Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Network Security

Exercise Sheet 02: Cryptographic Protocols

Prof. Dr. Mathias Fischer, August See, Finn Sell

Summer semester 2024

Goals and Objectives For programming tasks, submit the code as an extra file. Please write for each (sub) task how much time you needed. You should reach at least 50%.

1 Integrity and replay protection (25%)

Given the service S. Service S should only be used by authorized entities. To use a service, one needs to authenticate against the authentication server (AS). Alice (A) wants to use service S. She authenticates against the authentication server (securely).

1. Given AS returns $AS \to A : ticket = AS[(S, +K_A)]$ and Alice sends $A \to S : (ticket, \{n_A\}_{-K_A})$. Service S, saves and checks used nonces (n_A). How do you assess the security of this approach so far *regarding forgery*? If forgery is possible, how can it be prevented?

2. Given a valid ticket, S returns $S \to A : \{K_{A,S}\}_{+K_A}$, where $K_{A,S}$ is some session key used to secure further communication. How do you assess the security of this approach so far *regarding masquerade*? If masquerade is possible, how can it be prevented?

3. Assume Alices right to access Service S is withdrawn. What are the problems in the current protocol and how can they be fixed?

2 Cipher Malleability (25%)

The purpose of this exercise is to select a malleable cipher and demonstrate how its properties can be exploited. This will deepen your understanding of the security implications associated with cipher malleability. Choose a malleable cipher, e.g., one of the following schemes: AES-CBC, One-Time-Pad, or AES-CTR.

1. Implement Encryption and Decryption Functions
   Use an appropriate library for this!
   - *Encrypt Function:* Create a function that encrypts plaintext using the chosen

scheme.
- *Decrypt Function:* Create a function that decrypts ciphertext using the chosen scheme.

2. Create a function that can alter the ciphertext in a controlled way to demonstrate the malleability of the cipher.

3. Conduct an Experiment
- *Encrypt and Display:* Encrypt a sample plaintext and show the ciphertext.
- *Manipulate Ciphertext:* Use the manipulate function to alter the ciphertext.
- *Decrypt and Explain:* Decrypt the manipulated ciphertext, display the altered plaintext, and explain the results.

3 Padding Oracles (25%)

This exercise aims to explore the concept of a padding oracle attack through detailed explanations and practical pseudocode examples.

1. Explain what padding oracle attacks are and how they can be utilized to compromise cryptographic security.

2. Provide a Pseudocode Example
- *Setup:* Describe the cryptographic setup (e.g., block ciphers with PKCS#7 padding).
- *Pseudocode for Attack:* Write pseudocode showing the steps to exploit padding validation to deduce plaintext.

3. Step-by-Step Walkthrough of the Attack
Offer a step-by-step explanation of the pseudocode, detailing the logic behind each step and its role in decrypting the ciphertext.

4 Cryptographic algorithms (25%)

1. Implement Diffie Helmann by yourself. Use MODP group 14 for the prime and generator (`https://datatracker.ietf.org/doc/html/rfc3526#section-3`) and recommended exponent sizes (`https://datatracker.ietf.org/doc/html/rfc3526#section-8`). Comment your code adequately.

2. Implement Signals Symmetric Key Ratchet (Only the symmetric, not the DH ratchet). Use appropriate cryptographic primitives for this: `https://signal.org/docs/specifications/doubleratchet/#recommended-cryptographic-algorithms`.

   Simulate a conversation between Alice and Bob where each party sends at least two messages. Submit the code and logging output of the individual steps (Initial keys, ratchet turning, encrypting, decrypting), see Listing 1. More hints and recommendations are given in Listing 2 .

3. Explain the role of the DH ratchet in the Signal protocol.

```
Initial constant fffaabcc
Initial chain key abcdef00
Turning Alice send chain ratchet. Chain key f87efa4a..., message key 83e53b5e...
Encrypting b'Hi bob, how is netsec?' with key 83e53b5e... . ciphertext is 8a0771ef...
```

Listing 1: Example log output

```python
from typing import Tuple

def symmetric_ratchet(constant: bytes, chain_key: bytes) -> Tuple[bytes, bytes]:
  """Symmetric key ratchet
  https://signal.org/docs/specifications/doubleratchet/#symmetric-key-ratchet
  https://signal.org/docs/specifications/doubleratchet/#recommended-cryptographic-
  algorithms
  https://cryptography.io/en/latest/hazmat/primitives/key-derivation-functions/#hkdf

  Args:
    constant: Some constant used in the KDF
    chain_key: The chain key that is used in the KDF

  Returns:
    Tuple[bytes, bytes]: new chain_key, message_key
  """
  pass

def encrypt_message(message: bytes, message_key: bytes) -> bytes:
  """Encrypt the given message using the given message key
    https://cryptography.io/en/latest/hazmat/primitives/symmetric-encryption/
  """
  pass

def decrypt_message(message: bytes, message_key: bytes) -> bytes:
  """Decrypt the given message using the given message key
    https://cryptography.io/en/latest/hazmat/primitives/symmetric-encryption/
  """
  pass
```

Listing 2: Example structure with hints