The University of Melbourne

Department of Computing and Information Systems

COMP90043-CRYPTOGRAPHY AND SECURITY

November, 2021

Exam Duration: 15 minutes reading + 120 minutes Exam writing and uploading;

Authorised Materials: The exam is open book; You may only use course materials provided via the LMS or the text book but must not use any other resource including the Internet (See also next page). You can use any notes prepared by yourself. You may use calculators though calculators are not required to answer questions.

Instructions to Students:

- Total marks for the exam is 40 (Worth 40% of the final mark in the subject).
- Please note that the Canvas Exam is 2 hours and 15 minutes, this includes reading and writing time and when you are done you should upload. The Canvas will continue to accept submissions for another 30 minutes (we will accept even if marked late on canvas). You should have submitted by then. If you are later than that, then you will need to upload to OneDrive (link provided on the exam day), where penalties will be applied (1 final subject mark per minute). So to clarify:
  15:00 start combined reading & exam taking
  17:15 end combined reading & exam taking; start scan & upload to Canvas
  17:45 end Canvas upload; start of late submission time with -1 final subject mark per minute late, with only permitted late submission method as upload to OneDrive via provided link (available on the exam day)
  18:15 late submissions closed.
- Link for Late Submission (Note: Only if you fail to submit on Canvas on time):

  https://unimelbcloud-my.sharepoint.com/:f:/g/personal/udaya_unimelb_edu_au/Em7VbqOVYv5MqaNWLS6h3tUBgdrfvJ3YdJo3xcR-MT3ilg
- The exam will have two parts: Part A is a quiz on canvas, Part B is this assignment which has 8 questions.

 continued on next page

- You also must not contact or communicate with any other person (other than teaching team) or make use of the Internet.
- All answers must be written either using a pen on a blank A4 sheet of paper, or by a stylus on a tablet. In either case, the solutions should be hand written on a blank sheet with a new page per question, with the question number clearly marked. If you are using stylus for writing, it is your responsibility to keep track of pages.
- You must **not** type characters to your submission (i.e documents such as word or textfile are not allowed).
- Scanning instructions are already made available on Canvas in an announcement. You submit only one pdf file of scanned written work.
- A discussion forum will be available during the exam for any exam content related issues.
- During the exam, for any non-content-related support, please contact: Inside Australia: 13 6352 / Outside Australia: +61 3 9035 5511 [select option 1, then select option 1 again].

Declaration: By submitting this exam, you certify that you complied with "Declaration of Academic Honesty":

- The answers I am submitting for this assessment are my own unassisted work; and
- I have not made any use of communications devices or channels such as mobile phones, text messages, WeChat or WhatsApp, email, or other messaging technologies, while undertaking this assessment;
- I have not made use of any material outside of what is specified under Authorized Material of this assessment;
- I have not made use of any world-wide web or internet based resources, including google and other search services, Wikipedia, and StackOverflow;
- I have not taken any actions that would encourage, permit, or support other enrolled students to violate the Academic Honesty expectations that apply to this assessment.

 continued on next page

Part A

Please complete the Quiz on Canvas available at *Assignments - Semester 2 Exams, 2021 - Exam: Cryptography and Security. (COMP90043_2021_SM2) PartA*

Part B: This Assignment:

1. [2 marks] For each of the following ciphers, compute the number of possible non-trivial keys. A trivial key is the one which maps all elements to themselves.

   (a) $Cipher_1$: A Vigenère Cipher defined over the alphabet $\mathbf{Z}_{26}$ having a key of length between m_1 and m_2 characters.

   (b) $Cipher_2$: A transposition cipher over the alphabet $\mathbf{Z}_{26}$ with a key length m_3.

   (c) $Cipher_3$: A product cipher defined by product of $Cipher_1$ and $Cipher_2$, where the plaintext symbols first encrypted by $Cipher_1$ and then encrypted by $Cipher_2$.

2. [3 marks] Evaluate or simplify the following expressions. Show the steps in your calculations.

   (a) $a^p + (p-1)^a \bmod p$, where p is a prime number and a is an odd integer co-prime to p.

   (b) Solve x in $3x^{14} + 4x^{10} + 9x - 13 \equiv (0 \bmod 5)$.

3. [2 marks] In the RSA algorithm, if two primes $p = 29$ and $q = 41$ are used, what are the three smallest possible values for e? Please show your working (however, you don't need to show the steps for calculating modular inverses).

continued on next page

4. [3 marks] Consider $\mathbf{GF}(2^3) = \mathbf{GF}(2)[x]$ *mod* $(x^3 + x^2 + 1)$, a field with 8 elements.

   (a) Express all the elements of $\mathbf{GF}(3^2) = \mathbf{GF}(3)[x]$ *mod* $(x^2 + x + 2)$ as polynomials.

i	Elements:x^i	As Polynomials
$-\infty$	0	
0	1	
1	x	
2	x^2	
3	x^3	
4	x^4	
5	x^5	
6	x^6	
7	x^7	
8	x^8	

Table 1: Elements of $GF(3^2)$ as powers of x

   (b) Find the multiplicative inverse of the polynomial $x+1$ in the above field.

5. [6 marks] **Schnorr signatures** The Schnorr signature scheme is as follows:

   - A large prime p is chosen, with a smaller prime $q|(p-1)$. A generator g is chosen of order q.
   - A private key is a random integer $1 < x < q$, and the corresponding public key is $y = g^x \pmod{p}$.
   - To sign a message M:
     - a random value $1 < k < q$ is chosen and $r = g^k \pmod{p}$.
     - The first signature component is $e = H(r||M)$
     - The second signature component is $s = k - ex \pmod{q}$

 continued on next page

Note that there are several different notations for this scheme. Please use the notation provided above.

(a) Provide the verification equation for Schnorr using the notation above.

(b) It is important that an adversary should never use the same value k for two different signatures. Assume that a signer did so by mistake and issued two signatures (M_1, e_1, s_1), (M_2, e_2, s_2). Explain how an observer could detect the reuse of k, even though with Schnorr you would have $e_1 \neq e_2$ and $s_1 \neq s_2$.

(c) Show how, given two signatures (M_1, e_1, s_1), (M_2, e_2, s_2) with the same k value, an adversary can efficiently recover the secret key x.

(d) To avoid the risk of nonce reuse above, Jose has a clever idea to use a counter: he will pick a random z, and then for the i^{th} message he signs he will use $k_i = z + i$. Explain why this is a bad idea: show how an adversary given signatures (M_i, e_i, s_i), (M_j, e_j, s_j) for any i, j could detect this pattern and compute the secret key x (you may assume the adversary knows i and j).

6. [3 marks] This question is about hash function.

(a) Consider the following hash function based on RSA. The key $< n, e >$ is known to the public. A message M is represented by blocks of predefined fixed size $M_1, M_2, M_3, ..., M_m$, $m \geq 1$ such that $M_i < n$ and positive for all $i \leq m$. The hash is constructed as follows: take the square of the first block (mod$n$), XOR (denoted $\oplus$) with the square of the second block (mod$n$), take the result and XOR with the square of the third block (mod$n$), etc and then at the end, the final result is squared (mod$n$) as well. For example, the hash value of a message consisting of m blocks is calculated by

$$H(M) = H(M_1, M_2, ..., M_m) =$$
$$[(M_1)^2 \bmod n \ \oplus \ (M_2)^2 \bmod n \ \oplus \ ... \ \oplus \ M_m^2 \bmod n]^2 \bmod n$$

Does this hash function satisfy each of the following requirements? Justify your answers (with examples if necessary).

i. Variable input size
ii. Fixed output size
iii. Efficiency
iv. Preimage resistant
v. Second preimage resistant
vi. Collision resistant

 continued on next page

(b) What are the consequences of the above hash function being used in a public key signature algorithm?

7. [3 mark] Assume the textbook RSA signature which uses no hash for this question. Marvin (an adversary) accidentally discovers a series of message and signature pairs in Alice's computer.

$$(m_1, s_1), (m_2, s_2), (m_3, s_3), \text{ and } (m_4, s_4),$$

where $s_i = (m_i)^d \ mod \ n, 1 \leq i \leq 4$.

Marvin wishes to create some new messages that are the functions of the above discovered messages as follows:

Index	New message	Function
1	$f_1 =$	$m_1 + \ m_2 + \ m_3 + \ m_4$
2	$f_2 =$	$m_1^7 \ m_2^4$
3	$f_3 =$	$197 \ m_3$

Which of the above messages could he forge and which could he not? Explain both the reasoning and the construction steps involved in the forged signatures.

8. [8 marks] **RSA accumulators** An RSA accumulator is constructed as follows:

- a modulus $N = pq$ is chosen for primes p, q, just like for RSA encryption, except that **nobody** should know the factors p, q.
- A random element $1 \leq g \leq N$ is chosen as the generator.
- To commit to a set $X = \{x_1, x_2, \ldots, x_n\}$, compute the value $A = g^{x_1 \cdot x_2 \cdot \ldots \cdot x_n} \pmod N$.
- To prove the inclusion of a value $x_i \in X$, compute the value $\pi = \mathsf{Proof}(x_i, X) = A = g^{\frac{x_1 \cdot x_2 \cdot \ldots \cdot x_n}{x_i}} \pmod N$
- To verify such a proof π, compute $\mathsf{Verify}(x_i, \pi, A) = \left(\pi^{x_i} \stackrel{?}{=} A \pmod N\right)$

This question will ask about several aspects of this construction.

(a) Assuming a static set X, compare the efficiency of inclusion proofs for Merkle Trees and RSA accumulators. Quantify one way in which Merkle Trees are more efficient and one way in which RSA accumulators are more efficient. Hint: you may consider proof generation time, proof size and/or proof verification time. You may consider opportunities to exploit parallelism.

 continued on next page

(b) Consider the costs of changing the set X and updating the accumulator value A. Asymptotically, how expensive is it to update A if a value is added to the set X? How expensive is it to update A' if a value is removed from the set X?

(c) Show what would go wrong if an adversary was able to obtain the factors p, q. Specifically, show how given any accumulator value A the adversary could forge an inclusion proof for any value x'.

(d) Finally, to show why the scheme is considered secure, show that an adversary who can forge inclusion proofs for arbitrary x' must be able to break RSA encryption. Specifically, show how, given an algorithm $\mathsf{Forge}(A, x', N)$ which computes a valid inclusion proof for x', you can construct an algorithm which breaks RSA encryption by calling Forge as a black box. Explain exactly what values you would pass to Forge.

END OF EXAMINATION