==== OpenPKSD Trusted Key Server ====

Auther: Hironobu SUZUKI
Email: hironobu /at/ openpksd /dot/ org
hironobu /at/ h2np   /dot/ net

What's That

OpenPKSD Trusted Key Server, called OpenPKSD-TKS, supports to supply a public key that is permitted to distribute by the public key owner, called PKO.

All public keys in OpenPKSD-TKS are "trust key" because PKOs subimtted it with PKO's digital signature.

So, you can get a public key that is allowed by PKO.

Background

There are some OpenPGP public keyservers, pksd --- Horrowitz's version of OpenPGP public keyserver is well known and has running at many keyserver sites, OpenPKSD --- Hironobu's version of OpenPGP public keyserver, SKS --- Yaron Minsky's verion of OpenPGP public keyserver and others.

The keyserver accepts any public key that is submitted not only PKO but also someone who has a public key. As matter of fact, some public key is submitted against intention of PKO.

A public key has some someone's digital signature that is used for "web of trust". Any OpenPGP user can sign on someone's public key and submitted into keyserver. It means a public key is opened against PKOs will. When signer sign on a public key, they must check PKO carefully. According to the manner of the key signing party, photo ID is required to avoid spoofing. But many signer sign on without any kind of verifying. Many PKO don't think that public keyserver is not suitable place to distribute their own public key, so they use their web server instead of public keyserver.

This situation causes that public keyserver is considering useless place for my public key distribution.

Solution

OpenPKSD-TKS provides to show, to hide and to submit a public key by

only PKO. A submitted signed public key that submitted someone except
PKO will be held server and sent its notice to PKO. After that, PKO
can get from OpenPKSD-TKS.

Glossary

OpenPKSD --- OpenPGP public keyserver daemon, program that was
             developed by Hironobu SUZUKI.

TKS --- Trusted key server, something worthy of trusty :-)

PKO --- Public key owner.

$Id: 01_01_overview.txt,v 1.2 2005/08/31 01:12:57 hironobu Exp hironobu $INSTALL
=======

Before Starting Installtion

Step 1: Ruby, PostgreSQL and GNUPG

OpenPKSD-TKS need Ruby, PostgreSQL and GNUPG package and assume tools
are installed under local directory to avoid "Server system don't work
after my distribution is updated" situation.

So, you have to install Ruby, PostgreSQL and GNUPG.

Please read those documents before starting your installation.

```
 [ ] 02_01_postgresql-install.txt
 [ ] 02_02_gnupg-install.txt
 [ ] 02_03_ruby-install.txt
 [ ] 02_04_misc-install.txt
```

Step 2: Install Directory

Make install directory. Default is "/tks".

OpenPKSD-TKS Install

Step 1: Make /tks as install directory.

```
 # mkdir  /tks
 # chmod tks /tks
 # chgrp tks /tks
```

Step 2: OpenPKSD-KTS install it.

Simply, you type "tar"f, "./configure" and "make install".

```
  % cd /tks
  % tar zxvf /somewhere/openpksd-tks-"version-number".tar.gz
  % cd openpksd-tks-"version-number"
  % ./configure
  % make install
```

Step 3: Database initialization.

Database initialization is required before you run OpenPKSD-TKS
system. You must initialize new database if you don't have it.

    ** CAUTION ** DON'T DO THAT IF YOU ALREADY HAVE OpenPKSD-TKS
    DATABASE. IF YOU DO THAT, YOUR DATABASE MIGHT BE CORRUPTED.

```
  % /tks/bin/init_db
```

Step 4: Run Database (postgresql)

OpenPKSD-TKS is required PostgreSQL database engine. So, you must run
PostgreSQL daemon by "start_db" script before you start openpksd-tks
services.

```
  % /tks/bin/start_db
```

  ** NOTE ** Some directories (default: "/tks/var/...") must be
  permitted to create and to write for user who run "start_db".

Step 5: Stop Database

If you need to stop PostgreSQL daemon, use "stop_db" script.

  ** CAUTION ** DON'T USE "kill" COMMAND WHEN YOU STOP PostgreSQL
  DAEMON. IF YOU DO THAT, DATABASE MIGHT MIGHT BE CORRUPTED.

```
  % /tks/bin/stop_db
```

Step 6: Init Tables

If you run openpksd-tks first time on your site, you have to
initialize database tables.

```
  % /tks/bin/table_init
```

  Note: bin/start_db.sh is required before run this procedure.

Run OpenPKSD

After PostgreSQL daemon was started, you may run OpenPKSD start
script.

```
  % cd /openpksd
  % bin/start_openpksd.sh
```

Init DB By Your Pubring

Read doc/howto_init_openpksd.txt file before you do that.

cgi-bin command installation

See file docs/cgi-base_lookup.txt.

Acknowledgement

OpenPKSD-TKS project was funded by The Information-technology
Promotion Agency(IPA) in 2005.

The Information-technology Promotion Agency(IPA) was established on
October 1, 1970 by the Japanese statute "Law on Facilitation of
Information Processing." Its purpose was to promote information
processing through high-level computer applications, primarily by
encouraging the development and wider utilization of computer program
products and providing aid to companies in the information processing
service. In order to ensure that the agency would be amply infused
with the creativity and independent sprit of the private sector, it
was founded by eminent people drawn from the private sector, and was
authorized by the goverrnment as the sole official organization under
the Law on Facilitation of Information Procedding. IPA's original
capital was also jointly funded by the government and private sources.

More detail, see http://www.ipa.go.jp/ipa-e/index-e.html

$Id: 02_00_install.txt,v 1.2 2005/08/31 01:12:57 hironobu Exp hironobu $

$Id: 02_00_install.txt,v 1.2 2005/08/31 01:12:57 hironobu Exp hironobu $

How to compile and install postgresql
====================================

Step 1: Get PostgreSQL source from www.postgresql.org

Step 2: Install it.

```
        Example

       % tar zxvf postgresql-8.0.3.tar.bz2
       % cd postgresql-8.0.3
       % ./configure
       % make
       % su
       # make install
```

/usr/local/pgsql direcotry is a default install directory. And it is
a default PostgreSQL directory path for OpenPKSD-TKS.

If you want to install local version of PostgreSQL, use prefix option
when you run configure command as below.

Ex)
```
       ./configure --prefix=/tks/pgsql
```

--

Debian GNU/Linux
================

If you use Debian GNU/Linux, libreadline4-dev is required before
compiling.

```
# apt-get install libreadline4-dev
```

$Id: 02_01_postgresql-install.txt,v 1.1 2005/07/31 04:20:11 hironobu Exp $

How to compile and install GNUPG
================================

GNUPG will be used for some functionality of openpksd. Most of
openpksd source codes are written by Ruby and Ruby version of code
conversion is slower than native object codes like as gpg command. So,
gpg will be used.

Step 1: Get GNUPG source from www.gnu.org

Step 2: Install it.

```
        Example

       % tar zxvf gnupg-1.x.x.tar.gz
       % cd gnupg-1.x.x
       % ./configure
       % make
       % make install
```

If you want to install local version of gpg, use prefix option.

ex)
```
       % ./configure --prefix=/tks
```

--

/usr/local/bin/gnupg is the file path of default install command. And
it is a default path for OpenPKSD-TKS.

$Id: 02_02_gnupg-install.txt,v 1.1 2005/07/31 04:20:11 hironobu Exp $

RUBY
====

HOW TO GET LATEST RUBY
=========================

Ruby ftp directory is here. Get ruby at your own risk.

```
       ftp://ftp.ruby-lang.org/pub/ruby/
```

INSTALL
=======

Step 1: Extract Ruby tar ball file.

```
  % tar zxf ruby-1.8.2.tar.gz
```

Step 2: Do "./configure" and "make" at ruby-1.8.2 directory.

```
  % cd ruby-1.8.2
  % ./configure
  % make
```

Step 3: Install ruby into your system by root.

```
  % su
  # make install
```

PostgreSQL interface

====================

Step 1: You can get ruby module for PostgreSQL from URL as below.

```
  URL http://ruby.scripting.ca/postgres/
```

Step 2: Extract and make Makefile with ruby.

```
   %  ruby extconf.rb --with-pgsql-include-dir=/usr/local/pgsql/include  --with-pgsql-lib-dir=/usr/local/pgsql/lib
```

  If you don't use /usr/local/pgsql, type

```
  % ruby extconf.rb
```

Step 3: Make and install it into your system.

```
  % make
  % su
  # make install
```

Dynamic Loading Trouble
======================

There are some hits for dynamic loading trouble.

1) Check your loader path. Check /etc/ld.so.conf or load path environment
   variable.

```
    ex) Linux
    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

2) Check compiler version and binutils version. Gcc version 3.x is
   required newer version ld.

$Id: 02_03_ruby-install.txt,v 1.1 2005/07/31 04:20:11 hironobu Exp $
Bash (F.A.Q)
===========

"bash" is recommended for OpenPKSD-TKS installation. *BSD or/and
Solaris don't have "bash" by default. If your system haven't "bash",
install it.

$Id: 02_04_misc-install.txt,v 1.1 2005/08/31 01:12:57 hironobu Exp hironobu $
SHELL SCRIPTS
=============

deletekey.sh
       Delete a key from openpksd database.

```
        % deletekey.sh 0xFFFFFFFF

init_db.sh
        INIT Database.
        !!!!DON'T RUN AFTER SYSTEM INITIALIZATION WAS DONE!!!!

        % init_db.sh

openpksd_dump.sh
        Dump openpksd database to stdout. Note that this command always
        make full backup. So, db.out size become over 3GB!!

        % openpksd_dump.sh  > db.out

openpksd-mail.sh
        Filter for /etc/aliases to recieve sync site mail.

        /etc/aliases sample
        openpksd.pgp-public-keys: "|/openpgp/bin/openpksd-mail.sh"

openpksd-dequeue-mail.sh
        dequeue sync mail in incoming directory.
         Note: only 10 queue file will be removed for once.

        12 * * * *    sh openpksd-dequeue-mail.sh >& /dev/null

psql_db.sh
        Wrapper of psql for openpksd database. You should remind that
        this command access database directly. So, you can destroy
        openpksd database as easy as pie.

        % psql_db.sh
         openpksd=#              (psql prompt)

start_db.sh
        Start PostgreSQL daemon.

start_openpksd.sh
        Start openpksd daemon. So, you can use hkp (11371) protocol after
        started.

stop_db.sh
        Shutdown PostgreSQL daemon.
        !!NEVER SEND SIGNAL or USE KILL COMMAND TO STOP DATABASE!!
        If you kill PostgreSQL daemon process, database contents in cache
        will be breaken.

syncsite_dequeue.sh
        Send queued key which will be sent to sync sites.
        % syncsite_dequeue.sh   (send queued mails once)
        or
        % syncsite_dequeue.sh infinity_loop  (send queued mails forever)

template.sh
        Nothing to do. (see template.sh)

vacuum.sh
        PostgreSQL vacuum command for openpksd database. Clean up garbage
        in database and optimaize openpksd database.

        0 5 * * 0     sh /openpksd/bin/vacuum.sh
        (vacuum.sh run at 5 am every sunday)

dumpkey.sh
        Shell wrapper for tools/dump_openpgp_keys.rb.  Note! this command
        will dump whole of public keys. You should check empty disk size
        to avide disk full problem before you do this.


         % dumpkey.sh dumpedfile

$Id: 03_01_commands_script.txt,v 1.1 2005/08/31 01:12:57 hironobu Exp hironobu $
```

CGI-BIN BASE LOOKUP
===================

This tools will be used for cgi-bin command without openpksd daemon.
The interface page of html file is "openpksd-dist/src/html/findkey.html"

```
Step 1: Move src/ruby-codes/tools/cgi-bin directory and install
         cgi-bin command into cgi-bin directory.

      If cgi-bin is /home/apache/cgi-bin, then

      ---
      % make install
      ....
      install  lookup  /home/apache/cgi-bin
      install  ../ruby-codes/tools/lookup.rb /home/apache/cgi-bin
      ---

      If you must specify cgi-bin directory, then

      ---
      % make CGIBIN=/somewhere/specified/cgi-bin-direcotry install
      ---

Step 2: Make sure db_account value in etc/openpksd.conf. db_account
         must be user name who was start db_start.sh (process owner of
         PostgerSQL)
```

```
    File etc/openpksd.conf (default)
    ---
    #
    # Database User Account (default nil)
    #  This value must be set when you use cgi-bin lookup.rb
    # db_account   openpksd
    ---

   ex)
   ---
   #
   # Database User Account (default nil)
   #  This value must be set when you use cgi-bin lookup.rb
   db_account   hironobu
   ---

  ---
   % make pgpdump
  ---
```

PGPDUMP
========

Cgi-bin based pgpdump function is available. I think that this cgi-bin
command is an optional service.

Step 1: Check pgpdump command on your site. If you haven't it,
        get it from URL as below.

        http://pgp.iijlab.net/pgpdump.html

Step 2: Check PGPDUMP path in file pgpdump.in. If PGPDUMP path is
        wrong, edit it by yourself.

```
    File pgpdump.in (default)
    ---
    export PGPDUMP=/usr/local/bin/pgpdump
    ---

    ex)
    ---
    export PGPDUMP=/openpksd/bin/pgpdump
    ---
```

Step 3: After file pgpdump was installed into cgi-bin directory,
       change lookup in cgi-bin directory.

```
    File lookup.in (default)
    ---
    export LOOKUP_PGPDUMP_FORMAT=NO
    ---

    ex)
    ---
    export LOOKUP_PGPDUMP_FORMAT=YES
    ---
```

Step 4: Install lookup and pgpdump scripts into cgi-bin directory.
        You can install lookup script and can build pgpdump script by
        "make foo", but pgpdump script must copy by manually
        operation.

```
   ---
   % make lookup
   % make install
   % make pgpdump
   % cp ./pgpdump /home/apache/cgi-bin
   ---
```

$Id: 03_02_cgi-base_lookup.txt,v 1.1 2005/08/31 01:12:57 hironobu Exp hironobu $

Ticket

Ticket is use as "one time pad" tha PKO (public key owner) is correct
or not.

```
  PKO                           OpenPKSD-TKS
  ask ticket  -----https------>+
        [email/keyid/request]

                              make one-time pad
                             (encryted by keyid public key)
  decrypt     <------email-----+
                  [one-time pad]

  activate    ------https------> activate request
               [one-time pad]
```

Request

```
     Request    How to work
     ---------+---------------------------
     hide       hide my public key
     show       show my public key
     get        get public keys from key queue
```

Activation

==========

Your request will keep in ticket queue table. You must activate your
request in 24 hours. After 24 hours, your ticket will be waste.

You submit 'ticket id #' and 'one-time pad' then your request is
activated.

```
         ticket id #  --- 10 digit.
                          0123456789

         one-time pad --- 19hex
                          FFFF FFFF FFFF FFFF FFFF
```

"ticket id" is actually database index. one-time pad is random number.

```
--
--  OpenPKSD-TKS TABLE DEFINITION
--


-- KEYCONTENTS     :  used for storing public key
-- "longkeyid"     :  Master key ID,  (ex 0xBD58B69202912C53)
-- "shortkeyid"    :  Short (32bit) keyID, to use searching key.
-- "keycreattime" : This public key is created, not db entry time. It
--                   is UNIX time.
-- "hashvalue"    : A public key is same or not.
-- "fingerprint"  : To show public key fingerprit.
-- "keystatus"    : The status of this key. "hidden/published"
-- "createtime"   : Create time of this entry.
-- "updatetime"   : Update time of this entry.

CREATE TABLE KEYCONTENTS(
       LONGKEYID NUMERIC(20),
       SHORTKEYID NUMERIC(10),
       KEYCREATTIME INT8,
       HASHVALUE TEXT,
       FINGERPRINT TEXT,
       KEYSTATUS TEXT,
       PRIMARYMAILADDRESS TEXT,
       CREATETIME TIMESTAMP,
       UPDATETIME TIMESTAMP,
       BITLENGTH INT,
       CONTENTS TEXT
);

-- USERID is used for linking between a userid and a public key.  A
-- public key can have multiple userids.
```

```
CREATE TABLE USERID (
       USERID TEXT,
       LONGKEYID NUMERIC(20),
       CREATETIME TIMESTAMP
);

-- MAILADDRESS is used for linking between a mail address and a public
-- key. A public key can have multiple mail addressed.

CREATE TABLE MAILADDRESS (
       LONGKEYID NUMERIC(20),
       CREATETIME TIMESTAMP,
       MAILADDRESS TEXT
);

--  KEYQUEUE is used for queue that have third party submitted public
--  key.

CREATE TABLE KEYQUEUE (
       LONGKEYID NUMERIC(20),
       SUBMITTEDTIME TIMESTAMP,
       CONTENTS TEXT
);

-- KEYWORD is used for keyword searching for public key (not used yet)

CREATE TABLE KEYWORD (
       KEYWORD TEXT,
       LONGKEYID NUMERIC(20)
);

-- TICKET is used for ticketing


CREATE TABLE TICKET_QUEUE (
       ONETIMEPAD TEXT,
       REQUEST TEXT,
       KEYID NUMERIC(20),
       EMAIL  TEXT,
       ISSUEDATE TIMESTAMP,
       ACTIVATEDATE TIMESTAMP
);
Example of Bad Request
======================

Example of Bad Request

----

This is a bad request
```

```
http://pgp.nic.ad.jp:11371/pks/lookup?op=get&exact=on&search=0x02912CDD


---
Current version
================

Current version of Marc's pksd use HTML 1.0 format, Very simple
format. openpksd will use HTML 1.0 for the matter of compatibility.

HTTP/1.0 400 Bad Request
Server: pks_www/0.9.4+patch1
Content-type: text/html

<HEAD><TITLE>400 Bad Request</TITLE></HEAD><BODY></BODY>

HTML 2.0 Version
================

This is a HTML 2.0 version of "404 Not Found" results.  This is a just
sample to show HTML 2.0 format. openpksd don't use it.

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL /404_not_found was not found on this server.<P>
</BODY></HTML>

---


Sequence of communication
========================

This is a sample of sequence of communication between client and
server.  I'd like show two type of "get" option, first sample is
retrieved by keyid, another is retrieve by userid.

See also example-of-url.txt

----

GET /pks/lookup?op=get&exact=on&search=0x02912C53
HTTP/1.0 200 OK
Server: pks_www/0.9.4+patch1
Content-type: text/html

<title>Public Key Server -- Get ``0x02912C53
```

```
''</title><p>
<h1>Public Key Server -- Get ``0x02912C53
''</h1><p>
<pre>
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 5.0
Comment: PGP Key Server 0.9.4+patch1

mQGiBDc49ycRBACXq1NiBH+MuxolCljKYyCdQ9SNUBVwaON2PXvCdngoI7aaQLzP
.....
AgAMBQI3OPiwBQkDwmcAAAoJEL1YtpICkSxT3mUAn3n74KvwQJK6zCp//TC1PLDZ
ODrqAJ92+XiRSW64Q8W7ulnopjTWIzD+PA==
=atiE
-----END PGP PUBLIC KEY BLOCK-----
</pre>

---

GET /pks/lookup?op=get&exact=on&search=hironobu@h2np.net
HTTP/1.0 200 OK
Server: pks_www/0.9.4+patch1
Content-type: text/html

<title>Public Key Server -- Get ``hironobu@h2np.net
''</title><p>
<h1>Public Key Server -- Get ``hironobu@h2np.net
''</h1><p>
<pre>
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 5.0
Comment: PGP Key Server 0.9.4+patch1

mQCNAi7ri9IAAAEEALsCdefpinp9aVHq40FecEKMBhup3XgmiSQANBs1FANjX8wL
....
AiTykESM1VIovj2ICyp0ane/l8wEiUhQrbkv73SCs0wTO3B/yD/FP4GBQ5HixAMM
9aLhbYZgeG8c1kVT+v421Njpx8DCo9jL1gxqfLthICG2OCYQQwqrJT3WM3b1kW73
=MC49
-----END PGP PUBLIC KEY BLOCK-----
</pre>
```

Example of how to show it by HTML
=================================

Example of returned HTML page when index request was issued.

```
----------

<title>Public Key Server -- Index ``hironobu suzuki
''</title><p>
<h1>Public Key Server -- Index ``hironobu suzuki
```

```
''</h1><p>
<pre>
Type bits/keyID    Date       User ID
pub  1024/<a href="/pks/lookup?op=get&search=0xF2AFC486">F2AFC486</a> 1997/09/03 Hironobu SUZUKI
&lt;<a href="/pks/lookup?op=get&search=0xF2AFC486">hironobu@h2np.suginami.tokyo.jp</a>&gt;
pub  1024/<a href="/pks/lookup?op=get&search=0x2066FAFD">2066FAFD</a> 1994/12/11 Hironobu SUZUKI
&lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">hironobu@icat.or.jp</a>&gt;
                              &lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">
hironobu@h2np.suginami.tokyo.jp</a>&gt;
                              Hironobu SUZUKI &lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">
hironobu@h2np.net</a>&gt;
                              Hironobu SUZUKI &lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">
hironobu@sra.co.jp</a>&gt;
                              Hironobu SUZUKI &lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">
hironobu@h2np.suginami.tokyo.jp</a>&gt;
pub   512/<a href="/pks/lookup?op=get&search=0x1BE840B1">1BE840B1</a> 1993/09/26 *** KEY REVOKED
***
                              hironobu@sra.co.jp
                              Hironobu SUZUKI &lt;<a href="/pks/lookup?op=get&search=0x1BE840B1">
hironobu@sra.co.jp</a>&gt;
</pre>

----------

<title>Public Key Server -- Index ``0x02912C53 ''</title><p>
<h1>Public Key Server -- Index ``0x02912C53''</h1><p>
<pre>
Type bits/keyID    Date       User ID
pub  1024/<a href="/pks/lookup?op=get&search=0x02912C53">02912C53</a> 2001/01/11 Hironobu SUZUKI
&lt;<a href="/pks/lookup?op=get&search=0x02912C53">hironobu@h2np.suginami.tokyo.jp</a>&gt;
                              Hironobu SUZUKI (Independent Software Consultant) &lt;<a
href="/pks/lookup?op=get&search=0x02912C53">hironobu@h2np.net</a>&gt;
</pre>
```

```
Example of POST
================

Put ascii armor key into key server, like this.

---
POST /pks/add HTTP/1.0

Content-Length: 13303



keytext=-----BEGIN+PGP+PUBLIC+KEY+BLOCK-----%0AVersion%3A+GnuPG+v1.....
```

```
....
....
OWOHvIZBB3o5Ukpg9iZ1HC%0A%3DOSsb%0A-----END+PGP+PUBLIC+KEY+BLOCK-----%0A

--------------
```

Example of Search Index
=========================

Example of search index sequence between client and server.

```
---
GET /pks/lookup?op=index&exact=on&search=hironobu@h2np.net
HTTP/1.0 200 OK
Server: pks_www/0.9.4+patch1
Content-type: text/html

<title>Public Key Server -- Index ``hironobu@h2np.net
''</title><p>
<h1>Public Key Server -- Index ``hironobu@h2np.net
''</h1><p>
<pre>
Type bits/keyID    Date       User ID
pub  1024/<a href="/pks/lookup?op=get&search=0x2066FAFD">2066FAFD</a> 1994/12/11 Hironobu SUZUKI
&lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">hironobu@icat.or.jp</a>&gt;
                              &lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">
hironobu@h2np.suginami.tokyo.jp</a>&gt;
                              Hironobu SUZUKI &lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">
hironobu@h2np.net</a>&gt;
                              Hironobu SUZUKI &lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">
hironobu@sra.co.jp</a>&gt;
                              Hironobu SUZUKI &lt;<a href="/pks/lookup?op=get&search=0x2066FAFD">
hironobu@h2np.suginami.tokyo.jp</a>&gt;
</pre>
---
```

Example of URL
===============

get public key which has keyid 0x02912C53 from pgp.nic.ad.jp.

```
    http://pgp.nic.ad.jp:11371/pks/lookup?op=get&exact=on&search=0x02912C53
```

Each elements of URL
====================

"http://domain:11371/pks/lookup"

The formal port number is 11371. This port number is a changeable number.

"op="
Option has 3 type, "get", "index" and "vindex"

```
       "op=get"
       "op=index"
       "op=vindex"
```

"get" means "get a public key".
"index" means "show the public key information"
"vindex" means "show the public key detail information"
"index" and "vindex" retruns HTML format.

"search="
Search has 2 type, "str" and "0x-prefixed keyid number"

```
       "search=str"
       "search=0xFFFFFFFF"
```

"str" is a UTF-8 but this is a URL format. It's same as grep.
"0xFFFFFFFF" must be described following by "0x".

"fingerprint="
Fingerprint has 2 type. "on" or "off".
"off" is default, so "fingerprint=" is a same as "fingerprint=off".

```
    fingerprint=on
```

"exact="

Exact option effects "search=str" as exact-search. Default is "off" which behaves like as grep. My keyserver will ignore exact option and always "on". Because "grep" retrieves too many match keys, string match is always required. "off" is ignored and search option search exact matching.

```
    exact=on

---

"keytext="

Put pubkey. public keys must ascii-armored.

---

"get"

get tag for "op=" for get a public key.  "get" must use with searching
by keyid.

---

"index"

index tag for "op=" for searching key information.

---

"vindex"

verbose index tag for "op=" for searching key information.

---


"on"

switch keyword for fingerprint=, exact= options.
---

examples

http://localhost/pks/lookup?op=get&exact=on&search=marc@mit.edu
http://pgp.nic.ad.jp:11371/pks/lookup?op=get&exact=on&search=0x02912C53
http://pgp.nic.ad.jp:11371/pks/lookup?op=index&search=0x02912C53
http://pgp.nic.ad.jp:11371/pks/lookup?op=vindex&fingerprint=on&search=0x02912C53
---------

Retrun Status-code
==================


pksd supports as below

    HTTP/1.0 200 OK
    HTTP/1.0 404 Not Found
```

$Id: 04_03_example_of_hkp.txt,v 1.1 2005/08/31 01:12:57 hironobu Exp hironobu $