

White Paper

JumpStart for Wireless™ **Providing Easy WLAN Setup While** **Delivering Industry-Leading Security**



Contents

Executive Summary	1
WLAN Threat Scenarios	2
How Popular Security Protocols Fail	3
The JumpStart Alternative	5
JumpStart's Ease of Use	5
JumpStart Security Methods	5
Augmenting the Diffie-Hellman Protocol	5
Implementing JumpStart for Wireless	7
Appendix 1:	
Initial Affinitization between the AP and Station	8
Appendix 2:	
Incremental Addition of a Station	11

JumpStart for Wireless™

Providing Easy WLAN Set-Up While Delivering Industry-Leading Security

Executive Summary

The JumpStart for Wireless security protocol deals with a long-standing difficulty for personal wireless LANs (WLANs): A security setup method that is not easy to use is of little use because people often fail to use it. When people do use it, they can make errors. Either way, security is poor or non-existent. Poorly implemented security is often worse than no security, as it gives users a false sense of protection.

JumpStart solves these problems by providing a system that is both easy to use and as secure as today's standard protocols and encryption methods can make it. By leveraging the protocol underlying today's secure e-commerce transactions, the Advanced Encryption Standard (AES) and other industry-standard protocols, JumpStart builds upon an established and highly secure foundation.

Thanks to these proven protocols and algorithms, achieving a high level of security is not particularly difficult. However, designing a secure system that is also easy to use is challenging.

The innovative aspect of JumpStart thus lies in its ease of use. With simple observation of LEDs, users quickly establish and verify secure wireless connections. The creation of a user-friendly password then allows for simple access to the newly created JumpStart network. JumpStart performs the difficult parts of the setup behind the scenes without the need for direct user control. With these secure, easy-to-configure protocols, users greatly improve their ability to avoid threats such as identity theft and the compromise of any personal or business information.

For OEMs, easier security setup helps ensure that users make fewer customer-support calls and do not return equipment after becoming frustrated with a difficult setup process. Better/easier security can also be a competitive selling point (supported through use of the JumpStart logo and User Interface). From a broader perspective, maintaining high security is vital for preserving the credibility of wireless networks. JumpStart is straightforward for OEMs to implement, and Atheros is making the protocol freely available for use on any wireless client.

To put WLAN security issues in context, this technology backgrounder begins with a description of security threats and alternative protocols. Then JumpStart's capabilities are described, including its simple user interface and sophisticated algorithms that meet the security challenges of wireless networks.

WLAN Threat Scenarios

Security professionals have identified a number of possible threats to personal wireless networks. In the following descriptions of these threats, note that the term “access point” (AP) refers to any device that fulfills the wireless AP function, including home gateways, firewalls or wireless media servers:

- **Rogue AP**—A rogue AP tries to capture a station. The attacker AP lures the station into an association protocol. At the end of this procedure, the station affinitizes with the attacking AP instead of the intended one. The attacker benefits when the station sends or receives data.
- **Rogue station**—A rogue station affinitizing itself with an AP. The attacker benefits by becoming a participant in the wireless network and thus gaining the ability to send and receive data.
- **Man-in-the-middle**—The attacker logically inserts itself between two legitimate endpoints and actively encrypts and decrypts data for the endpoints. The attacker benefits by capturing a copy of the transmitted data, perhaps including the Pair-wise Master Key (PMK) used for encryption. The attacker can also modify or insert data in transmissions to one party in the name of the other party.
- **Passive monitoring**—The attacker monitors the communication among the stations and the AP in real time. The attacker benefits by capturing data.
- **Offline dictionary**—An eavesdropper records a session and later analyzes the recorded data to determine the PMK. The attacker uses the PMK to decrypt future sessions in real time. The attacker benefits by capturing data.
- **Brute force**—The attacker determines the session key by trying random sequences until finding one that works. The attacker benefits by capturing and sending data in real time.
- **Replay**—An attacker records and later plays back a session over the wireless network. The attacker benefits by gaining access to the network and thus gaining the ability to send and capture data.
- **Forgery attack**—An attacker deliberately tampers with the data exchanged between the intended endpoints. The attacker benefits when the legitimate endpoints receive data that is different from what was originally intended (similar to the man-in-the-middle results.)
- **Radio denial of service (DoS)**—An attacker overloads an AP in various ways so that the AP is unable to serve legitimate users. The attacker does not directly benefit but creates a nuisance.
- **Blind forwarding**—An attacker simply forwards frames without modifying them. As in a DoS attack, the blind-forwarding attacker does not directly benefit.

How Popular Security Protocols Fail

Security protocols must be judged by their ability to protect networks against the identified threats. Note that protocols are generally helpless to prevent the last two threats described above—denial of service and blind forwarding—but protocols should address the other threats to achieve a high level of security.

In real-world situations, achieving this security depends to some extent on the ease with which individuals can apply the protocol. Security protocols that are difficult for users to set up tend to fail because they are either setup incorrectly or not used at all.

Similarly, business users often fail to use protocols that require too much effort to roll out to numerous stations. Security protocols must therefore promote scalability and ease of expansion. A protocol is highly undesirable if it works well for a small network but degrades as the network expands.

With these criteria in mind, it is useful to evaluate today's security setup protocols for WLANs. [Table 1](#) summarizes the evaluations.

Table 1. Characteristics of Popular Security Protocols

Protocol	Ease of Use	Security	Scalability
In-band	Pass	Fail	Pass
Out-of-band	Fail	Pass	Fail
Answer to predefined questions	Pass	Fail	Fail
Attachment to existing computer	Fail	Pass	Fail
Digital Certificates	Fail	Pass	Fail
WPA Enterprise	Pass	Pass for enterprise Fail for home network	Fail
Pre-burned keys	Pass	Fail	Pass

The following descriptions generally use generic protocol names, but bear in mind that these protocols underlie all of today's WLAN security setup procedures:

- **In-band**—This protocol fails to provide adequate security because it exchanges encryption keys via the same communication channel that it is attempting to protect. An attacker can observe the exchange of encryption keys and use this information to decrypt subsequent communications.
- **Out-of-band**—This protocol exchanges encryption keys using a communication channel other than the network it attempts to protect (thus operating out of the regular communication band). To support this protocol, the equipment manufacturer usually associates a unique key with each device and provides a printed multi-digit number to the user. During device initialization or software/firmware upgrades, the user must enter the number directly into the device. While this protocol can be secure, it fails the ease-of-use and scalability requirements.
- **Answers to predefined questions**—This protocol creates an encryption key based on a user's answers to a few simple questions. The protocol is thus unsecure because encryption strength depends upon the key's randomness.
- **Attachment of wireless device to existing computers**—This protocol requires that the wireless device be attached to a computer by wire (USB, for example), enabling the computer to communicate with the device to create the encryption key. While fairly secure, this protocol fails the ease-of-use and scalability requirements.
- **Digital certificates**—This protocol requires that at least one party have a valid digital certificate, which is used along with its corresponding private key to exchange the encryption key. Unfortunately, certificate validation is complex and requires a trustworthy root as well as access to a fresh certificate revocation list. For these and other reasons, a protocol that uses digital certificates is quite secure but fails the ease-of-use and scalability requirements.
- **WPA Enterprise**—This protocol requires users to authenticate with their network security infrastructure before they can join the wireless network. This protocol is fairly secure and relatively easy to use, but it works only for enterprises. In home networks, WPA Enterprise uses pre-shared secret keys (PSKs), which are subject to dictionary attacks and generally unsecure.
- **Pre-burned keys**—This protocol uses the same key for all APs produced by a given vendor. Though the key can be random and of high quality, security decreases as the number of units increase because all deployed units are likely to use the same key to avoid the cost of matching pairs of devices.

Few WLAN users know which of these protocols they use and thus have no idea whether their security is adequate. JumpStart changes that situation by branding the security setup process. Users of JumpStart-compliant products know that they are getting specific security protocols that provide the necessary network protection.

The JumpStart Alternative

JumpStart for Wireless avoids the weaknesses of other protocols with easy-to-use, scalable procedures that protect against all of the identified threats (aside from the denial-of-service and blind forwarding threats that protocols cannot prevent). At the same time, JumpStart is easy to use.

JumpStart's Ease of Use

Before looking closely at JumpStart's security methods, consider the process from the user's point of view. The following sequence flows directly from the device installation process on the user's computer and asks the user to make choices with mouse clicks:

1. JumpStart asks whether the user wants to create a new wireless network or connect to an existing wireless network.
2. When creating a new wireless network, JumpStart asks the user to look at the target device to confirm that an LED is blinking just as shown on the computer screen.
3. JumpStart asks the user to type in a password which will be used to securely associate devices after the network is established.

With these few mouse clicks and a password, users complete their part of the JumpStart process. The JumpStart protocol handles all the other details of generating keys and secure handshaking between the wirelessly connected devices.

JumpStart Security Methods

The primary protocol used in JumpStart is the Diffie-Hellman (DH) key exchange protocol. Developed in 1976, this proven protocol underlies all secure Web-based e-commerce transactions, including Secure Sockets Layer (SSL), Secure Shell (SSH) and Internet Protocol Security (IPSec).

Additionally, JumpStart uses high-quality random values to negotiate the key material for the generation of the WPA Pairwise Master Key (PMK), the Key Encryption Key (KEK), and all message integrity check (MIC) keys. JumpStart derives all key material from a minimum of 1024 bits of securely-chosen data, thereby making all such keys highly resistant to attack.

JumpStart also uses the Advanced Encryption Standard (AES) and the Secure Hash Algorithm (SHA-1). Both government and private institutions recognize these protocols as strong algorithms for protecting data. Moreover, JumpStart uses sequence numbers and random values to thwart replay attacks.

Augmenting the Diffie-Hellman Protocol

For wireless networks, the Diffie-Hellman protocol alone does not offer complete security because it is subject to man-in-the-middle attacks. JumpStart guards against this threat via four defense mechanisms: out-of-band authentication using LEDs, protocol completion within a limited time window, the addition of a password and the serialization of protocol frames.

In the first of these defense mechanisms, the AP provides LED signals that indicate the AP's state in the setup process. If the user does not see the expected LED signals on the target AP, then it is clear that a rogue AP is participating in the setup process, and the process is terminated.

JumpStart for Wireless requires that an AP display two states using one or more LEDs. For example, the AP could have one LED that flashes a slow steady blink to convey "ready state" and another LED that provides a distinctively identifiable blink pattern, three rapid blinks followed by a pause, and then repeat.

- State 1—Slow blink
- State 2—Three rapid blinks followed by a pause

The second defense mechanism for augmenting Diffie-Hellman is the use of a limited time window. The user must complete each protocol sequence within this preset time, leaving an attacker a tiny interval for mounting a man-in-the-middle attack.

The third defense mechanism is the addition of a password. JumpStart authenticates some Diffie-Hellman key negotiations using a combination of the modified password and a random value as the authentication key. For this task, JumpStart uses the Secure Hash Algorithm (SHA-1) and the Message Authentication Code standard as specified in IETF RFC 2104.

After initial setup of the first station, an authenticated user must authorize the setup of all other stations. For stations that have a means to enter characters, JumpStart requires the user to enter the password. An attacker who does not know the password cannot mount a man-in-the-middle attack.

In the fourth defense mechanism, the AP enters a serialized mode for request messages after receiving a request message from a station to engage in the JumpStart protocol. In this mode, the AP accepts no more request messages until the protocol is completed. This serialization is important for preventing a rogue station from attacking the AP by hijacking a session from a legitimate station.

Finally, JumpStart makes use of the fact that wireless is a broadcast-oriented medium to detect a man-in-the-middle attack. During critical points in the protocol, the AP operates in promiscuous mode, allowing it to sense whether any other endpoints are attempting to negotiate JumpStart at the same time. If so, it avoids the possible attack by aborting the current operation.

Table 2 summarizes JumpStart's defense against each of the threat scenarios.

Table 2. JumpStart's Defenses Against Security Threats

Threat	Defense
Rogue AP	User verifies LED states during initial setup
Man-in-the-Middle	Require authenticated key-exchange protocol Serialize protocol frames and allow only one session Abort protocol if not completed within a pre-set time
Rogue Station	Require authenticated key-exchange protocol User verifies LED states during initial setup
Passive Monitoring	Use strong encryption and hashing algorithms
Offline Dictionary	Use high-quality encryption keys
Brute Force	
Forgery	Use strong encryption and hashing algorithms
Replay	Use high-quality encryption keys Use randomly generated tokens and replay numbers

Using the combination of techniques described so far, JumpStart for Wireless implements specific procedures such as the initial affinitization between the AP and a station or the incremental addition of a station. These two procedures are described in detail in “[Appendix 1: Initial Affinitization between the AP and Station](#)” on page 8 and “[Appendix 2: Incremental Addition of a Station](#)” on page 11.

Implementing JumpStart for Wireless

For OEMs, implementing JumpStart is straightforward. The Atheros software includes a set of layer-3 protocols and an application each for the AP and the station. No changes are needed to the network driver interface specification (NDIS) or any part of the MAC layer, so use of JumpStart reduces the risk of interoperability problems (such as might affect WiFi certification). In fact, Atheros has built the process to be compatible with legacy devices. JumpStart supports WEP, WPA, and WPA2 networks.

When JumpStart runs, it builds a security profile that is given to other management tools. The password collected during JumpStart's initial setup procedure is available for signing future Diffie-Hellman sessions. The use of Diffie-Hellman and other protocols eliminates the need for unique manufacturing states in either APs or stations. JumpStart provides a fully wireless solution, with no added hardware cost.

JumpStart for Wireless is an easy solution for OEMs and end users. With a few mouse clicks and a simple password, users get a level of wireless security equal to the best security available, and far better than most. JumpStart has launched WLAN security into a new era of simplicity and effectiveness.

Appendix 1: Initial Affinitization between the AP and Station

JumpStart's initial affinitization procedure assumes no shared states or secret between the AP and the station. When the user turns on the AP, its LED is in state 1 (ready to start), and the station is running the JumpStart for Wireless software (e.g., from a CD). The procedure runs as follows:

1. The station's JumpStart software displays a screen asking the user to select a task: create a new network or join an existing network. (see [Figure 1](#)). During the initial configuration process, users select create a new network on the interface.



Figure 1. User Selects Networking Task

2. The station detects the AP.
3. Station to AP: Association request.
This step is a complete 802.11 open authentication and association.
4. AP to Station: Association response.
5. AP to Station: *Capability menu, Diffie-Hellman public parameters, N*.
N is a sequence number that prevents replay attacks.
6. Station to AP: *Capability choice, Diffie-Hellman public value A, N*.
7. From this point the AP serializes all communications to prevent a rogue station from hijacking the session. The AP uses radio entropy to create a good random public value *B*. Before transmitting this public value, the AP can derive the Diffie-Hellman shared secret. The AP uses part of that secret value to produce the message authentication code, indicated in the next step as the value *MIC*. All messages are therefore authenticated using the Diffie-Hellman shared secret. If so configured, the AP enters promiscuous mode to prevent a rogue AP from mounting a man-in-the-middle attack by transmitting its own Diffie-Hellman public value.

8. AP to station: *Diffie-Hellman public value B, station's capability choice, N+1, MIC.*
9. The station uses the AP's public value B to calculate the shared secret. The station verifies the integrity of the message using the MIC.
10. Station to AP: *Capability menu from AP, N+1, Nonce, MIC.*
This acknowledgement message proves to the AP that the station knows the derived Diffie-Hellman shared secret. The message is a signed version of the AP's capability menu, which the station received in step 5.
11. If the AP is in promiscuous mode, it exits it, then, upon successful MIC verification from step 10, the AP sets its JumpStart LED to state 2 (*successful completion*).
12. The station's JumpStart software asks the user to confirm that the JumpStart LED is in state 2 ([Figure 2](#)).

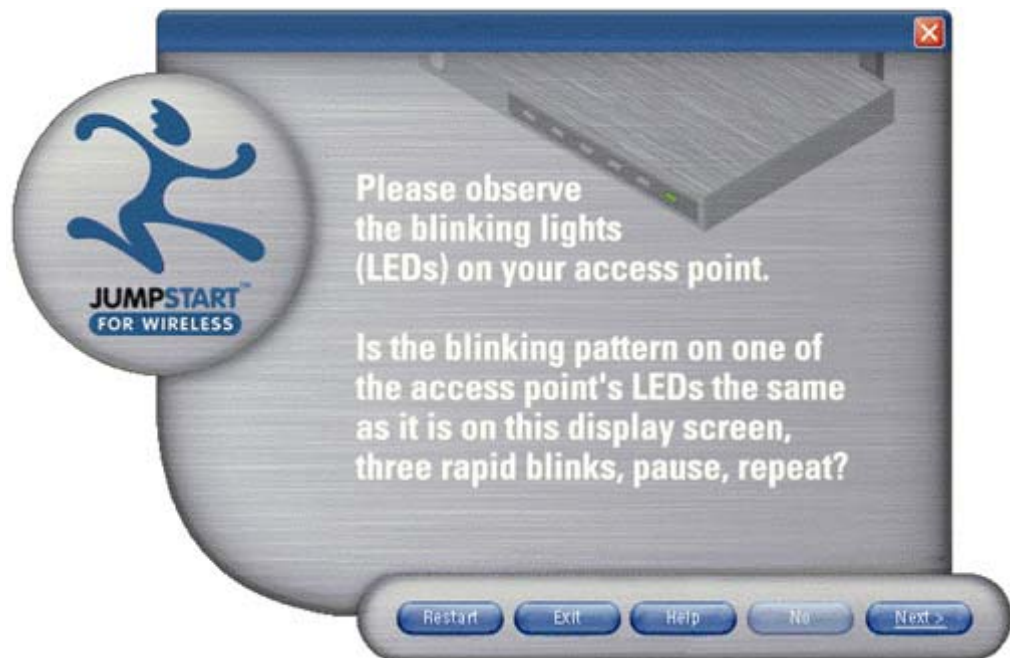


Figure 2. User Confirms Successful Affinitization

- The station's JumpStart software asks the user to enter a password (Figure 3). The software creates a randomized and expanded version of this password.



Figure 3. User is Prompted to Create Password

- Station to AP: Encrypted $\{\text{salted SHA-1}(\text{password}), \text{salt}\}_{\text{AES key}}$
The station derives a symmetric encryption key from the Diffie-Hellman shared secret and uses the AES algorithm to encrypt the message.
- The station and the AP each derive the same 256-bit Pair-wise Master Key (PMK) from the Diffie-Hellman shared secret and permanently store it. The AP also stores the randomized/expanded version of the password.

Table 3 summarizes the threats that this procedure mitigates.

Table 3. Threat Mitigation for Procedure 1

Procedure Step	Threats Mitigated	How Threat is Mitigated
6	Rogue AP	User confirms LED state during setup.
8	Man-in-the-Middle (Rogue Station)	AP serializes communication and will not accept request frames until protocol completion. AP aborts the protocol if not completed within a pre-set time limit.
	Man-in-the-Middle (Rogue AP)	AP enters promiscuous mode and aborts protocol if another station transmits frame.
5-16	Replay	The protocol uses replay numbers to detect replay attacks.
All	Brute Force	The protocol uses high-quality random values and strong encryption keys.
	Forgery	The protocol provides message authentication codes (the MIC).

Appendix 2: Incremental Addition of a Station

After using JumpStart to set up a network, users can run JumpStart again to add stations. The procedure for adding stations does not require JumpStart LEDs, but the user must know the password entered during the initial setup. The procedure runs as follows:

1. The JumpStart software on the station to be added to the network will display a screen which asks the user to select the task to be executed: create a new network or join an existing network. For station additions, users will wish to join an existing network (see [Figure 4.](#))



Figure 4. User Selects Networking Task

2. The station detects the AP.
3. Station to AP: Association request-This step is a complete 802.11 open authentication and association.
4. AP to Station: Association response.
5. AP to Station: *Capability menu, Diffie-Hellman public parameters, Diffie-Hellman public value B, salt, N, MIC*
Where N is a replay number and MIC is a message authentication code that uses the randomized/expanded version of the password to sign the protocol message.

- Station prompts user for the password. The station uses the password along with the randomized value it received in step 5 to compute the MIC. The station authenticates the message it received in step 5 by verifying the signature of the message. At this point the station is communicating with an AP that has proven possession of the randomized/expanded version of the password (see [Figure 5](#)).



Figure 5. User Provides Password Created During Initial Configuration

- Station to AP: *Capability choice, Diffie-Hellman public value A, N, MIC*
Where MIC is a message authentication code that uses the randomized/expanded version of the password to sign the protocol message. The AP is now communicating with a station that has proven possession of the randomized/expanded version of the password. The station can only compute this version if the user enters the correct password in step 6.
- AP and station each compute the Diffie-Hellman shared secret and use it to derive a temporary AES key.
- AP to station: $\{PMK\}_{AES\ key}$
After encrypting the PMK with the temporary AES key, the AP transmits the PMK to the station.

Table 4 summarizes the threats that this procedure mitigates.

Table 4. Threat Mitigation for Procedure 2

Procedure Step	Threats Mitigated	How Threat is Mitigated
4	Rogue AP	Station verifies signature of Diffie-Hellman public parameters and AP's public value.
6	Man-in-the-Middle (Rogue station)	AP verifies signature of station's Diffie-Hellman public value.
4-8	Replay	The protocol uses replay numbers to detect replay attacks.
All	Brute Force	The protocol uses high-quality random values and strong encryption keys.
	Forgery	The protocol provides message authentication codes (the MIC).



**Atheros Communications
Incorporated**
529 Almanor Avenue
Sunnyvale, CA 94086
t: 408/773-5200
f: 408/773-9940
www.atheros.com

Subject to Change without Notice