xLinka 13/10/2020
iPod Click wheel Games

iPod Games when Run load the game into ram, a key is generated from the .SINF file and compared to the iPods internal key(no clue how this is generated) but removing the function at 00136B84 (1.3 5.5 Enhanced) stops the game checking for further DRM protection allowing the use to run unsigned code.

Removing this and returning its own value by changing the references to 00 00 A0 E1 essentially removed the function out. After doing so the firmware checksums have to be recalculated then saved to file before flashing to the iPod or an error will occur.

The iPod games use an AES-128-CBC encryption algorithm, and the keys are loaded into the ram as a game is running.

When Tetris 1.0 was cracked the key was found at the following address:

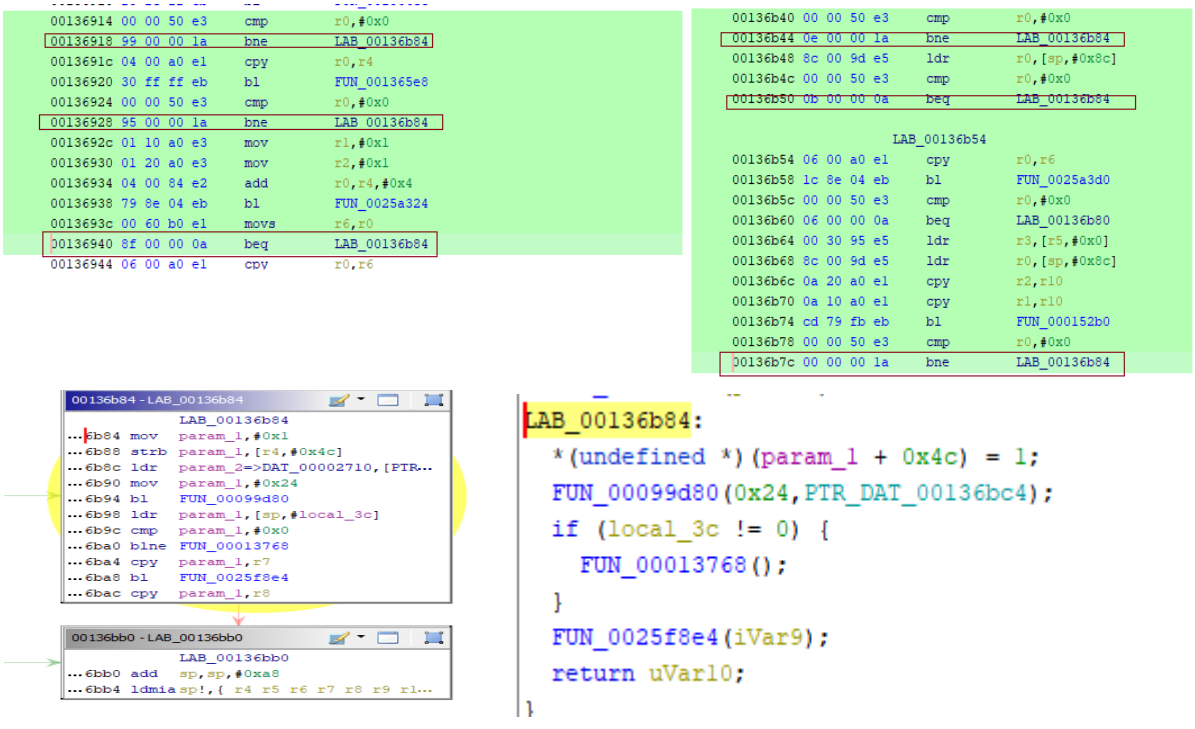
1. 13d05688 = KEY
2. 13b486cc (R10) = IV
   The key for this game was: 042DF36814CFC03B17716FC31538D767

We could assume the keys are loaded into the exact same place for the Keys and IV but not for certain.

The original 20 games used a modified OSOS to dump keys before crash to dump the memory of the iPod there's multiple ways to achieve this.

The key is unique per iPod so one key will not match another's game, so we need people who own the games and is able to sign them to iPod.

But as of 1.3 5.5 its safe to assume to key is calculated within 00136b84 as it loads other functions such as 00009d80 0x24 to a pointer data location at 00136bc4



xLinka 13/10/2020
iPod Click wheel Games

Classics/nanos:

As of this document the classic and nanos have not been patched but as of some full NAND DUMPS using Linux to dd of=/dev/sdbx of=ipod.iso

Games when synced to the iPod created a temp file within the iPod in `\iPod_Control\iTunes\Temp File.tmp`

But the file was long gone but accidently saved its data permanently to the NAND flash as shown even after formatting the device with a new pc and firmware flash the data remained no clue how this accident happed by using recover tools u can get back the TempFile.tmp but in this case all the game code was saved to the NAND and create a bin file from it – this was PACMAN on NANO3G