DIGITAL SIGNATURE ENCRYPTION

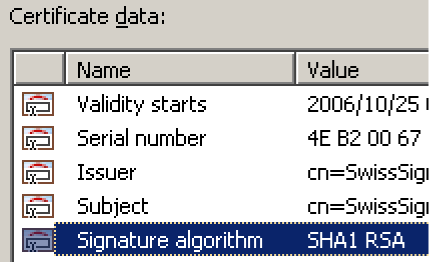
Message digest creation algorithms

A document is digested (hashed) using the algorithm specified by aSignHash or tSignHash (Defaults in **Red**)

	SubFilter value specified by aSignFormat, seed value, JavaScript, or PubSec handler				
V.	**adbe.pkcs7.detached**	**adbe.pkcs7.sha1**	**adbe.x509.rsa.sha1**	**ETSI.CAdES.detached**	**PDF #**
10.0	MD5, SHA1, **SHA256**, SHA384, SHA512, RIPEMD160	SHA1	Same as adbe.pkcs7.detached .	Same as adbe.pkcs7.detached	v.1.7
9.0	MD5, **SHA1**, **SHA256 (9.1)**, SHA384, SHA512, RIPEMD160			N/A	v.1.7
8.0	MD5, **SHA1**, SHA256, SHA384, SHA512, RIPEMD160				v.1.7
7.0	MD5, **SHA1**, SHA256				v.1.6
6.0	MD5, SHA1				v.1.3
4-5.0	MD5, SHA1				v.1.3

Digital signature algorithms

The digest above is encrypted using a DSA, RSA, or Elliptic Curve algorithm and using either a key of a certain length for DSA & RSA, or a number derived by coordinates of a point on a curve (which is much smaller than the fixed key size, which makes ECC much faster). The encryption algorithm used is the one specified by the signature algorithm in the certificate.



Certificate data:

Name	Value
Validity starts	2006/10/25
Serial number	4E B2 00 67
Issuer	cn=SwissSig
Subject	cn=SwissSig
Signature algorithm	SHA1 RSA

Acrobat uses the digital ID's public-key certificate (PKC) for both signature creation & validation, and thus has to understand the digest algorithm used to create the PKC so it can validate its signature. The PKC is signed by its issuer, and that signature has to be validated just as the signature over the document (PDF file) has to be validated.

Elliptic Curve Credential (ECC) usage: Only 11.x products support using ECC credentials for signing, validating an ECC signature, and receiving a document with an ECC credential. A document can be encrypted for mixed recipients on 9.x products and later (where recipients may have RSA credentials or ECC credentials). Products prior to 9.x only support RSA.

The table below listed is the minimum and recommended hash size to be used with each curve. The list includes the NIST Recommended Elliptic Curves defined in FIPS PUB 186-4: Digital Signature Standard (DSS) issued July 2013.

Version	**Encryption algorithms**	**PDF #**	**Digest creation compatibility**
11.0	RSA and DSA SHA1 up to 4096-bit ECDSA (Elliptic Curve) with specific named curves (NIST): • P256 with digest algorithm SHA256 • P384 with digest algorithm SHA384 • P521with digest algorithm SHA512 • Not supported on Windows (MSCAPI) and Mac (Keychain): P-192 (secp192r1), P-224 (secp224r1), P-256 (secp256r1), P-384 (secp384r1), P-521 (secp521r1), B-163 (sect163r2), K-163 (sect163k1), B-233 (sect233r1), K-233 (sect233k1), B-283 (sect283r1), K-283 (sect283k1), B-409 (sect409r1), K-409 (sect409k1), B-571 (sect571r1), K-571 (sect571k1)	v 1.7	DSA only supports SHA1 and adbe.pkcs7.detached. Only available with adbe.pkcs7.detached and ETSI.CAdES.detached
8.0-10.0	RSA up to 4096-bit and DSA SHA1 up to 4096-bit	v.1.7	DSA only supports SHA1 and adbe.pkcs7.detached. RSA supports all algorithms and signature types (subFilter values).
7.0	RSA up to 4096-bit and DSA SHA1 up to 4096-bit	v.1.6	
6.0	RSA up to 4096 bit	v.1.5	
4-5.0	RSA up to 1024 bit	v.1.3	

Acrobat supports the named curves recommended by NIST (see FIPS PUB 186-2 or later) which RSA also supports. The following curves are **not** supported by Acrobat or RSA:

- Using SHA1 (Acrobat does not support any SHA1 ECDSA curves):
    - secp160k1
    - secp160r1
    - secp160r2
    - BrainpoolP160r1
    - BrainpoolP160t1
- Using SHA224
    - secp192k1
    - BrainpoolP192r1
    - BrainpoolP192t1
    - secp224k1
    - BrainpoolP224r1
    - BrainpoolP224t1
- Using SHA256:
    - secp256k1
    - BrainpoolP256r1
    - BrainpoolP256t1
- Using SHA384:
    - BrainpoolP320r1
    - BrainpoolP320t1
    - BrainpoolP384r1
    - BrainpoolP384t1
- Using SHA512:
    - BrainpoolP512r1
    - BrainpoolP512t1