IT Security

Chapter 2: Encryption (Part 1)

- Symmetric encryption
- Base64 encoding
- Asymmetric encryption
- Practical aspects of encryption
- Random numbers
- Key Management

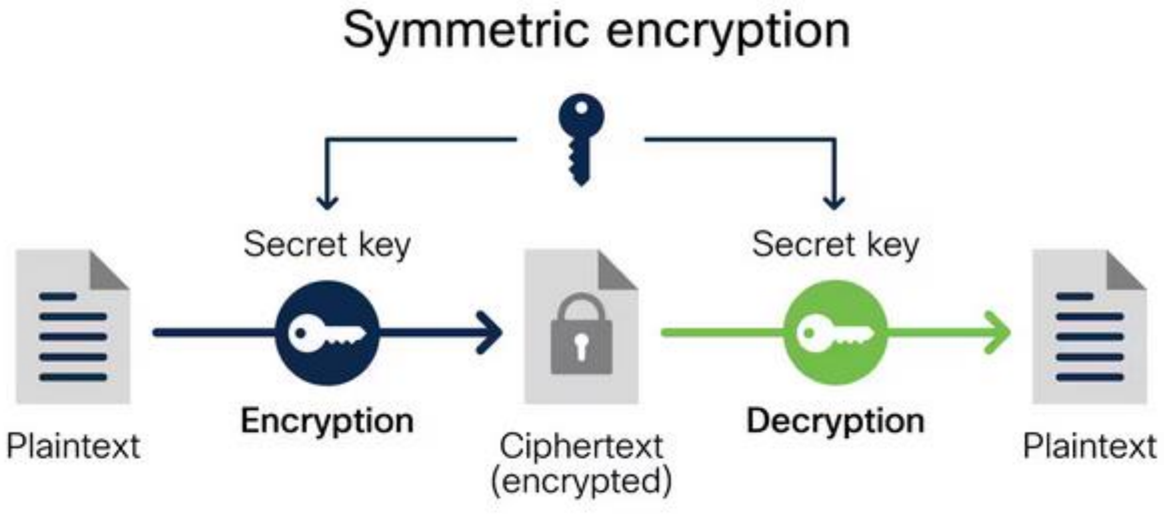
Prof. Dr. Reiner Hüttl, FH Rosenheim, © 2023, 15.03.23

What do we want to learn?

- What are the types of encryption?`
- What are the most important algorithms and procedures?
- How can I use encryption in practice?
- What do I have to pay attention to so that encryption really brings more security?

Symmetric encryption

- One central key for encryption and decryption
- Disadvantages: exchange of secret key required
- Advantages:
  - Very fast,
  - implementable in HW



Quelle: https://www.cisco.com/c/en/us/products/security/encryption-explained.html#~q-a

Symmetric Cipher Variants

- **Stream ciphers**
  - Take a key of fixed length
  - Generate a data stream of arbitrary length, consisting of pseudo-random numbers, from the key
  - Convert each bit of the plaintext to ciphertext using XOR
  - Examples: RC4 (UNSECURE since 2013!!!), ChaCha20

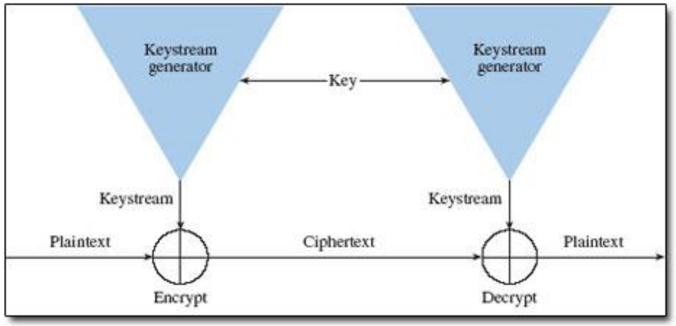
- **Block ciphers**
  - Process plaintext and ciphertext in blocks (e.g. 64 Bit, 128 Bit)
  - Have a mode (block cipher mode)
  - Examples: DES, DES3 (UNSECURE!!), AES, Twofish, Serpent

- **Padding**
  - Filling up the last incomplete block with "regular" patterns.
  - Reason: Some algorithms (mostly block ciphers) require full blocks
  - Examples: PKCS5 padding, W3C padding, ISO padding, ESP padding.

irregular but reconstructable

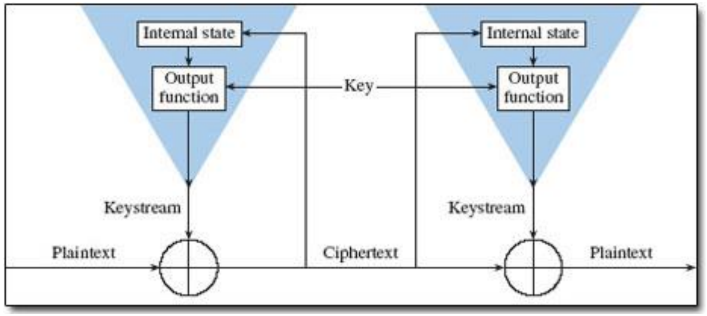
Stream cipher variants

- Synchronous stream ciphering
  - Generates the key stream independently of the clear text or key text



From Schneier, *1996, Figure 9.6*

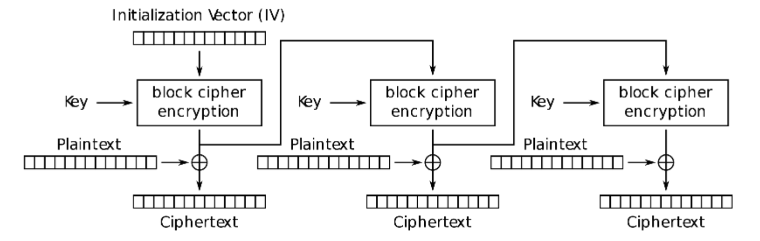
- Self-synchronizing stream ciphering
  - Key stream depends on previous encrypted bit



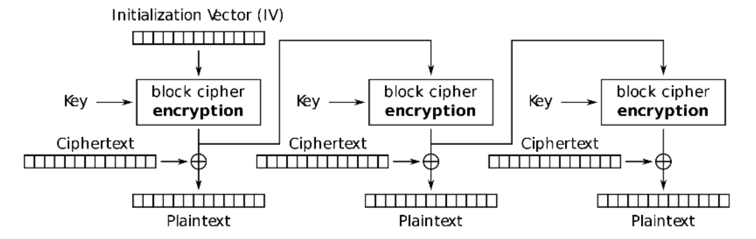
From Schneier, *1996, Figure 9.8*

Block ciphers can be implemented as stream ciphers

- **OFB, Output Feedback Mode**
- Is a synchronous stream ciphering mode
- The key stream can be precalculated



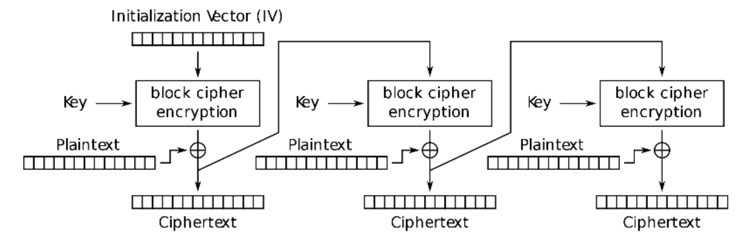
Output Feedback (OFB) mode encryption



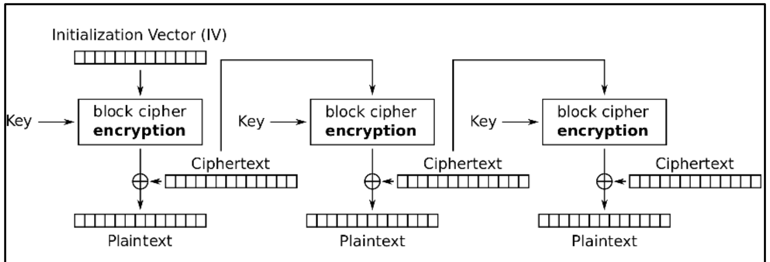
Output Feedback (OFB) mode decryption

https://de.wikipedia.org/wiki/Output_Feedback_Mode

- **CFB Cipher Feedback Mode**
- Self-synchronizing stream ciphering
- Errors in IV or ciphertext affect only two blocks



Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption

https://de.wikipedia.org/wiki/Cipher_Feedback_Mode

The standard encryption method Advanced Encryption Standard (AES)

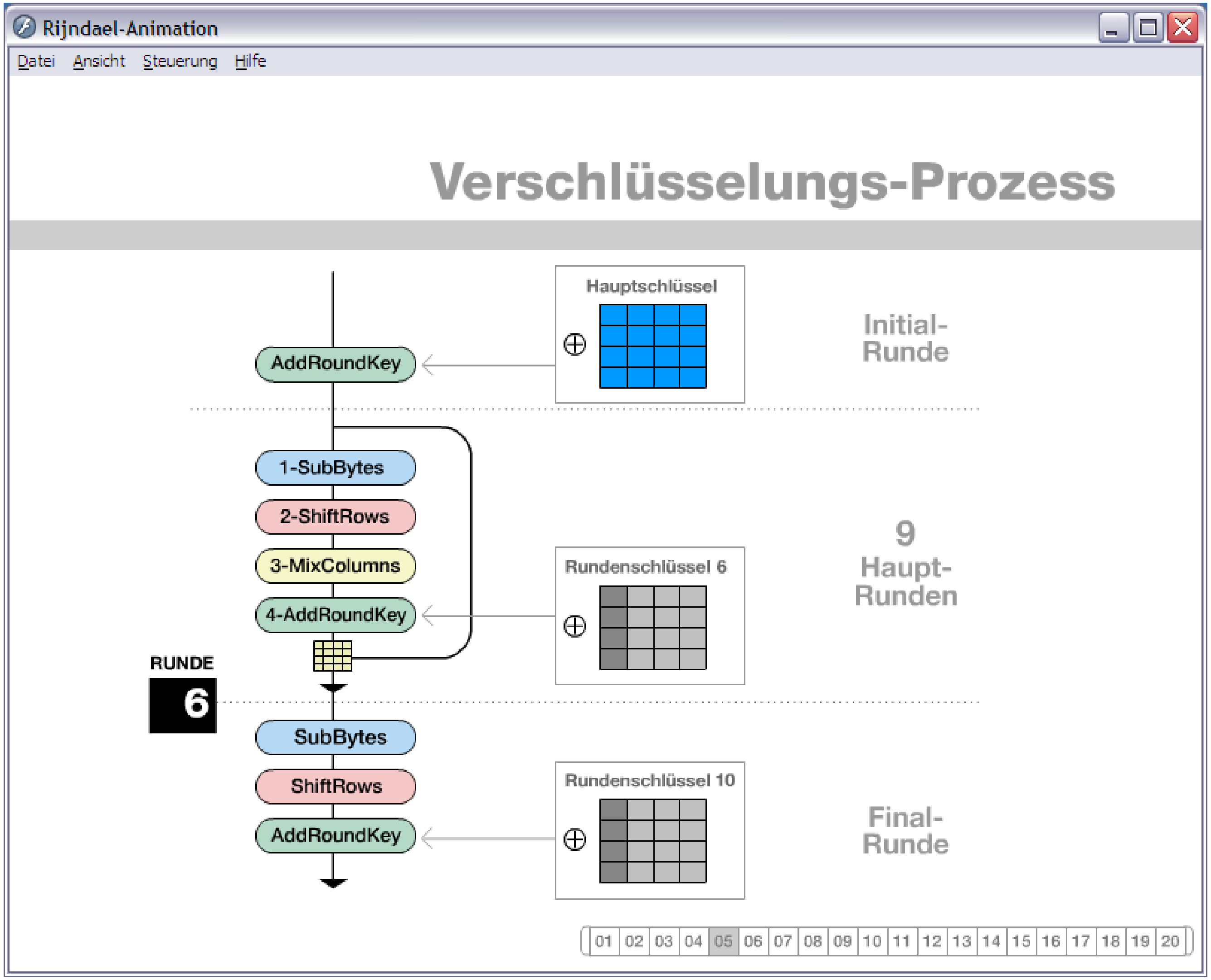
- Requirements for AES in competition from NIST 2001
  - Block cipher with 128, 192 and 256 bits
  - Mathematical justification of security
  - Simplicity of design
  - Flexibility in block sizes and key lengths
  - Efficiency
  - Easy to implement in HW and SW

Old standard, unsafe today, do not use anymore!!!

- Official successor to DES
  - Based on Rijndael algorithm
  - Selected from 15 proposals in an open process
  - All design criteria published
  - Encrypts blocks with fixed length of 128 bits (16 bytes)
  - Key length optionally 128, 192 or 256 bits

recommended key length

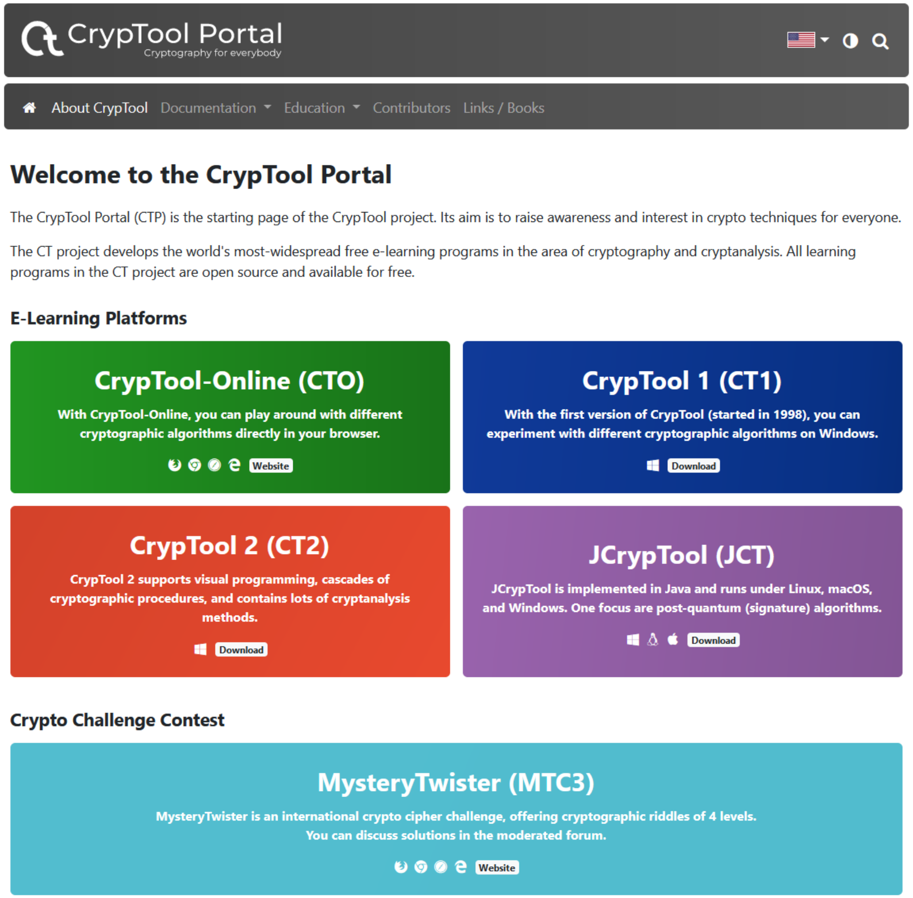
The Algorithm of AES



Quelle: Cryptool Portal,
https://www.cryptool.org/en/cto/aes-animation

Small exercise: View simulation of AES

- Go to CryptTool Portal https://www.cryptool.org/en/
- The CT project develops free e-learning programs in the area of cryptography and cryptanalysis
- Prof. Bernhard Esslinger,
  University of Siegen
- Go to CrypTool-Online (CTO)
  Go to AES Animation
  https://www.cryptool.org/en/cto/aes-animation
- Go through the animation step by step



Welcome to the CrypTool Portal

The CrypTool Portal (CTP) is the starting page of the CrypTool project. Its aim is to raise awareness and interest in crypto techniques for everyone.

The CT project develops the world's most-widespread free e-learning programs in the area of cryptography and cryptanalysis. All learning programs in the CT project are open source and available for free.

E-Learning Platforms

CrypTool-Online (CTO)
With CrypTool-Online, you can play around with different cryptographic algorithms directly in your browser.
Website

CrypTool 1 (CT1)
With the first version of CrypTool (started in 1998), you can experiment with different cryptographic algorithms on Windows.
Download

CrypTool 2 (CT2)
CrypTool 2 supports visual programming, cascades of cryptographic procedures, and contains lots of cryptanalysis methods.
Download

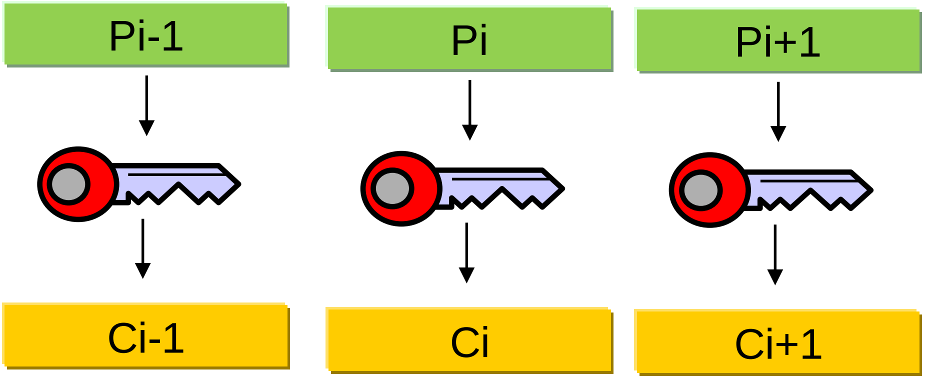
JCrypTool (JCT)
JCrypTool is implemented in Java and runs under Linux, macOS, and Windows. One focus are post-quantum (signature) algorithms.
Download

Crypto Challenge Contest

MysteryTwister (MTC3)
MysteryTwister is an international crypto cipher challenge, offering cryptographic riddles of 4 levels.
You can discuss solutions in the moderated forum.
Website

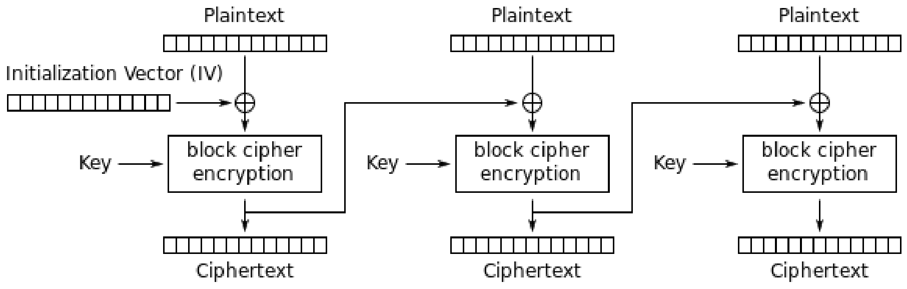
Operating mode for block cipher: ECB

- A symmetric method (e.g., AES) can be operated in different modes
- **Electronic Code Book (ECB)**
  - A plaintext block is encoded into a ciphertext block
  - Same plaintext blocks generate same ciphertext blocks
  - All blocks can be ciphered independently of each other
  - Vulnerable for cryptanalysis (stereotyped beginnings and endings)
  - Simple bit errors have no influence on other blocks
  - If bits are lost, resynchronization of block boundaries is required

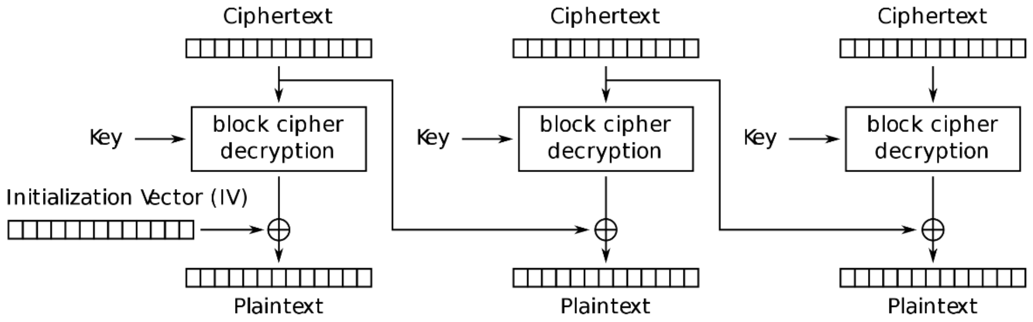


Operating mode for block cipher: CBC

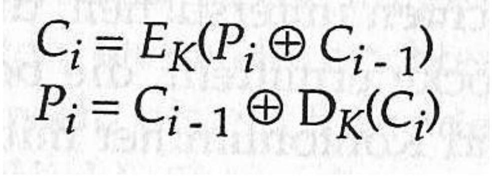
- **Cipher Block Chaining (CBC)**
  - Feedback: encryption of a block depends on predecessor blocks
  - Plaintext is XORed with previous ciphertext block before encryption
  - **Initialization vector** for first block necessary:
    not secret, uniquely reconstruct able for decryption, different for each message
  - Problem: error propagation
    Bit error: 1-bit error in ciphertext leads to error in block and following block
    Synchronization error: no recovery anymore



Cipher Block Chaining (CBC) mode encryption



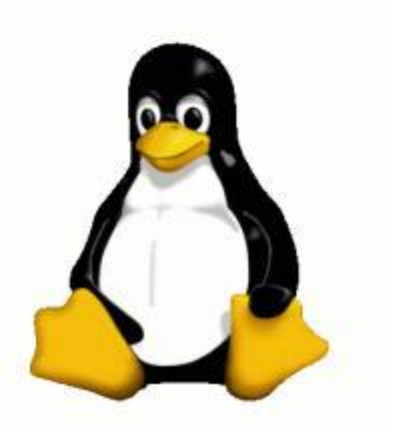
Cipher Block Chaining (CBC) mode decryption

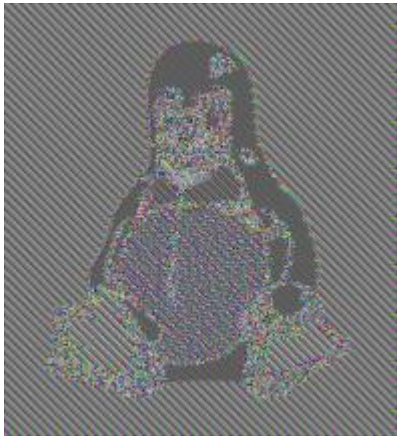


$$C_i = E_K(P_i \oplus C_{i-1})$$
$$P_i = C_{i-1} \oplus D_K(C_i)$$

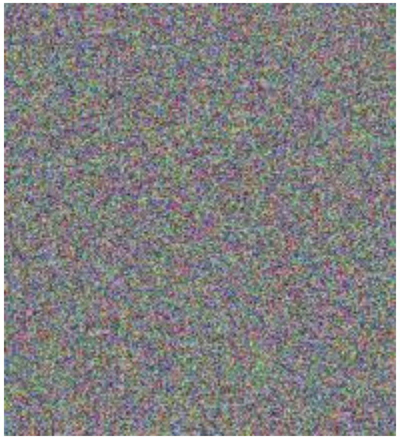
ECB vs. CBC

- Popular error with encryption: Use wrong mode
- **ECB is bad**, CBC is better
- But much better is GCM (comes later)





ECB



CBC

Attention when choosing the initialization vector

- Initialization vector
  - Does not have to be kept secret (just send it along with the ciphertext)
  - Should be random (e.g. SecureRandom)
  - Should be different for each message
  - Is the same size as the block size, for AES **always** 128 bit

- Popular errors
  - Use Key as IV
    - This may cause the key to be read in the worst case [1]
  - Set IV to 0 or other fixed value
    - IV brings randomness into the system - therefore roll dice randomly!
  - Generate IV from password
    - Better chosse IV randomly

[1] https://crypto.stackexchange.com/questions/16161/problems-with-using-aes-key-as-iv-in-cbc-mode

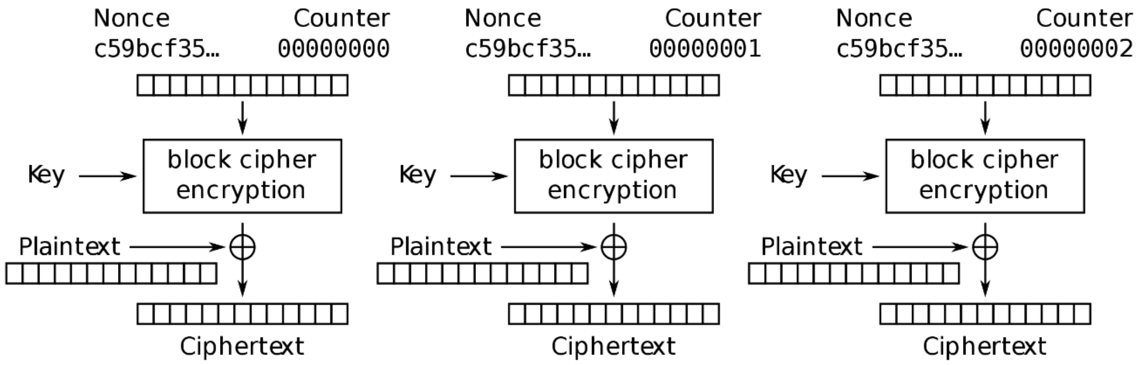
Operating mode CTR Counter Block Mode

- For each block a new unpredictable keystream block is calculated:
  Keystream block = IV(Nonce) + current counter + encryption key

Nonce = Number used only once

- Advantages:
  - Key does not depend on the key of the previous block
  - Encryption and decryption can be performed in parallel
  - Key stream can be precomputed (when operating as a stream cipher)
  - Bit errors affect only one block

$$C_i = P_i \oplus E_K(ctr_i)$$



Counter (CTR) mode encryption

Application of cryptography is difficult

- I use AES/CBC, all good? **NO!**
  - Attacker can modify ciphertext and thus influence plaintext
  - Attacker can perform **Padding Oracle Attack** and read plaintext, see https://www.arxumpathsecurity.de/blog/2019/10/16/cbc-mode-is-malleable-dont-trust-it-for-authentication

- Other popular error
  - Never use CBC, CTR, etc. without authentication!
  - Otherwise attacker can read / modify plaintext

- That‘s why: Secure encryption with authentication
  - MAC Message Authentication Code
  - MAC = Hash function using a secret key

Authentication
Goal of secure identification of a person or a machine

MAC is discussed in more detail in the Digital Signatures chapter

Authenticated Encryption

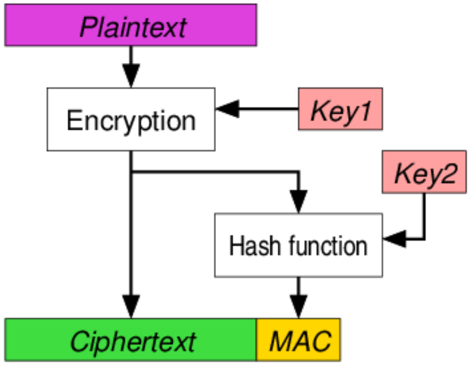
- Encryption methods are called authenticated if not only the confidentiality but also the integrity of the data to be encrypted is protected.

- Encrypt-then-MAC
  - MAC on ciphertext
  - **High Security** if both keys are different

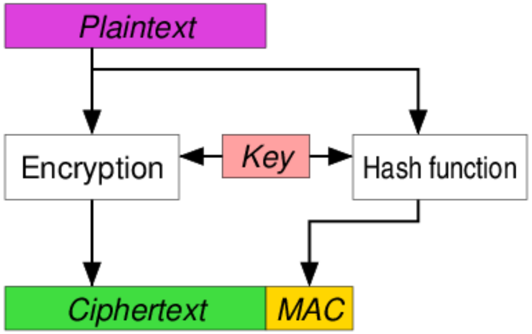
- Encrypt-and-MAC
  - MAC on plaintext
  - Plaintext is encrypted without MAC

- MAC-then-Encrypt
  - MAC on plaintext
  - MAC and plaintext are encrypted

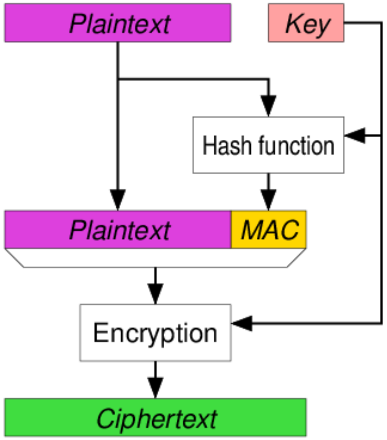
EtM



E&M



MtE



Quelle: https://en.wikipedia.org/wiki/Authenticated_encryption

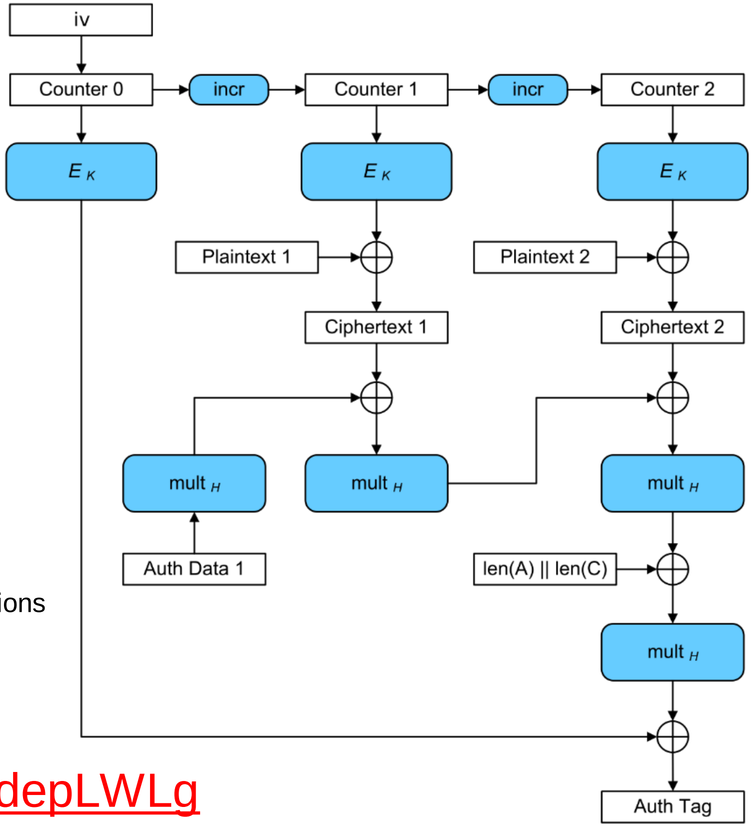
Authenticated Encryption

- Encrypt-then-MAC
  - AES+CBC+HMAC-SHA256
    https://www.mkammerer.de/blog/encrypt-something-with-aes-how-hard-can-it-be/
  - ChaCha20 (encrypt) + Poly1305 (MAC)
    https://github.com/phxql/chacha20-poly1305-java

- Popular erros
  - Use same key for AES and HMAC
    - Is ok in certain cases, if possible: don't do it
  - MAC-then-encrypt
    - AES(HMAC(Plaintext) || Plaintext)
    - There are padding attacks (e.g. with SSL)
  - MAC-and-encrypt
    - AES(plaintext) || HMAC(plaintext)
    - No integrity on ciphertext

Operating mode GCM Galois Counter Mode

- Authenticated encryption mode with associated data (AEAD)
  - Fast, as parallelization is possible
  - Can also be used as stand-alone MAC
  - Accepts IV (nonce) of any length
  - Auth Tag contains Auth Data, IV and ciphertext



Ek Encryption with Key k

Len (a) = len (Auth Data)

Len (C) = len (ciphertext)

Mult H Galois field multiplications

https://de.qwe.wiki/wiki/Galois/Counter_Mode

- Video illustrating GCM
  https://www.youtube.com/watch?v=R2SodepLWLg

AEAD Authenticated Encryption with Associated Data

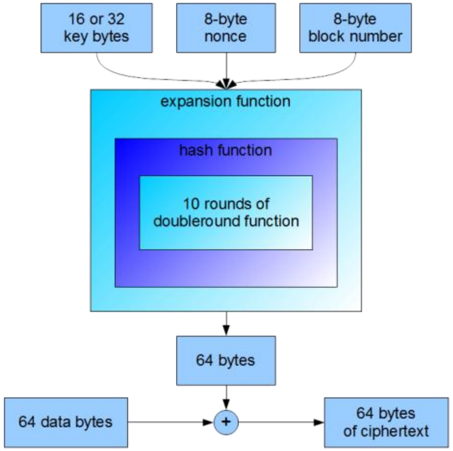
- AEAD combines confidentiality, authenticity and integrity
  - is automatically secure against ciphertext modification
  - Combine encryption and MAC into one integrated protocol
  - e.g. AES-GCM
    - Uses a 96 bit nonce
    - Nonce (IV) need not be random, but must NEVER be reused with same key
  - Further variant AES-EAX

- Video illustrating AEAD
  https://www.youtube.com/watch?v=od44W45sCQ4

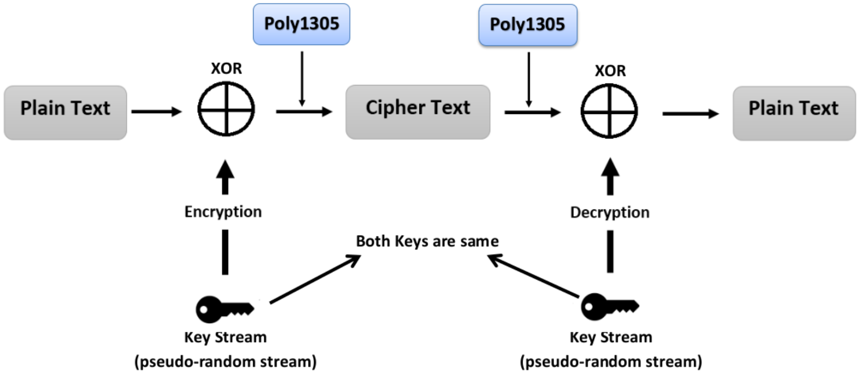
AEAD with stream cipher ChaCha20 and checksum Poly1305

- ChaCha20
  - Symmetric stream cipher based on Salsa20 by Daniel Bernstein
  - Implementation easier than AES-GCM
  - Used in IPSec, TLS, OpenSSL, OpenSSH

- Poly1305 is much faster as MAC than HMAC



Block Diagram of Salsa20 Algorithm
http://www.crypto-it.net/eng/symmetric/salsa20.html



ChaCha20 Poly1305 Encryption and Decryption scheme
https://javainterviewpoint.com/chacha20-poly1305-encryption-and-decryption/

Base64 Encoding

Base64 is an encoding and not an encryption!

- For transmission of binary data in sequences of 8-bit bytes across channels that only reliable support text content, e.g. the World Wide Web.
- Representation of binary data with 64 printable ASCII characters.
- Principle:
  - Split 24 bits into 4 parts of 6 bits
  - Extend each 6 bit sequence to 8 bits
- Disadvantage: message becomes 33% longer
- Application:
  - Keys, signatures and certificates are often stored or transmitted BASE64 encoded.
  - E-Mail Attachments

Example for Base64 Encoding

Base64 Alphabet

Wert	Zeichen	Wert	Zeichen	Wert	Zeichen	Wert	Zeichen
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v	64	=
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

Beispiel abcde			
Quelle	**Binärdarstellung**	**Base64**	
a	01100001		
b	01100010		
c	01100011		
24-Bit	011000010110001001100011		
6-Bit	011000	24	Y
	010110	22	W
	001001	9	J
	100011	35	j
d	01100100		
e	01100101		
	________		
24-Bit	011001000110010100______		
6-Bit	011001	25	Z
	000110	6	G
	010100	20	U
	______		=
YWJjZGU=			