Hochschule Bochum IT-Sicherheit – Praktikum 1

Labor für Informatik und Mathematik im Anwendungsfeld Industrie 4.0

Themenbereich: Kryptographie

Prof. Dr. Christian Scheffer
Stefan Hausotte, M.Sc.
Niklas Schütrumpf, B.Sc.

Dieses Praktikum kann in einer Gruppengröße von bis zu vier Personen durchgeführt werden. Ihre **Gruppengröße ist gleich die Anzahl der Aufgaben**, die Sie gemeinsam zu erledigen haben. Heißt sind Sie zweit, dürfen Sie sich zwei Aufgaben aussuchen. Sind Sie widerrum zu viert, müssen Sie alle Aufgaben gemeinsam lösen.

Die Aufgaben dürfen Sie in einer Programmiersprache Ihrer Wahl erledigen. Bitte einigen Sie sich jedoch für jedes Praktikum für eine Programmiersprache innerhalb der Gruppe. Kommentieren und Dokumentieren Sie Ihren Code, so dass ein Code-Review ohne große Hürde möglich ist.

Für jede Gruppe geben Sie einmalig alle Aufgaben **zusammen** bei Moodle ab.
Geben Sie immer das PDF in ausgefüllter Form ab sowie den Code mit Ergebnissen zu den bearbeiteten Aufgaben.

A1: Frequency Analysis

Schreiben Sie ein Programm, das eine Textdatei einliest und eine Analyse der Buchstabenhäufigkeit durchführt. Berechnen Sie die Häufigkeiten der Buchstaben im Text und stellen Sie die Ergebnisse grafisch dar. Versuchen Sie anschließend, den Text mithilfe von Letter Frequency Analysis zu entschlüsseln und den Klartext auszugeben.
Verwenden Sie hierfür die Datei *Aufgabe1.txt*.
Hinweis: Der Text ist in der Sprache Deutsch.

Entschlüsselter Text:

der geheimnisvolle fremde schritt durch die dunklen gassen der stadt.
der mond hing tief am himmel, sein fahles licht warf gespenstische schatten auf die alten gemäuer.
die stille wurde nur vom leisen klappern der schritte des fremden durchbrochen.
in seiner hand hielt er einen alten, vergilbten briefumschlag.
seine augen huschten nervös von einer dunklen ecke zur nächsten, als er sich dem vereinbarten treffpunkt näherte. ein hauch von angst lag in der luft, während er sich dem unbekannten schicksal entgegen bewegte.

A2: AES-CBC

Implementieren Sie den Advanced Encryption Standard (AES) im Cipher Block Chaining (CBC)-Modus. Schreiben Sie ein Programm, das die Binärdatei *Aufgabe2.bin* einliest, diese in Blöcke aufteilt und jeden Block gemäß dem AES-Algorithmus verschlüsselt. Speichern Sie den verschlüsselten Text in einer neuen Datei. Verwenden Sie als Padding *0*. Nutzen Sie als Schlüssel *ITS-Prakt2024*.

Was ist bei dem Schlüssel zu beachten?

Bei der Auswahl eines Schlüssels für den AES-Algorithmus ist es wichtig, sicherzustellen, dass er eine gültige Länge von 128, 192 oder 256 Bit hat, um eine angemessene Sicherheit zu gewährleisten. Um eine ausgewogene Balance zwischen Sicherheit und Effizienz zu erreichen, ist es ratsam, einen Schlüssel mit der größtmöglichen Länge zu wählen, die die Anwendung erlaubt.

Für eine optimale Sicherheit sollte der Schlüssel zufällig generiert sein. Daher ist es empfehlenswert, für die Generierung von Schlüsseln echte Zufallszahlen zu verwenden.

A3: Diffie-Hellman

Implementieren Sie den Diffie-Hellman-Schlüsselaustauschalgorithmus. Schreiben Sie eine Funktion, die es zwei Benutzern ermöglicht, gemeinsam einen geheimen Schlüssel auszutauschen, ohne diesen über einen unsicheren Kanal zu senden.

Wie stellen Sie sicher, dass Ihre Implementierung sicher gegen Man-in-the-Middle-Angriffe ist?

Um den Diffie-Hellman-Schlüsselaustausch gegen Man-in-the-Middle-Angriffe abzusichern, ist die Nutzung von Authentifizierungsmechanismen wie digitalen Zertifikaten effektiv. Diese Zertifikate, die von vertrauenswürdigen Zertifizierungsstellen ausgestellt werden, bestätigen die Identität der Parteien und sichern so die Echtheit der öffentlichen Schlüssel. Oft wird dies in Verbindung mit SSL/TLS verwendet, um sowohl die Identitäten zu verifizieren als auch die Datenübertragung zu verschlüsseln.

A4: One Time Pad

Gegeben sind zwei Dateien *Aufgabe4_1.txt* und *Aufgabe4_2.txt* welche einen HEX-String enthalten eines Deutschen in One Time Pad (OTP) verschlüsselten Satzes. Implementieren Sie ein Programm, das Ihnen dabei hilft die zwei HEX-Strings wieder vollständig zu entschlüsselt und den ursprünglichen Klartext wiederherzustellen.
Hinweis: Die Klartexte sowie der Schlüssel sind gleich lang. Für beide Klartexte wurde der identische Schlüssel verwendet. Die Texte beinhalten Wörter welche sich auch in diesem Dokument wiederfinden. Alle Wörter im Klartext wurden in CAPS geschrieben.

Welche Art von Angriff fahren Sie?

Wir nutzten einen Wörterbuch-Angriff. Dabei werden die Einträge einer vordefinierten Wort-Liste genutzt, um geheime Inhalte zu erraten (hier: Klartexte der verschlüsselten Nachrichten). Das ist oft effizienter als jede mögliche Kombination von Zeichen zu testen, besonders wenn man die wahrscheinlich im Geheimtext genutzten Worte auf eine übersichtliche Menge eingrenzen kann (hier: Wörter in der PDF).