Symmetrische Kryptografie

Betriebsmodi von Blockchiffren

und was man sonst damit machen kann

Martin

Chaos Computer Club Cologne e.V.
https://koeln.ccc.de

12. Oktober 2015

Definition

Krypto-System

Tupel (M, C, K, E, D)

Message, Ciphertext, Key,

Encryption-Function, Decryption-Function

Verschlüsselung

$\mathrm{Enc}_K(M) \Rightarrow C$

Entschlüsselung

$\mathrm{Dec}_K(C) \Rightarrow M$

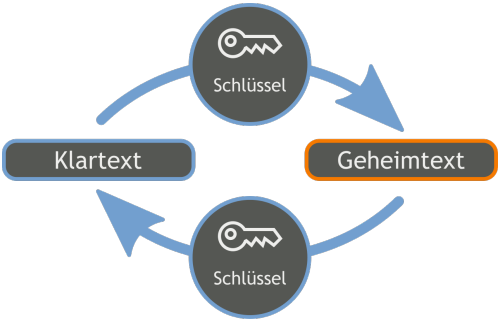


Abbildung: Wikipedia, Bananenfalter

How to blockcipher?

1. Die Welt einigt sich auf Algorithmus und Schlüssellänge $length
2. Wähle Schlüssel mit Länge $length
3. Teile Klartext in $length lange Blöcke - Padding nötig

How to blockcipher?

1. Die Welt einigt sich auf Algorithmus und Schlüssellänge $length
2. Wähle Schlüssel mit Länge $length
3. Teile Klartext in $length lange Blöcke - Padding nötig

Zerlegung und Padding

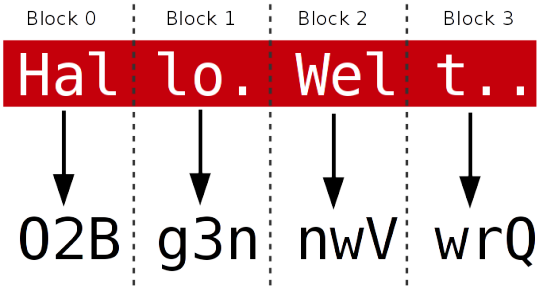
Aufgabe

- Klartext sei “Hallo Welt”
- Blocklänge $length sei 3

Zerlegung und Padding mit Beispiel

Aufgabe

- Klartext sei “Hallo Welt”
- Blocklänge $length sei 3



Block-Zerlegung

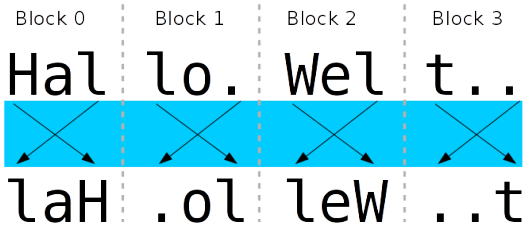
1. Die Welt einigt sich auf Algorithmus und Schlüssellänge $length
2. Wähle Schlüssel mit Länge $length
3. Teile Klartext in $length Blöcke - Padding nötig
4. Verschlüssele alle Blöcke - Aber Wie?

Block-Zerlegung

1. Die Welt einigt sich auf Algorithmus und Schlüssellänge $length
2. Wähle Schlüssel mit Länge $length
3. Teile Klartext in $length Blöcke - Padding nötig
4. Verschlüssele alle Blöcke - Aber Wie?

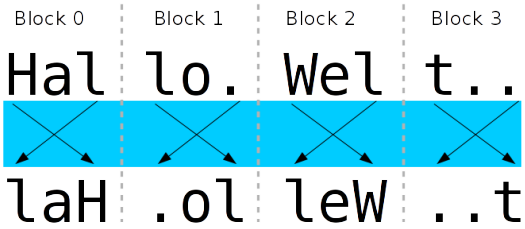
Beispiel

- Klartext sei “Hallo Welt”
- Blocklänge $length sei 3
- Algorithmus: tausche Buchstaben
- Schlüssel: {{0,2}} tausche ersten und letzten Buchstaben im Block
- Modus **ECB**

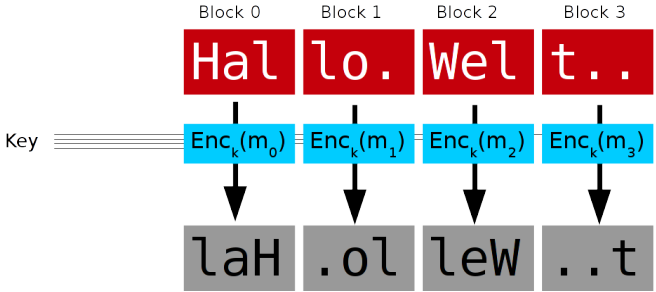


Beispiel

- Klartext sei “Hallo Welt”
- Blocklänge $length sei 3
- Algorithmus: tausche Buchstaben
- Schlüssel: {{0,2}} tausche ersten und letzten Buchstaben im Block
- Modus **ECB**



ECB - Electronic Codebook



Alle Blöcke werden der Reihe nach mit dem selben Schlüssel verschlüsselt

Angriff ECB

Klingt ganz cool... ist es aber nicht

Aufgabe

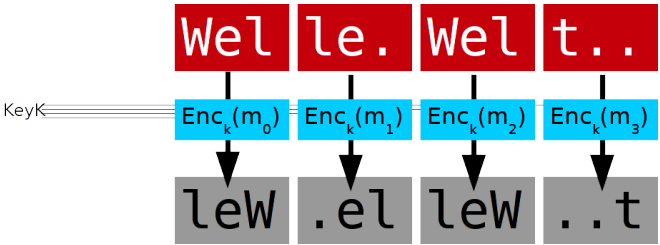
Warum?

Angriff ECB

Klingt ganz cool... ist es aber nicht

Aufgabe

Warum?



$$m_x = m_y \Longrightarrow E_k(m_x) = E_k(m_y) \Longleftrightarrow c_x = c_y$$

Angriff ECB mit bunten Bildchen

Aufgabe

1. Bitmap binär einlesen
2. Gegebenenfalls Padding beachten
3. Verschlüsseln
4. Dateiheader “wiederherstellen”

Angriff ECB - Lösung OpenSSL

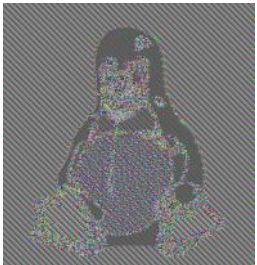
Lösung mit OpenSSL

1. openssl enc -aes-128-ecb -in plain.bmp -out encrypted.bmp
   - openssl: openssl aufrufen
   - enc: encryption
   - in: inputfile, out: outputfile
2. dd if=plain.bmp of=encrypted.bmp bs=1 count=32 conv=notrunc
   - dd: erstellt bitweise Kopie
   - if: inputfile, of: outputfile
   - bs: Wie viele Bytes sollen gleichzeitig kopiert werden?
   - count: Anzahl der zu Kopierenden Blöcke bmp header hat ca. 30 Byte...
   - conv=notrunc: outputfile nicht nach count Blöcken abschneiden

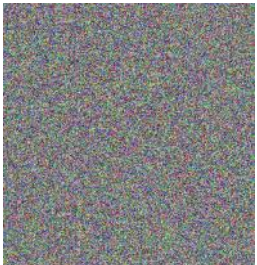
Python-Lösung kommt später... irgendwann...

Angriff ECB visualisiert

(a) Tux von Larry Ewing lewing@isc.tamu.edu mithilfe von The GIMP



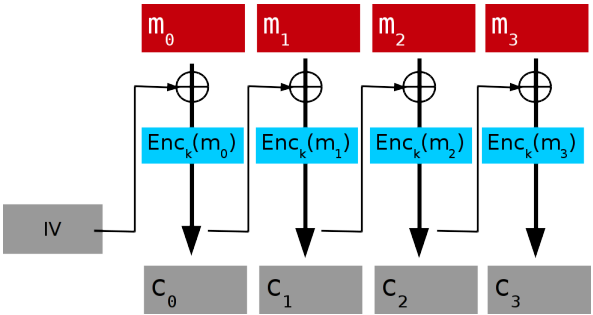
(b) Wikipedia, Lunkwill; Tux von Larry Ewing



(c) Wikipedia, Lunkwill; Tux von Larry Ewing

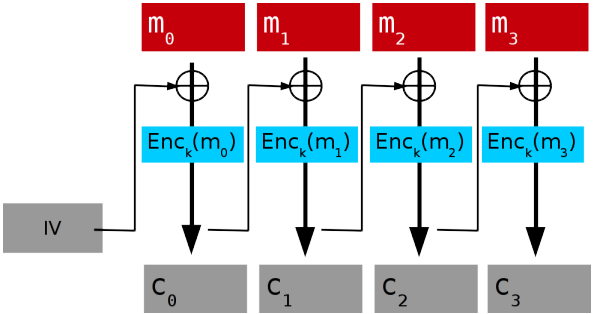
Abbildung: Block cipher mode of operation. Wikipedia, The Free Encyclopedia. Retrieved 16:51, October 3, 2015, from https://en.wikipedia.org/w/index.php?title=Block_cipher_mode_of_operation&oldid=683053162#Electronic_Codebook_.28ECB.29

CBC - Cipher Block Chaining



- Jeder Ciphertext geht in die nachfolgende Verschlüsselung mit ein
- Entschlüsseln nur noch in richtiger Reihenfolge; Blockchain
- IV muss nicht geheim gehalten werden ABER nicht doppelt verwenden sonst

CBC - Cipher Block Chaining



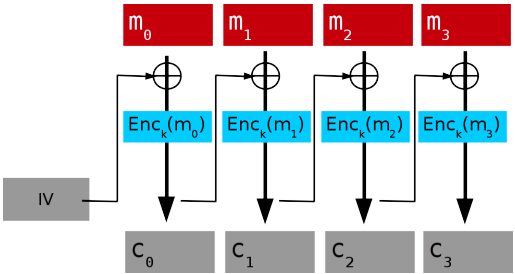
- Jeder Ciphertext geht in die nachfolgende Verschlüsselung mit ein
- Entschlüsseln nur noch in richtiger Reihenfolge; Blockchain
- IV muss nicht geheim gehalten werden ABER nicht doppelt verwenden sonst

CBC - Cipher Block Chaining

Aufgabe

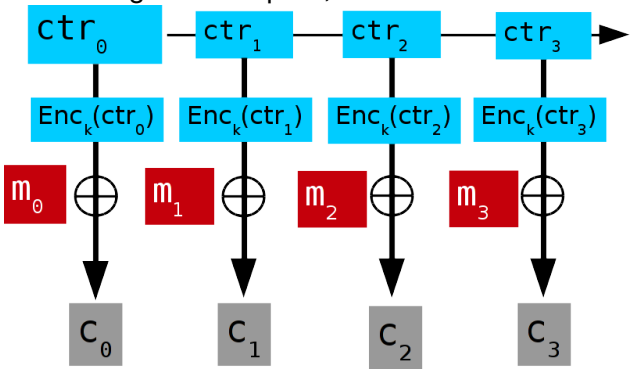
Was passiert bei Mehrfachverwendung von IV?

Zur Erinnerung:



CTR - Counter Mode

Böse Zungen behaupten, es handelt sich um einen Stream Cipher



CTR Notizen

- guter Counter beinhaltet eine Nonce: $(\text{Nonce} \parallel \mathit{Counter})$
- Ver- und Entschlüsselung mit selber Operation: Counter **ver**schlüsseln und XOR
- vgl. mit OTP (One-Time Pad); Schlüsselstrom kann durch Counter “errechnet” werden

Anmerkung: Gute Blockchiffren können als PRNG “missbraucht” werden

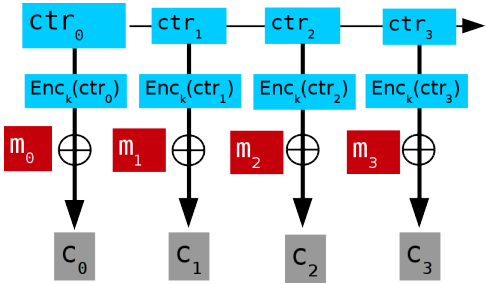
yanosz sagt...

Das gilt auch für AES Ciphertext-Blöcke

CTR Sicherheit?

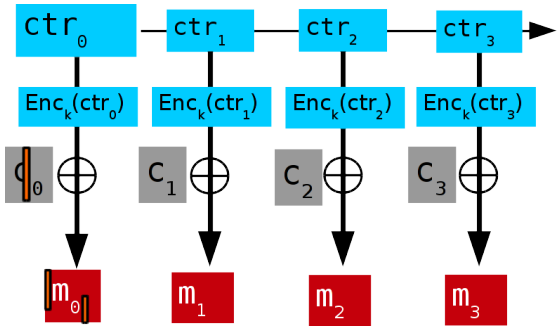
Aufgabe

OTP ist sicher, 1-Block-ECB ist sicher.
Ist CTR sicher?



Ja aber...

Hinweis: Das ist die Entschlüsselung



Modifikation im Ciphertext unentdeckt.

Sicherheitsziele

Aufgabe

Die drei klassischen Sicherheitsziele lauten...?

- Vertraulichkeit
- Integrität
- Verfügbarkeit

Was ist gegeben?

Sicherheitsziele

Aufgabe

Die drei klassischen Sicherheitsziele lauten...?

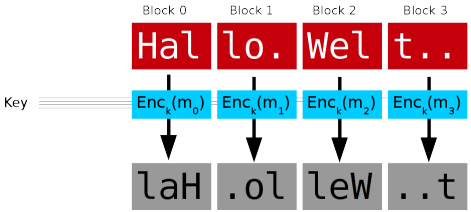
- Vertraulichkeit
- Integrität
- Verfügbarkeit

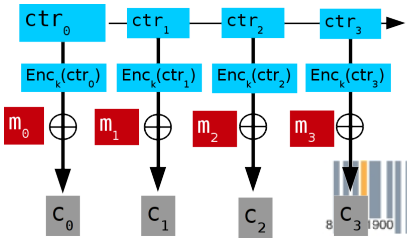
Was ist gegeben?

- Alice will Bob eine Nachricht schreiben
- Die Nachricht wird verschlüsselt übertragen
- Bob kann die Nachricht lesen

Aufgabe

Weiß Bob ob die Nachricht manipuliert wurde?





Message Authentication Codes

Wie kann man Integrität gewährleisten?

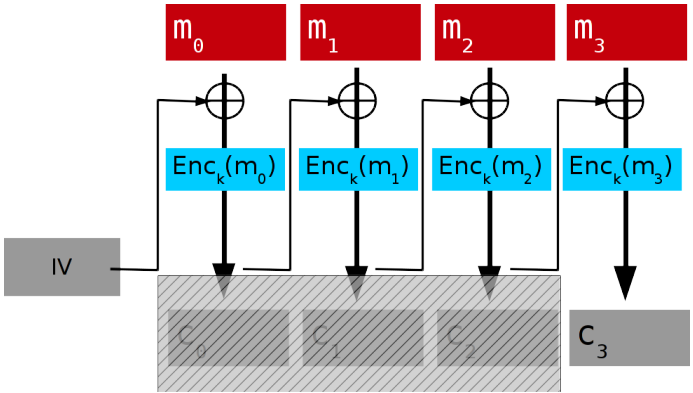
Message Authentication Code

- *Alice hängt generiert einen MAC:* $(m, \mathrm{MAC}_k(m))$
- *Bob empfängt die Nachricht:* (m, y)
- *Bob berechnet den MAC selbst und vergleicht:* $\mathrm{MAC}_k(m) == y?$

$\Longrightarrow$ *Alice und Bob müssen ein Geheimniss K teilen*

CBC-MAC

Mit einer Blockchiffre Integrität sicherstellen?

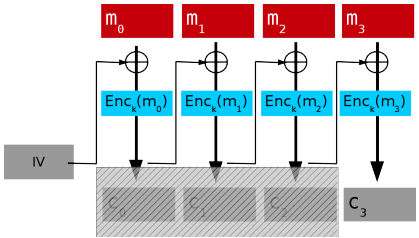


CBC-MAC

Aufgabe

Wie lang ist ein CBC-MAC?

Zur Erinnerung:



Angriff

XOR-Eigenschaften

$A \oplus B \oplus B = A$

1. $(m, \mathrm{MAC}_k(m))$ wurde von jemandem berechnet $m = (m_1 \parallel m_2 \parallel m_3)$
2. $((m_1 \parallel m_2 \parallel m_3), \mathrm{MAC}_k(m))$
3. $\Longrightarrow ((m_1 \parallel m_2 \parallel m_3 \parallel m_1 \oplus \mathrm{MAC}_k(m)) \parallel m_2 \parallel m_3), \mathrm{MAC}_k(m'))$

yanosz sagt...

Viel einfacher: Mallory könnte den ersten Ciphertext-Block entfernen und den IV durch c_0 ersetzen.

Achtung bei CBC und CBC-MAC

Für CBC und CBC-MAC niemals den selben Schlüssel verwenden

Nach 2,12 Minuten Gleichungen umformen kann man sehen, dass nur noch die Integrität des letzten Blocks sichergestellt ist und Eve die anderen Blöcke ersetzen kann.

Wen das interessiert: https://en.wikipedia.org/wiki/CBC-MAC

Außerdem: Blöcke anhängen immer noch möglich..... $\Longrightarrow$ CMAC benutzen

Vertraulichkeit und Integrität?

Wäre es nicht schön beides gleichzeitig zu haben?
Vertraulichkeit und Integrität
Authenticated Encryption

Vertraulichkeit und Integrität?

Wäre es nicht schön beides gleichzeitig zu haben?
Vertraulichkeit und Integrität
Authenticated Encryption

Auf dem Weg zu GCM

Was bisher geschah...

1. ECB Electronic Code Book Mode (unsicher für mehr als einen Block)
2. CTR Counter Mode
3. CBC Cipher Block Chain Mode
4. CBC-MAC für Integritätssicherung

Die Lösung: **GCM - Galois Counter Mode**

- G für GMAC (Galois MAC)
- CM für CTR Mode

Auf dem Weg zu GCM

Was bisher geschah...

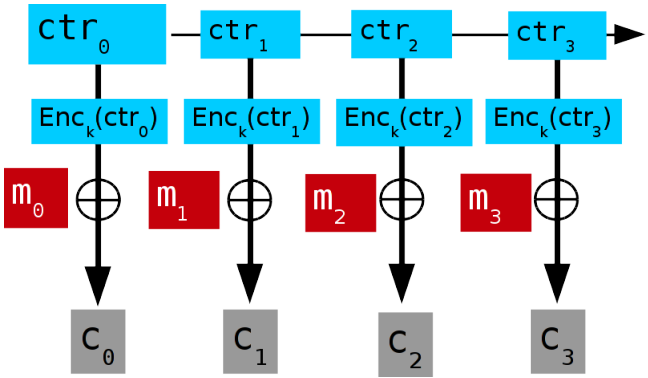
1. ECB Electronic Code Book Mode (unsicher für mehr als einen Block)
2. CTR Counter Mode
3. CBC Cipher Block Chain Mode
4. CBC-MAC für Integritätssicherung

Die Lösung: **GCM - Galois Counter Mode**

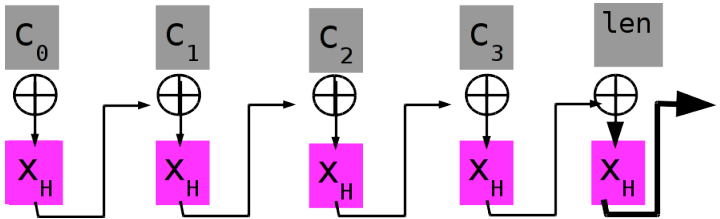
- G für GMAC (Galois MAC)
- CM für CTR Mode

Rückblick CTR

Zur Erinnerung: Counter Mode



GMAC - Galois Message Authentication Code

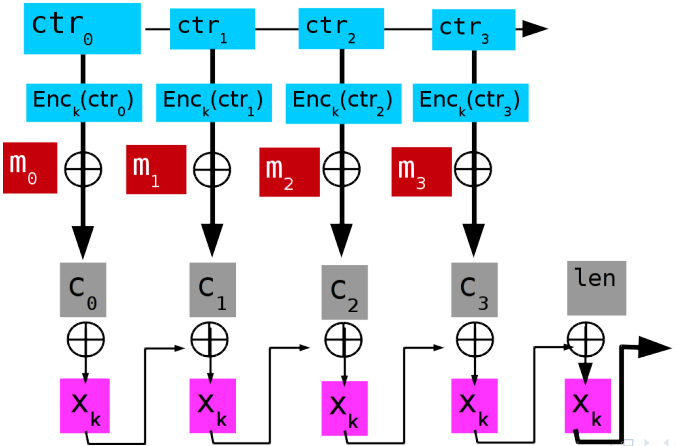


Warum Galois?

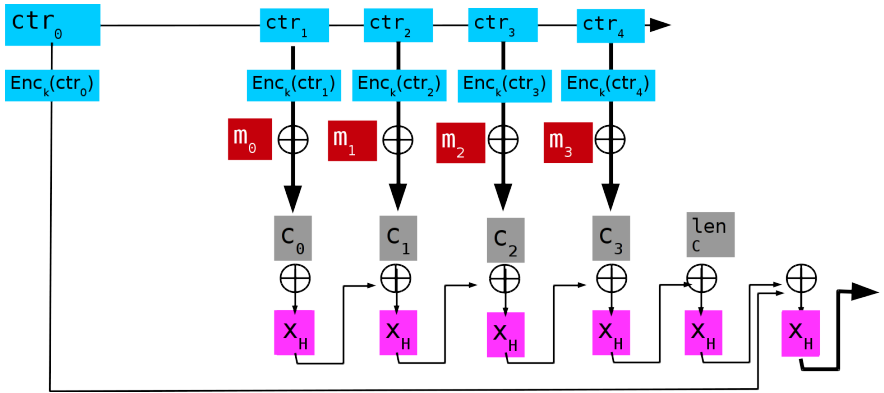
Definition

x ist eine Operation (Polynom-)Multiplikation in $\mathbb{GF}(2^{128})$
$H = \mathrm{Enc}_k(\text{Null-Block})$

CTR und GMAC

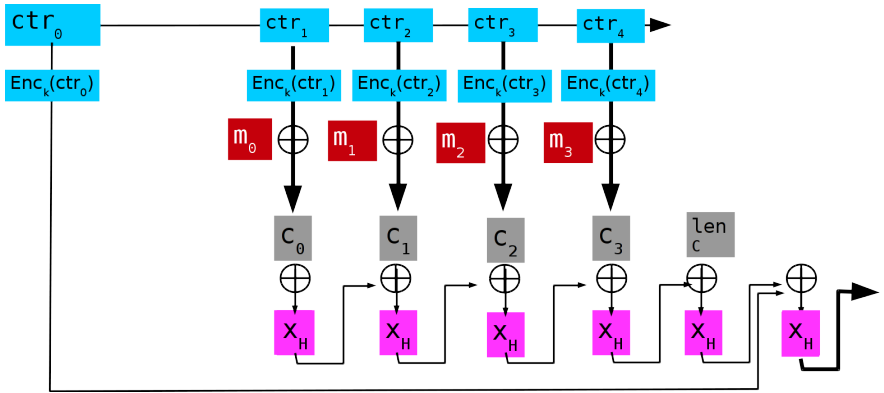

ctr0
ctr1
ctr2
ctr3
Enck(ctr0)
Enck(ctr1)
Enck(ctr2)
Enck(ctr3)
m0
m1
m2
m3
c0
c1
c2
c3
len
Xk
Xk
Xk
Xk
Xk

GCM Authenticated Encryption



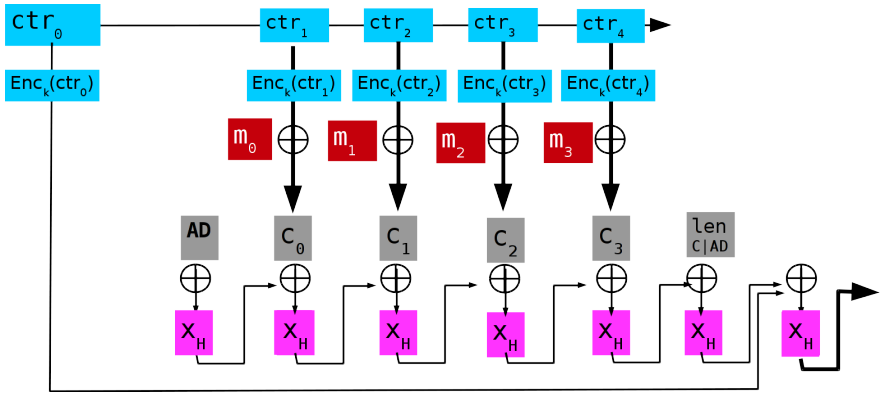
Authenticated Encryption - Aber GCM kann noch mehr

GCM Authenticated Encryption



Authenticated Encryption - Aber GCM kann noch mehr

GCM Authenticated Encryption with Associated Data



Authenticated Encryption with Associated Data

Multiplikation in $\mathbb{GF}(2^{128})$

X_h (lila Box) ist eine (Polynom-) Multiplikation im Galois-Körper$\mathbb{Z}(2^{128})$

1. Ciphertext in Polynom verwandeln
   - n Anzahl der Bits
   - Ciphertext C
   - C $= \sum_{i=0}^{n}(b_n * x^n)$
2. Ciphertext-Polynom mit H (dem “Tag”) multiplizieren
   - H ergibt sich aus Enc_k(Null-Block)
3. Gegebenenfalls Reduktion falls Ergebnis größer als Generatorpolynom $x^{128} + x^7 + x^2 + x + 1$