The University of Sydney
School of Computer Science
Dr Suranga Seneviratne
Senior Lecturer - Security

THE UNIVERSITY OF SYDNEY

INFO3616/CSEC3616/CSEC5616 — S2 2025
Assignment - 2

This is an **individual** assignment.

This assignment worths 10% of the final marks of the course and covers the content of Weeks 4-6 (inclusive).

Submit your final report as a `PDF` and codes as a `zip` file in Canvas. In Canvas, under Assignment 2, you will find two links to submit your report and code separately.

You should explain any details of how to run your code in the report.

Please read the below instructions carefully.

***** IMPORTANT *****

1) **Note the answer release date:** Please note the answer release date mentioned below. Any submission after that will get zero marks, instead of a late penalty.

2) **Typeset reports only:** We accept only typeset answers. Any hand written answers will get zero marks. This is because we can't do plagiarism checks for hand written answers.

3) **DO NOT repeat questions:** In your answer sheet DO NOT repeat the questions. Simply include the question number and your answer only. If you include question text in your answer sheet, your TurnItIn score will be high and there will be additional checks. This will cause a delay in releasing your marks. **We will also impose a penalty of 10% of the total marks.**

***** SUBMISSION *****

Final Report and Code: Due by Sunday the 28th of September, 2025 11:59 PM

Simple Extensions: For this assignment, you can apply for simple extensions following the university process. If your request is **approved**, the revised deadline will be **3rd of October, 2025 11:59 PM**. Please note that you have do it yourself and no need to contact the unit of study co-ordinator for this. We will get an automatic email from the system if your request was approved. Also note that, this system is not connected with Canvas, therefore it is normal for you the see the old deadline in Canvas, even if your simple extension was approved.

****Simple Extensions Update 04/09/2025**** - Please refer to the Week 5 announcements on Ed for updated information regarding simple extensions.

Answer release: The answers to this assignment will be automatically released on the **6th of October 00:00 AM**. Any submissions after that will get **zero marks**. If you have a legitimate reason that requires an extension beyond that you will need to go through the university special considerations process. If approved, what you will be grant is a mark adjustment not an extension.

1 Breaking the Vigenère Cipher (25 marks)

In the given `zip` file for Assignment 2 (`Question1.zip`), you will find a cipher text encrypted by the Vigenère cipher. Your task is to break it and find the original plain text. Please read the following instructions carefully.

- You need to write your own code for this. However, there will be some guesswork and analysis involved. So we do not expect your code to give the decrypted result straight away. For example, if you are checking various substitutions, it is acceptable that you run your code multiple times to see which substitution results in a readable text.
- The **spaces** and punctuations such as **.** and **'** are not encrypted. You can store their positions into an array, remove them, do the decryption using your method, and insert them back to the correct positions in the final plaintext.
- You will need ideas from `Kasiski Test` to break this. However there can be other possible solutions. You are not allowed to use any online decryption tool to assist you.
- The key length is less than 20 characters and contain only capital English characters. It is a random string, not a meaningful word. The final plain text is readable and meaningful.
- In the report include a description of your approach, the key length, the key, and the final plaintext you obtained. Also, explain how your code works and how to run it so that the tutors can run your code.

2 AES Calculation (25 marks)

In this question you are going to do some AES calculations **manually**. You are given the following information in HexaDecimal notation.

- Plaintext 0E0D0C0B0A09080706050403020100 0F
- First round key 0303030303030303030303030303030303.

Specifically, you will be **manually** calculating the initial steps in the AES calculation as illustrated in Figure 1.

Answer the following questions. For each step you need to explain what you are doing, your calculation steps, and the answer.

i Show the original contents of State, displayed as a 4 * 4 matrix. **(1 marks)**

ii Show the value of State after initial AddRoundKey. **(3 marks)**

iii Show the value of State after SubBytes. (3 mark) **(3 marks)**

iv Show the value of State after ShiftRows. (3 mark) **(3 marks)**

v Calculate and show the missing values of the first column of the State after MixColumns (see the last matrix in Figure 1). (You can use the remaining value of the column to check whether you did the correct calculations in the previous steps. The rest of the values of the state marked as NA are not applicable to the question.) **(15 marks)**

Note: Addition and multiplication for AES are done on $GF(2^8)$ with the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$.

Referring to following material will be helpful to complete this task

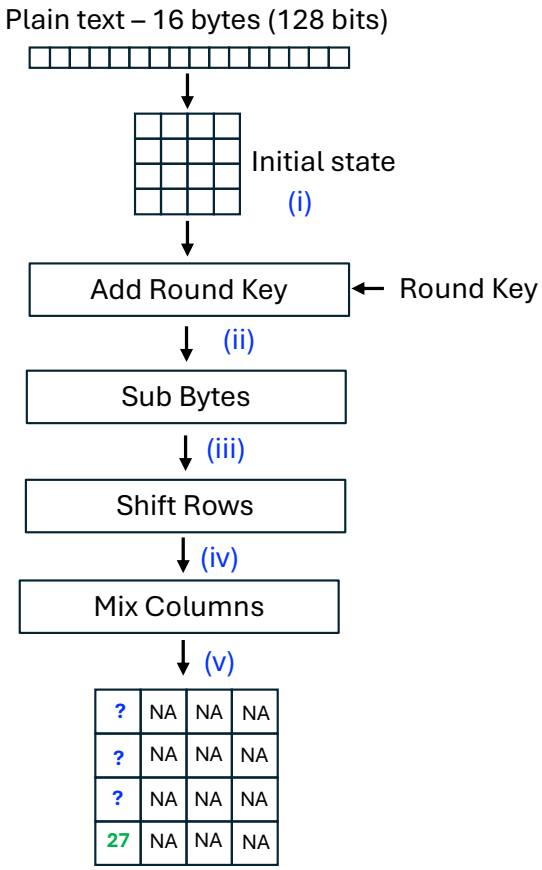

Figure 1: AES manual calculation

- AES Rijndael Cipher explained as a Flash animation
- AES wiki page Advanced Encryption Standard
- Cryptography and Network Security (Seventh Edition - William Stallings) - Chapter 6

3 RSA Calculation (25 marks)

Suppose Bob is generating a public and private key pair using RSA algorithm. Bob choses the prime numbers $p = 4337$ and $q = 6481$. He computed $n = pq = 28108097$ and $\phi(n) = (p-1)(q-1) = 28097280$. He then chooses the encryption exponent $e = 65537$.

i Find the private key. i.e., d where $ed \equiv 1 \mod \phi(n)$ using the Extended Euclidean algorithm **(14 marks - 10 marks for the missing values in the table and 4 marks for d)**.

First follow the explanation below.

Recall the Euclidean Algorithm for finding the GCD of a and b. It goes like this

1) Calculate $r_1 = a \mod b$ which satisfies $a = q_1 b + r_1$

2) Calculate $r_2 = b \mod r_1$ which satisfies $b = q_2 r_1 + r_2$

3) Calculate $r_3 = r_1 \mod r_2$ which satisfies $r_1 = q_3 r_2 + r_3$

...

i) Calculate $r_i = r_{i-2} \mod r_{i-1}$ which satisfies $r_{i-2} = q_i r_{i-1} + r_i$

...

n) Calculate $r_{n+1} = r_{n-1} \mod r_n = 0$ which satisfies $r_{n-1} = q_{n+1}r_n + 0$

When r_{n+1} becomes zero $gcd(a, b) = r_n$

Next we assume that each step we can find x_i and y_i that satisfy $r_i = ax_i + by_i$. We end up with the following sequence.

$a = q_1b + r_1$ and $r_1 = ax_1 + by_1$

$b = q_2r_1 + r_2$ and $r_2 = ax_2 + by_2$

$r_1 = q_3r_2 + r_3$ and $r_3 = ax_3 + by_3$

$r_{n-2} = q_nr_{n-1} + r_n$ and $r_n = ax_n + by_n$

$r_{n-1} = q_{n+1}r_n + 0$

We can rearrange the terms in above equations to obtain

$$r_i = r_{i-2} - r_{i-1}q_i \tag{1}$$

But we also know that $r_{i-2} = ax_{i-2} + by_{i-2}$ and $r_{i-1} = ax_{i-1} + by_{i-1}$

By substituting in Equation (1) we get.

$r_i = (ax_{i-2} + by_{i-2}) - (ax_{i-1} + by_{i-1})q_i$

$r_i = a(x_{i-2} - q_ix_{i-1}) + b(y_{i-2}) - q_iy_{i-1})$

Since we assumed $r_i = ax_i + by_i$ by co-efficient matching we can get

$x_i = x_{i-2} - q_ix_{i-1}$ and $y_i = y_{i-2} - q_iy_{i-1}$

Based on this information your task is to fill the blanks in Table 1 and find the value of d. Note here that initially you will get a negative value for d. You will have to do an extra calculation outside the table to find the positive value of d.

Table 1: Computing the decryption exponent d.

i	q_i	r_i	x_i	y_i
-1	NA	28097280	1	0
0	NA	65537	0	1
1	428	47444	1	-428
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-
5	-	-	-	-
6	-	-	-	-
7	-	-	-	-
8	-	-	-	-
9	-	-	-	-
10	-	-	-	-
11	-	-	-	-
12	5	0	NA	NA

ii Suppose Bob shared his public key e with Alice. If Alice wants to send the message 3 to Bob, find the encrypted message. Show your steps of exponentiation by squaring (You don't have to show all the steps but you can show first 2-3 steps of the exponential squaring and show the final answer - you can use an online calculator the find the final answer e.g. https://www.dcode.fr/modular-exponentiation). (**6 marks**).

iii Verify that Bob can decrypt what Alice sends. Describe the required computation and use an online calculator to obtain the final answer. (**2 marks**).

4 Message Authentication Codes (25 marks)

a) Message Authentication Codes vs. Hashes

i Explain what Authenticated Encryption (AE) is and why it is required. **(2 marks)**

ii Explain the difference between Authenticated Encryption (AE) and Authenticated Encryption with Associated Data (AEAD) **(3 marks)**

b) Protocol Analysis

Consider the following two protocols that are used by Alice to send a message to Bob.

Protocol X: $y = E_{k_1}(x||H(k_2||x))$

Here x is the plaintext message, k_1 and k_2 are shared keys between Alice and Bob, and H is a cryptographically secure hashing function such as SHA256. E stands for a AES encryption. Once y is computed Alice sends y to B. (Here, || denotes concatenation.)

Protocol Y: $x, y = E_{PK}(H(x))$

Here, PK is the public key of Bob and it is assumed Bob has access to the corresponding private key SK. H is again a cryptographic secure hashing function such as SHA256. E stands for RSA encryption. Once y is computed Alice sends x and y to Bob.

i Explain step by step what Bob will do after the reception of the message y in each of the protocols. **(5 marks)**

ii Explain whether confidentiality and integrity is achieved in each of the two protocols. **(5 marks)**

c) Flawed Hash

A number of proposals have been made for hash functions based on using a cipher block chaining technique but without using the secret key. One example, proposed is as follows. Divide a message M into fixed-size blocks $M_1, M_2, ..., M_N$ and use a symmetric encryption system such as DES to compute the hash code H as:

H_0 = Initial Value (Selected arbitrarily)

$H_i = H_{i-1} \oplus E(M_i, H_{i-1})$

$H = H_N$

Assume that DES is used as the encryption algorithm.

Now, DES has an interesting property. We didn't discuss this in the class. But it is easy to understand.

Complementarity property of DES: If $Y = E(K, X)$ then $Y' = E(K', X')$. Here $'$ means the inverse of a binary string. If $Y = 1001$, $Y' = 0110$

Use this property to show how a message consisting of blocks $M_1, M_2, ..., M_N$ can be altered without altering its hash code. **(10 marks)**