

Information Security

Assignment 4

Group 31

Stijn Kammer (s4986296) & Ramon Kits (s5440769)

October 24, 2022

EXERCISE 1

1. Consider a CRC that uses the divisor 10011. Find two collisions with 10101011, that is, find two other data values that produce the same CRC checksum as 10101011.

First we have to compute the CRC checksum of 10101011 using the divisor 10011.

```
101010110000
10011
-----
 11001
 10011
-----
 10101
 10011
-----
 11000
 10011
-----
 10110
 10011
-----
 1010
```

The CRC checksum is therefore 1010. Now we have to find two other data values that produce the same CRC checksum. The first place where zeros have been added has to be the same in order to find a collision. In this case, we have to find a value that produces a step where 11000 is used. That is, the first step where CRC bits are being added. So to make it easy to find a collision, we can just use 110 as value because this generates the step we need instantly. Also, we can use one of the other intermediate steps to find a collision, e.g. 10101. This way of finding collisions is very limited, only steps of the original value can be used. To find a collision with a value of the same length as the original, we can also use the divisor as the first part of the value so when a XOR is performed on that part, the first 5 bits will get discarded in a sense. At this point, we are left with 3 bits that have to be 110 to get the desired step we want. So the value we end up with is 10011110.

We now have found three collisions using two different methods. Collisions can also be found in more sophisticated ways, but with the simplicity of the question in mind, only two ways are discussed here. Our found values are 110, 10101 and 10011110. Which are more than two, but it makes it possible to demonstrate multiple ways of finding collisions.

2. Consider a CRC that uses the divisor 10011. Suppose the data value is 11010110. Trudy wants to change the data to 111****, where "*" indicates that she doesn't care about the bit in that position, and she wants the resulting checksum to be the same as for the original data. Determine all data values Trudy could choose. For this question, we have to find all data values that Trudy could choose. Now we cannot

use the same method as in the previous question because we want all possible data values that start with 111 and have a length of 8 bits. Trudy has the opportunity to generate any of $2^5 = 32$ possible values for the remaining 5 bits, of which only a select amount create the same CRC checksum as the original data value. Brute forcing all options is possible but not scalable. So we have to find a way to compute the possible values without having to check all of them.

We have to solve for the unknown bits in the following checksum calculation steps.

```

111*****0000
10011
 11...
10011
 1...
10011
  ...
  ...
 11000
10011
 10110
10011
   1010

```

For the one step before we reach 11000, we can say that there are a small group of possibilities for the unknown bits.

We can fill in the empty step in the only possible way, and we can remove that step all together. That gives us the two only solutions.

Filling these values in the checksum calculation steps, we get the following results.

111001110000		111101000000
<u>10011</u>		<u>10011</u>
11111		11011
<u>10011</u>		<u>10011</u>
11001		10000
<u>10011</u>	and	<u>10011</u>
10101		11000
<u>10011</u>		<u>10011</u>
11000		10110
<u>10011</u>		<u>10011</u>
10110		1010
<u>10011</u>		
1010		

The only two answers are **11100111** and **11110100**.

EXERCISE 2

EXERCISE 3

EXERCISE 4

EXERCISE 5

EXERCISE 6

EXERCISE 7

EXERCISE 8

EXERCISE 9

EXERCISE 10