Authentifizierte Verschlüsselung

Dozent: Prof. Dr. Michael Eichberg

Kontakt: michael.eichberg@dhbw.de

Version: 0.1.5

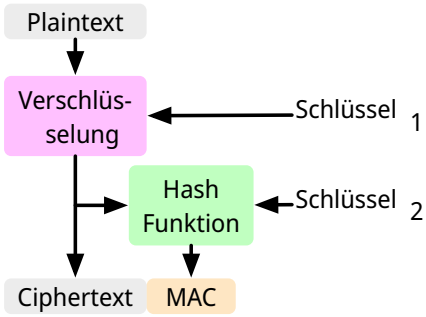
Teilweise basierend auf: *Cryptography and Network Security - Principles and Practice, 8th Edition, William Stallings*

Folien:
- **HTML:** https://delors.github.io/sec-authentifizierte-verschluesselung/folien.de.rst.html
- **PDF:** https://delors.github.io/sec-authentifizierte-verschluesselung/folien.de.rst.html.pdf

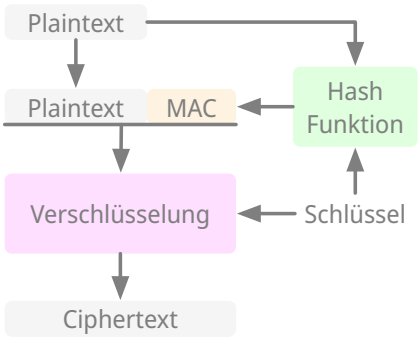
Fehler melden: https://github.com/Delors/delors.github.io/issues

Drei Ansätze in Hinblick auf *Authenticated Encryption*

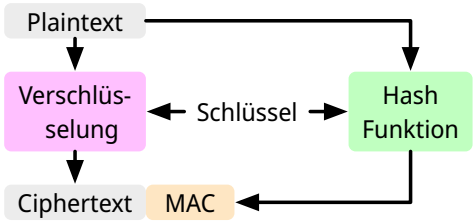
Encrypt-then-MAC



~~MAC-then-Encrypt~~



Encrypt-and-MAC



Modi

- **Encrypt-then-MAC**: Der Klartext wird verschlüsselt und dann wird ein MAC über den Chiffretext berechnet. Dieser Ansatz wird von IPSec und TLS 1.3 verwendet.
- **Encrypt-and-MAC**: Der Klartext wird verschlüsselt und ein MAC über den Klartext berechnet. Beides wird versendet. Dieser Ansatz wird von SSH verwendet. Es wurde gezeigt, dass kleinere Änderungen die Sicherheit weiter verbessern können.
- **MAC-then-Encrypt**: Ein MAC wird über den Klartext berechnet und dann wird der Klartext und der MAC verschlüsselt. Dies war bis TLS 1.2 der Standard. Aufgrund von erfolgreichen Angriffen - insbesondere gegen das Padding bei Verwendung des CBC Modus (siehe Padding-Oracle Attacks) - wird dieser Ansatz nicht mehr verwendet.

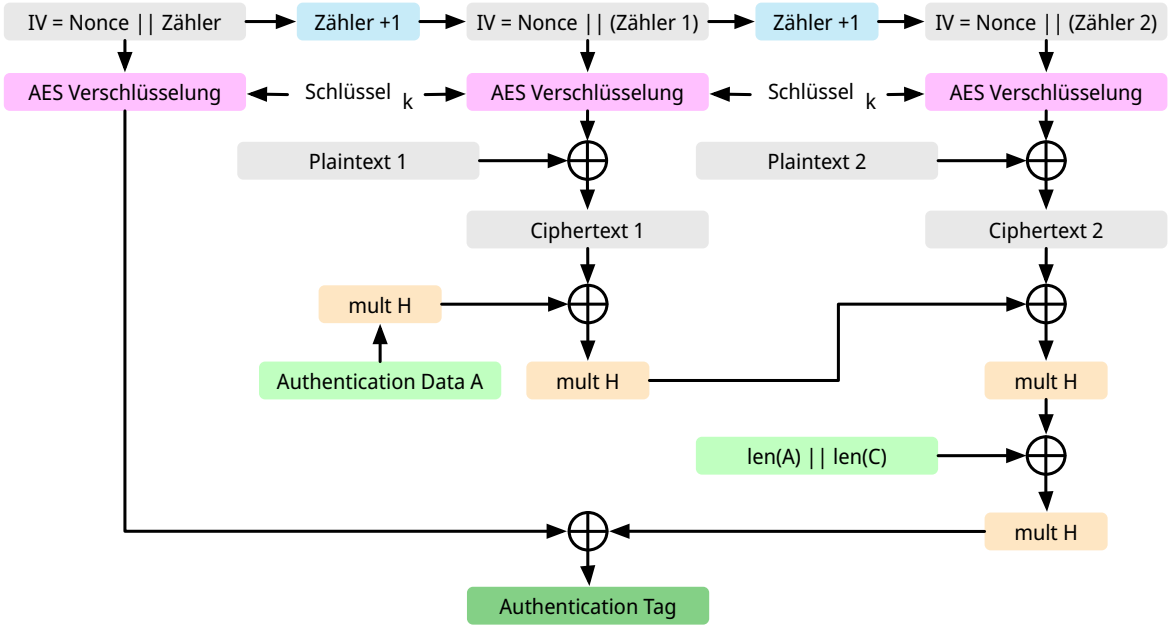
Das grundlegende Problem entsteht dann, wenn der Angreifer bei einer Manipulation der Nachricht (am Ende) nicht nur eine generische Fehlermeldung erhält (Decryption failed) sondern ggf. die Information, dass ein Padding-Fehler vorliegt. Er kann diese Information nutzen, um schrittweise eine Entschlüsselung vorzunehmen. Alternativ können ggf. Timing-Unterschiede im Verhalten verwendet werden.

Integrität und Authentizität

Es ist möglich Integrität ohne Authentizität zu gewährleisten. Durch einen einfachen Hash kann gewährleistet werden, dass die Daten während der Übertragung nicht verändert wurden (insbesondere durch einen Fehler). Wenn ich jedoch Authentizität gewährleisten möchte, dann muss ich einen MAC verwenden, der auf einem Schlüssel basiert, der nur dem Sender und dem Empfänger bekannt ist. Dies verhindert, dass ein Angreifer einfach die Daten verändert und den MAC neu berechnet.

Authentizität ohne Integrität ist nicht sinnvoll. Der Nutzen zu wissen, dass eine Nachricht von einer bestimmten Person kam, aber nicht zu wissen, ob die Nachricht verändert wurde, ist sehr gering.

AES-GCM Modus (Galois/Counter Mode)



Authenticated Encryption with Additional Data (AEAD)

- Standardisiert durch NIST in SP 800-38D.
- Es handelt sich um eine Verknüpfung des CTR-Modus und des Galois-Modus. Ziel ist eine hohe Parallelisierung und Effizienz.
- Der Algorithmus ist in der Lage, Authentizität (+ Integrität) und Vertraulichkeit zu gewährleisten.
- Die Eingabe in den Algorithmus ist der Klartext (*Plaintext*), der Schlüssel, ein Initialisierungsvektor (IV) und zusätzliche (optionale) authentifizierte Daten A.
- Das Authentication Tag wird mittels Arithmetik über dem Körper $GF(2^{128})$ berechnet und wird am Ende des Chiffretextes angehängt. Es wird das bekannte Polynom: $x^{128} + x^7 + x^2 + x + 1$ verwendet.
- Die Blockgröße ist 128Bit (d. h. die AES-Blockgröße).
- H ist der Hash Key: $H = E(K, 0^{128})$ (wobei E die AES-Verschlüsselung ist).
- $mult$ ist Multiplikation im Körper $GF(2^{128})$.
- Die optionalen authentifizierten Daten A werden zum Beispiel benötigt, um den Kontext einer Nachricht zu erfassen (und zum Beispiel Replay-Attacken vorzubeugen). Ein konkretes Beispiel könnte die Ziel-IP-Adresse sein, wenn die Nachricht über das Internet übertragen wird.

Hinweis

Die Visualisierung stellt nur zwei Schritte dar; eine Erweiterung auf n-Blöcke ist jedoch offensichtlich.

Übung

0.1. AES-GCM

Warum ist es wichtig, dass der IV bei AES-GCM nur einmal verwendet wird?