Cryptographic Systems in Operation

Software Security

Steffen Helke
Chair of Software Engineering

19th December 2018

Brandenburgische
Technische Universität
Cottbus - Senftenberg

Objectives of today's exercise

➜ Understanding the concepts *stream cipher* and *block cipher*

➜ Getting to know key criteria for evaluating operation modes such as *self-synchronization*, *error expansion* and *random access*

➜ Being able to describe fundamental modes of operation, e.g. *ECB*, *CBC*, *CFB*, *OFB* and *CTR*

Basic Terms

Block Cipher

- Encryption and decryption of strings with a *fixed* length

Stream Cipher

- Encryption and decryption of strings with a *variable* length
- Messages are encoded as a sequence of characters

Remarks

- Distinction is not always clear and depends on the used alphabet

  ➜ **Example**

  - for the alphabet $\{0, 1\}$ DES is a block cipher
  - for the alphabet $\{0, 1, 2, \ldots, 2^{64} - 1\}$ DES is a stream cipher

Deterministic vs. Indeterministic Cipher

Deterministic Block Cipher

- Ciphertext blocks and plaintext blocks have the *same length*
- Identical input parameters always result in the *same ciphertext*

Indeterministic Block Cipher

- Ciphertext blocks will be *longer* than the plaintext blocks
- Identical input parameters result in *different ciphertexts*

Synchronous vs. Self-Synchronizing Cipher

Synchronous Stream Cipher

- Encryption of a character depends on *all preceding characters*

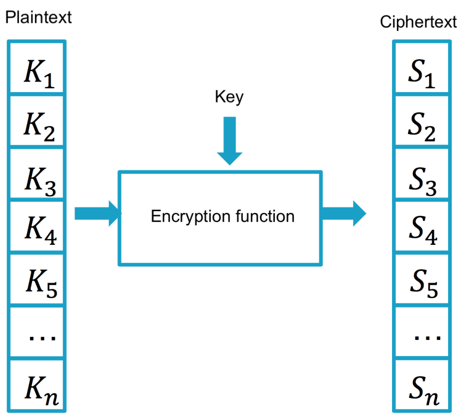
Self-Synchronizing Stream Cipher

- Encryption of a character depends only on a *certain number of preceding characters*
- Regular re-synchronization mechanism is implemented

Classification of Operation Modes

- Operation mode of the block cipher determines how the encryption of (long) plain texts is implemented
- Classification
  1. Block cipher is used to *encrypt a message directly*
  2. Block cipher is used to *generate a pseudo-random key stream*, which is used to encrypt the message by a stream cipher
- The basic concept of an operation mode is usually independent of the used block cipher (encryption system)

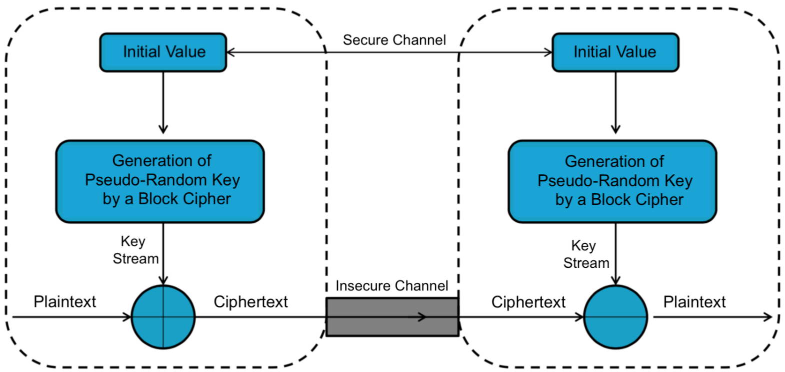
How to encrypt a message by a block cipher directly?



➜ The last block will be filled with *padding*

➜ We assume that the plaintext is fully available at the beginning of encryption

How to encrypt a message by a block cipher indirectly?



➜ Plaintext does not have to be fully available at the beginning of encryption

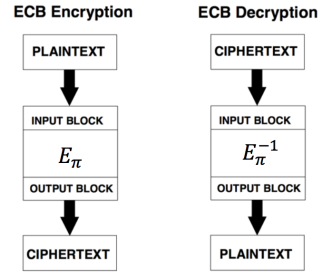
➜ An arbitrarily small block size for the encryption is possible

Operation Mode

– Electronic Codebook (ECB) –

ECB – Electronic Codebook

- Procedure for block ciphers, very naively implemented
- Splitting the plaintext into blocks of equal size
- Filling of the last block (*Padding*) is usually necessary
- Encryption and decryption of a block is *completely independent of other blocks*



Quelle: NIST

Encryption

$c_j = E_\pi(m_j)$ for $j = 1..n$

Decryption

$m_j = E_\pi^{-1}(c_j)$ for $j = 1..n$

ECB – Example

Assumption

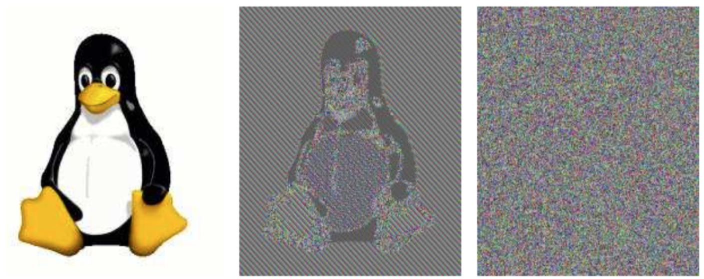
- Block size: $b = 4$
- Plaintext: $m = 101100010100101$ (15 Bits)
- Encryption function: here $E_\pi \mathrel{\widehat{=}}$ Permutation on bits
  ➜ in practice a modern encryption system is used, e.g. AES
- Key: $(1 \rightarrow 4),(2 \rightarrow 1),(3 \rightarrow 2),(4 \rightarrow 3)$ – just a *left shift*

Calculation

1. Splitting into blocks, including *Padding*
   $m_1 = 1011$, $m_2 = 0001$, $m_3 = 0100$, $m_4 = 1010$
2. Encryption
   $c_1 = E_\pi(1011) = 0111$, $c_2 = E_\pi(0001) = 0010$, ...
   ➜ $c = 0111\ 0010\ 1000\ 0101$

Disadvantages of the ECB Mode

➜ Same plaintext blocks are represented by the same ciphertext blocks, i.e. regularities in plaintext are also recognizable in the ciphertext



Original — Encrypted by ECB mode — Encrypted by another operation mode

➜ Note, the Bundestrojaner studied by the Chaos Computer Club in 2011 was implemented by AES & ECB

Evaluation of the ECB Mode

Advantages

+ Encryption and decryption can be parallelized
+ Errors in one block do not affect other blocks
+ Indeterministic block ciphers can be used

Disadvantages

- Regularities of the plaintext are recognizable in the ciphertext
  ➜ It is not recommended to use ECB

Neutral

+/- Self-synchronizing operation mode

Operation Mode

– Cipher Block Chaining (CBC) –

CBC – Cipher Block Chaining

- Split the plaintext into n blocks of the same size
- Use XOR to connect a plaintext block and the corresponding ciphertext block of the previous step and apply the encryption function E_π to the result.
- Use for the first plaintext block the initialization vector IV

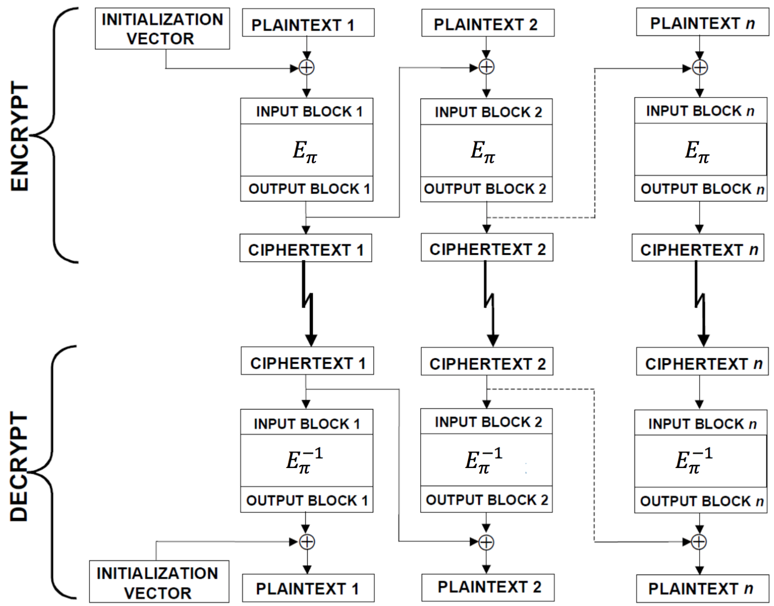
Encryption

$$c_1 = E_\pi(m_1 \oplus IV)$$
$$c_j = E_\pi(m_j \oplus c_{j-1}) \text{ for } j = 2 \,..\, n$$

Decryption

$$m_1 = E_\pi^{-1}(c_1) \oplus IV$$
$$m_j = E_\pi^{-1}(c_j) \oplus c_{j-1} \text{ for } j = 2 \,..\, n$$

How does the CBC operation mode work?



CBC – Example (Encryption)

Assumption

- Plaintext blocks:
  $m_1 = 1011, m_2 = 0001,\ m_3 = 0100,\ m_4 = 1010$
- Encryption function: $E_\pi \;\widehat{=}$ Permutation of Bits
- Key: $(1 \to 4),(2 \to 1),(3 \to 2),(4 \to 3)$ – just a *left shift*
- Initialization vector: $IV = 1010$

Calculation

- Steps

$$\begin{aligned} c_1 &= E_\pi(m_1 \oplus IV) = E_\pi(0001) = 0010 \\ c_2 &= E_\pi(m_2 \oplus c_1) = E_\pi(0011) = 0110 \\ c_3 &= E_\pi(m_3 \oplus c_2) = E_\pi(0010) = 0100 \\ c_4 &= E_\pi(m_4 \oplus c_3) = E_\pi(1110) = 1101 \end{aligned}$$

- Result

$$c = 0010\ 0110\ 0100\ 1101$$

CBC – Example (Decryption)

Assumption

- Ciphertext blocks:
  $c_1 = 0010,\ c_2 = 0110,\ c_3 = 0110,\ c_4 = 1101$
- Encryption function: $E_\pi \;\widehat{=}$ Permutation of Bits
- Key: $(1 \to 4),(2 \to 1),(3 \to 2),(4 \to 3)$ – just a *left shift*
- Initialization vector: $IV = 1010$

Calculation

- Steps

$$\begin{aligned} m_1 &= E_\pi^{-1}(c_1) \oplus IV = 0001 \oplus 1010 = 1011 \\ m_2 &= E_\pi^{-1}(c_2) \oplus c_1 = 0011 \oplus 0010 = 0001 \\ m_3 &= E_\pi^{-1}(c_3) \oplus c_2 = 0010 \oplus 0110 = 0100 \\ m_4 &= E_\pi^{-1}(c_4) \oplus c_3 = 1110 \oplus 0100 = 1010 \end{aligned}$$

- Result

$$m = 1011\ 0001\ 0100\ 1010$$

CBC – Error Propagation

Assumption

- Ciphertext block with *transmission error*
  $c_1 = 0011,\ c_2 = 0110,\ c_3 = 0110,\ c_4 = 1101$
- Encryption function: $E_\pi \;\widehat{=}$ Permutation of Bits
- Key: $(1 \to 4),(2 \to 1),(3 \to 2),(4 \to 3)$ – just a *left shift*
- Initialization vector: $IV = 1010$

Decryption

- Steps

$$\begin{aligned} m_1 &= E_\pi^{-1}(c_1) \oplus IV = 1001 \oplus 1010 = 0011 \\ m_2 &= E_\pi^{-1}(c_2) \oplus c_1 = 0011 \oplus 0011 = 0000 \\ m_3 &= E_\pi^{-1}(c_3) \oplus c_2 = 0010 \oplus 0110 = 0100 \\ m_4 &= E_\pi^{-1}(c_4) \oplus c_3 = 1110 \oplus 0100 = 1010 \end{aligned}$$

- Result

$$m = 0011\ 0000\ 0100\ 1010$$

Evaluation of CBC

Advantages

\+ Decryption can be parallelized

\+ Regularities of the plaintext will be not recognizable in the ciphertext

\+ Indeterministic block ciphers can be used

Disadvantages

\- Encryption can't be parallelized

\- Error of a block effects the successor block

Neutral

+/- Self-synchronizing operation mode

Operation Mode
– Cipher Feedback (CFB) –

CFB – Cipher Feedback

General Procedure

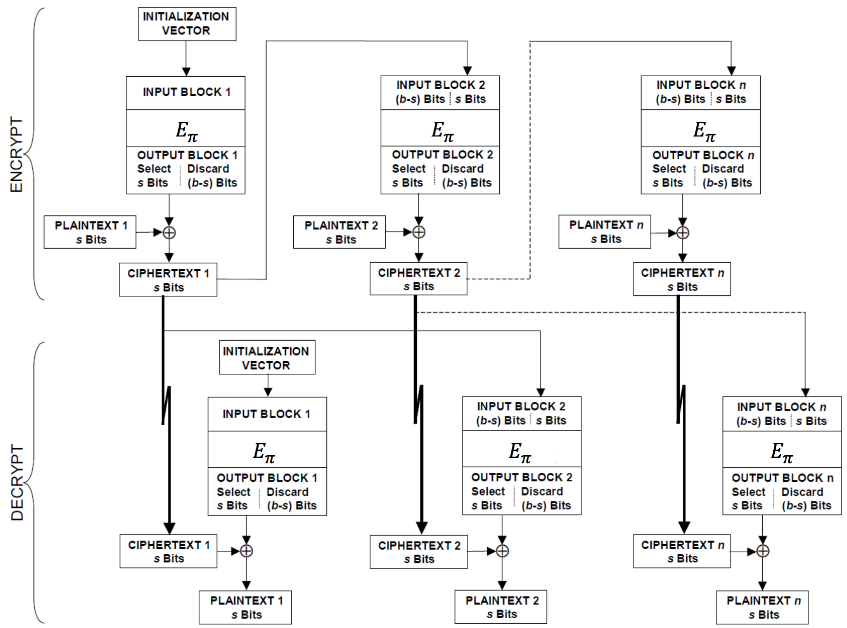
- Length s of the unit to be encrypted can be shorter than the block length b of the used encryption system
- The ciphertext of the previous block is encrypted by E_π and the result and the plaintext block will be connected by XOR
- The first plaintext block is encrypted with the initialization vector IV, which (at least) should have the length b

Calculation

➜ Set $I_1 = IV$ and perform the following steps for each j with $1 \leq j \leq n$

1. Set $O_j = E_\pi(I_j)$
2. Set t_j to the first s bits of O_j
3. Set $c_j = m_j \oplus t_j$ for encryption
   Set $m_j = c_j \oplus t_j$ for decryption
4. Set I_{j+1} to the last b bits of the concatenation of I_j and c_j

How does the CFB operation mode work?



CFB – Example (Encryption)

Assumption

- $s = 3$ and $b = 4$
- Plaintext blocks: $m_1 = 101$, $m_2 = 100$, $m_3 = 010$
- Encryption function: $E_\pi \mathrel{\widehat{=}}$ Permutation of Bits
- Key: $(1 \to 4),(2 \to 1),(3 \to 2),(4 \to 3)$ – just a *left shift*
- Initialization vector: $IV = 1010$

Calculation

j	I_j	O_j	t_j	m_j	c_j
1	1010	0101	010	101	111
2	0111	1110	111	100	011
3	1011	0111	011	010	001

CFB – Example (Decryption)

Assumption

- $s = 3$ and $b = 4$
- Ciphertext blocks: $c_1 = 111$, $c_2 = 011$, $c_3 = 001$
- Encryption function: $E_\pi \,\widehat{=}\,$ Permutation of Bits
- Key: $(1 \to 4),(2 \to 1),(3 \to 2),(4 \to 3)$ – just a *left shift*
- Initialization vector: $IV = 1010$

Calculation

j	I_j	O_j	t_j	c_j	m_j
1	1010	0101	010	111	101
2	0111	1110	111	011	100
3	1011	0111	011	001	010

CFB – Error Propagation (Decryption)

Assumption

- $s = 3$ and $b = 4$
- Ciphertext blocks with *transmission error*
  $c_1 = 110$, $c_2 = 011$, $c_3 = 001$
- Encryption function: $E_\pi \,\widehat{=}\,$ Permutation of Bits
- Key: $(1 \to 4),(2 \to 1),(3 \to 2),(4 \to 3)$ – just a *left shift*
- Initialization vector: $IV = 1010$

Calculation

j	I_j	O_j	t_j	c_j	m_j
1	1010	0101	010	110	100
2	0110	1100	110	011	101
3	0011	0110	011	001	010

Evaluation of CFB

Advantages

+ Decryption can be parallelized
+ Regularities of the plaintext will be not recognizable in the ciphertext
+ The length of the encryption units can be smaller than the block size of the used encryption system

Disadvantages

- Encryption can't be parallelized
- Error of a block effects $1 + \frac{b}{s}$ successor blocks
- Indeterministic block ciphers are not supported

Neutral

+/- Self-synchronizing operation mode

Operation Mode

– Output Feedback (OFB) –

OFB – Output Feedback

General Procedure

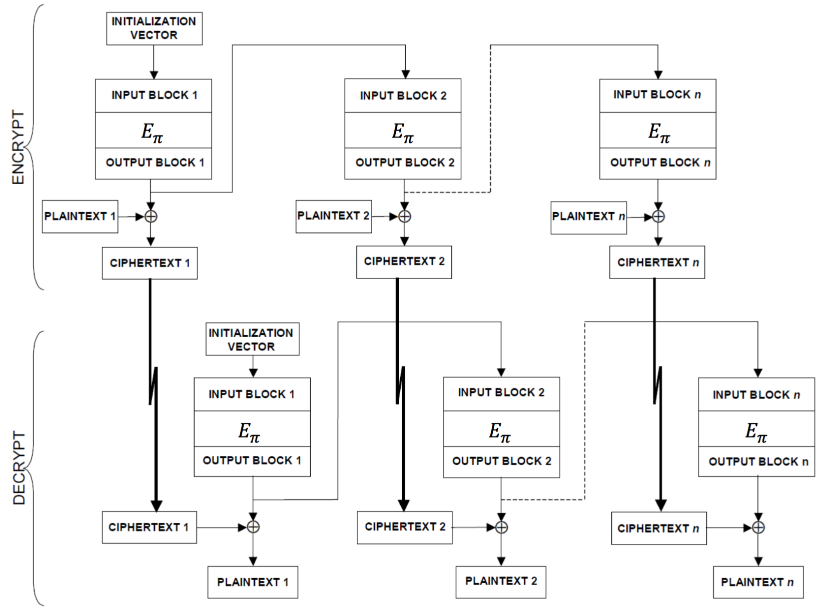
- Length s of the unit to be encrypted can be shorter than the block length b of the used encryption system
- The generated pseudo-random key stream depends only on the IV and the used main key
- Both encryption and decryption depend only on the position, not on the previous message

Calculation

➜ Set $I_1 = IV$ and perform the following steps for all j with $1 \leq j \leq n$

1. Set $O_j = E_\pi(I_j)$
2. Set t_j to the first s bits of O_j
3. Set $c_j = m_j \oplus t_j$ for encryption,
   Set $m_j = c_j \oplus t_j$ for decryption
4. Set $I_{j+1} = O_j$

How does the OFB operation mode work?



OFB – Example (Encryption)

Assumption

- $s = 3$ and $b = 4$
- Plaintext blocks: $m_1 = 101$, $m_2 = 100$, $m_3 = 010$
- Encryption function: $E_\pi \mathrel{\widehat{=}}$ Permutation of Bits
- Key: $(1 \rightarrow 4),(2 \rightarrow 1),(3 \rightarrow 2),(4 \rightarrow 3)$ – just a *left shift*
- Initialization vector: $IV = 1010$

Calculation

j	I_j	O_j	t_j	m_j	c_j
1	1010	0101	010	101	111
2	0101	1010	101	100	001
3	1010	0101	010	010	000

OFB – Example (Decryption)

Assumption

- $s = 3$ and $b = 4$
- Ciphertext blocks: $c_1 = 111$, $c_2 = 001$, $c_3 = 000$
- Encryption function: $E_\pi \mathrel{\widehat{=}}$ Permutation of Bits
- Key: $(1 \rightarrow 4),(2 \rightarrow 1),(3 \rightarrow 2),(4 \rightarrow 3)$ – just a *left shift*
- Initialization vector: $IV = 1010$

Calculation

j	I_j	O_j	t_j	c_j	m_j
1	1010	0101	010	111	101
2	0101	1010	101	001	100
3	1010	0101	010	000	010

Evaluation of OFB

Advantages

+ Regularities of the plaintext will be not recognizable in the ciphertext
+ The length of the encryption units can be smaller than the block size of the used encryption system

Disadvantages

- Encryption and decryption cannot be parallelized, but the key stream can be calculated in advance
- Result is a symmetric stream cipher, independent of the encryption system used, which could be asymmetric
- Indeterministic block ciphers are not supported

Neutral

+/- Synchronous operation mode

Operation Mode

– Counter Mode (CTR) –

CTR – Counter Mode

General Procedure

- Input of the key function is a counter value $T_1, T_2 \ldots T_n$, whereby T_j is unique
- Input is combined with a nonce for a single block

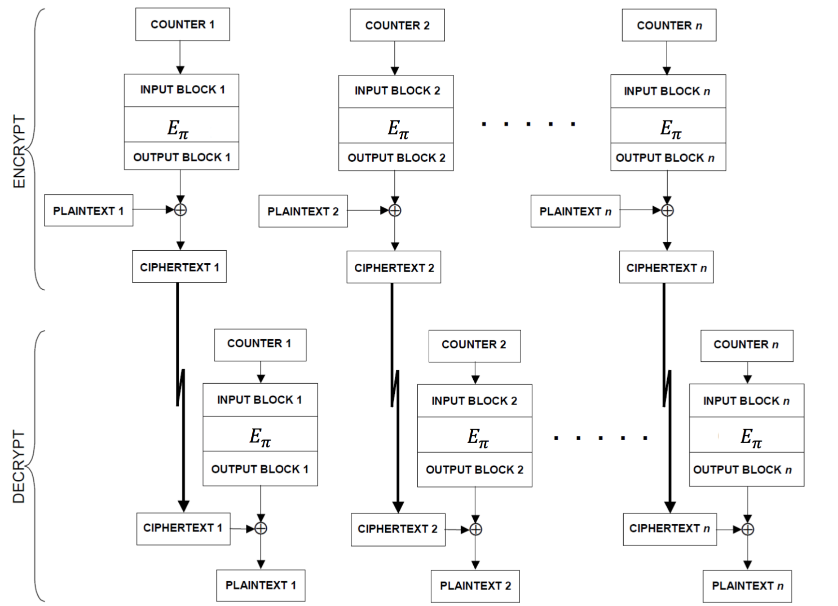
Encryption

1. Set $O_j = E_\pi(T_j)$ for $j = 1, 2, \ldots n$
2. Set $c_j = m_j \oplus O_j$ for $j = 1, 2, \ldots n-1$
3. Set $c_n = m_n \oplus O_n$, whereby for an incomplete block m_n only required bits of O_n will be used

Decryption

1. Set $O_j = E_\pi(T_j)$ for $j = 1, 2, \ldots n$
2. Set $m_j = c_j \oplus O_j$ for $j = 1, 2, \ldots n-1$
3. Set $m_n = c_n \oplus O_n$, whereby for an incomplete block c_n only required bits of O_n will be used

How does the CTR operation mode work?



Evaluation of CTR

Advantages

\+ Encryption and decryption can be parallelized

\+ Regularities of the plaintext will be not recognizable in the ciphertext

\+ The length of the encryption units can be smaller than the block size of the used encryption system

Disadvantages

- Indeterministic block ciphers are not supported
- Counter functions are predictable, therefore it is better to use an extension of CTR, e.g. the GCM (Galois Counter Mode)

Neutral

+/- Synchronous mode

Comparison

	ECB	CBC	CFB	OFB	CTR
usage of inde-terministic block ciphers	+ possible		- impossible		
asymmetrical block cipher produces	- asymmetrical stream cipher		- symmetrical stream cipher		
length of encryptable units	- specified by block length		+ arbitrary		
error expansion	inside one block	2 blocks	≤ 1 + b/s blocks	none with falsification	
synchrony	self-synchronized			synchronous	
random access	yes	only with decryption		no	yes
parallelizeable	yes	only with decryption		no	yes
calculation of the block cipher in advance	no		for one block at a time	yes	