现代密码学第二次作业

郑凯文

2021 年 5 月 24 日

1 AES-128

1.1 算法原理

在AES-128中，加密算法的伪代码为

```
RoundKey=KeyExpansion(CipherKey)
State=Plaintext
AddRoundKey(State, RoundKey0)
For i=1 to r-1
    SubBytes(State)
    ShiftRows(State)
    MixColumns(State)
    AddRoundKey(State, RoundKeyi)
End for
SubBytes(State)
ShiftRows(State)
AddRoundKey(State, RoundKeyr)
Ciphertext=State
```

其中$r = 10$，State是一个4x4的矩阵，每个元素长度为1字节。用$s_{i,j}$表示第i行第j列的元素，则对于一个长度为16字节的数据分组，按照先列后行进行对应：

$$\begin{bmatrix} in_0 & in_4 & in_8 & in_{12} \\ in_1 & in_5 & in_9 & in_{13} \\ in_2 & in_6 & in_{10} & in_{14} \\ in_3 & in_7 & in_{11} & in_{15} \end{bmatrix} \rightarrow \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} \rightarrow \begin{bmatrix} out_0 & out_4 & out_8 & out_{12} \\ out_1 & out_5 & out_9 & out_{13} \\ out_2 & out_6 & out_{10} & out_{14} \\ out_3 & out_7 & out_{11} & out_{15} \end{bmatrix} \tag{1}$$

AES-128的加密和解密是互逆的操作。RoundKey的生成过程不变，将加密算法中每一步的逆操作用前缀Inv标识，如InvShiftRows、InvSubBytes、InvMixColumns。由于异或的逆操作仍是本身，AddRoundKey不受影响，只是使用RoundKey的顺序相反。

解密算法的伪代码为

```
RoundKey=KeyExpansion(CipherKey)
State=Ciphertext
AddRoundKey(State, RoundKeyr)
For i=1 to r-1
    InvShiftRows(State)
    InvSubBytes(State)
    AddRoundKey(State, RoundKeyr-i)
    InvMixColumns(State)
End for
InvShiftRows(State)
InvSubBytes(State)
AddRoundKey(State, RoundKeyr-i)
Plaintext=State
```

1.1.1 SubBytes

SubBytes对状态矩阵中每一个元素（8比特）进行两步操作：

1. 在有限域$GF(2^8)$中求逆
2. 进行某一固定的仿射变换

最终的结果是一个固定的非线性可逆代换。

InvSubBytes为上述代换的逆代换。

1.1.2 ShiftRows

状态矩阵中，第i行（$i = 0, 1, 2, 3$）循环左移i位。

InvShiftRows为上述操作的逆。

1.1.3 MixColumns

对状态矩阵中每一列，进行如下变换

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{2}$$

InvMixColumns与其类似，可以求得上述操作的逆为

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{3}$$

1.1.4 AddRoundKey

状态矩阵逐列异或4个长度4字节的RoundKey。

1.1.5 KeyExpansion

将16字节的CipherKey转化为44个4字节的RoundKey，每轮使用4个。转化过程用到了SubBytes。

1.2 实现细节与优化

1.2.1 S盒

SubBytes只是一个大小为256的代换，可以使用lookup table加速。SubBytes(a)可以查表$S[a]$得到。同样的，解密中也可以使用逆S盒。

1.2.2 T表

在32位或64位处理器上，将4个字节作为一组统一处理会大大减小运算次数。实际上，加密的每轮可以在转化形式后成为简便的32位运算。

在每一轮中，将SubBytes、ShiftRows、MixColumns、AddRoundKey合为一步，原状态矩阵为$\{a_{i,j}\}$，新状态矩阵为$\{e_{i,j}\}$，对于第j列，可以表示为

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = S[a_{0,j}] \begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} \oplus S[a_{0,j+1}] \begin{bmatrix} 03 \\ 02 \\ 01 \\ 01 \end{bmatrix} \oplus S[a_{0,j+2}] \begin{bmatrix} 01 \\ 03 \\ 02 \\ 01 \end{bmatrix} \oplus S[a_{0,j+3}] \begin{bmatrix} 01 \\ 01 \\ 03 \\ 02 \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix} \tag{4}$$

这样，可以预处理得到如下4个T表

$$T_0[a] = \begin{bmatrix} S[a]\cdot 02 \\ S[a]\cdot 01 \\ S[a]\cdot 01 \\ S[a]\cdot 03 \end{bmatrix} \quad T_1[a] = \begin{bmatrix} S[a]\cdot 03 \\ S[a]\cdot 02 \\ S[a]\cdot 01 \\ S[a]\cdot 01 \end{bmatrix} \quad T_2[a] = \begin{bmatrix} S[a]\cdot 01 \\ S[a]\cdot 03 \\ S[a]\cdot 02 \\ S[a]\cdot 01 \end{bmatrix} \quad T_3[a] = \begin{bmatrix} S[a]\cdot 01 \\ S[a]\cdot 01 \\ S[a]\cdot 03 \\ S[a]\cdot 02 \end{bmatrix} \tag{5}$$

它们都是由256个4字节元素组成的表。我们用一个4字节变量存储状态矩阵中的一列，某一轮前的状态为a_0, a_1, a_2, a_3，变换后的状态为e_0, e_1, e_2, e_3，RoundKey为k_0, k_1, k_2, k_3，则

$$e_j = T_0[a_{0,j}] \oplus T_1[a_{1,j+1}] \oplus T_2[a_{2,j+2}] \oplus T_3[a_{3,j+3}] \oplus k_j \tag{6}$$

其中$a_{i,j}$为a_i第j个位置的8比特，可以用位运算取出。这样，只需16次异或，便可完成一轮。

对于解密过程，一轮中4个操作的顺序发生了变化：

```
InvShiftRows(State)
InvSubBytes(State)
AddRoundKey(State, RoundKeyr-i)
InvMixColumns(State)
```

注意到InvShiftRows和InvSubBytes的顺序无关紧要，而由于InvMixColumns是一个线性变换，满足

InvMixColumns(AddRoundKey(a))=InvMixColumns(a^k)=InvMixColumns(a)^InvMixColumns(k)

这样我们只需预先对除去第一轮和最后一轮的RoundKey进行InvMixColumns变换，就可以维持加密时InvSubBytes->InvShiftRows->InvMixColumns->AddRoundKey的顺序。于是，可以同理构造解密时的T表。

1.2.3 循环展开

我参照了OpenSSL的高效写法，在处理每个block时，将9个中间轮全部展开。由于状态矩阵只需要4个4字节变量来保存，直接定义了8个u32类型的局部变量s0, s1, s2, s3, t0, t1, t2, t3来中转状态（s->t->s->t->...），这样就免去了状态矩阵的复制开销。同时，局部变量可以尽量被保存在寄存器中，相比访存性能大幅提高。

1.3 正确性与性能

我随机生成key、iv与16Kbits的明文，共进行10次实验，每次将加密结果与OpenSSL的加密结果比较，并验证可以成功解密。

程序还会随机生成key、iv与16Kbits的明文，进行1024次加密和解密，从而较为准确地计算吞吐率。开启O2优化时，某次结果为

```
encrypt: 2692.696062 Mbps
decrypt: 2678.272514 Mbps
```

不开启时，某次结果为

```
encrypt: 841.618011 Mbps
decrypt: 851.335533 Mbps
```

测试在我的笔记本电脑上进行，处理器为AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz。

1.4 测试样例

以16进制给出测试样例：

```
plaintext:
dbc5739ebe478d64572c4f731c6c67c11be5133a0d3dc9ac912f2de595ab6b70
70de0e2e259b937dc7e2f0e34f57a56a3db8a44af56ef6879d236c33ced7a33e
b6b26ddb4d005815e248f831a09d9cdd5540274bae1dd24c403e7f0e15224ccb
d4b9a722b9ff379c482fcde8cd69c522aaec6d58093fa4497d23579346a35e1a
5d053c160573b24da3803570e9fa9293e600ecef3f9038bdb48f50fa32ae148f
b451a6b9c4580667d83bd7c2356a551b6a410aa9d2426686d1b6800365949319
e539d2aa91d8116a13e92c48538163bdc36d6695afcd1b80839b83e82f160115
4fd3bfe1abd04bbeb977060cf869c9bbd6305085fd6b0580068969369f6a4bef
3e0ad0e9da1ba89492aea08a186a46ee9a967497027917080 2803ea2eb899129
9361126e7cba020e69a298810cde6fa675e33d775d557f5fd5be01c04792e9db
f3fc496fb64b7d1fed16a0faf410a069f3dee0503360b0081eb1c96544b24037
ae89a765d42484c23a25bc2f355c98283a79796dd92976f7da3f5c1ef19d56a0
26fd05fb2189bd5cae798be3d5230c109c857d75aef36c8832c9a72466fdc48c
fac9871b52447701bd02e49326f0a3c27520382314a4ac466d536ad3502e604a
f7e7654a2cdd4be9df2f7c05201fc8954000b954a4659a12b805e5083345522b
2db7755994c04274efbf790fde41a51e415e72e6c30df87b12dd834523d57050
8ce5a921a5eb9595aa0ea4895049a791a71a776a276fe5394d687e703defc0ca
d469eb7a54800fff8eb388defd2f70a449e70f7057f4a9a45d28149a17d464eb
3d4f6591cf74905e28183c2548acc99194d802ebcdab8f2ad3a3c4ea7729d6b4
783b4548b0d6a6d8eee2fd368fc6c8239fca0e6c759d9649405a33b783096bfc
45b044f586eacd75cccaab5b90737e2f3d8c9bb32931fc698c2f200f398b0b7e
3c4f73c2394037060ae3619a56e0ca946c6547969743ff23722032abab3e29e7
8d9caac7dce1cde6c42e811b0e4baf7bb0f6114739106aab309d57dcdb80c368
1d6d2ff94ffce0132b612e39acddb45cd3c5a40cd60eb806ab0fe2868fa6efac
131ea6621b867646e7a47f938234ef55f99362cfa21ad64d29b8d4b85ec36572
e10bd4fc914a4278efc20b71f6fac6ef8e28bf3042957d6b4d5124ac14891ef6
94f2f2253d359d2cf7a89deda263dc308c9b60ce30de3a7e2f5e2a44e7483a7b
3a2ca077613da358e5404587a422b830bd18feeef6386c2696966a7ddea4f818
d0989032d5338aba74d04218f2fa48af12469d097f092f159f99937d3d8b960d
24263ff959cab4cd9af6e58cf02d3b0274d90bf3e23a0882d39bff1027951e4b
bb5d441527f8e2c1eec84ddef589e16962ec5c442765c6fa00c60b275b297217
86b72caeaf0e6f9ed6bd7ccc465d35a84a92ec71f7b36bf779761fd49f91eb26
4817d4f8264396fc0012c84670feeeba90db2b878e967e070d9ddbac2fc7d277
dea66f04ea0501ea18c93188c71f4257fa6dde88035d8f10fa6bbd29328fa110
```

```
3610152016160a2edf3bb6a75bf8fe5565ddde683a6d7934d8365e0ac5ff1bfb
0f301b25462653256109ccbc01cb1266a8f0cfe25d4816367e744043735b3f83
8b5aa8d180fcf7e205c39e078eb06d36a03c18fe842f3402a3744617d0859a5b
df422d603e244244e7e04b7691b8ac31f5c52f79f4637c97d8c2aea847480326
8b3086c954c80d3ca958b23a115e6b06239b7f17fefbafd6bd5d7e04a6822b31
b2b1fa077a08432360f55d7153c8777763f78e62f23d38b09bb7b44139df72eb
916cf20b74352ed52a8b467e53bef5b7b58319a7c151575c080c9d41eb0f2d7c
7b1f87f055b5c57f400bfd94c9f24b7e76642637b57d93be8930ff753f2cf1ba
4c79aaa12e6f206f7b1e0344104ec386b2e9bd67665025f0802565bf51567a9d
cf243efe945f6d0f7d70538dbe161470ffd1d76622fd56a222bb627311dc11e1
004fdf94ae4ca32bbcf7b97a0dcdea0d9ec173c0bec963e084c55495a16576a1
b4553663a1d98e5dd047d7de14c1ebb3835e734127d622ab9b76403cdbb7de8f
0c14f2aeed810bbec8e39cdda4879027e503690cda8bb77501f7b2dcae906bbb
a45e6991df744fa757eb84fc72142357188c63f2171a67181219f4c0a9607b4d
bee4df9d592e44b01ac9ac8cddd0e4f55c47e774624f8c7468813412e1b05f9f
943e3ced6d809e87494a13271af71c773f04eba1537715bbf849cdd9f92d788e
6bb47bd835195f7e6473a57e6ac2f5a9c6e04a19585fd450a9a22aa2cfa2303a
57ac138cc5720a29e5b0a850729df9387e4451d6a325264cc750ef96f31fd14a
cbe4d69156e0ba3c90628c0200853a7ec98b546db17ab978cba80fbec8e00893
c4de241abedf564f41e25141688cbf3117139ec88e584159005017c8301f5cf4
fd800ebb5f650aa1475ce2afe8a2e1ffb57fc843d7099cd859b3a089d2fc7dcf
7d8b8bdcf0957d38f160e7d902c8d9b748a1fb1faa97f7034b988c1d9409ed11
9478ee850d6bbdffcba4d8cd6db185b55280d4fc17ccff62648b80f8946d0a29
e5f8aef2636bf12f0fcafc7c7b8131ce0106ca19d2ca7b3655fb2eea6838134d
30c140942c31c33bfbbfb87741e94542ef0f5bc1d9d7f72fd226193b5e2c888f
edc82319fae654f5a50c6ce6f6b129e5c184a79a5b9ec92ec4e269230ef1b2fb
bad514b4bb69a9607516476bc7705188f4f8235096ec7e5bcfe77eddd830d992
05ed46c056f020cc066737cdd78856cc80791c17659a723481f0125920ebec25
d832e52f2205fb286d32f644bb4c103bc52c522ac6c45f47b471a1d45c8df934
bfde63e2e45e0a519100954c4ca68711d2da3c999e9be0530c8127680e219cce
key:
ff00b0e35eba34efbbca3b0770c31942
iv:
9d55db3bf0bc8efc3db6644cd7001ad6
ciphertext:
fc51975f7e72b6e1932ef212d0b571c777ed3f9784c752fe833f8f335322c9cf
ca9a72f9e1b91167cff5531a4d6709fd3e122b75079508b43c5a20a6226d9825
ca40469a3dfe990ac65effe390ebd508dc6edc8356ac680f9f643b8ace249b00
da8cc5be7a69f15d4e62daba4c9ce27ade4e29772b1b6f92c2c6d2624d346b75
3c2f2d982230d73478df77b8491a2a91f8714ad75eaba9407f193283978d6b95
dac7d45f6e543c2ebcfa6f05e1ef50da8156d9e561c2837674da37d869365a35
```

```
77b1fe343a0b0d133880c509f197e40b08c2234eec306a565c3732605a038595
399ef552995615e27b79e5030b30e7c2c544812a07f9c0f26e81ef72210fae8a
b081ef4421d98a3e227dced64e587011217e534858c497299af0123b75360051
91c4338f1ec48bb129b2e66a6f293dc1af7e0545a999261d397fe0d766823d3b
bb2e7d0903d7cc97fc98fee2f59e38965910fb6b9109403870874144b3b31e37
680b246d6bfbd6c91568584f445148602687e346e173c0daea46bf8126ecb95b
83e3c8f145169555c9f2dff2abb745c384010ea6dcb93935c07617d854451411
148f7484559a53f2c8e7047468178666bff5b7a5960436f70b79cf44cc763365
2463383211e0d1fa0a4bd2b4052d1e33c6ef33e9bf3400e62519960388a18896
7e931ed00006b4800f318e188ed4a8ea4cab222347470c1ff9a5db1d717fb61f
2ab4d63314895a63267c987320a0b7e4b377a572e2dd5f27dbdf47255c278150
c590ed055c8deb31fe58b48f7b7535ec142dee5202fc6316f81890d638b8c807
e341734886d07cac0b14011d5847dcac70762aa064f59ffca1e9578ede6b04a3
0c1948e1fff511dac290c55d8cbe3c0ab5b814b9a040120e54190b8ef3148d4e
8cc3887d719069eb21b9a030fe7fb0211025ef3fe891e65d1ad1283e22bef206
b96cbc2797b3994d067d1668eba2380f3aaab45bcefaa4d2f62f1ff1d695d538
8f607bbeeadbff1762d1056c67eaa0a443639a1b2c1408817c061c08404b6a6e
5e48398f5a32b368bcfe797c0f5c7c505d1c79827e9d522e4347006105e15742
2b5ae1f9c0f87ddc758e9e5221b44401428ad46102d2b2e4d3a85480209741c5
01feb64515ae14675851f70fe217589d220da2ed0f0f87134261cc1a080052e9
2297d398caca9c1f830ef51db0a1a657fe20c249021e8f43687d7654349514de
492f0ce3d9f654875b1b2261825615e0c185e90e5909d89986f08a33482ef822
70cf6333a93fb240284d9b2412fd7e2ffa7dc56490dc83dcccd9809ded423627
d38b31bc961264d9cb97afc624b16a47d181ec90bdf714539d6f7b73cde40ce6
040e71ebcd999de51fb741a5c3552f96b0427b3f404583b3557e24b6d0d9f1c7
b81fa877f64b08a9b9efe1379ced9e68fd6b21716128ca8f3f3672be7b08f342
02e6b16996531dca84795a1aab678a130f77f69984271ab4e17118a75472b3cc
ffb4e72868b898f510b7e1fefd0dd9ad08a4c9f33304e82c0fce866603bfdf62
e1e44dd226c342cb8b840a976a03d4ac70fa3c8172fbab73be19d32a6221b83e
8156e2ae58b474ef46003fcc73ca27d7a1df5c4328ab81cd7760981f55f1ac58
ba2dda0e2a6d97b45b41524d349a0210b5ca395df625f3ae0c3476a8ab41a771
b49cf74bdea5f89f77e4c8a7f5d3c0cb83dcca4febe53d7214bbca1dae3ef2e0
3e4d4f0b8c0f3e2ccc7fde1e21828a3d1e40ef7987d43208e3cfff1ac95a1a7c
ded5431b34137c48c04908d07aed067fe2d9024af71aeb02e2b952b10204125e
97381ac3d79c8cd40386c7763eac1a012b8064c03784673c7f37816310f3c206
3c9b751691fd30e5b5e3d5c76005bf103363e5032e4a17b4af2bb430b0667dfe
4ffd04b4ac659bd6be359b8fcb6f334f4f63dab69fba5448b4a2c59e0188c351
451773acba2786cf7eb51158dbdb6bc5a9595fce16105d3ade2ee9ba4c6b0555
9ce5d46a6dd38e1b434eb8a40654dc2b9959228630e8247b8ff0025d952a97cc
c8fbef11059fb6d2e41e5238ece36e56a616b540337a09acb97d16fc0f949279
09d3543e1404c66a96fea3fe637d6071c37d287a5881335b1b89a28a27dccf98
```

```
527f01761ca8ba14c9b8a805266b1ca40a97f817d770cbf627ab9acbc584195b
ba66c8eb8e4261be30a5a03060300002debd1a52b0b8e8129bc85ae7661c23fb
ef25fd05084607fd63acdcaec30d62b273f250fb0dfee300f02cf856e53f030f
409989bded82e7c8393b4bf49777e7f876ceba488dacc0678ac52eaaed501db6
e45566d3f5a21e56b1830b2161c3c2e1ce07de7433fcd4be2d953877458427f7
34c0b80d632dc34a646222e23907ffb7264d82afde7cd18bbaa12c2047c20b9f
26f355a6e7c10763560309a80fef9bfeb15afce695874c4d6b44087047c54848
3b6e4f15c27ef9d5a7076d19bbb81c45fb784a64c6e33396d635c1dba3beb6ad
4854d46c238309e94c735e65aae136393563eea963e54051fee219f610942813
cdf54759f40b2847356a0fac0a4b572eefd9b68feae2875c286af60c9842d139
149e0067ab87493215b68e5d1da54c05083fc90b76e974f146d3eef6517e6dfd
47d6c717d960b566bda14fa5806cbab6e7ca1ac0b30e6846ef87ad487b2d95c9
ac15b754ec22c65b3e6fa5dc1e6b9c0b18583d0f1a81fc76fa64adbf1a4f10bd
54eec9dff499107a90de023a38c78a2e71e8f92fe48b58f35e402745f9ec7424
767c9d8efad3305093801604bd7893b5f117e823a3e7a373c225abaa959e911a
41961db05af286c8ec666643d9b3313df6f0669939ec490338db3583b2950d93
09ae9b32a791676c836355c3eede97aef1c2b199b07130d85cb7b9ac3960e971
```

可在AES – Symmetric Ciphers Online进行验证。

2 RC4

2.1 算法原理

算法定义一个长度为256的u8状态数组S，首先使用key来初始化

```
for i from 0 to 255
    S[i] := i
endfor
j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod keylength]) mod 256
    swap(S[i], S[j])
endfor
```

之后生成字节流，和输入流异或得到输出流。字节流生成方式为

```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap(S[i], S[j])
    output S[(S[i] + S[j]) mod 256]
endwhile
```

2.2 实现细节与优化

由于算法比较简单，主要进行的是循环展开。经过测试，初始化时进行4个一组的展开，生成字节流时进行16个一组的展开，这样效率比较高。

2.3 正确性与性能

我随机生成128字节的key与16Kbits的明文，共进行10次实验，每次将加密结果与OpenSSL的加密结果比较。

程序还会随机生成128字节的key与16Kbits的明文，进行1024次加密，从而较为准确地计算吞吐率。开启O2优化时，某次结果为

```
4230.565838 Mbps
```

不开启时，某次结果为

```
1379.786133 Mbps
```

2.4 测试样例

以16进制给出测试样例：

```
input:
dea66f04ea0501ea18c93188c71f4257fa6dde88035d8f10fa6bbd29328fa110
3610152016160a2edf3bb6a75bf8fe5565ddde683a6d7934d8365e0ac5ff1bfb
0f301b25462653256109ccbc01cb1266a8f0cfe25d4816367e744043735b3f83
8b5aa8d180fcf7e205c39e078eb06d36a03c18fe842f3402a3744617d0859a5b
df422d603e244244e7e04b7691b8ac31f5c52f79f4637c97d8c2aea847480326
8b3086c954c80d3ca958b23a115e6b06239b7f17fefbafd6bd5d7e04a6822b31
b2b1fa077a08432360f55d7153c8777763f78e62f23d38b09bb7b44139df72eb
916cf20b74352ed52a8b467e53bef5b7b58319a7c151575c080c9d41eb0f2d7c
7b1f87f055b5c57f400bfd94c9f24b7e76642637b57d93be8930ff753f2cf1ba
4c79aaa12e6f206f7b1e0344104ec386b2e9bd67665025f0802565bf51567a9d
cf243efe945f6d0f7d70538dbe161470ffd1d76622fd56a222bb627311dc11e1
004fdf94ae4ca32bbcf7b97a0dcdea0d9ec173c0bec963e084c55495a16576a1
b4553663a1d98e5dd047d7de14c1ebb3835e734127d622ab9b76403cdbb7de8f
0c14f2aeed810bbec8e39cdda4879027e503690cda8bb77501f7b2dcae906bbb
a45e6991df744fa757eb84fc72142357188c63f2171a67181219f4c0a9607b4d
```

```
bee4df9d592e44b01ac9ac8cddd0e4f55c47e774624f8c7468813412e1b05f9f
943e3ced6d809e87494a13271af71c773f04eba1537715bbf849cdd9f92d788e
6bb47bd835195f7e6473a57e6ac2f5a9c6e04a19585fd450a9a22aa2cfa2303a
57ac138cc5720a29e5b0a850729df9387e4451d6a325264cc750ef96f31fd14a
cbe4d69156e0ba3c90628c0200853a7ec98b546db17ab978cba80fbec8e00893
c4de241abedf564f41e25141688cbf3117139ec88e584159005017c8301f5cf4
fd800ebb5f650aa1475ce2afe8a2e1ffb57fc843d7099cd859b3a089d2fc7dcf
7d8b8bdcf0957d38f160e7d902c8d9b748a1fb1faa97f7034b988c1d9409ed11
9478ee850d6bbdffcba4d8cd6db185b55280d4fc17ccff62648b80f8946d0a29
e5f8aef2636bf12f0fcafc7c7b8131ce0106ca19d2ca7b3655fb2eea6838134d
30c140942c31c33bfbbfb87741e94542ef0f5bc1d9d7f72fd226193b5e2c888f
edc82319fae654f5a50c6ce6f6b129e5c184a79a5b9ec92ec4e269230ef1b2fb
bad514b4bb69a9607516476bc7705188f4f8235096ec7e5bcfe77eddd830d992
05ed46c056f020cc066737cdd78856cc80791c17659a723481f0125920ebec25
d832e52f2205fb286d32f644bb4c103bc52c522ac6c45f47b471a1d45c8df934
bfde63e2e45e0a519100954c4ca68711d2da3c999e9be0530c8127680e219cce
ff00b0e35eba34efbbca3b0770c319429d55db3bf0bc8efc3db6644cd7001ad6
00caba5f84ee4e3fb88a47284d606beab54625a502b4a1406a058c4105a61706
70d165f4c0b334783d7ba18adb0c7490529a35554ed695b8db21f9e0c710e637
e24b2ba2ff5f1a3cdabbc7b5c73b451ad57a6f235004db2b25d40cece5f223c7
3e4e693dae8379883f403e067c832051fe8f754e93507ab82586a40a78c7d1b6
163af3c4bd6d4cfcad8a03290e237b0cb3f05a4640d4ff655aa36fd36b408981
7a7d4538ea9134971c37c12a5b3c360e2c90546c6553d2bff7419262821ce3fc
99283483b9691ad5a0dbfffb17350943c65eb02bb182eaa8c37d0a4599ed4232
1576b5cfdfcfa480aba47bc2d985069fe3b6ca9438b53dfb324741cb3583fd4a
f9b219d981be592c62d4ef3b59f5db3caba5d1e35a0ede8c551f578aa254d59c
06ee7588acceb40ea2a34afc98253843ca0926251705b16d2409f7c75dcc6364
bbd8ec67a6a0764844c044dce57d20af8646d49e4b860b708f0237eccf9a508a
723cf118dd67602127a5fd0c221dbca8649046af16511fa554569223f0e2ad62
1f9e7afc06db1d2d801a3aa238f64a9c86914b9de26b4236c1d459b2b70614d6
a58fd2ab6aefd8ea09128c4108d6dd8f67292c4a946e805543da07fae01cd085
aba230159109ff9a1b8bdc2461b9b3c9e2df13764d93cc906dd38a4eef5ad39a
fc04af8d0dae282839044c9bbdff64a0de77162c0ae2bc78b647c6a5a199409e
9def2baa9e53d3d7571f72151fd6b5fd4dcb2958aee6d0642d9609ce2f496ccd
399877d7eb4aae436a215889f70d8645d8b09d86966deac303f491323dfeff76
96774d81c1fcc42b1d1cb414293b5902ebf688816373446667d599a4d3981b69
0f68ebd164affc81ccb196f5eceff7d7e6805849f39cb05a7149fe45e119aef1
8299c2e649be68156ffe0a5bed0232d3828a1d7526cdcf9816cdddf7e78be869
25aa4f6e69b783d8b58d34a38f667611f19386176055af76238c6e0a1856733d
01c2ab6a7a2e422fbb76d24bdd495ccedce3e53d3895b35b2121653978d87679
9b21e3154f25440b9c17567960b2473c952c79cec12d29e34e8f1cc667933f02
```

```
b4221704485c0fe473655dd317a40fadd0897b92b6a475043391cb9b240a9dd9
2db5dd7511ec598451b657685a66152aef90bca53531aa68c37503e77fa1c0ac
569d2167897aebda3042438aa858b598e9713d1ea3e786665c8a4ddc2b0e8881
abaae83524d30f551552dfbdab9455940693b2a97a380fd7c25cb3ed6a3b6e16
e5564b0a295a5f3ead3efc58d351ecd9e49e825fd6913699ede9865824f56e0a
4bb914751373b3c0b1af188401045de5a2df4479707a125e6398b6888d2492d9
dda64ef01901b1cab1c94fb2ceac97708cdce9fc56fb5aba9410422134d4fa11
7a4802934ab35dfb7cacad4a5944bbe520a4e177a03c31344c735581475092c1
key:
989454e247b1ddc45e8a0eb7cfc99cef6e7d660eb99742060a978751e7191280
ae6662f51840b976cac82d9991c989ff46ef0d00874f0691e78de3cea6f54e54
5cb14a74f103eabbcb17555de0de5c26cd6a2654b92ce6a0b9c96f60bebdb41a
6efe8e5f02781bcd8f702a6f4e87961bf1bc70aae9564ba21fba02dd77b7f8e6
output:
4336bd01042761380d3669809915b3e7976ca4b360a54756d5db27e0e502140b
5aac625ec6c7f5df597e931f31b2bc5f31545722883653d34ef7989acf258c23
31d966f5f0f7613a9f4e6f1e820f5c308939b68ee06c2d8efa9908fd28ffeb6d
fe8eae1a89d8c422445933520148eae68e97bea2feb3077ec0313cafb1e1277d
e16108d65e7c63f863935fc5aecdedffeba34aa651b14aaa068dae38810f2abf
42948042ecae0f26ac87cd32842b6bd87af57c6e47dff1395609e9f9df18242f
09afdea1f3685cff2298d8593e3434ea342402528f2a1d660a565653054b5e56
08c37bd48fadf4a96f4a5d724e84e7470477e550dffa6f50d8fa44a9897b0c7e
e50773704dc9ca0a023f379eda39f4496273536f6a38d0a89bb0f06bbf558f25
c8fd7962053c7a8d25a49fce41bf169f305a7bee460b62f4642a3f255c9e8019
3acf5225a66b21fc54bbc8896cef1b0ea373d243d65b9558457d7f0fd71dc7af
12377ae73b3c9d9d3f1f7530b731b9cf380cdae9e79e09cc6d4ed1505e10d96f
4b6acd8f23b1f872645122b1dd23a0c864e2ca9f6deb7f5caa196d226dcb5b5a
7d0e81f48a8fb88a1fcf91a1f22e2af92f22483675f8d52426a346fda045c737
60e97f66aaac15c560537b618c9395d8c5b58d8bb0b3a6657e8b60d650655fc8
d0aae7c0db638f5f7ba30faff637cbce6898466ef614c8c6e69cd69ac626a048
4f78869e44bf3ceeca3539ae2efce3c21636fd13817d5a2981e7e48148df3d40
444b3d4b4102e8a070d8bb36b3995b40c5474b1c493c61b49ca891da89afa727
793fb4284ac6925cf066ec5377ac1b9922ddeaf2c4f9c6017be4bd1203eb7948
c066b931bbff4495cc9a45afab727ba9897fdcdbf732ea11ee68c22b01e848da
fb1c5a67ecf94980d3c90341805ff8112eda46b6822dcb2b3352cb73c9f84646
58947550fbc92e80bbdbc57f5d46a09fddc4c0bf2d473a7ee55574a46045ff7e
428143aeda8cc32f329fbd60c13077adf9b2c89d4422737b1f1c2c4b36afc8f4
f758f947b1391569b5813a0837535d3ca7e8b6d9bcaccfbc3aa32728db4b51bf
951bbf445a495b9724b261f195ba21f70c0f9ff06927cca467b28fd79412fb59
488fa85c4f18b5ecf3016849c4d73bbdbb928ee01b6e6e797a5f07f0723633fb
a139b129de0ae96b4f102c0dabaab324e2a9979f3b6a9ca9c27da5a0a1ee38a4
```

```
3720849efaedc14c96cd430d1f851316104fbb60e78aa521f8ecf3d03b00df79
62ad2f0e318126b9cffa4eb89767c9639bf6bcc2f740dc77a34d971d531bb2fa
59ca4d0f229aa040f406c773046cebe03adcae26517606c3b64443fc0e450343
3b65166558beb08f280f4d054c9247c2d38b2776d8fd39390e423eb8833c1605
4016f398c14b55d4f47d68181b9b5e8ca2b73e0bbfbab92e7557f96b09cad80a
63ff95f1d0bb088ead44dcff0770259fa21900f1220b047760654a9ba1e62875
d409659ba768466035a142d62c50ff8b16435abb5e98f0147dffcb5c770c7106
7a70eced28f5d29d4a19b2d2dfcd6780f07d07d4162c32f195408be2bc417373
01dc4128a23076b7feba95df115563aecf43fe120ccfbade1620c0e535bb40e6
d81fa2b7ae401daa4ccfe802c0d8310e00f4156d1169bd9914fbb5b3d3ed3322
3e5d686b778e9c6d95f52353e306daadb0452cfa7f98805203eca367a32eae41
b6d398ba816ec3ff3011d6709065ef6de1783a0433e72463907f0ef3e44d57a9
62a689eca05ed64724d7e63bac12c2424f1fbfcdc7ecad7fc822af24f666c0e8
34e54b1a466fd94209b9676028414f95c6f89c3d9de77b60732270626dd191a9
33d6a4e40400cd7b67af4a4a25968b3f5c9a8197c9f2fa9443241985c196a760
19f44399b7e4469ff12891c65fd690ba10507c9cd6d8ebe4dab4dbc727f7e0a4
5d894d36f6f85d2ad720f8d66de8b686560d4712082f9c6952f63643a2599897
95037fbbbe55f2623f4632d15739013c62d6baf8642f1e2dc206dd5ad4a1e5c1
89366830a747af12cbeb7bb3336f3884358c2ccb0277207afeffa77b209b6d85
838decafdfa95933d358ab359e578bb8c28bee449e2d70fb83e2e37ee6c4ac70
ab6d66ea10116c23390d013b102e4a59b77c4b6bc0820e0a16d1107975378c4d
c0b35c66a7659ed53315434804bd9db2523dab1378a62870a6ed495f57c64f0b
011f037a6b28f1acf7889a3d6d8c3747c8b6812fc3da0800d0615501650d7082
ee44b5474e4d61602a539114552a86f3aa2616f81c0a199b4278c1a7856bb528
928da589e05bf5d3cbf24ea09fbc9ca2e8847f1f77dcecc3fece7f9bf2fe96d5
a723e5d3328d7e04df21efff2c2e89ed7700695718a2f2395a3642c1e3432555
792f070b90f5c34f1c0fe1632689c1fbadd3bc81b80da8167c2d0cc875157e3f
6c3097c95eba50c9dae40cd391e111ae0130491615d589fbeae92e27ee3f52a6
85141d9f0028f3793e6280bfad9738965ea26dfb9cbaf0f634a951cc2434bf54
b2ab7451e7ad251946c4497421f085beac2b63a15281611daa9b8ba48847f6dd
2e791c39ea4a3d12084a0ff46a7e614e3ef851942a4574237f83053e506b9ff2
fe99c7d4c8c3c4a42fca24b776e8895175098732149 38bc8672085b1d0fc8e5b
73b3bbfa8085c98f289d033db13e6da4a62be20bf6fb48922ed44eb0b6ab65b4
aeaf50286c66ac9bcca6ab1e6ee138aa73df113dd63c00f4cbc3391ea8fec486
0b6c78547173449eba01fd2480325de525b4486a480a819b6e75e0ca6c94b017
fa85b3eb28e3a4339218f9abe55f4e9bd79708f38cdcc916c4ac2c6c0da0a54e
bffeb60aec9b7a64c0b3334c67110e02fe88cb8d221103c067bfb3e845c967c2
```

可在RC4 – Symmetric Ciphers Online进行验证。

3 SHA-3-256

3.1 算法原理

SHA-3-256可以用SPONGE结构表示为：

SHA3-256(M) = SPONGE[KECCAK-p[1600, 24], pad10*1, 1088] (M || 01, 256)

即对于明文M，在末尾添加01后，使用模式10. . . 01填充为1088位的整数倍。初始化S为1600位0，对于每个1088位的分组，0填充至1600位，与当前S异或，经过变换KECCAK-p[1600, 24]后得到新的S。最后取S的前256位作为Hash值。

KECCAK-p[1600, 24]的过程为，将1600位的输入转化为$[5, 5, 64]$比特的三维状态数组A，经过24轮$\theta \to \rho \to \pi \to \chi \to \iota$的变换得到新的状态$A'$，再转化为1600比特的输出。

3.2 实现细节与优化

实际实现中，输入以字节为单位，因此添加01两位后必然需要填充至136字节的整数倍。对于字节操作而言，这等价于在原先`u8`数组末尾添加0x06 0x00 . . . 0x00 0x80（填充超过1字节）或0x86（填充1字节）。

状态数组可利用64位处理器的优势，使用5x5的`u64`数组。

对于θ, π, χ，进行尽可能的循环展开。对于ρ，每个位置的偏移量$(t + 1)(t + 2)/2$可预计算并查表。对于ι，预计算每一轮的RC。

3.3 正确性与性能

首先测试了对空字符串的Hash：

sha3_256("") = "a7ffc6f8bf1ed76651c14756a061d662f580ff4de43b49fa82d80a4b80f8434a"

之后随机生成16Kbits的数据，共进行10次实验，每次将Hash结果与OpenSSL的Hash结果比较。

程序还会随机生成16Kbits的数据，进行1024次Hash，从而较为准确地计算吞吐率。开启O2优化时，某次结果为

```
1562.500000 Mbps
```

不开启时，某次结果为

```
243.694407 Mbps
```

3.4 测试样例

```
input:
b39ae3b9e965481ab43550cff787d870c66d37fffaf6fb63f87a82cdb450f867
ebdb20d44068eef49e3ec395c59b058b083d8b033386662b01e9f8b539f01c24
cb3df80ba5e6ff4325c2d9ea5dde76661b01694e87cf7988b8713df2615a162c
970f373cf536801af8590556377bbc537c25a103f41b8cad8cc99fee23b51aba
c452f7ba8877d481d0d9d70754935ad0b8fcd4ac176059a329f8914daeac0772
fefe2c8675010745dade4d2f71a7ff29a3d3d6ba332f5e5d28efaad69bb14899
b075202576276b5006b87f775f7fa1035277bd86a61be3ce0b8da4a63eed40ee
626014d8877f288d37a8059627a699791d57ffc372e2927d6f3624ae23649c85
c4b05d4b2f86d9662edefd558496cea1edce6460b0f6dd202d01ce50656ad629
1b33754ab94eb1e72cae3cb0440b5132d9b59289ac6fa9d9717729d6e2ff00fd
337547ecc3f8d4efa6109feb1bf01df4a5af7e511e272a8f9f54668153667e86
dbc5739ebe478d64572c4f731c6c67c11be5133a0d3dc9ac912f2de595ab6b70
70de0e2e259b937dc7e2f0e34f57a56a3db8a44af56ef6879d236c33ced7a33e
b6b26ddb4d005815e248f831a09d9cdd5540274bae1dd24c403e7f0e15224ccb
d4b9a722b9ff379c482fcde8cd69c522aaec6d58093fa4497d23579346a35e1a
5d053c160573b24da3803570e9fa9293e600ecef3f9038bdb48f50fa32ae148f
b451a6b9c4580667d83bd7c2356a551b6a410aa9d2426686d1b6800365949319
e539d2aa91d8116a13e92c48538163bdc36d6695afcd1b80839b83e82f160115
4fd3bfe1abd04bbeb977060cf869c9bbd6305085fd6b0580068969369f6a4bef
3e0ad0e9da1ba89492aea08a186a46ee9a96749702791708028 03ea2eb899129
9361126e7cba020e69a298810cde6fa675e33d775d557f5fd5be01c04792e9db
f3fc496fb64b7d1fed16a0faf410a069f3dee0503360b0081eb1c96544b24037
ae89a765d42484c23a25bc2f355c98283a79796dd92976f7da3f5c1ef19d56a0
26fd05fb2189bd5cae798be3d5230c109c857d75aef36c8832c9a72466fdc48c
fac9871b52447701bd02e49326f0a3c27520382314a4ac466d536ad3502e604a
f7e7654a2cdd4be9df2f7c05201fc8954000b954a4659a12b805e5083345522b
2db7755994c04274efbf790fde41a51e415e72e6c30df87b12dd834523d57050
8ce5a921a5eb9595aa0ea4895049a791a71a776a276fe5394d687e703defc0ca
d469eb7a54800fff8eb388defd2f70a449e70f7057f4a9a45d28149a17d464eb
3d4f6591cf74905e28183c2548acc99194d802ebcdab8f2ad3a3c4ea7729d6b4
783b4548b0d6a6d8eee2fd368fc6c8239fca0e6c759d9649405a33b783096bfc
45b044f586eacd75cccaab5b90737e2f3d8c9bb32931fc698c2f200f398b0b7e
3c4f73c2394037060ae3619a56e0ca946c6547969743ff23722032abab3e29e7
8d9caac7dce1cde6c42e811b0e4baf7bb0f6114739106aab309d57dcdb80c368
1d6d2ff94ffce0132b612e39acddb45cd3c5a40cd60eb806ab0fe2868fa6efac
131ea6621b867646e7a47f938234ef55f99362cfa21ad64d29b8d4b85ec36572
e10bd4fc914a4278efc20b71f6fac6ef8e28bf3042957d6b4d5124ac14891ef6
94f2f2253d359d2cf7a89deda263dc308c9b60ce30de3a7e2f5e2a44e7483a7b
```

```
3a2ca077613da358e5404587a422b830bd18feeef6386c2696966a7ddea4f818
d0989032d5338aba74d04218f2fa48af12469d097f092f159f99937d3d8b960d
24263ff959cab4cd9af6e58cf02d3b0274d90bf3e23a0882d39bff1027951e4b
bb5d441527f8e2c1eec84ddef589e16962ec5c442765c6fa00c60b275b297217
86b72caeaf0e6f9ed6bd7ccc465d35a84a92ec71f7b36bf779761fd49f91eb26
4817d4f8264396fc0012c84670feeeba90db2b878e967e070d9ddbac2fc7d277
dea66f04ea0501ea18c93188c71f4257fa6dde88035d8f10fa6bbd29328fa110
3610152016160a2edf3bb6a75bf8fe5565ddde683a6d7934d8365e0ac5ff1bfb
0f301b25462653256109ccbc01cb1266a8f0cfe25d4816367e744043735b3f83
8b5aa8d180fcf7e205c39e078eb06d36a03c18fe842f3402a3744617d0859a5b
df422d603e244244e7e04b7691b8ac31f5c52f79f4637c97d8c2aea847480326
8b3086c954c80d3ca958b23a115e6b06239b7f17fefbafd6bd5d7e04a6822b31
b2b1fa077a08432360f55d7153c8777763f78e62f23d38b09bb7b44139df72eb
916cf20b74352ed52a8b467e53bef5b7b58319a7c151575c080c9d41eb0f2d7c
7b1f87f055b5c57f400bfd94c9f24b7e76642637b57d93be8930ff753f2cf1ba
4c79aaa12e6f206f7b1e0344104ec386b2e9bd67665025f0802565bf51567a9d
cf243efe945f6d0f7d70538dbe161470ffd1d76622fd56a222bb627311dc11e1
004fdf94ae4ca32bbcf7b97a0dcdea0d9ec173c0bec963e084c55495a16576a1
b4553663a1d98e5dd047d7de14c1ebb3835e734127d622ab9b76403cdbb7de8f
0c14f2aeed810bbec8e39cdda4879027e503690cda8bb77501f7b2dcae906bbb
a45e6991df744fa757eb84fc72142357188c63f2171a67181219f4c0a9607b4d
bee4df9d592e44b01ac9ac8cddd0e4f55c47e774624f8c7468813412e1b05f9f
943e3ced6d809e87494a13271af71c773f04eba1537715bbf849cdd9f92d788e
6bb47bd835195f7e6473a57e6ac2f5a9c6e04a19585fd450a9a22aa2cfa2303a
57ac138cc5720a29e5b0a850729df9387e4451d6a325264cc750ef96f31fd14a
cbe4d69156e0ba3c90628c0200853a7ec98b546db17ab978cba80fbec8e00893
output:
ef3534300ae30d3ee96f838e1fd2a2052d98ec84d80624afde062038c4f1c814
```