Tutoriel	**AmigaCracking : Arkanoid – Revenge of Doh**
Protection	**Single Track Protection - Copylock**
Auteur Original	**Rob**
Soumit par (sur flashtro.com)	**Rob [2005-01-16]**
Version	**17/02/2017 Gi@nts**
Vérification/Correction	**V2**

ARKANOID – REVENGE OF DOH
* CRACK TUTORIEL *

Table des matières

Matériels nécessaire

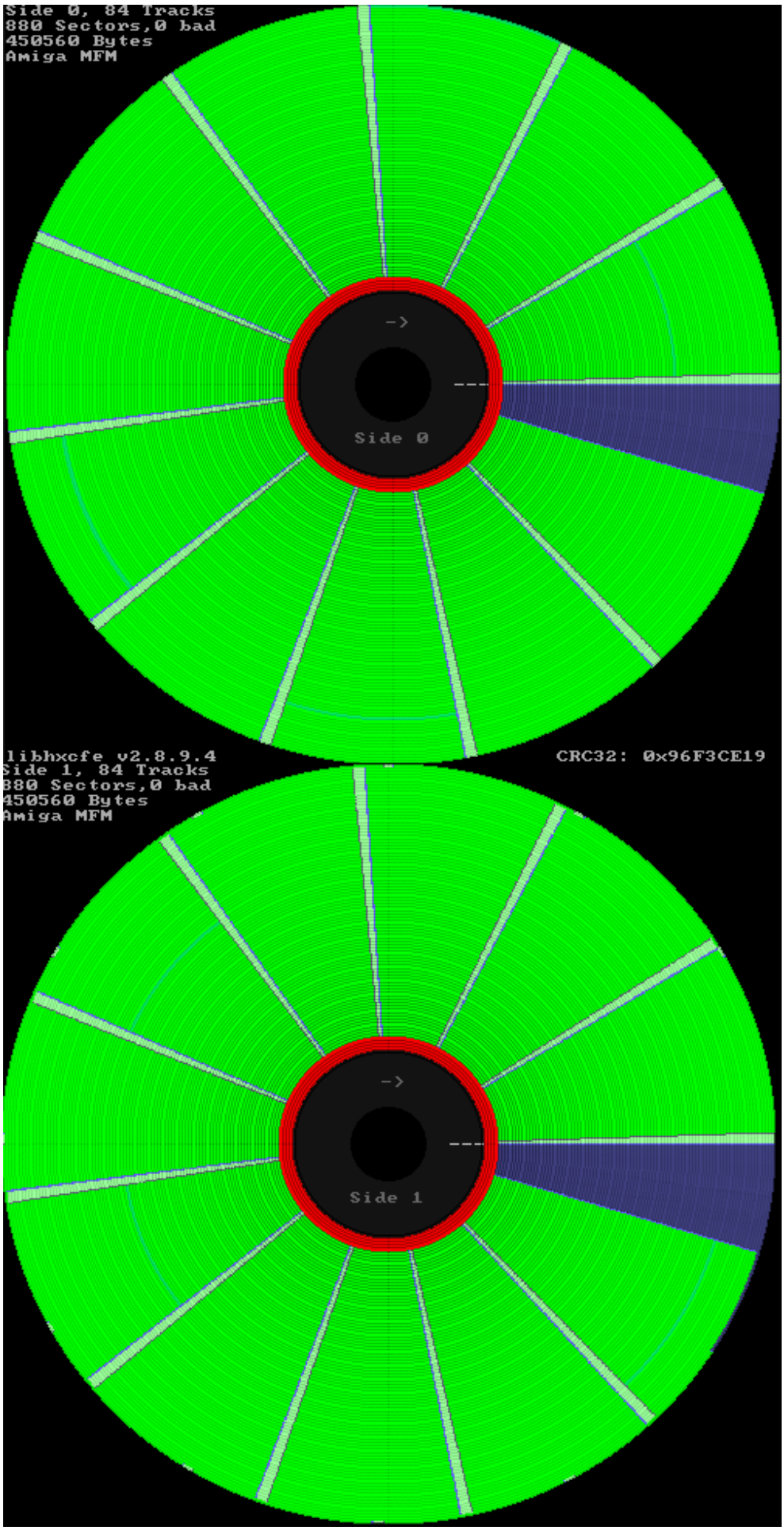
1) Un Amiga avec 512K (ou plus) ou l'émulateur WINUAE
2) Une Carte ACTION REPLAY MKIII (ou ça ROM Image)
3) Le jeu Original Arkanoid – Revenge Of Doh ou son image CAPS (SPS 0765)
4) Le logiciel Xcopy Pro en disquette ou image disk.

Général Info

Ce tutoriel Français est basé sur le tutoriel original de Rob.
Ce document n'est pas une traduction mot par mot de celui-ci mais plus une nouvelle version.
Suivit pas à pas avec des nouvelles informations.

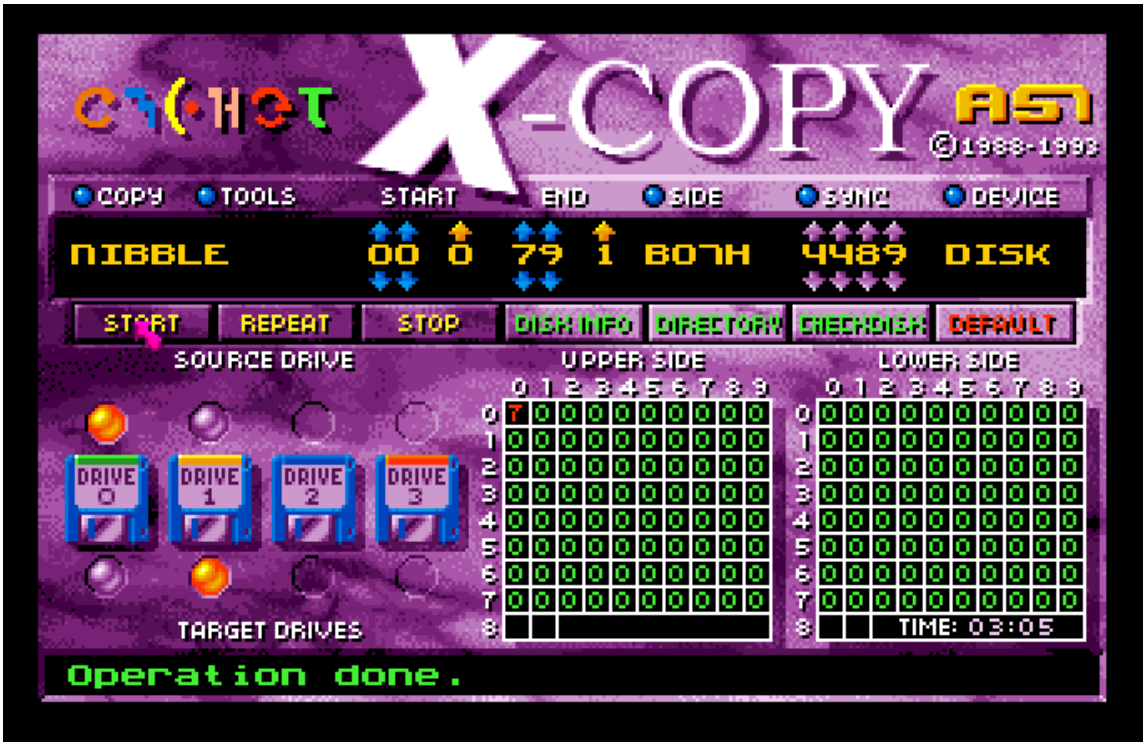
Un grand merci à **WayneK** sur le forum de **flashtro.com** qui m'a bien aidé sur les questio**ns que je me posais et du coup a** permis ce tuto.

Bon Tuto. **Gi@nts**


Side 0, 84 Tracks
880 Sectors,0 bad
450560 Bytes
Amiga MFM
->
Side 0
libhxcfe v2.8.9.4
Side 1, 84 Tracks
880 Sectors,0 bad
450560 Bytes
Amiga MFM
CRC32: 0x96F3CE19
->
Side 1

Au travail

Comme dans Tous bon hack qui se respecte, on va commencer par essayer de copier le disk original.
Démarrer votre logiciel de copie préféré, à savoir **Xcopy Pro**
Choisissez le mode **NIBBLE, insérez** une disquette **vierge** en **DF1** et la disquette du **jeu original** en **DF0**
Lancer la copie



Hummm, cela ressemble à une protection '**copylock**'

Bien sûr, cette copie ne fonctionnera pas, très vite elle fera crasher votre Amiga.
Mais gardons quand même sous le coude ce ***backup.***

Petit rappel des codes d'erreur de Xcopy :

1. *Less or more than 11 sectors*
2. *No sync found*
3. *No sync after gap found*
4. *Header checksum error*
5. *Error in header/format long*
6. *Data block checksum error*
7. *Long track*
8. *Verify error*

Insérez la disquette originale du jeu dans le lecteur de l'Amiga et démarrer dessus.

Au bout des quelques secondes de chargement.
Entrer dans votre **AR** et allons regarder si on trouve notre **'copylock'** en mémoire :
Taper `F 48 7A`

Aucun résultat.... Humm

On va faire autrement, à savoir : charger le **bootblock** en mémoire et regarder ça de plus près :

#RT alias Read Track, permet le chargement de la track 0 à 1 (1er piste de la face 0)
#D, alias Désassemble

Taper : `rt 0 1 10000` puis `d 10000`

```
d 10000
~010000 NEG.W   A7
~010002 SUBQ.B  #1,D0
~010004 CHK     A5,D4
~010006 MOVE.W  USP,-(A6)
~010008 ORI.B   #70,D0
~01000C MOVEM.L D0-D1/A0-A2,-(A7)
~010010 MOVE.L  #E00,D0
~010016 MOVE.L  #10002,D1
~01001C MOVEA.L 00000004.S,A6
~010020 JSR     -C6(A6)
~010024 TST.L   D0
~010026 BEQ     00010010
~010028 MOVE.L  D0,10(A7)
~01002C MOVEA.L C(A7),A1
~010030 MOVE.L  10(A7),D0
~010034 MOVE.L  D0,28(A1)
~010038 MOVE.L  #E00,24(A1)
~010040 MOVE.L  #400,2C(A1)
~010048 MOVE.W  #2,1C(A1)
~01004E MOVEA.L 00000004.S,A6
~010052 JSR     -1C8(A6)
~010056 MOVEA.L C(A7),A1
~01005A MOVE.B  1F(A1),D0
~01005E BNE     00010030
~010060 MOVEM.L (A7)+,D0-D1/A0-A1
~010064 RTS
;======================================
```

Il semblerait qu'on ait notre petite routine de déplacement de donnée ici :

010034 MOVE.L D0,28(A1) <== Adresse de destination en **A1**
010038 MOVE.L #E00,24(A1) <== Nbr de donnée a copier : **$E00**
010048 MOVE.L #400.2C(A1) <== Adresse source de lecture : **$400**

Cela va donc copier **$E00** de donnée à partir de **$400** vers l'adresse contenue dans **D0**

Voyons voir de plus près ce qui se trouve en ce moment à cette adresse.
#M alias memory read, permet de voir les données en mémoire.
Taper `n 103F0`

```
M 103f0
:0103F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
:010400 60 00 00 7C 50 72 6F 74 65 63 74 69 6F 6E 20 28 `..|Protection (
:010410 43 29 43 6F 70 79 72 69 67 68 74 20 31 39 38 38 C)Copyright 1988
:010420 20 52 6F 62 20 4E 6F 72 74 68 65 6E 20 43 6F 6D  Rob Northen Com
:010430 70 75 74 69 6E 67 2E 20 41 6C 6C 20 52 69 67 68 puting. All Righ
:010440 74 73 20 52 65 73 65 72 76 65 64 2E 00 00 00 00 ts Reserved.....
:010450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
:010460 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF ................
```

Pourtant... c'est bien du **'copylock'**, le code doit être nettoyé après chargement sinon, on l'aurai trouvé.

On va modifier notre petit '**bootblock**' pour éviter le **RTS** en **$10064** histoire d'en savoir plus.
Taper :

#BRA, Instruction du 68000 qui permet de se brancher à l'adresse indiqué, ici une boucle sur nous-même.
#BOOTCHK, permet de calculer le checksum d'un bootblock en mémoire
#WT, alias Write Track, permet d'écrire une zone mémoire sur la disquette à l'adresse indiqué en cylindre.

 # On modifie le code à partir de 10064
`A 10064`

#On change le RTS par un BRA sur soit même, la fameuse boucle infini.
`BRA 10064` [**ENTRER**] puis [**ESCAPE**]

#On calcule le nouveau checksum de ce bootblock
`BOOTCHK 10000`

#On sauve le tout sur disquette
Insérez la disquette de **backup** préalablement crée et taper :
`WT 0 1 10000`

Effectuer un **reboot** de votre Amiga tout en laissant la disquette de backup dans celui-ci.

Après un court chargement (celui du **bootblock**), l'Amiga semble ne plus rien faire.
Il est entré dans notre boucle sans fin.

Entrer dans votre **AR**
Taper : `D`

~0015BC BRA 000015BC

On est bien dans notre boucle sans fin, impeccable.
On va voir si maintenant on arrive à trouver notre petite signature de '***copylock***' en mémoire.
Taper `F 48 7A`

```
d
~0015BC BRA     000015BC

f 48 7a
Search from: 000000 to: C80000
005A70  _
```

Bingo, trouver !

On jette un petit coup d'œil en mémoire à l'adresse indiquer :
Taper `n 5A70`

```
n 5a70
.005A70 Hz..#ß... @..| .].\.#.... ?.NhC..X".L.......A...H...A..d#.....Jü
.005AB0 ü..[...[...[...$..'=.A'.'.A.+...=..'... '..... .....4...0'... '/'../.\.§
.005AF0 .X.q.....F.............Ö..H...A..4#....$A...#.... ........../....
.005B30 .o....C..&g. Q ...'&.|..H...C... Q ... o.."#P.. (.üF.H@..Lß..Ns
.005B70 ..........................D.ß.o...t!..\j.....$..,...Ye...l..r.ü.
.005BB0 ..ü.ü....E....L.....\...)Ö..ovÖ..ö..ö..?Fc..f.Cw.Bf.ö....n(o...=
.005BF0 8..*.$.....6.pK..K~.cK8...Ö/.6..... ....}{.b.....ü....J.bä.. cü_
```

Maintenant on va tous simplement continuer l'exécution du **bootblock**
Cela passe bien sûr par la suppression de notre modification.

Taper `A 15BC` puis `RTS` [**ENTRER**] puis [**ESCAPE**]

Et avant de retourner à l'exécution de notre code, on **remplace** la **disquette de backup** par la **disquette originale** de notre **jeu** dans le lecteur Amiga, puis :

#X permet le retour au code Amiga en court.
Taper `X`

```
a 15bc
^0015BC RTS
^0015BE

X_
```

Le code continue son chemin et charge les données du jeu.

Assez rapidement, une image du jeu est affichée.

Entrer dans votre **AR**
On va aller voir si les données ont changé à l'adresse **$5A70**
Taper `n 5A70`

```
.005A70 Hz..#ß... @..I .].\.#.... ?.NhC..X".L.......A...H...A..d#.....Jü
.005AB0 ü..[...[...[...$..'=.A'..A.+...=..`...`.....4...0'...'/'../.\.§
.005AF0 .X.q.....F.............Ö..H...A..4#....$A...#.... ........../....
.005B30 .o....C..&g. Q ...'&.I..H...C... Q ... o.."#P.. (.üF.H@..Lß..Ns
.005B70 ..........................D.ß.o...t!..\j.....$.;...Ye...l..r.ü.
.005BB0 ..ü.ü....E....L.....\...)Ö..oyÖ..ö..ö..?Fc..f.Cw.Bf.Ö....n(o...=
.005BF0 8..*.$.....6.pK..K~.cK8...Ö/.6..... ....}(.b.....ü....J.bä.. cü_


n 5A70
.005A70 ..3ü...ß..By.ß..3üB..ß.. I..▒.0(.."I.ß..2.Q..ü"M#I.....(#I.....$
.005AB0 #I.....3I....,x..N..8"M.)..f.3I....#I.....$,x..N..83ü...ß..3ü.
.005AF0 .ß..#..X...13ü. .ß..0(.. I.ß..BXQ..üN......9...ß..g(#ü.....ß..#ü
.005B30 ...@.ß.ä#ü.....ß..#ü.....ß..N..ü.............................
.005B70 ........................................................................
.005BB0 ........................................................................
.005BF0 ........................................................................
_
```

Il semblerait qu'il y ait eu du changement.
Le '***copylock***' a sûrement décodé les données.

Taper `D 5A70`

Remonter au début de cette 'partie' de code, à savoir : **$59EE**

Le code juste au-dessus nous envoie en **$59E0**, qui nous renvoie rapidement en **$59F4**

```
~0059DE TST.L   -5556(A2)
~0059E2 LINEA
~0059E4 ORI.B   #0,D0
~0059E8 BRA     000059F4
;=====================================
~0059EA SUBQ.B  #1,D0
~0059EC BRA     000059E0
;=====================================
~0059EE LINEF
~0059F0 ORI.B   #70,D0
~0059F4 MOVEA.L A1,A5
~0059F6 MOVE.L  #77FDE,28(A1)
~0059FE MOVE.L  #7E00,24(A1)
~005A06 MOVE.L  #2C00,2C(A1)
~005A0E MOVE.W  #2,1C(A1)
~005A14 MOVEA.L 00000004.S,A6
~005A18 JSR     -1C8(A6)
~005A1C MOVEA.L A5,A1
~005A1E MOVE.B  1F(A1),D0
~005A22 BNE     000059F6
~005A24 MOVE.W  #A0,00DFF096
~005A2C MOVE.W  #20,00DFF09A
~005A34 MOVE.W  #8100,00DFF096
~005A3C LEA     5B4C(PC),A0
~005A40 MOVE.L  0000006C,2(A0)
```

$59E8 semble être une bonne adresse de départ, beaucoup de chose se passe tout de suite après cette adresse.
Allons regarder ce qu'il y a exactement en mémoire à cet endroit.
Taper `N 59E8`

```
n 59e8
.0059E8 '.S.'..C...p*I#|..▒..(#|..~..$#|.,.,.,3|....,x..N..8"M.)..f.3ü..
.005A28 .β..3ü. .β..3ü...β..A...!y...|..A...#....13ü. .β..3ü<..β..3ü...β
.005A68 ..3ü.0.β..3ü...β..By.β..3üB..β.. |..▒.0<.."|.β..2.Q..ü"M#|.....(
.005AA8 #|.....$#|.....,3|....,x..N..8"M.)..f.3|....#|.....$,x..N..83ü..
.005AE8 .β..3ü. .β..#..X...13ü. .β..0<.. |.β..BXQ..üN.....9...β..g(#ü..
.005B28 ...β..#ü...@.β.ä#ü.....β..#ü.....β..N..ü.....................
.005B68 ........................................................................
```

Le code semble se terminer vers **$5B68**

Comme on a pu le remarquer plus haut, dans le code du '***bootblock***', des données étaient accédé à l'adresse **$400**
En fait, dans le cas d'une protection '***copylock***', cet adresse mémoire sert de tampon a ce qui est décodé.
En l'occurrence ici, la track 0

Maintenant qu'on a ses données décodées en mémoire, on va tout simplement les déplacer ses données décryptées vers cette adresse et réécrire les données sur disque.

Taper :
#On lie et charge en mémoire notre bootblock
`RT 0 2 70000`

#On transfère nos datas décryptée vers la zone tampon en question (70000+400)
`TRANS 59E8 5B68 70400`

Remplace la **disquette du jeu original** par la **disquette de BACKUP** dans le lecteur Amiga, puis :

Taper :
#Et on sauve le tout sur disquette,
`WT 0 2 70000`

Maintenant je le jeu devrait être fonctionnel et surtout copiable simplement avec xcopy
Redémarrer votre Amiga et apprécier ce superbe jeu !