Technical Brief

Opportunistic Key Caching

Jon Green
Product Management, Aruba Networks

George Voon
Software Engineering, Aruba Networks

May, 2007

Introduction

One of the great wireless revolutions in recent history has been the IEEE 802.11i standard, which paved the way for secure enterprise deployments of wireless LAN technology. As a replacement for the legacy WEP (Wired Equivalent Privacy) encryption specification, 802.11i provides robust security by mandating both strong encryption as well as strong authentication using IEEE 802.1x. However, this greater security did not come without a cost. Each time a wireless station (STA) connects to an access point, it must go through an authentication procedure. This authentication procedure, while absolutely necessary for security, introduces delay in the process of establishing connectivity. The delay is primarily the result of the 802.1x authenticator contacting an external authentication server. For most data applications, the delay is negligible and is barely noticed by end users. For delay-sensitive applications such as voice and video, however, the delay can be significant enough to disrupt communication.

Fortunately, the 802.11i standard provides a method to speed up authentication for a roaming client using a technique known as PMK (Pairwise Master Key) caching. There are two distinct but similar techniques available:

- **PMK Caching** is defined by 802.11i and is a technique available for authentication between a single AP and a station. If a station has authenticated to an AP, roams away from that AP, and comes back, it does not need to perform a full authentication exchange. Only the 802.11i 4-way handshake is performed to establish transient encryption keys.
- **Opportunistic Key Caching (OKC)** is a similar technique, not defined by 802.11i, available for authentication between multiple APs in a network where those APs are under common administrative control. An Aruba deployment with multiple APs under the control of a single controller is one such example. Using OKC, a station roaming to any AP in the network will not have to complete a full authentication exchange, but will instead just perform the 4-way handshake to establish transient encryption keys.

Although OKC is not a part of the 802.11i standard, several wireless vendors have adopted the technique and have achieved interoperability. Most notably, Microsoft has provided support for OKC in the Windows XP and Vista 802.1x supplicant. This document explains the Aruba Networks implementation of Opportunistic Key Caching with a goal of achieving wider interoperability between multiple client devices and an Aruba wireless network.

OKC Operation

The operation of PMK caching is defined in IEEE 802.11i section 8.4.1.2.1:

"A STA roaming within an ESS establishes a new PMKSA by one of three schemes:

- A STA (AP) can retain PMKs for APs (STAs) in the ESS to which it has previously performed a full IEEE 802.1X authentication. If a STA wishes to roam to an AP for which it has cached one or more PMKSAs, it can include one or more PMKIDs in the RSN information element of its (Re)Association Request frame. An AP whose Authenticator has retained the PMK for one or more of the PMKIDs can skip the 802.1X authentication and proceed with the 4-Way Handshake. The AP shall include the PMKID of the selected PMK in Message 1 of the 4-Way Handshake. If none of the PMKIDs of the cached PMKSAs matches any of the supplied PMKIDs, then the Authenticator shall perform another IEEE 802.1X authentication. Similarly, if the STA fails to send a PMKID, the STA and AP must perform a full IEEE 802.1X authentication."[1]

The language in this section specifies that PMK caching is available between stations and APs that have previously performed a full 802.1x authentication – it does not say anything about APs to which a client has not previously authenticated. OKC works by attempting to use PMK caching for any station-AP pair that have not previously communicated.

Aruba Implementation

In an Aruba deployment, both PMK caching and OKC are available:

- PMK caching is **always on** and cannot be disabled for WPA2 ESSIDs
- OKC is a **configurable** option for WPA2 ESSIDs and **enabled** by default

Note that PMK caching and OKC are available within a single controller. At the time of this writing, OKC between multiple Aruba mobility controllers is not possible – stations roaming to a new controller will be required to perform a full 802.1x authentication. Inter-controller OKC is targeted for a future ArubaOS release.

IEEE 802.11i also allows for a station performing PMK caching to send multiple PMKIDs in the association or reassociation frame. At the time of this writing, ArubaOS does not support lists of PMKIDs in association frames, nor is Aruba aware of any client implementations that behave this way. This behavior is subject to future change.

Full 802.1x Authentication

A *standard* association/authentication process, without key caching, is shown in Figure 1 below for reference purposes. Differences from this process will be illustrated in following sections. Note that the "EAP Exchange" line in the figure is a simplification of

[1] IEEE Std. 802.11i-2004, Medium Access Control Security Enhancements to Std. 802.11-1999.

multiple exchanges back and forth between the station and the authentication server – the number and format of these messages varies depending on the EAP type in use.

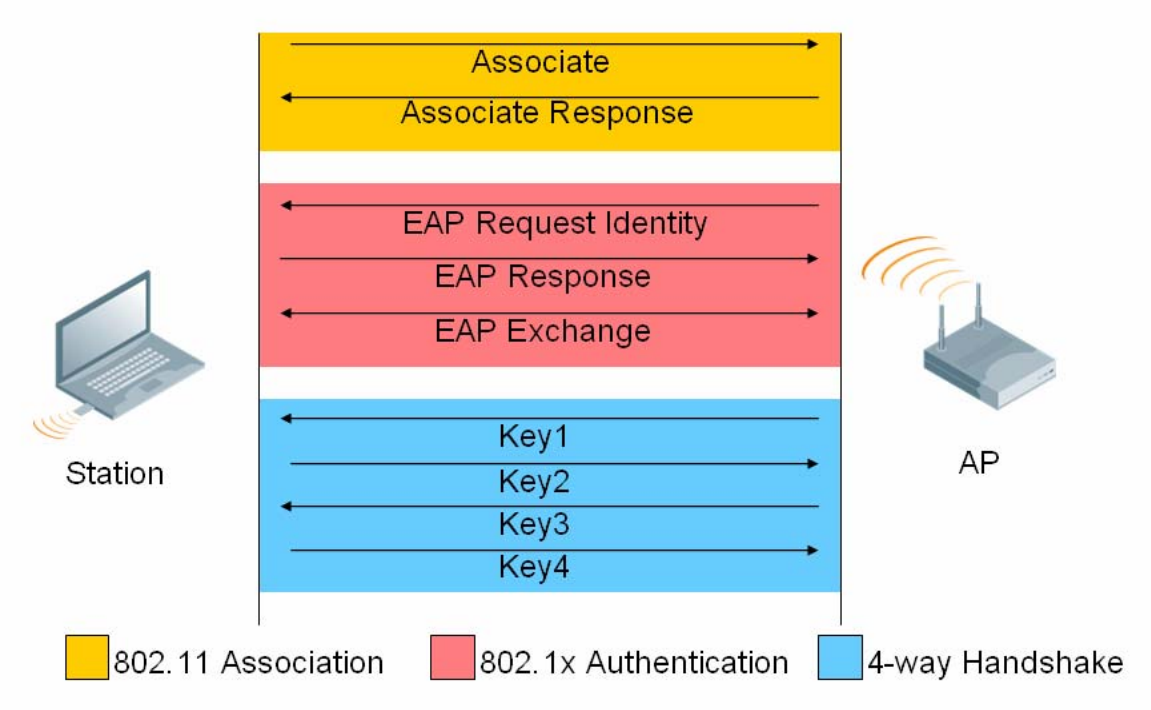

Figure 1 - Standard 802.11i Association Process

PMK Caching Implementation

Clients wishing to use PMK caching should send the PMKID in the association (or re-association) frame, as specified by 802.11i. If the PMKID matches the one cached by the controller for the station's MAC address, the authentication step is skipped and the system proceeds directly to key exchange, as in Figure 2 below. If the PMKID is not sent, or does not match the one cached by the controller, a full authentication process (Figure 1) is performed.

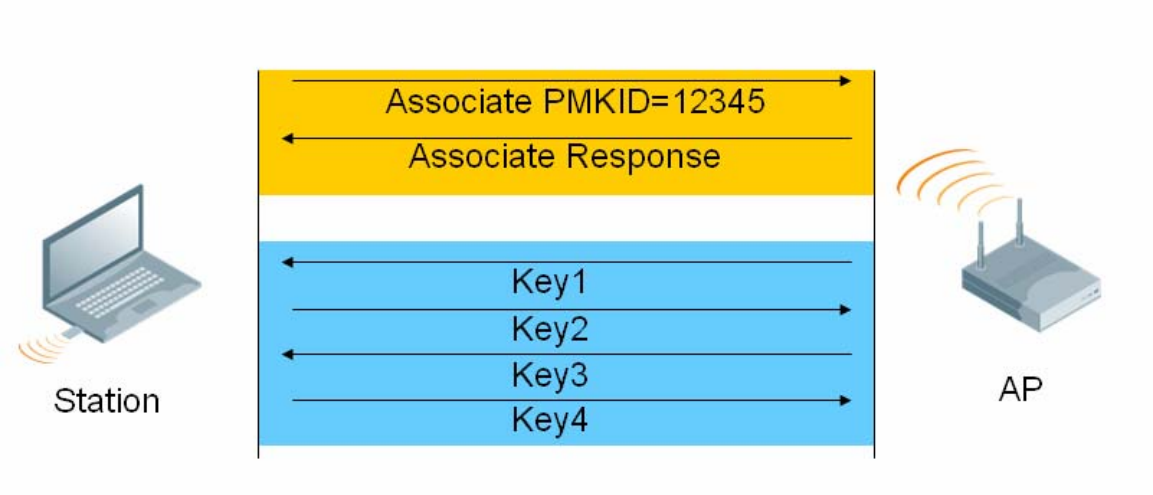

Figure 2 - PMK Caching with Cache Hit

A sample debug output from an Aruba controller performing PMK caching would look like the following:

Initial association with AP

```
Mar 15 16:43:46  station-up                *  00:16:cf:05:7f:a8
00:0b:86:c1:83:06            -    -      wpa2 aes
Mar 15 16:43:46  eap-id-req               <-  00:16:cf:05:7f:a8
00:0b:86:c1:83:06            1    5
```

Re-associating with AP after lost connection

```
Mar 15 16:48:37  station-up                *  00:16:cf:05:7f:a8
00:0b:86:a1:69:f0            -    -      wpa2 aes
Mar 15 16:48:37  station-data-ready       *  00:16:cf:05:7f:a8
00:00:00:00:00:00            2    -
Mar 15 16:48:37  wpa2-key1                <-  00:16:cf:05:7f:a8
00:0b:86:a1:69:f0            -    117
Mar 15 16:48:37  wpa2-key2                ->  00:16:cf:05:7f:a8
00:0b:86:a1:69:f0            -    135
Mar 15 16:48:37  wpa2-key3                <-  00:16:cf:05:7f:a8
00:0b:86:a1:69:f0            -    167
Mar 15 16:48:37  wpa2-key4                ->  00:16:cf:05:7f:a8
00:0b:86:a1:69:f0            -    95
```

Opportunistic Key Caching Implementation

OKC is enabled on an Aruba controller in an 802.1x authentication profile using the following CLI commands:

```
aaa authentication dot1x <profile name>
    opp-key-caching
```

A station wishing to perform OKC should send a PMKID in the association or reassociation frame. At the time of this writing, most client implementations that support

OKC do *not* send the PMKID in their association request, meaning that OKC would normally not be possible. In order to achieve interoperability with these clients, the Aruba controller checks to see if a PMKID has been previously cached for a client's MAC address, and if one is found *always* attempts to perform OKC, regardless of whether the client has sent a PMKID during association. This behavior is illustrated in Figure 3 below. In this illustration, the controller ignores the presence or lack of a PMKID in the association frame, but finds a matching PMK cached for this station. The controller thus skips the 802.1x authentication step and proceeds directly to the 4-way handshake.

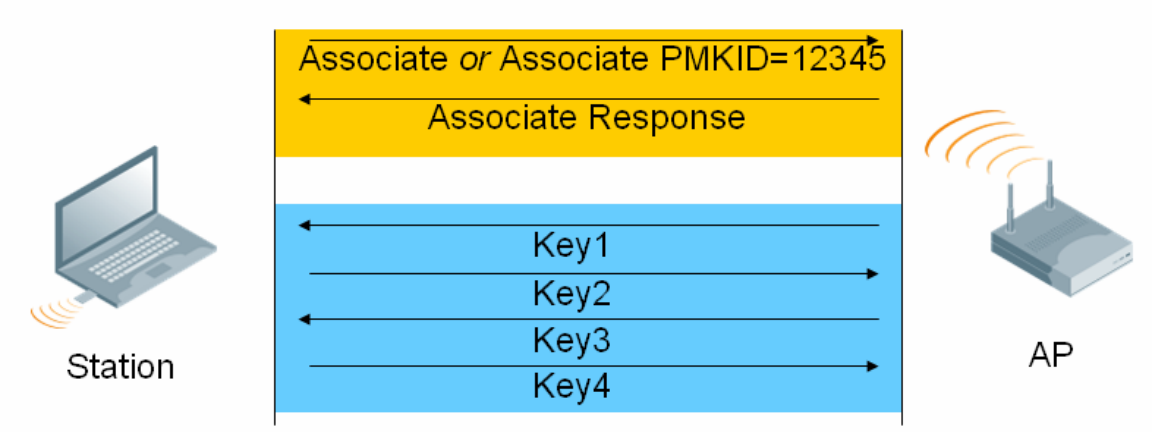

Figure 3 - OKC with Cache Hit

Under some conditions, the controller may have a PMK cached for a given station, but the station has recently been rebooted and no longer has the PMK cached. Under these circumstances, a full 802.1x authentication must be performed. The same situation holds true for a station that does not support OKC – the Aruba controller will attempt OKC with this client, and the client should reject the request by starting 802.1x authentication with an "EAPOL Start" frame. This behavior is illustrated in Figure 4 below.

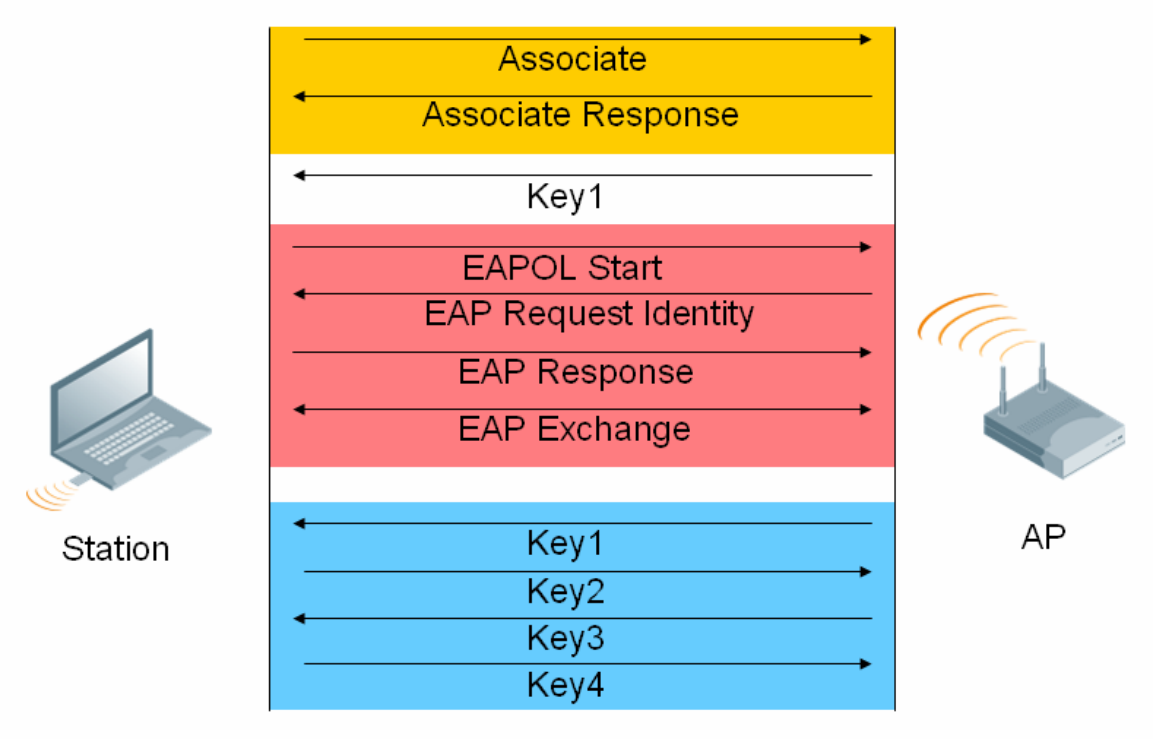

Figure 4 - OKC with Unsupported Client

A sample debug output from an Aruba controller attempting OKC with a client that does not support OKC follows:

Station roams from AP1 to AP2

```
Feb 22 20:10:03  station-up                *  00:16:cf:05:7f:a8
00:0b:86:22:72:90            -    -      wpa2 aes
Feb 22 20:10:03  station-data-ready       *  00:16:cf:05:7f:a8
00:00:00:00:00:00            2    -
Feb 22 20:10:03  wpa2-key1                <-  00:16:cf:05:7f:a8
00:0b:86:22:72:90            -    117
Feb 22 20:10:03  eap-start                ->  00:16:cf:05:7f:a8
00:0b:86:22:72:90            -    -
Feb 22 20:10:03  eap-id-req               <-  00:16:cf:05:7f:a8
00:0b:86:22:72:90            2    5
```

Some client devices do not respond to the Key1 message with an EAPOL Start frame, and instead go into an undefined state from which they must be rebooted. In order to work with these clients, Aruba has implemented a feature to force checking of the PMKID. This feature instructs the Aruba controller to specifically check for OKC support rather than assuming it. When this option is implemented, a station wishing to perform OKC *must* send the PMKID in the associate or reassociate frame – otherwise a full 802.1x authentication will take place. The PMK validation feature is enabled using the following CLI commands:

```
aaa authentication dot1x <profile name>
     opp-key-caching
```

```
validate-pmkid
```

Figure 5 and Figure 6 below illustrate the behavior when this configuration option is used.

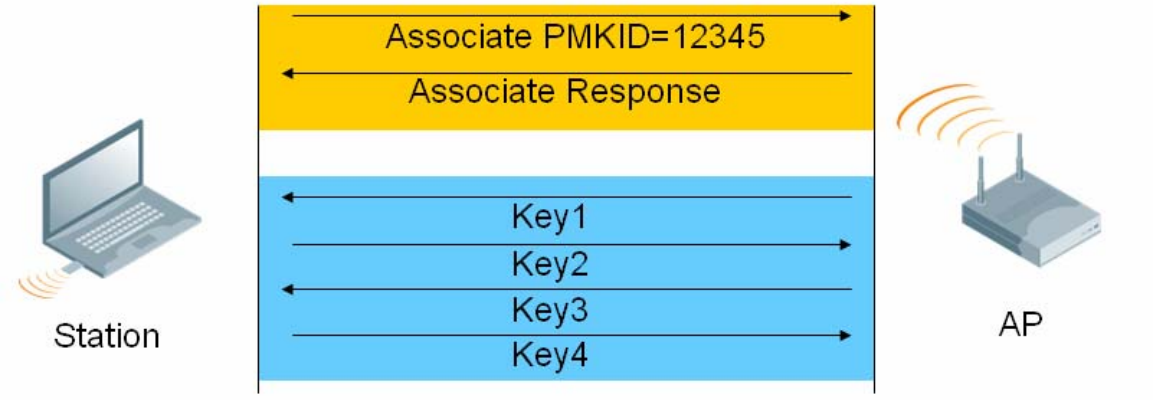

Figure 5 - OKC with Supported Client

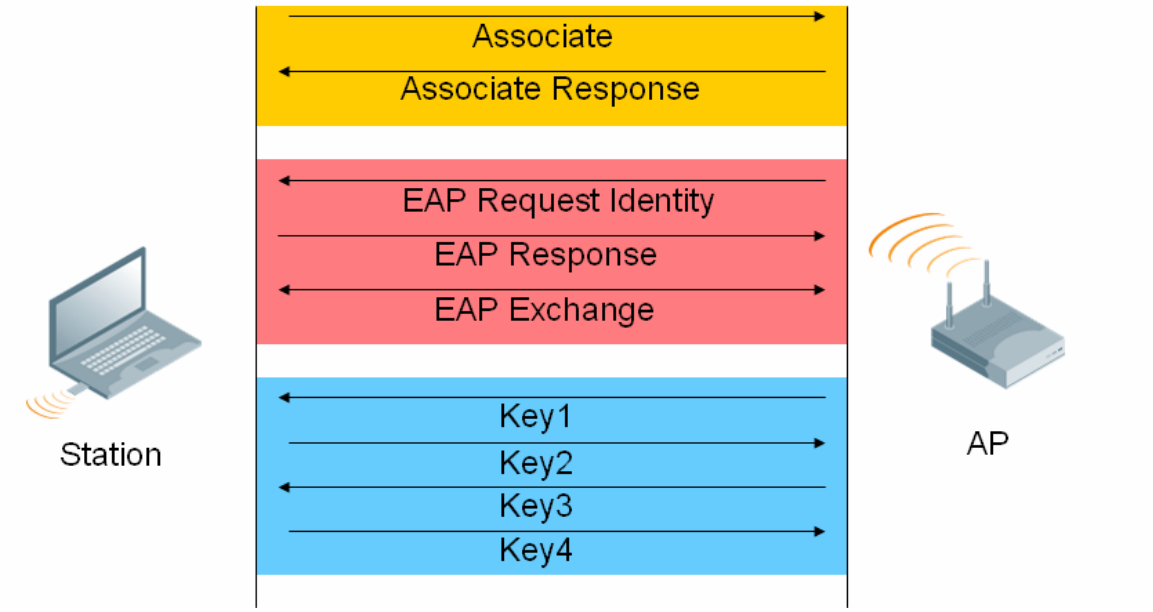

Figure 6 - OKC with validate-PMK and Unsupported Client

Once a client has successfully connected to an AP using OKC, if the client roams away and comes back to that same AP, PMK caching would be used, not OKC. OKC only applies to the first time a station and AP communicate.

Summary

The following table presents a summary of Aruba's OKC behavior for various conditions.

OKC Enabled	Client sends PMKID	Validate-PMKID Enabled	Action
N	N	N/A	Full 802.1x authentication (Figure 1)
N	Y	N/A	If same AP: PMK Caching (Figure 2) If new AP: Full 802.1x authentication (Figure 1)
Y	N	N	Attempt OKC (Figure 3)
Y	N	Y	Full 802.1x authentication (Figure 1)
Y	Y	N	Attempt OKC (Figure 3)
Y	Y	Y	Attempt OKC (Figure 5)

Table 1 - OKC Behavior Summary

About Aruba Networks, Inc.

Aruba Networks is a fast-growing enterprise infrastructure company enabling the Mobile Edge, an evolutionary new network architecture that addresses three top concerns of IT managers—mobility, security, and convergence. The Mobile Edge extends the reach of enterprise networks, providing secure access to information and voice services anywhere a user needs them, enabling new applications, allowing organizations to compete more effectively, and bringing about dramatic economic benefits. To deliver the Mobile Edge, Aruba manufactures and markets a complete line of fixed and modular mobility controllers, wired and wireless access points, and an advanced mobility software suite. Based in Sunnyvale, California, Aruba has operations in the United States, Europe, the Middle East, and Asia Pacific, and employs staff around the world. To learn more, visit Aruba at http://www.arubanetworks.com