IT8x27 Cryptography and Security Mechanisms

Assignment 1:

[Total 100 marks]

Due Date: Monday, 19th August, 11:55 pm

This assignment is worth 30% of total course assessment.

Submission:

Submit your comprehensive solutions, including any relevant CrypTool code snippets or diagrams, using the submission folder on Moodle course page. Ensure your work adheres to academic integrity guidelines.

A Note on Plagiarism

I. Please be aware that dishonest practices will not be tolerated and will be dealt in accordance with WelTec policy.
II. Work that is not original is usually very easy to identify.

The use of generative AI tools or any other automated tools is strictly prohibited for this assignment. All work must be completed independently and manually by the student. Any violation of this policy will be considered academic misconduct and will result in disciplinary action according to the Whitireia/WelTec academic integrity guidelines.

References

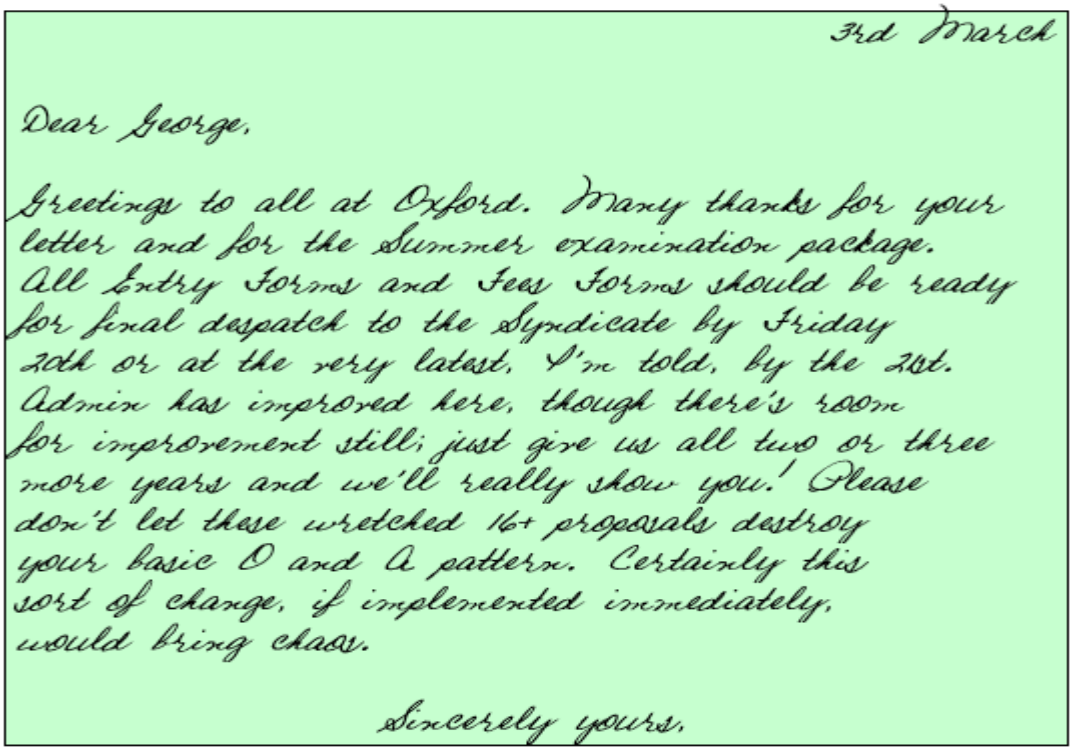
1) Ferguson, N., Schneider, B., & Kohno, T. (2010). Part II Message Security. In *Cryptography Engineering: Design Principles and Practical Applications* (1st ed.). Wiley.
2) William Stallings, *Cryptography and Network Security: principles and practice*, Sixth edition, Pearson.
3) Wenliang Du, Computer & Internet Security, A hands on approach, 2nd edition

You are a member of the cybersecurity team at WelTec SecureTech, a company specializing in secure communications for high-profile clients, including financial institutions and government agencies. Recently, there has been a series of sophisticated cyber-attacks targeting the company's encrypted data streams. Your team is tasked with critically evaluating and enhancing the encryption in use. This assignment will briefly evaluate your comprehension and ability to analyze and enhance encryption methods and cryptographic applications as they are utilized in real-world situations.

Part I [50 marks]

Q1 [5 marks]

A suspicious image file was intercepted during an attack attempt on the company's network. Analyze the image for any hidden messages that may have been embedded using steganography techniques. Show all the workings. [2]



3rd March

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the Summer examination package. All Entry Forms and Fees Forms should be ready for final despatch to the Syndicate by Friday 20th or at the very latest, I'm told, by the 21st. Admin has improved here, though there's room for improvement still; just give us all two or three more years and we'll really show you! Please don't let these wretched 16+ proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours,

Figure 2.9 A Puzzle for Inspector Morse
(from *The Silent World of Nicholas Quinn*, by Colin Dexter)

Q2 [7 marks]

In the Vigenère cipher, the key is a stream of random numbers between 0 and 26.
For example, if the key is 3 19 5 . . . , then the first letter of plaintext is encrypted with a shift of 3 letters, the second with a shift of 19 letters, the third with a shift of 5 letters, and so on.

1. Encrypt the plaintext 'cryptography' with the key stream
   7 0 3 5 21 17 19 12 09 11 4 10 7
2. Using the ciphertext produced in part (a), find a key so that the cipher text decrypts to the plaintext 'cryptanalysis.

Show all the workings.

Q3 [8 marks]

A generalization of the Caesar cipher, known as the *affine Caesar cipher*, has the following form: For each plaintext letter *p*, substitute the ciphertext letter *C*:
$C = E([a, b], p) = (ap + b) \bmod 26$

A basic requirement of any encryption algorithm is that it be one-to-one.
That is, if $p \neq q$, then $E(k, p) \neq E(k, q)$.

Otherwise, decryption is impossible, because more than one plaintext character maps into the same ciphertext character. The affine Caesar cipher is not one-to-one for all values of *a*.
For example, for $a = 2$ and $b = 3$, then $E([a, b], 0) = E([a, b], 13) = 3$.

1. Are there any limitations on the value of *b*? Explain why or why not.
2. Determine which values of *a* are not allowed.
3. Provide a general statement of which values of *a* are b and are not allowed. Justify your statement.

Q4 [15 marks]

Use CrypTool and encrypt your name with 3-DES in the ECB mode under the following two encryption keys:

First key stream (K1) = 11 22 33 44 55 66 77 88 AA BB CC DD EE FF FF FF
Second Key stream (K2) = 11 22 33 44 55 66 77 88 11 22 33 44 55 66 77 88

Decrypt resulting ciphertexts using 1-DES cipher. Provide any intermediate results that you obtain. One among the keys K1 and K2 enables "fast" decryption with the 1-DES cipher (a single application of 1-DES). Which one? You should describe your answer using screenshots and detailed explanations

Q5 [10marks]

Alice encrypts a message using AES, and sends the ciphertext to Bob. Unfortunately, during the transmission, the 2nd bit of the third block in the ciphertext is corrupted. How much of the plaintext can Bob still recover if the mode of encryption is one of the followings: EBC,CBC, CFB, OFB, or CTR? Discuss the impact of each encryption mode on error propagation.

Q6 [5marks]

Describe how a known plaintext attack could be performed on 3-DES. Discuss the feasibility of such an attack and the measures that can be taken to mitigate it.

Part II [50 marks]

Q7 [5marks]

In Linux, the password hash is produced by applying a hash function for many rounds (e.g., 5000 rounds for SHA-512). This seems to waste time, Why does Linux do this? [3]

Q8 [5marks]

A developer writes the following in a post: “I am writing a login for a forum, and I would like to hash the password at the client side in JavaScript before sending it to the server. If the hash matches with the one stored on the server, the user will be allowed to log in.” The developer believes that by sending the hash of the password, instead of sending the password directly, can improve the security. Do you agree or not, why?

Q9 [5marks]

You are safe guarding a gate; anybody passing the gate has to tell you a name and password. Since there is no sound protection, anybody nearby can hear what they say and get the password. To solve that problem, you are requesting that each password can only be used once. Initially, you give each authorized person a list of passwords, and ask them to cross one out each time when a password is used, so each password is used only once. However, you do not want to maintain such a list yourself; you only want to remember one number for each person (up to 32 bytes). You do not mind changing this number each time when a secret password is used. Please describe how you can do this. [3]

Q10 [5marks]

Charlie has arranged a blind date for Alice and Bob, who are both cryptographers, and they do not know each other before. Bob wants to make sure that the person he is dating is actually Alice, not somebody else. Before going to the dating, he got a copy of Alice's public key from Charlie. Please describe how Bob can ask Alice to prove that she is Alice (please do not try this in the real life, or you will probably never see your date again) [3]

Q11[5+5 marks]

In the chip technology used in credit cards, how does the terminal know that a credit card is issued by an authorized bank? Also, how does the card issuer know that the owner of the card has approved a payment? [3]

Q12 [15 marks]

A client from a small bank seeks to implement a system where documents require signatures from two signatories. The bank uses RSA for data protection and digital signatures but needs to adjust its procedures.

1) The first signatory signs the document and passes it to the second signatory.
2) The second signatory verifies the first signature and adds their own. The document must show signatures from both signatories, but only the second signatory should verify the initial signature. The bank prefers to use existing RSA modules.

Explain how this can be accomplished with a generalization of RSA digital signatures.

Q $2^{infinity}$[5 marks]

Please describe a way so we can encrypt a message using the DES algorithm, and then decrypt it using the AES algorithm.