taken on the site. However, cookies normally part of Facebook usage are not available to Facebook anymore in Figure 6-2—the outside application requests information from the platform without the help of a user's browser. To fix this, we establish on Facebook a *session key* mapping, as shown in Example 6-11.

EXAMPLE 6-11. A session key mapping

```
get_session: {user_id, application_id} -> session_key
```

The client of the web service simply sends the `session_key` along with every request, to let the web service know on which viewer's behalf the request executes. If the user (or Facebook) disables this application or has never used this application, this fails the security checks, returning an error. Otherwise, the outside application site will carry this session key around with its own user records or in a cookie for that user.

But how does one obtain this key in the first place? The example function `establish_facebook_session` in the *http://fettermansbooks.com* application code (Example 6-5) is a placeholder for this flow. Every application has its own unique "application key" (also called an `api_key`) that begins the application authorization flow (Figure 6-3):

1. The user is redirected to the Facebook login with a known `api_key`.
2. The user enters her credentials on Facebook to authorize the application.
3. The user is redirected to a known application landing site with the session key and user ID.
4. The application is now authorized to make calls to the API endpoint on the user's behalf (until the session expiration or deletion).

To help the user initiate this flow, a link or button could be rendered:

```
<a href="http://www.facebook.com/login.php?api_key=abc123">
```

with that application key (say, "`abc123`"). If a user agrees to authorize this application using the password form *on Facebook* (note that the password is the last piece of data Facebook would export), the user is directed back to this application site with a valid `session_key` and his Facebook user ID. This session key is quite private, so for further verification, each call made by the application is accompanied by a hash generated from a shared secret.

Assuming the developer has stashed his `api_key` and application secret, `establish_facebook_session` can be written quite simply from the flow in Figure 6-3. Though the details between these kinds of handshake systems can vary, it is important that no user can be authorized unless he enters his password in the crucial step on Facebook. Interestingly enough, some early applications simply used this authorization handshake as their own password system, not employing any Facebook data at all.