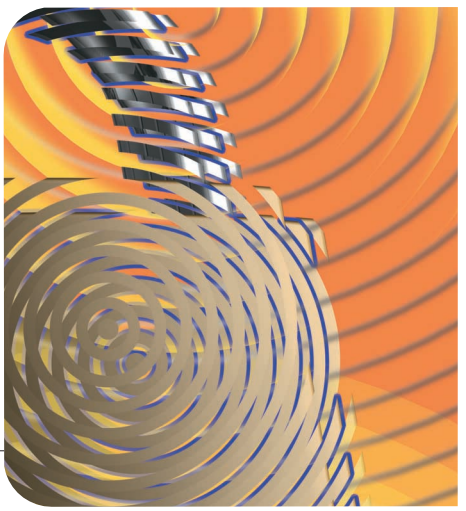
Overview of IEEE 802.16 Security

When creating the new wireless communication standard IEEE 802.16, designers attempted to reuse a security scheme designed for wired media. The authors review the standard, enumerate its flaws, and outline changes that could defend it against threats.



David Johnston and Jesse Walker
Intel

IEEE 802.16,[1] the standard for constructing wireless metropolitan area networks (WMANs), was originally developed to address the "last mile" problem. Until recently, most of the wireless industry and its users mistakenly believed that the standard's major security weakness was its use of 56-bit Data Encryption Standard (DES). In fact, the key size is one of the standard's most insignificant security weaknesses.

The IEEE 802.16 Working Group sought to avoid the design mistakes[2–5] of IEEE 802.11[6] by incorporating a pre-existing standard into IEEE 802.16. The pre-existing standard, DOCSIS (Data Over Cable Service Interface Specifications), was designed to solve the last mile problem for cable. Because cable is a wired technology and IEEE 802.16 is wireless, the two enjoy different threat models. Consequently, IEEE 802.16 security fails to properly protect an IEEE 802.16 link.

This article overviews IEEE 802.16 security and discusses its major security flaws. We also suggest modifications to protect the standard against attack. A glossary defines many of the IEEE 802.16 acronyms used in this article.

Media access control and physical layers

Each of the four main modes of the IEEE 802.16 physical layer (PHY) offers significant flexibility. This flexibility allows operation across a wide range of spectrum allocations, including variations in channel bandwidth, frequency division duplex, and time division duplex. However, all modes support a common feature set, including initial ranging, registration, bandwidth requests, and connection-oriented channels for management and user data. IEEE 802.16 security protocols are the same, regardless of PHY type.

IEEE 802.16 divides on-air communications into frames. Downlink frames (from base station (BS) to subscriber station (SS)) contain a frame header, which includes two slot maps, one for the downlink (DL_MAP) and one for the uplink (UL_MAP). The maps indicate the location, size, and encoding of all the slots in the downlink and uplink frames.

The MAC is connection oriented. Each slot belongs to a certain connection, identified by a connection ID. *Management* connections handle broadcast data, initial ranging, bandwidth requests, and general management messaging. For each SS, a secondary management connection carries Internet Protocol management packets. All other connections are *transport* connections. The IEEE 802.16 link management function dynamically creates transport connections to carry user packets.

IEEE 802.16 protects only transport connections and the secondary management channel.

MPDU packet format. IEEE 802.16 creates packets—or *MAC protocol data units*—to transport data over the connections. MPDUs come in two forms, differentiated by the MPDU header, as Figure 1 shows:

- *bandwidth request header* (BRH), where the header is the entire packet; and
- *generic MAC header* (GMH), which is followed by a payload and optional cyclic redundancy checking (CRC).

A management connection ID identifies management packets. Each management MPDU carries a single MAC management message. Transport connections

carry *MAC service data units*—data units passed by the network stacks above the MAC. IEEE 802.16 affords much flexibility as to how MPDUs carry MSDUs.

Network entry. Network entry involves a sequence of actions:

1. The SS scans for a suitable BS downlink signal, which it uses to establish channel parameters.
2. Initial ranging allows the SS to set PHY parameters correctly and establish the primary management channel with the BS. This channel is used for capability negotiation, authorization, and key management.
3. The privacy and key management (PKM) protocol authorizes the SS to the BS.
4. The SS registers by sending a request message to the BS. The BS's response assigns a connection ID for a secondary management connection.
5. The SS and BS create transport connections using a `MAC_create_connection` request. A request to create a dynamic transport connection indicates whether MAC-level encryption is required.

Security algorithms

IEEE 802.16 security is implemented as a privacy sublayer at the bottom of the MAC protocol's internal layering. Its goal is to provide access control and confidentiality of the data link.

The IEEE 802.16 security architecture uses five components, described in the following subsections.

Security associations. Security associations (SAs) maintain the security state relevant to a connection. IEEE 802.16 uses two SA types but explicitly defines only the data SA, which protects transport connections between one or more SSs and a BS.

The data SA consists of

- A 16-bit SA identifier, or SAID.
- A cipher to protect the data exchanged over the connection. The standard uses DES in cipher block chaining (CBC) mode,[7] but the design is extensible to other algorithms.
- Two traffic encryption keys (TEKs) to encrypt data: the current operational key and a TEK for when the current key expires.
- Two 2-bit key identifiers, one for each TEK.
- A TEK lifetime. The default value for this parameter is half a day and assumes a minimum value of 30 minutes and a maximum value of seven days.
- A 64-bit initialization vector for each TEK.
- An indication of the type of data SA. *Primary* SAs are established during link initialization; *static* SAs are configured on the BS; and *dynamic* SAs are constructed as needed for dynamic transport connections.

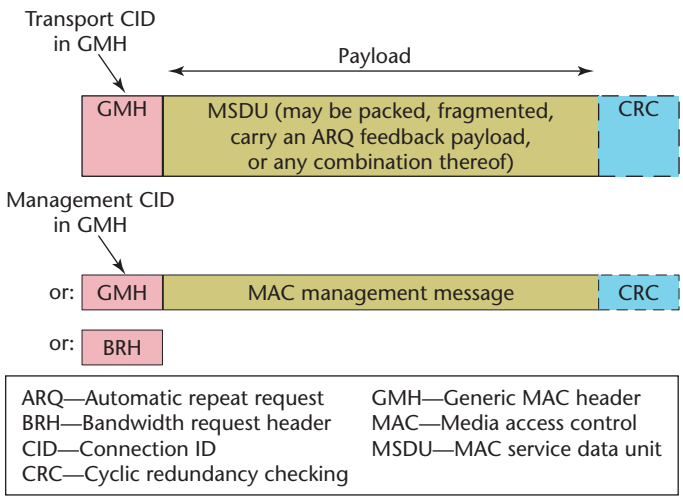


Figure 1. IEEE 802.16 MAC protocol data units. An MPDU has either a bandwidth request header or a generic MAC header.

To secure a transport connection, an SS first initiates a data SA using a `create_connection` request. To support multicast, the standard lets many connection IDs share an SA. On network entry, IEEE 802.16 automatically creates an SA for the secondary management channel. A fixed SS therefore typically has two or three SAs, one for the secondary management channel and either one for both uplink and downlink transport connections or separate SAs for uplink and downlink connections. Each multicast group also requires an SA to share among group members.

The authorization SA, which the standard never explicitly defines, consists of

- An X.509 certificate identifying the SS.
- A 160-bit authorization key (AK). Correct use of this key demonstrates authorization to use IEEE 802.16 transport connections.
- A 4-bit quantity to identify the AK.
- An AK lifetime, ranging from one to 70 days. The default lifetime is seven days.
- A key encryption key (a 112-bit Triple-DES key) for distributing TEKs. The KEK is constructed as: KEK = Truncate-128(SHA1(((AK | 0^{44}) $\oplus$ 53^{64})), where Truncate-128(·) means to discard all but the first 128 bits of the argument, $a \mid b$ denotes the concatenation of strings a and b, $\oplus$ denotes exclusive OR, a^n denotes the octet a repeated n times, and SHA1 is defined by the secure hash standard.[5]
- A downlink hash function-based message authentication code (HMAC) key providing data authenticity of key distribution messages from the BS to the SS. This key is constructed as: Downlink HMAC key = SHA1((AK | 0^{44}) $\oplus$ $3A^{64}$).
- An uplink HMAC key providing data authenticity of

Table 1. Terms used in a privacy key management (PKM) authorization message exchange.

TERM	DESCRIPTION
A → *B*: *M*	Entity *A* sends *B* the message with value *M*
Cert(*Manufacturer*(*SS*))	An X.509 certificate identifying *SS*'s manufacturer
Cert(*SS*)	An X.509 certificate with the *SS* public key
Capabilities	*SS*-supported authentication and data encryption algorithms
SAID	The secure link between *SS* and *BS* (the connection ID)
RSA-Encrypt(*k, a*)	Instruction to RSA-OAEP encrypt its second argument *a* under the key *k*
PubKey(SS)	The SS's public key, as reported in *Cert(SS)*
AK	Authorization key
Lifetime	A 32-bit unsigned number giving the number of seconds before *AK* expires
SeqNo	A 4-bit value for *AK*
SAIDList	A list of SA descriptors, each including an SAID, the SA type—primary, static, or dynamic—and the SA cipher suite

key distribution messages from the SS to the BS. The uplink HMAC key is constructed as: Uplink HMAC key = SHA1((AK | 0^{44}) ⊕ $5C^{64}$).
- A list of authorized data SAs.

An authorization SA is state shared between a particular BS and a particular SS. The design assumes these two stations maintain the AK as a secret. Base stations use authorization SAs to configure data SAs on the SS.

X.509 certificate profile. X.509 certificates identify communicating parties. The standard defines an X.509 certificate profile requiring the following fields:

- X.509 certificate format version 3.
- Certificate serial number.
- Certificate issuer's signature algorithm Public Key Cryptography Standard 1—that is, RSA encryption with SHA1 hashing.[8]
- Certificate issuer.
- Certificate validity period.
- Certificate subject—that is, the certificate holder's identity, which, if the subject is the SS, includes the station's MAC address.
- Subject's public key, which provides the certificate holder's public key, identifies how the public key is used, and is restricted to RSA encryption.
- Signature algorithm, which is identical to the certificate issuer's signature algorithm.
- Issuer's signature, which is the digital signature of the Abstract Syntax Notation 1 Distinguished Encoding Rules (ASN.1 DER) encoding of the rest of the certificate.

IEEE 802.16 does not define X.509 certificate extensions.

The standard defines two certificate types: manufacturer certificates and SS certificates. It does not define BS certificates. A manufacturer certificate identifies the manufacturer of an IEEE 802.16 device. It can be a self-signed certificate or issued by a third party. An SS certificate identifies a particular SS and includes its MAC address in the subject field.

Manufacturers typically create and sign SS certificates. The BS typically uses the manufacturer certificate's public key to verify the SS certificate, and hence identify the device as genuine. This design assumes the SS maintains the private key corresponding to its public key in some sort of sealed storage, preventing attackers from easily compromising it.

PKM authorization. The PKM authorization protocol distributes an authorization token to an authorized SS. The authorization protocol consists of a three-message exchange between a subscriber station *SS* and a base station *BS*. *SS* initiates the protocol by sending the first two messages, and *BS* responds with the third message (see Table 1 for definitions of the terms used):

```
Message 1:
SS → BS Cert(Manufacturer(SS))
Message 2:
SS → BS Cert(SS) | Capabilities |
        SAID
Message 3:
BS → SS RSA-Encrypt(PubKey(SS), AK) |
        Lifetime | SeqNo | SAIDList
```

SS uses Message 1 to push its X.509 certificate *Cert*(*Manufacturer*(*SS*)) to *BS*, which uses it to decide whether *SS* is a trusted device. The design assumes that all devices from a recognized manufacturer can be trusted. IEEE 802.16 lets *BS* ignore this message as its security policy might allow access only to devices known a priori.

SS sends Message 2 immediately after Message 1. Message 2 consists of *SS*'s X.509 certificate *Cert*(*SS*), its security capabilities, and the identity *SAID* of what will be its primary *SS*. *Cert*(*SS*) lets *BS* determine whether *SS*

Table 2. Terms used in a privacy key management (PKM) protocol message exchange.

TERM	DESCRIPTION
[…]	Indicates an optional message
SeqNo	The AK used for the exchange
SAID	The ID of the data SA being created or rekeyed
HMAC(1)	The HMAC-SHA1 digest of *SeqNo* | *SAID* under *AK*'s downlink HMAC key
HMAC(2)	The HMAC-SHA1 digest of *SeqNo* | *SAID* under *AK*'s uplink HMAC key
OldTEK	The previous-generation *TEK*'s initialization vector, remaining lifetime (in seconds), and sequence number for the data SA specified by SAID (the TEK sequence number is a 2-bit quantity)
NewTEK	The next *TEK*'s initialization vector, lifetime (in seconds), and sequence number for the data SA specified by SAID (the TEK sequence number is 1 greater, modulo 4, than the *OldTEK* sequence number)
HMAC(3)	The HMAC-SHA1 digest of *SeqNo* | *SAID* | *OldTEK* | *NewTEK* under *AK*'s downlink HMAC key

is authorized, and the *Cert*(*SS*) public key lets *BS* construct Message 3.

If *BS* can verify *Cert*(*SS*) and *SS* is authorized, it responds with Message 3, which instantiates an authorization SA between the two stations. Correct use of this AK demonstrates authorization to access the WMAN channel. The design assumes that only *BS* and *SS* possess the AK—that is, the key is never revealed to any other party. IEEE 802.16 never constrains this key's generation.

Privacy and key management. A PKM protocol instance establishes a data SA between *BS* and *SS*. The PKM protocol consists of a two- or three-message exchange between *SS* and *BS*. *BS* uses the first message, which is optional, to force rekeying. Otherwise, *SS* initiates the protocol by sending the second message, and *BS* responds with the third message (see Table 2 for definition of the terms used):

```
[Message 1:
BS → SS: SeqNo | SAID | HMAC(1)]
Message 2:
SS → BS: SeqNo | SAID | HMAC(2)
Message 3:
BS → SS: SeqNo | SAID | OldTEK |
         NewTEK | HMAC(3)
```

BS never uses Message 1 unless it wants to rekey a data SA or create a new SA. By computing the value *HMAC*(1), it allows *SS* to detect forgeries.

SS uses Message 2 to request SA parameters. *SS* must take *SAID* from the authorization protocol *SAIDList* or from a Message 1 with valid *HMAC*(1). *SS* generates a separate Message 2 for each data SA. It computes the value *HMAC*(2) to allow *BS* to detect forgeries.

If *HMAC*(2) is valid and *SAID* identifies one of *SS*'s SAs, *BS* configures the SA using Message 3. The *OldTEK* value reiterates the active SA parameters while the *NewTEK* value prescribes parameter values to be used on expiry of the current TEK. *BS* Triple DES encrypts the old and new TEKs under the authorization SA *KEK*, using electronic code book (ECB) mode. The standard imposes no TEK generation requirements. Computing the value *HMAC*(3) lets *SS* detect forgeries.

A valid *HMAC*(2) value authenticates *SS* to *BS*. Two assumptions support this claim:

- only *SS* can unwrap the AK sent in Message 3 of the authorization protocol, and
- *AK* is unpredictable.

The protocol admits no comparable authentication of *BS* to *SS*; indeed, correct *HMAC*(1) and *HMAC*(3) values demonstrate only that a party that knows the AK value received by *SS* in Message 3 constructed key management Messages 1 and 3.

Encryption. DES-CBC encryption, operating over the payload field, enciphers a plaintext MPDU, but not the MPDU GMH or the CRC, as Figure 2 depicts.

The MPDU GMH carries two bits to indicate the TEK being used. It does not carry the CBC mode initialization vector. To calculate the MPDU initialization vector, the IEEE 802.16 encryption module XORs the SA initialization vector with the contents of the PHY synchronization field from the most recent GMH. Because the SA initialization vector is constant and public for its TEK, and because the PHY synchronization field is highly repetitive and predictable, the MPDU initialization vector is also predictable.

IEEE 802.16 provides no data authenticity.

Analysis of IEEE 802.16 security

Several errors exist in IEEE 802.16 security.

WMAN threat model

Security threats apply to both the PHY and MAC levels of IEEE 802.16. Because IEEE 802.11 security operates

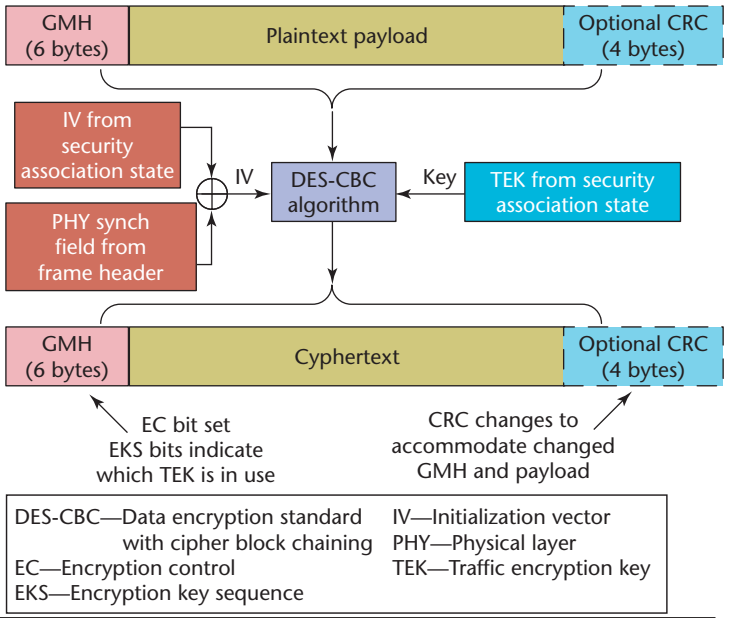


Figure 2. IEEE 802.16 encryption process. The Data Encryption Standard in cipher block chaining mode enciphers the multiprotocol data units but not the MPDU generic MAC header or the cyclic redundancy checking.

wholly at the MAC level, it does nothing to protect against PHY-level attacks.

A typical threat arises from the *water torture* attack, in which an attacker sends a series of frames to drain the receiver's battery. Another threat involves an attacker jamming a radio spectrum, thus denying service to all parties. Because available techniques for defending against PHY-level attacks are insufficient to merit standardization, we focus exclusively on MAC-level threats.

IEEE 802.16's progression from a fixed, line-of-sight, point-to-multipoint, high-frequency system to a lower frequency, near line-of-sight, and, in the future, mobile system, increases the number of threats to users. Attacks against the original standard, IEEE 802.16-2001, required an adversary to physically place the attacking equipment between the SS and BS and be able to operate at the comparatively high frequencies of 10 to 66 GHz.

The IEEE 802.16a standard introduced lower frequency operation, reducing the radio circuit implementation complexity and the physical placement constraints. Mesh modes in IEEE 802.16a introduce new security assumptions that the existing security mechanisms do not address well, such as the trustworthiness of the next-hop mesh node.

Adding mobility to the standard through IEEE 802.16e makes the attacker's life even easier. The attacker's physical location is not greatly constrained, making management messages more vulnerable than in IEEE 802.11. The need to maintain a secured state while a mobile SS moves between BSs introduces new vulnerabilities.

Several threats are generic to any wireless medium. Because IEEE 802.16 uses radio, anyone with a properly positioned radio receiver can intercept messages sent on a wireless channel. Hence the design must define a confidentiality mechanism.

IEEE 802.16 designers failed to address another threat: anyone with a correctly placed and configured radio transmitter can write to a wireless channel. Because of this vulnerability, an attacker could forge new frames and capture, modify, and retransmit frames from authorized parties. The design must therefore also provide a data authenticity mechanism.

An attacker can also resend a valid, already-sent frame unmodified. Interference and distance could allow an attacker to communicate with two authorized parties who cannot communicate directly with each other, and reorder and selectively forward frames. Thus, the design must detect replayed frames.

Lack of explicit definitions

The most striking thing about the IEEE 802.16 design is its failure to explicitly define the authorization SA, meaning it never receives the same attention data SAs receive. Threats against data SAs apply directly to the authorization SA, so this failure will likely lead to problems.

For example, the SA state never distinguishes one authorization SA instance from another, leaving the protocol suite open to replay attacks. In addition, the authorization SA does not include the BS identity, so the SS cannot distinguish authorized from unauthorized BSs. Although hiding the BS identity from the user might be desirable, hiding it from the SS prevents key management and encryption from protecting the SS from forgery and replay attacks.

This causes a related problem for data SAs. Because the SS cannot distinguish reused authorization SAs, it also cannot recognize reused data SAs. The encryption scheme is therefore vulnerable to attack through encryption key reuse.

The safest way to correct the replay vulnerability is to add a random value from the BS and SS to the authorization SA. Requiring input from both parties can protect their contributions. An authenticated BS identity also eliminates the threat against SSs due to the credentials' asymmetry.

Inattention to replay also appears in the data SA definition. The standard treats the 2-bit key identifier as a circular buffer, allowing an attacker to interject reused TEKs. The standard's designers should expand the key identifier space to permit as many key identifiers as can be transported by the largest AK lifetime value. Because an AK can last for up to 70 days, whereas a TEK lifetime can be as short as 30 minutes, a data SA can consume up to 3,360 TEKs over the AK's lifetime, requiring the SAID space to grow from 2 to at least 12 bits.

This raises the related question of when a TEK should expire. In the current standard, the TEK expires after a con-

figurable period of time. Although this is certainly necessary, it is not sufficient. IEEE 802.16 default TEK lifetime is half a day, and the standard permits a maximum TEK lifetime of seven days. These numbers can lead to problems.

Recall that IEEE 802.16 uses DES in CBC mode for encryption. DES uses a 64-bit block size—that is, it operates on 64-bit data blocks to effect each encryption or decryption operation. One theorem says that a CBC mode using a block cipher with an n-bit block cipher loses its security after operating on $2^{n/2}$ blocks with the same encryption key.[9] For DES, $n = 64$, so IEEE 802.16 can safely protect at most 2^{32} 64-bit blocks. An average throughput of 6.36 Mbps produces 2^{32} 64-bit blocks in half a day; an average throughput of 455 Kbps produces 2^{32} 64-bit blocks in the maximum allowed seven days. If the average data rate exceeds that allowed by the lifetime parameter setting, the utility of the encryption scheme is greatly diminished.

Need for mutual authentication

The most obvious flaw of the entire IEEE 802.16 security design is the lack of a BS certificate. The only way to defend the client against forgery or replay attack is to replace the standard's authentication scheme with a scheme providing mutual authentication. Mutual authentication is required for any wireless medium; cabling cost reduction translates into increased credential management costs.

Authorization vulnerabilities

Consistent with the authorization SA's weak design, the PKM authorization protocol that manages it possesses vulnerabilities.

The IEEE 802.16 design's lack of a means for authenticating the BS to the SS leaves the PKM protocol open to forgery attacks. In a forgery attack, the SS cannot verify that any authorization protocol messages it receives were generated by an authorized BS. The BS constructs the authorization protocol responses it sends to an SS using entirely public information, so any rogue BS can create a response. Requiring the SS to authenticate to the BS can eliminate this vulnerability.

The authorization protocol subjects the SS to replay attacks. The simplest way to prevent such an attack is to require the SS to generate a random challenge in Message 2 of the authentication protocol, and the BS to include the challenge in the state it returns authenticating itself to the SS.

A related problem is the protocol's failure to allow participants to distinguish one instance of the protocol from another. This will become important as IEEE 802.16e facilitates mobility and roaming. By exchanging public random numbers, participants can uniquely identify the protocol instance as the 4-tuple <BS's certified identity, SS's certified identity, BS's public random number for this instance, SS's public random number for this instance>. Participants could use this information to tie key management protocol instances to the governing authorization instance.

The authorization protocol exhibits a serious AK-related problem. The standard imposes no requirements on AK generation, even though its later use assumes random generation—that is, an AK is selected using a uniform probability distribution on the space of 160-bit strings. The standard should make this assumption explicit.

Another weakness exists because the BS contributes all of the bits in an AK. This common design means the SS must trust that the BS always generates a new AK that is cryptographically separated from all other AKs generated by all BSs. It also means that the BS's random number generator must be perfect—if it exhibits significant bias, it could expose the AK and hence all the TEKs. A safer design would compute the AK from bits both parties contribute—for example, AK = HMAC-SHA1(BS's AK, some random value generated by SS). Including even a public random value generated by an SS in the AK computation would assure the SS that its keys are fresh.

Finally, the protocol assumes that certificates are correctly issued—that is, no parties with different public or private key pairs are certified to use the same MAC address. If this condition isn't met, each party can masquerade as the other. The specification should explicitly call out its assumption that every certified MAC address is distinct.

The problems with the authorization protocol represent a catastrophic failure of the IEEE 802.16 security design. The key management and encryption portions of IEEE 802.16 security offer no assurance because the security of both rests on the authorization protocol's correctness. This failure demonstrates that security algorithms cannot be transferred from one context to another without great care.

Key management failures

Given the failures of its authorization protocol, it does not matter whether the IEEE 802.16 key management protocol is correct. However, if the authorization protocol design errors were corrected, problems in the key management protocol would still undermine security.

The most serious key management protocol problem is its use of TEK sequence space. The protocol identifies each TEK with a 2-bit sequence number, wrapping the sequence number from 3 to 0 on every fourth rekey. The protocol's use of the sequence number to distinguish messages subjects it to replay attack. If replay succeeds—and nothing in the protocol lets the SS detect this attack—encryption reuses the TEK and initialization vector in the encryption, exposing both the TEK and the subscriber data.

The standard fails to specify that TEKs are randomly generated using a uniform probability distribution and a cryptographic-quality random number generator. Because the encryption scheme requires this condition, the standard should call it out explicitly.

Similarly, the key distribution scheme offers no TEK freshness assurance. This is certainly unavoidable for multicast, but not for unicast. Again, using a key-derivation

Alphabet soup: A guide to IEEE 802.16 terminology

AES: Advanced Encryption Standard
AK: Authorization key
BRH: Bandwidth request header
BS: Base station
CBC: Cipher block chaining
CCM: Counter with cipher-block-chaining MAC
CRC: Cyclic redundancy checking
DES: Data Encryption Standard
DOCSIS: Data Over Cable Service Interface Specifications
DL_MAP: Downlink map
EAP: Extensible authentication protocol
ECB: Electronic code book
GMH: Generic MAC header
HMAC: Hash function-based message authentication code
KEK: Key encryption key
MPDU: MAC protocol data unit
MSDU: MAC service data unit
PKCS: Public Key Cryptography Standard
PKM: Privacy and key management
SA: Security association
SAID: Security association identifier
SHA: Secure hash
SS: Subscriber station
TEK: Traffic encryption key
UL_MAP: Uplink map
WMAN: Wireless metropolitan area network

scheme to mix SS randomness into the BS-supplied TEK easily corrects this problem.

Finally, to prevent replays from succeeding against the key management protocol, the standard should tie messages to a particular protocol instance.

Data protection errors

People understand that DES fails to provide strong data confidentiality. The data protection scheme suffers from much more severe problems, however.

The most important of these problems is the scheme's failure to protect against forgeries or replies, the most serious threats against any wireless data protection scheme. Just like IEEE 802.11's Wired Equivalent Privacy (WEP) protocol, the data protection scheme doesn't defend against forgery. Encryption only read-protects the WMAN channel; it doesn't protect the channel from writes, even by someone without the encryption key.

The protocol also exhibits a severe error in its use of encryption. IEEE 802.16 uses DES in CBC mode. CBC mode requires a random initialization vector to secure the scheme,[9] but IEEE 802.16 uses a predictable initialization vector. Correcting this problem requites generating each per-frame initialization vector randomly and inserting them into the payload. Although this increases the encryption overhead, no other alternative exists.

A way forward

IEEE 802.16 ongoing activities (see the "Alphabet soup" sidebar) afford an opportunity to address the standard's security issues. Designers of the new and proposed security methods share five main goals:

- Use the Advanced Encryption Standard[10,11] as the encryption primitive. Use this in a well-understood mode of operation, such as counter with cipher block chaining MAC (CCM).
- Introduce a more flexible authentication scheme based on the Extensible Authentication Protocol (EAP).[12]
- Promote the authorization SA as a first-class co.ncept within the specification.
- Fix native authorization and key management.
- Permit low cost re-authentication during roaming.

Designers have included some changes to the security mechanisms in 802.16e and might address some of the base standard's flawed security before IEEE 802.16d is completed. All the changes merit close scrutiny by the security community prior to approval of the updated standard.

New data protection scheme

The IEEE 802.16e amendment recently adopted AES-CCM—that is, AES in CCM mode, as a new data link cipher. CCM[13] combines counter mode encryption for data confidentiality with the CBC-MAC for data authenticity. Hence, correct use of AES-CCM addresses the most fundamental deficiency in the original data protection scheme—the lack of a data authenticity mechanism.

Designers chose AES-CCM for a variety of reasons, including its use in IEEE 802.11i and subsequent scrutiny. The US National Institute of Standards and Technology has indicated that CCM will become an approved mode of operation for AES. CCM protects associated data (that is, authenticated but unencrypted data), which lets the encryption scheme protect GMH. No intellectual property claims have been made against CCM.

AES-CCM requires that the transmitter construct a unique nonce, which is a per-packet encryption randomizer. Consistent with the IEEE 802.11i solution, IEEE 802.16e inserts a packet number into each MPDU to ensure each nonce's uniqueness. A receiver validates that received packets correctly decrypt under AES-CCM and have a monotonically increasing packet number.

IEEE 802.11e also specifies AES in ECB mode to re-

Table 3. Terms used in the adapted authentication and key management messages.

TERM	DESCRIPTION
SS-Random	An unpredictable value *SS* generates. *SS-Random* serves two functions: it becomes *SS*'s protocol instance identifier, and it is used by a new AK derivation scheme to guarantee *SS* that the resulting *AK* is fresh.
BS-Random	An unpredictable value generated by BS servicing the same two functions for *BS*.
pre-BS	The base keying material distributed by *BS* to the *SS*. This is identical to the current standard, with the added requirement that *pre-BS* is generated randomly (that is, it's unpredictable)
Cert(*BS*)	An X.509 certificate identifying the BS. It can be implemented with the profile used for the *SS*, but it is used only for signatures in the BS's public/private key pair.
Sig(*BS*)	BS's signature over the other items in Message 3.

place the Triple-DES key wrapping in the PKM protocol. A better choice would be NIST's AES key-wrap algorithm.[14]

EAP authentication

Task group e considered two options for the EAP-based authentication method. The first uses IEEE 802.1X to transport EAP messages. The task group rejected this option because IEEE 802.1X encodes EAP messages as data frames, which assumes that a fully operational data link exists—an untrue assumption for any wireless medium prior to link establishment. The second approach encodes EAP messages directly into IEEE 802.16 management frames. This approach permits authentication during link establishment. IEEE 802.16e introduces two additional PKM messages to transport EAP: `PKM-EAP-REQ` and `PKM-EAP-RSP`.

IEEE 802.16e does not define the authentication method used, and EAP methods to support the needs of wireless networking security are still a research area. However, designers are beginning to articulate generally accepted requirements.[15]

Adapting native authentication and key management

To save the native IEEE 802.16 PKM, we add one field to Message 2 and four fields to Message 3, and compute the AK in a novel way:

```
Message 1:
SS → BS: Cert(Manufacturer(SS))
Message 2:
SS → BS: SS-Random | Cert(SS) |
         Capabilities | SAID
Message 3:
BS → SS: SS-Random | BS-Random | RSA-
         Encrypt(PubKey(SS), pre-AK) |
         Lifetime | SeqNo | SAIDList |
         Cert(BS) | Sig(BS)
```

Table 3 defines the terms used in these messages.

Computing the AK using the new scheme *AK* = HMAC-SHA1(*pre-AK*, *SS-Random* | *BS-Random* | *SS-MAC-Addr* | *BS-MAC-Addr* | 160) creates a new AK equal in bit length to the existing AK. Including *SS-Random* and *BS-Random* in the result guarantees both *SS* and *BS* that the resulting AK is fresh, regardless of peer contributions. Including the MAC addresses in the AK derivation binds the key to this set of peers, and including the bit length protects the construction against extension attacks. Using HMAC-SHA1 instead of vanilla SHA1 for key derivation future-proofs the design against attacks on the hash function. This protocol corrects the security mistakes in the authorization scheme.

To correct the key management mistakes, we suggest adding the instance identifier *SS-Random* | *BS-Random* and expanding *SeqNo* to at least 12 bits, adding the proviso that it never wraps:

```
[Message 1:
BS → SS: SS-Random | BS-Random |
         SeqNo12 | SAID | HMAC(1)]
Message 2:
SS → BS: SS-Random | BS-Random |
         SeqNo12 | SAID | HMAC(2)
Message 3:
BS → SS: SS-Random | BS-Random |
         SeqNo12 | SAID | OldTEK |
         NewTEK | HMAC(3)
```

If the data SA is for unicasts, we suggest deriving TEK instead of distributing it: *TEK* = HMAC-SHA1(*pre-TEK*, *SS-Random* | *BS-Random* | *SS-MAC-Addr* | *BS-MAC-Addr* | *SeqNo*12 | 160)

We plan to propose these changes to Task group e.

Low-cost roaming

Authentication is an expensive operation, and experience shows that meeting the performance requirements for many applications requires amortizing the authentication cost over connections with several BSs. For voice call migration, for example, the ITU recommends that the time between leaving one BS and reestablishing the con-

IEEE 802.16 progression

The initial IEEE 802.16-2001 standard, published in April 2002, defines a point-to-multipoint fixed wireless access system operating in the 10- to 66-GHz frequency range. The nature of radio propagation at these frequencies means that 802.16 wireless connections are line of sight between a base station and a subscriber station.

IEEE 802.16c, published in January 2003, describes system profiles and conformance criteria. IEEE 802.16a-2003, published in April 2003, enables 2- to 11-GHz operation, and lets near line-of-sight operation cope with the propagation environment at these frequencies. It also introduces a mesh mode, enabling nodes to forward traffic to adjacent nodes.

The IEEE 802.16 Working Group has two ongoing activities:

- A revision standard, IEEE 802.16d, merging the base IEEE 802.16-2001 standard with approved amendments a and c to form what is likely to be called IEEE 802.16-2004. IEEE 802.16d also corrects errors in the base standard and enables compatibility with IEEE 802.16e.
- Amendment IEEE 802.16e adds mobility support to IEEE 802.16. The resulting standard will define a system that simultaneously supports both fixed and mobile services.

Completion of IEEE 802.16d is expected later this year; completion of IEEE 802.16e is anticipated in 2005.

text at another BS never exceed 30 milliseconds. On the other hand, a protocol design security assumption is that BSs share neither AKs nor TEKs; otherwise, compromising one BS compromises the SS at all BSs it visits during the same session. Solving this problem requires extensive back-end development, which is outside the scope of IEEE 802.16e. Algorithmic development to address this problem is ongoing in the IETF and IRTF.

Future work

IEEE 802.16 has the potential to achieve great market success, but its security issues are likely to brake its adoption. We plan to work within IEEE 802.16e and the IETF to address the security deficiencies identified here. □

References

1. *IEEE Std. 802.16-2001, IEEE Standard for Local and Metropolitan Area Networks*, part 16, "Air Interface for Fixed Broadband Wireless Access Systems," IEEE Press, 2001.
2. W.A. Arbaugh, N. Shankar, and Y.C. Wan, "Your 802.11 Network Has No Clothes," Mar. 2001; www.cs.umd.edu/~waa/wireless.pdf.
3. N. Borisov, I. Goldberg, and D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11," Feb. 2001; www.isaac.cs.berkeley.edu/wep-faq.html.
4. S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Schedule Algorithm of RC4," *Proc. 8th Ann. Workshop Selected Areas of Cryptography*, Springer-Verlag, 2001, pp. 1-24.
5. J. Walker, "Unsafe at Any Key Size," Oct. 2000; http://grouper.ieee.org/groups/11/Documents/DocumentHolder/0-362.zip.
6. *IEEE Std 802.11-1999, IEEE Standard for Local and Metropolitan Area Networks*, part 11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," IEEE Press, 1999.
7. *FIPS PUB 180-1, Secure Hash Standard*, Nat'l Inst. of Standards and Technology, Apr. 1995; http://csrc.nist.gov/CryptoToolkit/tkhash.html.
8. *FIPS PUB 46-3, Data Encryption Standard (DES)*, Nat'l Inst. of Standards and Technology, Oct. 1999; http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.
9. RSA Cryptography Standard, *RSA Public Key Cryptography Standard #1 v. 2.0*, RSA Laboratories, Oct. 1998, www.rsasecurity.com/rsalabs/pkcs/pkcs-1/.
10. M. Bellar et al., "A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation," *Proc. 38th IEEE Symp. Foundations of Computer Science*, IEEE CS Press, 1997, pp. 394-403.
11. *FIPS PUB 197, Advanced Encryption Standard (AES)*, Nat'l Inst. of Standards and Technology, Nov. 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
12. L. Blunk and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)," RFC 3748, Internet Eng. Task Force, 2004.
13. D. Whiting, R. Housley, and N. Ferguson, "Counter with CBC-MAC (CCM)," RFC 3610, Internet Eng. Task Force, Sept. 2003.
14. R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm," RFC 3394, Internet Eng. Task Force, Sept. 2002.
15. B. Aboba, D. Stanley, and J. Walker, "IEEE 802.11 EAP Requirements," Internet draft, work in progress, Jan. 2004.

David Johnston *is senior staff engineer with Intel Corporation's Wireless Protocols Lab in Hillsboro, Oregon. His current research centers on novel implementation architectures for wireless medium access controllers. He is actively involved in the development of the IEEE 802.11 and 802.16 standards and recently helped found and chaired the IEEE 802.21 working group. Johnston has a BSc in computer science from Manchester University, UK. Contact him at david.johnston@ieee.org.*

Jesse Walker *is a principal engineer with Intel Corporation's Mobile Networking Lab in Hillsboro, Oregon. His research focuses on network security protocols. Walker has a PhD in mathematics from the University of Texas at Austin. He is the editor for IEEE 802.11i and is active in many other IEEE 802 and IETF security standards. Contact him at jesse.walker@intel.com.*