Grundlagen der Kryptographie

Praktikum 3

Prof. Dr. Christoph Saatjohann

18.11.2025 - 09.12.2025

1 Aufgabe 2: Galois Fields

Generieren Sie eine Multiplikationstabelle für das Galois Field $GF(2^3)$ mit dem irreduziblen Polynom $P = x^3 + x + 1$.

Berechnen Sie dafür die Multiplikationen mit anschließeneder Reduction. Das Ergebnis der Reduktion ist der Rest der Polynomdivision (bzw. das Ergebnis nach Einsetzen der umgestellten Terme von $P(x)$ wie in der Vorlesung gezeigt).

2 Aufgabe 3: AES

Berechnen Sie die AES-128-Verschlüsselung vom Start aus bis in die erste Runde, **VOR** dem Key Addition Layer der Runde. D.h. Sie müssen nicht mehr k_1 aus der Schlüsseltransformation berechnen! Gegegen ist ein Plaintext der nur aus Einsen besteht. Der Schlüssel besteht ebenfalls nur aus Einsen:
$Plaintext = 0x11111111111111111111111111111111$
$Key = 0x11111111111111111111111111111111$

3 Block Cipher Modi - CBC

Szenario:
In der Firma *AllSecure* wird automatisch der komplette Netzwerkverkehr automatisiert mittels AES-256 im CBC Modus verschlüsselt. Die Verschlüsselung läuft Datei-basiert, d.h. für jede übertragende Datei wird der aktuelle IV genutzt. Dieser IV wird einmal morgens für den kompletten Tag neu generiert. Der AES-Schlüssel ist ein Langzeit-Key und wird nicht automatisch neu generiert.

3.1 IV-Rekonstruktion

Sie konnten aus einem alten Backup den Langzeitschlüssel extrahieren. Weiterhin konnten Sie die heutigen übertragenden Dateien mitschneiden, allerdings

fehlt Ihnen noch der tagesaktuelle IV zum Knacken der übertragenden Dateien. Sie wissen aber von einer Dateiübertragung einer temporären 16 Byte Datei, welche ausschließlich aus 0xFF besteht. Beschreiben Sie wie sie mit diesen Informationen die restlichen Dateien knacken können.

3.2 IV-Rekonstruktion: Praxis

Recovern Sie den IV und entschlüsseln Sie eine weitere abgefangene Nachricht. Welchen Inhalt hat diese Datei?
Key:

000102030405060708090a0b0c0d0e0f

Temporäre, verschlüsselte, Datei:

0e6bcb21f0dbf708284b788059bb3b5b

Die weitere, verschlüsselte, Datei die Sie entschlüsseln möchten:

f42672e2e0fce236d844515409df8132
d8b757a75cd8930dd8be61255cd118dd
c30f2ceae04c505f05feff694742db3d

Hinweis: Sie dürfen sowohl online-AES-Tools nutzen oder eine eigene Implementierung. Sowohl im ECB als auch im CBC Modus müssen die Eingangsdaten ein Vielfaches der Blockgröße sein (AES: 16 Bytes). Im unseren Szenario sind alle alle Dateien ein Vielfaches von 16 Bytes. In der Praxis ist dies allerdings selten der Fall! Daher werden hier sogennate Padding-Verfahren (`https://en.wikipedia.org/wiki/Padding_%28cryptography%29#Byte_padding`) genutzt, welche die Daten bis auf die benötige Größe "auffüllen". Abhängig der Paddingverfahren werden hier auch bei Datenblöcken welche ein Vielfaches der Blockgröße sind Paddingbytes angehängt, wodurch der Output größer als der Input wird. Berücksichtigen Sie dieses Verhalten bei der Nutzung von Online-Tools.

4 Toy-Cipher

Gegeben sei eine, unsichere, Permutationscipher: $e(b_1b_2b_3b_4b_5) = (b_2b_5b_4b_1b_3)$

4.1 AES Modi Implementierung

Implementieren Sie diese Cipher mit den 5 verschiedenen Block-Cipher-Modi: ECB, CBC, CFB, OFB and CTR

4.2 AES Modi Ciphertexte

Verschlüsseln Sie die Nachricht x mit diesen 5 Modi und geben Sie den resultierenden Ciphertext an.

IV = 10101

Nachricht x = 01101 11011 11010 00110