bcgit / **bc-java** Public mirror

mirrored from https://www.bouncycastle.org/repositories/bc-java

Code | Issues 299 | Pull requests 40 | Discussions | Actions | Projects

Commit b425743

dghgit committed on Dec 4, 2025

```
refactored counter code.
```

main · r1rv84

1 parent 5a4ccfd commit b425743

2 files changed +71 -9 lines changed

Top

Filter files…

- core/src
  - main/java/org/bouncycastle/crypto/modes
    - G3413CTRBlockCipher.java
  - test/java/org/bouncycastle/crypto/test
    - GOST3412Test.java

2 files changed +71 -9 lines changed

Search within code

...astle/crypto/modes/G3413CTRBlockCipher.java

```
@@ -13,8 +13,6 @@
13 13   public class G3413CTRBlockCipher
14 14       extends StreamBlockCipher
15 15   {
16    -
17    -
18 16       private final int s;
19 17       private byte[] CTR;
20 18       private byte[] IV;
```

```
@@ -24,7 +22,6 @@ public class G3413CTRBlockCipher
24  22      private int byteCount = 0;
25  23      private boolean initialized;
26  24
27      -
28  25      /**
29  26       * Basic constructor.
30  27       *
@@ -184,27 +181,34 @@ protected byte calculateByte(byte in)
184 181          if (byteCount == s)
185 182          {
186 183              byteCount = 0;
187     -            generateCRT();
        184 +        generateCTR();
188 185          }
189 186
190 187          return rv;
191 188
192 189      }
193 190
194     -    private void generateCRT()
        191 +    private void generateCTR()
195 192      {
196     -        CTR[CTR.length - 1]++;
        193 +        int start = CTR.length - 1;
        194 +        while (++CTR[start] == 0)
        195 +        {
        196 +            start--;
        197 +            if (start == IV.length - 1)
        198 +            {
        199 +                throw new IllegalStateException("attempt to process too many
    blocks");
        200 +            }
        201 +            CTR[start]++;
        202 +        }
203 203      }
204 204
205 205
206 206      private byte[] generateBuf()
```

```
201  207         {
202       -
203  208             byte[] encryptedCTR = new byte[CTR.length];
204  209             cipher.processBlock(CTR, 0, encryptedCTR, 0);
205  210
206  211             return GOST3413CipherUtil.MSB(encryptedCTR, s);
207       -
208  212         }
209  213
210  214
```

.../bouncycastle/crypto/test/GOST3412Test.java

```
     •••    @@ -1,12 +1,14 @@
1    1      package org.bouncycastle.crypto.test;
2    2
       3  + import org.bouncycastle.crypto.StreamBlockCipher;
3    4      import org.bouncycastle.crypto.engines.GOST3412_2015Engine;
4    5      import org.bouncycastle.crypto.modes.G3413CBCBlockCipher;
5    6      import org.bouncycastle.crypto.modes.G3413CFBBlockCipher;
6    7      import org.bouncycastle.crypto.modes.G3413CTRBlockCipher;
7    8      import org.bouncycastle.crypto.modes.G3413OFBBlockCipher;
8    9      import org.bouncycastle.crypto.params.KeyParameter;
9    10     import org.bouncycastle.crypto.params.ParametersWithIV;
       11 + import org.bouncycastle.util.Arrays;
10   12     import org.bouncycastle.util.encoders.Hex;
11   13     import org.bouncycastle.util.test.SimpleTest;
12   14
     ↓ ↑    @@ -85,12 +87,68 @@ public String getName()
85   87         public void performTest()
86   88             throws Exception
87   89         {
88        -        super.performTest();
       90 +        //super.performTest();
89   91
       92 +        ctrTest();
90   93     //        cfbTest();
91   94     //        ofbTest();
92   95         }
```

```
93  96
        97  +     private void ctrTest()
        98  +         throws Exception
        99  +     {
       100  +         StreamBlockCipher sb = new G3413CTRBlockCipher(new
              GOST3412_2015Engine(), 128);
       101  +
       102  +         sb.init(true, new ParametersWithIV(new
              KeyParameter(Hex.decode("8899aabbccddeeff0011223344556677fedcba9876543210012345
              6789abcdef")),
       103  +             Hex.decode("0001020304050607")));
       104  +
       105  +         byte[] block = Hex.decode("000102030405060708090a0b0c0d0e0f");
       106  +         byte[] output = new byte[16];
       107  +         byte[] last = new byte[16];
       108  +
       109  +         sb.processBytes(block, 0, block.length, last, 0);
       110  +
       111  +         for (int i = 1; i < 255; i++)
       112  +         {
       113  +             sb.processBytes(block, 0, block.length, output, 0);
       114  +             if (Arrays.areEqual(last, output))
       115  +             {
       116  +                 fail("cipher text repeats 1");
       117  +             }
       118  +         }
       119  +
       120  +         sb.processBytes(block, 0, block.length, output, 0);
       121  +         if (Arrays.areEqual(last, output))
       122  +         {
       123  +             fail("cipher text repeats 2");
       124  +         }
       125  +
       126  +         sb = new G3413CTRBlockCipher(new GOST3412_2015Engine(), 128);
       127  +
       128  +         sb.init(true, new ParametersWithIV(new
              KeyParameter(Hex.decode("8899aabbccddeeff0011223344556677fedcba9876543210012345
              6789abcdef")),
       129  +             Hex.decode("0001020304050607")));
       130  +
```

```
131 +          sb.processBytes(block, 0, block.length, last, 0);
132 +
133 +          for (int i = 1; i != ((1 << 15) - 1); i++)
134 +          {
135 +              sb.processBytes(block, 0, block.length, output, 0);
136 +              if (Arrays.areEqual(last, output))
137 +              {
138 +                  fail("cipher text repeats 3");
139 +              }
140 +              byte[] tmp = last;
141 +              last = output;
142 +              output = tmp;
143 +          }
144 +
145 +          sb.processBytes(block, 0, block.length, output, 0);
146 +          if (Arrays.areEqual(last, output))
147 +          {
148 +              fail("cipher text repeats");
149 +          }
150 +      }
151 +
94  152       public static void main(
95  153           String[] args)
96  154       {
```

Comments 0

Please sign in to comment.