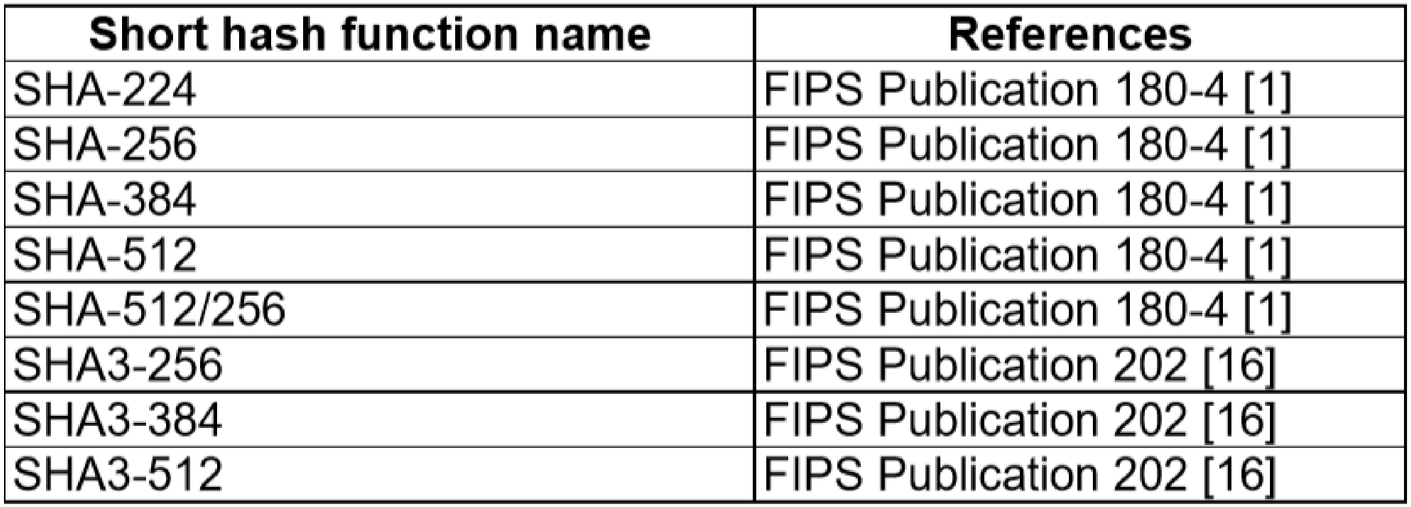
Overview of crypto standards

Zdeněk Říha

Hash functions

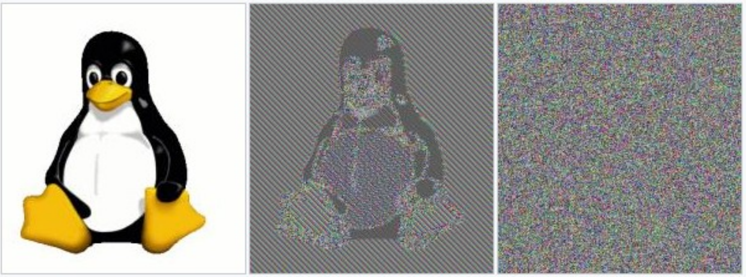
- MD5 (128 bit output) – defined v RFC 1321
- RIPEMD-128/RIPEMD-160 in ISO/IEC 10118-3
- BLAKE2b, BLAKE2s defined in RFC 7693.



Short hash function name	References
SHA-224	FIPS Publication 180-4 [1]
SHA-256	FIPS Publication 180-4 [1]
SHA-384	FIPS Publication 180-4 [1]
SHA-512	FIPS Publication 180-4 [1]
SHA-512/256	FIPS Publication 180-4 [1]
SHA3-256	FIPS Publication 202 [16]
SHA3-384	FIPS Publication 202 [16]
SHA3-512	FIPS Publication 202 [16]

Symmetric crypto

- Modes of operation (FIPS 81)
  - ECB (Electronic Code Book)
  - CBC (Ciper Block Chaining)
  - CFB (Cipher Feedback Mode)
  - OFB (Output Feedback Mode)



Original image

Encrypted using ECB mode

Modes other than ECB result in pseudo-randomness

See: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

- Newer modes of operation
  - CTR (Counter Mode) [FIPS SP 800-38A]
  - CMAC [FIPS SP 800-38B], CCM [FIPS SP 800-38C], GCM [FIPS SP 800-38D], XTS-AES [FIPS SP 800-38E]
  - Other in FIPS SP 800-38F, FIPS SP 800-38G

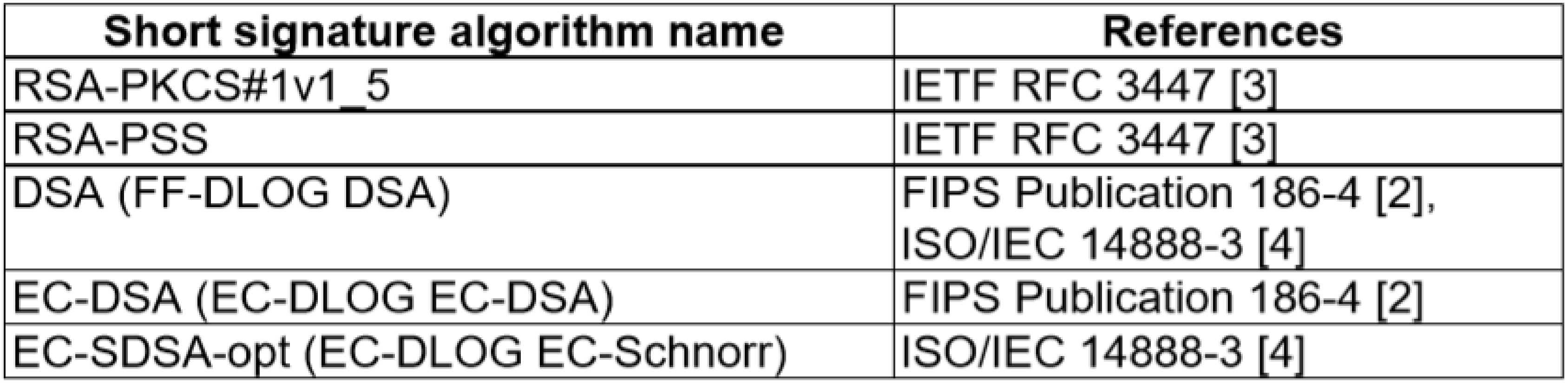
Padding

- **ISO 9797 method 1** padded with values 0x00
  - to remove the padding the length of the original message is needed
- **ISO 9797 method 2** (ISO 7816-4, EMV'96) – first the value 0x80 is added, then bytes of 0x00 are added
  - *PS = '80 00', if 2 bytes are needed*
  - *PS = '80 00 00 00 00 00 00 00', if 0 bytes are needed (full block added)*
- • **PKCS#5** – the padding string is made from value n-(||*M*|| mod n)
  - *for (3)DES n=8, AES n=16*
  - *e.g. PS = 02 02 - if 2 bytes are needed*
  - *e.g. PS = 08 08 08 08 08 08 08 08 – if 0 bytes are needed and n=8 (3DES)*

Symmetric crypto

- DES – defined in FIPS PUB 46 (-1 a -2)
  - key 56 bits, block 64 bits
- 3DES – defined in FIPS PUB 46-3
  - key either 112 or 168 bits, block 64 bits
- AES – (Rijndael), defined v FIPS PUB 197
  - key 128, 192 or 256 bits, block 128 bits

Asymmetric crypto



Short signature algorithm name	References
RSA-PKCS#1v1_5	IETF RFC 3447 [3]
RSA-PSS	IETF RFC 3447 [3]
DSA (FF-DLOG DSA)	FIPS Publication 186-4 [2], ISO/IEC 14888-3 [4]
EC-DSA (EC-DLOG EC-DSA)	FIPS Publication 186-4 [2]
EC-SDSA-opt (EC-DLOG EC-Schnorr)	ISO/IEC 14888-3 [4]

- Certificates X.509
  - ITU-T, ISO/IEC, RFC
- DER / PEM

PKCS

- PKCS#1 – defines RSA encryption
- PKCS#3 – defines Diffie-Hellman protocol
- PKCS#5 – symmetric encryption based on a password
- PKCS#7 – format for digital signatures and asymmetric encryption
- PKCS#8 – defines the private key format
- PKCS#10 – defines format for certificate requests
- PKCS#11 – API for communication with cryptographic tokens
- PKCS#12 – format for storing private keys including public key certificates, all protected by a password
- PKCS#13 – defines encryption based on elliptic curves
- PKCS#15 – defines cryptographic token information format

RSA Padding

- E.g. RSA 2048 bits
  - Modulus *n* is 2048 bits, public exponent *e* usually small
  - Message *m* is 2048 bits in total, usual hash functions provide hashes much shorter. Therefore we need padding.
- BTW No padding needed for DSA and ECDSA

RSA Padding algorithms

- **ANSIX 9.31**
  - 6b bb … bb ba || Hash(M) || 3x cc
    (where x=3 for sha1, x=1 for ripemd160)
- **PKCS#1 v1.5**
  - 00 01 ff … ff 00 || HashAlgID || Hash(M)
- **PSS**
  - 00 || H || G(H) $\oplus$ [salt || 00 … 00] (where H = Hash(salt, M), salt is random, and G is a mask generation function)

Assignments

1. Write a program (in any programming language) that will prepare a padded block for RSA signature with PKCS#1 v1.5 padding. Input is a file and RSA key size; output is the padded octet string (print it in hex). Use SHA-256 as the hash function. Do not use crypto library for the padding itself [5 points].

2. Write a program that will generate 2048 bit DH parameters in DER format. Use any cryptolibrary and any programming language (no shell script). Check whether the optional privateValueLength is included (submit a screenshot). Recommendation: Openssl & C & functions DH_new, DH_generate_parameters_ex, i2d_DHparams_bio. [5 points].

Good luck

- Good luck and good fun while reading the standards
- Email: zriha@fi.muni.cz