- [Protect an individual page](#)
- [Protect a group of pages](#)
- [Protect the site](#)

[PmWiki](#) has built-in support for password-protecting various areas of the wiki site. Authors generally want to be able to apply passwords to individual pages or to [wiki groups](#). Wiki Administrators can apply passwords to individual pages, to wiki groups, or to the [entire site](#). Setting an edit password on a [page](#) or [group](#) (or [the entire site](#)) is one of the most common ways to stop [spam](#). As with any access control system, the password protection mechanisms described here are only a small part of overall system and wiki [security](#).

As an author editing pages...

An author will generally set 3 types of passwords:

1. to control who can see a page or group, use `read` passwords
2. to control who can edit a page or group, use `edit` passwords
3. to control who can alter the passwords used to protect a page or group, use `attr` passwords

If required most [page actions](#) can be password protected.

Protect an individual page

To set a password on an individual wiki page, add the [page action](#)

```
?action=attr
```

to the page's URL (address) to access its attributes. Using the form on the attributes page, you can set or clear the `read`, `edit`, or `attr` passwords on the page. In the form you enter the passwords as cleartext; [Pm Wiki](#) encrypts them for you automatically when it stores them.

Additional options:

- Leaving a field blank will leave the attribute unchanged.
- To remove a password from a page (*reverting back* to the group's or site's default), enter

```
clear
```

- To indicate that the page can be edited *even if a group or site password is set*, enter

```
@nopass
```

- To lock a page for everybody but the admin, enter

```
@lock
```

- To assign the site's site-wide passwords to the `read`, `edit`, or `attr` password for the page, enter

```
@_site_edit, @_site_read or @_site_upload
```

Protect a wiki group of pages

To set a password on a [wiki group](#) is slightly more difficult -- you just set the passwords on a [special page?](#) in each group called

[GroupAttributes?](#)

First, you can get to the attributes page for GroupAttributes by entering a URL (address) like

```
http://example.com/pmwiki/pmwiki.php?n=GroupName.GroupAttributes?action=attr
```

Replace example.com with your domain name, and [Group Name?](#) with the name of the group

Then, using the form on the attributes page, you can set or clear the `read`, `edit`, or `attr` passwords for the entire group. In the form you enter the passwords as cleartext; Pm Wiki encrypts them for you automatically.

Additional options:

- To remove a password from a group (*reverting back* to the site's default), enter

clear

- To indicate that the group can be edited *even if a site password is set*, enter

@nopass

- To lock a group for everybody but the admin, enter

@lock

- (Beginning with Ver 2.2.3) To assign the site's site-wide passwords to the `read`, `edit`, or `attr` password for the group, enter

```
@_site_edit, @_site_read or @_site_upload
```

Passwords

Passwords may consist of any combination of characters, except double "quotes" or 'apostrophes'. Passwords with spaces or colons must be entered using quotes, eg "foo bar" or "foo:bar".

Obviously longer is better, and on some systems passwords need to have 4 or more characters.

Multiple passwords

Multiple passwords for a page, group or site are allowed.
Simply enter multiple passwords separated by a space. This allows you to have a read password, a write password, and have the write password allow read/write access. In other words, if the read password is

alpha

and the edit password is

beta

then enter

```
 Set new read password: alpha beta Set new edit password: beta
```

This says that either

alpha

or

beta

can be used to read pages, but only

beta

may edit. Since Pm Wiki checks the passwords you've entered since the browser has been opened, entering a read password that is also a write password allows both reading and writing.

Protect the site

Passwords can be applied to the entire wiki website in *config.php*.
See passwords administration for details.

administrator

As an administrator ...

You can set passwords on pages and groups exactly as described above for authors. You can also:

1. set site-wide passwords for pages and groups that do not have passwords
2. use `attr` passwords to control who is able to set passwords on pages
3. use `upload` passwords to control access to the file [upload](#) capabilities (if uploads are enabled)
4. use an `admin` password to override the passwords set for any individual page or group
5. use [SiteAdmin.AuthList](#) to view the permissions settings for pages that have permissions set.

For more information on password options available to administrators, see [PasswordsAdmin](#).

Which password wins?

In Pm Wiki, page passwords override group passwords, group passwords override the *default* passwords, and the `admin` password overrides all passwords. This gives a great deal of flexibility in controlling access to wiki pages in Pm Wiki.

The [special page?](#) [SiteAdmin.AuthList](#) is a page list of all pages with access permissions set.

Opening access to pages in protected groups/sites

Sometimes we want to "unprotect" pages in a group or site that is otherwise protected. In these cases, the special password

@nopass

is used to indicate that access should be allowed to a page without requiring a password.

For example, suppose Main.GroupAttributes has an edit password set, thus restricting the editing of all pages in Main. Now we want Main.WikiSandbox to be editable without a password. Using

clear

for the edit password for Main.WikiSandbox *doesn't unprotect the page*, because the password is being set by the group. Instead, we set the edit password for Main.WikiSandbox to the special value

@nopass

which tells Pm Wiki to ignore any site-wide or group-level passwords for that page.

How can I password protect all the pages and groups on my site? Do I really have to set passwords page by page, or group by group?

Administrators can set passwords for the entire site by editing the config.php file; they don't have to set passwords for each page or group. For example, to set the entire site to be editable only by those who know an "edit" password, an administrator can add a line like the following to local/config.php:

```
$DefaultPasswords['edit'] = crypt('edit_password');
```

For more information about the password options that are available only to administrators, see [PasswordsAdmin](#).

I get http error 500 "Internal Server Error" when I try to log in. What's wrong?

This can happen if the encrypted passwords are not created on the web server that hosts the Pm Wiki.
The crypt function changed during the PHP development, e.g. a password encrypted with PHP 5.2 can not be decrypted in PHP 5.1, but PHP 5.2 can decrypt passwords created by PHP 5.1.
This situation normally happens if you prepare everything on your local machine with the latest PHP version and you upload the passwords to a webserver which is running an older version.
The same error occurs when you add encrypted passwords to local/config.php.

Solution: Create the passwords on the system with the oldest PHP version and use them on all other systems.

How can I create private groups for users, so that each user can edit pages in their group, but no one else (other than the admin) can?

Modify the edit attribute for each group to id:username, e.g. set the edit attribute in [Jane Doe.Group Attributes?](#) to id:[Jane Doe?](#).

There is a more automatic solution, but it's probably not a good idea for most wikis. Administrators can use the [AuthUser](#) recipe and add the following few lines to their local/config.php file to set this up:

```
$group = FmtPageName('$Group', $pagename);
 $Default Passwords?['edit'] = 'id:'.$group;
 include_once("$Farm D?/scripts/authuser.php");
```

This automatically gives edit rights to a group to every user who has the same user name as the group name. Unfortunately it also gives edit rights to such a user who is visiting a same-named group not just for pages in that group, but for any page on the wiki that relies on the site's default edit password. This can create security holes.

 How come when I switch to another wiki within a farm, I keep my same authorization?

Pm Wiki uses PHP sessions to keep track of authentication/authorization information, and by default PHP sets things up such that all interactions with the same server are considered part of the same session.

An easy way to fix this is to make sure each wiki is using a different cookie name for its session identifier. Near the top of one of the wiki's local/config.php files, before calling authuser or any other recipes, add a line like:

```
session_name('XYZSESSID');
```

You can pick any alphanumeric name for XYZSESSID; for example, for the cs559-1 wiki you might choose

```
session_name('CS 559 SESSID?');
```

This will keep the two wikis' sessions independent of each other.

Is it possible to test the password level for display and/or if condition? Example: * (:if WriterPassword:) (display Edit link) (:ifend:)

You can use `(:if auth edit:)`. See [ConditionalMarkup](#).