Network Security and Cryptography, Summative Assessment 1
Submission Deadline 14 November 2024, Thursday 9.59 AM UK time

You must type your answers in a word processing system, and create a PDF. We will not accept handwritten and scanned or photographed answers. Please submit your PDF on Canvas.

1. Consider the AES-128 key schedule algorithm, with the key `4247316644F4823070F4744FA232172C`.

   Find the subkeys k_1 and k_2. **[4 points]**

2. The file at

   `https://canvas.bham.ac.uk/files/17789736/download`

   is encrypted using AES-128 in CTR mode, using:

   $$\begin{aligned} \text{key} &= \texttt{2B75B85C6CD84D0AB432D228ADC2060D} \\ \text{nonce (a.k.a. IV)} &= \texttt{85C2B686921E512AFF0DB3C67F6D8D97}. \end{aligned}$$

   Decrypt it and describe the image you get. *Hint. The plaintext file is an image in JPG format. To decrypt, use any suitable software or online tool you like. One option is* `openssl`. **[4 points]**

3. In 60 words or less, explain why an encryption system that satisfies IND-CPA can't be broken by machine learning. **[4 points]**

4. Find a printable string x such that the first 20 bits of SHA-256(x) are zero. *Note: the x you submit must be different from the x that everyone else submits.* **[4 points]**

5. Fred proposes to define a new hash function which takes an input of up to 2^{32} bits, as follows:

   ```
   INPUT: bitstring message m of length < 2**32 bits
   Pad m with a 32-bit encoding of the length of m
   c := emptystring
   while there are still bits to take from m:
      Take the next bit from m and append it to c
      c := c << 1
      Take the next |c| bits from m and xor them into c

   Truncate c to 64 bits, if necessary
   OUTPUT: c
   ```

   Here, `|c|` means the length of c, and `<< 1` means rotate the string left by one bit. If there are insufficiently many bits to take from m, then we pad m with 0s. Prove to Fred that his idea is inadequate, by finding a collision for Fred's proposed function. **[4 points]**

6. The "ciphers.zip" file (linked from the Canvas page) contains some text files with RSA PKCS encrypted ciphertext. The files can be decrypted using the secret key key.pem (linked from the Canvas page) and openssl rsautl.

(a) What is the modulus N of the secret key in key.pem. **[2 point]**

(b) Decrypt the file matching the last 4 digits of your student id and write down the result. **[3 point]**

7. Consider the following RSA based encryption scheme that we will call BadModPKC. H_1 and H_2 are two hash functions mapping onto $\mathbb{Z}_n^*$

Procedure $\mathsf{Keygen}(1^\lambda)$	Procedure $\mathsf{Encrypt}(PK, m)$
01 : Choose two random $\lambda/2$-bit primes p and q	// We assume $m \in \mathbb{Z}_n^*$
02 : $n = p \cdot q$	01 : $r \xleftarrow{\$} \mathbb{Z}_n^*$
03 : $\phi = (p-1)(q-1)$	02 : $c_1 = r^e \mod n$
04 : Select e such that $1 < e < \phi$ and $\gcd(e, \phi) = 1$	03 : $c_2 = m \cdot H_1(r) \mod n$
05 : Compute d such that $1 < d < \phi$ and $ed \equiv 1 \pmod{\phi}$	04 : $c_3 = r \cdot H_2(m) \mod n$
06 : Set $PK = (e, n)$	05 : **return** $c = (c_1, c_2, c_3)$
07 : Set $SK = (d)$	
08 : **return** (PK, SK)	

Show that BadModPKC is not IND-CPA secure. **[5 point]**