NAME

pcp2-generate-rsakeys − RSA key generation for the Pure Crypto Project

SYNOPSIS

pcp2-generate-rsakeys [-size numrandombytes] enc | sig

DESCRIPTION

pcp2-generate-rsakeys reads a hash key (used for SDLH) and additionally generates two large strong prime numbers P and Q that form a modulus for either an encryption key or a signing key.
Before **pcp2** can be used, two individual RSA keys must be generated, an encryption key and a signing key. It is a precondition that a hash key "./hashkey" is available that can be incorporated into both of these RSA keys. The program **sdlh-generate-hashkey** can be used to produce a hash key file. When a RSA signing key is generated, the default size ensures that the signing key's modulus is at least 512 bits larger than the hash modulus which is stored in the hash key file.

When a RSA key is generated the public and private part is stored in a file "encryptionkey" or "signingkey" dependent on the first parameter "enc" or "sig". Initially, the private part (the RSA decryption exponent) will be stored in these files unprotected. All pieces of information (except the decryption exponent) will then be used to create a securityhash value that represents all public information about the RSA key including the user ID string. Please note, that the user ID string of any signing key must include the word 'signing'.

A second file "encryptionkey.pub" or "signingkey.pub" holds all public information that can be shared with other people who use **pcp2**.

To protect the private part of a RSA key with a passphrase the program **pcp2-protect-privatekey** can be used.

Both public keys "encryptionkey.pub" and "signingkey.pub" should also be copied into the directory $HOME/.pcp/trusted-keys and both keys should then be signed with the option "-s" to indicate that this public key is regarded as a trusted key. Note, that the RSA key's securityhash is part of the signed key file.

OPTIONS

−size numrandombytes

The number of random bytes taken from /dev/random are used to find a strong prime. For every probable prime P that can be found, the value (P-1)/2 is checked. If this value is also prime, P is regarded as a strong prime number.
The minimum number of random bytes differ for encryption and signing keys. If larger prime numbers than the default values (1440 bits for encryption keys and 2150 bits for signing keys) are required, the size of the RSA modulus can be changed in a range specific to the key type.

NOTES

Full documentation https://senderek.ie/pcp2

BUGS

Please report bugs to innovation@senderek.ie

AUTHORS

pcp2-generate-rsakeys is written by Ralf Senderek <innovation@senderek.ie>.

COPYRIGHT

SEE ALSO

sdlh, pcp2, pcp2-protect-privatekey, sdlh-generate-hashkey