

White Paper

Wireless Security

Introduction

Security within a SkyPilot Carrier-Class Broadband Wireless System achieves four critical objectives: confidentiality, message integrity, node authentication and network authentication. Confidentiality provides privacy by preventing others from accessing network traffic in its cleartext form. Message integrity ensures that network traffic is delivered unaltered to the intended recipient. Node authentication provides trust between two SkyPilot nodes on the network. Without node authentication, confidentiality and message integrity are difficult to achieve. Node authentication also ensures that only authorized nodes are permitted to join the network. Network authentication ensures nodes join only trusted networks.

The two primary technologies required to implement these security provisions are encryption, which uses keys, and authentication handshake protocols, which may employ digital certificates. SkyPilot supports the Advanced Encryption Standard (AES or Rijndael) to convert cleartext into ciphertext (and vice versa) using a secret session key derived from the authentication handshake negotiation. MD5 is used for generating one-way hashes or message digests.

Managing these security provisions involves the use of a shared network key, which is installed during the manufacture of the SkyPilot nodes. The SkyPilot Command Line Interface (CLI) provides the commands required to configure and verify these shared network keys. Additional authentication is provided through the use of digital certificates signed by SkyPilot as the certificate authority (CA). A certificate is a digital statement of information that links a public key with its owner. To ensure its validity, the certificate is signed by a trusted authority designated as the certificate authority. A certificate chain can optionally exist with a hierarchy of certificate authorities to a root CA. The CA public key is contained within the certificate and installed on the nodes prior to deployment.

This document describes the use of shared static network keys in securing a SkyPilot Carrier-Class Broadband Wireless System. This self-contained model, which requires identification (ID) certificates, a root public key and a rigorous authentication handshake protocol, is both necessary and sufficient to secure a SkyPilot network. The material is organized into two main sections covering Key and Certificate Generation and the Network/Node Authentication. The final section of this document describes security standards employed in the authentication and encryption of end-user data across 802.11 b/g/a radios when using the SkyPilot family of SkyExtender DualBand and TriBand products.

Key and Certificate Generation

Cryptography requires one or more keys of various lengths to enable open standards, like AES, to be used securely. One of the most secure methods of key management is for each node to generate its own public/private key pair. This approach ensures that the private key is known only by its owner because the only credential that needs to be passed from any node is its public key for authentication by a CA. The protection of private keys is paramount to the success of any security model. In the SkyPilot system, public/private keys are generated by SkyPilot during manufacturing and stored on each device. The private keys are stored in non-volatile RAM and obfuscated to prevent any pattern detection.

Identification Certificates

All SkyPilot nodes have an identification (ID) certificate installed during their manufacture. This certificate is part of a certificate chain that has been signed all the way up to the root certificate authority (CA). The root CA is SkyPilot. The ID certificate proves that the node indeed owns a unique MAC address and aids in node authentication. Along with the ID certificate, the root CA's public key is also installed.

White Paper

The data contained within the ID certificate is as follows:

- Type of certificate
- Initial valid date of certificate
- Expiration date of certificate
- Identity of certificate owner
- Public key of certificate owner
- Identity of signer
- Signature of signer

Shared Network Keys

The most straightforward security model employs a static shared network key. Under this model, all nodes within the same network share the same network key. The shared network key is a static key optionally configured by the network operator prior to deploying each node. The network key is then used by each node to prove (via an authentication handshake described in the next section) that it is part of a particular network. Only nodes that possess the identical network key are permitted to join the network. The network key becomes a basis, therefore, of both node and network authentication.

The shared network key model also allows network operators to segment a network. Each network is distinguished with its own shared network key. Multiple keys, therefore, enable the deployment of multiple networks or network segments. The keys are kept secret, of course, to prevent unauthorized nodes in the vicinity from joining the “wrong” network.

To help ensure network integrity, the shared network key can only be modified through the CLI of each SkyPilot device. In the rare event that the shared network key was to become compromised, a new key would need to be configured on all nodes.

Network/Node Authentication

The authentication handshake process uses a rather rigorous algorithm for ensuring that nodes are valid members of a network. When a new node is deployed, it establishes links with other devices around it. With basic communications enabled, the node initiates the handshake negotiation (details of which are provided later in this section). The node uses a single task model that listens on a message queue for inter-node and inter-task messages; four separate inter-node messages are employed during the authentication handshake:

1. “Hello” to initiate the handshake negotiation
2. “Challenge” to request knowledge of a shared secret
3. “Response Challenge” to respond to the challenge with the shared secret
4. “Response” to acknowledge success and complete the handshake negotiation.

The node also manages the secret session keys created between nodes. Once the handshake negotiation has completed successfully and the secret session key is established, the secret session key is used in the block cipher to encrypt and decrypt all inter-node traffic.

The Authentication Handshake

The Strong Authentication mode described here employs a two-way, seven-step handshake algorithm to satisfy two requirements of securing a network with shared static keys. The first is to enable two nodes to authenticate each other, and thereby, their eligibility to join the network. The second is to establish the shared secret session key that is used by the driver in the block cipher.

White Paper

The seven-step authentication handshake (depicted in Figure 1) proceeds as follows—unless and until it is terminated by the unsuccessful completion of any single step:

Step 1: After basic link connectivity has been established, handshake negotiation is then initiated in the node with the smaller MAC address (Node A in this example). The node with the larger MAC address (Node B in this example) waits and listens for the initial message. To initiate the negotiation, Node A generates a “Hello” message that contains its ID certificate, and sends this message to Node B.

Step 2: Upon receiving the “Hello”, Node B verifies that the ID certificate in the message has been signed by the correct CA using its root public key. This step provides assurance to Node B that Node A possesses a valid certificate. If node B successfully verifies the ID certificate, it then computes a random number. This random number is one of the elements used later to generate the secret session key. If Node B does not receive the “Hello” message within 60 seconds or if the verification effort fails, then Node B determines that authentication has failed for this link. The link is then adjusted to an authentication failed state accordingly.

Step 3: Node B encrypts the random number it generated in Step 2 with the shared network key. It then encrypts the ciphertext with Node A’s public key. The purpose of this double encryption is to verify that Node A has the same shared network key as Node B. It also ensures that only Node A can decrypt the message. The final ciphertext is then combined with Node B’s ID certificate and placed in a “Challenge” message sent to node A. Node B’s ID certificate is used to prove its authenticity to Node A.

Step 4: When Node A receives the “Challenge” message, it verifies Node B’s ID certificate using its root public key. Node A then decrypts Node B’s random number first using its private key and then using the shared network key. It then computes its own random number, and uses the two random numbers (Node B’s and its own) to compute a shared secret session key.

Step 5: Node A then encrypts its own random number using the shared network key. It then encrypts this ciphertext along with Node B’s decrypted random number using Node B’s public key. As before, this double encryption proves to Node B that Node A has the same shared network key. It also ensures that only Node B can decrypt the message. The resulting ciphertext is placed in a “Response Challenge” message and sent to Node B.

Step 6: When Node B receives the “Response Challenge” message, it attempts to decrypt the random numbers. If Node B can decrypt the random numbers, it then verifies that the decrypted random number from the received message matches the one it generated in Step 2. If the random numbers match, then Node A will have proven to Node B that they both possess the same shared network key. Node B then generates the secret session key using the two random numbers. Node B encrypts Node A’s decrypted random number using Node A’s public key. Node B then sends Node A a “Response” message containing the ciphertext.

Step 7: When Node A receives the “Response” message, it decrypts the random numbers using its private key. It then compares the decrypted random number with the one it generated in Step 4. If the numbers match, then Node B will have proven to Node A that they both now share the same secret session key. Once the handshake completes successfully, the node delivers the computed secret session key to its driver; this key is then used to encrypt/decrypt all traffic traversing the link. The node also records that link authentication has completed successfully. The provisioning agent task then commences with appropriate provisioning and configuration tasks.

White Paper

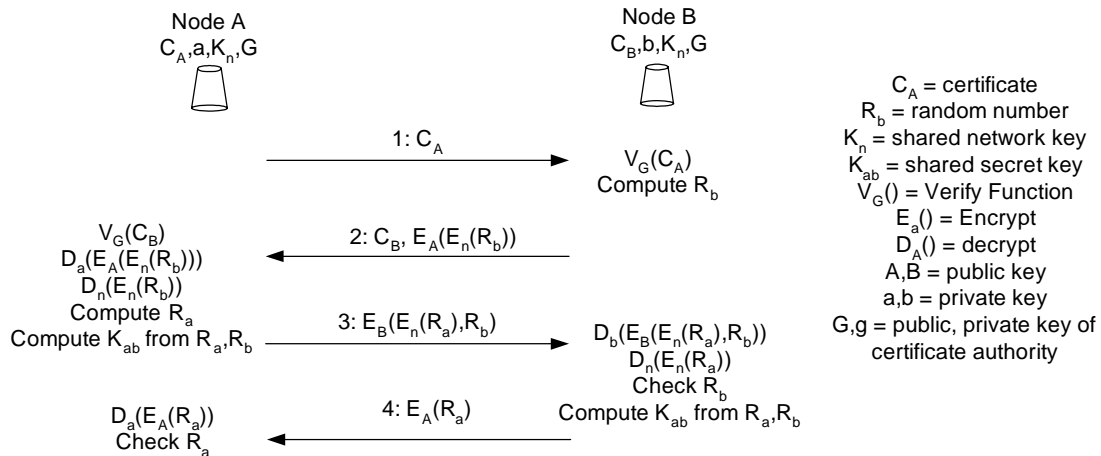


Figure 1: Strong Authentication Handshake

End-user Authentication and Encryption

The SkyExtender DualBand and TriBand products provide several different authentication methods for end-user devices across the 802.11b/g/a access radios. All options are industry standard security implementations that may or not be supported depending on the specific end-user device and software. The following standards are supported:

- WEP
- 802.1x
- WPA
- WPA2

These standards are described in both chronological order of standardization as well as least to most sophisticated in their security methods. When configuring SkyPilot access points, multiple SSIDs may be created on the same radio, with each SSID employing its own security mechanism. It is possible to present two SSIDs in which one uses WEP while the other uses WPA2.

Wired Equivalent Privacy (WEP)

WEP is part of the IEEE 802.11 standard ratified in September 1999. WEP uses the stream cipher RC4 for confidentiality and the CRC-32 checksum for integrity. WEP relies on a pre-shared key and cannot provide for per-user authentication.

Standard 64-bit WEP uses a 40 bit key, to which a 24-bit initialization vector is concatenated to form the RC4 traffic key. At the time that the original WEP standard was being drafted, US Government export restrictions on cryptographic technology limited the key size. Once the restrictions were lifted, all of the major manufacturers eventually implemented an extended 128-bit WEP protocol using a 104-bit key size.

White Paper

Key size is not the major security limitation in WEP. Cracking a longer key requires interception of more packets, but there are active attacks that simulate the necessary traffic. There are other weaknesses in WEP, including the possibility of initialization vector collisions and altered packets, which are not helped at all by a longer key. Despite the weaknesses, WEP provides a level of security that can deter casual snooping. SkyPilot does not recommend using WEP if possible but provides it as an option if end-user devices require it.

IEEE 802.1x

IEEE 802.1x is an IEEE standard for providing login/password-based authentication rather than a shared network key. 802.1x is based on the EAP, Extensible Authentication Protocol (RFC 2284). 802.1x is configured to authenticate hosts which are equipped with supplicant software, denying unauthorized access to the network at the data link layer.

This protocol addresses some of the security vulnerabilities of WEP by performing authentication through a third-party RADIUS server configured to respond to login requests by validating the username and password of each session. This provides for strong mutual authentication but suffers from relying on the same encryption flaws of WEP.

Wi-Fi Protected Access (WPA)

WPA is designed to encrypt data using the RC4 stream cipher, with a 128-bit key and a 48-bit initialization vector. The major improvement in WPA over WEP is the Temporal Key Integrity Protocol (TKIP), which dynamically changes keys as the system is used. This defeats the well-known key recovery attacks on WEP.

In addition to improved encryption, WPA also provides vastly improved payload integrity. The cyclic redundancy check (CRC) used in WEP is inherently insecure; it is possible to alter the payload and update the message CRC without knowing the WEP key. A more secure message authentication code is used in WPA, an algorithm named "Michael". This method includes a frame counter, which prevents replay attacks being executed; this was another weakness in WEP.

By increasing the size of the keys and initialization vectors, reducing the number of packets sent with related keys, and adding a secure message verification system, WPA makes breaking into a wireless network far more difficult. The Michael algorithm was the strongest that WPA designers could come up with that would still work with older network cards; however it is subject to a packet forgery attack. To limit this risk, WPA networks shut down for 60 seconds whenever an attempted attack is detected.

WPA was formulated as an intermediate step towards improved 802.11 security. WPA firmware upgrades have been provided for the vast majority of wireless network interface however some client devices may not have the correct software to make use of this standard.

WPA can best be thought of as an improved encryption technique, making use of either pre-shared key or 802.1x as the authentication mechanism. This allows the operator to employ group or individual authentication as required.

Wi-Fi Protected Access 2 (WPA2 or 802.11i)

IEEE 802.11i, also known as WPA2, is an amendment to the 802.11 standard specifying security mechanisms for wireless networks. The draft standard was ratified on 24 June 2004 with the intention of superseding WEP. WPA2 makes use of the Advanced Encryption Standard (AES) block cipher while WEP and WPA use the RC4 stream cipher. Like WPA, WPA2 offers the choice of either pre-shared key or 802.1x as the authentication mechanism.

White Paper

Combined with 802.1x and the appropriately configured RADIUS server, end-users are authenticated on a per-user basis and all traffic is encrypted using the AES method. This is the most sophisticated combination of authentication and encryption techniques available over 802.11b/g/a networks.

The choice of which security mechanism to employ depends entirely on what is supported in the target customer base. In a deployment in which all clients will use an approved and managed client device such as a specific laptop or client radio, WPA2 might be the most appropriate. But in an open access application in which client devices are not regulated and users are not tracked as closely, WEP or 802.1x could be a more appropriate choice.

Also, the use of 802.1x requires that a RADIUS server be configured and maintained with user names and passwords. This presents a greater burden than employing the use of a pre-shared key.

SkyPilot's recommendation is to make use of the full WPA2 standard whenever possible in order to employ the best possible authentication and encryption techniques available.

Conclusion

The SkyPilot Carrier-Class Broadband Wireless System affords substantial flexibility for securing the network with strong encryption and authentication. Network operators have a choice of security models. The most straightforward model is completely self-contained, requiring no third-party authentication servers or certificate authorities. Various end-user security models permit flexibility in choosing security provisions, allowing the SkyPilot network to be integrated into an existing network security infrastructure. These security provisions afford robust protection that is both cost-effective, and relatively easy to implement and maintain.

To learn more about the SkyPilot Carrier-Class Broadband Wireless System, visit SkyPilot at www.skypilot.com or call SkyPilot at 866-SKYPILOT (866-759-7456) or +1 408-764-8000.

SkyPilot Networks, Inc.
2055 Laurelwood Road
Santa Clara, CA 95054-2747
408.764.8000
US Toll Free 866 SKYPILOT
sales@skypilot.com

© 2006 SkyPilot Networks, Inc. All rights reserved. SkyConnector, SkyControl, SkyExtender, SkyGateway, SkyPilot, SkyPilot Networks, SkyProvision, the SkyPilot logo, and other designated trademarks, trade names, logos, and brands are the property of SkyPilot Networks, Inc. or their respective owners. Product specifications are subject to change without notice. This material is provided for informational purposes only; SkyPilot assumes no liability related to its use and expressly disclaims any implied warranties of merchantability or fitness for any particular purpose.