**User guide**

Introduction:

Integration of different systems raises the need to provide a secure channel between those systems. For creating secure channel in the system, the application must to share a secret (in case of symmetric cryptography - private key. And in the case of public cryptography- each application need to know the public key of the other , and must to Ensure the authenticity of this public key ) .

To solve the above problems we build key management system. In this system we allow the system administrator capability of - impose a task of creating keys on one program, and deliver this keys securely to other programs .

The system will be divided into two parts:

**Agent** - software that is installed on each client. This software will run continuously on the client computer. This program will get tasks from the system administrator, and will perform those tasks.

There are two kinds of tasks. Task that related to keys (like generate new secret key – or install trust certificate in a keystore). And other task that related to the agent management (like change the agent configuration ..) .

Because a number of programs (that use our system) may run on one computer, and each program may use different type of keystore, the agent program will support plugins architecture, when each plugin supports different type of keystore. For each program that needs to use keystore, we will create an instance of the suitable plugin , that will take care of this keystore .

All the tasks that related to keys will be made via the suitable implementor instance. So each implementor’s plugin need to support operation of -creating new keys and store keys that deliver from the server.

**Server** - provide a web site for the system admin. From this site the admin can impose tasks on the agents, and also follow the agents operations. The web-server also delivers the tasks and the keys to the agent. (Details and examples about the server UI operations found later in this document)

**Installation instructions for the server**

*Windows :*

1. Install apache with php and openssl on your computer

(make sure – that php enable the openssl function – you can check it with the “phpinfo()” command)

1. Install mysql on your computer
2. Enable mode ssl in apache
3. Run the install.jar and follow the instructions
4. Run the batch file “run.bat”
5. Restart apache

*Linux:*

1. Install apache with php and openssl on your computer
2. Install mysql on your computer
3. Enable mode ssl in apache
4. Add project directory to htdocs on your apache
5. Use this tutorial to create keys for your sever in open ssl) it also useful for configure apache) : http://www.cafesoft.com/products/cams/ps/docs30/admin/ConfiguringApache2ForSSLTLSMutualAuthentication.html
6. Change the apache httpd-ssl conf file that so the serverCert and caCert will be the new public key file that you just created , and also the server private key – to the private that you just created(in step 5).
7. Add this lines to the end of the apache httpd-ssl conf file :

<Location /*projectRoot*>

SSLRequireSSL

</Location>

<Location /*ProjectRoot*/trusted>

SSLRequireSSL

SSLVerifyClient require

SSLVerifyDepth 10

</Location>

1. Run from the command line the script on ProjectRoot/saveUser.php with your admin username and password (example “php saveUser.php ben 12345” )
2. Create database in mysql and run the creation script that found in : sql/serverCeateTables.sql
3. Update the database properties in the configuration file “conf.cnf”
4. Restart apache
5. Run the certificateDb.jar

**After installation of the server the relevant URL-s are:**

Register Agent- yourServerHost/ projectRoot/a.php

Check trusted connection - yourServerHost / projectRoot /trusted/trustCon.php

Send ACK - yourServerHost / projectRoot /trusted/getAck.php

Get tasks- yourServerHost / projectRoot /trusted/getTasks.php

Inform about new implementer- yourServerHost/ projectRoot /trusted/newImp.php

Login into the UI - yourServerHost/ projectRoot /UI/login.php

You can change to more convenient url-s by adding “.htaccess” file in the folder “htdocs” .

An example of “.htaccess” file:

*RewriteEngine On*

*RewriteRule ^regAgent/?$ MA/a.php*

*RewriteRule ^chekTrust/?$ MA/trusted/trustCon.php*

*RewriteRule ^SendAck/?$ MA/trusted/getAck.php*

*RewriteRule ^getTasks/?$ MA/trusted/getTasks.php*

*RewriteRule ^newImp/?$ MA/trusted/newImp.php*

**Agent**

Installation instructions for agent (windows\linux):

1. Edit the configuration file “conf.cnf” of agent install ( a json file ) .

(the arguments that not appear in the configuration file - can be supplied in the argv (like -agentName name) or inserted manually on program loading ).

**The parameters are:**

agentName- the unique id of the agent

agentServicePath- full path to directory to agent service

serverCaPath- path to the certificate of the server

instName – the user name of the installer

instPassword – the password for the installer

serverRegAgentPath – the url of agent registration

serverChekTrustConectionPath- the url of the server for check trust connection

ksPassword- the password to the key store

1. Run the agent-install jar (from cmd ) .
2. Edit the URL section in the configuration file of the agent service.
3. Add the directory of jnotify to the system path (found in projDir/agent/bin/service/jnotify)

(If you don’t want to change the system path you I can run stage 5 with –Djava.library.path=”path to jnotfiy”

(like java –Djava.library.phath=”c:\blabla\jnofiy” –jar as.jar))

1. Run the agent service jar (from cmd )

Adding new plugin to agent :

The plugin has to be a jar file that contains a class that implements the implementor interface. For install this new plugin, you just need to put the jar file in the directory of plugins in the agent service folder. The operation can be done even when the agent-service is running, and the plugin will be loaded dynamically.

You can find existing eclipse project which is framework for creating new plugins here :

<https://github.com/eyal-leshem/pluginframework>

(The agent program comes with plugin for java key-store , and plugin for apache built-in )

Adding new instance of implementer:

This must be done from the website (See the explanation about the server) .

**Server UI operations**

Login **-**

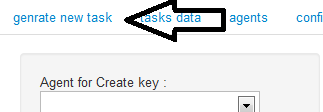
****

Insert the user name and password and press submit .

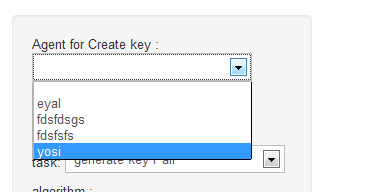
Generate new task

There are 4 kinds of tasks:

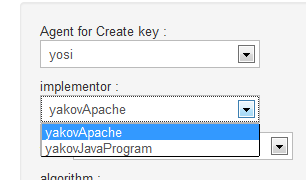
1. Generate key pair – the “agent for create key” creates key pair of public key and private key. The agent returns the public key in x.509 certificate . This certifcate will be delivered to the “implementer for share“ that will install it as a trusted certificate.
2. Generate secret key - “agent for create key” creates secret key . and the key will be delivered to the “implementor for share“ .
3. Remove certificate – delete certificate according it's serial number.
4. Add to CRL – add this serial number to certificate revocation list.
5. Choose the generate new task tab :



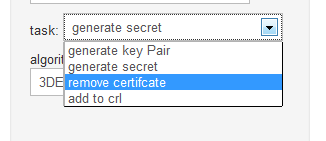
1. Select an agent :

****

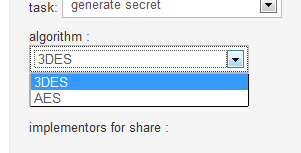
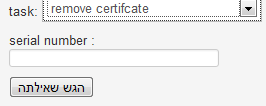
1. Select the implementor :



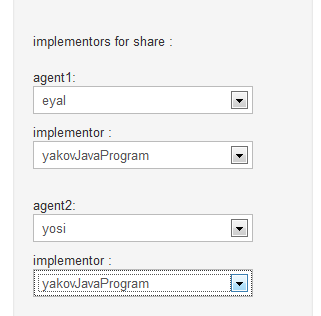
1. Choose task :



1. choose algorithm: (or Provide serial number -depends on the tasks kind)

** **

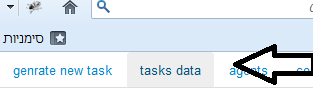
1. Choose the implementors that will share this key/ certificate



7.Submit the form.

**Get Tasks data:**

1. Choose the “tasks data” tab:

****

1. Check the tasks tables:



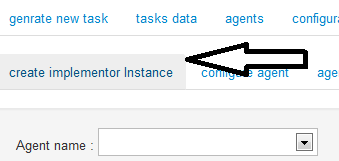
Tasks in queue – table that contains the data about tasks that are waiting to be sent to the relevant implementor.

Done tasks – data about tasks that done successfully.

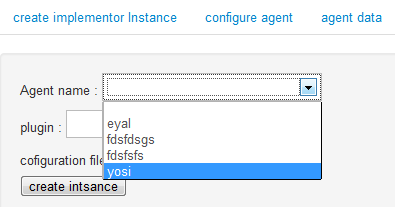
Failed tasks - tasks that failed and will not be sent to agents again .

**Create new implementor instance:**

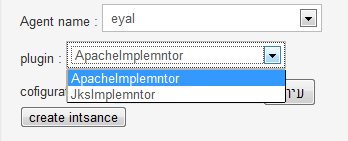
1. Go to the tab of –“agents”/ “create new implementor instance”



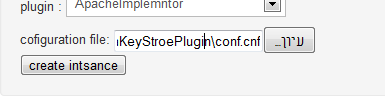
1. Choose agent id :

****

1. choose plugin name :



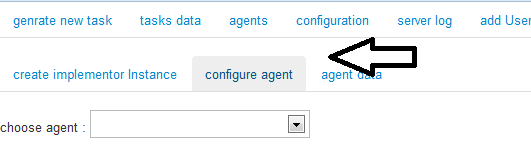
1. upload configuration file (a json string that will be given to the constractor of the instance )

****

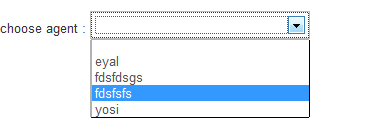
1. Create the instance - new task of “create new instance“ where added to the database. When the implementor will perform this task , he will send ack. Then the implementer will be shown in task generation window and it will be possible to send to this implementor new key tasks.

**Change agent configuration :**

1. Choose the “agent”/”configure agent” tab

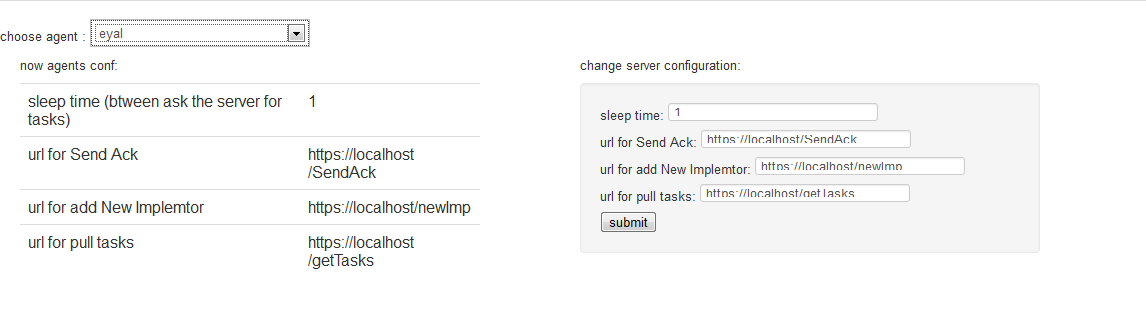
****

1. Choose agent –



1. In the left side is the current configurations now.

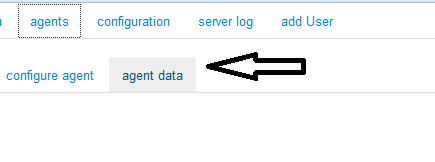
In the right side is the from for creating new configurations



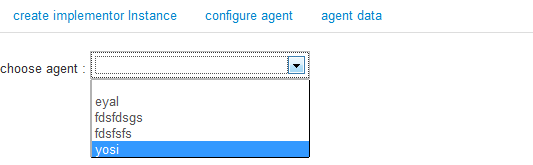
1. After submitting the form new task of “change conf” will be created in the server . when the required agent ask for update – he will get this task and change his configuration.

**Check agent data :**

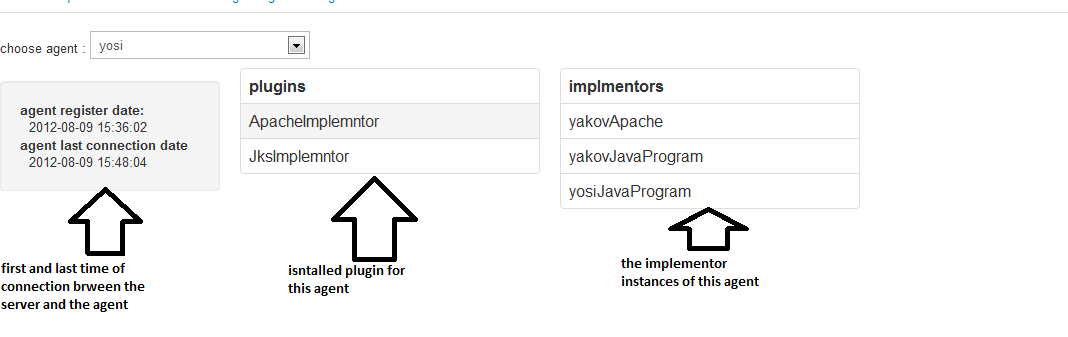
1. Choose the tab “agents/ agent data” :



1. Choose the relevant agent :

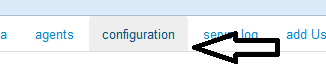


1. Check the agent data:

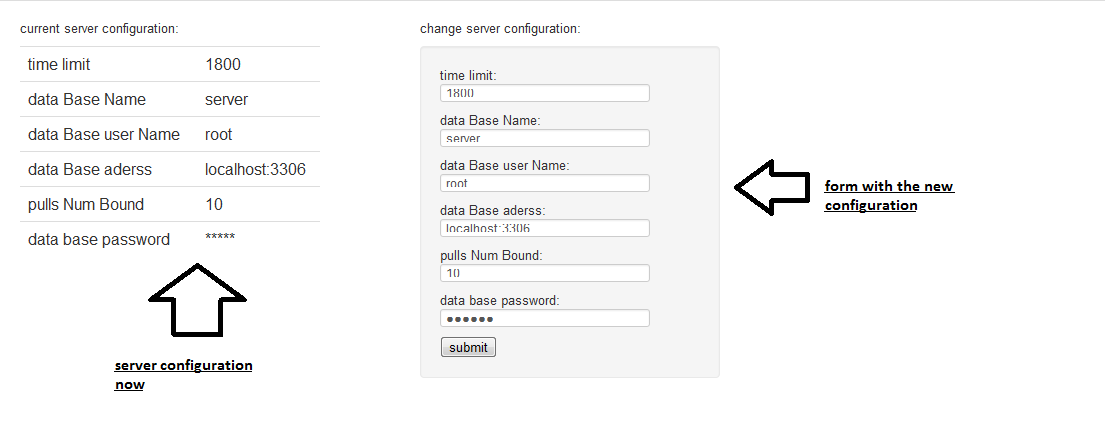


**Change server configuration:**

1. Choose the “configuration” tab :



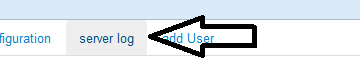
1. Change the server configuration – and submit the task :



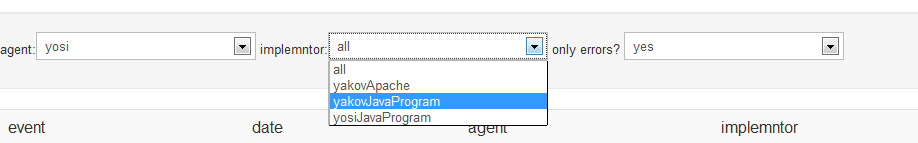
**Warning : in case of bad parameters of configuration the web site will stop to work**

**Check server log:**

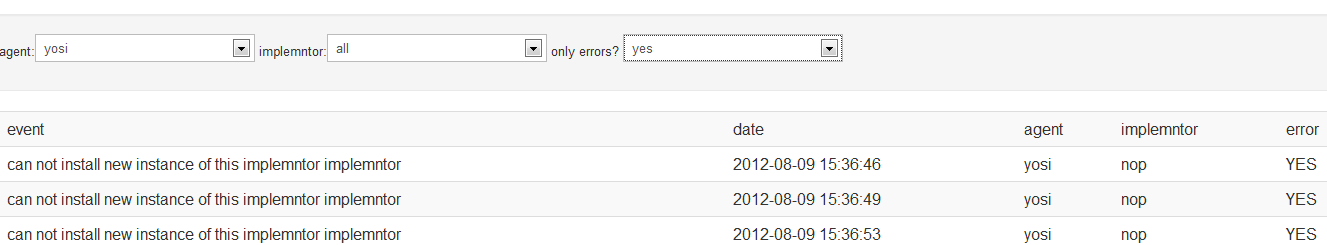
1. Choose the “server log” tab :

****

1. Use the filters to filter the data :



1. Check the data in the table :

****

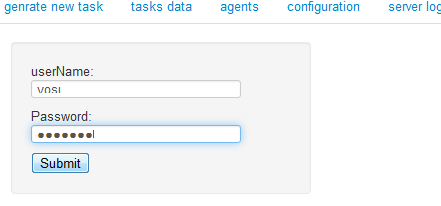
**Add new user :**

Use this to add new system manager user. The user can use his username and password to get into the website , and to register new agents .

1. Choose the add user tab :



1. Insert user name and password:



1. Submit the form.