UNIVERSITY OF BRISTOL

January 2023

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 5 questions over 11 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.

TURN OVER ONLY WHEN TOLD TO START WRITING

Preamble

This exam is composed of 5 questions, *all* of which you must answer. Questions are roughly ordered by increasing difficulty (as *we* perceive it). Within each question, question parts are roughly ordered by increasing difficulty (as *we* perceive it). Indicative difficulty is based on the complexity of the material, the depth of understanding required, *and* the level of guidance given. Some questions we mark as difficult may seem easy if you have engaged throughout. Conversely, some questions we mark as easy may seem difficult if you have not taken opportunities to practice the skills we teach or receive formative feedback.

Use of calculators and computers. You are free to use a computer or calculator throughout, but *must* show your working where specified. Wolfram Alpha[1] is sufficient for simple calculations. For more advanced questions, we expect you to use Sage or Python. In a bind, you can use Sage online via CoCalc.[2]

Open Book and Referencing. This is an open book exam conducted with access to an internet-connected computer. 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.

Marking MAQs. Multiple Answer Questions (Question 1) have between 0 and 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 time and space to iterate on more complex questions.

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

[2] https://cocalc.com/

Q1 — Multiple Answer Questions [30 marks]

[6 marks] **1.a)** (*) Alice and Bob roll one 6-sided die each.

A. Alice and Bob have probability $\frac{1}{6}$ of rolling the same value.

B. Alice and Bob have probability $\frac{1}{6}$ of both rolling a 1.

C. Alice and Bob have probability $\frac{1}{6}$ of rolling different values.

D. Alice has probability $\frac{1}{6}$ of rolling 1.

[6 marks] **1.b)** (*) Which of the following are valid RSA public keys (e, n)? You should use a computer for this question.

A. $(62, 5767)$

B. $(103, 1232)$

C. $(101, 2067)$

D. $(1073, 3233)$

[6 marks] **1.c)** (**) Alice and Bob share a 256-bit value k, known only to them, and generated uniformly at random. let E be a correct enciphering scheme that has perfect secrecy.

A. Without any further interaction, an adversary has probability 2^{-256} of recovering the value k.

B. An adversary who sees some message m and its enciphering $c = E_k(m)$ has probability 2^{-256} of recovering the value k.

C. An adversary who sees some message m and its enciphering $c = E_k(m)$ has probability 1 of recovering the value k.

D. Without any further interaction, an adversary has probability 1 of recovering the value k.

[6 marks] **1.d)** (**) Consider a cryptographic hash function with 256-bit digests. Which of the following hold?

A. The hash function is expected to have collision resistance, preimage resistance and second preimage resistance.

B. There exists a generic attack that computes the hash function q times and finds a collision with probability roughly $\frac{q}{2^{256}}$.

C. There exists a generic attack that computes the hash function q times and finds a collision with probability roughly $\frac{q^2}{2^{256}}$.

D. There exists a generic attack that computes the hash function q times and finds a preimage with probability roughly $\frac{q}{2^{256}}$.

[6 marks] **1.e)** (***) For which of the following values of p and $f(x) \in (\mathbb{Z}/p\mathbb{Z})[x]$ is $(\mathbb{Z}/p\mathbb{Z})/(f(x))$ a finite field?

A. $(p, f) = (5, x^2 + 2)$
B. $(p, f) = (3, x^2 + 1)$
C. $(p, f) = (3, x^4 + 2x^2 + 1)$
D. $(p, f) = (2, x^2 + 1)$

```
$\mathsf{CBC.Enc}^{\mathsf{n}}_{\mathsf{k}}(\mathsf{m}[1]\|\ldots\|\mathsf{m}[b])$
$\mathsf{c}[0] \leftarrow \mathsf{n}$
for $i \in [1 \ldots b]$
    $\mathsf{c}[i] \leftarrow \mathsf{E}_{\mathsf{k}}(\mathsf{m}[i] \oplus \mathsf{c}[i-1])$
return $\mathsf{c}[1]\|\ldots\|\mathsf{c}[b]$
```

Figure 1: Encryption for CBC.

```
$\mathsf{C^*\text{-}MAC.Tag}_{(\mathsf{k}_1,\mathsf{k}_2)}(\mathsf{m}[1]\|\ldots\|\mathsf{m}[b])$
$\mathsf{c}[0] \leftarrow \mathsf{n}$
for $i \in [1 \ldots b]$
    $\mathsf{c}[i] \leftarrow \mathsf{E}_{\mathsf{k}_1}(\mathsf{m}[i] \oplus \mathsf{c}[i-1])$
return $\mathsf{E}_{\mathsf{k}_2}(\mathsf{c}[b])$
```

Figure 2: Tagging for C*-MAC.

```
$\mathsf{BadAE.Enc}^{\mathsf{n}}_{(\mathsf{k}_1,\mathsf{k}_2)}(\mathsf{m}[1]\|\ldots\|\mathsf{m}[b])$
$\mathsf{c}[0] \leftarrow \mathsf{n}$
for $i \in [1 \ldots b]$
    $\mathsf{c}[i] \leftarrow \mathsf{E}_{\mathsf{k}_1}(\mathsf{m}[i] \oplus \mathsf{c}[i-1])$
$\mathsf{t} \leftarrow \mathsf{E}_{\mathsf{k}_2}(\mathsf{c}[b])$
return $(\mathsf{c}[1]\|\ldots\|\mathsf{c}[b], \mathsf{t})$
```

Figure 3: Encryption for BadAE.

Q2 — Authenticated Encryption [20 marks]

This question explores the security of an integrated construction for authenticated encryption. We will consider only messages (including those generated by the adversary) whose length is a multiple of the block length.

Consider the construction BadAE shown in Figure 3, where E is a blockcipher with block length ℓ. Figure 1 recalls the encryption algorithm for the CBC mode of operation over the blockcipher E. Figure 2 recalls the tagging algorithm for the C*-MAC construction over the blockcipher E.

[2 marks] **2.a)** (*) Assume the key space for the blockcipher E is $\mathcal{K}$ (and the message space is $\{0,1\}^\ell$, as standard). What are the key space, message space, and ciphertext space for BadAE as an authenticated encryption scheme.

Solution: The key space is $\mathcal{K}$, the message space is $\mathcal{M} = \left(\{0,1\}^\ell\right)^*$, and the ciphertext space is $\mathcal{M} \times \{0,1\}^\ell$.

[4 marks] **2.b)** (*) Write the decryption algorithm for BadAE. Remember: check the tag.

Solution:

```
$\mathsf{BadAE.Dec}^{\mathsf{n}}_{(\mathsf{k}_1,\mathsf{k}_2)}(\mathsf{c}[1] \| \ldots \| \mathsf{c}[b], \mathsf{t})$
$\mathsf{t}' \leftarrow \mathsf{E}_{\mathsf{k}_2}(\mathsf{c}[b])$
if $\mathsf{t} = \mathsf{t}'$
    $\mathsf{c}[0] \leftarrow \mathsf{n}$
    for $i \in [1 \ldots b]$
        $\mathsf{m}[i] \leftarrow \mathsf{c}[i-1] \oplus \mathsf{D}_{\mathsf{k}_1}(\mathsf{c}[i])$
    return $\mathsf{m}[1] \| \ldots \| \mathsf{m}[b]$
return $\perp$
```

[4 marks] **2.c)** (**) Express BadAE as a composition of CBC and C*-MAC. Which generic composition is it closest to? How does it differ from it?

Solution:

```
$\mathsf{BadAE.Enc}^{\mathsf{n}}_{(\mathsf{k}_1,\mathsf{k}_2)}(\mathsf{m})$
$\mathsf{c} \leftarrow \mathsf{CBC.Enc}^{\mathsf{n}}_{\mathsf{k}_1}(\mathsf{m})$
$\mathsf{t} \leftarrow \mathsf{C}^*\text{-}\mathsf{MAC.Tag}^{\mathsf{n}}_{(\mathsf{k}_1,\mathsf{k}_2)}(\mathsf{m})$
return $(\mathsf{c}, \mathsf{t})$
```

This is most like an Encrypt-and-MAC composition, but the keys used in encryption and authentication are not independent, and the nonce is not authenticated.

[5 marks] **2.d)** (**) Describe a nonce-respecting adversary that produces a ciphertext that decrypts successfully to a message that was not queried to the encryption oracle. Your adversary may make a chosen-plaintext query before producing her forgery.

Solution: An adversary in possession of a ciphertext-tag pair for a known plaintext can easily craft a new ciphertext that will successfully decrypt to a different message, simply by modifying any ciphertext block except the last.

[5 marks] **2.e)** (***) Describe a nonce-respecting adversary against the IND-security of BadAE as a nonce-based encryption scheme. Your adversary may make two CPA queries.

Solution: Given messages m and m' and nonces n and n' such that $\mathsf{m}[1] \oplus \mathsf{n} = \mathsf{m}'[1] \oplus \mathsf{n}'$, the BadAE encryptions of m with n and m' with n' will share their first block. This only occurs with low probability in the ideal world.

Q3 — Key Exchange and Digital Signatures [20 marks]

This question is about ElGamal encryption and ElGamal signatures. You should use a computer for this question.

3.a) (*) You want to send an encrypted message to Alice using ElGamal encryption. Alice sends you her public key $(p, g, pk_A) = (31, 3, 15)$.

[1 mark] i. Using Sage, Python, or a program of your choice, generate a random secret key sk_B and a random message $m \in \{1, \dots, p-1\}$ for yourself. (If you are unable to generate these randomly, just choose some $sk_B, m > 7$ to use for the rest of this question).

[3 marks] ii. Using square-and-multiply, compute your shared secret with Alice. You may do this on pen-and-paper or write a computer program. In either case, show your work (copy down code where relevant).

[3 marks] iii. Using double-and-add, compute your encrypted message. You may do this on pen-and-paper or write a computer program. In either case, show your work (copy down code where relevant).

[7 marks] **3.b)** (**) You ask Alice, who has public key $(p, g, pk_A) = (31, 3, 15)$, to sign two messages $m_1 = 12$ and $m_2 = 23$ using the ElGamal signature scheme. She sends you two signatures

$$(r_1, sig_1) = (24, 6) \text{ and } (r_2, sig_2) = (24, 23).$$

Find Alice's secret key. Show your working. You do not need a computer for this part but you may use one if you wish.
Hint: the possible multiples of $12 \pmod{30}$ are

$$\{12, -6, 6, -12, 0\}.$$

3.c) (***)

[2 marks] i. Let Alice's public key be as above in parts (a) and (b). Suppose that she sends you another signed message. If you wanted to recover, by brute-force, the nonce that she used, what is the maximum number of nonces that you have to check?

[2 marks] ii. How would you construct a (large) prime p to use in the setup for an ElGamal signature such that there are as little as possible valid choices for a valid nonce?

[2 marks] iii. Suppose p is specially constructed as in part (ii) and that g generates $\mathbb{F}_p^*$ as a multiplicative group. Is brute-force the best algorithm to find a nonce k given $r = g^k \pmod{p}$? Justify your answer.

Solution:

(ii)/(iii) 1 mark for binary decomp. 1 mark for splitting up into square and multiply (resp double and add). 1 mark for correct answer.

- One mark for observing nonce reuse. One mark for mod 30. One mark for attempting to solve correct simultaneous equations. One mark for noticing that $12 \equiv 12a \pmod{30} \not\Rightarrow a = 1$. One mark for correct answer. Two marks for correct reasoning from $12 \equiv 12a$ to $a = 21$. (e.g. $a \equiv 1 \pmod 5$ using hint and then checking the options mod 5).

-
    - Hope is that part(b) made tham think about (non)existence of inverses even if they couldn't solve it. One mark for all the invertible ones mod $p-1$, one mark for correct computation of $\varphi(p-1)$ (via any method).
    - one mark for smooth $p-1$, one mark for justification
    - one mark for no, one mark Pohlig-Hellman + justification

Q4 — Cryptanalysis of the DLP [20 marks]

This question is about algorithms to solve the Discrete Logarithm Problem. You should use a computer for this question.

[4 marks] **4.a)** (*) Using index calculus with factor base $\{2, 3, 5\}$, find a such that $492^a \equiv 507 \pmod{569}$. You may use without proof that 492 is a generator of $\mathbb{F}_{569}^*$. Show your work (copy down code where relevant).

[8 marks] **4.b)** (***) Re-solve part (a) using Pohlig-Hellman, additionally making use of a square-root complexity algorithm to find discrete logarithms in the subgroup of $\mathbb{F}_{569}^*$ of order 71. Show your work (copy down code where relevant).

4.c) (***)

[2 marks] i. Comment on the concrete complexity of the algorithm you used for (a) versus the algorithm you used for (b).

[2 marks] ii. For 5 random choices of generator g for $\mathbb{F}_{569}^*$, by looking at factorizations of $g^i \pmod{569}$ for small i, comment on the ease of finding a suitable factor base.

If you are not sure how to compute generators, you can sacrifice one mark and instead look at 5 random numbers in $\mathbb{F}_{569}^*$.

Show your work (copy down code where relevant).

[2 marks] iii. How would you expect Pohlig-Hellman (including a square-root complexity subroutine) to perform with respect to index calculus in the computation of logarithms base g, where the g are those you found in part (ii)?

[2 marks] iv. Suppose that G is a cyclic group that is not the unit group of a finite field, and that the discrete logarithm problem is hard in G. Suppose further that G has 105663913 elements. How would you go about computing discrete logarithms in G? Justify your answer.

Solution:

- One mark for correct modulus throughout (and only penalize once for incorrect moduli). One mark for searching through powers of g^i to find $\log 2$, $\log 3$, $\log 5$. One mark for solving simultaneous equations and finding correct answers. One mark for applying to find $\log 507$.
- One mark for naming a correct square-root complexity algorithm (BSGS or Pollard-rho). 3 marks for correct application of Pohlig-Hellman. 3 marks for correct application of BSGS/Pollard-rho. 1 mark for correctly combining the algorithms.

```
$\mathsf{Exp}^1(\mathbb{A})$
$\mathsf{a} \leftarrow\!\!\$ \; (\mathbb{Z}/p\mathbb{Z})^\times$
$\mathsf{b} \leftarrow\!\!\$ \; (\mathbb{Z}/p\mathbb{Z})^\times$
$\mathsf{r} \leftarrow\!\!\$ \; \mathbb{A}(g^{\mathsf{a}}, g^{\mathsf{b}})$
```

$$\mathsf{Adv}^1(\mathbb{A}) = \Pr\left[\mathsf{Exp}^1(\mathbb{A}) : \mathsf{r} = g^{\mathsf{a}\cdot\mathsf{b}}\right]$$

Figure 4: Experiment 1

```
$\mathsf{Exp}^2(\mathbb{A})$
$\mathsf{a} \leftarrow\!\!\$ \; (\mathbb{Z}/p\mathbb{Z})^\times$
$\mathsf{r} \leftarrow\!\!\$ \; \mathbb{A}(g^{\mathsf{a}})$
```

$$\mathsf{Adv}^2(\mathbb{A}) = \Pr\left[\mathsf{Exp}^2(\mathbb{A}) : \mathsf{r} = \mathsf{a}\right]$$

Figure 5: Experiment 2

```
$\mathsf{Exp}^{\text{3-real}}(\mathbb{A})$
$\mathsf{a} \leftarrow\!\!\$ \; (\mathbb{Z}/p\mathbb{Z})^\times$
$\mathsf{b} \leftarrow\!\!\$ \; (\mathbb{Z}/p\mathbb{Z})^\times$
$\mathsf{c} \leftarrow \mathsf{a}\cdot\mathsf{b}$
$\mathsf{r} \leftarrow\!\!\$ \; \mathbb{A}(g^{\mathsf{a}}, g^{\mathsf{b}}, g^{\mathsf{c}})$
```

```
$\mathsf{Exp}^{\text{3-ideal}}(\mathbb{A})$
$\mathsf{a} \leftarrow\!\!\$ \; (\mathbb{Z}/p\mathbb{Z})^\times$
$\mathsf{b} \leftarrow\!\!\$ \; (\mathbb{Z}/p\mathbb{Z})^\times$
$\mathsf{c} \leftarrow\!\!\$ \; (\mathbb{Z}/p\mathbb{Z})^\times$
$\mathsf{r} \leftarrow\!\!\$ \; \mathbb{A}(g^{\mathsf{a}}, g^{\mathsf{b}}, g^{\mathsf{c}})$
```

$$\mathsf{Adv}^3(\mathbb{A}) = \left|\Pr\left[\mathsf{Exp}^{3-\text{real}}(\mathbb{A}) : \mathsf{r}\right] - \Pr\left[\mathsf{Exp}^{3-\text{ideal}}(\mathbb{A}) : \mathsf{r}\right]\right|$$

Figure 6: Experiment 3

Q5 — Asymmetric Assumptions and Reductions [10 marks]

Let p be a prime, and g be a generator of $\mathbb{Z}/p\mathbb{Z}$. Consider the three experiments and associated advantages displayed in Figures 4, 5, and 6.

[2 marks] **5.a)** (**) Which of these experiments defines:

i. the Discrete Logarithm Problem in $\mathbb{Z}/p\mathbb{Z}$?

ii. an indistinguishability property?

Solution:

i. Experiment 2

ii. Experiment 3

Experiment 2 is the DLP, Experiment 1 is Computational Diffie-Hellman, Experiment 3 is Decisional Diffie-Hellman.

[4 marks] **5.b)** (***) Order the three assumptions by descending order of hardness. For example, recall from Lectures 1 and 4 that Key Recovery is harder than One-Wayness, which is harder than Indistinguishability.

Solution: Experiment 2 is harder than Experiment 1, which is harder than Experiment 3.

[4 marks] **5.c)** (***) Prove either one of the two hardness comparisons from Q5.b. Marks will be given for laying out the reduction logic, for writing down the reduction, and for analysing it.

Solution: The easiest one to prove is that the Discrete Logarithm Problem is harder than the Computational Diffie-Hellman problem. Given an adversary $\mathbb{A}$ that solves the DLP, we must construct an adversary $\mathbb{B}$ that solves CDH with a similar complexity and probability. Given its input (A, B), $\mathbb{B}$ runs $\mathbb{A}(\mathsf{A})$ to obtain a value $\mathsf{a} \in (\mathbb{Z}/p\mathbb{Z})^{\times}$, then returns B^{a}.

If $\mathbb{A}$ does solve the DLP, then $\mathbb{B}$ does output $g^{\mathsf{a} \cdot \mathsf{b}}$, and its advantage is therefore at least that of $\mathbb{A}$. For complexity, $\mathbb{B}$ simply runs $\mathbb{A}$ and computes a single modular exponentiation. This is reasonable.

THIS IS THE END OF THE EXAM