Block Ciphers

1. We show that encryption and decryption steps are inverse in every round. A round of a Feistel network is defined by

$$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus f_{k_i}(R_{i-1}).$$

Hence one recovers R_{i-1} by $R_{i-1} = L_i$, and L_{i-1} by

$$L_{i-1} = R_i \oplus f_{k_i}(L_i) = L_{i-1} \oplus f_{k_i}(R_{i-1}) \oplus f_{k_i}(R_{i-1})$$

using the formulas for decryption. Decryption takes (R_r, L_r) as input, and r Feistel rounds yield (R_0, L_0). The final permutation recovers the plaintext (L_0, R_0).

2. The hexadecimal numbers `01`, `02` and `03` correspond to the polynomials 1, x and $1 + x$ mod $x^8 + x^4 + x^3 + x + 1$, respectively. Obviously, the inverse of 1 is 1. Furthermore, `8D` corresponds to $x^7 + x^3 + x^2 + 1$ and `F6` corresponds to $x^7 + x^6 + x^5 + x^4 + x^2 + x$. Multiplying these polynomials with x and $1 + x$, respectively, and reducing mod $x^8 + x^4 + x^3 + x + 1$ shows that the inverse of `02` is `8D` and the inverse of `03` is `F6`. Then $S_{RD}(\texttt{01}) = A \cdot \texttt{01} + b = \texttt{7C}$, $S_{RD}(\texttt{02}) = A \cdot \texttt{8D} + b = \texttt{77}$ and $S_{RD}(\texttt{03}) = A \cdot \texttt{F6} + b = \texttt{7B}$.

3. Since $(x + y)^2 = x^2 + 2xy + 2y \equiv x^2 + y^2 \bmod 2$, the functions $f(x) = x^2$, $f(x) = x^4$, x^8, $\ldots$ are $GF(2)$-linear. However, if n is not a power of 2 then $f(x) = x^n$ is not $GF(2)$-linear. Consider $n = 254$. Since $GF(2^8)^*$ is a group of order 255, we have $x^{255} = 1$ and thus $x^{254} = x^{-1}$ (for $x \neq 0$). The function $f(x) = x^{254}$ is not linear: $\texttt{01} \oplus \texttt{02} = \texttt{03}$, but $\texttt{01}^{-1} \oplus \texttt{02}^{-1} = \texttt{01} \oplus \texttt{8D} = \texttt{8C}$ is not equal to $\texttt{03}^{-1} = \texttt{F6}$.

4. One has $S_{RD}(a) = y = Aa^{-1} + b$ for $a \in GF(2^8)^*$. This gives $y - b = Aa^{-1}$ and

$$a^{-1} = A^{-1}(y - b) = A^{-1}(y + b).$$

Note that $+1 = -1$ over extension fields of $GF(2)$. One computes:

$$A^{-1} = \begin{pmatrix} 0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \\ 0&0&1&0&0&1&0&1 \\ 1&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1 \\ 1&0&1&0&0&1&0&0 \end{pmatrix}, \quad A^{-1}b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Therefore, $S_{RD}^{-1}(y) = a = (A^{-1}(y + b))^{-1}$ for $y \neq b$. So the inverse SubByte operation is given by an affine map followed by multiplicative inversion in $GF(2^8)$. For $y = b$, one sets $S_{RD}^{-1}(b) = \texttt{00}$.

5. Let $a = (a_7 a_6 \ldots a_1 a_0)$ be an 8-bit string. Then $\texttt{01} \cdot a = a$. Now consider multiplication by `02`. Since `02` corresponds to the polynomial x, we have $\texttt{02} \cdot (0a_6 \ldots a_1 a_0) = (a_6 \ldots a_1 a_0 0)$. If $a_7 = 1$ then we use the relation $x^8 \equiv x^4 + x^3 + x + 1$ and obtain $\texttt{02} \cdot (1a_6 \ldots a_1 a_0) = (a_6 \ldots a_1 a_0 0) \oplus (00011011)$. Multiplication by `03` can be implemented using the equation $\texttt{03} \cdot a = (\texttt{02} \oplus \texttt{01}) \cdot a = (\texttt{02} \cdot a) \oplus a$.

The MixColumns M matrix only contains the entries 01, 02 and 03. Hence multiplication by M can be efficiently implemented using the above binary operations.

We use SageMath to compute the inverse MixColumn matrix M^{-1} over $GF(2^8)$.

```
K.<a>=GF(2^8, name='a', modulus=x^8+x^4+x^3+x+1)
M=matrix(K,[[a,1+a,1,1],[1,a,1+a,1],[1,1,a,1+a],[1+a,1,1,a]])
M.inverse()
  [a^3 + a^2 + a  a^3 + a + 1    a^3 + a^2 + 1  a^3 + 1        ]
  [      a^3 + 1  a^3 + a^2 + a  a^3 + a + 1    a^3 + a^2 + 1 ]
  [a^3 + a^2 + 1  a^3 + 1        a^3 + a^2 + a  a^3 + a + 1   ]
  [  a^3 + a + 1  a^3 + a^2 + 1  a^3 + 1        a^3 + a^2 + a ]
```

In hexadecimal notation, the inverse MixColumn operation is given by multiplication with the matrix

$$M^{-1} = \begin{pmatrix} \texttt{0E} & \texttt{0B} & \texttt{0D} & \texttt{09} \\ \texttt{09} & \texttt{0E} & \texttt{0B} & \texttt{0D} \\ \texttt{0D} & \texttt{09} & \texttt{0E} & \texttt{0B} \\ \texttt{0B} & \texttt{0D} & \texttt{09} & \texttt{0E} \end{pmatrix}.$$

The first four bits of all entries of M^{-1} are zero. This allows an efficient implementation of multiplication with this matrix.

6. Use SageMath to verify that all square sub-matrices are nonsingular.
7. We give a high-level description of the AES decryption function f_k^{-1} in pseudocode:

```
Inverse-Rijndael(State, CipherKey)
{
   KeyExpansion(CipherKey, ExpandedKey)
   // Inverse of Final Encryption Round
   AddRoundKey(State,ExpandedKey[Nr])
   Inverse ShiftRows(State)
   Inverse  SubBytes(State)
   for(i = Nr - 1; i >= 1 ; i--)  {
      // Inverse of Encryption Round i
      AddRoundKey(State,ExpandedKey[i])
      Inverse MixColumns(State)
      Inverse ShiftRows(State)
      Inverse SubBytes(State)
    }
    AddRoundKey(State,ExpandedKey[0])
}
```

8. a) The initial words are $W_0 =$ 01 00 00 00, $W_1 =$ 00 00 00 00, $W_2 =$ 00 00 00 00 and $W_3 =$ 00 00 00 00. Since $sh(W_3) =$ 00 00 00 00 and $S($00$) =$ 63, the first round of key expansion gives:

$$W_4 = (\texttt{01 00 00 00}) \oplus (\texttt{63 63 63 63}) \oplus (\texttt{01 00 00 00}) = \texttt{63 63 63 63}$$
$$W_5 = W_6 = W_7 = \texttt{63 63 63 63}$$

Therefore, $k_0 =$ `01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00` and $k_1 =$ `63 63 63 63 63 63 63 63 63 63 63 63 63 63 63 63`.
b) The SageMath code given prints out the round keys k_0, k_1, ..., k_{10} and the output block:

```
01000000000000000000000000000000
63636363636363636363636363636363
9A989898F9FBFBFB9A989898F9FBFBFB
91979701686C6CFAF2F4F4620B0F0F99
EFE1792A878D15D07579E1B27E76EE2B
C7C988D940449D09353D7CBB4B4B9290
5486E86A14C2756321FF09D86AB49B48
9992BA688D50CF0BACAFC6D3C61B5D9B
B6DEAEDC3B8E61D79721A704513AFA9F
2DF3750D167D14DA815CB3DED0664941
28C8F67D3EB5E2A7BFE951796F8F1838
Output: 09F292268EDAAB9D6308EF2723590C18
```

9. The *diffusion* operations that spread small changes throughout the complete state of 128 bits would be missing. AES would be reduced to addition of the round key and the S-Box transformation. Changing *one plaintext byte* would change only *one ciphertext byte*. Encryption of each of the 16 plaintext bytes in a block would be independent of each other. The output of 256 chosen input blocks `00...00`, `01...01`, ..., `FF...FF` were sufficient to determine the output of any input block. The security of AES would be significantly reduced. The modified block cipher is not a pseudorandom permutation, since an adversary can easily distinguish it from a random permutation. An adversary could choose two input blocks which differ only in one byte. If the output blocks differ also only in one byte, then the modified block cipher was used, since a random permutation would not have this property. The adversary therefore wins the *prp* experiment.

   If only the ShiftRows operations were missing, then the transformation of the four state columns would be independent of each other. Similarly, if the MixColumns operations were missing, then the transformation of the four state rows would be independent of each other. On the same lines as above, an adversary can distinguish the modified block ciphers from a random permutation.

10. All other operations are affine and their composition is also affine. Thus the encryption map would be affine altogether and *confusion* would be missing. A known plaintext attack with 129 plaintext-ciphertext pairs can be conducted. The attack requires only linear algebra and is therefore very efficient. The modified cipher is not a pseudorandom permutation: an adversary can easily find out whether a function is affine or not if they can choose input blocks and get the corresponding output blocks.

11. The nonlinearity of encryption and decryption is essential, since otherwise the cipher would be completely insecure. The nonlinearity of the key schedule is desirable.

12. $W_0, \ldots, W_7$ and k_0, k_1 are all-zero. Since $RC_1 =$ `01` and $S_{RD}($`00`$) =$ `63`, we obtain

$$W_8 = W_9 = W_{10} = W_{11} = \texttt{62 63 63 63}$$

and therefore

$$k_2 = \texttt{62 63 63 63 62 63 63 63 62 63 63 63 62 63 63 63}.$$

Furthermore, $S_{RD}(\texttt{62}) = \texttt{AA}$, $S_{RD}(\texttt{63}) = \texttt{FB}$ and hence

$$W_{12} = W_{13} = W_{14} = W_{15} = \texttt{AA FB FB FB}.$$

Then

$$k_3 = \texttt{AA FB FB FB AA FB FB FB AA FB FB FB AA FB FB FB}.$$