Betriebsmodi bei Blockchiffren

Dozent: Prof. Dr. Michael Eichberg

Version: 2024-03-08

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

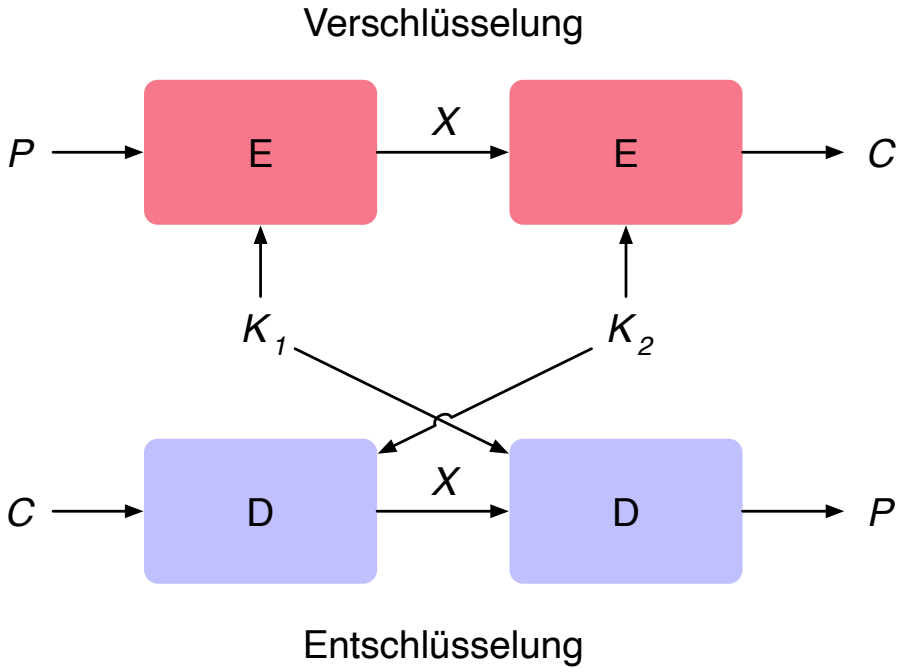
Folien: https://delors.github.io/sec-blockchiffre-operationsmodi/folien.rst.html
https://delors.github.io/sec-blockchiffre-operationsmodi/folien.rst.html.pdf

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

1. Einschub: Mehrfachverschlüsselung

Prof. Dr. Michael Eichberg

Doppelte Verschlüsselung

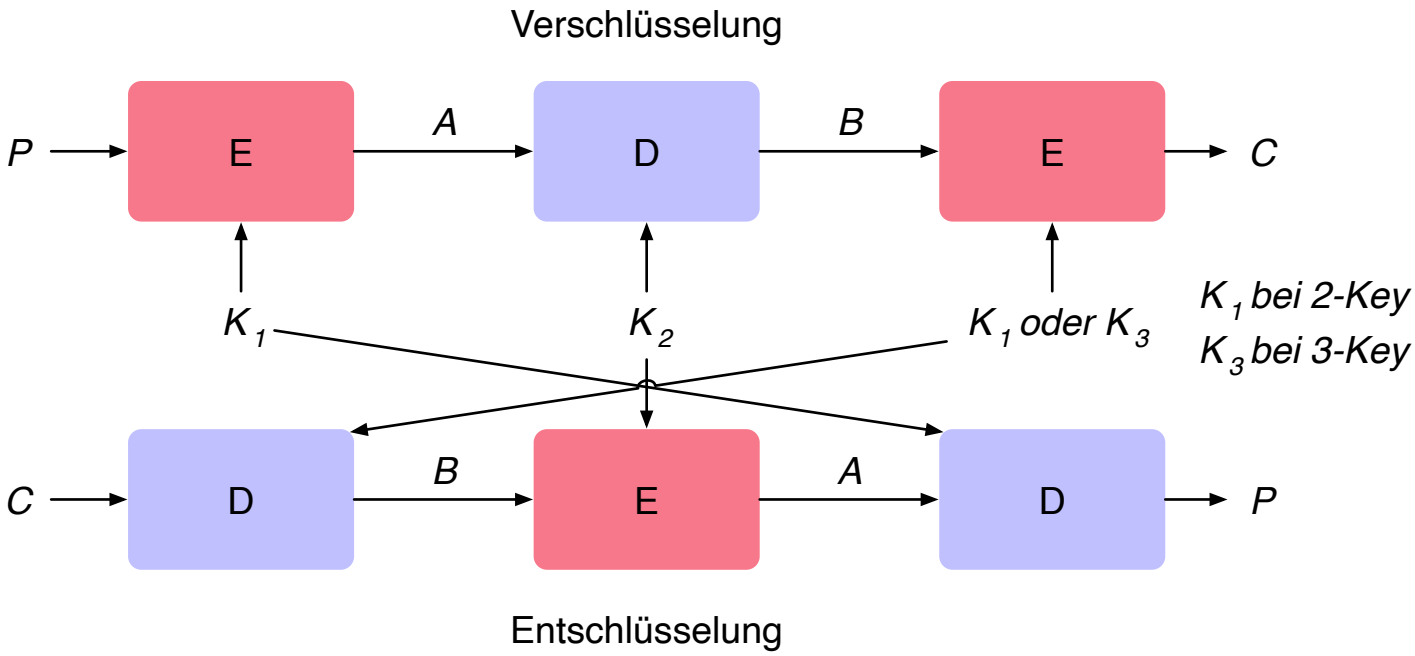


Meet-in-the-Middle-Angriff

- Beobachtung: $E(K_2, E(K_1, P)) = E(K_3, P)$ ist nicht gültig. D. h. die zweifache Anwendung von DES führt zu einer Abbildung, die nicht äquivalent zu einer einfachen DES-Verschlüsselung ist.
- Der Meet-in-the-Middle-Algorithmus greift dieses Verfahren an. Er hängt nicht von einer bestimmten Eigenschaft von DES ab, sondern funktioniert gegen jede Blockchiffre.
- Das Ergebnis ist, dass ein bekannter Klartextangriff gegen Doppel-DES mit einem Aufwand in der Größenordnung von 2^{56} erfolgreich ist, verglichen mit 2^{55} für einen einfachen DES.

Dreifache Verschlüsselung

(Z. B. Triple-DES (3DES) mit drei Schlüsseln)



Triple-DES mit zwei Schlüsseln

Die offensichtliche Antwort auf den *Meet-in-the-middle*-Angriff ist die dreifache Verschlüsselung mit drei verschiedenen Schlüsseln.

- Dies erhöht die Kosten des *Meet-in-the-Middle*-Angriffs auf $\mathbf{2^{112}}$, was jenseits dessen liegt, was praktikabel ist.
- Das hat den Nachteil, dass eine Schlüssellänge von $\mathbf{56\,bits \times 3 = 168\,bits}$ erforderlich ist, was etwas unhandlich sein kann.
- Als Alternative schlug Tuchman eine dreifache Verschlüsselungsmethode vor, die nur zwei Schlüssel verwendet.
- 3DES mit zwei Schlüsseln war eine Alternative zu DES und wurde in die Schlüsselverwaltungsstandards ANSI X9.17 und ISO 8732 aufgenommen.

Triple-DES mit drei Schlüsseln

- Es wurden mehrere Angriffe gegen 3DES mit 2 Schlüsseln entwickelt, die jedoch (noch) nicht praktikabel sind.
- Viele Forscher sind inzwischen der Meinung, dass 3DES mit drei Schlüsseln die bevorzugte Alternative ist.
- 3DES mit drei Schlüsseln hat eine effektive Schlüssellänge von 168 Bit und ist definiert als:

$$C = E(K_3, D(K_2, E(K_1, P)))$$

- Rückwärtskompatibilität mit DES ist gegeben, wenn man $K_3 = K_2$ oder $K_1 = K_2$ einsetzt.

2. Betriebsmodi bei Blockchiffren

Prof. Dr. Michael Eichberg

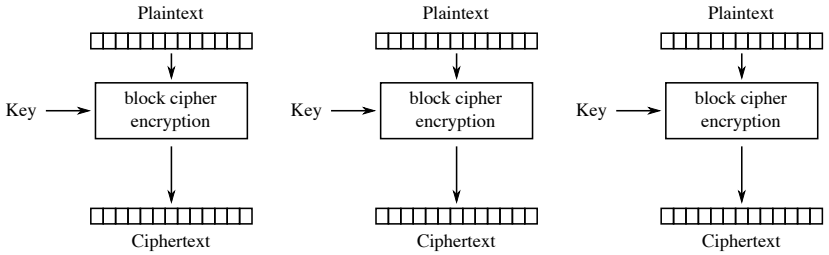
Betriebsmodi

- Eine Technik zur Verbesserung der Wirkung eines kryptografischen Algorithmus oder zur Anpassung des Algorithmus an ein Anwendungsszenario. Insbesondere in Abhängigkeit von der Länge des Klartexts.
- Um eine Blockchiffre in einer Vielzahl von Anwendungen einsetzen zu können, hat das NIST fünf Betriebsmodi definiert.
  - Die fünf Modi decken eine breite Palette von Verschlüsselungsanwendungen ab, für die eine Blockchiffre verwendet werden kann.
  - Diese Modi sind für die Verwendung mit jeder symmetrischen Blockchiffre vorgesehen, einschließlich 3DES und AES.

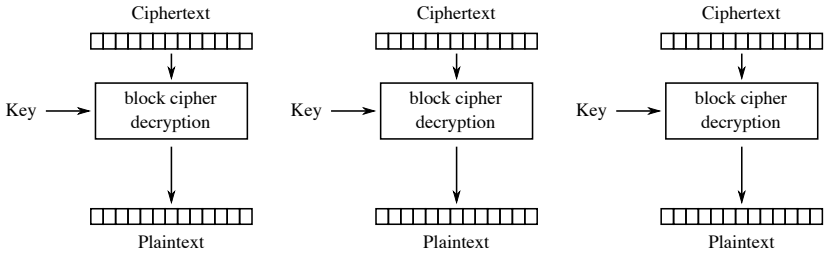
Betriebsmodi - Übersicht

Modus	Beschreibung	Typische Anwendung
Electronic Codebook (ECB)	Jeder Block von Klartextbits wird unabhängig voneinander mit demselben Schlüssel verschlüsselt.	▪ Sichere Übertragung einzelner Werte (z. B. eines Verschlüsselungsschlüssels)
Cipher Block Chaining (CBC)	Die Eingabe für den Verschlüsselungsalgorithmus ist die XOR-Verknüpfung des nächsten Klartextblocks mit dem vorangegangenen Chiffretextblock.	▪ Universelle blockorientierte Übertragung ▪ Authentifizierung
Cipher Feedback (CFB)	Die Eingabe wird Bit für Bit verarbeitet. Der vorhergehende Chiffretext wird als Eingabe für den Verschlüsselungsalgorithmus verwendet, um eine pseudozufällige Ausgabe zu erzeugen, die mit dem Klartext XOR-verknüpft wird, um die nächste Einheit des Chiffretextes zu erzeugen.	▪ Allgemeine stromorientierte Übertragung ▪ Authentifizierung
Output Feedback (OFB)	Ähnlich wie CFB, mit dem Unterschied, dass die Eingabe für den Verschlüsselungsalgorithmus die vorangegangene Verschlüsselungsausgabe ist, und volle Blöcke verwendet werden.	▪ Stromorientierte Übertragung über verrauschte Kanäle (z. B. Satellitenkommunikation)
Counter (CTR)	Jeder Klartextblock wird mit einem verschlüsselten Zähler XOR-verknüpft. Der Zähler wird für jeden nachfolgenden Block erhöht.	▪ Blockorientierte Übertragung für allgemeine Zwecke ▪ Nützlich für Hochgeschwindigkeitsanforderungen

Electronic Codebook



Electronic Codebook (ECB) mode encryption



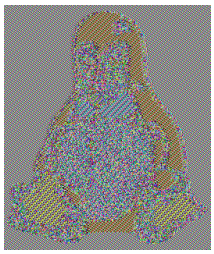
Electronic Codebook (ECB) mode decryption

Autor: https://commons.wikimedia.org/wiki/User:WhiteTimberwolf

Probleme bei der Verwendung der Verschlüsselung im ECB-Modus

ECB-Tux - der Linux-Pinguin verschlüsselt im ECB-Modus:

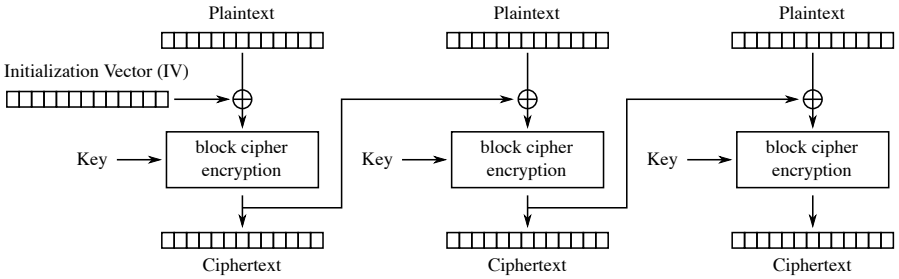
Quelle: https://github.com/robertdavidgraham/ecb-penguin



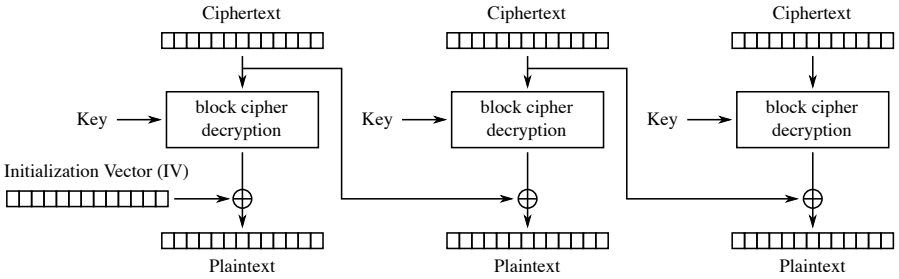
Kriterien und Eigenschaften für die Bewertung und Konstruktion von Blockchiffre-Betriebsarten, die ECB überlegen sind.

- Overhead
- Fehlerbehebung
- Fehlerfortpflanzung
- Streuung
- Sicherheit

Cipher Block Chaining



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Autor: https://commons.wikimedia.org/wiki/User:WhiteTimberwolf

Konvertierung von Blockchiffren in Stromchiffre

Bei AES, DES oder jeder anderen Blockchiffre erfolgt die Verschlüsselung immer Block-für-Block mit Blockgrößen von b Bits:

- Im Fall von (3)DES: $b = 64$
- Im Fall von AES: $b = 128$

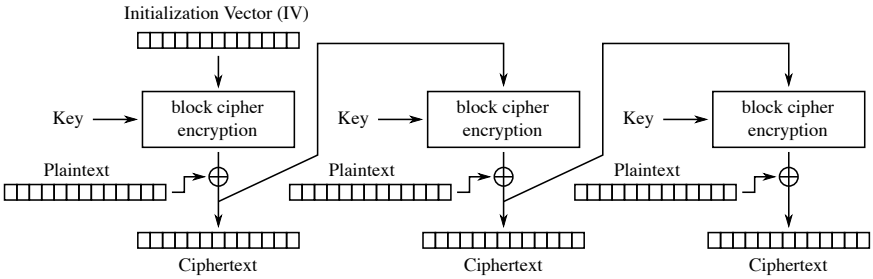
Hinweis

Es gibt drei Modi, die es ermöglichen, eine Blockchiffre in eine zeichenorientierte Stromchiffre umzuwandeln:

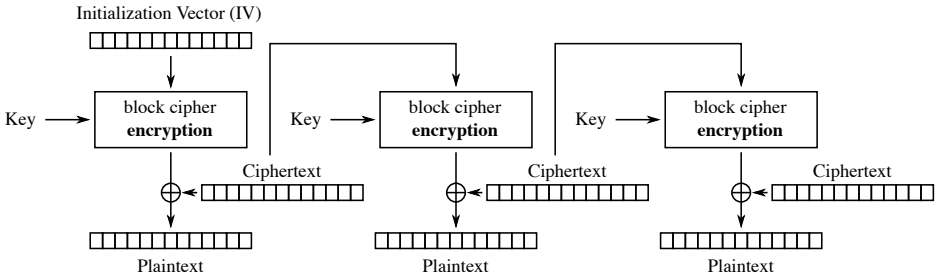
- Cipher Feedback Mode (CFB)
- Output Feedback Mode (OFB)
- Counter Mode (CTR)

D. h., es ist kein Auffüllen (*Padding*) erforderlich, wenn die Nachricht nicht ein Vielfaches der Blockgröße ist.

Cipher Feedback Mode



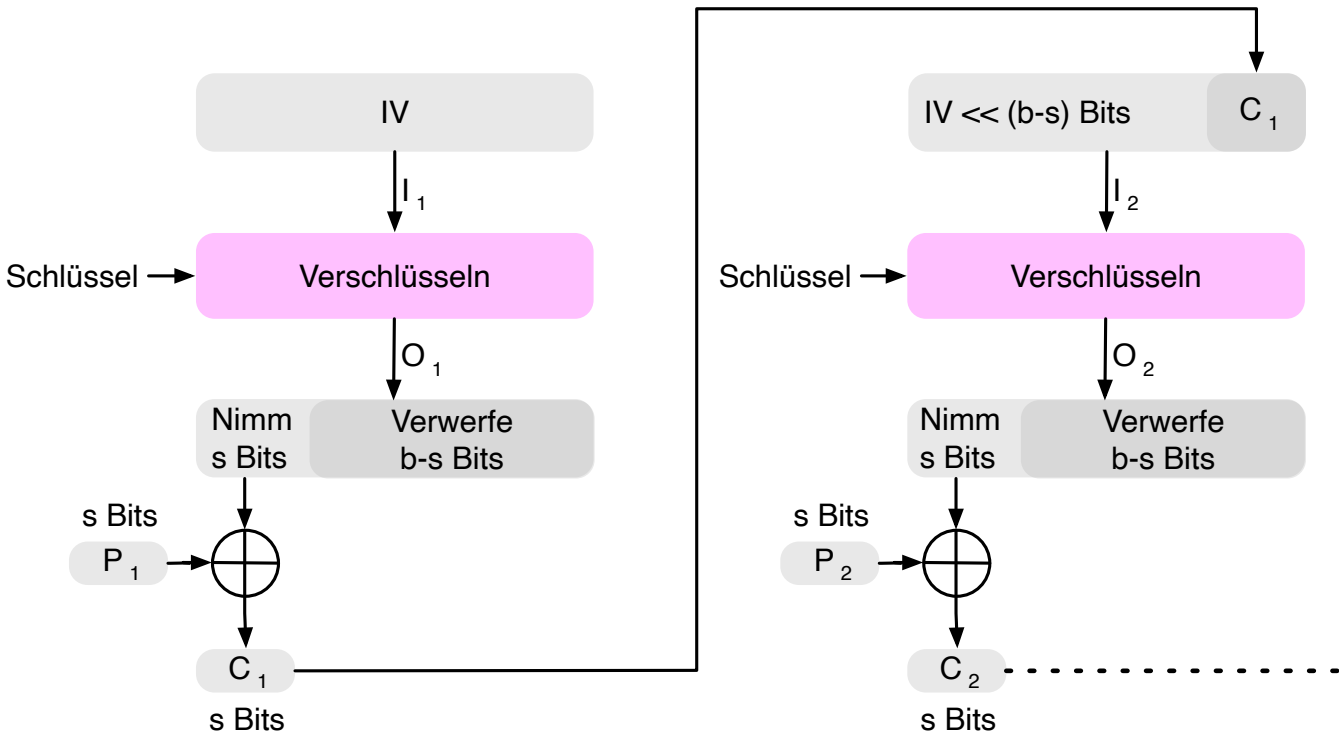
Cipher Feedback (CFB) mode encryption



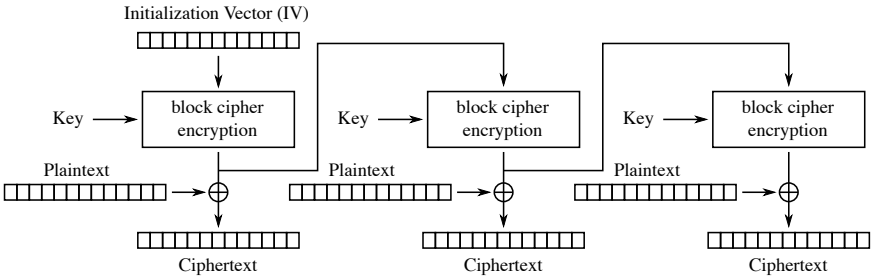
Cipher Feedback (CFB) mode decryption

Autor: https://commons.wikimedia.org/wiki/User:WhiteTimberwolf

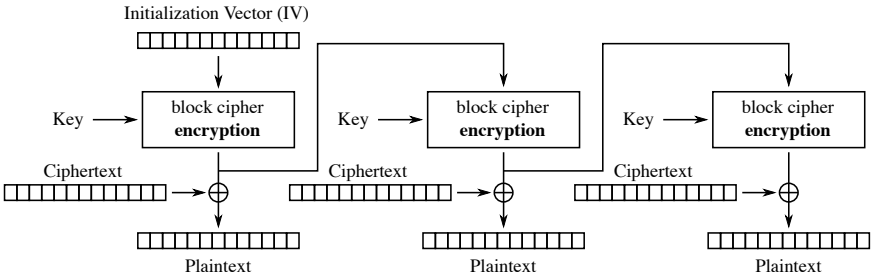
Cipher Feedback Mode als Stromchiffre



Output Feedback Mode



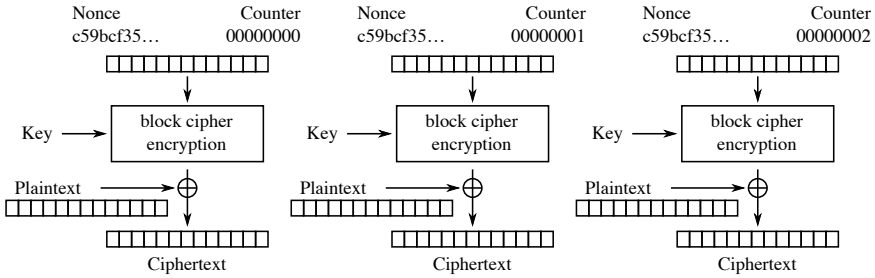
Output Feedback (OFB) mode encryption



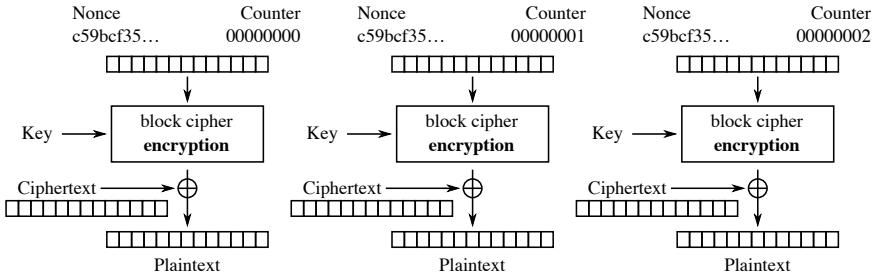
Output Feedback (OFB) mode decryption

Autor: https://commons.wikimedia.org/wiki/User:WhiteTimberwolf

Counter Mode



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Autor: https://commons.wikimedia.org/wiki/User:WhiteTimberwolf

Counter Mode - Vorteile

Hardware-Effizienz:
kann von der Parallelisierung der Hardware profitieren

Software-Effizienz:
leicht parallelisierbar in Software

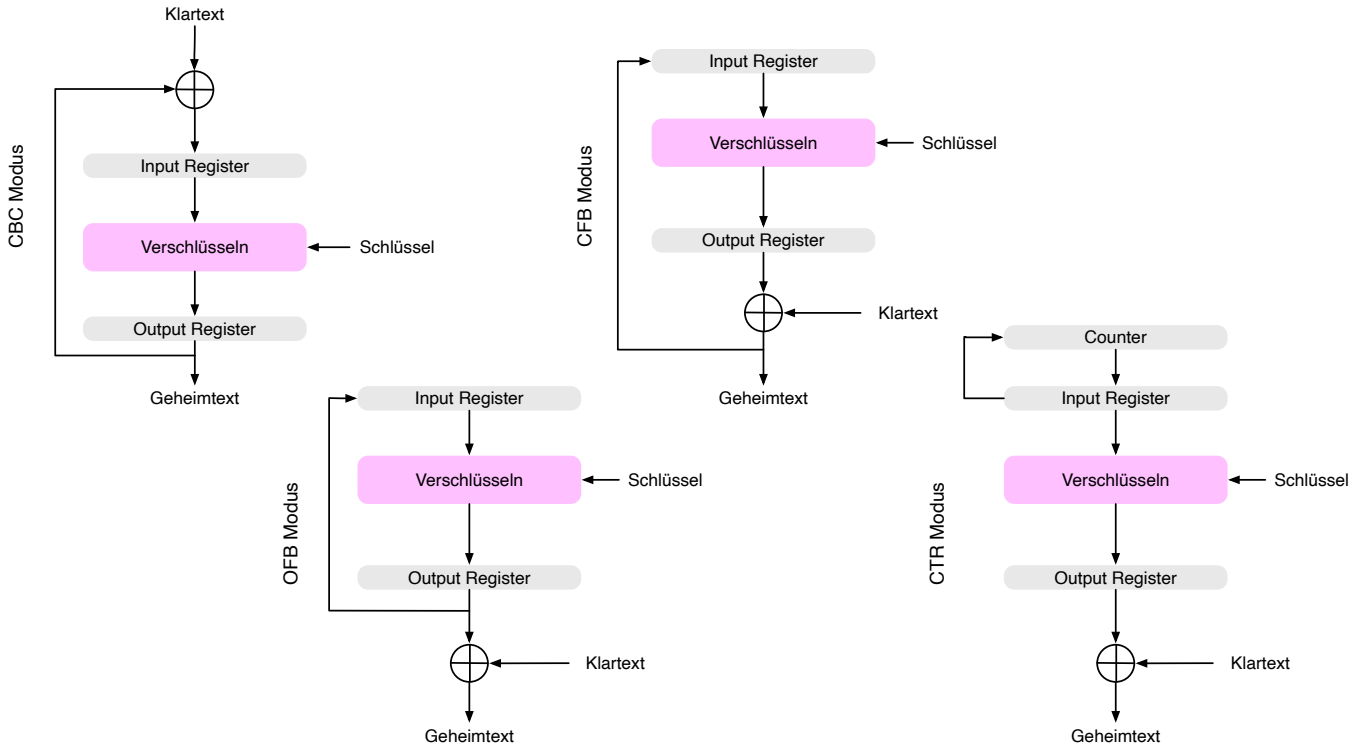
Vorverarbeitung: die Verschlüsselung der Zähler

Zufälliger Zugriff: Der i-te Block des Klartextes/des Chiffretextes kann im Zufallszugriff verarbeitet werden

Nachweisbare Sicherheit:
genauso sicher wie die anderen Verfahren.

Einfachheit: Es wird nur der Verschlüsselungsalgorithmus benötigt.

Rückkopplungseigenschaften[1] der Betriebsmodi



[1] (*Feedback Characteristics*)

XTS-AES Modus für blockorientierte Speichergeräte

2010 vom NIST als zusätzlicher Blockchiffre-Betriebsmodus genehmigt.

Modus ist auch ein IEEE-Standard, IEEE Std 1619-2007

- Die Norm beschreibt eine Verschlüsselungsmethode für Daten, die in sektorbasierten Geräten gespeichert sind, wobei das Bedrohungsmodell einen möglichen Zugriff des Gegners auf die gespeicherten Daten beinhaltet.
- Hat breite Unterstützung der Industrie erhalten.

Frage

Welche potenziellen Bedrohungen sind relevant?

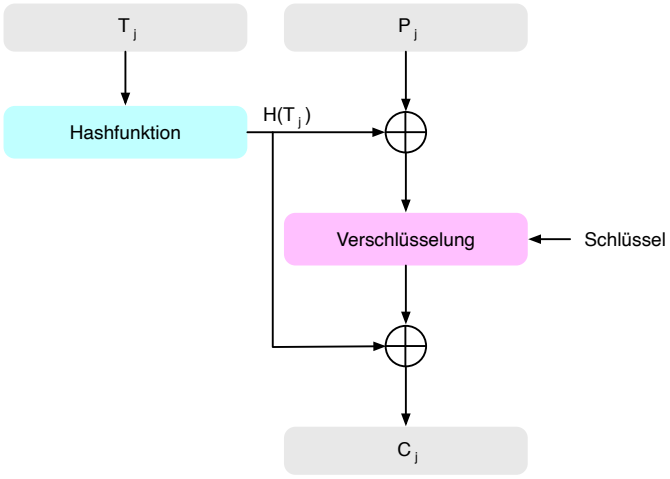
Tweakable Blockchiffren

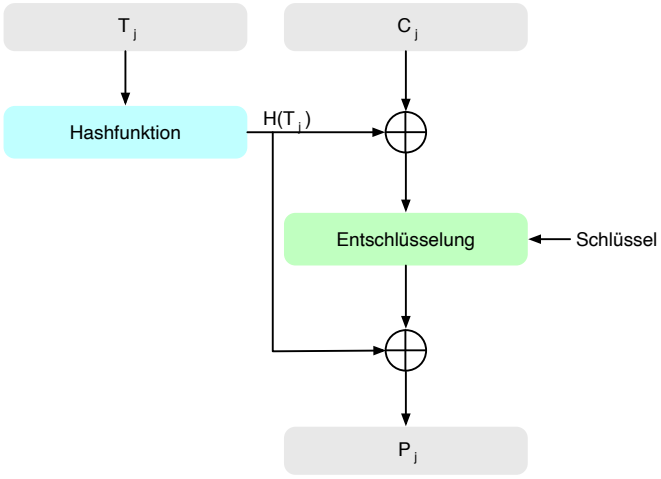
- Der XTS-AES-Modus basiert auf dem Konzept einer veränderbaren (🇺🇸 *tweakable*) Blockchiffre.
- Allgemeine Struktur:

  Um Chiffriertextes a zu berechnen, wird benötigt:

  - **Klartext**
  - **Symmetrischer Schlüssel**
  - **Tweak**
- Der *Tweak* muss nicht geheim gehalten werden; der Zweck ist, Variabilität zu bieten.

Tweakable Blockchiffren





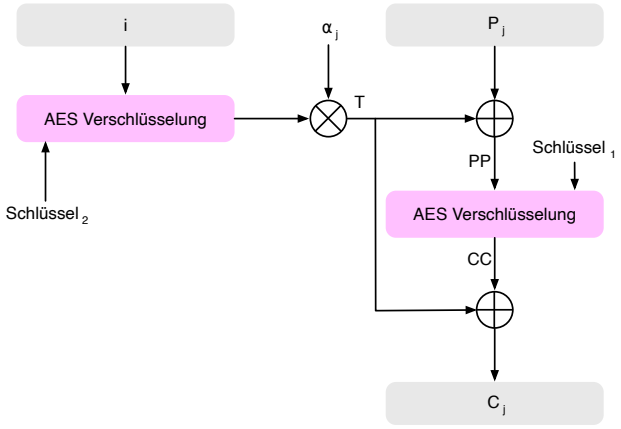
Anforderungen an die Speicherverschlüsselung

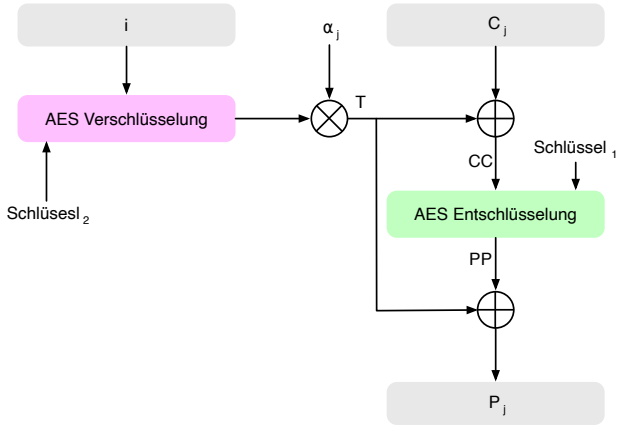
Die Anforderungen an die Verschlüsselung gespeicherter Daten, die auch als *data at rest* bezeichnet werden, unterscheiden sich von denen für übertragene Daten.

Die Norm P1619 wurde in Hinblick auf folgende Eigenschaften entwickelt:

- Der Chiffretext ist für einen Angreifer frei verfügbar.
- Das Datenlayout wird auf dem Speichermedium und beim Transport nicht verändert.
- Der Zugriff auf die Daten erfolgt in Blöcken fester Größe und unabhängig voneinander.
- Die Verschlüsselung erfolgt in 16-Byte-Blöcken, die unabhängig voneinander sind.
- Es werden keine weiteren Metadaten verwendet, außer der Position der Datenblöcke innerhalb des gesamten Datensatzes.
- Derselbe Klartext wird an verschiedenen Stellen in verschiedene Chiffretexte verschlüsselt, aber immer in denselben Chiffretext, wenn er wieder an dieselbe Stelle geschrieben wird.
- Ein standardkonformes Gerät kann für die Entschlüsselung von Daten konstruiert werden, die von einem anderen standardkonformen Gerät verschlüsselt wurden.

XTS-AES Operation auf einem Block





- Key: Der Schlüssel wobei gilt: $Key = Key_1 \,||\, Key_2$
- P_j: Der j-te Block des Klartexts. Alle Blöcke haben eine Länge von 128 bits. Eine Klartextdateneinheit - in der Regel ein Festplattensektor - besteht aus einer Folge von Klartextblöcken.
- C_j: Der j-te Block des Chiffretextes.
- j: Die fortlaufende Nummer des 128-Bit-Blocks innerhalb der Dateneinheit.
- i: Der Wert des 128-Bit-Tweaks.
- α: Ein primitives Element des $GF(2^{128})$ welches dem Polynom x (d. h. 0000...0010) entspricht.
- α^j: α j mal mit sich selbst multipliziert im Körper $GF(2^{128})$
- $\oplus$ Bitwise XOR
- $\otimes$ Modulare Multiplikation mit Binärkoeffizienten modulo $x^{128} + x^7 + x^2 + x + 1$.

Übung

Warum ist es bei CBC wichtig, den Initialisierungsvektor (IV) zu schützen?

In welchen Betriebsarten ist eine Auffüllung (🇺🇸 *Padding*) notwendig?

Was geschieht im Falle eines Übertragungsfehlers (einzelner Bitflip im Chiffretext) bei ECB, CBC, CFB, OFB, CTR?

Warum muss der IV im Falle von OFB eine Nonce (🇺🇸 *Number used ONCE*) sein (d. h. eine Zahl, die nur einmal für die Ausführung des Verschlüsselungsalgorithmus verwendet wird)?

Sie möchten feststellen, ob ein Programm zur Verschlüsselung von Dateien den ECB-Modus verwendet. Was müssen Sie tun?

Warum ist es bei CBC wichtig, den Initialisierungsvektor (IV) zu schützen?

🔑

In welchen Betriebsarten ist eine Auffüllung (🇺🇸 *Padding*) notwendig?

🔑

Was geschieht im Falle eines Übertragungsfehlers (einzelner Bitflip im Chiffretext) bei ECB, CBC, CFB, OFB, CTR?

🔑

Warum muss der IV im Falle von OFB eine Nonce (🇺🇸 *Number used ONCE*) sein (d. h. eine Zahl, die nur einmal für die Ausführung des Verschlüsselungsalgorithmus verwendet wird)?

🔑

Sie möchten feststellen, ob ein Programm zur Verschlüsselung von Dateien den ECB-Modus verwendet. Was müssen Sie tun?

🔑

Übung

Verwenden Sie den OFB-Modus in Kombination mit einer Caesar-Chiffre. Die Blockgröße ist ein einzelnes Zeichen. Der Schlüssel ist die Anzahl der Zeichen, um die Sie ein Zeichen verschieben wollen - wie zuvor. Die IV ist ein Zeichen. Damit sie ein XOR durchführen können, ordnen wir jedem Zeichen einen Wert zu und erweitern das Alphabet um die Ziffern 1 bis 3, "!", "?" und das "_". Auf diese Weise ist es immer möglich, ein sinnvolles Zeichen auszugeben.

Daraus ergibt sich die folgende Kodierung:

Index	Zeichen	Binärdarstellung
0	A	00000
1	B	00001
2	C	00010
3	D	00011
4	E	00100
5	F	00101
6	G	00110
7	H	00111
8	I	01000
9	J	01001
10	K	01010

Index	Zeichen	Binärdarstellung
11	L	01011
12	M	01100
13	N	01101
14	O	01110
15	P	01111
16	Q	10000
17	R	10001
18	S	10010
19	T	10011
20	U	10100
21	V	10101

Index	Zeichen	Binärdarstellung
22	W	10110
23	X	10111
24	Y	11000
25	Z	11001
26	1	11010
27	2	11011
28	3	11100
29	!	11101
30	?	11110
31	_	11111

Verschlüsseln Sie nun einige Nachrichten mit dieser Chiffre. Welchen Effekt hat die Anwendung des OFB-Modus auf die Nachrichten?

Verwenden Sie den OFB-Modus in Kombination mit einer Caesar-Chiffre. Die Blockgröße ist ein einzelnes Zeichen. Der Schlüssel ist die Anzahl der Zeichen, um die Sie ein Zeichen verschieben wollen - wie zuvor. Die IV ist ein Zeichen. Damit sie ein XOR durchführen können, ordnen wir jedem Zeichen einen Wert zu und erweitern das Alphabet um die Ziffern 1 bis 3, "!", "?" und das "_". Auf diese Weise ist es immer möglich, ein sinnvolles Zeichen auszugeben.

Daraus ergibt sich die folgende Kodierung:

Index	Zeichen	Binärdarstellung
0	A	00000
1	B	00001
2	C	00010
3	D	00011
4	E	00100
5	F	00101
6	G	00110
7	H	00111
8	I	01000
9	J	01001
10	K	01010

Index	Zeichen	Binärdarstellung
11	L	01011
12	M	01100
13	N	01101
14	O	01110
15	P	01111
16	Q	10000
17	R	10001
18	S	10010
19	T	10011
20	U	10100
21	V	10101

Index	Zeichen	Binärdarstellung
22	W	10110
23	X	10111
24	Y	11000
25	Z	11001
26	1	11010
27	2	11011
28	3	11100
29	!	11101
30	?	11110
31	_	11111

Verschlüsseln Sie nun einige Nachrichten mit dieser Chiffre. Welchen Effekt hat die Anwendung des OFB-Modus auf die Nachrichten?