Stream Ciphers and Block Ciphers

Cryptography I – Fall 2013

Ruben Niederhagen

September 19th, 2013

Introduction

Stream Ciphers:

- symmetric-key cipher
- state-driven: operates on arbitrary message length
- commonly used stream ciphers: A5/1 and A5/2 (GSM), RC4 (SSL, WEP), eSTREAM Project

Introduction

Stream Ciphers:

- symmetric-key cipher
- state-driven: operates on arbitrary message length
- commonly used stream ciphers: A5/1 ! and ~~A5/2~~ (GSM), RC4 ? (SSL, WEP), eSTREAM Project

Introduction

Stream Ciphers:

- symmetric-key cipher
- state-driven: operates on arbitrary message length
- commonly used stream ciphers: A5/1 and ~~A5/2~~ (GSM), RC4 (SSL, WEP), eSTREAM Project

Synchronous Stream Ciphers:

Given key K and initial state σ_0:

state:	σ_{i+1}	$=$	$f(\sigma_i, K)$	with next-state function f
key stream:	z_i	$=$	$g(\sigma_i, K)$	with key-stream function g
cipher stream:	c_i	$=$	$h(z_i, m_i)$	with output function h

Introduction

Stream Ciphers:

- symmetric-key cipher
- state-driven: operates on arbitrary message length
- commonly used stream ciphers: A5/1 and ~~A5/2~~ (GSM), RC4 (SSL, WEP), eSTREAM Project

Synchronous Stream Ciphers:

Given key K and initial state σ_0:

state:	σ_{i+1}	$=$	$f(\sigma_i, K)$	with next-state function f
key stream:	z_i	$=$	$g(\sigma_i, K)$	with key-stream function g
cipher stream:	c_i	$=$	$z_i \oplus m_i$	

Introduction

Stream Ciphers:

- symmetric-key cipher
- state-driven: operates on arbitrary message length
- commonly used stream ciphers: A5/1 and ~~A5/2~~ (GSM), RC4 (SSL, WEP), eSTREAM Project

Self-Synchronizing Stream Ciphers:

Given key K and initial states $\sigma_0 \ldots \sigma_t$:

state:	σ_{i+1}	$=$	$(c_i, c_{i-1}, \ldots c_{i-t})$	
key stream:	z_i	$=$	$g(\sigma_i, K)$	with key-stream function g
cipher stream:	c_i	$=$	$h(z_i, m_i)$	with output function h

Introduction

Stream Ciphers:

- symmetric-key cipher
- state-driven: operates on arbitrary message length
- commonly used stream ciphers: A5/1 and ~~A5/2~~ (GSM), RC4 (SSL, WEP), eSTREAM Project

Self-Synchronizing Stream Ciphers:

Given key K and initial states $\sigma_0 \ldots \sigma_t$:

state:	σ_{i+1}	$=$	$(c_i, c_{i-1}, \ldots c_{i-t})$	
key stream:	z_i	$=$	$g(\sigma_i, K)$	with key-stream function g
cipher stream:	c_i	$=$	$z_i \oplus m_i$	

Introduction

Block Ciphers:

- symmetric-key cipher
- memoryless: operates on a fixed-length block size
- commonly used block ciphers: DES, Triple-DES, AES

Introduction

Block Ciphers:

- symmetric-key cipher
- memoryless: operates on a fixed-length block size
- commonly used block ciphers: ~~DES~~, Triple-DES, AES

Introduction

Block Ciphers:

- symmetric-key cipher
- memoryless: operates on a fixed-length block size
- commonly used block ciphers: ~~DES~~, Triple!-DES, AES

An n-bit block cipher is a function $E : \{0,1\}^n \times \mathfrak{K} \rightarrow \{0,1\}^n$.
For each fixed key $K \in \mathfrak{K}$ the map

$$E_K : \{0,1\}^n \rightarrow \{0,1\}^n, M \mapsto E_K(M)$$

is invertible (bijective) with inverse $E_K^{-1} : \{0,1\}^n \rightarrow \{0,1\}^n$.

From Block Ciphers to Stream Ciphers

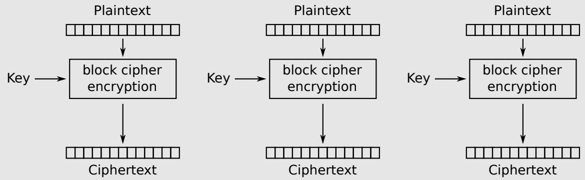
Mode of Operation:

- Electronic codebook (ECB) mode
- Cipher-block chaining (CBC) mode
- Cipher feedback (CFB) mode
- Output feedback (OFB) mode
- Counter (CTR) mode

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Electronic codebook (ECB) mode:
  - Encryption:
    obtain ciphertext $C_1, \ldots, C_t$ as
    $C_i = E_K(M_i), \; i = 1 \ldots t$

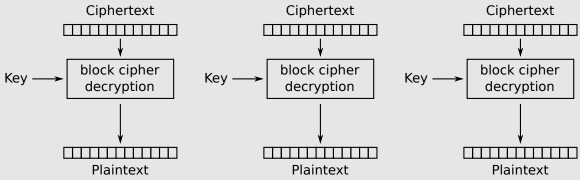


Electronic Codebook (ECB) mode encryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Electronic codebook (ECB) mode:
  - Decryption:
    obtain plaintext $M_1, \ldots, M_t$ as
    $M_i = E_K^{-1}(C_i),\ i = 1 \ldots t$

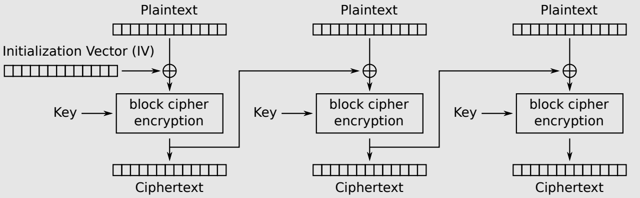


Electronic Codebook (ECB) mode decryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Cipher-block chaining (CBC) mode:
  Use a (non-secret) initialization vector (IV) of length n bits.
  - Encryption:
    obtain ciphertext $C_1, \ldots, C_t$ as
    $C_i = E_K(M_i \oplus C_{i-1}),\ i = 1 \ldots t,\ C_0 = IV$

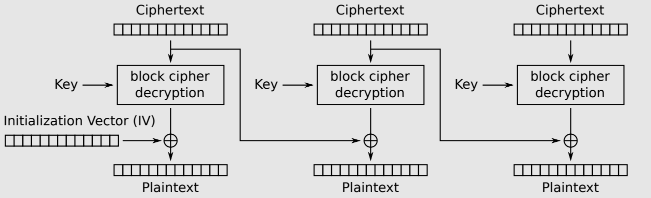


Cipher Block Chaining (CBC) mode encryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Cipher-block chaining (CBC) mode:
  Use a (non-secret) initialization vector (IV) of length n bits.
  - Decryption:
    obtain plaintext $M_1, \ldots, M_t$ as
    $M_i = E_K^{-1}(C_i) \oplus C_{i-1},\ i = 1 \ldots t,\ C_0 = IV$

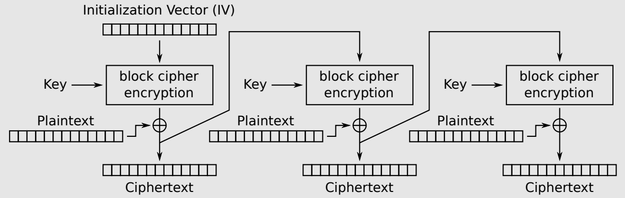


Cipher Block Chaining (CBC) mode decryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Cipher feedback (CFB) mode:
  Use a (non-secret) initialization vector (IV) of length n bits.
  - Encryption:
    obtain ciphertext $C_1, \ldots, C_t$ as
    $C_i = E_K(C_{i-1}) \oplus M_i,\ i = 1 \ldots t,\ C_0 = IV$

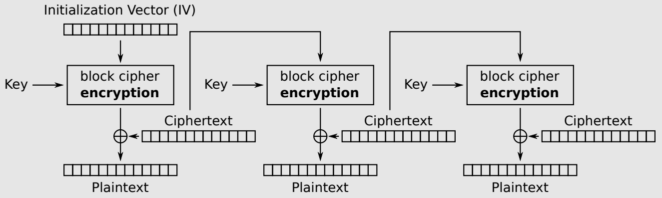


Cipher Feedback (CFB) mode encryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Cipher feedback (CFB) mode:
  Use a (non-secret) initialization vector (IV) of length n bits.
  - Decryption:
    obtain plaintext $M_1, \ldots, M_t$ as
    $M_i = E_K(C_{i-1}) \oplus C_i,\ i = 1 \ldots t,\ C_0 = IV$

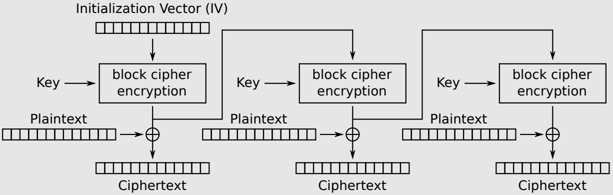


Cipher Feedback (CFB) mode decryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Output feedback (OFB) mode:
  Use a (non-secret) initialization vector (IV) of length n bits.
  - Encryption:
    obtain ciphertext $C_1, \ldots, C_t$ as
    $C_i = O_i \oplus M_i,\ i = 1 \ldots t,$
    $O_i = E_K(O_{i-1}),\ O_0 = IV$

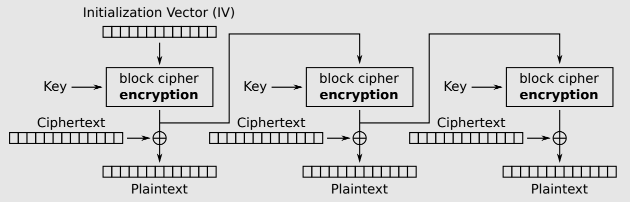


Output Feedback (OFB) mode encryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Output feedback (OFB) mode:
  Use a (non-secret) initialization vector (IV) of length n bits.
  - Decryption:
    obtain plaintext $M_1, \ldots, M_t$ as
    $M_i = O_i \oplus C_i, \; i = 1 \ldots t,$
    $O_i = E_K(O_{i-1}), \; O_0 = IV$

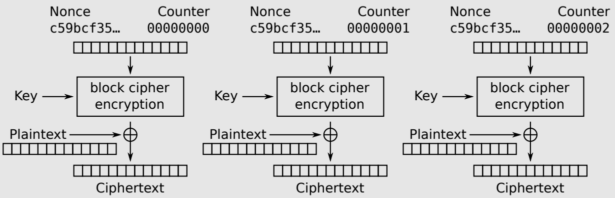


Output Feedback (OFB) mode decryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Counter (CTR) mode:
  Use a (non-secret) initialization vector (IV) of length n bits.
  - Encryption:
    obtain ciphertext $C_1, \ldots, C_t$ as
    $C_i = E_K(N_i) \oplus M_i, \; i = 1 \ldots t,$
    $N_i = N_{i-1} + 1 \mod 2^n, \; N_0 = IV$

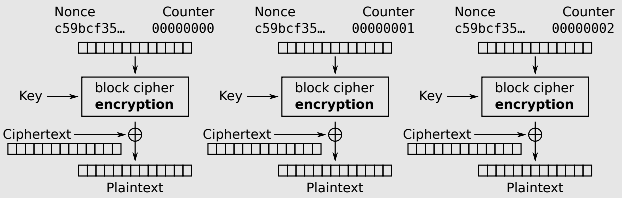


Counter (CTR) mode encryption

From Block Ciphers to Stream Ciphers

Mode of Operation:

- Counter (CTR) mode:
  Use a (non-secret) initialization vector (IV) of length n bits.
  - Decryption:
    obtain plaintext $M_1, \ldots, M_t$ as
    $M_i = E_K(N_i) \oplus C_i,\ i = 1 \ldots t,$
    $N_i = N_{i-1} + 1 \mod 2^n,\ N_0 = IV$



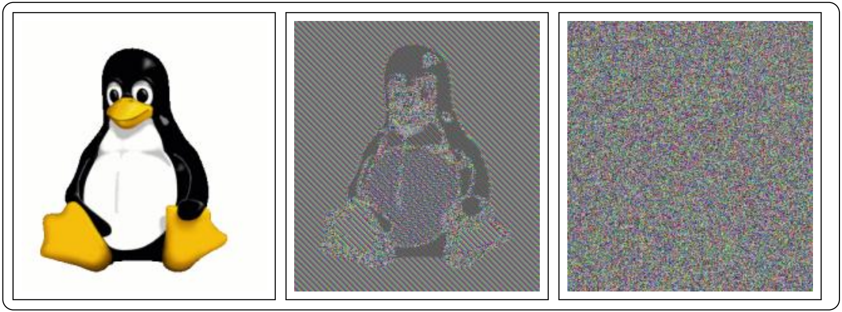
Counter (CTR) mode decryption

Properties of the Block-Cipher Modes of Operation

- ECB is considered insecure if applied to more than one block: Identical input blocks are mapped to identical output blocks.

Properties of the Block-Cipher Modes of Operation

- ECB is considered insecure if applied to more than one block: Identical input blocks are mapped to identical output blocks.



Properties of the Block-Cipher Modes of Operation

- ECB is considered insecure if applied to more than one block: Identical input blocks are mapped to identical output blocks.
- In CBC and CFB mode, the last ciphertext block C_t depends on all message blocks $M_1, \ldots, M_t$, in ECB, OFB, and CTR mode each block of ciphertext C_i only on message block M_i.

Properties of the Block-Cipher Modes of Operation

- ECB is considered insecure if applied to more than one block: Identical input blocks are mapped to identical output blocks.
- In CBC and CFB mode, the last ciphertext block C_t depends on all message blocks $M_1, \ldots, M_t$, in ECB, OFB, and CTR mode each block of ciphertext C_i only on message block M_i.
- CBC, CFB, and OFB encryption can not be performed in parallel on several blocks, ECB and CTR encryption can. CBC and CFB decryption also can be performed in parallel.

Properties of the Block-Cipher Modes of Operation

- ECB is considered insecure if applied to more than one block: Identical input blocks are mapped to identical output blocks.
- In CBC and CFB mode, the last ciphertext block C_t depends on all message blocks $M_1, \dots, M_t$, in ECB, OFB, and CTR mode each block of ciphertext C_i only on message block M_i.
- CBC, CFB, and OFB encryption can not be performed in parallel on several blocks, ECB and CTR encryption can. CBC and CFB decryption also can be performed in parallel.
- Only ECB and CTR allow random access to the ciphertext.

Properties of the Block-Cipher Modes of Operation

- ECB is considered insecure if applied to more than one block: Identical input blocks are mapped to identical output blocks.
- In CBC and CFB mode, the last ciphertext block C_t depends on all message blocks $M_1, \dots, M_t$, in ECB, OFB, and CTR mode each block of ciphertext C_i only on message block M_i.
- CBC, CFB, and OFB encryption can not be performed in parallel on several blocks, ECB and CTR encryption can. CBC and CFB decryption also can be performed in parallel.
- Only ECB and CTR allow random access to the ciphertext.
- CBC and ECB require padding of the input to a multiple of the block size, CFB, OFB, and CTR don't.

Properties of the Block-Cipher Modes of Operation

- ECB is considered insecure if applied to more than one block: Identical input blocks are mapped to identical output blocks.
- In CBC and CFB mode, the last ciphertext block C_t depends on all message blocks $M_1, \ldots, M_t$, in ECB, OFB, and CTR mode each block of ciphertext C_i only on message block M_i.
- CBC, CFB, and OFB encryption can not be performed in parallel on several blocks, ECB and CTR encryption can. CBC and CFB decryption also can be performed in parallel.
- Only ECB and CTR allow random access to the ciphertext.
- CBC and ECB require padding of the input to a multiple of the block size, CFB, OFB, and CTR don't.
- For OFB, CFB, and CTR mode each two messages encrypted with he same key must use a different IV.

Properties of the Block-Cipher Modes of Operation

- ECB is considered insecure if applied to more than one block: Identical input blocks are mapped to identical output blocks.
- In CBC and CFB mode, the last ciphertext block C_t depends on all message blocks $M_1, \dots, M_t$, in ECB, OFB, and CTR mode each block of ciphertext C_i only on message block M_i.
- CBC, CFB, and OFB encryption can not be performed in parallel on several blocks, ECB and CTR encryption can. CBC and CFB decryption also can be performed in parallel.
- Only ECB and CTR allow random access to the ciphertext.
- CBC and ECB require padding of the input to a multiple of the block size, CFB, OFB, and CTR don't.
- For OFB, CFB, and CTR mode each two messages encrypted with he same key must use a different IV.
- Most widely used modes are CBC and CTR.

An Example for Block Ciphers: AES

History:

- **September 1997:** NIST issued a public call for a new block cipher, supporting a block length of 128 bits and lengths of 128, 192, and 256 bits.
- **August 1998 and March 1999:** AES1 and AES2 conferences organized by NIST.
- **August 1999:** NIST announces 5 finalists:
  - MARS (IBM)
  - RCG (Rivest, Robshaw, Sidney, Yin)
  - Rijndael (Daemen, Rijmen)
  - Serpent (Anderson, Biham, Knudsen)
  - Twofish (Schneier)
- **April 2000:** AES3 conference
- **October 2nd, 2000:** NIST announces that Rijndael has been selected as the proposed AES

An Example for Block Ciphers: AES

Parameters:

- fixed block size of 128bit
- variable key size (in bits): AES-128, AES-192, AES-256

Animation:

`http://www.cs.bc.edu/~straubin/cs381-05/blockciphers/rijndael_ingles2004.swf`

An Example for Block Ciphers: AES

Rijndael S-box:

For y in $GF(2^8) = GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$ compute

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

with $z = y^{-1}$.

An Example for Block Ciphers: AES

Rijndael S-box:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

An Example for Block Ciphers: AES

Optimizations for 32-bit Architectures:

- Lookup tables $T_0, \ldots, T_3$ combining all steps.

An Example for Block Ciphers: AES

Optimizations for 32-bit Architectures:

- Lookup tables $T_0, \ldots, T_3$ combining all steps.

Security Concerns:

- Theoretical attacks reduce security of AES-128 to $2^{126.1}$.
- Cache-timing attacks are practical attacks but require precise timing measurements.
  $\rightarrow$ AES implementations must be resistant to timing attacks!

An Example for Block Ciphers: AES

Optimizations for 32-bit Architectures:

- Lookup tables $T_0, \dots, T_3$ combining all steps. ⚠

Security Concerns:

- Theoretical attacks reduce security of AES-128 to $2^{126.1}$.
- Cache-timing attacks are practical attacks but require precise timing measurements.
  $\rightarrow$ AES implementations must be resistant to timing attacks!

An Example for Block Ciphers: AES

Optimizations for 32-bit Architectures:

- Lookup tables $T_0, \ldots, T_3$ combining all steps. ⚠

Security Concerns:

- Theoretical attacks reduce security of AES-128 to $2^{126.1}$.
- Cache-timing attacks are practical attacks but require precise timing measurements.
  $\rightarrow$ AES implementations must be resistant to timing attacks!

High-Speed Implementations:

- NaCl: `http://nacl.cr.yp.to/features.html`
- `http://cryptojedi.org/crypto/index.shtml#aesbs`

Cryptographic Attack Methods

Plaintext-Based Attacks:

- known plaintext
- chosen plaintext
- adaptive chosen plaintext

Cryptographic Attack Methods

Plaintext-Based Attacks:

- known plaintext
- chosen plaintext
- adaptive chosen plaintext

Ciphertext-Based Attacks:

- ciphertext only
- chosen ciphertext
- adaptive chosen ciphertext

Cryptographic Attack Methods

Plaintext-Based Attacks:

- known plaintext
- chosen plaintext
- adaptive chosen plaintext

Ciphertext-Based Attacks:

- ciphertext only
- chosen ciphertext
- adaptive chosen ciphertext

Linear Cryptanalysis:

- known plaintext attack
- statistical analysis against large amounts of plaintext

Cryptographic Attack Methods

Plaintext-Based Attacks:

- known plaintext
- chosen plaintext
- adaptive chosen plaintext

Ciphertext-Based Attacks:

- ciphertext only
- chosen ciphertext
- adaptive chosen ciphertext

Linear Cryptanalysis:

- known plaintext attack
- statistical analysis against large amounts of plaintext

Differential Cryptanalysis:

- chosen plaintext attack
- statistical analysis of the difference of two inputs and the difference of the outputs

Literature

Stream and Block Ciphers:

Chapter 6 and 7, *Handbook of Applied Cryptography*,
A. Menezes, P. van Oorschot, and S. Vanstone,
CRC Press, 1996.

AES:

AES Proposal Rijndael,
Joan Daemen, Vincent Rijmen

Linear and Differential Cryptanalysis:

A Tutorial on Linear and Differential Cryptanalysis,
Howard M. Heys