That’s an interesting title, eh? You wonder „ssh public key distribution“ with **dokuwiki**? Yes. Keep reading.

Task: Create a simple and lean SSH key distribution solution for multiple linux-servers. Do this quick and don’t spend much time doing it.

Why: We need a simple solution to distribute ssh public keys of people allowed to access certain servers and keep that updated as easy as possible. And we need a nice overview of the current situation at any times.

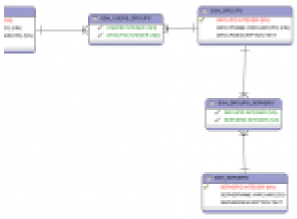
===== The dokuwiki-part =====

First thing to think about is a UI for this. Having the task to not spend much time thinking about it, I came to a very simple but IMHO really sophisticated method:

We naturally need some kind of database for this. The keys should be stored and the connection between the key and a server should be done. That’s a very simple database layout.

For the UI-part we need a simple [[http://en.wikipedia.org/wiki/Create,_read,_update_and_delete|CRUD]]-solution for this database, nothing more. We’re using a [[http://www.dokuwiki.org|dokuwiki]] at our site and I came across the wonderful [[http://www.dokuwiki.org/plugin:database2|Database2-plugin]]. With this plugin you can easily create a CRUD-UI for any [[http://de3.php.net/manual/en/book.pdo.php|PDO-enabled database]] very quickly.

So I created the following database design:



(I actually used database2 and it created the tables for me. Did I mention, that it was a neat plugin?)

That completes the UI.

===== The lbows-part =====

My „baby“ [[http://www.lbows.org|lbows]] was used for the second part: Bringing the keys to a form like the //authorized_keys//-file used by SSH to the server. That should’nt include much requirements on the server side. A simple download would be great.

I created a small lbows module for that (see [[https://github.com/dploeger/lbowsmodules/wiki/SshKeyDistribution]]) and even updated lbows‘ rest module to handle my requirement (so if you want to use it, you’ll have to update lbows).

After installing and configuring, I got my backend server for the key distribution. My „key distribution center“, if you like and the kerberos guys don’t kill me.

===== The server-part =====

On the server I would simply create a cronjob that downloads my authorized_keys file by doing:

```
wget -q -O- „http://lbows/index.php/SshKeyDistribution/rest/getAuthorizedKeys/`hostname -s`?response“ > /root/.ssh/authorized_keys
```

Nice.

===== Conclusion =====

When it comes to complex administrator tasks, it’s often wise to use the KISS-strategy („Keep it simple, stupid“; whereas some folks rather say „Keep it simple, sweetie“). Some complex tasks are actually very simple if you think outside the box.