



Zertifikatsanfrage ECDSA

Damit über die interne CA ein ECDSA Request signiert werden kann, muss zuvor eine geeignete Vorlage erstellt und veröffentlicht werden.

Eigenschaften von ECDH256WEB

Schlüsselnachweis	Antragstellename	Ausstellungsvoraussetzungen
Abgelöste Vorlagen	Erweiterungen	Sicherheit
Allgemein	Kompatibilität	Anforderungsverarbeitung
		Kryptografie

Anbieterkategorie: Schlüsselspeicheranbieter

Name des Algorithmus: ECDH_P256

Minimale Schlüsselgröße: 256

Auswählen der für Anforderungen verwendbaren Kryptografieanbieter

Verwendung aller auf dem Computer des Antragstellers verfügbaren Anbieter für Anforderungen möglich

Für Anforderungen muss einer der folgenden Anbieter verwendet werden:

Anbieter:

- Microsoft Software Key Storage Provider

Anforderungshash: SHA256

Alternatives Signaturformat verwenden

OK Abbrechen Überehmen Hilfe

Die Vorlage ist veröffentlicht.

Konsole1 - [Konsolenstamm]\Zertifizierungsstelle (Lokal)\ISSUINGCA\Zertifikatvorlagen

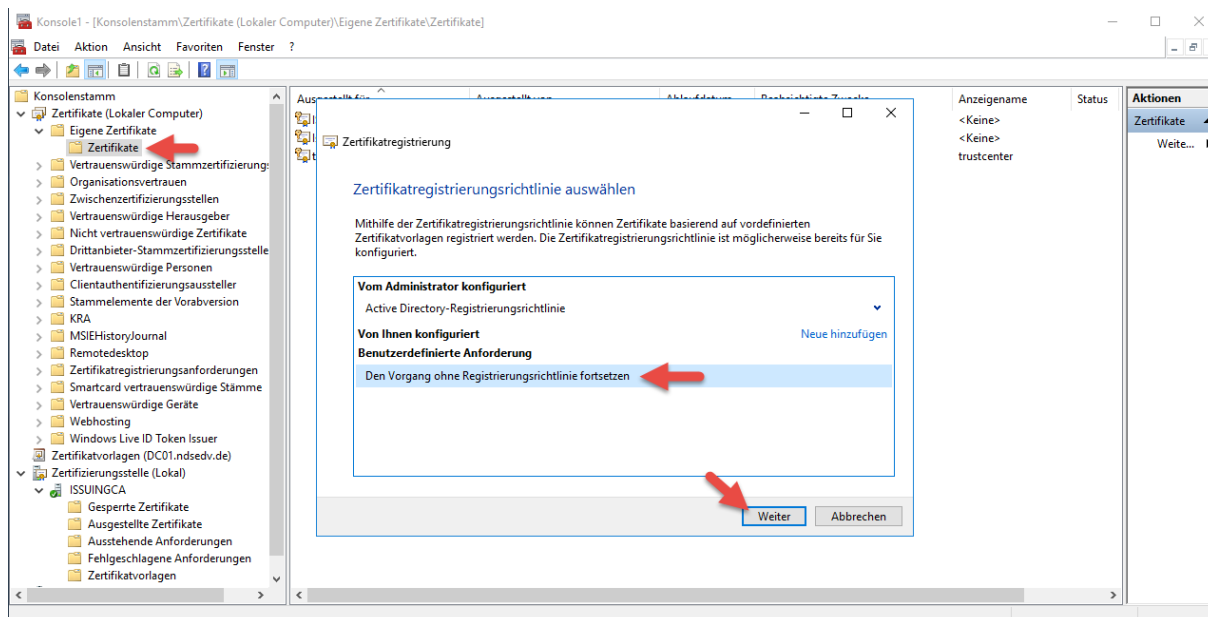
Name	Beabsichtigter Zweck
ECDH256WEB	Serverauthentifizierung
DSA256WEB	Serverauthentifizierung
WEBDSA384	Serverauthentifizierung
RDPSHA256	Serverauthentifizierung
Key Recovery Agent	Key Recovery Agent
WEBSHA256V3	Serverauthentifizierung
WebserverV3	Serverauthentifizierung
KRA2	Key Recovery Agent
DCV4	Serverauthentifizierung, Clientauthentifizierung
ComputerV3	Serverauthentifizierung, Clientauthentifizierung
CodesignaturV2	Codesignatur
BenutzerV2	Verschlüsselndes Dateisystem, Sichere E-Mail, Clientauthentifizierung



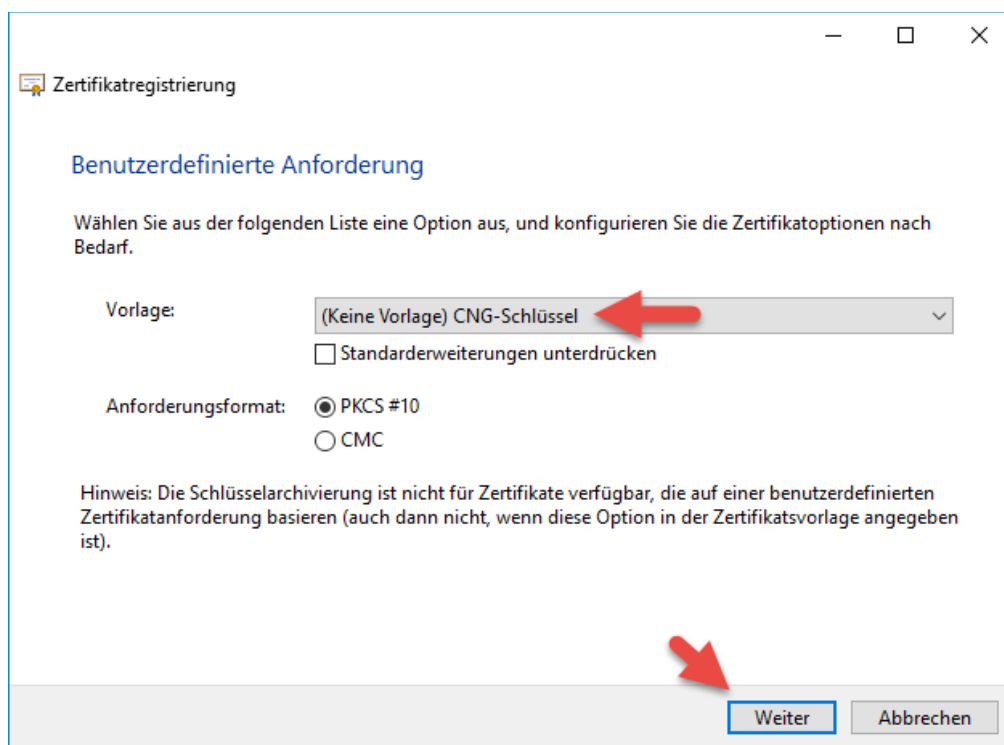
Zertifikatsanfrage ECDSA

Die Zertifikatsanfrage erstellen wir wie folgt:

Die Anfrage erfolgt ohne Registrierungsrichtlinie...



...und ohne Vorlage.





Zertifikatsanfrage ECDSA

Klicken auf Details und dann auf Eigenschaften.

Zertifikatsregistrierung

Zertifikatsinformationen

Klicken Sie auf "Weiter", um die bereits für diese Vorlage ausgewählten Optionen auszuwählen, oder klicken Sie auf "Details", um die Zertifikatanforderung anzupassen, und klicken Sie anschließend auf "Weiter".

Benutzerdefinierte Anforderung STATUS: Verfügbar Details ^

Die folgenden Optionen beschreiben die Verwendung und den Gültigkeitszeitraum, die auf diesen Zertifikattyp zutreffen:

- Schlüsselverwendung:
- Anwendungsrichtlinien:
- Gültigkeitszeitraum (Tage):

Eigenschaften

Weiter **Abbrechen**

Das Zertifikat ist wie gewohnt mit den Informationen zu füllen die für die Bestätigung der Identität wichtig sind.

Zertifikateigenschaften

Allgemein **Antragsteller** Erweiterungen Privater Schlüssel

Der Antragsteller eines Zertifikats ist der Benutzer oder Computer, für den das Zertifikat ausgestellt ist. Geben Sie Informationen über die zulässigen Antragstellernamen und alternative Namenswerte ein, die in einem Zertifikat verwendet werden dürfen.

Zertifikatsantragsteller
Der das Zertifikat empfangende Benutzer oder Computer

Antragstellername:

Typ: Allgemeiner Name Hinzufügen >

Wert: < Entfernen

Alternativer Name:

Typ: DNS Hinzufügen >

Wert: < Entfernen

CN=issuingca

DNS
issuingca
issuingca.ndsedv.de

OK Abbrechen Übernehmen



Zertifikatsanfrage ECDSA

Über den Reiter Privater Schlüssel gelangen wir nun zu der Auswahl der Kryptographischen Anbieter. An dieser Stelle wählen wir ECDSA_P256 aus. Höher können wir nicht gehen, da die CA nur bis 256 Bit signieren kann.

Zertifikateigenschaften

Allgemein Antragsteller Erweiterungen Privater Schlüssel

Kryptografiedienstanbieter
Bei einem CSP handelt es sich um ein Programm, das ein öffentlich-privates Schlüsselpaar erzeugt, das in vielen zertifikatbezogenen Prozessen Verwendung findet.

Wählen Sie den Kryptographischen Dienstanbieter (CSP) aus:

- ECDSA_x962P256v1, Microsoft Software Key Storage Provider
- ECDSA_P256, Microsoft Software Key Storage Provider
- ECDSA_P384, Microsoft Software Key Storage Provider
- ECDSA_P521, Microsoft Software Key Storage Provider
- RSA, Microsoft Passport Key Storage Provider
- RSA, Microsoft Smart Card Key Storage Provider

Alle CSPs anzeigen

Schlüsseloptionen
Legen Sie die Schlüssellänge und die Exportoptionen für den privaten Schlüssel fest.

Schlüsselgröße: 256

Privaten Schlüssel exportierbar machen

OK Abbrechen Übernehmen

Speichern die Zertifikatsanfrage und übergeben diese zum Signieren an die CA.

Zertifikatregistrierung

Wohin möchten Sie die Offlineanforderung speichern?

Möchten Sie eine Kopie der Zertifikatanforderung speichern, oder die Anfrage später verarbeiten, speichern Sie die Anfrage auf der Festplatte oder auf mobilen Speichermedien. Geben Sie Standort und Namen der Zertifikatanforderung ein, und klicken Sie anschließend auf "Fertig stellen".

Dateiname:
C:\Users\NDS\Desktop\SUBCA_ECDSA.req Durchsuchen...

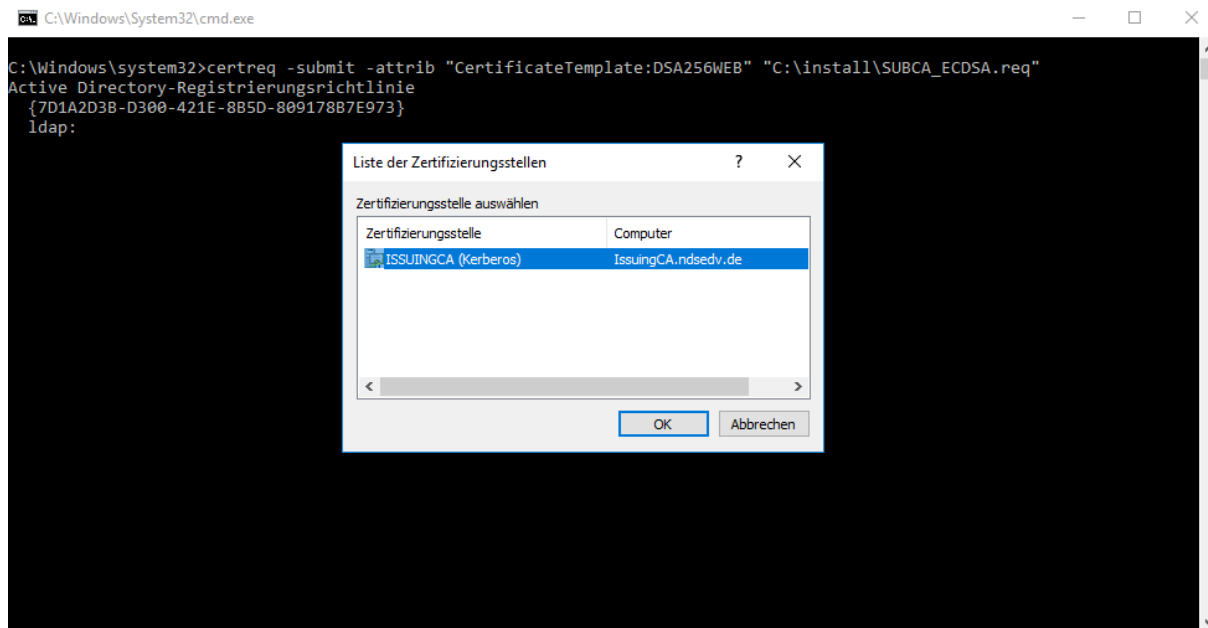
Dateiformat:
 Base 64
 Binär

Fertig stellen Abbrechen

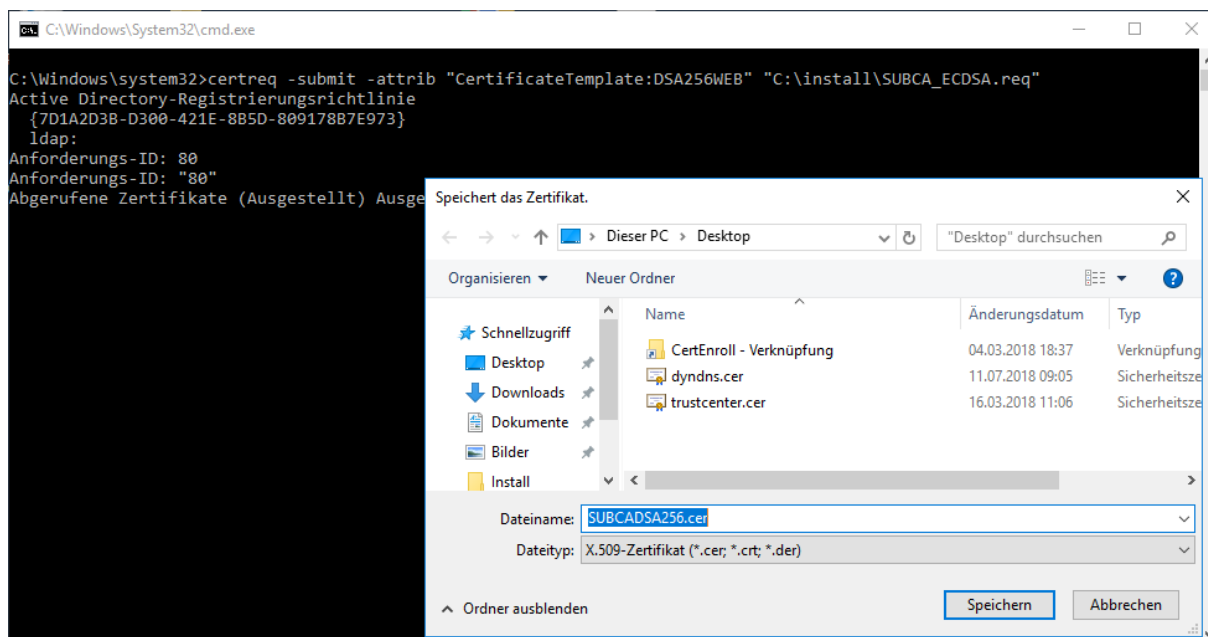


Zertifikatsanfrage ECDSA

Die Zertifikatsanfrage wird nun über die interne CA des Administrators geprüft und signiert.



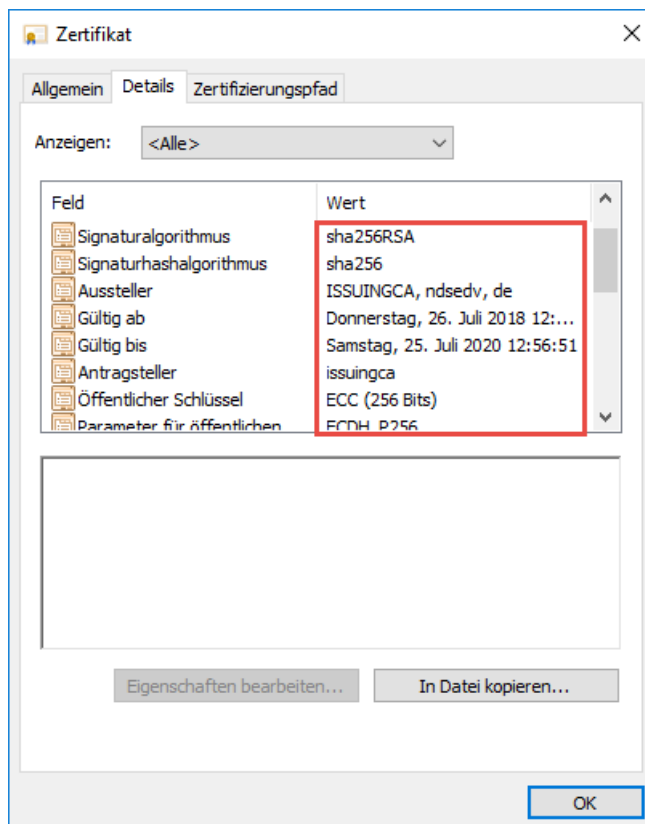
Speichern das Zertifikat ab.



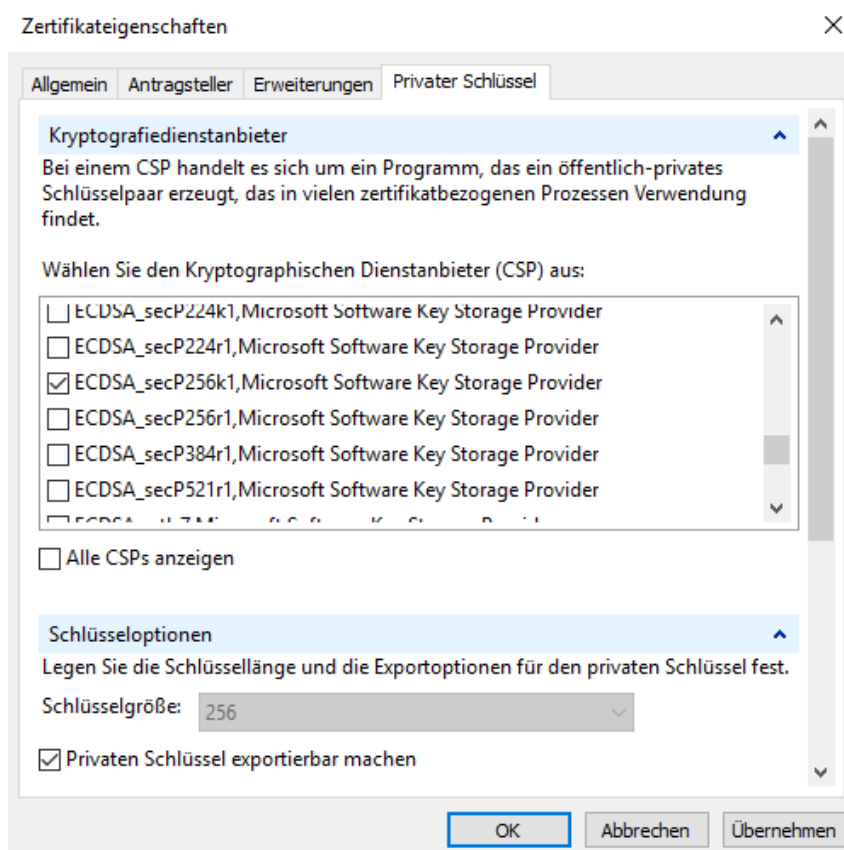


Zertifikatsanfrage ECDSA

Schauen wir uns nun die Eigenschaften des Zertifikats im Bereich Signatur und Öffentlicher Schlüssel an.



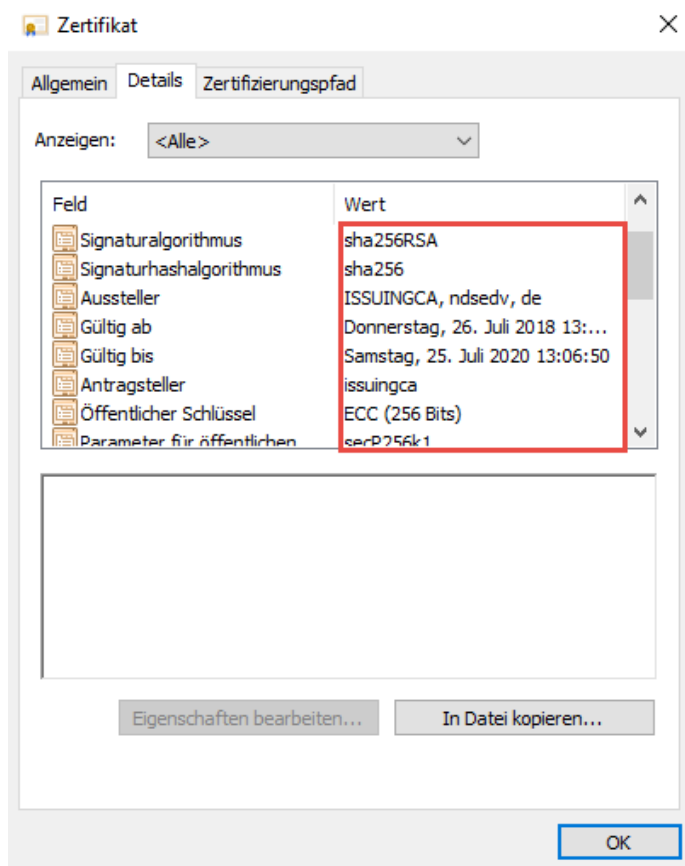
Alternativer Provider:





Zertifikatsanfrage ECDSA

Hier sehen wir das der Öffentliche Schlüssel mit der Signatur ECC (256) Bits ausgestattet ist.





Zertifikatsanfrage ECDSA

Die beiden ausgestellten Zertifikate sind nun für den Einsatz von Forward Secrecy geeignet und unterstützen alle TLS bis hin zur Versionen 1.3.

Kann mein System denn nun was mit den Zertifikaten anfangen?

Das Ganze prüfen wir mit diesem Befehl:

- CertUtil.exe -DisplayEccCurve

Das getestete System unterstützt die Curve secP256k1

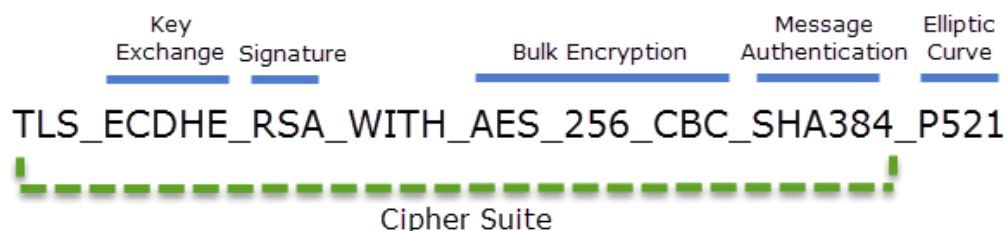
```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 10.0.17134.191]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>CertUtil.exe -DisplayEccCurve
Microsoft SSL Protocol Provider:
-----
Curve Name                Curve OID                Länge des öffentlichen SchlüsselsCurveType    EccCurveFlags
-----
curve25519                2.253.1.3.3.1.1.1.1.1  255                29                0xa
nistP256                  1.2.840.10045.3.1.7    256                23                0x7
nistP384                  1.3.132.0.34           384                24                0x7
brainpoolP256r1          1.3.36.3.3.2.8.1.1.7   256                26                0x7
brainpoolP384r1          1.3.36.3.3.2.8.1.1.11 384                27                0x7
brainpoolP512r1          1.3.36.3.3.2.8.1.1.13 512                28                0x7
nistP192                  1.2.840.10045.3.1.1    192                19                0x7
nistP224                  1.3.132.0.33           224                21                0x7
nistP521                  1.3.132.0.35           521                25                0x7
secP160k1                 1.3.132.0.9            160                15                0x7
secP160r1                 1.3.132.0.8            160                16                0x7
secP160r2                 1.3.132.0.30           160                17                0x7
secP192k1                 1.3.132.0.31           192                18                0x7
secP192r1                 1.2.840.10045.3.1.1    192                19                0x7
secP224k1                 1.3.132.0.32           224                20                0x7
secP224r1                 1.3.132.0.33           224                21                0x7
secP256k1                 1.3.132.0.10           256                22                0x7
secP256r1                 1.2.840.10045.3.1.7    256                23                0x7
secP384r1                 1.3.132.0.34           384                24                0x7
secP521r1                 1.3.132.0.35           521                25                0x7
```

ECDHE = ist die Abkürzung für Elliptic Curve Diffie-Hellmann und bezeichnet das DH-Verfahren auf Basis elliptischer Kurven

ECDSA = ist die Abkürzung für Elliptic Curve DSA (Digital Signature Algorithm) und dieses lässt sich auch mit elliptischen Kurven durchsetzen, ist aber noch langsamer als RSA.

Jede Cipher Suite besteht aus einem Schlüsselaustauschalgorithmus, einem Cipher Algorithmus und einem MAC (Hash) Algorithmus. Message Authentication (MAC) sind Algorithmen die Hashes und Signaturen erzeugen um die Integrität einer Nachricht sicherzustellen.



<https://www.der-windows-papst.de/2018/03/07/ecccurve-unterstuetzung-windows-10-2016/>