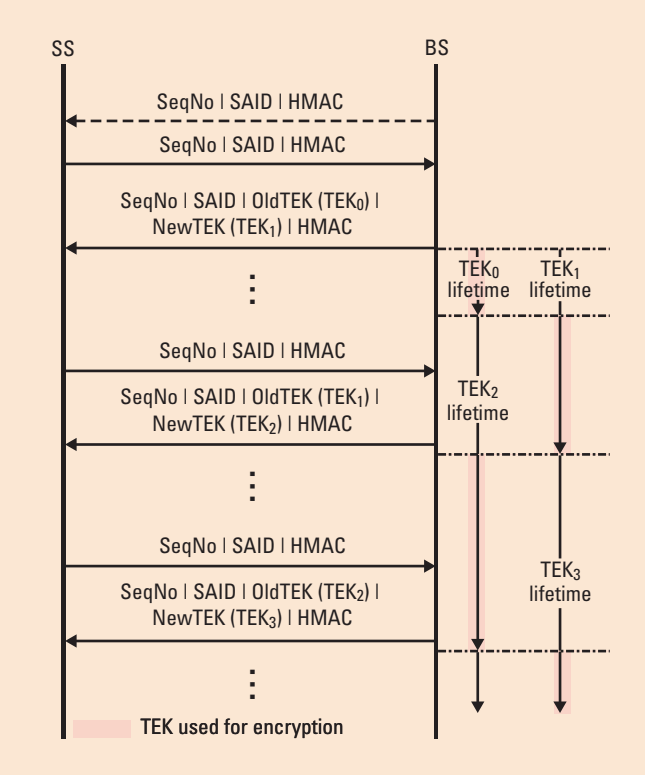

Figure 3. The key management protocol in the IEEE 802.16 standard. Periodically, the SS sends a Key-Request message to the BS, while the BS responds with a Key-Reply message, containing the BS's active keying material. Optionally, if the BS sees an urgent need for rekeying, it can send a rekeying request as shown in the dashed message.

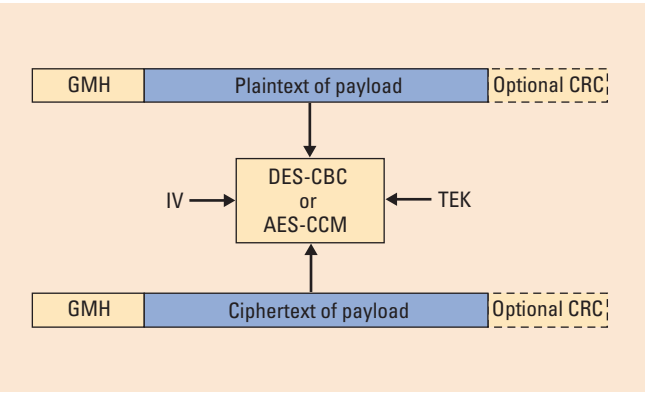

Figure 4. Encryption in the IEEE 802.16 standard. The standard specifies two main cryptographic suites for encryption: Data Encryption Standard-Cipher Block Chaining (DES-CBC) and Advanced Encryption Standard in Counter with CBC-MAC (AES-CCM).

Further Reading on Broadband Wireless Access Security

The following list offers a good overview of security issues and technologies for broadband wireless access networks.

- IEEE Std. 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE Press, 2004.
- IEEE Std. 802.16e-2005, IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, IEEE Press, 2005.
- C. Eklund et al., WirelessMAN: Inside the IEEE 802.16 Standard for Wireless Metropolitan Networks, IEEE Press, 2006.
- C.-T. Huang and M.G. Gouda, Hop Integrity in the Internet, Springer, 2006.
- C.-T. Huang et al., "Efficient and Secure Multicast in WirelessMAN: A Cross-layer Design," J. Comm. Software and Systems, vol. 3, no. 3, 2007, pp. 199–206.
- D. Johnston and J. Walker, "Overview of IEEE 802.16 Security," IEEE Security & Privacy, vol. 2, no. 3, 2004, pp. 40–48.
- S. Xu and C.-T. Huang, "Attacks on PKM Protocols of IEEE 802.16 and Its Later Versions," Proc. 3rd Int'l Symp. Wireless Communication Systems (ISWCS 2006), 2006, pp. 185–189.
- S. Xu, M.M. Matthews, and C.-T. Huang, "Security Issues in Privacy and Key Management Protocols of IEEE 802.16," Proc. 44th ACM Southeast Conf. (ACMSE 2006), ACM Press, 2006, pp. 113–118.

algorithm (MBRA) to refresh traffic keying material. Prior to receiving multicast service, an SS must register and authenticate with a BS via the PKM protocol. The BS and SS exchange PKM messages through the *primary management connection*, except that the PKMv2 Group-Key-Update-Command is transferred over the broadcast connection. The BS sends multicast traffic to all SSs in the multicast group and encrypts it using a single group-wide session key called the group traffic encryption key (GTEK). Because every SS must have the current GTEK to decrypt multicast data, the challenge is how to efficiently distribute and update it to all the SSs in the multicast group. A trivial solution is to let the BS securely distribute the updated GTEK to each SS individually when a new SS wants to join the group, a member wants to leave the group, or the current GTEK is about to expire, but this solution isn't scalable because of the multitude of unicast key exchanges.

The MBRA offers an improvement. To ensure timely distribution of the new GTEK before the current one expires, the MBRA uses a group key encryption key (GKEK) to encrypt the new GTEK and broadcast to all the SSs. An SS gets the initial GTEK, which the BS uses to encrypt the multicast traffic, via Key-Request and Key-Reply messages over the primary management connection. A BS updates and distributes the traffic keying material periodically by sending two Group-Key-Update-Command messages: one for the GKEK update mode and the other for the GTEK update mode. Intermittently, a BS transmits the Key-Update-Command message for the GKEK update mode to each SS through its primary management connection. This message contains the new GKEK encrypted with the KEK, which is derived from the AK established during authentication. Then, the BS transmits the Key-Update-Command message for the GTEK update mode through the *broadcast connection*, which contains the new GTEK encrypted with the corresponding GKEK. We can specify the protocol as follows:

$$\mathrm{BS} \rightarrow \mathrm{SS} : \{\mathrm{GKEK}\}_{\mathrm{KEK}} \quad (1)$$

$$\mathrm{BS} \Rightarrow \text{all SS} : \{\mathrm{GTEK}\}_{\mathrm{GKEK}} \quad (2)$$

where $\rightarrow$ stands for a unicast message, and $\Rightarrow$ stands for a broadcast message.

However, this protocol has two problems: first, it isn't scalable because it still needs to unicast to each SS, and, second, it doesn't address the issue of backward and forward secrecy. When a new member receives the current GTEK, for example, it can decrypt all previous messages that were multicast during the same GTEK's lifetime. Nothing in this protocol prevents an SS that's leaving the group from receiving the next GKEK or decrypting the next GTEK.

GTEK lifetime as specified by the IEEE 802.16e standard has great leverage in the relationship between scalability and forward and backward secrecy—currently, the range is set as 0.5 hours

minimum, 12 hours by default, and seven days maximum. A long enough lifetime must be maintained to give the BS enough time to individually update the GKEK and broadcast the new GTEK. However, longer GTEK lifetimes imply much greater lapses in backward and forward secrecy during member join and leave events, respectively, because more messages are encrypted via the given GTEK.

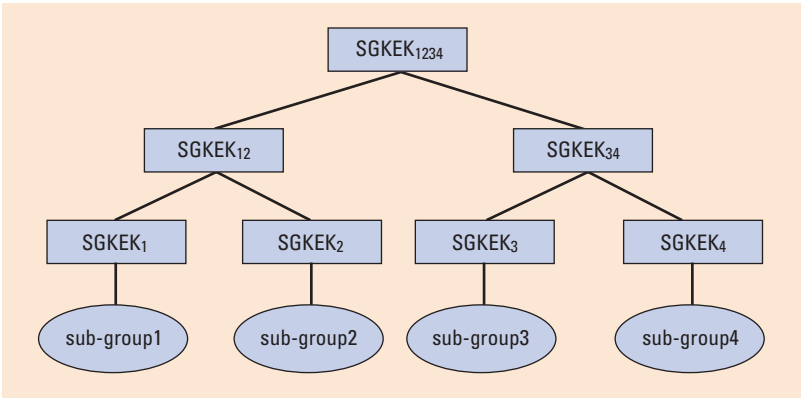

Figure 5. A sample key hierarchy with four subgroups. The members of each subgroup remember the keys on the path from the leaf represented by the subgroup to the key tree's root.

Improving Efficiency with Subgroup Rekeying

We propose a solution—the efficient sublinear rekeying algorithm with perfect secrecy (Elapse)—to address MBRA's two primary problems. Elapse is based on subgrouping the SSs so that the GKEK isn't maintained by unicasting to individual SS but by broadcasting to subgroups. For every cell consisting of a BS and many SSs subscribing to a multicast application, the SSs will be divided into $N = 2^k$ subgroups, with each subgroup maintaining k keys. The implementer of a given application can determine the exact value of N to achieve the application's best performance. When a new SS requests keying material, it moves into the subgroup with the lowest member count to keep the subgroups balanced in size. Otherwise, one subgroup could be much larger than the others, and the efficiency of rekeying will drop significantly.

Each subgroup maintains a hierarchy of subgroup KEKs (SGKEKs) rather than a single GKEK. According to a binary tree hierarchy, each SS within a subgroup will store k SGKEKs; Figure 5 shows the case for $N = 4$. In the figure, subgroup 1 stores $SGKEK_1$, $SGKEK_{12}$, and $SGKEK_{1234}$, which functions as the traditional GKEK did.

In the simplest case of rekeying, no members join or leave during the GTEK's lifetime, and the BS sends only one broadcast:

$$BS \Rightarrow \text{all SS} : \{GTEK\}_{SGKEK1234} \quad (3)$$

In the case of rekeying due to a member join, the joining SS sends a key request to the BS, which sends the joining SS a Key-Reply with a new hierarchy of SGKEKs. For example, when a new SS joins and subgroup 2 currently has the one with the fewest members, the Key-Reply is like Message 4, except that all keys need to be updated:

$$BS \rightarrow \text{new SS} : \{SGKEK_{1234}, SGKEK_{12}, SGKEK_2\}_{KEK} \quad (4)$$

Message 4 is also delivered via unicast to all existing SSs inside subgroup 2. While Message 4 is being delivered, the BS rekeys all existing SSs in other subgroups with new versions of appropriate keys using Messages 5 and 6:

$$BS \Rightarrow SS_{SG3}, SS_{SG4} : \{SGKEK_{1234}\}_{SGKEK34} \quad (5)$$

$$BS \Rightarrow SS_{SG1} : \{SGKEK_{1234}, SGKEK_{12}\}_{SGKEK1} \quad (6)$$

where SS_{SGi} means the collection of all SSs within subgroup i.

For performance reasons, the updated GTEK isn't included in these messages. In a multijoin event, if more SSs attempt to join during the updates, the BS waits until all joining SSs arrive and then places them in the same subgroup. The only addition in a multijoin would be another Message 4 to each additional SS joining the multicast service. At the conclusion of all SGKEK updates during a join or multijoin, the BS broadcasts the new GTEK to all SSs with Message 7:

$$BS \Rightarrow \text{all SS} : \{GTEK\}_{SGKEK1234} \quad (7)$$

Rekeying after a member's departure is much like when a member joins. If a member from group 2 left, the BS would unicast Message 4b

to all remaining SSs in subgroup 2. Next, the BS would broadcast Messages 5b and 6b to the members outside subgroup 2. The difference is that if an SS decides to leave after it has already received new SGKEK material in the middle of another leave process, no rekeying can be combined; instead, another rekeying process must commence. Therefore, the BS directly distributes the new GTEK along with SGKEK rekeying messages:

$$BS \rightarrow SS: \{SGKEK_{1234}, SGKEK_{12}, SGKEK_{2}, GTEK\}_{KEK} \quad (4b)$$

$$BS \Rightarrow SS_{SG3}, SS_{SG4}: \{SGKEK_{1234}, GTEK\}_{SGKEK34} \quad (5b)$$

$$BS \Rightarrow SS_{SG1}: \{SGKEK_{1234}, SGKEK_{12}, GTEK\}_{SGKEK1} \quad (6b)$$

Node mobility is an essential feature of 802.16e. If the multicast group consists of a large number of fast-moving nodes that often join and leave the multicast group, the BS will have to frequently perform the group rekeying, which will hurt overall performance. To mitigate this problem, BS can use the mobility information gathered in the application layer to differentiate the subgroupings. The BS can designate some specific subgroups for fast-moving SS nodes that are projected to leave the multicast group sooner than a predefined duration. The BS can make these special subgroups smaller than other subgroups by selecting an appropriate duration—then, every time a node tagged as fast moving requests to join the multicast group, it will go to one of the designated subgroups rather then the regular subgroup with the lowest member count. This way, when members join or leave, the total number of Message 4 sent by the BS in Elapse (the unicast message) will decrease for fast-moving subgroups, which constitute most of the joins and leaves.

Security Issues in Multihop Communications

When the communication between two end nodes is beyond the range of one BS, a possible solution is to deploy multiple relay stations to relay the traffic between the source and destination BSs. Currently, an IEEE task group is in the process of finishing IEEE 802.16j addendum, the Multi-hop Relay Specification for 802.16. In this case, the system manager must consider the authentication and integrity between each pair of adjacent relay stations in order to provide end-to-end security. A mechanism called *hop integrity* is sufficient to meet these requirements. Encryption and decryption at every hop isn't necessary, but two adjacent relay stations at each hop must establish trust relationships and share a secret that the two relay stations can use to compute an HMAC that the receiving relay station can use to authenticate and check the received message's integrity.

An alternative and more cost-effective solution is to deploy a mesh network, in which the system manager recruits some SSs as mesh nodes to extend coverage. Hop integrity can still be applied to provide the required authentication and integrity between adjacent BSs and mesh nodes and between pairs of adjacent mesh nodes, but the system manager must take the construction of trust relationship between adjacent nodes into account, especially when the mesh nodes have mobility.

Compared to the security standards found in Wi-Fi networks, the protocols we've introduced here incorporate new schemes for authentication and key distribution, but they still need to address the issues of efficiency, scalability, and forward and backward secrecy before they'll be practical for real applications. We've introduced some solutions to address these issues, but they must be further verified and standardized. IT

Chin-Tser Huang is an assistant professor of computer science and engineering at the University of South Carolina. His research interests include network security, network protocol design and verification, and distributed systems. Huang has a PhD in computer science from the University of Texas at Austin. Contact him at huangct@engr.sc.edu.

J. Morris Chang is an associate professor of electrical and computer engineering at Iowa State University. His technical interests include wireless networks, object-oriented programming languages, and embedded computer systems. Chang has a PhD in computer engineering from North Carolina State University. Contact him at morris@iastate.edu.