# Cryptography Exam

duncan.pf2

May 2021

## 1 Question 1

### 1.1

Generation of a one-time pad key is $k \leftarrow (0,1)^l$, Encryption is $Enc_k(m,k) = k \oplus m$. Decryption is $Dec_k(c.k) = k \oplus c$. Reusing one-time pads are dangerous as if $M_1$ and $M_2$ use the same key $k$, then xoring $M_1$ and $M_2$ together would produce the xor of messages $m_1$ and $m_2$, leaking information in plaintext. If we know one of the messages, we can then xor that to decrypt the other message. All shown below.

$Enc_k(m_1,k) = k \oplus m_1 = M_1$
$Enc_k(m_2,k) = k \oplus m_2 = M_2$
$M_1 \oplus M_2 = m_1 \oplus m_2$
$m_1 \oplus M_1 \oplus M_2 = m_2$

### 1.2

According to Shannon's theorem, the length of the key must at least be the length of the message. As the key is 128 bits and the blocksize is 256 bits, this is not true. To achieve perfect secrecy, the following needs to hold. $Pr[Enc_k(m_0) = c] = Pr[Enc_k(m_1) = c]$. As there is only one k that $Enc_k(m) = k \oplus m = c$, therefore $Pr[Enc_k(m) = c] = Pr[k \oplus m = c] = Pr[k = m \oplus c] = 2^{-128}$ which is lower than their definition of perfect secrecy.

### 1.3

To decrypt AES128-OFB ciphertexts, we were told that encryption occurred with $c_i = r_i \oplus m_i$, also known as $c_i = AES128_k(r_{i-1}) \oplus m_i$. Therefore similar to output feedback mode, to decrypt $m_i = c_i \oplus AES128_k(r_{i-1})$

### 1.4

$r_0$ is a 128 bit key that is chosen randomly. There are $2^{128}$ possible choices of keys, if $r_i = r_j$ then that means the probability of that occurring is $2^{-128} * 2^{-128}$. As there are only 1024 options then the overall probability of it occurring in the small sample space is $2^9 * 2^{-128} * 2^{-128}$ or $2^{-247}$.

### 1.5

To be IND-CPA secure, there is a negligible function negl that for all $\lambda$ that $Pr[PrivK^{eav}_{A,\pi}(n) = 1] \leq 1/2 + negl(\lambda)$ also written as $Pr[PrivK^{eav}_{A,\pi}(n) = 1] - 1/2 \leq negl(\lambda)$. As the probability of an attacker using two values of r that are the same is computationally negligible $2^{-247}$, it is safe to assume that this value does not impact the IND-CPA security of AES128-OFB as it does not considerably affect the $Pr[PrivK^{eav}_{A,\pi}(n)]$.

## 2 Question 2

### 2.1

Collision resistance is when the $Pr[HashColl_{A,H}(\lambda) = 1 \leq negl(\lambda)$. Preimage resistance is let $s \leftarrow Gen(1^\lambda), x \leftarrow (0,1)^*$ and $h = H^s(x)$ then it is infeasible for any PPT adversary A knowing s and h to find y such that $h = H^s(y)$. Second-preimage resistance is let $s \leftarrow Gen(1^\lambda), x \leftarrow (0,1)^*$ and $h = H^s(x)$ then it is infeasible for any PPT adversary A knowing s and x to find $y \neq x$ such that $h = H^s(y)$
In words, collision resistance means that there is a negligible probability for two messages to hash to the same value. Pre-image resistance is that it is sufficiently hard to discover a message given it's hash and second preimage resistance means that given a message, it is sufficiently hard to find another message that hashes to the same value.

### 2.2

```
q = squareroot of 2^128 # approximate number required for a 50% chance of finding a collision
m = [size q] # list of arbitrary inputs of size q
list = [] # hash list

for x in q
    list add H(m)

for y in list:
  for z in list: # starts one space ahead of y to prevent false collisions
    if y == z:
       collision found # print collision values
```

### 2.3

If $H_1$ is collision resistant then $H_{(1,2)}$ is dependent on $H_2$ in order to determine collision resistance. So if $H_1$ is collision resistant, this means that $Pr[HashColl_{A,H}(\lambda) = 1 \leq negl(\lambda)$, therefore hashes produced by $H_1$ have negligible probability of being the same. As a Hash function takes in arbitrary input, this is true for any input received by $H_1$. As $H_{(1,2)}$ hashes the value of $H_2$ which is arbitrary input and is stated to be uniformly distributed, however we are unaware if $H_2$ is collision resistant or not. If $H_2(m_1) = H_2(m_2)$ and $m_1 \neq m_2$, then the hashes of $H_1$ would be the same. Therefore $H_{(1,2)}$ is dependent on the collision resistance of $H_2$.

## 3 Question 3

### 3.1

The prime factors are 7 and 191

### 3.2

$\mathbb{G} = \mathbb{Z}^*_{191}$
$\phi(191) = 190$

### 3.3

The number of generators of $\mathbb{G}$ is equivalent to the number of integers that are co-prime to the order of $\mathbb{G}$. Therefore $\phi(190) = \phi(2) * \phi(5) * \phi(19) = 1 * 4 * 18 = 72$

### 3.4

The possible group orders of $\mathbb{H}$ are 1,2,5,10,19,38,95,190 as the order of any subgroup must divide $|\mathbb{G}|$. Subgroup of order 1 is trivial as to be a subgroup $\mathbb{H}$ has to contain the identity element. A non-trivial subgroup is the subgroup of order 2, as that is the identity element and it's inverse (1,190). The last trivial subgroup is the subgroup of order 190 which is the group $\mathbb{G}$

### 3.5

$\mathbb{H}$ with $< 7 >$ is a subgroup.

The subgroup of generator $< 7 >$ has an order of 10. It's elements are $7^x \mod 191 = 1, 7, 49, 152, 109, 190, 184, 142, 39, 8$
with orderings of 1,10,5,10,5,2,5,10,5,10 respectively as shown below:
$1^1 \mod 191 = 1$
$7^{10} \mod 191 = 1$
$49^5 \mod 191 = 1$
$152^{10} \mod 191 = 1$
$109^5 \mod 191 = 1$
$190^2 \mod 191 = 1$
$184^5 \mod 191 = 1$
$142^{10} \mod 191 = 1$
$39^5 \mod 191 = 1$
$82^{10} \mod 191 = 1$

## 4 Question 4

### 4.1

$Open(sk_R, pk_S, C, \sigma) = m \leftarrow PKE.Dec(sk_R, c)$ and $x \leftarrow DS.Verify(pk_S, \sigma, m)$

Correctness $x = 1$ if $1 = DS.Verify(pk_S, \sigma, m)$ then this means that the authenticity and the confidentiality of the message is preserved.

### 4.2

IND-CPA security revolves around the following property. To be IND-CPA secure, there is a negligible function negl that for all $\lambda$ that $Pr[PrivK_{A,\pi}(n) = 1] \leq 1/2 + negl(\lambda)$ also written as $Pr[PrivK_{A,\pi}(n) = 1] - 1/2 \leq negl(\lambda)$. However, CB is not IND-CPA secure as the $pk_S$ of the digital signature is public and being used as "encryption" for the message $m$. This does not break the EUF-CMA security as another signature is not found that matches the message. However, the actual message itself and not a hash of the message is being encrypted, meaning that once the attacker receives the token they can use the corresponding public key to decrypt and read the message. This allows them to win the IND-CPA game with a probability of 1. In the first step, the adversary picks two messages such that $m_0 \neq m_1$. Once it receives the challenge cipher text $c$, the adversary simply retrieves the public $pk_S$ key and decrypts the ciphertext revealing the chosen message, winning the game with probability 1.

### 4.3

The new specification for the CB should be as follows.

$Gen(\lambda)$ : Returns $(sk_R, pk_R) \leftarrow PKE.Gen(\lambda)$ and $(sk_S, pk_S) \leftarrow DS.Gen(\lambda)$

$Seal(pk_R, sk_s, m)$ : Returns $(c) \leftarrow PKE.Enc(pk_R, DS.Sign(sk_S, m))$

$Open(sk_R, pk_S, C, \sigma) : m \leftarrow DS.Verify(pk_S, PKE.Dec(sk_R, c), m)$

This is IND-CPA secure as the attacker must first decrypt the message before verifying the authenticity of the message. This does not allow the attacker to simply use the Digital Signature public key to verify the token and retrieve the message. Inside it uses the user's secret key, which the attacker does not have. As the public-key encryption scheme is already IND-CPA secure, the digital signature value inside can be treated as an arbitrary value that does not affect the encryption scheme.