Digital Bank (DIBA)
Transport Routines

The transport sub directory of the DIBA project contains tools to encrypt / decrypt communications. They can be used as standalone utilities. These can be freely downloaded. See the links section at the end of the document. The utilities contain statically linked dependencies, it should work on every windows platform without any additional files.

There are four main utilities. 1.) dibakeygen 2.) dibakeyinfo 3.) dibaencrypt 4.) dibadecrypt

1.) dibakeygen

This is the key generator. Typical usage is dibakeygen keyname. It will generate a key pair in two files, “keyname.pub”, “keyname.key”. Respectively the public and private keys.

```
RSA key generation (of 2048 bits) can take a few minutes. Your computer needs to gather random entropy.

Please wait .................................

RSA key generation complete.

Please enter a password to lock your key pair.
This password must be retained for later use. Do not loose this password.

Warning! Weak option specified, recommended for testing only.
Enter  keypair  pass: ****
Confirm keypair pass: ****
Key 'UVvgx0hA6sMaELeSfGRGpjBK1wxKHeTM'  successfully saved to 'testkey.key' and 'testkey.pub'.
```

This utility has a number of options. It can generate keys from minimal input. The main options influence key strength, provide additional information for the key, check the integrity, check library integrity.

This is an example command line to generate strong key pair into the files mykey.pub, and mykey.key:

dibakeygen.exe -k 4096 mykey

The utility will prompt you for a password, which is used to encrypt the private key. Please remember this password, as without it the key is unrecoverable. Two different and strong encryption algorithms are used to protect your key, for extra safety. This tool (by default) will force you to enter a strong password. As an example, upper and lowercase letters, numbers and punctuation marks. You can specify the -w (week) option to allow the utility to accept any password with four characters of

longer.

You might also decorate the key with personalized information, like giving a name for the key, providing a description, and overriding the creator's name. Here is the information that the program itself provides you:

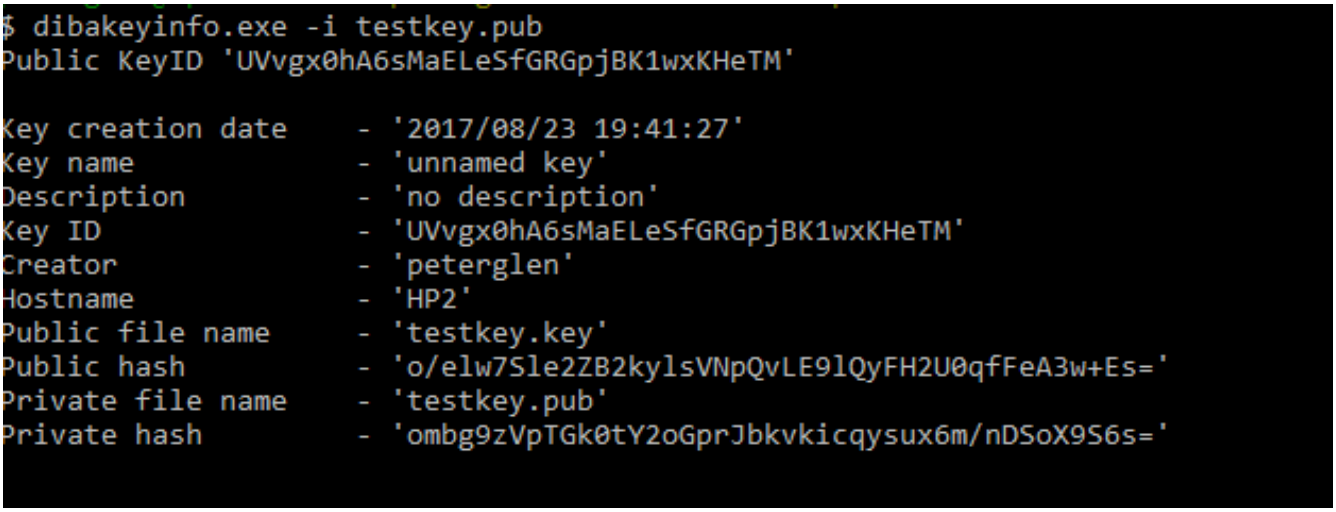
```
Generate Public / Private keypair into a set of key files.
Usage: dibakeygen [options] keyfile
Where 'keyfile' is the basename for .key .pub files. [keyfile.pub, keyfile.key]
Options can be:
               -k             --keylen      - key length in bits (default 2048)
               -v             --verbose     - Verbosity on
               -V             --version     - Print version numbers and exit
               -u             --dump        - Dump key to terminal
               -t             --test        - run self test before proceeding
               -s             --sum         - print sha sum before proceeding
               -f             --force       - force clobbering files
               -w             --weak        - allow weak pass
               -n             --nocrypt     - do not encrypt key (testing only)
               -p val         --pass val    - pass in for key (@file reads pass from file)
               -m name        --keyname nm  - user legible key name
               -d desc        --desc  desc  - key description
               -c name        --creator nm  - override creator name (def: logon name)
```

Most of the arguments are self-explanatory. Please remember when you provide a password on the command line, it may be visible by others. If you prefix the password with the at sign '@', the utility will read the password from a file. Although this might be a little more secure but still depending on a file that can only be read by the user of the utility.

The key generation itself contains version information, so future versions of this program can examine the generator origin. The file formats itself is flexible enough to accommodate changes without versioning.

2.) dibakeyinfo

With this utility one can check the generated key. Here's a sample output of the key we generated when we made the previous screenshot (above).



This utility will show various information about a DIBA key. The creator of the key with the correct password has access to every detail contain within the keys. Without password only public key information, and the public portion of the private key info is available.

```
Usage: dibakeyinfo [options] keyfile

Where keyfile is the basename for .key .pub files. Individual files may be specified also.
Example: file.key or file.pub

Options can be:

               -p val         --pass val     - pass in for key (@file for pass file)
               -v             --verbose      - Verbosity on
               -t             --test         - gcrypt self Test on
               -c             --check        - check key signature(s)
               -i             --pinfo        - print key info
               -k             --pkey         - print public / private key
               -l             --plen         - print key length
               -n             --nocrypt      - do not decrypt key


Option with argument needs one option per command line item.
```

The same warnings about passwords apply here, as was the case of the key generator. Two additional functions are included here: the ability to check key integrity, and the ability to print the key information, and the keys themselves to the screen (stdout). Please note, that if you enter the password for the private key and instruct the program to print the key, your secret key information becomes visible. Only use these functions entrusted terminals.

3.) dibaencrypt

Encrypt a file or a the standard input (from a pipe). Uses a public key generated by dibakeygen.

```
Encrypt files with Public key

Usage: asencrypt [options] keyfile
Options can be:
               -i <filename>  --infile <filename>     - input file name
               -o <filename>  --outfile <filename>    - output file name
               -k <filename>  --keyfile <filename>    - key file name
               -V             --version     - Print version numbers and exit
               -r             --stdin                 - use stdin as input
               -v             --verbose               - Verbosity on
               -d             --dump                  - Dump buffers
               -t             --test                  - test on
               -v             --ppub              - print public key

               -?             --help        - displays this help
               -h             --help        - displays this help
```

4.) dibadecrypt

Decrypt a file or standard input (from a pipe). Uses a public key generated by dibakeygen.

```
Decrypt file with Private key.

Usage: asdecrypt [options] keyfile
Options can be:
               -i <filename>  --infile <filename>     - input file name
               -o <filename>  --outfile <filename>    - output file name
               -p             --pass                - pass in for key (testing only)
               -k <filename>  --keyfile <filename>    - key file name
               -r             --stdin                 - use stdin as input
               -v             --verbose               - Verbosity on
               -V             --version       - Print version numbers and exit
               -t             --test                  - test on
               -n             --nocrypt               - do not decypt private key
               -x             --printpub              - print public key

               -?             --help        - displays this help
               -h             --help        - displays this help
Option with argument needs one option per command line item.
```

Checking and securing:

The utilities have a builtin check sum facility. Use the -s option to check it. Please note, that if you build the diba transport tools, the sum will change. These are the sums at the time of writing (Aug 2017)

dibakeygen.exe -s
Executable sha hash: 'IvDom+OOUOfl/zMlEjgHr1EuA4LQsoCxkddTeiEgAU0='

dibakeyinfo.exe -s
Executable sha hash: '23y5wUmXqNnmtPXRDfXiY7EvknE7T2ZNlFu2Et4jKQM='

dibaencrypt.exe -s
Executable sha hash: 'ngpQQPewmnLyDoAe0S0tk50gryA0fQ9IJHVKKT1S2VY='

dibadecrypt.exe -s
Executable sha hash: 'mTddjulAzX4exQQesW+Bvef74urNlfT1Py3O3hLZQiw='

Links and URLs:

The source(s) and executables for public/private key gen / key info / encryption / decryption :

https://github.com/pglen/diba

in the 'transport' sub directory