Multiple Examples of DSA

July 31, 2003

FIPS 186-2 with Change Notice #1 dated October 5, 2001 allows the PRNG to be either the one originally specified in Appendix 3 of the standard or the one specified in the Change Notice. Additionally, these PRNG's specify the use of a one-way function G(t,c). These examples include both the original and Change Notice versions of the PRNG's, as well as two different G(t,c) functions (the first is based on SHA-1 and the second is based on DES).

1. 1024-bit Modulus with Original PRNG, G(t,c) using SHA-1

1.1 Constructing a set of Domain Parameters (P, Q, and G)

The following SEED value is from step 1 in Appendix 2.2:

```
SEED=     1444c1df 2830a665 9a80ed71 c06e9de5 0652f7ea
```

The rest of the algorithm in Appendix 2.2 results in the following values for *q* and *p*:

```
q=        b5afd2f9 3246b1ef cd1f3a7c 240c1e9e 21a3630b

p=        a65feaab 511c61e3 3df38fdd daf03b59 b6f25e1f
          a4de57e5 cf00ae47 8a855dda 4f3638d3 8bb00ac4
          af7d8414 c3fb36e0 4fbdf3d3 166712d4 3b421bfa
          757e8569 4ad27c48 f396d03c 8bce8da5 8db5b820
          39f35dcf 857235c2 f1c73b22 26a36142 9190dcb5
          b6cd0edf b0ff6933 900b02ce cc0ce692 74d8dae7
          c6948043 18d6d6b9
```

With the following counter value:

```
counter=  24
```

To compute the value of *g* using the method in Appendix 4 (not a requirement of the standard), the following value of *h* was selected in step 3:

```
h=        2
```

And produces:

```
g=        007bbd2c 5dc917a5 e08b9c2f 80a49fb6 3fcd5c05
          78ba701e 254fe353 0dedd3b6 680a6e5a fb3280b5
          3f154028 bafff73d 1ba0fdb0 004b9eb0 dbf24b29
          5bf2a356 913cd1c0 be03c510 3a1da8b7 3e7670b5
          6d716ed5 547af67b 5061311e ea245e2e 5c337843
          cbc135b9 b9c18775 d5d56cfd a31b747e 2449861a
          df3b3f72 7189c0a3
```

1.2 *Computing a Private and Public Key Pair*

Using the algorithm in Appendix 3.1 for computing *x* values:

```
XKEY=     bd029bbe 7f51960b cf9edb2b 61f06f0f eb5a38b6

XSEED=    00000000 00000000 00000000 00000000 00000000

XVAL=     bd029bbe 7f51960b cf9edb2b 61f06f0f eb5a38b6
```

Using the routine in Appendix 3.3, Constructing The Function G From SHA-1, in step 3.c of Appendix 3.1 provides the following values from the G function and for *x*:

```
G()=      2070b322 3dba372f de1c0ffc 7b2e3b49 8b260614

x=        2070b322 3dba372f de1c0ffc 7b2e3b49 8b260614
```

The following value is the updated XKEY value from step 3.d:

```
XKEY=     dd734ee0 bd0bcd3b adbaeb27 dd1eaa59 76803ecb
```

The value of *x* computed above results in the following *y* value:

```
y=        87c9b20a aef34afc bd6ffb55 09e7cb3b 43f8bec5
          6ba74ad0 89d2ac26 59b9fa8f 895d51b5 9891f0a5
          afe8b2e1 1ae133ac 16529ffc 031eedf7 834f6c1b
          ce2604c4 e5cc750d f577d29c 08f0a6e4 f7e190d2
          1b683fb6 e08f4d9e a6ea1f03 d7720cea 0a97c039
          69118dea 97d3efc3 0d0dcd80 495cf2ea 84eac1b4
          4fb3d2b8 e25e0bd8
```

1.3 *Generating a Signature*

```
M=        ASCII form of "abc"

SHA-1(M)= a9993e36 4706816a ba3e2571 7850c26c 9cd0d89d
```

Using the algorithm in Appendix 3.2 for computing *k* values:

```
KKEY=     687a66d9 0648f993 867e121f 4ddf9ddb 01205584
```

Using the routine in Appendix 3.3, Constructing The Function G From SHA-1, in step 3.a of Appendix 3.2 provides the following values from the G function and for *k*:

```
G()=      fd00cee3 87e139fd ea1da3fd 07685fba b979711e

k=        4750fbea 559a880e 1cfe6980 e35c411c 97d60e13
```

The following value is the updated KKEY value from step 3.d:

```
KKEY=     afcb62c3 5be381a1 a37c7ba0 313bdef7 98f66398
```

The method in Appendix 4 for computing the multiplicative inverse provides the following value of k^{-1}:

k^{-1}= 74884c0c 2bccf6c1 f2f225e8 cb5352e9 b5590aa5

Computing the Digital Signature as specified in Section 5 yields:

```
r=        b1d237b1 af083174 9cfa5557 edf2327b 84835270

s=        1711e4c0 94cfc31a 33b2fc71 e8cc7061 7b31ab52
```

1.4 *Verifying a Signature*

Verifying the Digital Signature as specified in Section 6 provides:

```
w=        01f8f24c c5249634 db93f9b0 ab2ecacd 63868502

u1=       49db8fa7 93c1726f 025b6354 6094927b 55b1fb89

u2=       3a05c13e 9f9a9dc7 b5525be4 84574fa4 535588bb
```

g^{u1} mod p= 77bb0678 dd781c4e 95b43893 26d43749 f6604981
41e56695 c724c06b 96f2e15c 3bfe605b eaf55c7e
7f42b0b5 9696da5f 2ee65cb8 c83b2f19 2bad08b1
814be325 36c4b819 8b04c2c0 019d5a79 eb7cbf9d
6f47b063 a1bbca1f 39a2212f a3f4b4b5 261dc041
43ea563c cfcc1504 a6cd6ac1 08fcc407 fff6c0ae
c6091c90 49e40b4b

y^{u2} mod p= 8799225f d0943d2e c39d25f8 f10e737e c0ef0c6b
9c9574a7 37f48aa2 90417c87 5bcce77f 7a472b8a
d8c4fbd7 1d7cb990 3f2524e1 f710acce 3f868ff7
2a580e38 ce04434c a95e22de 74770119 1355f063
14a5dbf5 4ce25d8b 0208b515 12acb9e9 bb818924
ddd86e84 7e100c98 012466a2 4084c5f0 616bb574
1a8ebf80 7bc71887

```
v=        b1d237b1 af083174 9cfa5557 edf2327b 84835270
```

Since v=r’, the signature is verified.

2. 1024-bit Modulus with Change Notice PRNG, G(t,c) using SHA-1

2.1 Constructing a set of Domain Parameters (P, Q, and G)

The following SEED value is from step 1 in Appendix 2.2:

```
SEED=     1444c1df 2830a665 9a80ed71 c06e9de5 0652f7ea
```

The rest of the algorithm in Appendix 2.2 results in the following values for q and p:

```
q=        b5afd2f9 3246b1ef cd1f3a7c 240c1e9e 21a3630b

p=        a65feaab 511c61e3 3df38fdd daf03b59 b6f25e1f
          a4de57e5 cf00ae47 8a855dda 4f3638d3 8bb00ac4
          af7d8414 c3fb36e0 4fbdf3d3 166712d4 3b421bfa
          757e8569 4ad27c48 f396d03c 8bce8da5 8db5b820
          39f35dcf 857235c2 f1c73b22 26a36142 9190dcb5
          b6cd0edf b0ff6933 900b02ce cc0ce692 74d8dae7
          c6948043 18d6d6b9
```

With the following counter value:

```
counter=  24
```

To compute the value of g using the method in Appendix 4 (not a requirement of the standard), the following value of h was selected in step 3:

```
h=        2
```

And produces:

```
g=        007bbd2c 5dc917a5 e08b9c2f 80a49fb6 3fcd5c05
          78ba701e 254fe353 0dedd3b6 680a6e5a fb3280b5
          3f154028 bafff73d 1ba0fdb0 004b9eb0 dbf24b29
          5bf2a356 913cd1c0 be03c510 3a1da8b7 3e7670b5
          6d716ed5 547af67b 5061311e ea245e2e 5c337843
          cbc135b9 b9c18775 d5d56cfd a31b747e 2449861a
          df3b3f72 7189c0a3
```

2.2 *Computing a Private and Public Key Pair*

Using the revised algorithm found in the Change Notice for computing x values:

```
XKEY=     bd029bbe 7f51960b cf9edb2b 61f06f0f eb5a38b6

XSEED=    00000000 00000000 00000000 00000000 00000000
```

The first loop through step 3.2 provides:

```
     XVAL=     bd029bbe 7f51960b cf9edb2b 61f06f0f eb5a38b6
```

Using the routine in Appendix 3.3, Constructing The Function G From SHA-1, in step 3.2.b of the Change Notice algorithm for computing values of *x* provides:

```
     w[0]=     2070b322 3dba372f de1c0ffc 7b2e3b49 8b260614
```

The following value is the updated XKEY value from step 3.2.c:

```
     XKEY=     dd734ee0 bd0bcd3b adbaeb27 dd1eaa59 76803ecb
```

The second loop through step 3.2 provides:

```
     XVAL=     dd734ee0 bd0bcd3b adbaeb27 dd1eaa59 76803ecb
```

Using the routine in Appendix 3.3, Constructing The Function G From SHA-1, in step 3.2.b of the Change Notice algorithm for computing values of *x* provides:

```
     w[1]=     3c6c18ba cb0f6c55 babb1378 8e20d737 a3275116
```

The following value is the updated XKEY value from step 3.2.c:

```
     XKEY=     19df679b 881b3991 6875fea0 6b3f8191 19a78fe2
```

Step 3.3 provides the following values:

```
w[0] || w[1]=  2070b322 3dba372f de1c0ffc 7b2e3b49 8b260614
               3c6c18ba cb0f6c55 babb1378 8e20d737 a3275116

X=        47c27eb6 16dba413 91e5165b e9c5e397 7e39a15d
```

The value of X computed above results in the following *y* value:

```
y=        6f072da4 e8787a82 d1a37e23 ac7f843f 4f3696b1
          92f4f062 f3687084 a3185669 8dea64bd 3710e151
          64b52a33 e6a6080f c5b00596 ddbc0bc1 aa4aaba3
          61666dee efba6a93 cb99c9fb 37e3c2a2 4b20922a
          7add1dd4 d5cf23a6 9d6a285a 27d18ed0 824ae59c
          7a8228ac c259e5f7 f10d163f 08858dee b897c22b
          9abb4282 16e51c47
```

2.3 *Generating a Signature*

M= ASCII form of "abc"

SHA-1(M)= a9993e36 4706816a ba3e2571 7850c26c 9cd0d89d

Using the revised algorithm found in the Change Notice for computing values of *k*:

KKEY= 687a66d9 0648f993 867e121f 4ddf9ddb 01205584

The first loop through step 3.1 provides:

> Using the routine in Appendix 3.3, Constructing The Function G From SHA-1, in step 3.1.a of the Change Notice algorithm for computing values of *k* provides:
>
> w[0]= fd00cee3 87e139fd ea1da3fd 07685fba b979711e
>
> The following value is the updated KKEY value from step 3.1.b:
>
> KKEY= 657b35bc 8e2a3391 709bb61c 5547fd95 ba99c6a3

The second loop through step 3.1 provides:

> Using the routine in Appendix 3.3, Constructing The Function G From SHA-1, in step 3.1.a of the Change Notice algorithm for computing values of *k* provides:
>
> w[1]= bfb1c43b a8c9326c 16d8a3aa a3e35b30 b4349b31
>
> The following value is the updated KKEY value from step 3.1.b:
>
> XKEY= 252cf9f8 36f365fd 877459c6 f92b58c6 6ece61d5

Step 3.2 provides the following values:

```
w[0] || w[1]=  fd00cee3 87e139fd ea1da3fd 07685fba b979711e
               bfb1c43b a8c9326c 16d8a3aa a3e35b30 b4349b31
```

k= 952127c8 c4b38b8b ffb0defa 5ff6af91 a2a81296

The method in Appendix 4 for computing the multiplicative inverse provides the following value of k^{-1}:

k^{-1}= 088388e1 34d2da64 3c54844b 0febe082 c196e815

Computing the Digital Signature as specified in Section 5 yields:

r= 6b4fbbc0 98a514a2 3bb89c67 0587bced aae1fa69

```
s=         84737ccd 7b3ef5f6 2806c4c5 87d9e97f 2dfac5cc
```

2.4 *Verifying a Signature*

Verifying the Digital Signature as specified in Section 6 provides:

```
w=         03512709 097081db d023d9bb 63a25808 b345e7bc

u1=        86d68eba 67a36c27 0dc74cab 64f7b37f fce29071

u2=        6c332119 44b43db8 3589d044 a66ab573 58b2b305
```

g^{u1} mod p=

```
73394ea6 ebbfbceb 81c3b466 786aacfb 27f1187b
5e37905d 6526c793 11d9c6ce 40a0515a b7cc882e
9134d050 f0c0ea79 a3ab3dfa a9daabff 0531bbdf
d1f92482 edbfa8d5 907b5678 61f24386 ee785b69
3786f01e 9a7b1ce3 693bdf56 31df92bb 0b644b4d
acaf8ffc 9b141690 e82cb51d 46af6747 0b848e4b
2db2e8bc 13362f9f
```

y^{u2} mod p=

```
6f4433be 3c89f6a7 ec02d08b d495ca51 5c91159a
cb962035 8ab3e48a 97aebb7e 733660c7 0128ba8c
00ec5365 46d9f9d2 1373f259 346600fe b59f239f
1dbc25ed 82b58430 e9980570 193dd8e9 299fe62a
0b867392 b1979bfe aae916a9 0b7e8906 e7177eb4
d2ef78c7 395c4a8f 68334a7e f5735f34 13fb5be8
79016955 97c3a799
```

```
v=         6b4fbbc0 98a514a2 3bb89c67 0587bced aae1fa69
```

Since *v*=*r*’, the signature is verified.

3. 1024-bit Modulus with Original PRNG, G(t,c) using DES

3.1 Constructing a set of Domain Parameters (P, Q, and G)

The following SEED value is from step 1 in Appendix 2.2:

```
SEED=      1444c1df 2830a665 9a80ed71 c06e9de5 0652f7ea
```

The rest of the algorithm in Appendix 2.2 results in the following values for *q* and *p*:

```
q=         b5afd2f9 3246b1ef cd1f3a7c 240c1e9e 21a3630b
```

```
p=         a65feaab 511c61e3 3df38fdd daf03b59 b6f25e1f
           a4de57e5 cf00ae47 8a855dda 4f3638d3 8bb00ac4
           af7d8414 c3fb36e0 4fbdf3d3 166712d4 3b421bfa
           757e8569 4ad27c48 f396d03c 8bce8da5 8db5b820
           39f35dcf 857235c2 f1c73b22 26a36142 9190dcb5
           b6cd0edf b0ff6933 900b02ce cc0ce692 74d8dae7
           c6948043 18d6d6b9
```

With the following counter value:

```
counter=   24
```

To compute the value of *g* using the method in Appendix 4 (not a requirement of the standard), the following value of *h* was selected in step 3:

```
h=         2
```

And produces:

```
g=         007bbd2c 5dc917a5 e08b9c2f 80a49fb6 3fcd5c05
           78ba701e 254fe353 0dedd3b6 680a6e5a fb3280b5
           3f154028 bafff73d 1ba0fdb0 004b9eb0 dbf24b29
           5bf2a356 913cd1c0 be03c510 3a1da8b7 3e7670b5
           6d716ed5 547af67b 5061311e ea245e2e 5c337843
           cbc135b9 b9c18775 d5d56cfd a31b747e 2449861a
           df3b3f72 7189c0a3
```

3.2 *Computing a Private and Public Key Pair*

Using the algorithm in Appendix 3.1 for computing *x* values:

```
XKEY=      bd029bbe 7f51960b cf9edb2b 61f06f0f eb5a38b6

XSEED=     00000000 00000000 00000000 00000000 00000000

XVAL=      bd029bbe 7f51960b cf9edb2b 61f06f0f eb5a38b6
```

Using the routine in Appendix 3.4, Constructing The Function G From DES, in step 3.c of Appendix 3.1 provides the following values from the G function and for *x*:

```
G()=       ea11d565 f78d7f7b ea5a8f4d fe0336ff 1166516e

x=         3462026c c546cd8c 1d3b54d1 d9f71860 efc2ee63
```

The following value is the updated XKEY value from step 3.d:

```
XKEY=      f1649e2b 44986397 ecda2ffd 3be78770 db1d271a
```

The value of *x* computed above results in the following *y* value:

```
y=        6d1dbbc3 4b83189f 5bf06dc2 1c017a7f c28a2cc9
          92e0edf9 0b5d7ed2 72c7ce37 b84aff8c 4c7d5166
          86e8addc 93778100 df526fc6 c001e912 45e3a576
          30731f15 cc2e4e57 a87cb935 1f7ad191 5216b3cd
          1e9b3c8e e550087a a83ed879 129f1d83 f48515d5
          dc4e6000 d6491483 0219eee8 0c6f1d58 76a175f7
          09bb9c37 87720383
```

3.3 *Generating a Signature*

```
M=        ASCII form of "abc"

SHA-1(M)= a9993e36 4706816a ba3e2571 7850c26c 9cd0d89d
```

Using the algorithm in Appendix 3.2 for computing *k* values:

```
KKEY=     687a66d9 0648f993 867e121f 4ddf9ddb 01205584
```

Using the routine in Appendix 3.4, Constructing The Function G From DES, in step 3.a of Appendix 3.2 provides the following values from the G function and for *k*:

```
G()=      48520cee 821bc35e 1cc5acef df04cec2 38b5a952

k=        48520cee 821bc35e 1cc5acef df04cec2 38b5a952
```

The following value is the updated KKEY value from step 3.d:

```
KKEY=     b0cc73c7 8864bcf1 a343bf0f 2ce46c9d 39d5fed7
```

The method in Appendix 4 for computing the multiplicative inverse provides the following value of k^{-1}:

k^{-1}= a738dbb7 29bf0efb 93efc4b8 ef4db656 597849f7

Computing the Digital Signature as specified in Section 5 yields:

```
r=        1de96c74 70f8d68c 4ebbcb1c 9868818c 98cfee4c

s=        a44fcb3f 6c777c35 ea47d054 6f1ca89d 5d7b038d
```

3.4 *Verifying a Signature*

Verifying the Digital Signature as specified in Section 6 provides:

```
w=         0ec19c3d e7ba59d3 45deb285 bac4a5d3 dd21592a

u1=        4e5b8a0b d6557311 1a20691d 5474b224 21fe1da1

u2=        5846d94f bc73b417 d3148196 76220709 04c8dec0
```

g^{u1} mod p=

```
28e72acf 33740e4f 75961fbd 31ad5bc2 cf7cbe50
13d16713 7925964a 0e783635 1a442632 edb716b3
267bf185 b8105880 9b521450 8c6df030 1af1a286
c3f2080e 8c2ce34a 57389cf6 85a89dbd 6cf6d675
e3679c35 b5d546f8 3d8cf20d bba930ce 5c40b9ce
3a9db942 82281034 10936cd0 65becfe0 dcde464d
87759850 0692bd6f
```

y^{u2} mod p=

```
95d6a1bd 09c37a7c 489f9f13 f3191ef2 a3dbd166
7b8150cb 572aa45e 60123710 bafd5ca3 aae44005
1991427b c49951f5 2574ad70 9dbf4c41 a1451433
a0e6495f ebe7a73d 1b5ff8fe 4cf504cb 0f609395
57e8361a ec37dc4d 87c5aa95 f29d7ab5 2f99ca73
5ba7dea5 16a850c6 4fe9794a f570be3b 28280f50
e53ce1cf 46ca8902
```

```
v=         1de96c74 70f8d68c 4ebbcb1c 9868818c 98cfee4c
```

Since *v*=*r*’, the signature is verified.

4. 1024-bit Modulus with Change Notice PRNG, G(t,c) using DES

4.1 Constructing a set of Domain Parameters (P, Q, and G)

The following SEED value is from step 1 in Appendix 2.2:

```
SEED=      1444c1df 2830a665 9a80ed71 c06e9de5 0652f7ea
```

The rest of the algorithm in Appendix 2.2 results in the following values for *q* and *p*:

```
q=         b5afd2f9 3246b1ef cd1f3a7c 240c1e9e 21a3630b

p=         a65feaab 511c61e3 3df38fdd daf03b59 b6f25e1f
           a4de57e5 cf00ae47 8a855dda 4f3638d3 8bb00ac4
           af7d8414 c3fb36e0 4fbdf3d3 166712d4 3b421bfa
           757e8569 4ad27c48 f396d03c 8bce8da5 8db5b820
           39f35dcf 857235c2 f1c73b22 26a36142 9190dcb5
           b6cd0edf b0ff6933 900b02ce cc0ce692 74d8dae7
           c6948043 18d6d6b9
```

With the following counter value:

```
counter=  24
```

To compute the value of *g* using the method in Appendix 4 (not a requirement of the standard), the following value of *h* was selected in step 3:

```
h=         2
```

And produces:

```
g=         007bbd2c 5dc917a5 e08b9c2f 80a49fb6 3fcd5c05
           78ba701e 254fe353 0dedd3b6 680a6e5a fb3280b5
           3f154028 bafff73d 1ba0fdb0 004b9eb0 dbf24b29
           5bf2a356 913cd1c0 be03c510 3a1da8b7 3e7670b5
           6d716ed5 547af67b 5061311e ea245e2e 5c337843
           cbc135b9 b9c18775 d5d56cfd a31b747e 2449861a
           df3b3f72 7189c0a3
```

4.2 *Computing a Private and Public Key Pair*

Using the revised algorithm found in the Change Notice for computing *x* values:

```
XKEY=      bd029bbe 7f51960b cf9edb2b 61f06f0f eb5a38b6

XSEED=     00000000 00000000 00000000 00000000 00000000
```

The first loop through step 3.2 provides:

```
     XVAL=      bd029bbe 7f51960b cf9edb2b 61f06f0f eb5a38b6
```

Using the routine in Appendix 3.4, Constructing The Function G From DES, in step 3.2.b of the Change Notice algorithm for computing values of *x* provides:

```
     w[0]=      ea11d565 f78d7f7b ea5a8f4d fe0336ff 1166516e
```

The following value is the updated XKEY value from step 3.2.c:

```
     XKEY=      a7147124 76df1587 b9f96a79 5ff3a60e fcc08a25
```

The second loop through step 3.2 provides:

```
     XVAL=      a7147124 76df1587 b9f96a79 5ff3a60e fcc08a25
```

Using the routine in Appendix 3.4, Constructing The Function G From DES, in step 3.2.b of the Change Notice algorithm for computing values of *x* provides:

```
    w[1]=     a3198afc a3ffd705 20fd68c3 2abdfe47 c305c2f8
```

The following value is the updated XKEY value from step 3.2.c:

```
    XKEY=     4a2dfc21 1adeec8c daf6d33c 8ab1a456 bfc64d1e
```

Step 3.3 provides the following values:

```
w[0] || w[1]=  ea11d565 f78d7f7b ea5a8f4d fe0336ff 1166516e
               a3198afc a3ffd705 20fd68c3 2abdfe47 c305c2f8

X=        2e108577 08b9fde1 2af830d1 2a028ff6 df7c5c8f
```

The value of *x* computed above results in the following *y* value:

```
y=        96bed4a2 87c8660c 8b829bd8 6cd5fde7 0d0fd04c
          a92417a4 622b54a0 ab3c9ae2 271d7aad 962d1244
          502f1fc8 ef1c7ae9 75c05ee0 5ae1f957 5052b4d5
          97ad3c3b caf19195 68efa227 205052dc 08a24b0a
          c4b622ef c13e50f8 9582e539 437531f5 697e6e77
          e2d4be4c af272e62 d8e1d262 479f63a7 4230f380
          c54d903c c7e9c5f2
```

4.3 *Generating a Signature*

```
M=        ASCII form of "abc"

SHA-1(M)= a9993e36 4706816a ba3e2571 7850c26c 9cd0d89d
```

Using the revised algorithm found in the Change Notice for computing *k* values:

```
KKEY=     687a66d9 0648f993 867e121f 4ddf9ddb 01205584
```

The first loop through step 3.1 provides:

Using the routine in Appendix 3.4, Constructing The Function G From DES, in step 3.1.a of the Change Notice algorithm for computing values of *k* provides:

```
    w[0]=     48520cee 821bc35e 1cc5acef df04cec2 38b5a952
```

The following value is the updated KKEY value from step 3.1.b:

```
    KKEY=     b0cc73c7 8864bcf1 a343bf0f 2ce46c9d 39d5fed7
```

The second loop through step 3.1 provides:

Using the routine in Appendix 3.4, Constructing The Function G From DES, in step 3.1.a of the Change Notice algorithm for computing values of *k* provides:

```
w[1]=     02e273fa 19931c33 56995269 e912baa2 e046ba2d
```

The following value is the updated KKEY value from step 3.1.b:

```
KKEY=     b3aee7c1 a1f7d924 f9dd1179 15f72740 1a1cb905
```

Step 3.2 provides the following values:

```
w[0] || w[1]=  48520cee 821bc35e 1cc5acef df04cec2 38b5a952
               02e273fa 19931c33 56995269 e912baa2 e046ba2d

k=        ab2ab897 ce90c05a 343cc115 4afa19a5 0170d7ef
```

The method in Appendix 4 for computing the multiplicative inverse provides the following value of k^{-1}:

k^{-1}=   730dbb67 1eaba1e1 690e4d6a b9639bc4 85d55b46

Computing the Digital Signature as specified in Section 5 yields:

```
r=        5575e318 c46c106b 28d5f267 17b6774b 658ac833

s=        63465d98 63a44b6f 0ccd8fab 7efaeb4f e84793df
```

4.4 *Verifying a Signature*

Verifying the Digital Signature as specified in Section 6 provides:

```
w=        17f8bc2f 19a08e2e b6690b02 956e5037 1b0ac7c6

u1=       a32472ed 4e93ec36 33ff6855 54a58001 1387bd89

u2=       77f2b06c b75b3537 c1d58787 bbdb2f9d d7a3f834
```

g^{u1} mod p= 94cb35a0 f2a87250 99ca0f50 5fe93244 756b3ec9
6c0e118b 9fe98799 da861e82 f0b0b760 49442d72
8e1124e9 2715272e f7726614 541a182f 08df3ede
b1eab19d 29dfcf3b 3f4016c0 f41bdded 11f3ca06
edc5dc76 f00342d0 4e94e14f 11241e82 878751df
a2cb196b 80486b26 acb04965 330c5450 c77b2528
78edabda ff12bb22

y^{u2} mod p= 212ba68a 63f428c3 78439ae7 291334e8 f488086d
24bd0c94 e42a81e5 93a29420 15557b46 1b4f3433

```
            fd9c1be7 00d5e876 bf387ede ccd10836 61f23fa6
            0b25fd3b e8e07cb4 c31d05d2 271489e6 9cd1efcd
            2c10c505 e95a1da6 5c5b4832 0ceaceec a463c7c5
            486cf044 021f8390 9f5177bf 66d18823 3a39c468
            fb1ef932 2e85e864

v=          5575e318 c46c106b 28d5f267 17b6774b 658ac833
```

Since $v=r'$, the signature is verified.