Vic Ludwig

I claim that for the RC6 block cipher $P = D_K(E_K(P))$, where P is the plaintext input and D_K is the decryption function and E_K is the encryption function both with appropriate keys.

The procedure for encryption is:

```
B = B + S[0]
D = D + S[1]
for i = 1 to r do {
        t = ( B x ( 2B + 1 ) ) <<< lg w
        u = ( D x ( 2D + 1 ) ) <<< lg w
        A = ( ( A ⊕ t ) <<< u ) + S[2i]
        C = ( ( C ⊕ u ) <<< t ) + S[2i + 1]
        ( A, B, C, D ) = ( B, C, D, A )
}
A = A + S[2r + 2]
C = C + S[2r + 3]
```

The procedure for decryption is:

```
C = C – S[2r + 3]
A = A – S[2r + 2]
for i = 1 to r do {
        ( A, B, C, D ) = ( D, A, B, C )
        u = ( D x ( 2D + 1 ) ) <<< lg w
        t = ( B x ( 2B + 1 ) ) <<< lg w
        C = ( ( C – S[2i + 1] ) >>> t ) ⊕ u
        A = ( ( A – S[2i] ) >>> u ) ⊕ t
}
D = D – S[1]
B = B – S[0]
```

For this proof I will assume that r = 1 as demonstrating the reversibility of the function for additional rounds would not shed any additional light on the subject. As the rounds are all identical in structure if one may be decrypted effectively then a larger number could be also.

As the input data for RC5 is broken into four 32 bit registers saying that $P = D_K(E_K(P))$ is equivalent to saying $[A, B, C, D] = D_K(E_K([A, B, C, D]))$. So if i could demonstrate that the same data ended up in the same register after decryption as was there before encryption I would have proved that $D_K = E_K^{-1}$.

Regardless of what else may have happened to the data in registers A and C (including being switched with other registers), the last operation of the encryption sequence is a simple addition. So the encryption function may be written in this manner:

$E_K(A) = E_{K1}(A) + S[2r + 2]$
$E_K(C) = E_{K2}(C) + S[2r + 3]$

This could be thought of more intuitively as, “things happen to register A and then S[2r + 3] is added to it.”

Note also that the first step of the decryption process is a subtraction from registers A and C which may be written like this:

$D_K(A) = D_{K1}(A - S[2r + 2])$
$D_K(C) = D_{K2}(C - S[2r + 3])$

Or that, “S[2r + 2] is subtracted from A and then things happen to it.”

From this one can see that the last two steps of encryption and the first two steps of decryption cancel one another out.

$A = D_{K1}((E_{K1}(A) + S[2r + 2]) - S[2r + 2])$
$A = D_{K1}(E_{K1}(A) + S[2r + 2] - S[2r + 2])$
$A = D_{K1}(E_{K1}(A))$

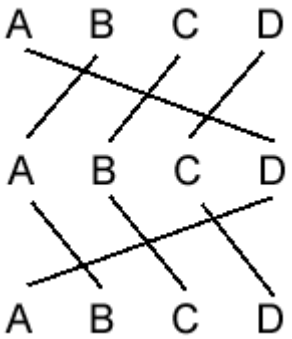
$C = D_{K2}((E_{K2}(C) + S[2r + 3]) - S[2r + 3])$
$C = D_{K2}(E_{K2}(C) + S[2r + 3] - S[2r + 3])$
$C = D_{K2}(E_{K2}(C))$

From here we may concern ourselves only with the remaining parts of the encryption and decryption functions.

The next operations to observe are the shifting of the registers. This occurs immediately prior to the final addition in the encryption function and immediately after the opening subtraction in the decryption function.

(A, B, C, D) = (B, C, D, A)
(A, B, C, D) = (D, A, B, C)

It is fairly easy to show that these two operations are inverses of one another. One can see that the first operation moves each register to the left in a circular manner and that the second moves each right in a circular manner.



So one may see that these register switching operations are also inverses.

At this point the operations become more interesting and some must be taken out of sequence. The first step in a round, whether it be an encryption or decryption round is to calculate two variables "u" and "t." This must be found first as they are used in the other operations for that round.

$t = (B \times (2B + 1)) \lll \lg w$
$u = (D \times (2D + 1)) \lll \lg w$

Note that these calculations are the same for the encryption and decryption functions. Also the values of "t" and "u" will be the same in either function. This is the case because no operations are done of "B" or "D" during any round other than the register exchange which occurs later in the encryption process and has already been undone in the decryption process.

The next set of operations make use of the values of "u" and "t" calculated in each round. In the encryption process, "A" is XORed with "t," then are circular shifted to the left by an amount defined by "u" and finally S[2i] is added to "A". A similar operation occurs for "C."

$A = ((A \oplus t) \lll u) + S[2i]$
$C = ((C \oplus u) \lll t) + S[2i + 1]$

In the decryption process, S[2i] is subtracted, "A" is circular shifted to the right by "u" and then XORed with t. Again a similar reversing process occurs for "C."

$A = ((A - S[2i]) \ggg u) \oplus t$
$C = ((C - S[2i + 1]) \ggg t) \oplus u$

This may be written in a function notation as:

$A = D_{K3}((((((E_{K3}(A) \oplus t) \lll u) + S[2i]) - S[2i]) \ggg u) \oplus t)$
$A = D_{K3}(((((E_{K3}(A) \oplus t) \lll u) + S[2i] - S[2i]) \ggg u) \oplus t)$
$A = D_{K3}((((E_{K3}(A) \oplus t) \lll u) \ggg u) \oplus t)$

$A = D_{K3}((E_{K3}(A) \oplus t) \oplus t)$
$A = D_{K3}(E_{K3}(A))$

The simplification goes similarly for "C." Additionally, no other operations occur on "A" or "C" during the previous parts of encryption or in the following parts of decryption so $A = D_{K3}(E_{K3}(A))$ simplifies to A = A, the intended result. The same goes for "C."

That operation completes the reversal of the round part of the function the only remaining part is the initial addition in the encryption and the final subtraction in the decryption.

The first operation that occurs during encryption is that addition of S[0] to "B" and S[1] to "D." Not surprisingly, that last operation that occurs in decryption is the subtraction of S[0] from "B" and S[1] from "D."

B = B + S[0]
D = D + S[1]

D = D – S[1]
B = B – S[0]

In the form of a function:

$B = D_{K1}((E_{K1}(B) + S[0]) – S[0])$
$B = D_{K1}(E_{K1}(B) + S[0] – S[0])$
$B = D_{K1}(E_{K1}(B))$

"D" simplifies in a similar manner. In this case D_{K1} and E_{K1}do not represent any other operations. So the equations really simplify to B = B and D = D.

So we can see that all four registers do end, after decryption, with the same data as they held before encryption. Additionally, we were able to demonstrate this by showing that each part of the encryption process was reversed by the corresponding part of the decryption process. Which proves that the decryption function is the inverse of the encryption function.