密码学第四次作业

计 24 2012011335 柯均洁

一、编程实现 SHA-1、SHA-2、SHA-3、SM3 算法， 对每个算法给出以下字符串的杂凑值，并提交每个算法的 C 语言程序

i. SHA-1 算法

源程序见 src/SHA1/

1. 空字符串

- SHA-1 四轮 a、b、c、d、e 的输出值

	a	b	c	d	e
初始值	67452301	efcdab89	98badcfe	10325476	c3d2e1f0
第一轮	e758e8da	0b3088dd	8a2a5483	c1afe45c	f63f5951
第二轮	9bdbdd71	62273351	ec805e22	413c1d9a	2aeeae62
第三轮	9b9d2913	982bcbca	b86beac8	c5a3382e	af9292fa
第四轮	72f480ed	6e9d9f84	999ae2f1	852dc41a	ec052519

- 输出值为：da39a3ee5e6b4b0d3255bfef95601890afd80709

2. ‘abc’

- SHA-1 四轮 a、b、c、d、e 的输出值

	a	b	c	d	e
初始值	67452301	efcdab89	98badcfe	10325476	c3d2e1f0
第一轮	fd9e1d7d	dc64901d	20aa99ca	d3a49608	c82f758b
第二轮	32de1cba	4c986405	f718e5cf	03d447f6	f72eec32
第三轮	3f52de5a	09d785fd	3498bfd4	f211824f	d79915ab
第四轮	42541b35	5738d5e1	21834873	681e6df6	d8fdf6ad

- 输出值为：a9993e364706816aba3e25717850c26c9cd0d89d

3. ‘kejunjie2012011335’

- SHA-1 四轮 a、b、c、d、e 的输出值

	a	b	c	d	e
初始值	67452301	efcdab89	98badcfe	10325476	c3d2e1f0
第一轮	01c14432	c7585d1c	de6a32d5	bbc42e51	78076c99
第二轮	bcb7db4f	6ff06403	c527e9d3	9548b103	3591f981
第三轮	5c145c33	adbb3ef2	44f6791b	3c37cbf2	443c812e
第四轮	363ddd6f	78c1446f	1378f032	5c7dc386	5c4b296f

- 输出值为：9d830070688eeff8ac33cd306cb017fc201e0b5f

4. ‘ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789’
   - 输出值为：761c457bf73b14d27e9e9265c46f4b4dda11f940
5. ‘a’重复 10^6 次
   - 输出值为：34aa973cd4c4daa4f61eeb2bdbad27316534016f

ii. SHA-256 算法

源程序见 src/SHA256/

1. 空字符串
   - 输出值为：e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
2. ‘abc’
   - 输出值为：ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad
3. ‘kejunjie2012011335’
   - 输出值为：333842f6a14a24db08c8962d7c5ae5f5da638f5a3bfbda6d8aec1d2ab6f02855
4. ‘ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789’
   - 输出值为：db4bfcbd4da0cd85a60c3c37d3fbd8805c77f15fc6b1fdfe614ee0a7c8fdb4c0
5. ‘a’重复 10^6 次
   - 输出值为：cdc76e5c9914fb9281a1c7e284d73e67f1809a48a497200e046d39ccc7112cd0

iii. SHA-512 算法

源程序见 src/SHA512/

1. 空字符串
   - 输出值为：cf83e1357eefb8bdf1542850d66d8007d620e4050b5715dc83f4a921d36ce9ce47d0d13c5d85f2b0ff8318d2877eec2f63b931bd47417a81a538327af927da3e
2. ‘abc’
   - 输出值为：ddaf35a193617abacc417349ae20413112e6fa4e89a97ea20a9eeee64b55d39a2192992a274fc1a836ba3c23a3feebbd454d4423643c

e80e2a9ac94fa54ca49f

3. ‘kejunjie2012011335’
   - 输出值为：
     66e4a5066eaedfa689858e6db8a7d837163ad03689a48ce37f1864def78780ce6ed0cadaf85ae3944edf072de2d82585b5c746a873ae8a3ee1258da30763259d
4. ‘ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789’
   - 输出值为：
     1e07be23c26a86ea37ea810c8ec7809352515a970e9253c26f536cfc7a9996c45c8370583e0a78fa4a90041d71a4ceab7423f19c71b9d5a3e01249f0bebd5894
5. ‘a’重复 10^6 次
   - 输出值为：
     e718483d0ce769644e2e42c7bc15b4638e1f98b13b2044285632a803afa973ebde0ff244877ea60a4cb0432ce577c31beb009c5c2c49aa2e4eadb217ad8cc09b

iv. SHA3-512 算法

源程序见 src/SHA512/

1. 空字符串
   - 输出值为：
     0eab42de4c3ceb9235fc91acffe746b29c29a8c366b7c60e4e67c466f36a4304c00fa9caf9d87976ba469bcbe06713b435f091ef2769fb160cdab33d3670680e
2. ‘abc’
   - 输出值为：
     18587dc2ea106b9a1563e32b3312421ca164c7f1f07bc922a9c83d77cea3a1e5d0c69910739025372dc14ac9642629379540c17e2a65b19d77aa511a9d00bb96
3. ‘kejunjie2012011335’
   - 输出值为：
     1bae3aae5c31c4a16339264e540a22e6cca449912c5303b0beedad1b30db80ef211ba86ad8bfbb9831b5871356d1c160836314c6692be13bfd7c2dae16d464df
4. ‘ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789’
   - 输出值为：
     d5fa6b93d54a87bbde52dbb44daf96a3455daef9d60cdb922bc4b72a5bbba97c5bf8c59816fede302fc64e98ce1b864df7be671c968e4

3d1bae23ad76a3e702d

5. ‘a’重复 10^6 次
   - 输出值为：
     5cf53f2e556be5a624425ede23d0e8b2c7814b4ba0e4e09cbbf3c2fac7056f61e048fc341262875ebc58a5183fea651447124370c1ebf4d6c89bc9a7731063bb

v. SM3 算法

源程序见 src/SM3/

1. 空字符串
   - 输出值为：
     1ab21d8355cfa17f8e61194831e81a8f22bec8c728fefb747ed035eb5082aa2b
2. ‘abc’
   - 输出值为：
     66c7f0f462eeedd9d1f2d46bdc10e4e24167c4875cf2f7a2297da02b8f4ba8e0
3. ‘kejunjie2012011335’
   - 输出值为：
     8f908d7eabbad79c9420baa854e775b35d73da1320acdbdb1b82c92d92becb68
4. ‘ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789’
   - 输出值为：
     2971d10c8842b70c979e55063480c50bacffd90e98e2e60d2512ab8abfdfcec5
5. ‘a’重复 10^6 次
   - 输出值为：
     c8aaf89429554029e231941a2acc0ad61ff2a5acd8fadd25847a3a732b3b02c3

二、使用 DES、AES 分别在 CBC、OFB、CTR 模式下加密‘aaaaaaa…..’ (将字符 a 重复 10^6 次), 算法中使用的 IV 可设置为 0。（提交程序和密文）

1. DES
   - 源程序：
     见 DES/lib/
   - 结果：
     密文见 DES/res/目录下的 DES_CBC.txt、DES_OFB.txt、DES_CTR.txt
2. AES
   - 源程序：

见 AES/lib/

- 结果：

  密文见 DES/res/目录下的 AES_CBC.txt、AES_OFB.txt、AES_CTR.txt

3. SM4

三、使用 AES-GCM 对字符串

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789

进行认证加密。附加数据 A 设为空字符串。

源程序见 AES_GCM_py/

输入：

```
master_key = 0x00000000000000000000000000000000
plaintext =b'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefgh
            ijklmnopqrstuvwxyz0123456789'
auth_data = b''
init_value = 0x000000000000000000000000
```

结果：

```
ciphertext = b'\x42\xca\x99\x8a\x25\xf0\xe4\xda' + \
             b'\xba\x62\x89\xf5\x3c\xfc\xb1\x28' + \
             b'\xa6\xc7\xf9\xff\x1c\x1d\x0e\x7b' + \
             b'\xae\xa7\xe8\x9d\xf7\xef\xa4\x86' + \
             b'\x47\x6a\x78\x4b\x25\x1f\xf9\xb4' + \
             b'\x4f\xf9\xc7\xde\xa3\xe7\xde\x96' + \
             b'\xbe\x35\xdb\x63\x21\xbf\x1b\x4e' + \
             b'\x4f\x4b\x8a\x8b\xf1\xfa'
auth_tag = 0xd2752fd2c4310e79669058f1d6344c1f
```