NAME

X509_digest, X509_digest_sig, X509_CRL_digest, X509_pubkey_digest, X509_NAME_digest, X509_REQ_digest, PKCS7_ISSUER_AND_SERIAL_digest - get digest of various objects

SYNOPSIS

    #include <openssl/x509.h>

    int X509_digest(const X509 *data, const EVP_MD *type, unsigned char *md,
            unsigned int *len);
    ASN1_OCTET_STRING *X509_digest_sig(const X509 *cert,
                      EVP_MD **md_used, int *md_is_fallback);

    int X509_CRL_digest(const X509_CRL *data, const EVP_MD *type, unsigned char *md,
            unsigned int *len);

    int X509_pubkey_digest(const X509 *data, const EVP_MD *type,
              unsigned char *md, unsigned int *len);

    int X509_REQ_digest(const X509_REQ *data, const EVP_MD *type,
            unsigned char *md, unsigned int *len);

    int X509_NAME_digest(const X509_NAME *data, const EVP_MD *type,
              unsigned char *md, unsigned int *len);

    #include <openssl/pkcs7.h>

    int PKCS7_ISSUER_AND_SERIAL_digest(PKCS7_ISSUER_AND_SERIAL *data,
                    const EVP_MD *type, unsigned char *md,
                    unsigned int *len);

DESCRIPTION

X509_digest_sig() calculates a digest of the given certificate *cert* using the same hash algorithm as in its signature, if the digest is an integral part of the certificate signature algorithm identifier. Otherwise, a fallback hash algorithm is determined as follows: SHA512 if the signature algorithm is ED25519, SHAKE256 if it is ED448, otherwise SHA256. The output parameters are assigned as follows. Unless *md_used* is NULL, the hash algorithm used is provided in **md_used* and must be freed by the caller (if it is not NULL). Unless *md_is_fallback* is NULL, the **md_is_fallback* is set to 1 if the hash algorithm used is a fallback, otherwise to 0.

X509_pubkey_digest() returns a digest of the DER representation of the public key in the specified

X509 *data* object.

All other functions described here return a digest of the DER representation of their entire *data* objects.

The *type* parameter specifies the digest to be used, such as **EVP_sha1()**. The *md* is a pointer to the buffer where the digest will be copied and is assumed to be large enough; the constant **EVP_MAX_MD_SIZE** is suggested. The *len* parameter, if not NULL, points to a place where the digest size will be stored.

RETURN VALUES

    X509_digest_sig() returns an ASN1_OCTET_STRING pointer on success, else NULL.

All other functions described here return 1 for success and 0 for failure.

SEE ALSO

    EVP_sha1(3)

HISTORY

    The **X509_digest_sig**() function was added in OpenSSL 3.0.

COPYRIGHT

    Copyright 2017-2023 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.