

## Password Based Cryptography

### Outline

1. Preface
2. Summary
3. Message Format
  - 3.1. XML Schema
4. Padding
5. Salt Generation
6. Deriving the Iteration Count
7. Initialization Vector
8. Key Derivation Function
9. HMAC
10. Encryption
11. Decryption
12. Multiple File Encryption
13. File Destruction
14. Base64
15. References
16. Test Vectors

### 1.0 Preface

The following document provides a guideline for implementation of password-based (symmetric) cryptography. The goal of this guideline is to provide a relatively simple and secure implementation. The document follows the PKCS#5 v2 standard in all aspects except for the message format. The message is instead stored in XML rather than ASN.1.

#### 1.1 Introduction

Encryption algorithms can be divided into two distinct categories: symmetric and asymmetric. These are also often referred to as conventional and public key encryption algorithms. Conventional encryption uses the same key for both encryption and decryption. On the positive side conventional encryption is much faster, easier to implement and manage. However it is almost impossible to use for communication between two parties without prior contact. The problem is how do you get the encryption key to the other user? This problem is solved by public key encryption. However public key encryption has its own problems as it is very slow and often requires a third party to verify keys. In reality most cryptographic systems use public key encryption for exchanging keys and conventional encryption for data encryption.

There are however many cases where a full cryptographic system is not needed and conventional encryption will do. This often happens when the encrypted data does not need to be exchanged between two users. A good example of this is hard disk encryption. This document focuses on data encryption using conventional encryption and omits public key encryption entirely.

It is surprisingly difficult to encrypt data considering the last two decades of advances in modern cryptography. One would presume that simply using a modern encryption algorithm such as AES or 3DES would be sufficient. Unfortunately this is not the case. There are many small details that need to be considered. For example how do we derive the encryption key? Do we rely on the user to enter binary data,

or do we ask him to enter a pass phrase and later generate a key? If so what algorithm can we use to generate this key? How do we prevent an attacker to simply keep guessing the most likely pass phrases and eventually decrypt the file? The list of these issues goes on.

Fortunately there are well defined and tested standards that we can follow to successfully secure our data. Unfortunately this information spans many separate documents and is hard to collect and follow without any background information. One of the goals of this document is to collect all relevant information into a single source that is easy to follow for even an inexperienced user.

## 2.0 Summary

The system outlined in this document requires an implementation various encryption, message digest algorithms.

A minimum of one of the following encryption algorithms will be needed:

- AES-128
- AES-192
- AES-256

A minimum of one of the following message digest algorithms will be needed:

- SHA-1
- SHA-256
- SHA-512

The document also uses various supporting methods and modes.

- CBC Mode of operation
- HMAC Signature
- Key Derivation Function
- XML Message Format

The end of this document provides various references to the above-mentioned algorithms and supporting methods.

## 3.0 Message Format

The encrypted cipher text as well as the supporting information is encapsulated in XML. The XML tags are not encrypted, only the certain values between them.

Here is a sample XML message:

```
<?xml version="1.0"?>
<EncryptedData>
  <DataID>1</DataID>
  <EncryptionMethod>aes256-cbc</EncryptionMethod>
  <DocumentName>Test.txt</DocumentName>
  <IterationCount>1000</IterationCount>
  <SaltValue>DFEEBEBF</SaltValue>
  <CipherData>
    <CipherValue>DEADBEEF</CipherValue>
  </CipherData>
  <Signature>
    <SignatureMethod>hmac-sha256</SignatureMethod>
    <SignatureValue>cOQGJE3d3fXi1Bfdvr1v6tz/4lt9xGzfnfDPXEv4Q=</SignatureValue>
  </Signature>
</EncryptedData>
```

The only optional value in the message is <DocumentName>; the value of that field can be left blank although the XML tag should always be present.

Some of the values must be encoded in base64 in order to be visible using variants of ASCII and EBCDIC characters. This is done so that the encrypted message can be viewed as text in a web browser, email or even just notepad.

The following XML tags contain values encoded in base64:

```
<SaltValue>  
<CipherValue>  
<SignatureValue>
```

### 3.1 XML Schema (XSD)

The purpose of the XML Schema is to define what elements and types are legal in a given XML document. The XML Schema is usually saved in a separate file with the extension XSD. For those people not familiar with XSD documents the important portions to note are the legal algorithm types for the Encryption and Signature Method fields. The schema file does not have to be attached to the encrypted file; it could be strictly used during development.

```
<?xml version="1.0"?>  
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="1.0">  
  <xsd:element name="EncryptedData">  
    <xsd:complexType>  
      <xsd:sequence>  
        <xsd:element name="DataID">  
          <xsd:simpleType>  
            <xsd:restriction base="xsd:integer">  
              <xsd:minInclusive value="1"/>  
            </xsd:restriction>  
          </xsd:simpleType>  
        </xsd:element>  
        <xsd:element name="EncryptionMethod" type="EncryptionAlgorithm"/>  
        <xsd:element name="DocumentName" type="xsd:string"/>  
        <xsd:element name="IterationCount">  
          <xsd:simpleType>  
            <xsd:restriction base="xsd:integer">  
              <xsd:minInclusive value="0"/>  
            </xsd:restriction>  
          </xsd:simpleType>  
        </xsd:element>  
        <xsd:element name="SaltValue" type="xsd:string"/>  
        <xsd:element name="CipherData">  
          <xsd:complexType>  
            <xsd:sequence>  
              <xsd:element name="CipherValue" type="xsd:string"/>  
            </xsd:sequence>  
          </xsd:complexType>  
        </xsd:element>  
        <xsd:element name="Signature">  
          <xsd:complexType>  
            <xsd:sequence>  
              <xsd:element name="SignatureMethod" type="SignatureAlgorithm"/>  
              <xsd:element name="SignatureValue" type="xsd:string"/>  
            </xsd:sequence>  
          </xsd:complexType>  
        </xsd:element>  
      </xsd:sequence>  
    </xsd:complexType>  
  </xsd:element>  
  <xsd:simpleType name="EncryptionAlgorithm">  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="aes128-cbc"/>  
      <xsd:enumeration value="aes192-cbc"/>  
      <xsd:enumeration value="aes256-cbc"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
  <xsd:simpleType name="SignatureAlgorithm">  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="hmac-sha1"/>  
      <xsd:enumeration value="hmac-sha256"/>  
      <xsd:enumeration value="hmac-sha512"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:schema>
```

```
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

## 4.0 Padding

Due to the fact that most encryption algorithms process data in blocks, we need to make sure that the plain text is a multiple of the encryption algorithm block size. The only requirement for this process is that the action of padding must be reversible during decryption. Therefore the padding values are relevant. The following padding function is outlined in PKCS#5 v2. The padding function will return a padding string **PS** that will be appended to the plaintext in order to make it equal to the block size of the encryption algorithm.

Algorithm	Block Size (bytes)
AES-128	16
AES-192	16
AES-256	16

### Input:

- Plaintext Message Length
- Encryption Algorithm Block Size (in bytes)

### Output:

- Padding String

### Notation:

**PL**: Padding Length (stored as a single byte)

**ML**: Plaintext Message Length

**PS**: Padding String

**AB**: Encryption Algorithm Block Size in bytes

### Steps:

1. Determine the Padding Length by getting the remaining bytes after dividing the plaintext length by the block length.

$$PL = AB - (ML \text{ mod } AB)$$

2. Create a padding string that will hold PL number of bytes.

$$PS(PL)$$

3. Assign the value PL to each item in PS.

$$PS(1 \dots PL) = PL$$

4. Return **PS**.

### Examples in Hex

AB=8, ML=15 then PS=01

AB=8, ML=14 then PS=0202

AB=8, ML=16 then PS=0808080808080808

## 5.0 Salt Generation

The salt is an 8 byte random number. There are two ways of generating the Salt.

1. By using a cryptographic pseudo random number generator such as Yarrow or Fortuna.
2. By computing the salt using the Key Derivation Function

$$S = KDF(P, M)$$

Where **P** is the pass phrase and **M** is the plaintext message.

Out of the above steps, step number 1 is the preferred method of generating the salt; however in cases where the pseudo random number generator is not available procedure number 2 will provide sufficient security.

**Warning:** Do not use the compiler build in Rand() function to calculate the salt.

## 6.0 Deriving the Iteration Count

If not specified the default iteration count is set to 1000.

The iteration count is highly dependent on the environment and implementation needs. The higher the iteration count the longer it will take the attacker to perform a dictionary attack on the cipher text. However if the iteration count is too high it could make the decryption of a single file too slow to be practical. To add to the dilemma most PCs vary in speeds. What could be a perfectly reasonable iteration count on one PC could bring another one down to a halt. If in doubt use the default iteration count of 1000.

## 7.0 Initialization Vector

The size of the IV is equal to the block size of the specified encryption algorithm.

Algorithm	IV Size (bytes)
AES-128	16
AES-192	16
AES-256	16

There are two ways of generating the IV.

1. By using a cryptographic pseudo random number generator such as Yarrow or Fortuna.
2. By computing the salt using the Key Derivation Function

$$IV = KDF(P, M)$$

Where **P** is the pass phrase and **M** is the plaintext message.

Out of the above steps, step number 1 is the preferred method of generating the IV; however in cases where the pseudo random number generator is not available procedure number 2 will provide sufficient security.

The IV is stored as the first block of cipher text.

**Warning:** Do not use the compiler build in Rand() function to calculate the IV.

## 8.0 Key Derivation Function

The key derivation function produces a key from a pass phrase. The key size produced by the Key Derivation Function depends on the encryption algorithm used in the encryption. The following Key Derivation Function is outlined in PKCS#5 v2

Algorithm	Key Size (bytes)
AES-128	16
AES-192	24
AES-256	32

### Input

- Pass-phrase
- Salt
- Iteration Count
- Derived Key Size (bytes)

### Output

- Derived Key

### Notation

**HMAC**: Keyed-Hashed Message Authentication Code function described in section 11.0

**HLEN**: The length in bytes of the HMAC function output. Equal to the length of the Message Digest Algorithm used in the HMAC.

**P**: Pass phrase

**S**: Salt

**C**: Iteration Count

**DK**: Derived Key

**DKLEN**: Derived Key Size

### Operators

**||**: Concatenation of 2 strings

**XOR**: Exclusive or bitwise operator.

**INT(I)** is a four-byte encoding of the integer I, most significant byte first

**⌈ ⌋**: Round Up to ex: 1.5=2, 1.3=1, 1.7=2, 1=1

### Steps

1. If **DKLEN** >  $(2^{32}-1) * \mathbf{HLEN}$  "derived key too long" and stop.
2. Let **L** be **DKLEN** byte blocks of the **DK** rounded up. Let **R** be the number of bytes in the last block.

$$L = \lceil \mathbf{DKLEN} / \mathbf{HLEN} \rceil,$$

$$R = \mathbf{DKLEN} - (L - 1) \times \mathbf{HLEN}$$

3. Each block of **DK** is the result of a function **F**

$$T_1 = F(P, S, C, 1)$$

$$T_2 = F(P, S, C, 2)$$

...

$$T_L = F(P, S, C, L)$$

Function **F** can be defined as an exclusive or of the results of the HMAC function called **C** times.

$$F(P, S, C, I) = U(P, S \parallel \mathbf{INT(I)})_1 \mathbf{XOR} U(P, U_1)_2 \dots \mathbf{XOR} U(P, U_{C-1})_C$$

Function **U** can be described as

$$U(K,P) = \mathbf{HMAC}(K, P)$$

Notice the first time the **U** function is called the input parameters are **P** along with the salt and block counter **I**. However the second time the **U** function is called the input parameter is **P** along with the output of the previous call to the **U** function.

4. Concatenate the blocks and extract **DKLEN** bytes to produce **DK**.

$$\mathbf{DK} = T_1 || T_2 || \dots T_L [1 \dots R-1]$$

5. Output the derived key **DK**

## 9.0 HMAC

HMAC stands for Keyed-Hashed Message Authentication Code. The length of the HMAC output depends on and it is equal to the output of the message digest algorithm used. More information on the HMAC algorithm can be found in FIPS-198a

### Input:

- Message Digest Algorithm
- Key
- Plaintext

### Output

- HMAC

### Notation

- MD Cryptographic Message Digest Function
- B Block size of the Message Digest Function
- K Key
- KP Key After the preprocessing to make it B size
- P Text

### Key Preprocessing

The object of the preprocessing is to make the key size equal to the block size of the cryptographic message digest function. The preprocessed key is stored in variable: **KP**

1. If the length of the key is equal to the block size of the cryptographic message digest function then **KP=K**.
2. If the length of **K>B** then create a message digest of the key and append zeros to create a string **KP** that is equal in bytes to the block size of the cryptographic message digest function. (In the case of SHA-1 that is 64).
3. If the length of **K<B** then append zeros to create a string **KP** that is equal in bytes to the block size of the cryptographic message digest function (In the case of SHA-1 that is 64).

### HMAC Computation

1. Apply the bitwise exclusive or operator to every byte of **KP** against HEX 36 to produce a new string.
2. Append the text **P** to the result of step 1.
3. Apply the message digest algorithm to the result of step 2.
4. Apply the bitwise exclusive or operator to every byte of **KP** against HEX 5c to produce a new string.
5. Append result of step 3 to the result of step 4.
6. Apply the message digest algorithm to the result of step 5

The result of step 6 is the **HMAC**.

### Formula:

^= the bitwise Exclusive OR function

+ = Concatenation of 2 strings.

[] = specifies every item in array

**HMAC=MD((KP[] ^ 0x36) + MD((KP[] ^ 0x5c) + P))**

Detailed specifications on calculating the HMAC can be found in FIPS-198a

## 10.0 Encryption

Takes a variable length plaintext and encrypts it using the specified encryption algorithm.

1. Select a Salt, Iteration Count and IV as described in section 5.0, 6.0, 7.0
2. Perform the Key Derivation function to get the key from the pass phrase salt and iteration count as described in section 8.0
3. Compute the HMAC using the plaintext and key as described in section 9.0. Encode the result of the HMAC function in base64 to produce the digital signature.
4. The plaintext is padded to produce padded plaintext as described in section 4.0
5. The padded plaintext is divided into blocks, each block equal to the block size of the encryption algorithm.
6. The encryption is performed using Cipher Block Chaining mode. The each plaintext block is XOR-ed with the ciphertext of the previous block, the result is encrypted and the output of the encryption is the ciphertext for the block. This procedure is repeated until there are no more plaintext blocks. For the very first block, the IV is used instead of the ciphertext of the previous block. The IV is generated as specified in section 9.0.
7. The IV is inserted as the first block of the cipher text.
8. The ciphertext is stored in the XML message as described in section 3.0. The XML message will contain other fields and values used during encryption such as the iteration count, salt, digital signature and optionally original file name.
9. If requested delete the original plaintext as described in section 12.0

## 11.0 Decryption

Takes a variable length ciphertext and decrypts it using the specified decryption algorithm.

Prior to decryption the several values must be retrieved from the XML document containing the ciphertext such as:

Original File Name  
Encryption Method  
Signature Method  
Signature Value  
Iteration Count  
Salt

1. Before decryption, the ciphertext length is verified. If the length of ciphertext is not an integral multiple of the encryption algorithm block size, the decryption process is halted and an appropriate exception noted.
2. The Key Derivation function is performed to derive the key from the pass phrase as described in section 8.0
3. The IV is retrieved from the first block of the cipher text.
4. The second ciphertext block is decrypted, the decryption output is XOR-ed with the IV, and the result is the first plaintext block. The decryption is performed using the algorithm specified in the header.



5. For each subsequent block, the ciphertext block is decrypted, the decryption output is XOR-ed with the previous ciphertext block and the result is the plaintext block.
6. After the decryption is complete, the padding added to the end of the message is striped. The amount of padding to strip is determined by the last value in the decrypted plaintext.
6. The last step is to verify if the decrypted ciphertext is equal to the original plaintext, this is done by completing the following steps:
  1. Create an HMAC of the key and the decrypted plaintext as described in section 9.0
  2. Compare to the resulting HMAC to the Signature Value

If the computer HMAC is equal to the signature value then the ciphertext was decrypted successfully and the user can be notified as such. If the HMAC is not equal to the signature value then it is probable that one of the two following things happened.

- The key used to decrypt the ciphertext was incorrect.
- The ciphertext was damaged or altered.

7. If provided the decrypted plaintext can be saved under the original file name stored in the header.

## 12.0 Multiple File Encryption

There are instances where the user will require multiple cipher texts to be encrypted and saved as a single output. In these cases the encrypted cipher texts are placed concurrently and written to disk as a single file. The XML tags will separate each ciphertext.

It is important that all implementation will check if another instance of the XML <EncryptedData> will appear during decryption.

### Example:

```
<?xml version="1.0"?>
<EncryptedData>
  <DataID>1</DataID>
  <EncryptionMethod>aes256-cbc</EncryptionMethod>
  <DocumentName>Test.txt</DocumentName>
  <IterationCount>1000</IterationCount>
  <SaltValue>DFEEBEBF</SaltValue>
  <CipherData>
    <CipherValue>DEADBEEF</CipherValue>
  </CipherData>
  <Signature>
    <SignatureMethod>hmac-sha-1</SignatureMethod>
    <SignatureValue>cOQGJE3d3fXi1Bfdvr1v6tz/4lt9xGzfyDPXEv4Q=</SignatureValue>
  </Signature>
</EncryptedData>
<EncryptedData>
  <DataID>2</DataID>
  <EncryptionMethod>aes256-cbc</EncryptionMethod>
  <DocumentName>Test2.txt</DocumentName>
  <IterationCount>1000</IterationCount>
  <SaltValue>DFEEBEBF</SaltValue>
  <CipherData>
    <CipherValue>DEADBEEF</CipherValue>
  </CipherData>
  <Signature>
    <SignatureMethod>hmac-sha256</SignatureMethod>
    <SignatureValue>cOQGJE3d3fXi1Bfdvr1v6tz/4lt9xGzfyDPXEv4Q=</SignatureValue>
  </Signature>
</EncryptedData>
```

## 13.0 File Destruction

All temporary files used in must be overwritten at least two times preferably with random data before being deleted. In cases where the user requests the plaintext file to be replaced with the encrypted version the same rule applies.

## 14.0 Base64

When attempting to display a stream of binary data using ASCII characters some of the data will be lost. This causes a problem when attempting to communicate the cipher text using a medium that can only display ASCII characters. To solve the problem all binary data is converted using the base64 algorithm. The encoded data will be about 33% longer.

The encoding is performed by replacing each 6 bits of data in a stream with a corresponding ASCII character. 6 bits give us a total of 63 possibilities plus 1 for 0, hence the name base64. The replacing of data is done using a lookup table.

### Base64 Lookup table:

	0	1	2	3
0	A	Q	g	w
1	B	R	h	x
2	C	S	i	y
3	D	T	j	z
4	E	U	k	0
5	F	V	l	1
6	G	W	m	2
7	H	X	n	3
8	I	Y	o	4
9	J	Z	p	5
A	K	a	q	6
B	L	b	r	7
C	M	c	s	8
D	N	d	t	9
E	O	e	u	+
F	P	f	v	/

There fore hex 00 will be replaced with the character A. While Hex 3F (111111 in binary) will be replaced by /.

### Input

Base64 encoding algorithm takes a 3 byte input and returns 4 encoded characters.

### Steps:

1. Divide 3 bytes data into subsets of 6 bits.
2. Lookup each 6 bit value in the base 64 table
3. Output the 4 character result of the table lookups.

There are some cases where the last input to the base64 encoding algorithm is less then 3 bytes. In this case for every missing byte the output is replaced with the padding character “=”.

The total encoded stream of data must be represented in maximum of 76 bytes per line. The decoding algorithm must ignore all line breaks and characters not included in the Base64 lookup table. Decoding is performed by reversing the table lookup and stripping all “=” or not found characters.

## 15.0 References

Password-Based Cryptography Standard (PKCS#5 v2):  
<http://www.rsasecurity.com/rsalabs/pkcs/pkcs-5/>

W3C XML Encryption Working Group:  
<http://www.w3.org/Encryption/2001/>

The Keyed-Hash Message Authentication Code (HMAC) (FIPS-198a):  
<http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf>

Advanced Encryption Standard (FIPS-197)  
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

Secure Hash Standard (FIPS-180-2)  
<http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>

Data Encryption Standard (DES); specifies the use of Triple DES (FIPS 46-3)  
<http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>

Blowfish Encryption Algorithm  
<http://www.schneier.com/blowfish.html>

Base64  
<http://www.fourmilab.ch/webtools/base64/rfc1341.html>

RFC 1341 MIME (Multipurpose Internet Mail Extensions)  
<http://www.faqs.org/rfcs/rfc1341.html>

Key Iterations and Cryptographic Salts  
<http://www.abisoft.net/documents/KeyIterations&CryptoSalts.htm>

Advanced Encryption Algorithm by Example  
<http://www.abisoft.net/documents/AESbyExample.htm>

Test Vectors by Tom St. Denis  
[http://iahu.ca:8080/cipher\\_tv.txt](http://iahu.ca:8080/cipher_tv.txt)  
[http://iahu.ca:8080/hash\\_tv.txt](http://iahu.ca:8080/hash_tv.txt)

## 16.0 Test Vectors

The following test vectors were originally posted by Tom St Denis on sci.crypt.

The input of N bytes long (00,01,02,...,NN-1). Were applicable the initial key is of the same format (the same length as the output size). The HMAC key in step N+1 is the HMAC output of step N.

### HMAC-SHA-1

```
0: 06E8AD50FC1035823661D979E2968968CECD03D9
1: 0CE34DEAAD5CF1131D9528FAB8E46E12F8FE3052
2: 23924849643D03BBEAC71755A878A83BD83F5280
3: 6119DD9A7024A23F293A3B67EFA2BF1D82EC0220
4: 379DC76AC2D322FD8E5117CCA765391BC0E10942
5: 7897CC86CFF17A3F95C7AF02CCA03546F5CC2368
6: 1FA1EF3980E86B8DF2C8E744309381727ED10E8E
7: 03B2B726D71DAC6A2BEE63EAA09631DA78F5958B
8: B8CAC4C104997A547374803B5898057B3F8110A9
9: E165E07F8D542FB288C7D367198D0618DE3C9917
10: 18125F046C675F434B3C53A28C301FB2D91B5D34
11: FAAB993F2FEAE442D28FDBB613D2C768ED13342D
12: B657E7EE3A65C6484D007E21484813D9AED1264C
13: EEE2CBB7BAC158742711ED13090FA20462A5E5C0
14: 12367F3A4E1501D32D1731B39CD2DB2C5DF5D011
15: 57DD9DA36E7A4E567A2C5AE9F6230CF661855D90
16: E37110DDD295D93990C4531D95564E74C0EBE264
17: B2115C4E923EC640E5B4B507F7BC97FE700E12DD
18: ED20C67345867AB07E9171B06C9B3B2928F43188
19: 6CA7DFC9F8F432DED42E4EFE9F2D70D82507802D
20: B39EB4D2C190E0CE8FA2C994E92D18CFBCD8F736
21: 91BE5ABF1B35F6227772E36337F258420CF51314
```

22: EB957199EF666C6D0EACC64FC4261D11C715BB23  
23: 2A18D8D4AB1F8C528C9D368BF5A7CFFC2168D067  
24: D4DC370D482D82932701DF8CEAC9337682C2551B  
25: DB9665A6A26DBDE20238F04E9F1A368D26564E4F  
26: D5AE212C9E543F2656699B59DEED54CAACA9A071  
27: BE8890F9DEC6A02AE2848D8505B6408E884E6D1A  
28: E8D9DD9FAA3080560B0EDEF98B745FEE2A1E5479  
29: E219219D2CB8C363C2687F578446ADE1C0404287  
30: E8E7767B35ED8D0965F68272ACE61924CB044262  
31: 1B26689C1EF55448A61DFAEF98B6E7206A9675EA  
32: FE850390864E98A17FC43C3C871383169741B46D  
33: 3F63068D536A282C53E5C003BCEEC96646CF7455  
34: 2962C292CE247F11ACB7E1F981447C51E9BBE63C  
35: B28909A2B7B2E0E13FDCB1124B0BDC31D7D2FEDE  
36: 8DA0FC30C8322DABD67D61E82FC92351894789AC  
37: 543DAC6D449FE2DDC3201927D08695F68F832905  
38: 371540F3092F77867F0CA9DA69318C7673F68388  
39: 7EAF32204EA5993C87E9A12C67ADA4C85D253281  
40: FC4994BAA05F592901085ED7DA188EC3A9BF36E3  
41: EBF77592EF34E81BDA05305876411484DC0744F  
42: 25F64E8F076305D6F5741EA58232F68B725B8F6E  
43: 5DBA03F7E4B4226666F0D8D5BF49FEE77951D121  
44: 98E1D56D723DCACF227D2AC67BF2D6E7FD013497  
45: 53550BC55A367D87416FFA25261362E7D4618DA2  
46: B18434BCCCC5F08B35397C1A6684D60F4F3A452F  
47: FF2BF38DFC6909B46A01E055D173F67A7E456341  
48: DAFA445432ED37FEC99059DB8A0BC528E788E95D  
49: 7FF823C570F8B4C0E483165C076AEA7B5E727632  
50: BC4FC948AB621FE1419CF6006DC04E7D7B32FA23  
51: 1678AFCC3FBD1063E7C82CACAD5B6A933A93091A  
52: 97DC2F9F56738FDAFFD555BF09274153FC2FD009  
53: 74F5CB4F0900441B7AFFC278C01A3038DF3D60C8  
54: 021F66143270C9D58F26AB193DBA81A811917CBC  
55: F486D1C8127813FEEEA8A693C4B8ECB5BB53C3A2  
56: 8397CAB8EED5B2164FEC6BE688971DFA2138934E  
57: E4477CE9BF8CC5A4CCDE039B4E3000F1A0F4153A  
58: D6D2D1E3EE4D643AC4B38836AE54E846F99B376D  
59: 9545B2C6279371D4D928AEE24328121D43DE1E5E  
60: 947ED38EC087C4E53F417E8216408863A8EBFCB2  
61: 32518A2326ACDE1E962B3D0D2BF950F318894E83  
62: 5D21D368FB9D879ADC27B341D608BCF860AB14F4  
63: E2BEDD94D565A51915B1EC6FA9DE18C62D12533A  
64: 15ABF657DB6473C9E2F017C7A2F4DBA3CE7F33DD  
65: 0C9DAF8D959DAE3B66FF8A21A94BAFC523ABC462  
66: A36BE72B501D435CB627C4555A426C4ADAF3D666  
67: 1C171979D67A014A0422D6C3561C817A354CF67D  
68: B75485B08ED052A1F4C3BACCE3C563DF4BA82418  
69: 17297624219C5955B3AF81E5ED61C6A5D05BD54D  
70: 38A9AC8544F0EF24A623433C05E7F068430DA13E  
71: 1E9EEEAAD73E736D7B4F5ABB87BA0FABA623FB2E5  
72: 4B9D59879EAC80E4DAB3537E9CA9A877F7FAE669  
73: 7F76F2F875B2674B826C18B118942FBBF1E75BE55  
74: 1716A7804A9A5ABC9E737BDF5189F2784CE4F54B  
75: 168027EDF2A2641F364AF5DF1CB277A6E944EA32  
76: FBC67DED8C1A1EBBBBC974E4787D2BA3205F2B1B  
77: 33DD26C53F3914FECF26D287E70E85D6971C3C41  
78: 97906268286CD38E9C7A2FAF68A973143D389B2F  
79: 45C55948D3E062F8612EC98FEE91143AB17BCFC8  
80: AE1337C129DF65513480E57E2A82B595096BF50F  
81: CEC4B5351F038EBCFDA4787B5DE44ED8DA30CD36  
82: 6156A6742D90A212A02E3A7D4D7496B11ABCFC3C  
83: 3040F072DF33EBF813DA5760C6EB433270F33E8E  
84: EE1B015C16F91442BAD83E1F5138BD5AF1EB68E7  
85: A929C6B8FD5599D1E20D6A0865C12793FD4E19E0  
86: C0BFB5D2D75FB9FE0231EA1FCE7BD1FDAF337EE0  
87: AB5F421A2210B263154D4DABB8DB51F61F8047DB  
88: 1B8F5346E3F0573E9C0C9294DD55E37B999D9630  
89: 09DAA959E5A00EDC10121F2453892117DD3963AF  
90: ACB6DA427617B5CD69C5B74599D0503B46FC9E44  
91: 9E1BB68B50BD441FB4340DA570055BBF056F77A2  
92: D3E0C8E0C30BCB9017E76F96EEC709BF5F269760  
93: BE61BB1BC00A6BE1CF7EFE59C1B9467D414CF643  
94: 19D693B52266A2833ECA2BB929FBF4FCE691A5C9  
95: B99816886D9FE43313358D6815231E50C3B62B05  
96: 7A73EE3F1CF18B5E2006A20BB9E098E98B6513CA  
97: DEC620F008EF65A790A7D1139ACE6E8B8EFCFA5E  
98: B6BA0EBD215CF1B35742A41EB81A269ACB67C9A4  
99: 3A0FAAD14D3B64BE4EDB9D5109DC05DFFA7680E2  
100: 12E62CE53283B5422D3EA5D8D00BC7F0AE8A127C

101: AA36F0CC6B50AB30286BA52BCB9BB5C1BD672D62  
102: 55120C68B419FE5E12DB526D4ABFC84871E5DEC9  
103: 372BF92A9A2507509C3D3932B32444B7BE1C9BAC  
104: 7AB4B04EEC091F4ADA0807DDD743609BCD898404  
105: 20CB412425E88482E7D184EFEF79577BE97BAFDA  
106: DEB91399A7BFB8323BC8E6A5F4045125277C1335  
107: 6769F41624E553B3092F5E6390E4D983B851C98C  
108: 716760E4F99B59E90A4F914E1FB72A6D2C4B607A  
109: DA0AA5548B5C0AF0CC494F34CAB662A30372DD11  
110: 17A0E2CA5EF666EB34E2ED9C10EBC5DDCD0D9BBB  
111: 1B3614AF749EE359F64F3BE3650210CC3C3498ED  
112: 346E604622CF8D6B7D03B9FE74E7A684AECCA999  
113: 629E46882D214F9BD78418C2A97900B2049F1C83  
114: 765F86114E942214E099E684E76E94F95E279568  
115: 002ED578F79094B3D7E28CC3B06CD230163F1586  
116: 52CC9748778AF5C8E8B41F9B948ABCECF446BE91  
117: 9326190BF3A15A060B106B1602C7A159E287FD4C  
118: 18A5DFBAE6E7C9418973D18905A8915DCEF7B95B  
119: 6D25BF1E8F1244ACB6998AA7B1CB09F36662F733  
120: 5F9806C0C1A82CEA6646503F634A698100A6685D  
121: C3362CE612139290492225D96AB33B2ADFF7AF1E  
122: 3D42A5C1EAF725FF0907B600443EEF70E9B827E  
123: 7FF97FFC5D4F40650D7A7E857E03C5D76EDD6767  
124: 3A92F2A18E8F593E6A8287921E15E2914DF651EF  
125: CDE6F2F58166285390B71640A19BD83CA605C942  
126: 21A227A8DA7A9F5D15C41354196D79FE524DE6F0  
127: EBE93AB44146621BAAB492823A74210D3E9FD35C  
128: 6560BD2CDE7403083527E597C60988BB1EB21FF1

## HMAC-SHA-256

0: D38B42096D80F45F826B44A9D5607DE72496A415D3F4A1A8C88E3BB9DA8DC1CB  
1: 12B06C3218C858558CAD1DA6FE409898C31014D66CBE4ECD47C910EC975E104D  
2: EDBEF6AA747C951F25AB6AAA0D874648CF18FFEC4C9159F8FC71E971FAC6D21  
3: 03436338A166E9051599AB268CD74867C6159378069A9FF46FC07CAE375EDA68  
4: 634758DF0774A587F3AC6AD7988D0965524DE24EBE4DFF07FE622BCB8DA71ACD  
5: 0C08E52C7CFF8B5F70781197069DC8F209552D241687BA0D24661CCCC28D3937  
6: 749F473E0D934694AB9917569A61591CA50BEF18CABDED51666DF243DE879D53  
7: B1E12CFE0273F5D27192D1A4B70EEC4DDC714B66C8BB1921C63381F78CEC5219  
8: 1C60F13A1C519788E989BAC2EBD4F8E126EE6ED82C2E25817C63B2B633FABD33  
9: 5643F445B2C0656A49BB3DB5088C9E2E4B2082C2B611BBA0DAE5791F2FAA5D43  
10: C467F47251DAD4694C9C7A6758E54CEBD68FC933C7C57458020774A2A2B4288B  
11: 85C90CF2719BEBF40EF8D501FDA20C342BC406E728551BC0275ADA1747BD981F  
12: 06B72DAC895B008DA249B7B1D8A5133F09D86BF82DE2C4251BFA6C3D8C4CF03F  
13: 49EDB6714A556DF324E41A3CE5B57006E38FD7CA8B90FEEA2ACAB429204747BE  
14: 7411921D759DA0B491D6D4CC372DB79CC163F146C345B4A73D93EEB4C262A1DF  
15: 5C37FFBD1F0512AF443265B2F3E8B6D01AD9B45FF6F373D2CD0A7C6E48D03E26  
16: 773165FD16D51E51CD8A958E548902B47BBD0A6E156C31B6FEA036F6D8C4A90C  
17: 5B4BE909754EBC8ECBB8B5DA6298B8341B35D92E17CE7281909EBA1EF568347  
18: C6EEF2D12F54815561EEED3426D7AA7E671E26D42384B9478D91FC6B14CC76F8  
19: 4C9FA0575CD96BB1DEF6EA79F5EC7A1F0478E86352812F690C2C2BDB70028BC  
20: 7F87BA45FC41EC30E76F61E4EADEC013CE2B4C49CA6FE6D2FA525F6BBD45E103  
21: 9B8CA1D70339A0894E16CE4E76F6655ADDD3EEB598F3DD80FECC5EEEF3F638C3  
22: E4608AEA430A638799991B748BB858C91AF58F56B226E1901D28336B30498279  
23: AF4F9C52079B28546FBB44EEBA20C7AF0BF493D34EF6967B07CA32FC4DE25ADB  
24: FE51F3A9313EEDAAA991350AB41D7045D42AACF3AC7155DA3AD9A2F1DE3A73E  
25: C1F5AED9D77F8540A4A4B308A139D33F351B20C91A738E698BD8182F124D96C82  
26: 3CAC12A252B93B7D724AF9119FD3C18E85E88401F93BFF42AA05711B9833B1F6  
27: E61D4E94C212324A64B1A0C04B2237A9A1C5CC003D83EA80BCEB45452DCB42F2  
28: D01BA47DABCE4704B6820EC0ECDBEF137B9C4ACB80DC99B7C9220CFD9F9CE363  
29: AED502C53A8B2C76F671376CDDBD0596376B3664B917CD9C9ADBC489543D4721  
30: 3405AFD96584C5E5963362948D112A70155877BE3B5EFD479F226B73351ABAF0  
31: 5FA0290DC68B72B1FA27DBAF157923C706B3F52CDE9C4EE38CDA31D376B0BC0D  
32: C1391C694C985CCBA707A8C78AD05E2180AF6B4DA5BB877AAC5E2AB33B4890E2  
33: B018E7B15F92DBEC58F767633BCA3BD0D84B6D5B9443784DC1757166D7AA1C16  
34: 8D9E2C84967004E3957DF59D502BC11CF8C8959368117EC5DB56AC958A3E791B  
35: B0EAF9C0E869D7A304DDB30061A73C580B0A6F9D49E15442ECFBB3B5A851855B  
36: 0B48B0D8C3ACF7B4F9ECF8E46563C921B1B6720B6C650D72DD1126B6763CD595  
37: 8879D239EDB09F6606957D96A1F4BF37EAC0F3419881EEA79E8BF1364FB3FF6D  
38: CC663E436DE42E32EA110F9D90EB990D9151C9F06D51243D2076B0CC45361736  
39: 732DC3B1F809E55C498C53FC75A23966CAEA16BE984F795CB1BC94D026FAB30E  
40: F1F0EEC77D97A0234D0F19B2FB12A96B6E2FF8626F79A74D4AF26CDE1344D838  
41: 75C9D8C7344668C478D8AE6D9E2C41E336E7A2504CDD43B73CCBF78B4C05EEB1  
42: 4B149BCA6429408B242E76C52C4D3A0A5F5437EC0AB6D24D71EB1AC5496D75BA  
43: EDB65EBEBC0411B4F4DAF186033E306AD500711CCB80E770E99523BB2672A237A  
44: D1BBFF5A48346A0DFD5CFFAA7A2AF08C27F3FC2908D7A5D2F575E07CA9E72474  
45: E8EFB6373DD3457610E57750738358A50026D2C6704A98148CDD69BFF7B70551

46: 8E3733B729CEB97444BCCA405044B98F45FC59BBA86444A3FC0F4DF4854B5C4D  
47: 868F3EE8F4D4DFEDC3FFAE1FA069F5FBB2CB818E63C28151C1566634189234  
48: 3F5396115DC7F17AAB19A3A9779CFCCA57DE7A7C1A42F748FEC49B7D8C2B82D  
49: DC2A5E3E176A693AD8CAE551A505729B78FBDE778B526E28953BC1A56B54840E  
50: DC91FD745E9A7A9D0B41C79B3B3939B84BDF78BEB007F9AAF8FF82084759223A  
51: E73DC5413F17D4ECCE813DC060EF907C2E952AF92DD247A0AE2BE798E6A40B  
52: 696B5EE4C1E1D86B0015EEA2389C9A35088022FFF10034D0D09FA722A2A3E6  
53: F86C07265389512B2CE240A89EA29D61C6C79C2738FACA157B0DE43294485682  
54: DB31CBBFD28D6F8564219911EFB748A5663E482DBA26E38634E8E27E3CF65707  
55: 2F9675313AAB7A940AE77CA906D0342A448FDBA3F7589D14B1344D586EA157DE  
56: 7D829FD994258EF2ADFDEF22C8CD5CC1D29A9A55B62847B3B6F5DB630421CF999  
57: A6CDB9BC9AF75EA4680E895E8EDDCE76F536F7CCA571D62781A06DDB3424FA50  
58: 1B4186A34EB07F5B3127F2BE0F3943610679DB0F6BABC7DA03B416FA577D36E2  
59: 7B5DFF3459DC10B9B7AA2B2829094F97706DB5B2F133B8BF9F48D90253D68359  
60: 2ABB68160300028BBF3B5A414970D11DF4FD6F4B4A35029DEF8492ADF19A480  
61: B1B13ABF9D20C42E755D63EC63C016126259C8A6C3F9AB3F0F6AC5D0BD44ECA2  
62: 9ADDD17E5CF407CDBB12E5E52A50CE134F1B48A2A2AF90D7308344FB5A70485F  
63: 6A4C06DF40BA515C56476471D4A94F87A2B91EAF6C66510892F2F20A342B736  
64: 555D42426C003BAD0B08BEEA76DFC81B307C979BB6E4F15325B2ECD37E04423  
65: 8A58733E0B990D0D82F93F77DF36E30DCFD03B3181B73C544BB097A3A73B6AC9  
66: 6FCCCA4172E30A281A702E36E7BCA07370D4B57272385077A44D5F7933DD2FC  
67: 3B1A91E49AF88B1832F8E91109C7CC5DBEE2847D9ACD2A57404DBB565480AC75  
68: 69584075C278763CB0E09D4C9E15E9300A191BF99907049F14EC8DE24D86C121  
69: 2EE24340D13E68B10B95C3F77D55027F98BDE6BA5328D0C02CF89965687C062B  
70: C04B37F5932F427D40E21EEAB7C9594B16BFCF4F5FE2BF175CD63C62F2CEEA2  
71: 058E1AC8971ADD261744BF7D02B46A8B74A4D52B25643DF9729A1E7DF6CCC86F  
72: 18001F246ABC760197482E25F3AC64B14A795E55B41B505D6027261BFD7C52C  
73: 4AEAAED524B173E08E54A83E2D9A8B8824E6E2F1B89203D698E9BCE7C3242F8F  
74: 7D82CFB1D7427302889CADBA23A99154CBAC0C9ADEC94EAF29EB07DC86B0B7E2  
75: 18D42E92BA532A409CEDA8E3A07E751B430800827F5A9F14D93E3ED231BA08AF  
76: 8CFBA378D8595372DCBE09A6E726C23512F84C01EC3C66ADF6B6C55DF63936A  
77: DE1A6E280A9054C91B826785928F37A16E1D2A9A3CEC831185B26D2B8EDE158C  
78: 920C40B4204C7F3D4775176BD245BA0276604C568B3C29943C9AEF1A1C93428A  
79: 935BB39E5FBC5C4A15AC2A854475578CF80308E531CA86818DABE69BED8824A  
80: D608E561471CC09EC0865C826242CA26AA1C90BDF1625E1A38B96E3EE0CC5F04  
81: EFE2A8D806A1A71596A05A2F5F48D18CFD4A742247B04E8089FAB27291A8DD50  
82: 80235BE35DDEA5D49F124D8BE3D143F87CCBA7D0608C7E2CABBAAB01BB95E477  
83: E9410E0DC14F3BE36A49A5CA673C12E18CBE4F0817E0C1CBD2069349F8A09BBE  
84: B2042A81A36F27B4CB96DBB52A61F701A815869FF5AA0CDDAC0327E1ED1C2F22  
85: E9E5A9501B24952DCFB9D59CF95A9A9E6A27FB7315EB472D1E2B7F523D06D42  
86: 99193B4FAFEFFC932B261EF169250B96901ABF877424FF667CC0DA0154C50498  
87: 1D9C7F7E681D20E1E0324E7F1C8B6913FE8CA87EE52E443335115AB2C458E7F  
88: 7308DB7E2591D23E2109C5084B1174F07D289FBE91472FB2D8C06DF39F826B84  
89: 90F06ADC29070DC50A23D3F093007E273E783491A70A2F0AD6BA40E34F02518D  
90: E676DEEDC972019F10FEC24B4AEAC0A97870E924F7B1D6D3ECF91EF38A2AC544  
91: B5DA3B40FBF373795E67A6338F9AC3AD742741F34048930D9336D429D02EE78F  
92: 6FDE20988863CE157042EE52065EDA233BB2E6EC0464B9DCF2AAC1F3A18971F  
93: 428D4CF7477F0F379F634D1E7C15E4CE6DA067ADC45221A860C9C3AC4235753  
94: 9EC80B57E921DA3F81D13B65AA851F5971E4074C96E0D8B64E50A7F5089C1FC8  
95: 9088151BEF766D0896A48EB6DCC8A09D151C3396FBF3A9FE193C5E7BF9030B01  
96: 86D85302A4762536666316F363BB867EFE25FBD03BDD28EA7522973A1A1BD95C  
97: 007104BD935B532BA4702A78C505D67B41358A61DB8069585B91B1445DC346B5  
98: 5C5709F6202948E805FAC25C454ECFADFAC693955864494E511F0CD1FC9CFDCF  
99: 0B010F71C5323CC96D3B8DF71170968096E44969EA55B4C3DAC632D30D81D529  
100: 54621EC4F31CC7F6273601D81674612B44726B5CC4A76EAD2BBC3D32DBF62A9D  
101: 28EFE1AB745BE64E5DD7286C97360FF2D287F862ADBE44380F85E1388008079F  
102: 831BFA684C25542676AD52819249A10D9EF9C2505D69CC1397D0D39D08B39E5D  
103: EF7922C40CD96A47C5E7AE4D958B495F1D6954EDC20596E303CFBA43190A9EFA  
104: 3A0262EBC746A7C044C1DB043951F7EAC645C40F554898D3D7B2B7AAC4EBD396  
105: 1F2CFBA7275639A12DA7CD1986F920C47850DE3FE13C931618C0FAC765820ED5  
106: 7AC8913C0975101E187FDADDAC5B5EC467A25869C4E630EADBB42DD2DFE4958A  
107: D386591F326C91D274FE625A667B6F9F6F7D99CF56ACB365A218F1CF8E167A70  
108: 66286CB1B61156B005CBFC94C2CAB1A6694D7F123411B8A123F2ACD821C291F2  
109: 844D1038E710690050DA737D56FD6B17C261C7BE512713E62033384B53C40902  
110: 7EF970C40080F554851277F4E950C6F378B0A3DA3CD1BE250D976162F8A4EE79  
111: 9BC20A2B67566688BCAC77FCF30259F11D9B2FD2277D033E6AAE19E36058A353  
112: 796C72D95BBA1A4341C6B0397E165DD21CFBEF55555B35C717CE33B66ADE490  
113: 1E6A9C1F78AFF266EF8B25C32C1FDFB4A0F64AFFD046D257470BF6DAEF61D6D  
114: 0E1AD927AD658C5E0321333AF8AE4ED69903B4F22C5DFF90AC93268507A7C86B  
115: 07B7A778E2931704E7FECA284FF3B14071E255A2B824AD0A2272D21448579CEE  
116: A8D810DF06368A0E825D6DB4394916E43E217BEE9303AD4096A8E1CAD37B8703  
117: 6A9C7D302CCA1EE170366F355D8F40AE3A20D28BFCB2BA163DC68E08DACB748  
118: 40C3A8B08FF9F767491E4243D1808572FDAF1D8CD21AB47115849531513D0750  
119: F26EA6760AA80360398371855783815BCD34431E0CCCEC58A34A67997ACE43CEF  
120: EEA78D68A509988ED67E3F27FC22F3EBCD570EF0FE242A0251457EAC4C31F4  
121: AF977819B87F2E63C0E131DFA2A31C555AD831ADCA6DE0FC1BE48D21A1E7E666  
122: 846A75DF3691B2BF224FB0E66E360A2E8BB1DA32422190F2B319B73E6900AD42  
123: FFA997FCFABC9FCAD4B58B0EF848890FB23B974CD57FA07223037450C371B116  
124: 0028C776965A0AE5E9E70D9B833BF328BDCD06C5A12A2F1C510911E60AA304A









# Hash: SHA-1

0: DA39A3EE5E6B4B0D3255BFEF95601890AFD80709  
1: 5BA93C9DB0CFF93F52B521D7420E43F6EDA2784F  
2: 3F29546453678B855931C174A97D6C0894B8F546  
3: 0C7A623FD2BCC05B06423BE359E4021D36E721AD  
4: A02A05B025B928C039CF1AE7E8EE04E7C190C0DB  
5: 1CF251472D59F8FADEB3AB258E90999D8491BE19  
6: 868460D98D09D8BBB93D7B6CDD15CC7FBEC676B9  
7: 6DC86F11B8CDBE879BF8BA3832499C2F93C729BA  
8: 67423EBFA8454F19AC6F4686D6C0DC731A3DDD6B  
9: 63BF60C7105A07A2B125BBF89E61ABDABC6978C2  
10: 494179714A6CD627239DFEDED2DE9EF994CAF03  
11: 2C7E7C384F7829694282B1E3A6216DEF8082D055  
12: CFF9611CB9AA422A16D9BEEE3A75319CE5395912  
13: E51F9799C4A21BBA255CF473BAF95A89E1B86180  
14: F741644BA6E1BCF5FEE6D3C1B6177B78468ECE99  
15: FB1D9241F67827CE6DD7AC55F1E3C4E4F50CAA03  
16: 56178B86A57FAC22899A9964185C2CC96E7DA589  
17: 0A0315EC7B1E22A79FC862EDF79BDA2FC01669E3  
18: 32AF8A619C2566222BB0BA0689DABCC480C381D5  
19: D35B5AFBC48A696897C084E6E71AAE67C7CD9417  
20: 602C63D2F3D13CA3206CDF204CDE24E7D8F4266C  
21: A3C6FBE5C13E8B41FADC204C0CF26F3F214189F4  
22: 25E480E9E0CA2B610105CD1424B8A35F63FB3981  
23: 45412D51D3CA7BCF452D1612720EE88F9D2427C3  
24: ED6A95036E3E046931597A457DB7A78B7309C4C0  
25: B4FE0256D346700783420E08A4A6F7992B1E36C9  
26: 33E1799E98280E5A9ACE5509477A2048607C5537  
27: CF193837F6DE43F8E38000ACFCF764FA8D8FDE22  
28: 7C8DE247DDA83599AF2EC2EE2D29E20583DAC34B  
29: F38A076F70613FC251C4D21E6435AD08341A8A99  
30: DCD68E6174BD74BA180DA047A7345E8D111F85FD  
31: 43BBACB5F62A0482CBDB564171B04365CA6E27C0  
32: AE5BD8EFEA5322C4D9986D06680A781392F9A642  
33: EB90BCE364635C4C23B49F493F0043579BC85C17  
34: 2942C7AFA65444C43D0592D0DC73CA71DB729205  
35: ABF726F5FDA729FB7F3F0484D7C94B3107AA02AE  
36: 75DB4F6BCC05A781DDA9D17C46717286DD53654B  
37: A82CB42D89DAF5FBC1D4A48476229C495782F98D  
38: FC1A69683744AF823CD69E8A1E3F460591714028  
39: DC68DB44B48521B0700A864896A00E17777AEA83  
40: CC9AD99E917042381B0F99588896CBF236AA8ED3  
41: EC7A68484A749C7065C6B746F9C465DCB414F370  
42: C627C449DEFF14AE7ED807293D30846F061DA5B8  
43: 4782F2A19B6DBB0882D656DE86C3D21A7317F768  
44: 02D4EED99E7307BEA39AF5330BF7FB388D48B496  
45: B3D99B9D90A69E50FD4365704F5AB2EAB7BC9763  
46: 9B1C07176BB227F73E8A4E173071D39302061DE2  
47: D79097DDAC552A6E02A52CE7AAF494D2D73B2557  
48: DF7F23B160E75B9BAE5EA1E62B43A5A34A260127  
49: F598F3780D8C374D97957B9B62D56106E9E0B2D2  
50: 0BD98598F9AB29C1359EF5460A206DD1370515E3  
51: E6C320834F69D81689E1ECD5ABC808D49D9C4E07  
52: FD5EE7588CD129E12B886974621FD29FACC78E19  
53: 2A9C28EF61EB536D3BBDA64AD95A132554BE3D6B  
54: CFAE6D86A767B9C700B5081A54265FB2FE0F6FD9  
55: 8AE2D46729CFE68FF927AF5EEC9C7D1B66D65AC2  
56: 636E2EC698DAC903498E648BD2F3AF641D3C88CB  
57: 7CB1330F35244B57437539253304EA78A6B7C443  
58: 2E780486F64BC91FBFA2785EC1CA5C9E3CC07939  
59: 4A7713D44E97D9F09AE1D786199C58AE2BFAF3EB  
60: C98714B16F92C8A770E9FC229DF834D1688E282F  
61: AACE3DD6F54A2A255ABA920F5FFC8CF04B85A69A  
62: CF8563896A3B0A0775985D8289444C4BBC478DA7  
63: 6D942DA0C4392B123528F2905C713A3CE28364BD  
64: C6138D514FFA2135BFCE0ED0B8FAC65669917EC7  
65: 69BD728AD6E13CD76FF19751FDE427B00E395746  
66: CE705B7C60D46E7E36FE073DB8822698579CA410  
67: C717EBBF6A2BF1BB33DA6257352D5085BEE218B3  
68: 86151D140AAF9A4B5877D3FBB49014FE5906E57  
69: 7446B5A6BBCC58BC9662451A0A747D7D031F9A7D  
70: C24887924F92ADAC5AE367995D12691C662B7362  
71: 5AF83CFD42D61967778889CA911CFB6C14339BA7  
72: 587D4F6E6B4E21343423E434679009CBD3D24DCF  
73: AC65DD946C5CC432D4D624CAEB53C7363F96B7AF  
74: FA71E70750674C0F6B4AA19D0BE717B2936C83FD  
75: C9EFE6DD0A019315F73F3962DE38B6C848A1705B  
76: D1D05649B952C8F6EB016BE08FE1544AAC5D5925

77: CC3081AC1D695BAE51CFD5B44B9FB3A230733CC3  
78: EB9DE332558953792687D9A7F598B5D84BF0A46B  
79: 39DE5EFD92E3D3678F24D2CF545BA4D172D003D  
80: 399DBC9F721E44A992A0DEF42D999B32AF449ADC  
81: 996A2817C8ACBC667E1C4C27B8F4E9952736DD7A  
82: 3EF8189CE1BCC0D65AA182B1A81534635EDFDF2B  
83: D676714C6A6FF4E17A60C0511C25AA8B164FA606  
84: 4DB6E3381E1B9290267C1539E1053793C8B81FA1  
85: 3A34D35B0296FE4D83EDA39B742A9D8F4B13A958  
86: 54F3B45304EF1287F54B877FCCE3285E154F9D6C  
87: B1EA96216E025377AB5AA845238FC8BC65DD60E1  
88: BC6C7488145485DEDE1AE1D43B594F0046BCDA0F  
89: 3D9A0619ECF88C84CE86213E9AA91D9A252CBC32  
90: 92CCAA0B4CE89E2BD80A61B9BAFD5AC58AB7B588  
91: 3EB326B5BF4440FB3A88E3DCB05C1DB5EA01AC5C  
92: 989C63E819B13D4CADFB33F8DEAFBC57C1992A12  
93: AE944552C20CF16F07A5C357713832C9D72D0C6B  
94: 46723E982569A1E2D9EDCED5498FC1F46F7D63FC  
95: 3BC5DAE7907C83A0693F87FD8372EFD1DF53E09  
96: 96D281BA44EB21ECFB1663C8AC5752C48686A927  
97: FA0EF18178880A72B51C26555C10F5210DAB4390  
98: 0C7ECAC32B8ED6D9835D381BF069568722A276E1  
99: 649E44ECBA85C0938EC09229CEE4BB69388EC642  
100: 1E6634BFAEBC0348298105923D0F26E47AA33FF5  
101: AF2AF2734BB2BAA288940CB62109F4849DAA347F  
102: 22D14BC045CC9A3794C99BEEE7ABE278BF24D6D8  
103: C3164CCBED75B82ED3F59F4A47FE09B256025549  
104: C27B5BC7CD24DE4913614A769A442E9CC9FB0E08  
105: F44D48D98CAC77522FF6B9E1B9CBB8489E58E588  
106: EA19A71FFBEC9572F6CD65523ACAF865EC05AB52  
107: CDA0EB9D310247BD1E8B3EA10D9B9DEFF6FBABA9  
108: 449DFCE971B9D65D69FBC72940E9A885E8DDE9CE  
109: 96EEBB6B95A9DA99C58190CBD77CD6FBCF638A79  
110: 670F7A869E90CE86E0A18232A9D4B1F97C1C77D0  
111: BC544E24573D592290FDAFF8ECF3F7F2B00CD483  
112: E4CE142D09A84A8645338DD6535CBFAAF800D320  
113: 1C26461E26EB697CCC36A98714EE70CAA87A84E  
114: 51C5B1C25A71FF00394A84AB48B5733C8955551E  
115: 84803504181C0AE33A511C49AF5015A5B1892BFD  
116: 7CC8BCA120C2635ABFEA82DD203112B5C7E165DA  
117: 44E2519A529D7261F1BEBEDC8ED95E1182CAE0DC  
118: 2A81372DA39C1DF4251539A9922717B7CF5F0334  
119: 41C89D06001BAB4AB78736B44EFE7CE18CE6AE08  
120: D3DBD653BD8597B7475321B60A36891278E6A04A  
121: 3723F8AB857804F89F80970E9FC88CF8F890ADC2  
122: D031C9FB7AF0A461241E539E10DB62ED28F7033B  
123: E0B550438E794B65D89B9EE5C8F836AE737DECF0  
124: FB3998281C31D1A8EEA2EA737AFD0B4D6AB6AC2  
125: 7A914D8B86A534581AA71EC61912BA3F5B478698  
126: A271F71547442DEA7B2EDF65CD5FBD5C751710AA  
127: 89D7312A903F65CD2B3E34A975E55DBEA9033353  
128: E6434BC401F98603D7EDA504790C98C67385D535

## Hash: SHA-256

0: E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855  
1: 6E340B9CFFB37A989CA544E6BB780A2C78901D3FB33738768511A30617AFA01D  
2: B413F47D13EE2FE6C845B2EE141AF81DE858DF4EC549A58B7970BB96645BC8D2  
3: AE4B3280E56E2FAF83F414A6E3DABE9D5FBE18976544C05FED121ACCB85B53FC  
4: 054EDECD1D0211F624FED0CBCA9D4F9400B0E491C43742AF2C5B0ABEBF0C990D8  
5: 08BB5E5D6EAAC1049EDE0893D30ED022B1A4D9B5B48DB414871F51C9CB35283D  
6: 17E88DB187AFD62C16E5DEBF3E6527CD006BC012BC90B51A810CD80C2D511F43  
7: 57355AC3303C148F11AEF7CB179456B9232CDE33A818DFDA2C2FCB9325749A6B  
8: 8A851FF82EE7048AD09EC3847F1DDF44944104D2CBD17EF4E3DB22C6785A0D45  
9: F8348E0B1DF00833CBBBD08F07ABDECC10C0EFB78829D7828C62A7F36D0CC549  
10: 1F825AA2F0020E77CF91DFA30DA4668D791C5D4824FC8E41354B89EC05795AB3  
11: 78A6273103D17C39A0B6126E226CEC70E33337F4BC6A38067401B54A33E78EAD  
12: FFF3A9BCDD37363D703C1C4F9512533686157868F0D4F16A0F02D0F1DA24F9A2  
13: 86EBA947D50C2C01570FE1BB5CA552958DABBDB59B0657F0F26E21FF011E5C7  
14: AB107F1BD632D3C3F5C724A99D024F7FAA033F33C07696384B604BFE78AC352D  
15: 7071FC3188FDE7E7E500D4768F1784BEDE1A22E991648DCAB9DC3219ACFF1D4C  
16: BE45CB2605BF36EBDE684841A28F0FD43C69850A3DCE5FEDBA69928EE3A8991  
17: 3E5718FEA51A8F3F5BACA61C77AFAB473C1810F8B9DB330273B4011CE92C787E  
18: 7A096CC12702BCFA647EE070D4F3BA4C2D1D715B484B55B825D0EDBA6545803B  
19: 5F9A753613D87B8A17302373C4AEE56FAA310D3B24B6AE1862D673AA22E1790F  
20: E7AEBF577F60412F0312D442C70A1FA6148C090BF5BAB404CAEC29482AE779E8  
21: 75AEE9DC9FBE7DDC9394F5BC5D38D9F5AD361F0520F7CEAB59616E38F5950B5

22: 22CB4DF00CDD6067AD5CFA2BBA9857F21A06843E1A6E39AD1A68CB9A45AB8B7  
23: F6A954A68555187D88CD9A026940D15AB2A7E24C7517D21CEEB028E93C96F318  
24: 1D64ADD2A6388367C9BC2D1F1B384B069A6EF382CDA89771DD103E28613A25  
25: B729CE724D9A48D3884DBFCBEE1D3793D922B29FA9D639E7290AF4978263772B  
26: B858DA80D8A57DC546905FD147612EBDD3C9188620405D058F9EE5AB1E6BC52  
27: D78750726155A89C9131D0ECF2704B973B8710865BF9E831845DE4F2DCBC19DA  
28: DC27F8E8EE2D08A2BCCBB2DBD6C8E07FFBA194101FC3458C34DED55F72C0971A  
29: D09BEA65DFF48928A14B79741DE3274B646F55AC898B71A66FA3EAE2D9FACD77  
30: F2192584B67DA35DFC26F743E5F53BB0376046F899DC6DABD5E7B541AE86C32F  
31: 4F23C2CA8C5C962E50CD31E221BFB6D0ADCA19111DCA8E0C62598FF146DD19C4  
32: 630DCD2966C4336691125448BBB25B4FF412A49C732DB2C8ABC1B8581BD710DD  
33: 5D8FCFEFA9AEEB711FB8ED1E4B7D5C8A9BAFA46E8E76E68AA18ADCE5A10DF6AB  
34: 14CDBF171499F86BD18B262243D669067EFBDBB5431A48289CF02F2B5448B3D4  
35: F12DD12340CB84E4D0D9958D62BE7C59BB8F7243A7420F043177AC542A26AAA  
36: 5D7E2D9B1DCB8E5F78C90036A2CF29FE7B66554F2DF08CEC6AA9C0A25C99C21  
37: F4D285F47A1E4959A445EA6528E5DF3EFAB041FA15AAD94DB1E2600B3F395518  
38: A2FD0E15D72C9D18F383E40016F9DDC706673C54252084285AAA47A812552577  
39: 4ABA23AEA5E2A91B7807CF3026CDD10A1C38533CE55332683D4CCB88456E0703  
40: 5FAAAEEC3611556812C2D74B437C8C49ADD3F910F10063D801441F7D75CD5E3B  
41: 753629A6117F5A25D338DF10F4DD3D07E63EECC2EAF8EABE773F6399706FE67  
42: 40A1ED73B46030C8D7E88682078C5AB1AE5A2E524E066E8C8743C484DE0E21E5  
43: C033843682818C475E187D260D5E2EDF0469862DFA3BB0C116F6816A29EDBF60  
44: 17619EC42506F65F083E2314EF30AF796B6F1198D0FDDFB0F272930BF9BB991  
45: A8E960C769A9508D098451E3D74DD5A2AC6C861EB0341AE94E9FC273597278C9  
46: 8EBFEB2E3A159E9F39AD7CC040E6678DADE70D4F59A67D529FA76AF301AB2946  
47: EF8A7781A95C3207A292EBF511EDA3DC6E273BE59CB0F9E20A4F84D54F41427791  
48: 4DBDC2B262CB0F07985BC84202236DBC3777D74660611B8E58812F0CFDE6C3  
49: 7509FE148E2C426ED16C990F22FE8116905C82C561756E723F63223ACE0E147E  
50: A622E13829E488422EE72A5FC92CB1D25C3D0F185A1384B8138DF5074C983BF  
51: 3309847CEE454BF99DCFE8FDC5511A7BA168CE0B6E5684EF73F9030D009B8B5  
52: C4C6540A15F1C140A784056FE6D9E13566FB614ECB2D9AC0331E264C386442ACD  
53: 90962CC12AE9CDAE32D7C33C4B93194B11FAC835942EE41B98770C6141C66795  
54: 675F28ACC0B90A72D1C3A570FE83AC56555DB358CF01826DC8EEFB2BF7CA0F3  
55: 463EB28E72F82E0A96C0A4CC53690C571281131F672AA229E0D45AE59B598B59  
56: DA2AE4D6B36748F2A318F23E7AB1DFDF45ACDC9D049BD80E59DE82A60895F562  
57: 2FE741AF801CC238602AC0EC6A7B0C3A8A87C7FC7D7F02A3FE03D1C12EAC4D8F  
58: E03B18640C635B338A92B82CCE4FF072F9F1ABA9AC5261EE1340F592F35C0499  
59: BD2DE8F5DD15C73F68DFD26A614080C2E323B2B51B1B5ED9D7933E535D223BDA  
60: 0DDDE2840838EF6F9853E887F597D6ADB5F40EB35D5763C52E1E64D8BA3BFF  
61: 4B5C2783C91CECCB7C839213BCBB6A902D7FE8C2EC866877A51F433EA17F3E85  
62: C89DA82CBCD76DDF220E4E9091019B9866FFDA72BEE30DE1EFFFE6C99701A2221  
63: 29AF2686FD53374A36B0846694CC342177E428D1647515F078784D69CDB9E488  
64: FDEAB9ACF3710362BD2658CDC9A29E8F9C757FCF9811603A8C447CD1D9151108  
65: 4BFD2C8B6F1EEC7A2AFEB48B934EE4B2694182027E6D0FC075074F2FABB31781  
66: B6DFD259F6E0D07DEB658A88148F8253F9BBB74DDD6DB3EDBE159A56BC35073  
67: 8FA5913B62847D42B4B464E00A72C612D2AB0DF2AF0B9A96AF8D323FA509077  
68: 7DED979C0153EB9EF28A15A314D0B27B41C4F8EED700B54974B48EB3ECA91C  
69: 1CF3AA651DCF35DBFE296E770AD7EBC4E00BCCDD0224DB296183DC952D0008C9  
70: 5767D69A906D4860DB9079EB7E90AB4A543E5CB032FCE846554AEF6CEB600E1D  
71: 8189E3D54767D51E8D1942659A9E2905F9EC3AE72860C16A66E75B8CC9BD2087  
72: 107DE2BC788E11029F7851F8E1B0B5AFB4E34379C709FC840689EBD3D1F51B5B  
73: 169F6F093A9BE82FEBE1A6A4471425697EC25D5040B472C5B1822AEEA2625988  
74: 2087EBD358AE3EA2A092FC19C2DFEE57C5F0860296BC7B057C14E1227C5CB9D1  
75: 182AB56F7739E43CEE0B9BA1E92C4B2A81B088705516A5243910159744F21BE9  
76: 081F6C68899A48A1BE455A55416104921D2FE4BDAE696F4B72F9D9626A47915E  
77: 5CE02376CC256861B78F87E34783814BA1AEC6D09AB500D579ED8EE95C8AFC8  
78: B93E407404E3E95F20FD647365E0E7F46AFABE9AF1FF083AF996135E00D54009  
79: E81FA832B37BE8ED8F79DA29987AA4D61310DCB14B2859DEDF8FB1DAA2541FD3  
80: C56705FEA5B114B8DC63688533CED21167E628017387C885423B835A55EDD5EF  
81: C2226285D08A245A17058ED2D24AD095B714F608AE364FDDF119E0A7DF890540  
82: F9C270DA8793221A6809AC685FDD4F5387E0FE1EE6AAF01C74F1E0A719621614  
83: E69BEFD6EF7F685C36E343AC1702D87AD6A0E4AC8C0D5C521D04AAD4EF0B7458  
84: 4E3033562AD74A7D43EB5FF5FC2382622C6307CB10E245AD62DA77C4C63CB178  
85: 2EA17629472564A59E5EB845A2CDD04F442DF2FF26BCC866E400F77158D612A1  
86: B90223DF74DD49A8A1461F340F2D7A90F96903CCBB5BC3C74EA3658FC8948B20  
87: E0209F42B927EC9C0F6D6A76007ED540E9BDD6E427B3368A1EA6C5E7565972DD  
88: 10D9BD424114319C099ADF6288F74060CD8918EF1228827A6269B2BF0F0880C  
89: 7D1978A65AC94DBBCC62E3D81850299FE157DD9B7BD9E01B170156210D2815A  
90: E052DFF9E1C94AAA49556F86FAD55029A4875839FDA57F5005F4C4403876B256  
91: 58D29459B2130A2E151252D408B95E6DAC424C564062EB911CC76440CB926CA0  
92: 4E4530C392316F598E1BD07F32166380A8F712A33A48E9EB4247131EC5DC05D3  
93: A09C9D3E42342CF7DEA44EDB4AEB48CF6727CAD8032A12CF77A25829FC249D32  
94: EB978D0F1AC03CE5C3510B5F4A16073A7A2BDC15C4AB7777DCF01030CC316667  
95: 7D1905A3ACE827EA1AC51C4FA08C281ED3BE87E7F4E928D696BFDE35C8F2DC0F  
96: 08359B108FA5675D5CF319FA3434DA6ABBC1D595F426372666447F09CC5A87DC  
97: A7B3830FFAB0F2BBABBEF6DF0B169A7917008BF23880BBF8C20B8E000077312  
98: B4F5D9B1555994C5EBAEBD82918D560A3BF82962A171A1614E7551939E943366  
99: 014ECAEA1B378900F1212898C6DD01565D81AF1D0EF78DF5E28D46E9CAF7CFC  
100: BCE0AFF19CF5AA6A7469A30D61D04E4376E4BBF6381052EE9E7F33925C954D52







230: 01C7D098DCE4E40A69DE14682587FF2A40BAF9833BDCC6413AB54DB0E64262F290D584CD5B21C6558682C50E1E27BF53A18A16D72ABDE878C3522156C9F04DE3  
231: E863DA51CAE09500F589BE05CAAD5788587E2017907444D76F547D6F30632AC658EB8585733BB815D2E19EA046369ED3B81AA773FBFFAC316162389E015A71  
232: FD8232F7B79BDF9C52F5F0D5DE1C565E9D659BF19769096895D182A88028C1CDB7387DD240128A7EFCDF2708EBA7E9E3C676D6E2A036E1B993940F5CCDF1A736A  
233: 3BF8572C0C7B825CE7F3222A3DB87F1C52FBD1A8229B957ACFCF2047C560567483C479603A3C0B0F1B2DD265BEC257D1A32C651508D7A4DF501BC015657DCAC0  
234: 23F5C30B031136A17B8B2FCB55046DE7271312EE3E77851FBD05F78A294815CB2169079168E07647A2BD5D05C1BC2B1EF1B64B929DAA1F9CE723D448C936FEC  
235: 83D10057C7FB494FAAD289B4FE5F093DB2A0C7D79A298173DA735CD5063232BF9E5327A7B4AA795C99F323045790B554476F37EB9D04FE3DF40C047E4113A720  
236: 0AA201EDF4124F421D4515554A1A642E3B9D18C70E09E83A886D6F0CAB0750D9BA1FEB9C587F3ACAB0D8B9C1D83D789102F0E2A6CFF885C50F485929DF4602D  
237: B85CC52981751513B917F58305AFFDDC7D901CB3BB1D1BF5DAB058DE88CDDC2DAE543D73EC6AE0889C9D785F9178D207059D994E1C80706EB28AE65AA100C  
238: 068FED72E55444AE108E8FBD59A96DA4EA3D81A6642742C388BD4EAAEDA6EE21FB8702C2F95152F1F997A5F40F06C54619481F2EC343AD33400913D6FDB4FB  
239: CB4C7FD522756D5781AD3A4F590A1D862906B960E7720136CB3FB36B563CAA1EA5689134291FA79C80CC2B4092B41DF32EBCD36DBE79DB483440228C1622A8  
240: 6C48466C9F6C07E4AB762C696B7EEB35CFE236FCA73683E5FAB873AC3489B4D2EB3D7AFCCCE7E8165DBBF37ADED3B580C889C0B7E0F1790A833D8677429D91A5  
241: 4F663484EFC4758D670147758A5D4D9E5933FE22C0A1DC01F954738FF8310A6515B3EC42094449075ED678C55EE001A4FB91B1081DFAE6A8B83860B7B4C7B4AB  
242: 81A7040857420638D72672A2DF5A49D52B99F9F38B385D8C5129D6A2B82A682CFEAFE6509266E4B00F6B6A07341C2F64E4D4F2152583ED143E3DCF14C1C2126F  
243: 21F655A1334E1A45584F12A22E03B09E3C69E0E1D0FD573AD0D56F9C86862299E333ABE78590E97EEAA5C2FB14DC9F34FEF64DAF6E7A9BFBF68CA6631195CE5  
244: B62C5102F97E5C4D7554790A4CF53A58D3EF44C83142D6E009BD1F6CF8F3A19AA1B89DA8DD9BD1310827A5BF662BE7CAC750C48E6ED91313E940D7D9E5EB9C22  
245: 380023C0BA4C9524FF6778BE80CDF195E36FC460E8CF1BF04E5C2FE08E38C35F183FBCDC3726FF26423F351C507279F6258F2319EA1403B6C8A3DCB384AC7F  
246: 473FC167C7C4BC40B17DA039E09FF3DE884879557E40C52C1981AC419CE021A090BBAE014822D05714077008988D74FF151C927A443E886D63FF2CCD2012AF4  
247: 0060866E1959B1D66C72E754427EAD5E1D6C02D8409F5C32B2F5AE448F54682B504A1ABC0346CCF39BF66A8C7B69081E886B47A7D0B02291462391C95351EE40  
248: 3828B2ED548CFD0B74BB34A1FEAE030E267222198D7E387E7FE3ED503905A25D4C3301A9A47E78372F685B05847062476C507708CD75580ADB579E4C4DC79AA0  
249: C26A7D58B103EDFEAE2F1201BE58AAC127F69AE378DB04156074E991745D4AA5AAB3BA064407DFDA8D34E573B7EC1F9F37CEFO1ADC17FAF393C262A09F2C4736  
250: DCF82307195035A668097514FF1A10E0BF0E802B4945A702D2E17AF6DE1D3D9BA49616DFD16D802054B5219CA37884385E87A713B4EF5C7FCB69661C7F56D5E3  
251: 46049EAD0FA5C49429E15626AF4F2C0EA9DD2F308B99BA6E6E3F3088250A146870FD0B53228D5A1F1BF9859480E1B7A3D3DA180AEF4D5D41BD2951C4E19426C  
252: C0A1FB6C0A65A0D1AF46A5FE86C8A88E8A86F83E36317F435542927C98E74833C887CA3AB5E792CE5E3E21CC6C6AF437349F5A66FAF4CA79742491C643901F9  
253: DCDD20CD47B7C7D011E9DF7855B08336BD5007C4435208BD3B914D7E503B8399164A155697E68A1B88A0600BDCF847A114D98FB773C81FEC817B92057A6998A9  
254: E2DA07644DA73B66C1B6FBCEAE7FF28E3B9024F0BC5408FE02C18E3744CF9BD6DD54EA7BFA1F6F3A81C8560FB938FDF9A38A29853A3A819B58D10213A290EC  
255: 15025C9D135861FF5A549DF0BF6C398FD126613496D4E97627651E68B7B1F80407F187D7978464F0F78BFEEA787600FAAEBBE991EDDB661CD0CE874F0A744  
256: 1E7B80C8EDC552C8FEEB2780E111477E5BC70465FAC1A77B29B35980C3F0CE4A036A6C9462036824BD56801E62AF7E9FEBA5C22ED8A5AF877BF7DE117DCAC6D

## AES-128

0: 0A940BB5416EF045F1C39458C653EA5A  
1: 2B20AF92A928562CF645B1B824F2E6D9  
2: F2C29C3356937ECC3159D8D6EF5E883A1  
3: 4C07B5A2EF31A3229C87AB2E4DE88602  
4: 93AFA1147E793FFCC3D852695A62D008  
5: D4BCC317DC9AFE0E6C7AD1E76F79DBE9  
6: FEDB3371F3C65162AFCCDC6D18C79A65  
7: 4AF2A76F93F07C14161C16B5C176E439  
8: 00A1A596AF7CF44FD12981FA12CB1515  
9: 8013D7006AB38AEBD40D0DC10328751C  
10: 81A077F3A262FA4D00D98EE4D1BEC390  
11: 0CCBC99A3135F26D2BE824D633C0366F  
12: CDBB5568610AD428706408B64DB66E50  
13: CE94461EB0D57C8DB6AEB2BC8E8CE1D2  
14: 06F14868F4298979462595C0FBF33F5A  
15: FE22A7097513246074B7C8DFD57D32B2  
16: 0F2D936610F6D9E32C0E624568BB8E6F  
17: F32BCD92B563D475E98322E5850AC277  
18: 6E6FCB72930D81469F9E05B20FD406C0  
19: 42FF674CBA6C19C4AD84D42816173099  
20: 41C12474A49B6B2B5E7D38E03A4DD4E0  
21: F9E234E3CE3FCED184C775B6140AD733  
22: 7EB5CC6B183D8B3EB4FBA4717CD8838A  
23: CB6C5D78F9721E5BF8E980F0EDCAD4AF  
24: B3F20EF6C26FD9301576D82DA6D50809  
25: F9375037377D86599FB4F241166C43E9  
26: 98BAF9AB7402479C2DA356F5DAE35D5F  
27: 58D1A8E0DC3BC53FD995BB0F60F25FE7  
28: 0A75C0D22D2627C97BA2A7344B9B8C74  
29: 88C299B2F8C9EDAF86A301BBF534BDA7  
30: 755E3A17420281F2C619588A6B521FF9  
31: 0E540DD25C0C147461146E11F832A63D  
32: DC5B58691C6BA5B243036A41301BD7D1  
33: E9299A7336C2D8A51D6C7E2BD1B8F054  
34: 78CA6F682FC649DB289DD62D28D3A22D  
35: 98D96EDA081DE416083650B22BD3869D  
36: E747DE96D122CE1EF6F89BDE0FAE75FF  
37: E48DDF2EDDEB54C861A1E42F5B649EEE  
38: C650C2CF1E903B7F9C8536A2446CA762  
39: CF0BCDCE0F1FE7EB40016C1231FB2962  
40: 37B1C8BE3812147E0D5D115A797663EF



41: 45DD8184581049C4B28FBC0809690C5D  
42: 11B0D889F96E677EEC2E934E9F7F5398  
43: CEC30BC1128A96CD506E406B5ADFAE19  
44: DE67D5439BF83D5338D53F362FCF79B6  
45: 724FBB2D95CBEABC568AA44941D9B6E5  
46: C63F480DA3C73B2A661F1FBC3E4D1F89  
47: 225CD18789D18FF09C982EF38AEF0AAF  
48: B493DEC7E3D11911DEF8788102453670  
49: 23E0B12A67DF025CB77CBDF9E295FCAF

## AES-192

0: 0060BFFE46834BB8DA5CF9A61FF220AE  
1: 597FA00D03EDDC81C2575B4DD6B6AEFD  
2: 4881E4EF69005DCB9110BA327CAC8460  
3: FC4A968AF65FCFF45E4698455918673D  
4: 3079D7B27A3DA5C0805A61CC37109EE0  
5: 9B3F2C7C35806276E2C53826EC1B3C84  
6: FCDFCB1FD9FCF1B63E1AB6737FC154E8  
7: 4A8012AFD410D29CE2CEE0FD195EF9DA  
8: 9F4201C4174C71A3AEF8FD6822197D67  
9: DE3E5E98DA60E895389A1C17E3D50DA1  
10: 20C9064A076C01D1BC121A5A2A1F913C  
11: BA41A36CD24B515545B8B464B244E5BE  
12: 2CC1DE9DBCAC45269C6DBBC9203095F4  
13: 2ED2499CFEB30203E6305B3E1C329C4D  
14: FD709FC0AB48B204C95B74AD189C8832  
15: 7ED298B472C53A4CB7A3BAE588805E86  
16: CB0C6FE2BA76901F9EDE752634DCC31D  
17: 6C5CA6EFCF7101881507AB8770ACF1DE  
18: DEC3C5209E98BBFAA469C5FE6C02A674  
19: CFAC040C1198C8264679CACEAA7E9DE7  
20: EF990992EBA8ECA7E5F95E3B9D69D3A4  
21: 8FC1B640EB55A96D08D83D1184B77769  
22: E1F3DFB9D055BCB2D6CED6DCB8361BFB  
23: 6621F47057706F2A079819DBC0197B9C  
24: 882611AC68778CBD6A46FB5DD4611A37  
25: F35E1367A283CC641FBCE26512A8F2F1  
26: 5A4A71F69056CFBAB67DDA777F5CD945  
27: C446F2BFAD060A9E9E17F71B05ADABD0  
28: 1F0E50F71A67FAA7D169A7A1017FFD65  
29: A6A38588848915509451A2354D2AAC8E  
30: 4C887574F2C5DB00ED4FBAF814A70302  
31: 1B642944162A049CCA9FD0284D7AB4C3  
32: 431BD9293C5BFD12F948C255C838880B  
33: 32CD23A30039AE2FB80B804B905362B1  
34: EBB30E07E7517580A645CD1B5F664182  
35: 292F2BB28BB172620B05C7621BA347D6  
36: 46C06E1223F392D57B98EFCF4C832C18  
37: 451DFBAD2AA92080204F85432236A42C  
38: 768D6206D2B3DD1B9C26FAA5977A6477  
39: 3705F9CEBFE8F91ECE07F84578C05494  
40: 085EB0DCF360F5403FF1E7402A0F7A03  
41: 2A0D56F2E7C7FCE3095F511BDE4AD9A2  
42: A8AB2F3643A61AF164F99FEFAE2CE1B4  
43: E73FD4B1FAE0E5E6A6A31CCC2AF96386  
44: 578E84FD1AA16FF350374E4FD5FDD529  
45: EEAE301DD57084801DB01F8B9C4036CE  
46: 1C44A93B404298327857F71962E0604C  
47: B5F64CD5835C85A68DC23E26D4B8FF80  
48: 6C6F97850A87088AF195D0500B3F5D78  
49: 0BAB3A60A25CD1A750C2C443AA01C57A

## AES-256

0: 5A6E045708FB7196F02E553D02C3A692  
1: 5F7229D6AACF0DAFE3B518C0D4ADBAB4  
2: 96477F47C0A6F482AC4036D2C60FAAD8  
3: 7F791D54914F12E9F0D92F4416EFBEC0  
4: 87DDB19415BEDC42BD361FE380553C5A  
5: 8EDB2A09DC8731DB76D9F67A03AC4D9E  
6: 269A7C08C28D5E4D9355DDBA161F862E  
7: 042A3397BA5029C443DD76755008DB2A  
8: 469C82A94BC5F7B2DF57F0CE1716EE74  
9: 5A84A93077FA19146078310035F4B7E4  
10: 28CAF1C0D811F86CFD3C5EFC30DF79F9  
11: 05B575D06C2D593B708F7C695CE97571  
12: B7E8CACF0A0BD7F2F5DA0B09CC8B8AEC  
13: 0ADDE90F66F1BCF38CEC63EFBF9DBD46  
14: 9BF99E7F5B8F176DD686AF017D5196E2  
15: ABC189EE80D4A4588B3D54DDACCD9778  
16: A57405378580B1E8A8D877791300374C  
17: D1EF03F72FAB3DB68022FC60A2CEC13D  
18: 3D2406231BA17FF7CC973C5E203872DF  
19: C3E07233BD101502953D6186001838E4  
20: DC281C0CE02A83098C80D6C9463F3449  
21: A923023D2390B2230FCE9217776AAAFc  
22: 92E28E69009959FB84046C5ED1B64D1A  
23: CEF8F684EC64A31C651280CDC942DFC2  
24: 5A954684B22691F9CFC60442A654EF61  
25: 56A38A0D93188BAA50DFAF2CB799A76C  
26: 54503340C5DE26679AA5F35215DE85EA  
27: E74BFAF64946DFD699583FF9C47A1EAF  
28: 01F234F9868B085E9B1A2EC84738E2DB  
29: BBCA3DAEAB24EF25BC7B623F4D9FD680  
30: 3956C880F7F7D94ABC259D3D86157F27  
31: 4672C2149054C839C537BDA1F5BBF8F4  
32: CF1E9ACBEB391062593BD88C7948F64D  
33: CA5B4E867AE9D8BA2D4416C908EB99F1  
34: 36666180C768636CF1708CC5C85A6875  
35: 53E396D2755218675494C7AA515A2310  
36: C2B7D31A59A602A65E155F80353DB83D  
37: 0EBCE19FF6FC03E327A2602F858D835E  
38: E47CC2A5E6C7FEF185806E2CFB304D91  
39: D61F15FF75E0F523FA3872132A09AF42  
40: DCC25495052980986AE30756BA0417DA  
41: 451BF5B7C1F1AED9F3D5E18A391EA4DA  
42: 1B6B105C580083D23F3A8EACE41B7984  
43: 8C2F86CD6C86B67C9EBDCAFc5720E4F8  
44: 41360BDB3E4C6836BE0D15B659CEC5AA  
45: F972104AD851BAE0AD963817A3F03F58  
46: 396095F7C102B5A238110DD3D6D4ADFF  
47: F58391AEB9A5D8BB32A3055B37556E81  
48: A23789B146CE89C876F3C331901261D8  
49: 2684AF345C4B13FA154E93A3E2CD2A90