NAME

CURLOPT_SSL_SIGNATURE_ALGORITHMS − signature algorithms to use for TLS

SYNOPSIS

```
#include <curl/curl.h>

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSL_SIGNATURE_ALGORITHMS, char *list);
```

DESCRIPTION

Pass a char pointer, pointing to a null−terminated string holding the list of signature algorithms to use for the TLS connection. The list must be syntactically correct, it consists of one or more signature algorithm strings separated by colons.

A valid example of a signature algorithms list with OpenSSL is:
"DSA+SHA256:rsa_pss_pss_sha256"

The application does not have to keep the string around after setting this option.

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to disable its use again.

Works with OpenSSL and its BoringSSL fork (added in 8.14.0).

DEFAULT

NULL, use built−in list

PROTOCOLS

This functionality affects all TLS based protocols: HTTPS, FTPS, IMAPS, POP3S, SMTPS etc.

This option works only with the following TLS backends: OpenSSL

EXAMPLE

```
int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode result;
    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
    curl_easy_setopt(curl, CURLOPT_SSL_SIGNATURE_ALGORITHMS,
                     "DSA+SHA256:rsa_pss_pss_sha256");
    result = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
  }
}
```

HISTORY

OpenSSL support added in 8.14.0.

AVAILABILITY

Added in curl 8.14.0

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non−zero means an error occurred, see *libcurl−errors(3)*.

SEE ALSO

CURLOPT_SSLVERSION(3), **CURLOPT_SSL_CIPHER_LIST**(3), **CURLOPT_SSL_EC_CURVES**(3), **CURLOPT_USE_SSL**(3)