CS 427, Assignment 1

Cody Malick
`malickc@oregonstate.edu`

January 20, 2017

1

We can find the value of k by xoring m and c

$m \oplus c$:

110101100010110
100111011011111

010010111001001

Then, $k = 010010111001001$. We can use this info to then xor the value of m' with the value of k:

$m' \oplus k$:

010010111001001
001000000101111

011010111100110

Then our encrypted m', lets call it $c' = 011010111100110$

2

We can show that these two libraries are not interchangable by inspecting the distribution of each library:

$\mathcal{L}_1$ has a probability distribution of $\{0,1\}^\lambda - 0^\lambda$

$\mathcal{L}_2$, on the other hand, has a probability distribution of $\{0,1\}^\lambda$

This makes it quite obvious that there is a difference between the two distributions. Specifically, that $\mathcal{L}_2$ can produce the key consisting of only 0's, while $\mathcal{L}_1$ cannot. While a key consisting of only zeroes isn't ideal, it is a difference in behavior which Eve can exploit. In an example case:

If calling program $F(string\ m)$ which calls $VIEW(m)$, the adversary Eve can't choose which function, but she can choose the message m. Using function F:

```
F(string m) {
    result = VIEW(m);
    if m == result {
        return true;
    }

    return false;
}
```

When F calls $VIEW(m)$ with $\mathcal{L}_1$:$P(VIEW(m))$ returning false $= 1$
When F calls $VIEW(m)$ with $\mathcal{L}_2$:$P(VIEW(m))$ returning false $= \frac{1}{2^\lambda}$

In the case of $\mathcal{L}_2$, it simply returns a random ciphertext $c \leftarrow \{0,1\}^\lambda$, while $\mathcal{L}_1$ returns an OTP encrypted ciphertext with a random key without the possibility of an all zero key. Because the probabilities differ, there is a case where Eve can find a difference in behavior, specifically that $\mathcal{L}_1$ cannot produce an all zero key.

3

We can show these two functions, $\mathcal{L}_1, \mathcal{L}_2$ are interchangable by slowly changing bits of $\mathcal{L}_1$ to look like $\mathcal{L}_2$. Our end goal is to show that $\mathcal{L}_1$ produces values are in the same range that $\mathcal{L}_2$ does.

First, it is important to note the probability distribution of each function. $\mathcal{L}_1$ takes in a base n integer, and outputs a random value $c \in \mathbb{Z}_n$. Given that integer input, $\mathcal{L}_1$ then generates a random key, adds the key to the integer input, and uses a modulus function to maintain the domain of $\mathbb{Z}_n$. Then the probability of a single output occuring in the function is $\frac{1}{n}$.

$\mathcal{L}_2$ has identical input behavior, but different functionality inside the function. It simply returns a random ciphertext in the domain specified, $\mathbb{Z}_n$. The probability of a single result then, is $\frac{1}{n}$.
We can show these functions are equivilant by slowly changing one to look like the other:

1. First, we can add a line to $\mathcal{L}_2$, setting $c = c \bmod n$. This does not change the behavior of the library, as n is the size of the input domain, $\mathbb{Z}_n$. Our function now looks like this:

```
VIEW(M):
    c <-- Z_n
    c = c % n
    return c
```

2. Change initial c assignment to k to store initial key value in a seperate variable. This does not alter the functionality of the library as we are simply storing the random value in a different variable. Our function now looks like this:

```
VIEW(M):
    k <-- Z_n
    c := k % n
    return c
```

3. Lastly, we can then add x to k without consequence to the library as the modulus function maintains the domain of output. The randomness of output is also not changed as it simply offsets all possibilities by a constant amount. Offsetting a uniform distribution by a constant amount does not change the resulting value. Now our function looks like this:

```
VIEW(M):
    k <-- Z_n
    c := (x + k) % n
    return c
```

Our functions are now identical, then $\mathcal{L}_1 \equiv \mathcal{L}_2$, and therefore interchangable.