UNIVERSITY OF BRISTOL

January 2022

Faculty of Engineering

Examination for the Degrees
of
Bachelor of Science
Master of Engineering
Master of Science

COMS30021(J)
Cryptology

TIME ALLOWED:
3 Hours

This paper contains 6 questions over 16 pages.
Answer all the questions.
The maximum for this paper is 100 marks.

Other Instructions

1. **This is an open book exam.**
2. **Automated and programmable computing devices are permitted.**
3. **After completion of this exam, you will have 30 minutes to upload your submission to Blackboard.**

Preamble

This exam is composed of 6 questions, *all* of which you must answer:

- 2 regarding *symmetric cryptography*;

Question	Points
Symmetric Cryptography — MAQs	30
Message Authentication Codes	10
Total:	40

- 1 at the interface between symmetric and asymmetric cryptography; and

Question	Points
Security Notions for Digital Signatures	10
Total:	10

- 3 regarding *asymmetric cryptography*.

Question	Points
Asymmetric Cryptography — MAQs	12
Key-exchange and digital signatures	21
Cryptanalysis of the Discrete Logarithm Problem	17
Total:	50

Use of calculators. You are free to use a computer or calculator throughout, but *must* show your working where specified. Wolfram Alpha[1]is sufficient for most of the questions in this exam (at least those that are made easier by having access to a calculator), but feel free to use other tools.

Open Book and Referencing. This is an open book exam conducted online. If you reference external material (material that we did not provide during the course of the unit), you *must* include clear references, in line with the University's academic integrity policy.

[1] https://www.wolframalpha.com/

Marking MAQs. Multiple Answer Questions (in Questions 1 and 4) may have 0 to 4 correct answers. For each proposed answer, mark whether it is True or False, and provide a short justification (max 1 sentence). Marking for each question starts with full marks, and two marks are removed for each incorrect classification (each invalid answer selected, and each valid answer missed), down to a minimum of 0 marks.

Marking Scale. Partial marks will be given for answers that demonstrate general understanding but get details wrong (or forget them). In general (and where possible without fractional marks), getting 50% of the way to a full answer should net you roughly 70% of the marks. Effort beyond that will offer diminishing returns, so plan your work accordingly, and give yourself space and time to iterate on complex questions.

Q1 — Symmetric Cryptography — MAQs [30 marks]

Please recall the rules for marking MAQs stated in the preamble.

[6 marks] **1.a)** Alice and Bob share a 256 bit key known only to them. They have never used it in the past. Alice wants to encrypt and authenticate a single file to send to Bob. The file contains 2.4 Terabytes of recorded lecture material.

A. Alice can securely send the entire file by encrypting it using AES in ECB mode and computing a MAC over the ciphertext and nonce.

B. Alice can securely send the entire file in one message by encrypting it using AES in CTR mode, and computing a MAC over the ciphertext and nonce.

C. Alice can use the shared key as a One-Time Pad to encrypt the data, and as a MAC key to authenticate it.

D. Alice can use an integrated Authenticated Encryption algorithm, as long as it supports long messages.

Solution: ECB is insecure, and can't be used to send anything larger than a single block securely.

A single CTR mode message without a nonce can be 2^{128} blocks long, with each block containing 16 bytes. This (2^{132} bytes) is well over any amount of data we would count in Terabytes. (In fact, even with a 64-bit nonce, we could send the whole precious dataset in one go.)

256 bits is quite a bit less than 2.4 TB, so a One-Time Pad is out of the question.

The use of an integrated Authenticated Encryption algorithm is possible. Some (AES-GCM) do not deal with very long messages particularly securely because authentication tags are too short.

[6 marks] **1.b)** Your friend Bozo designed a cool new blockcipher. It uses 128-bit keys, which are split into two 64-bit subkeys. For a (128-bit) key $k = (k_1, k_2)$ (where k_1 and k_2 are the subkeys) and a message m, the blockcipher computes its output as follows $\mathsf{Enc}_k(m) = \mathsf{Enc}'_{k_1}(\mathsf{Enc}'_{k_2}(m))$, where Enc' is a blockcipher with 64-bit keys.

Assume that the best chosen plaintext attack against Enc' recovers the key in 2^{64} encryptions.

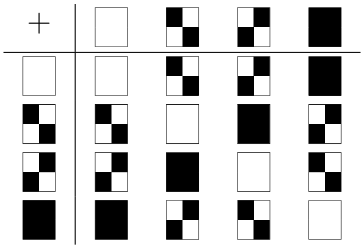
A. A brute-force key recovery attack on Enc takes 2^{128} operations on average.

B. Enc is a lot more secure than Enc'.

C. The best attack on Enc takes 2^{128} operations.

D. The best attack on Enc takes 2^{65} operations.

Solution: Enc is not a lot more secure than Enc$'$ due to meet-in-the-middle attacks, which takes 2^{65} operations instead of the naive 2^{128}.



$+$	□	▚	▞	■
□	□	▚	▞	■
▚	▚	□	■	▞
▞	▞	■	□	▚
■	■	▞	▚	□

Figure 1: Definition for the $+$ operator.

[6 marks] **1.c)** Consider the set $\mathcal{B} = \{$□, ▚, ▞, ■$\}$ and the operator $+ \in \mathcal{B} \times \mathcal{B} \to \mathcal{B}$ defined by the table in Figure 1. Which of the following statements hold?

A. $\Pr\left[b \leftarrow\!\!\$\, \mathcal{B} : b \in \{$▚, ▞$\}\right] = \frac{1}{2}$.

B. $\forall b \in \mathcal{B},\ b + b =$ □.

C. $+$ with a fresh uniformly random key has perfect secrecy.

D. $\Pr\left[b_1 \leftarrow\!\!\$\, \mathcal{B}; b_2 \leftarrow\!\!\$\, \mathcal{B} : b_1 = b_2 =$ ▞$\right] = \frac{1}{4}$

Solution: Sampling uniformly at random in $\mathcal{B}$ yields a half-coloured box in 2 out of 4 cases, so with probability $\frac{1}{2}$. It is clear from its definition that $+$ is nilpotent (the main diagonal is all blank). It is also clear from its definition that $+$ is a group operation ($\mathcal{B}$ is in fact isomorphic to $\mathbb{Z}_4$, although it is easier to see it as the set $\{0, 1\}$ equipped with bitwise XOR—the first bit tells us whether the main diagonal is shaded or not, the second bit gives the state of the second diagonal), so it is invertible and can be used with uniformly sampled keys for perfectly secret enciphering. The probability of sampling twice the same element is $\frac{1}{4}$, but drops to $\frac{1}{8}$ once that element is fixed in advance, as is the case here.

Mary Poppins keeps lots of things in her handbag. She can summon objects out of her bag using the image of their name by some (possibly keyed) function f. The next two choices correspond to this scenario. e

[6 marks] **1.d)** Assume that f is a cryptographically secure pseudorandom function with 256-bit outputs, which Mary uses with a uniformly random key known only to herself. How many items can Mary store in her handbag before the probability that two items have the same tag becomes greater than $\frac{1}{2}$?

A. 2^{128}

B. 2^{256}

C. It depends on the length of the objects' names.

D. 2^{64}

Solution: This is a straightforward birthday bound.

[6 marks] **1.e)** Miss Poppins wants to use the mechanism for self-defence, and wants to hide from would-be muggers that she is summoning her pepper spray. She is particularly worried about repeat offenders hearing her summon her pepper spray once, and later recognizing the string she uses to summon it. Which of the following functions offer the right level of security in this scenario?

A. A collision-resistant hash function.

B. A secure nonce-based encryption scheme.

C. A secure IV-based encryption scheme.

D. A secure deterministic MAC.

Solution: Anything deterministic will allow the would-be mugger to recognize that Mary is summoning the same object as that time she made them cry.

Nonce-based and IV-based encryption schemes both provide indistinguishability from random and will provide security here.

Now you know where "supercalifragilisticexpialidocious" comes from, and you know to run when you hear it.

Q2 — Message Authentication Codes [10 marks]

2.a) We consider the security of MACs. Consider the following experiment, parameterized by a MAC scheme $\mathsf{M} = (\mathsf{Kg}, \mathsf{Tag}, \mathsf{Vfy})$.

```
$\mathsf{Exp}^{\text{seuf-cma}}_{\mathsf{M}}(\mathbb{A})$
    $K \leftarrow\$ \mathsf{Kg}$
    $(\hat{M}, \hat{T}) \leftarrow\$ \mathbb{A}^{\mathcal{T}(\cdot)}$

    $\mathcal{T}(M)$
        $T \leftarrow \mathsf{Tag}_K(M)$
        return $T$
```

The *strong existential unforgeability under chosen message attack* advantage of some adversary $\mathbb{A}$ against some MAC scheme M is defined as follows, where a message-tag pair (M, T) is said to be fresh if the tag T was not produced for message M by the CMA oracle.

$$\mathsf{Adv}^{\text{seuf-cma}}_{\mathsf{M}}(\mathbb{A}) = \Pr\left[\mathsf{Exp}^{\text{seuf-cma}}_{\mathsf{M}}(\mathbb{A}) : \begin{array}{l} \mathsf{Vfy}_K\left(\hat{M}, \hat{T}\right) = \top \\ \wedge \left(\hat{M}, \hat{T}\right) \text{ is fresh} \end{array}\right]$$

[3 marks] i. Justify the adjective strong by proving that any sEUF-CMA secure MAC scheme is also EUF-CMA secure.

[2 marks] ii. Argue that the two notions are equivalent for deterministic MAC schemes.

Solution:

i. The sEUF and EUF experiments are identical, and the advantages differ only in the freshness condition. In particular, it is clear that if the message $\left(\hat{M}\right)$ then for any $\hat{T}$, the message-tag pair $\left(\hat{M}, \hat{T}\right)$ is also fresh. Therefore, any adversary that wins EUF-CMA also wins sEUF-CMA with the same forgery.
1 mark for the reduction logic. 1 mark for the reasoning. 1 mark for quality.

ii. Deterministic MACs have unique tags, so the freshness of a message is equivalent to the freshness of the unique valid message-tag pair for that message.
1 mark for the idea. 1 mark for a well-formed argument.

2.b) Consider the following MAC scheme (simplified to take in a list of blocks, to avoid having to pad), constructed over a pseudorandom permutation E.

```
Tag_K(M[1]||...||M[n])
$X[1] \leftarrow K$
for $i$ in $\{2, \ldots, n+1\}$
    $X[i] \leftarrow E_{X[i-1]}(M[i-1])$
return $X[n+1]$
```

[3 marks] i. Show that the scheme is not EUF-CMA secure by describing a concrete forgery.

[2 marks] ii. Under what condition on the size of messages is the scheme UUF-CMA secure? Informally justify your answer. You can assume that E is a secure PRP.

Solution: The balance of marks is slightly skewed since marks for the second subpart are locked behind the first subpart.

i. The tag for message $M[1] \| M[2]$ is $E_T(M[2])$, where $T = E_K(M[1])$ is the tag for $M[1]$. Therefore, an adversary can mount an existential forgery by requesting a tag (say, T) for some one block message M and extending it to any two-block message $M \| N$ by computing $E_T(N)$.
1 point for the observation that tags are intermediate values used in the computation of tags for longer messages. 1 point for describing the full concrete attack. 1 point for quality.

ii. When single-block messages are not allowed, the target is necessarily multi-block, and the attack described above can be mounted as a UUF attack.
When both single-block and multi-block messages are allowed, we cannot directly conclude with a claim of security or insecurity (although there is probably plenty wrong with the construction; I pulled it out from deep in stupid-land).
When only single-block messages are allowed, the attack does not work. The tag is in fact indistinguishable from random, so the scheme is trivially secure.

1 mark for the condition. 1 mark for the justification. Partial marks could be given for a condition such as fixed-length messages, but there remain issues with the construction that this would not ad-

dress.

Q3 — Security Notions for Digital Signatures [10 marks]

In this question, through some simple problems, we explore some of the security definitions for asymmetric cryptography, and how they relate to symmetric notions.

Syntax and Security of Signature Schemes A signature scheme is a triple of algorithms $\mathsf{S} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vfy})$, where Kg probabilistically produces a *keypair* (split into a public key pk and a secret key sk), Sign probabilistically produces a *signature* given a secret key sk and a message m (we call this a signature on m under sk), and Vfy checks the validity of a signature given a public key, the signature and a message.

Consider the following experiments, parameterised by a signature scheme S and an adversary $\mathbb{A}$. In both cases, the adversary wins if the signature they output is both valid and fresh.

```
$\mathsf{Exp}^1_{\mathsf{S}}(\mathbb{A})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{Kg}$
$(\hat{m}, \hat{\sigma}) \leftarrow\!\$\ \mathbb{A}^{\mathcal{S}(\cdot)}(\mathsf{pk})$

$\mathcal{S}(m)$
$\sigma \leftarrow\!\$\ \mathsf{Sign}_{\mathsf{sk}}(m)$
return $\sigma$
```

```
$\mathsf{Exp}^2_{\mathsf{S}}(\mathbb{A})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{Kg}$
$m^* \leftarrow\!\$\ \mathcal{M}$
$\hat{\sigma} \leftarrow\!\$\ \mathbb{A}^{\mathcal{S}(\cdot)}(\mathsf{pk})$

$\mathcal{S}(m)$
$\sigma \leftarrow\!\$\ \mathsf{Sign}_{\mathsf{sk}}(m)$
return $\sigma$
```

[1 mark] **3.a)** Why is the adversary given the public key as input? Justify this decision based on principles of modern cryptography.

Solution: Kerckhoffs' principle, plus the fact that the key is meant to be public...

[1 mark] **3.b)** Both experiments capture unforgeability notions. Which of Exp^1 and Exp^2 captures existential unforgeability? You may want to refer to similar symmetric notions for comparison.

Solution: Exp^1 is existential unforgeability: the adversary is in control of the message and wins even if the message they manage to forge a signature for is some random gibberish.

[4 marks] **3.c)** Referring to similar notions of unforgeability for symmetric MACs, define advantages for both experiments. In particular, clarify the notion of

validity and freshness.

Solution: The notions here are identical to the symmetric notions, except for the fact that validity must verify with the public key. Existential freshness requires that the message $\hat{m}$ has not been queried to the signing oracle. Universal freshness requires that the message m^* does not get queried to the signing oracle.

[4 marks] **3.d)** Relate the two notions just defined by showing that one implies the other.

Solution: Any EUF-CMA signing scheme is also UUF-CMA. The proof follows the same pattern as that in the symmetric case, but the following details might cause a loss of marks (in addition to not coming up with the right reduction): the reduction must forward the public key to the UUF-CMA adversary, detailed use of public and secret keys.

Q4 — Asymmetric Cryptography — MAQs [12 marks]

Please recall the rules for marking MAQs stated in the preamble.

[6 marks] **4.a)** Which of the following are valid key pairs? You may use a computer for this question, e.g. WolframAlpha will suffice (www.wolframalpha.com).

A. RSA secret key $(56, 154)$ with RSA public key $(13, 154)$.

B. Diffie-Hellman with group generator $g = 2 \pmod{59}$. Secret key 8 with public key 20. You may assume that 2 generates the whole group $\mathbb{Z}/59\mathbb{Z} - \{0\}$.

C. Diffie-Hellman with group generator $g = 2 \pmod{59}$. Secret key 9 with public key 21. You may assume that 2 generates the whole group $\mathbb{Z}/59\mathbb{Z} - \{0\}$.

D. RSA secret key $(5, 91)$ with RSA public key $(73, 91)$.

[6 marks] **4.b)** Which of the following statements are true:

A. For small parameters (e.g., computing discrete logarithms in a cyclic group of about 20 bits), on average, baby-step-giant-step and Pollard-rho will perform similarly.

B. For large parameters (e.g., computing discrete logarithms in a cyclic group of about 256 bits), on average, baby-step-giant-step and Pollard-rho will perform similarly.

C. Index calculus is the most efficient algorithm to solve the discrete logarithm problem in a group of size 2^{100}.

D. Index calculus can be used to solve the discrete logarithm problem in any group.

Q5 — Key-exchange and digital signatures [21 marks]

This question is about asymmetric primitives. Part (a) is about RSA, part (b) about ElGamal encryption, and part (c) is about ElGamal signatures.

[4 marks] **5.a)** You want to send an encrypted message to a friend with RSA public key $(e, n) = (13, 77)$. Using square-and-multiply, encrypt the message $m \equiv 3 \pmod{77}$ with their public key. You do not need a computer or calculator for this question but you may use one if you wish. Show your working.

5.b) Bob has sent you a message encrypted using ElGamal encryption with the integers mod $p = 103$. Your secret key is 59, Bob's public key is 94, your shared secret is 69, and the encrypted message is 86.

[5 marks] i. Using Euclid's algorithm to compute the inverse of the shared secret, decrypt the message. You do not need a computer or calculator for this question but you may use one if you wish. Show your working.

[5 marks] ii. You ask Bob to share the message with Alice, whose public key is $pk_A = 10$. You observe that Bob sends the ElGamal-encrypted message $(pk_B, \mathsf{enc}_m) = (94, 70)$ to Alice. Encrypt the message $m = 4$ and send it to Bob as if you were Alice (using ElGamal encryption). You do not need a computer or calculator for this question but you may use one if you wish. Show your working.

[7 marks] **5.c)** You send messages $m_1 = 100 \pmod{226}$ and $m_2 = 11 \pmod{226}$ to your friend and they sign them using the ElGamal signature algorithm. They return the signed messages $(r_1, sig_1) = (171, 154)$ and $(r_2, sig_2) = (171, 3)$. Compute your friend's secret key. You do not need a computer or calculator for this question but you may use one if you wish. Show your working. You may use the following identities:

$$3 \cdot 151 = 226 \cdot 2 + 1$$

and

$$-112/171 \equiv 76 \pmod{226}.$$

Solution:

(a) One mark for attempting the correct computation ($3^{13} \pmod{77}$). One mark for either the binary expansion of 13 or $3^{13} = 3^{2^3} \cdot 3^{2^2} \cdot 3$ (or similar). One mark for "square" step. One mark for correct answer (38).

(bi) One mark for Euclid forwards, one mark for reverse. One mark for $ss^{-1} = 3 \pmod{103}$. One mark for $m = ss^{-1} \cdot \mathsf{enc}_m$. One mark for correct answer: 52.

(bii) One mark for attempting to compute Alice and Bob's shared secret. EITHER One mark for observing that the inverse of $m = 52$ is 2, one mark for computing shared secret as $\mathsf{enc}_m \cdot m^{-1}$ OR two marks for computing the shared secret in another way (correct answer is 37).

One mark for computing $\mathsf{enc}_m = 4 \cdot 37 \equiv 45 \pmod{p}$. One mark for sending $(10, 45)$ to Bob.

(c) One mark for same r implies same nonce. One mark for $k = (m_1 - m_2)/(sig_1 - sig_2)$. One mark for $k = 41$, one mark for computing k mod 226. One mark for computing $151^{-1} \equiv 3 \pmod{226}$ with any correct method (using hint, Euclid, or on a computer). One mark for correction equation for a, one mark for correct answer $a \equiv 76$. Penalise once only for incorrect modulus.

Q6 — Cryptanalysis of the Discrete Logarithm Problem[17 marks]

[5 marks] **6.a)** i. Using Pollard-rho, find an integer a such that $3^a \equiv 14 \pmod{19}$. You do not need a computer or calculator for this question but you may use one if you wish. Show your working.

[4 marks] ii. Consider the following discrete logarithm problem of finding an integer b such that $14^b \equiv 13 \pmod{19}$. By applying index calculus on the factor base $\{2, 3\}$, we find the three equations

$$14^2 \equiv 2 \cdot 3 \pmod{19},$$

$$14^9 \equiv 2 \cdot 3^2 \pmod{19},$$
$$14^4 \cdot 13 \equiv 2^2 \cdot 3 \pmod{19}.$$

Use these to compute b. Show your working.

[2 marks] iii. Using your answers to parts (i) and (ii), calculate an integer c such that $3^c \equiv 13 \pmod{19}$.

[1 mark] iv. Does there exist an integer d such that $9^d \equiv 14 \pmod{19}$? Justify your answer.

[5 marks] **6.b)** Consider the finite field

$$\mathbb{F}_{3^2} = \{a + bx : a, b \in \mathbb{Z}/3\mathbb{Z}, x^2 + 2x + 2 \equiv 0 \pmod 3\}.$$

The multiplicative group $\mathbb{F}_{3^2}^*$ has order 8 and is generated by $g = x$. Let $h = 2x + 2$. Using Pohlig-Hellman, compute an integer n such that $g^n = h$. You may use the following identities:

$$g^2 = x + 1, \qquad h^2 = 2.$$

Solution: TODO