Archived:CBC mode encryption and decryption of data

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

This article needs to be updated: If you found this article useful, please fix the problems below then delete the {{ArticleNeedsUpdate}} template from the article to remove this warning.

Reasons: hamishwillee (10 May 2011)
SymbianCryptographyLibrary appears to be unavailable. Needs to be found, or if no longer available this article needs to be deleted.

Overview

Cipher-block chaining (CBC) is an encryption mode where each block of plain text is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block is dependent on all plain text blocks processed up to that point. Also, to make each message unique, an initialisation vector must be used in the first block.

Description

The class `CModeCBCEncryptor`, available in the Symbian Cryptography library, can be used for encryption of data. It is initialised with a subclass of `CBlockTransformation`, in this case `CAESEncryptor`, which it subsequently owns. The following code snippet shows how to encrypt and decrypt a block of data in CBC mode.

Solution

CBC mode encryption of data:

```
void CCBCExAppUi::CBCEncryption(TPtr8& aDataPtr)
{
 CAESEncryptor* aesEncryptor = CAESEncryptor::NewL(iCipherkey);
 CModeCBCEncryptor* cbcEncryptor = CModeCBCEncryptor::NewLC(aesEncryptor, iIV);<br>
for(TInt i = 0; i<3; i++)
{
  TPtr8 tempDataptr = aDataPtr.MidTPtr(16*i,16);
  cbcEncryptor->Transform(tempDataptr);
  cbcEncryptor->SetIV(tempDataptr);
}<br>
CleanupStack::PopAndDestroy();
}
```

CBC mode decryption of data:

```
void CCBCExAppUi::CBCDecryption(TPtr8& aDataPtr)
{
CAESDecryptor* aesdecryptor = CAESDecryptor::NewL(iCipherkey);
CModeCBCDecryptor* cbcDecryptor = CModeCBCDecryptor::NewLC(aesdecryptor, iIV);<br>
TBuf8<16>tempBuf;
for(TInt i = 0; i<3; i++)
{
  TPtr8 tempDataptr = aDataPtr.MidTPtr(16*i,16);
  tempBuf.Copy(tempDataptr);
```

```
    cbcDecryptor->Transform(tempDataptr);
    cbcDecryptor->SetIV(tempBuf);
  }<br>
  CleanupStack::PopAndDestroy();
  }
```

Example application

File:CBCEx.zip