School of Science

COSC2536/2537 Security in Computing and Information Technology

Assignment 1

	Assessment Type: Individual assignment; no group work. Submit online via Canvas→Assignments→Assignment 1. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
	Due date: Week 4, Friday the 14th Aug 2020 11:59pm Deadlines will not be advanced, but they may be extended. Please check Canvas→Syllabus or via Canvas→Assignments→Assignment 1 for the most up to date information. As this is a major assignment in which you demonstrate your understanding, a university standard late penalty of 10% per each working day applies for up to 5 working days late, unless special consideration has been granted.
	Weighting: 15 marks (Contributes 15% of the total Grade)

1. Overview

The objective of Assignment 1 is evaluating your knowledge on the topics covered in Lecture 1-4. Topics include Basic Cryptographic Techniques (symmetric-key cryptography, hash, and cryptanalysis), and Public-Key Cryptography (RSA, ElGamal and Paillier cryptosystems). Assignment 1 will focus on developing your abilities in application of knowledge, critical analysis and decision making. Assignment 1 contains several problems related to the topics mentioned above. You are required to prepare the solutions and upload them as a single PDF or Word document in CANVAS.

In this assignment, there are 4 (four) questions in total. The first question Q1 is on designing a **cryptographic algorithm for a secure vault** with a sophisticated digital keypad. In this question, a scenario is given that describes how a secret key for the digital keypad is generated and the digital keypad works. You need to design an algorithm that satisfies the requirements of the security of the digital keypad.

The second question Q2 is about designing an algorithm to perform ***cryptanalysis*** on a captured encrypted text. The term Cryptanalysis is used to breach cryptographic security systems and gain access to the contents of encrypted messages, even if the cryptographic key is unknown. Therefore, you are expected to apply cryptanalysis in to obtain plaintext from the given ciphertext in Q2.

The third question Q3 is about the designing a *Secure Online Property Auction System* using the ***hash algorithm***. In Q3, you are expected to design an *Online Bidding System* where an attacker cannot determine the bid values of participants and the **hash algorithm** based bidding would work.

The fourth question Q4 is related to **breaking** the ***RSA Encryption algorithm***. **Only for this question, you can submit the solution individually or in a group. In the case of a group submission, the maximum group members can be 3 (three), and you must mention the names of group members in the solution of this question.** In this question, you are expected to design an algorithm that would perform prime factorization using the computational power of 10 computers and determine the private-key ***d*** from the public-key ***(n, e)***. You should demonstrate the detail steps with explanations how the RSA encryption algorithm can be broken. Marks will be deducted if you fail to show the detail computations correctly, skip the computational steps, or do not provide explanations.

Develop this assignment in an iterative fashion (as opposed to completing it in one sitting). You should be able to start preparing your answers immediately after the Lecture-1 (in Week-1). At the end of each week starting from Week-1 to Week-4, you should be able to solve at least one question.

If there are questions, you must ask via the relevant Canvas discussion forums in a general manner.

Submission instructions are detailed in Section 2.

2. Submission Instructions

Overall, you must follow the following special instructions:

- You must use the values provided in the questions.
- Hand-written answers are not allowed and will not be assessed. Compose your answers using any word processing software (e.g. MS Word or Latex).
- You are required to show all of the steps and intermediate results for each question.
- Upload your solution as a single PDF or Word document in CANVAS.

3. Assessment Criteria

This assessment will determine your ability to:

- Follow requirements provided in this document and in the lessons.
- Independently solve a problem by using cryptography and cryptanalysis concepts taught over the first four weeks of the course.
- Meeting deadlines.

4. Learning Outcomes

This assessment is relevant to the following Learning Outcomes:

1. CLO 1: explain the functioning of security services in computing environments and the security issues in networked applications.
2. CLO 2: discuss various types of data integrity and confidentiality mechanisms including public key cryptography.
3. CLO 3: describe basic system security mechanisms and protocols, such as those used in operating systems, file systems and computer networks.

5. Assessment details

Please ensure that you have read **Section 1** to **3** of this document before going further. Assessment details (i.e. question Q1 to Q4) are provided in the **next page**.

Q1. Designing Cryptographic Algorithm for Secure Vault (3 Marks)

One day, three friends (Alice, Bob, and Laura) miraculously found huge number of ancient gold coins of equal size while bushwalking. They decided to equally divide those coins and bring them home. However, given that homes may not be safe to store the coins, they decided to put them in a strong vault in a bank (see **Figure-1.1**).

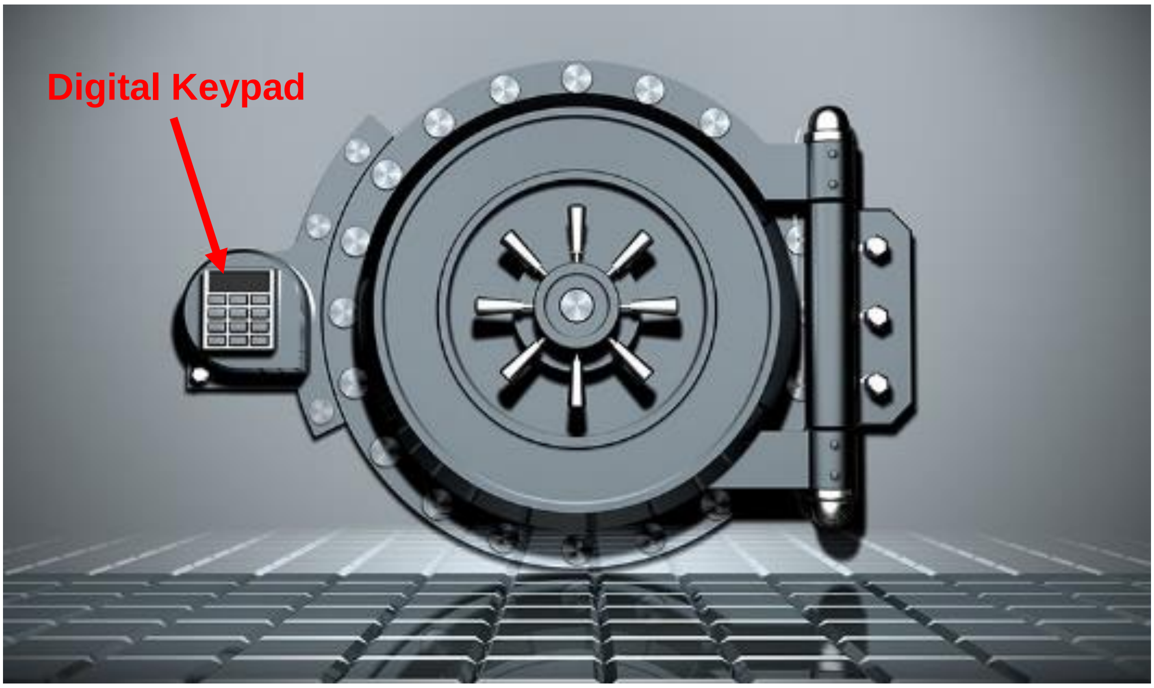


Figure-1.1: A vault with digital keypad

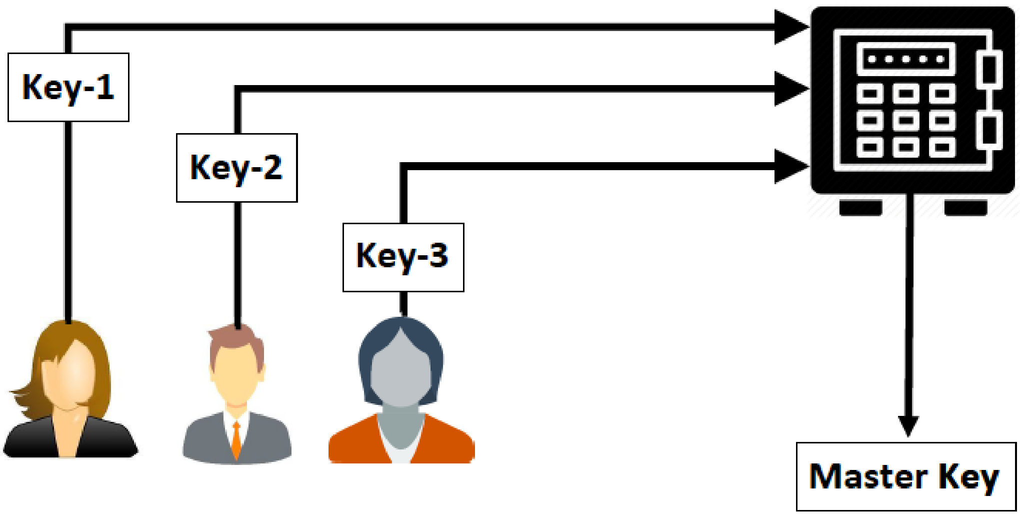


Figure-1.2: Master Key generation at vault from three keys

The vault has a digital keypad (see **Figure-1.1**) which is used to enter secret password for opening it. However, this keypad is very sophisticated and specially designed for the three friends. It can accept three secret keys one after another. Each secret key is an integer number of 5 digits.

When the keypad is initialized each friend enters individual secret key without anyone knowing that number. Once all three friends enter their secret numbers, the sophisticated logic in the keypad performs a mathematical operation and generates a master key by using the three numbers (see **Figure-1.2**). It then stores the master key in the memory and deletes the individual secret keys.

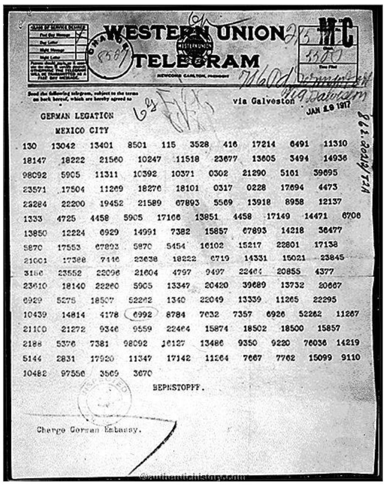
Once the digital keypad is initialized, they can come anytime but they must come all at the same time and enter the secret keys one after another. Similar to the initialization phase, keypad performs a mathematical operation and generates a new master key by using the three numbers. The new master key is then compared with old master key saved in the keypad. If they are same, the vault opens.

Explain the algorithm with an example to design the sophisticated keypad for the excellent vault which has gold coins!

[Note: If you are interested to implement a broader version of this system as a Capstone project, please contact the Lecturer]

Q2. Designing Algorithm for Cryptanalysis with Missing Encrypted Text (3 Marks)

On January 16, 1917, British code breakers intercepted an encrypted message from Zimmermann intended for Heinrich von Eckardt, the German ambassador to Mexico. The challenge was, the encrypted message had many missing ciphertext. The ciphertext and decoded message of Zimmermann is shown in **Figure-2**. In spite of missing encrypted text, the British cryptographic office known as "Room 40" decoded the Zimmermann Telegram and handed it over to the United States in late-February 1917.



WESTERN UNION TELEGRAM

NEWCOMB CARLTON, PRESIDENT

Send the following telegram, subject to the terms on back hereof, which are hereby agreed to

via Galveston JAN 19 1917

GERMAN LEGATION

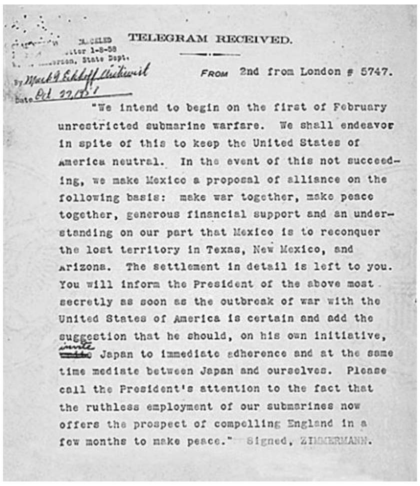
MEXICO CITY

130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
22284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670

BERNSTORFF.

Charge German Embassy.

(a) Encoded Message



TELEGRAM RECEIVED.

FROM 2nd from London # 5747.

"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, invite Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, ZIMMERMANN.

(b) Decoded Message

Figure-2: Zimmermann Telegram

In this task, you have to decrypt an encrypted message. However, here we have encrypted a long English message a bit differently. Every single alphabet in the message has been substituted by another unique alphabet. While the encrypted message was captured, some of the alphabets were missing. A missing encrypted alphabet is marked as '_'. The encrypted message is shown below:

EFA OBE_HA FBK OA_D IBNGDN BHH JBM G_ EFA JGKEBDRA BDJ _ BP SDBOHA EC BKRAIEBGD LFAEFAI BDMEFGDN FBK OAAD NBGDAJ CI DCE. _ FCL G HCDN QCI EFGK LBI EC ADJ. FCL G HC_N QCI TABRA. FCL LG_H G FBGH EFA JBM LFAD G IAESID EC EFA OCKCP CQ PM QBPGHM. P_ JABI G FCTA EC KAA MCS.

You need to perform the followings:

a) Decipher and find out the actual message. Show step-by-step processes.

b) Provide the decryption algorithm (in pseudocode or actual code in any programming language).

Q3. Designing Secure Online Property Auction System using Hash Algorithm (4 Marks)

Covid-19 has changed the way we conduct business these days. This is true for property auctions as well. The Prime minister of Australia recently announced a ban on in-person auctions and open-for-inspections. Large number of sellers and property agents are opting for online auctions. Based on an article published (URL: https://www.domain.com.au/news/saturday-auctions-how-will-they-work-now-they-are-all-online-944545/), we would like to highlight few facts about the current practice in online auctions:

- "Online auctions run like a mix between a live stream and a traditional auction, with buyers registering and placing bids while watching the video as if they were there."
- "Another method involves buyers sending off bids, similar to eBay, and the time allotted for the auction is extended by five minutes every time a bid is entered."

Obviously, there are many issues with online auction, but one of the critical issues is trust – the way online bidding process is conducted. We want to make sure the online bidding process is trustworthy, and nobody can cheat to win.

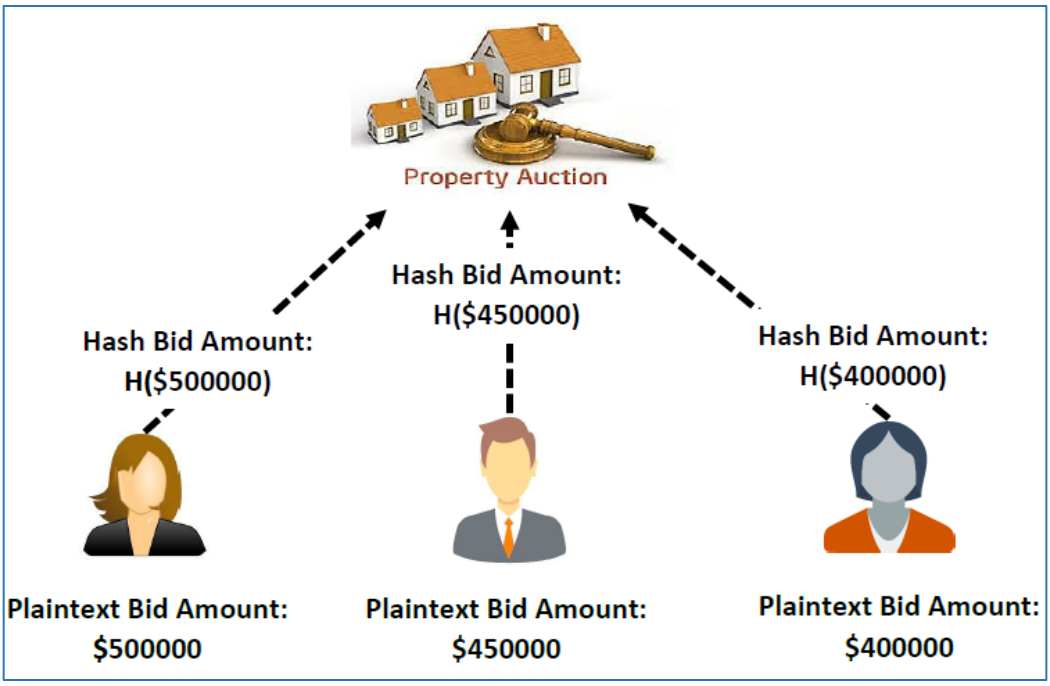


Figure-3: Cryptographic Hash Function based Online Bidding Application

Design a cheating-proof online property auction system using cryptographic hash function with the following requirements:

- A bidder can only bid with the hash value of the bid amount.
- The bidder can bid only once.
- Guessing the plaintext bid amount should be difficult.

Show step-by-step process with concrete examples.

[Note: If you are interested to implement a broader version of this system as a Capstone project, please contact the Lecturer]

Q4. Breaking RSA Key Faster with Multiple Servers (5 Marks)

[Note: Only for this question, you can submit the solution individually or in a group. In the case of a group submission, the maximum group members can be 3 (three), and you must mention the names of group members in the solution of this question.]
It has been found that a quantum computer with 4099 perfectly stable qubits could break the RSA-2048 encryption in 10 seconds, while a classic computer of present days requires 300 trillion years. It means, the powerful computers make the RSA cryptosystem vulnerable.

RSA cryptosystem is mainly built on the concept of prime numbers. The public-key component (***n***) of RSA cryptosystem is an integer that is the product of two prime numbers. Hence, prime factorization is a technique that can be used for breaking RSA private-key (***d***).

Prime factorization or integer factorization of a number is breaking a number down into the set of prime numbers which multiply together to result in the original number. This is also known as prime decomposition. Assume a number ‘77’ has two prime factors. That is, ‘77’ is a product of two prime numbers: 7 and 11 (i.e., 77 = 7 X 11).

The First 10,000 Primes
(the 10,000th is 104,729)
For more information on primes see http://primes.utm.edu/

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541
547	557	563	569	571	577	587	593	599	601
607	613	617	619	631	641	643	647	653	659
661	673	677	683	691	701	709	719	727	733
739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863
877	881	883	887	907	911	919	929	937	941
947	953	967	971	977	983	991	997	1009	1013
1019	1021	1031	1033	1039	1049	1051	1061	1063	1069

……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………

103087	103091	103093	103099	103123	103141	103171	103177	103183	103217
103231	103237	103289	103291	103307	103319	103333	103349	103357	103387
103391	103393	103399	103409	103421	103423	103451	103457	103471	103483
103511	103529	103549	103553	103561	103567	103573	103577	103583	103591
103613	103619	103643	103651	103657	103669	103681	103687	103699	103703
103723	103769	103787	103801	103811	103813	103837	103841	103843	103867
103889	103903	103913	103919	103951	103963	103967	103969	103979	103981
103991	103993	103997	104003	104009	104021	104033	104047	104053	104059
104087	104089	104107	104113	104119	104123	104147	104149	104161	104173
104179	104183	104207	104231	104233	104239	104243	104281	104287	104297
104309	104311	104323	104327	104347	104369	104381	104383	104393	104399
104417	104459	104471	104473	104479	104491	104513	104527	104537	104543
104549	104551	104561	104579	104593	104597	104623	104639	104651	104659
104677	104681	104683	104693	104701	104707	104711	104717	104723	104729

end.

Figure-4: Partial list of first 10000 Prime Numbers

However, a simple method to find the prime factors is to take a list of prime numbers, and start dividing a number by each prime number starting from '2' in the prime number's list. For example, first 10 prime numbers are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. Now, to find out the prime factors of '77', you should divide '77' by each prime number in the above list as follows unless you get another prime number as a quotient:

77 / 2 = Quotient is NOT a prime number
77 / 3 = Quotient is NOT a prime number
77 / 5 = Quotient is NOT a prime number
77 / 7 = 11 (Quotient is a prime number)

Hence, 7 and 11 are two prime factors of 77.

As you know from Lecture-3 and Tutorial-3, the public-key component (n) of the RSA cryptosystems is an integer that has two prime numbers. Assume that you have found the RSA public-key as: **n = 10772542097 and e = 95177**. You want to find the **private-key (d)** for the above RSA public-key.

Say, you have the list of first 10000 prime numbers as partially shown in **Figure-4**. A complete list of first 10000 prime numbers can be found in the URL: https://primes.utm.edu/lists/small/10000.txt. Assume that you have **10 computers**. How can you take advantage of the 10 computers and perform the integer factorization tasks mentioned above to break RSA faster? Explain your algorithm and show detail steps. Please note that we are not interested in any established approach found in textbooks to find prime factors. A simple brute-force method should do the work.

[https://www.quintessencelabs.com/blog/breaking-rsa-encryption-update-state-art/]

6. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge, and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarized, paraphrased, discussed, or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviors, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the University website.

7. Assessment declaration

When you submit work electronically, you agree to the assessment declaration.

8. Rubric/assessment criteria for marking

All of the computations must be correct and only provided values must be used. Instructions must be followed.

Criteria The characteristic or outcome that is being judged.						**Total**
Question 1 Designing Cryptographic **Algorithm**	The answer is correct and the explanation is up to the mark **3 Marks**	The answer is correct, but the explanation is not up to the mark **2 Marks**	The answer is partially correct and the explanation is not up to the mark **1 Marks**	The question is attempted with the correct approach but the answer is not correct. **0.5 Marks**	Not answered. **0 Marks**	**3 Marks**
Question 2 Designing Algorithm for Cryptanalysis	Plaintext is correct Steps are shown in a systematic way and algorithm is presented well. **3 Marks**	Plaintext is correct Steps are shown in a systematic way, but algorithm is not presented well or somewhat incorrect. **2 Marks**	Plaintext is partially correct Or Plaintext is correct. Steps are not shown in a systematic way and algorithm is not presented. **1 Marks**		Not answered **0 Marks**	**3 Marks**
Question 3 Cryptographic Hash Algorithm	The answer is correct, and the explanation is up to the mark **4 Marks**	The answer is correct, but the explanation is not up to the mark **3 Marks**	The answer is partially correct, and the explanation is not up to the mark **2 Marks**	The question is attempted but the answer is not correct. **1 Marks**	Not answered **0 Marks**	**4 Marks**
Question 4 Breaking RSA Encryption algorithm	Step-by-step processes of private-key computation are shown with a distributed algorithm. All of the computations are shown correctly in detail **5 Marks**	Step-by-step processes of private-key computation are shown with a distributed algorithm. Not all of the computations are shown correctly in detail **4 Marks**	Step-by-step processes of private-key computation are shown correctly and distributed algorithm is not convincing or somewhat incorrect. However, private-key computation steps are not shown or incorrectly shown **2 Mark**	Step-by-step processes of private-key computation are shown that are partially correct/ completely wrong. Distributed algorithm is not discussed. **1 Marks**	Not answered **0 Marks**	**5 Marks**