Rocky Enterprise Linux 9.2 Manual Pages on command 'jose-jwe-enc.1'

$ man jose-jwe-enc.1

JOSE-JWE-ENC(1) JOSE-JWE-ENC(1)

NAME

jose-jwe-enc - Encrypts plaintext using one or more JWK/password

SYNOPSIS

jose jwe enc [-i JWE] -I PT -k JWK [-p] [-r RCP] [-o JWE] [-O CT] [-c]

OVERVIEW

The jose jwe enc command encrypts data using one or more JWK (-k) or password (-p). When specifying more than one JWK or password, decryption will succeed with any one of the provided keys.

A detached JWE can be created by specifying the -O option. In this case, the decoded ciphertext will be written to the output specified and will not be included in the JWE.

If only one key is used (-k or -p), the resulting JWE may be output in JWE Compact Serialization by using the -c option.

This command uses a template based approach for constructing a JWE. You can specify templates of the JWE itself (-i) or for the JWE Recipient Object (-r). Attributes specified in either of these templates will appear unmodified in the output. One exception to this rule is that the

JWE Protected Header should be specified in its decoded form in the template. This command will automatically encode it as part of the encryption process.

If you specify a JOSE Header Parameter (via either the -i or -r options) that affects the construction of the JWE, this command will attempt to behave according to this parameter as if it were configuration. For example, specifying the "zip" parameter in the JWE Protected Header will cause the plaintext to be compressed before encryption. Currently, jose will modify its behavior for the "alg", "enc" and "zip" JOSE Header Parameters (see RFC 7516 Section 4.1.3), as well as the algorithm-specific parameters for the algorithms we support (see RFC 7518 Section 4).

However, it is not necessary to provide any templates: jose jwe enc will automatically fill in the "alg" and "enc" parameters by inferring the correct algorithms from the provided input keys (JWK or password). Therefore, the -i and -r options should generally be used for providing extended JWE metadata.

OPTIONS

- -i JSON, --input=JSON : Parse JWE from JSON
- -i FILE, --input=FILE : Read JWE from FILE
- -i -, --input=-: Read JWE from standard input
- -I FILE, --detached=FILE : Read decoded ciphertext from FILE
- -I -, --detached=- : Read decoded ciphertext from standard input
- -r FILE, --recipient=FILE : Read JWE recipient template from FILE
- -r -, --recipient=- : Read JWE recipient template from standard input
- -k FILE, --key=FILE : Read JWK(Set) from FILE
- -k -, --key=- : Read JWK(Set) from standard input
- -p, --password : Prompt for an encryption password
- -o FILE, --output=FILE : Write JWE to FILE
- -o -, --output=- : Write JWE to stdout (default)
- -O FILE, --detach=FILE : Detach ciphertext and decode to FILE
- -O -, --detach=- : Detach ciphertext and decode to standard output

? -c, --compact : Output JWE using compact serialization

EXAMPLES

Encrypt data with a symmetric key using JWE JSON Serialization:

$ jose jwk gen -i '{"alg":"A128GCM"}' -o key.jwk

$ jose jwe enc -I msg.txt -k key.jwk -o msg.jwe

Encrypt data with a password using JWE Compact Serialization:

$ jose jwe enc -I msg.txt -p -c -o msg.jwe

Please enter an encryption password:

Please re-enter the previous password:

Compress plaintext before encryption:

$ jose jwe enc -i '{"protected":{"zip":"DEF"}}' ...

Encrypt with two keys and two passwords: $ jose jwk gen -i {"alg":"ECDH-ES+A128KW"} -o ec.jwk $ jose jwk gen -i {"alg":"RSA1_5"} -o rsa.jwk $ jose jwe enc -I msg.txt -p -k ec.jwk -p -k rsa.jwk -o msg.jwe Please enter a password: Please re-enter the previous password: Please enter a password: Please re-enter the previous password:

AUTHOR

Nathaniel McCallum <npmccallum@redhat.com>

SEE ALSO

jose-jwe-dec(1), jose-jwe-fmt(1)

08/09/2021 JOSE-JWE-ENC(1)