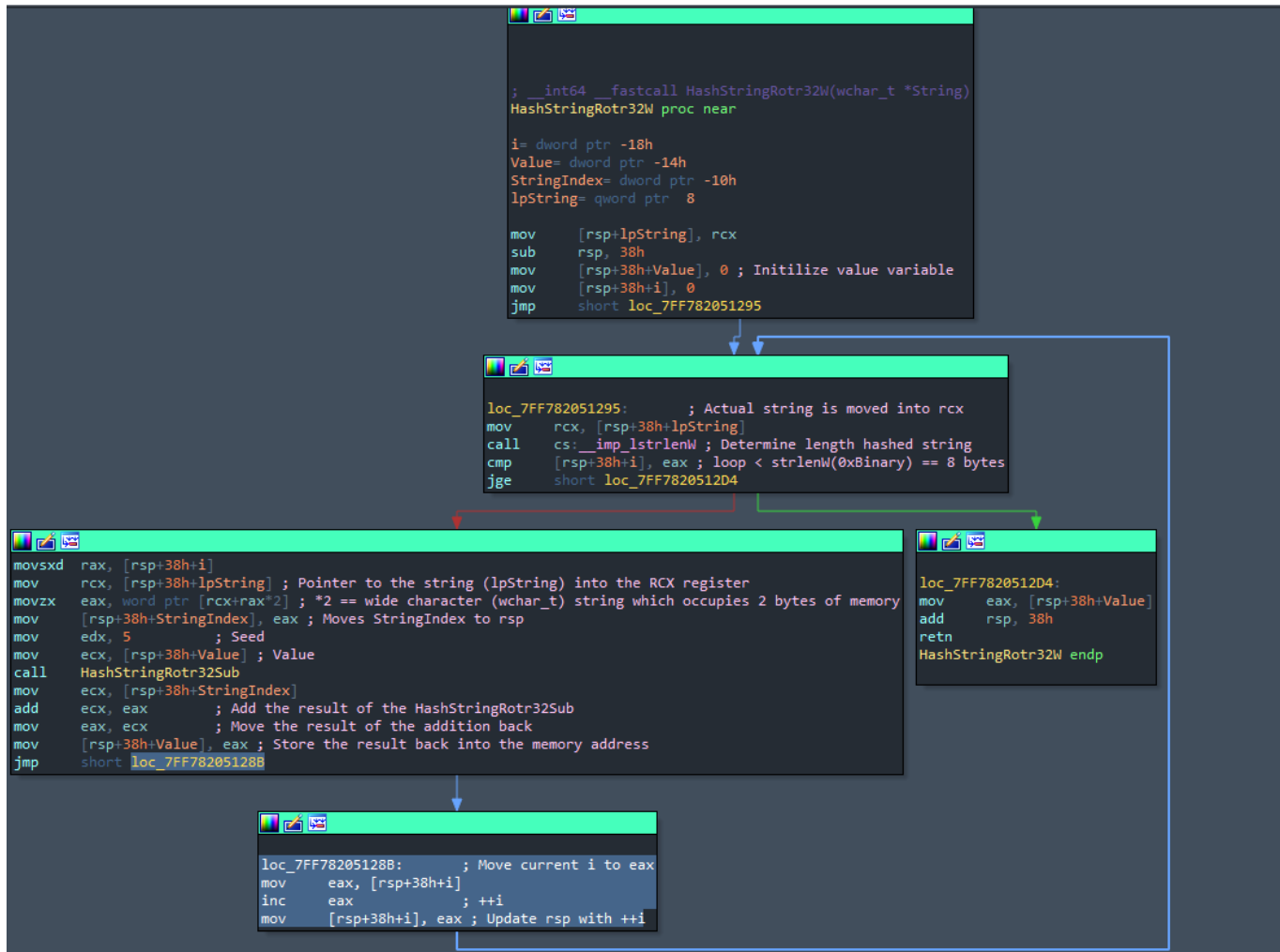


ROTR 32 Hashing Algorithm

ROTR 32 Subroutine

I explained in great detail how this subroutine works.



00007FFB2914E2D	CC	INC EAX	INC EAX
00007FFB2914E2E	CC	INC EAX	INC EAX
00007FFB2914E2F	CC	INC EAX	INC EAX
00007FFB2914E30	48 85C9	TEST RCX, RCX	rcx! "0xBinary"
00007FFB2914E31	74 1B	JZ kernelbase.7FFB2914E50	
00007FFB2914E32	48 C7C0 FFFFFFFF	MOV RAX, FFFFFFFF	
00007FFB2914E33	F14D 00	INC RAX	Increment till 8 bytes (0xBinary length)
00007FFB2914E34	48 FFC0	INC RAX	
00007FFB2914E35	90 02 04 4E	SHL WORD PTR DS:[RCX+RAX*2], 1	
00007FFB2914E36	75 F6	JNE kernelbase.7FFB2914E40	
00007FFB2914E37	EB 02	JMP kernelbase.7FFB2914E4C	
00007FFB2914E38	33C0	XOR EAX, EAX	
00007FFB2914E39	C3	RET	
00007FFB2914E3A	CC	INC EAX	
00007FFB2914E3B	33C0	XOR EAX, EAX	
00007FFB2914E3C	C3	RET	

HashStringRotr32Sub

This subroutine performs a rotation operation on the input **Value** by **Count** positions to the right while ensuring that the count stays within the valid bit range.

This code block rotates the bits of **Value** to the right by **Count** positions (**Value >> Count**). It then rotates the bits of **Value** to the left by the complement of **Count** (effectively rotating left by

32 - Count positions) and masks the result with Mask to ensure it stays within the valid bit range. Finally, it combines the two results using bitwise OR (|).

```
; __int64 __fastcall HashStringRotr32Sub(unsigned int Value, char Count)
HashStringRotr32Sub proc near

mask= dword ptr -18h
arg_0= dword ptr 8
arg_8= dword ptr 10h

mov     [rsp+16], edx
mov     [rsp+arg_0], ecx
sub     rsp, 18h
mov     [rsp+18h+mask], 31
mov     eax, [rsp+18h+mask]
mov     ecx, [rsp+18h+arg_8]
and     ecx, eax
mov     eax, ecx
mov     [rsp+18h+arg_8], eax
mov     eax, [rsp+18h+arg_8]
movzx   ecx, al
mov     eax, [rsp+18h+arg_0]
shr     eax, cl
mov     ecx, [rsp+18h+arg_8]
neg     ecx
and     ecx, [rsp+18h+mask]
mov     edx, [rsp+18h+arg_0]
shl     edx, cl
mov     ecx, edx
or      eax, ecx
add     rsp, 18h
retn
HashStringRotr32Sub endp
```