OpenSSL - pkeyutl

Utilitaire de manipulation de clé publique

OPTIONS

-in filename Spécifie le nom du fichier d'entrée

-out filename Spécifie le fichier de sortie

-inkey file fichier clé d'entrée, par défaut devrait être une clé privée.

-keyform PEM|DER Format de clé PEM, DER ou ENGINE

-passin arg Source du mot de passe de la clé en entrée

-peerkey file Fichier de clé pair, utilisé par les opérations de dérivation de clé

-peerform PEM|DER Format du fichier de clé pair PEM, DER ou ENGINE

-engine id pkeyutil va tenter d'obtenir une référence fonctionnelle de ce moteur.

-pubin Le fichier d'entrée est une clé publique

-certin L'entrée est un certificat contenant une clé publique

-rev Renverse l'ordre du tampon d'entrée. Utile pour certains libraires comme CryptoAPI qui présente l'entrée au format little endian

-sign Signe les données d'entrée et sort le résultat signé, requière une clé privée.

-verify Vérifie les données en entrée avec le fichier de signature et indique si l'opération à réussie ou non

-verifyrecover Vérifie les données entrée et sort les données récupérées

-encrypt Chiffre les données en entrée en utilisant un clé publique

-decrypt Déchiffre les données en entrée en utilisant une clé privée

-derive Dérive une clé partagée en utilisant une clé paire

-hexdump Dump en hexa les données sorties

-asn1parse asn1parse les données en sortie. Utilisé avec -verifyrecover

Notes

Les opérations et options dépendent de l'algorithme de clé et de son implémentation. Tous les algorithmes supportent l'option **digest :alg** qui spécifie le digest à utiliser pour les opérations de signature et vérification.

RSA

RSA supporte les opérations de chiffrement, déchiffrement, signature et vérification.

-rsa_padding_mode :mode Définis le padding RSA. (pkcs1, sslv23, none oaep, x931 et pss). oeap supporte uniquement le chiffrement et déchiffrement. pss support uniquement la signature et la vérification. rsa_pss_saltlen :len pour le mode pss, spécifie la longueur du salt. (Valeurs spéciales : -1 - le salt est à la longueur du digest, -2 - valeur maximum permise).

DSA

DSA support les opérations de signature et de vérification uniquement.

DH

DH supporte uniquement les opérations de dérivation

EC

EC supporte les opérations de signature, de vérification et de dérivation. La signature et vérification utilise ECDSA et la dérivation ECDH.

Exemples

Signer des données en utilisant une clé privée :
openssl pkeyutl -sign -in file -inkey key.pem -out sig
Récupérer des données signées :
openssl pkeyutl -verifyrecover -in sig -inkey key.pem
Vérifier la signature :
openssl pkeyutl -verify -in file -sigfile sig -inkey key.pem
Signer les données en utilisant un message digest :
openssl pkeyutl -sign -in file -inkey key.pem -out sig -pkeyopt digest :sha256
Dériver une clé secrète partagée :
openssl pkeyutl -derive -inkey key.pem -peerkey pubkey.pem -out secret