Ungraded Homework Solutions

CSC 152 – Cryptography

Please notify me of any errors you find. If you need help, ask.

1) *Let's say you are using a polynomial hash function* $k^{n+1} + x_0k^n + x_1k^{n-1} + \ldots + x_{n-1}k \bmod p$ *to hash the three-byte data* `0x 26 14 04`*, and let's say that* $p = 257$*,* k *is randomly chosen to be* `0x55`*, and that the data is broken into 8-bit chunks before hashing. What is the resulting value?*

Using Python as my calculator, the result appears to be 84.

```
k=0x55
p=257
y = (k**4 + 0x26 * k**3 + 0x14 * k**2 + 0x04 * k**1) % p
print(y)
```

2) *Can you find another data string (of any length) that yields the same output value?*

Perhaps the fastest way to do this is to assume a one-byte data can cause an output of 84 and simply use a for-loop to find it. The following outputs 46.

```
k=0x55
p=257
for i in range(256):
    if ((k**2 + i * k) % p == 84):
        print(i)
```

There are algebraic ways to find an answer, but for such a small problem brute-force is easiest.

3) *We saw that an authentication tag can be generated by combining a universal hash like the one above with a random function: TagGen(x, n) = h(x) op f(n). (The operation used depends on the specifics of h and f.) Because it's readily available, let's say we are using the AES S-box for f, the hash function listed above with k=0x55 for h, and addition mod p=257 for the TagGen operation. What authentication tag is generated for the three-byte data* `0x 26 14 04` *when the nonce used is 0x10?)*

We know that that the hash value is 84. The S-box produces 0xCA for input 0x10. Since 0xCA is 202, the tag is 84 + 202 = 286 mod 257 = 29.