Lab on Testing Different Modes in Symmetric Ciphers

Symmetric key cryptography provides several modes of operation, including Electronic Codebook (ECB), Cipher-Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter Mode (CTR), as shown in Figure 1. Modes of operation have been devised to encipher text of any size employing either DES or AES. Two important properties of these encryption modes that this lab will explore are pattern preservation and error propagation. Pattern preservation means that a block of plaintext is encrypted into a block of cipher text the same way every time; e.g. if Eve finds out that cipher text blocks 1, 5, and 10 are the same, she knows that plaintext blocks 1, 5, and 10 are the same. Error propagation means that a single bit error in transmission of a cipher text block creates errors in not only the decryption of the affected block, but propagates to the following blocks of the message.

Figure 1. Modes of Operation

Lab Tasks

Create an application to encrypt and decrypt messages using DES or AES ciphers using a programming language/cryptographic package of your own choice. Java has a mature offering in the form of its Java Cryptography Extension, which is integrated with the Java 2 SE SDK. An article on using AES with Java can be found here:

http://java.sun.com/developer/technicalArticles/Security/AES/AES_v1.html. BSAFE from RSA is available under share project. You can choose a language/package that allows for the selection of operation mode (ECB, CBC, etc.) for encryption/decryption.

Task 1 Implement DES and AES ciphers.

Create an application in the language of your choice that implements encryption of a plaintext series of bytes, and decryption of the created cipher text.
Task 2 Investigate Properties of Modes in DES and AES

Your application should use either AES or DES encryption, and employ each of the following algorithm modes: ECB, CBC, CFB, OFB, and CTR. Your application should test pattern preservation by encrypting plaintext that includes a pattern, and examining the cipher text to see if the pattern is preserved. Your program should test error propagation by modifying one byte of the cipher text prior to decryption, and then examining the decrypted plaintext. The output from your program should resemble the following:

opmode: CFB
input : 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 00 01 02 03 04 05 06 07
cipher: 61 a1 f8 86 ff 9b c7 09 4f c0 bc 1b 17 3a d7 bb c7 d7 1a 36 61 45 dd a8
Modifying random byte: bb->ba
plain : 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0e 34 8b 0c bf fb 7f 9c de

Prepare a written report of your examination of the two discussed properties of the different cryptographic modes of operation. Include a completed version of the table below in your report (fill in each block with a yes, no, or other comment). Include the source code of your application with your report.

<table>
<thead>
<tr>
<th></th>
<th>ECB</th>
<th>CBC</th>
<th>CFB</th>
<th>OFB</th>
<th>CTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>preservation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>propagation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hint:
To test pattern preservation property, you can include repeated blocks in your plaintext and observe the results of cipher text.
To test error propagation property, you can encrypt plaintext first and then modify one bit in ciphertext and check the decryption results.