COSC412: Assignment 2

Due: 11:59pm Monday 25/09/2023

Instructions

- All work is to be submitted by email to dme@cs.otago.ac.nz.
- Please submit documents in PDF format—your working can be done by hand and scanned into digital form where needed.
- There are a total of fifteen marks available (this will be scaled to the ten points the assessment is worth in your final grade).

Problems

1. Your task is to design a system that supports the following scenario: a user stores some of their digital assets on a resource server and they want to delegate to a helper service—akin to a third-party 'client' in OAuth 2.0—limited access to the user's assets on the aforementioned resource server. Ensure that you cover the points discussed below in your answer.

    - Specify a set of Kerberos services that can support this interaction, achieving a goal similar to the OAuth 2.0 'authorization code' workflow.
    - Provide a diagram showing how the different services within your design interact.
    - Give a step-by-step breakdown of your design's operation.
    - Indicate advantages and disadvantages of this Kerberos approach compared to using OAuth 2.0 directly.

    [6 points]

2. In a similar manner to the worked examples in the lecture notes on block ciphers, research the output feedback mode (OFB), and demonstrate its operation, following the steps below.

    (a) Create and display two different plaintext messages, each of which comprises two 128-bit blocks.

    (b) Choose and show an initialisation vector (IV) and an encryption key.

    (c) Apply AES-128 as the encryption function within the OFB mode to encrypt each of your messages, but using the same IV and key for both messages. Show the output produced.

    (d) Demonstrate, using the data that you have just encrypted, the problem that was caused by reusing the IV and the key.

    (You may treat the inner workings of AES as a black box.) [4 points]

3. Explain the role of ‘authenticators’ within the Kerberos protocol, and why they are necessary. [1 points]

4. Create and show a plaintext message that is between 130 and 250 bits in length. Demonstrate two different approaches that you can use to pad this message, showing how it gets encrypted by an AES block cipher function, and then decrypted and restored to its original plaintext form. [4 points]