Exercise 5

Question 1

Consider the following five events. What is the order of these events from most likely to least likely?

1. Correctly guessing a random 128-bit AES key on the first try.
2. Winning a lottery with 1 million contestants (the probability is $1/10^6$).
3. Winning a lottery with 1 million contestants 5 times in a row (the probability is $(1/10^6)^5$).
4. Winning a lottery with 1 million contestants 6 times in a row.
5. Winning a lottery with 1 million contestants 7 times in a row.

Question 2

Suppose that using commodity hardware it is possible to build a computer for about $200 that can brute force about 1 billion AES keys per second. Suppose an organization wants to run an exhaustive search for a single 128-bit AES key and was willing to spend 4 trillion dollars ($\$4 * 10^{12}$) to buy these machines (this is more than the annual US federal budget). How long would it take the organization to brute force this single 128-bit AES key with these machines? Ignore additional costs such as power and maintenance.

1. More than a million years but less than a billion (10^9) years.
2. More than a month but less than a year.
3. More than a 100 years but less than a million years.
4. More than a day but less than a week.
5. More than a billion (10^9) years.

Question 3

Let $R := \{0, 1\}^4$ and consider the following function $F : R^5 \times R \to R$ defined as follows:

$$F(k,x) := \begin{cases} t = k[0] \\ \text{for } i = 1 \text{ to } 4 \text{ do} \\ \qquad \text{if } (x[i-1] == 1) \quad t = t \oplus k[i] \\ \text{output } t \end{cases}$$

That is, the key is

$$k = (k[0], k[1], k[2], k[3], k[4]) \in R^5$$

and the function at, for example, 0101 is defined as $F(k, 0101) = k[0] \oplus k[2] \oplus k[4]$.

For a random key k unknown to you, you learn that

$$F(k, 0110) = 0011 \quad \text{and} \quad F(k, 0101) = 1010 \quad \text{and} \quad F(k, 1110) = 0110.$$

What is the value of $F(k, 1101)$? Note that since you are able to predict the function at a new point, this function is insecure.

Problems

4.1. Since May 26, 2002, the AES (Advanced Encryption Standard) describes the official standard of the US government.

1. The evolutionary history of AES differs from that of DES. Briefly describe the differences of the AES history in comparison to DES.
2. Outline the fundamental events of the developing process.
3. What is the name of the algorithm that is known as AES?
4. Who developed this algorithm?
5. Which block sizes and key lengths are supported by this algorithm?

4.2. For the AES algorithm, some computations are done by Galois Fields (GF). With the following problems, we practice some basic computations.

Compute the multiplication and addition table for the prime field $GF(7)$. A multiplication table is a square (here: 7×7) table which has as its rows and columns all field elements. Its entries are the products of the field element at the corresponding row and column. Note that the table is symmetric along the diagonal. The addition table is completely analogous but contains the sums of field elements as entries.

4.3. Generate the multiplication table for the extension field $GF(2^3)$ for the case that the irreducible polynomial is $P(x) = x^3 + x + 1$. The multiplication table is in this case a 8×8 table. (Remark: You can do this manually or write a program for it.)

4.4. Addition in $GF(2^4)$: Compute $A(x) + B(x) \bmod P(x)$ in $GF(2^4)$ using the irreducible polynomial $P(x) = x^4 + x + 1$. What is the influence of the choice of the reduction polynomial on the computation?

1. $A(x) = x^2 + 1, B(x) = x^3 + x^2 + 1$
2. $A(x) = x^2 + 1, B(x) = x + 1$

4.5. Multiplication in $GF(2^4)$: Compute $A(x) \cdot B(x) \bmod P(x)$ in $GF(2^4)$ using the irreducible polynomial $P(x) = x^4 + x + 1$. What is the influence of the choice of the reduction polynomial on the computation?

1. $A(x) = x^2 + 1, B(x) = x^3 + x^2 + 1$
2. $A(x) = x^2 + 1, B(x) = x + 1$

4.6. Compute in $GF(2^8)$:

$$(x^4 + x + 1)/(x^7 + x^6 + x^3 + x^2),$$

where the irreducible polynomial is the one used by AES, $P(x) = x^8 + x^4 + x^3 + x + 1$. Note that Table 4.2 contains a list of all multiplicative inverses for this field.

4.7. We consider the field $GF(2^4)$, with $P(x) = x^4 + x + 1$ being the irreducible polynomial. Find the inverses of $A(x) = x$ and $B(x) = x^2 + x$. You can find the inverses either by trial and error, i.e., brute-force search, or by applying the Euclidean algorithm for polynomials. (However, the Euclidean algorithm is only sketched in this chapter.) Verify your answer by multiplying the inverses you determined by A and B, respectively.

4.8. Find all irreducible polynomials

1. of degree 3 over $GF(2)$,
2. of degree 4 over $GF(2)$.

The best approach for doing this is to consider all polynomials of lower degree and check whether they are factors. Please note that we only consider monic irreducible polynomials, i.e., polynomials with the highest coefficient equal to one.

4.9. We consider AES with 128-bit block length and 128-bit key length. What is the output of the first round of AES if the plaintext consists of 128 ones, and the first subkey (i.e., the first subkey) also consists of 128 ones? You can write your final results in a rectangular array format if you wish.

4.10. In the following, we check the *diffusion properties* of AES after a single round. Let $W = (w_0, w_1, w_2, w_3) = (\texttt{0x01000000}, \texttt{0x00000000}, \texttt{0x00000000}, \texttt{0x00000000})$ be the input in 32-bit chunks to a 128-bit AES. The subkeys for the computation of the result of the first round of AES are $W_0, \ldots, W_7$ with 32 bits each are given by

$$\begin{aligned}
W_0 &= (\texttt{0x2B7E1516}),\\
W_1 &= (\texttt{0x28AED2A6}),\\
W_2 &= (\texttt{0xABF71588}),\\
W_3 &= (\texttt{0x09CF4F3C}),\\
W_4 &= (\texttt{0xA0FAFE17}),\\
W_5 &= (\texttt{0x88542CB1}),\\
W_6 &= (\texttt{0x23A33939}),\\
W_7 &= (\texttt{0x2A6C7605}).
\end{aligned}$$

Use this book to figure out how the input is processed in the first round (e.g., S-Boxes). For the solution, you might also want to write a short computer program or use an existing one. In any case, indicate all intermediate steps for the computation of *ShiftRows*, *SubBytes* and *MixColumns*!

1. Compute the output of the first round of AES to the input W and the subkeys $W_0, \ldots, W_7$.
2. Compute the output of the first round of AES for the case that *all* input bits are zero.
3. How many output bits have changed? Remark that we only consider a single round — after every further round, more output bits will be affected (*avalanche effect*).

4.11. The MixColumn transformation of AES consists of a matrix–vector multiplication in the field $GF(2^8)$ with $P(x) = x^8 + x^4 + x^3 + x + 1$. Let $b = (b_7x^7 + \ldots + b_0)$ be one of the (four) input bytes to the vector–matrix multiplication. Each input byte is multiplied with the constants 01, 02 and 03. Your task is to provide exact equations for computing those three constant multiplications. We denote the result by $d = (d_7x^7 + \ldots + d_0)$.

1. Equations for computing the 8 bits of $d = 01 \cdot b$.
2. Equations for computing the 8 bits of $d = 02 \cdot b$.
3. Equations for computing the 8 bits of $d = 03 \cdot b$.

Note: The AES specification uses "01" to represent the polynomial 1, "02" to represent the polynomial x, and "03" to represent $x + 1$.

4.12. We now look at the gate (or bit) complexity of the MixColumn function, using the results from problem 4.11. We recall from the discussion of stream ciphers that a 2-input XOR gate performs a $GF(2)$ addition.

1. How many 2-input XOR gates are required to perform one constant multiplication by 01, 02 and 03, respectively, in $GF(2^8)$.
2. What is the overall gate complexity of a hardware implementation of one matrix–vector multiplication?
3. What is the overall gate complexity of a hardware implementation of the entire Diffusion layer? We assume permutations require no gates.

4.13. We consider the first part of the ByteSub operation, i.e, the Galois field inversion.

1. Using Table 4.2, what is the inverse of the bytes 29, F3 and 01, where each byte is given in hexadecimal notation?
2. Verify your answer by performing a $GF(2^8)$ multiplication with your answer and the input byte. Note that you have to represent each byte first as polynomials in $GF(2^8)$. The MSB of each byte represents the x^7 coefficient.

4.14. Your task is to compute the S-Box, i.e., the ByteSub, values for the input bytes 29, F3 and 01, where each byte is given in hexadecimal notation.

1. First, look up the inverses using Table 4.2 to obtain values B'. Now, perform the affine mapping by computing the matrix–vector multiplication and addition.
2. Verify your result using the S-Box Table 4.3.
3. What is the value of $S(0)$?

4.15. *Derive* the bit representation for the following round constants within the key schedule:

- $RC[8]$
- $RC[9]$
- $RC[10]$

4.16. The minimum key length for the AES algorithm is 128 bit. Assume that a special-purpose hardware key-search machine can test one key in 10 ns on one processor. The processors can be parallelized. Assume further that one such processor costs \$10, including overhead. (Note that both the processor speed and the prize are rather optimistic assumptions.) We assume also that Moore's Law holds, according to which processor performance doubles every 18 months.

How long do we have to wait until an AES key search machine can be built which breaks the algorithm on average in one week and which doesn't cost more than \$1 million?

4.17. For the following, we assume AES with 192-bit key length. Furthermore, let us assume an ASIC which can check $3 \cdot 10^7$ keys per second.

1. If we use 100,000 such ICs in parallel, how long does an average key search take? Compare this period of time with the age of the universe (approx. 10^{10} years).
2. Assume Moore's Law will still be valid for the next few years, how many years do we have to wait until we can build a key search machine to perform an average key search of AES-192 in 24 hours? Again, assume that we use 100,000 ICs in parallel.