NATIONAL UNIVERSITY OF SINGAPORE

CS4236 - CRYPTOGRAPHY THEORY AND PRACTICE
(FEEDBACK)

(Semester 1: AY2021/2022)

Time allowed: 2 hours

INSTRUCTIONS TO STUDENTS

This assessment paper contains **FIVE (5)** sections totalling **FORTY** (**40**) marks, and comprises **TEN (10)** printed pages including this one.

This is an **OPEN BOOK** assessment, and you are to answer **ALL** questions. You may cite any result in the lecture notes or tutorials. Answer **ALL** questions within the space provided in this booklet (write on the backs of pages if you need more room).

Please write your Student Number below. (Do not write your name).

STUDENT NO: ________________________________

This portion is for examiners use only.

Question		**Marks**	**Remark**
General topics, short answers	Q1 (8)		
MAC and HASH	Q2 (10)		
Symmetric encryption	Q3 (7)		
Asymmetric encryption	Q4 (8)		
Signatures and secrets	Q5 (7)		
Total:	Q1-5 (40)		

Q1 (Short Answer Questions) (8 marks)

In the following short questions, each answer is worth either 1 (ONE) or 2 (TWO) marks.

1.1 Calculate the bias for $x \oplus y$ (with x, y independent) when $\varepsilon(x) = 0.2$ and $\varepsilon(y) = 0.3$. Show your working. (2 marks)

> I expected to see you using the piling up lemma, and have the correct answer - for example: $\varepsilon(x \oplus y) = 2^1 \times 0.2 \times 0.3 = 0.12$.

1.2 Explain what limits are usually placed on decryption oracles used in adversarial games for defining properties of encryption schemes. (1 mark)

> Normally they cannot decrypt a ciphertext that has been seen before.

1.3 Many proof-of-work schemes involve finding something with a lot of zeros. Explain in your own words what is meant by this. What do you need to do to prove you have "done the work:"? (1 mark)

> Clear understanding of the process of finding a value r such that $\mathscr{H}(m+r)$ has some number of leading or trailing zeroes.

1.4 Show that $G(x) \overset{\text{def}}{=} x \bmod p$, cannot be a PRG. (2 marks)

> The short answer is that the length of $G(x)$ is not larger than x - $\ell(G(x)) \leq \ell(x)$, so by definition it cannot be a PRG. Other arguments about the randomness, or using distinguishers were weaker in my view - why do the long involved thing, when there is a shorter proof? In addition, the argument that an adversary could check if something is less than p and output 1 is very weak.

1.5 Calculate the entropy in bits/symbol, of a source emitting the 4 symbols E, X, A and M, with the probability of E being 0.5, X being 0.25, and A and M having equal (0.125) probabilities. Show your working. (2 marks)

> Working clear, correct answer, and units: 1.75 bits/symbol.

Q2 (MAC and HASH) (10 marks)

2.1 Assume you were investigating a new MAC scheme based on $\text{Enc}_k(H(m))$, where the HASH is SHA3, and the encryption is AES in CBC mode. Show how you could forge a fresh (m',t) pair for an m', but without finding a (HASH) collision. You have an example of a valid (m,t) pair, and control over an AES in CBC encryption mechanism $\text{Enc}_{k,\text{IV}}(p)$. Show clearly each step in your attack. (4 marks)

Since the attacker has an encryption oracle that allows control over the IV, the attacker can control the input to the first stage. This is like the concatenation attack. The attacker can use the IV to pre-set the input to the first stage in CBC.

2.2 Rainbow tables and the birthday attack are both attacks applied to hashes. However, each attempts a different task. Explain clearly what each attack attempts to do. (2 marks)

Rainbow tables reduce the brute force problem of finding something that hashes to a specific hash, trading space and time. Birthday attacks are just for finding a collision - any collision. Find a pre-image vs find-a-pair.

2.3 Show/prove that there exists a MAC that is secure (existentially unforgeable) but that is not strongly secure. (4 marks)

The secure MAC game pays attention to the previous message(s) m, whereas the strongly secure game is recording the previous (m,t) pairs. There were several ideas possible here. A MAC that is secure, but probabilistic, will not be strongly secure. It is also possible to imagine, or construct a MAC with extra bits that are ignored by the verifier. This may be secure, but since you can change these bits at will, it will not be strongly secure.

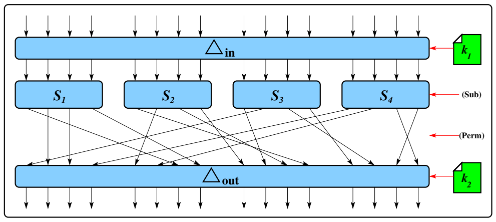
Q3 (Symmetric encryption) (7 marks)

3.1 In an authenticated encryption scheme, why is the **encrypt-and-authenticate** scheme considered to be unsafe (or to have issues)? (2 marks)

Encrypt-and-authenticate does not provide integrity of the ciphertext (it only provides it for the plaintext). As a result, for example, chosen ciphertext attacks may still be possible. In addition, there may be leakage of plaintext information through the MAC - for example, even if the encryption system used an IV, we can see if the same message is being encrypted.

3.2 Briefly explain why in symmetric systems based on rounds, the rounds include both substitutions and permutations. Why could we not have a system based just on rounds of substitution or permutation alone? (2 marks)

The problem is that two substitutions is just one different substitution, and similarly two permutations is just one different permutation.



3.3 Shown above is an example of a single stage of differential analysis, based on the substitution and permutation from the toy example. If the input bits of interest for S_1 were `1001` (i.e. 9), which (16-bit) input and output bits would be most useful for differential analysis, and what would the differential probability $\Pr[\langle\Delta_{\text{in}},\Delta_{\text{out}}\rangle]$ of this be? (3 marks)

Looking at the table, the most significant pairing is $\Pr[\langle 1001,0111\rangle] = \frac{1}{4}$, and higher than any other. The relevant 16 bit input and outputs are $\Pr[\langle\Delta_{\text{in}},\Delta_{\text{out}}\rangle] = \Pr[\langle 1001000000000000, 0110000100000000\rangle] = \frac{1}{4}$.

Q4 (Asymmetric encryption) (8 marks)

4.1 Textbook RSA is deterministic, and so if someone sends the same message as before, an adversary can know this. Assuming that a challenger never sent the same message twice though, an adversary might still be able to differentiate between (say) two messages it was expecting:

$$\begin{aligned} m_1 &= \text{“Attack at dawn”} \\ m_2 &= \text{“Attack at noon”} \end{aligned}$$

where each message encodes to a single 2048 bit integer m and the adversary snoops the ciphertext $c = m^e modN$. Explain how an attacker might be able to identify which of the two messages is in the ciphertext, even if the attacker does not know the public or private keys used. (3 marks)

This is another example of weaknesses in RSA. By using the Jacobi of each message, and comparing with the Jacobi of the ciphertext, the attacker may be able to identify which message is sent.

4.2 Explain why you might use $g = X$ rather than $g = Y$ for a generator for DHKE. (2 marks)

For example - In traditional DHKE based on a cyclic group mod p, use a generator $\langle g^2 \rangle$ to prevent Jacobi attacks. Or perhaps use a generator with a long cycle - higher order.

4.3 Alice is going to send a message to Bob using ECC *encryption*. Bob has a private/secret key $K_S = \langle 4, E_{31}(1,1), (0,1) \rangle$, and public key $K_P = \langle (22,21), E_{31}(1,1), (0,1) \rangle$. If Alice encoded her message as the point $(4,21)$, and chooses a random value $k = 2$, what message does she send to Bob? Show your working. (3 marks)

(SAME as homework 4 this year).

Q5 (Signatures and secrets) (7 marks)

5.1 Explain why canonical verification of a signature is not possible. (2 marks)

Canonical verification requires to checker to use the same process, to re-compute the signature, but for signatures this involves the use of private key. The verifier has no access to this key.

5.2 Explain why **hash-and-sign** is better than (say) **sign-and-hash**. (2 marks)

There are several reasons - firstly efficiency - signing is time-costly, and if you have to sign a long message it would take a long time - hashes by contrast are efficiently computed and short. Consider also the case where a signature scheme might be homomorphic, or partially homomorphic - in these cases, it may be possible to forge the signature for $2m$ from the signature for m.

5.3 In Feldman's VSS scheme, the key k is masked as $c = \mathscr{H}(a_0) \oplus k$, rather than just being a_0. Explain why this is done in this scheme (and not in Shamir's for example)? (3 marks)

Because in this scheme, the key must be delivered so it cannot be repudiated - and the participants must be able to check that the dealer is not cheating. Note also that g^{a_0}may reveal something about a_0(think Jacobi again).

=== END OF PAPER ===