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 $\sigma_0$:

state: $\sigma_{i+1} = f(\sigma_i, K)$ 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 $h$
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 $\sigma_0$:

- state: $\sigma_{i+1} = f(\sigma_i, K)$ 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: \( \sigma_{i+1} = (c_i, c_{i-1}, \ldots c_{i-t}) \)
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 \( h \)
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: $\sigma_{i+1} = (c_i, c_{i-1}, \ldots c_{i-t})$
- 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 $E : \{0, 1\}^n \times \mathcal{K} \rightarrow \{0, 1\}^n$. For each fixed key $K \in \mathcal{K}$ 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
    \[
    C_i = E_K(M_i), \quad i = 1 \ldots t
    \]
From Block Ciphers to Stream Ciphers

Mode of Operation:

- **Electronic codebook (ECB) mode:**
  - Decryption:
    - obtain plaintext $M_1, \ldots, M_t$ as $M_i = E_k^{-1}(C_i), \ i = 1 \ldots t$
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 diagram]
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
    $$M_i = E_K^{-1}(C_i) \oplus C_{i-1}, \ i = 1 \ldots t, \ C_0 = IV$$
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
  $$C_i = E_K(C_{i-1}) \oplus M_i, \ i = 1 \ldots t, \ C_0 = IV$$
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
Output feedback (OFB) mode:
Use a (non-secret) initialization vector (IV) of length $n$ bits.

Encryption:
obtain ciphertext $C_1, \ldots, C_t$ as

\[
C_i = O_i \oplus M_i, \quad i = 1, \ldots, t,
\]

\[
O_i = E_K(O_{i-1}), \quad O_0 = IV
\]
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
    \[
    M_i = O_i \oplus C_i, \ i = 1 \ldots t,
    \]
    \[
    O_i = E_K(O_{i-1}), \ O_0 = IV
    \]
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:
Use a (non-secret) initialization vector (IV) of length \( n \) bits.

Decryption:
Obtain plaintext \( M_1, \ldots, M_t \) as
\[
M_i = E_K(N_i) \oplus C_i, \quad i = 1 \ldots t,
\]
where \( N_i = N_{i-1} + 1 \mod 2^n \), \( N_0 = IV \).
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 $M_1, \ldots, M_t$. In ECB, OFB, and CTR mode, each block of ciphertext $C_i$ only depends on message block $M_i$.
- CBC, CFB, and OFB encryption cannot 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 the same key must use a different IV.
- Most widely used modes are CBC and CTR.
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.

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 the same key must use a different IV.

Most widely used modes are CBC and CTR.
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 $M_1, \ldots, M_t$, 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.
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.
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 the 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 $M_1, \ldots, M_t$, 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 the 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

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

with \( z = y^{-1} \).
An Example for Block Ciphers: AES

Rijndael S-box:

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

AES:
*AES Proposal Rijndael*,
Joan Daemen, Vincent Rijmen

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