Introduction to Armadillo Key Tool

Abstract
This paper is meant to explain how to use Armadillo Key Tool in it's full potential. It's good to be at least a little familiar with Armadillo's registration system, but it is not necessarily needed.

Background
In Armadillo, you have (as of now) four registration schemes:

1. *Unsigned*
2. *Signed V2*
3. *Signed V3*
4. *Short V3*

Unsigned keys do not have a digital signature and their *keybytes* (actual key data) is encrypted with Blowfish. The keys are decrypted using the name of the user and they are the only key type that support hardware transfer. Unsigned keys only support one level and can only store one 2 bytes of *otherinfo*.

Signed V2 keys use the ElGamal signature system. The keybytes are protected with a (relatively) simple XOR-encryption, based on the name of the user. Unlike unsigned keys this type of key has four levels (1 – 4). Good to know is that the signature is not encrypted using the name. Signed V2 keys can store 10 bytes of otherinfo when needed.

Signed V3 keys are basically the same as Signed V2 keys. The only actual difference is that there are 9 levels (1 – 9).

Short V3 keys are recommended to use by SiliconRealms because they support *nameless keys* and *keystrings*. Levels 1 – 9 are (just like Signed V2 & V3) signed with the ElGamal signature scheme. Level 10 is digitally signed using Elliptic Curve Cryptography (ECC) and this type of signature is harder to break then the ElGamal variant. Another big difference is that Short V3 keys are Base32 encoded.

Keybytes are bytes of data that contain information about the key. They are encrypted with the username the key was made for and a digital signature scheme is used to prevent *modification* of names and other info in keys.

Otherinfo is data that the program uses after it's extracted from the key. It can be really used for anything (expiration dates, expiration version, number of copies, extrainfo, etc). The key system has support for five WORDS of data, but in practice the fifth WORD is never actually passed to the program (you have a maximum of 32 EXTRAINFO bits).

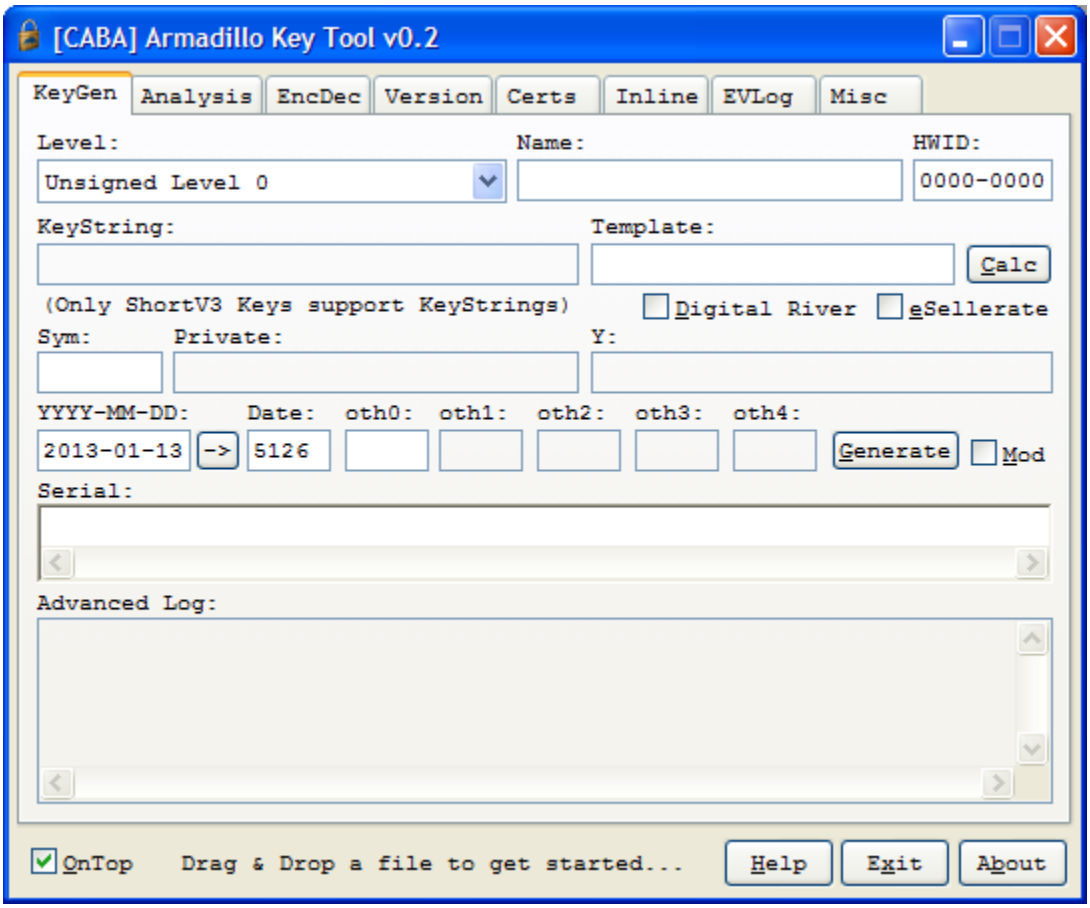
Nameless keys are keys that can be installed without entering a name. The term 'nameless' is actually wrong, because the name is simply stored in the key instead of being entered by the user. Nameless keys can easily be identified by the fact that (dashes and prepended zeroes removed) nameless keys always start with a '2'. The only keytype that supports nameless keys is Short V3. This is because of the Base32 encoding. Signed V2 & V3 keys are simply HEX bytes with dashes, meaning that prepending a '2' would totally change the key.

Keystrings are strings that can be embedded into the serial. They are passed to the KEYSTRING environment variable for use of the user. The maximum length of a keystring is 255 bytes (+1 for the size of the string). Keystrings are placed right after the symmetric key and otherinfo of the

keybytes in reversed order. The length is placed just before the signature. In real life we cannot use keystrings much longer then 100 characters, Armadillo will get a buffer overflow because the allocated buffer was too small...

Modification keys are any type of keys with the raw date value set to 0xFF40, 0xFF20 or 0xFF10 they are used to for example expand trails without giving out a completely new valid key.

Symmetric key this is a DWORD, based on an encryption template. It is used to decrypt certain parts of packed programs and it is the most important part of a key, actually you cannot create a valid key that runs the program without it.



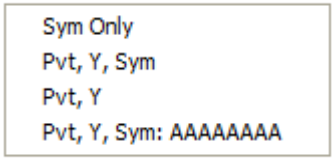
The KeyGen tab

Tab 1: KeyGen

As the name already says: the tab that can generate keys. In here you can select the key level (Unsigned – Short V3 Level 10), set the name to generate keys for (you can leave it empty with Short V3 keys), set the hardware ID of the user that installs the key, set the symmetric key, set the keystring (in case of Short V3 again), set the key creation date and set the otherinfo parameters.

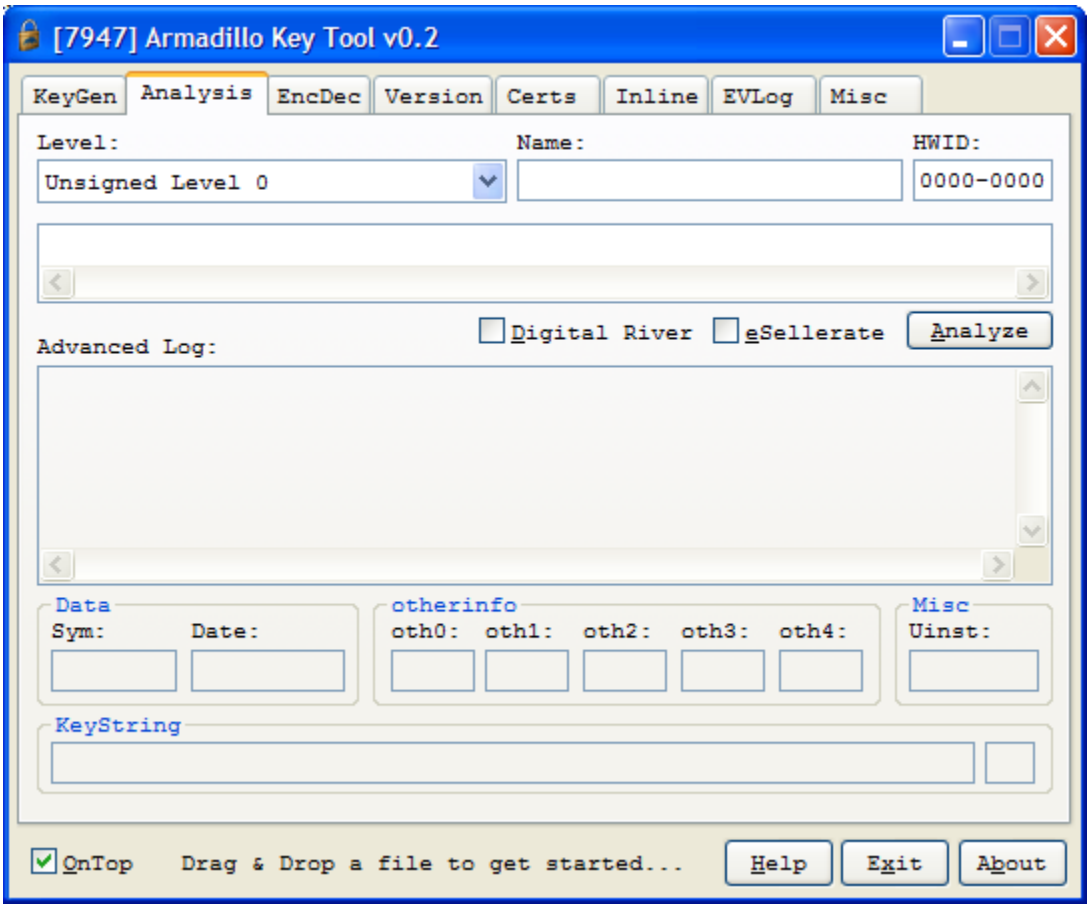
As you can see the GUI should be quite self-explanatory (enter values to use and click Generate to make keys), there are just a few things that you should know to properly use this tab.

The 'Calc' button is used to generate data from an encryption template. You should know that clicking the 'Generate' button NEVER uses the string provided in 'Template'. You can use the 'Calc' button to generate values from an encryption template and put them in the needed text boxes. Another thing you should know about this button is that clicking it opens a menu. The first option (Sym Only) only calculates the symmetric key from the encryption



template and leaves the rest of the values as they are. The second option calculates all possible values that can be calculated from an encryption template. The third option calculates the details needed for the signature, but leaves the symmetric intact and the last option (not always present) used the symmetric from the clipboard and calculates the values needed for signing keys from the encryption template.

Another (and last) thing you should know is that using the Digital River and eSellerate options should only be used when use symmetric keys generated from an encryption template. Brute forced symmetics already dealt with this protection layer and it is therefore not needed to check one of the boxes.



The Analysis tab...

Tab 2: Analysis

Also this tab should be quite self-explanatory. It is used to retrieve all possible data from keys with a name/key/hardware ID (not always the case) combination. Simply enter your key, the name the ***key*** (not your name) was generated for and the hardware ID (when applicable).

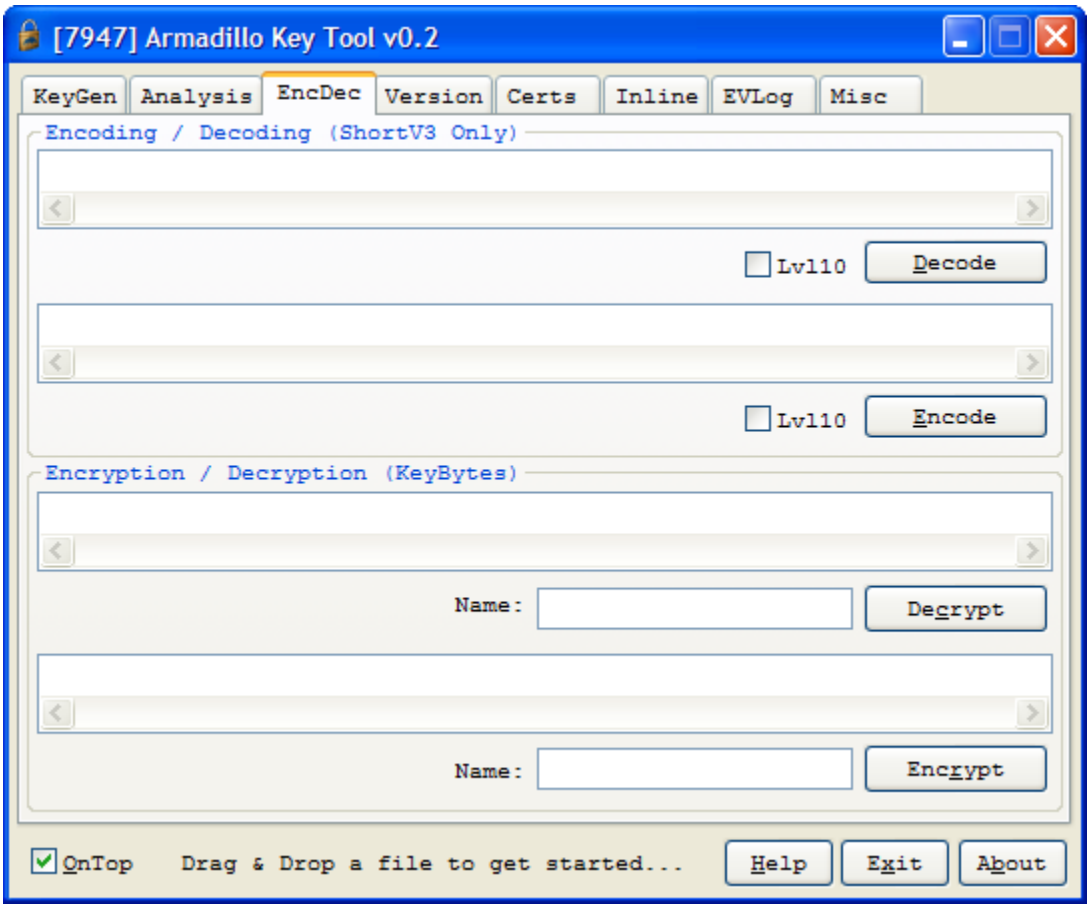
Again: avoid clicking the Digital River or eSellerate checkboxes. They are only present for completeness, not because their useful.

Tab 3: EncDec

This time the name is not so obvious as you might suspect. This tab was intentionally used for debugging only, but it helped me so much that I decided to keep it.

The controls are quite simple. Enter a Base32 (Short V3) encoded key in the first text box and hit Decode to decode it to it's original byte form. Click the Lvl10 checkbox if the key is Level 10.
The same goes for encoding Short V3 keys. Enter the keybytes you want to encode and hit Encode (and don't forget to check the Lvl10 checkbox if you're dealing with Level 10 keys).

The decryption part of the tab is used to decrypt keybytes so you can analyse them with your favourite text editor. Enter keybytes to encrypt/decrypt in their text boxes, enter the name you want to En/Decrypt with and hit En/Decrypt.



The EncDec tab...

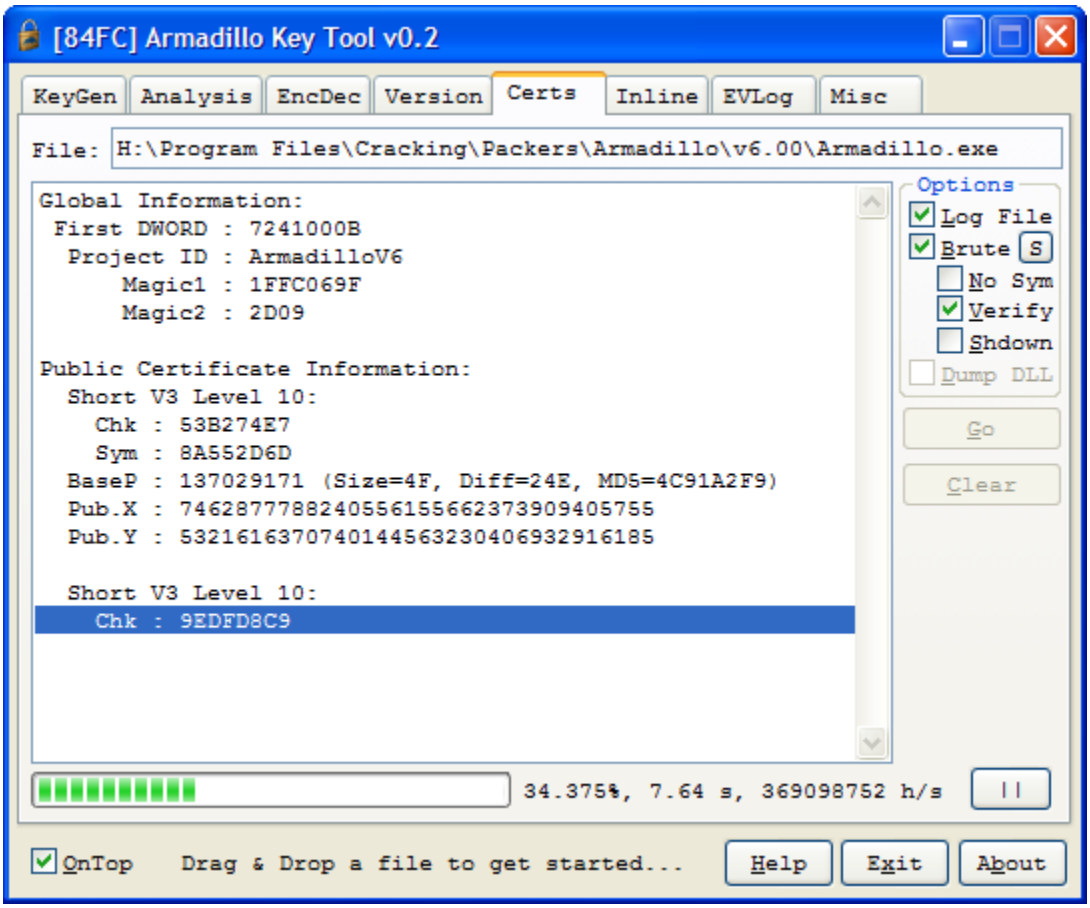
Tab 4: Version
For this tab everything is obvious. You drag & drop a file and all the needed version and protection info will be given to you. The special thing about this tab is that is can retrieve certain Other options that ArmaFP cannot detect.

Tab 5: Certs
This is what I personally consider the most important tab of all. It is used to retrieve public key information and symmetric key checksums from protected files. Simply drag & drop a file and hit 'Go' to start retrieving this info. Another good thing about this tab is that you can brute force symmetric keys of signed keys (unsigned keys do not have a checksum). Simply hit the 'Brute' checkbox and also the 'Verify' checkbox for the most optimized bruting process. Notice that the 'Verify' checkbox will only work for Armadillo v5.00 and higher, older versions are not supported. While bruting there is a progress bar and a pause button and you can also let the tool shut down the computer when the bruting process is finished (Armadillo v7.40 and higher).

If you only want to solve the ElGamal parameters you should check the 'No Sym' checkbox. This is useful if you (for example) already have a valid key and don't want to waste your time on brute forcing it all over again.

In the near future the listbox the data is displayed in will support right click options. This is useful if you want (for example) to brute force or copy a certain part of the public certificates.

Another options that isn't enabled yet, but is on the To Do list is the 'Dump DLL' option. It will be used to dump the internal security.dll out of the program. Replacing the DLL is **not** on the To Do list.



The Certs tab in action. You can nicely see that the bruting process is stopped when a valid symmetric is found, this is because of the Verify checkbox.

Tab 6: Inline

This tab is also used quite often. The usage is quite simple: you drag and drop an (unmodified) Armadillo protected EXE to retrieve CRC values and other needed data. Then you click 'Inline' and the tool should ask you where to save the modified EXE. After it's saved you open Olly with the Multimate Assembler plugin. On your clipboard is the skeleton for the inline. If you accidentally removed this data from your clipboard: don't worry, the 'Copy' button will be glad to provide it to your clipboard again.

Now what to do with this skeleton? It's actually quite easy: patching security DLL or parts of the program. The place that says: “Place your code after this, security base is in [REGISTER]” can be used to patch security dll. It's upto your imagination what you do with it (patching ECDSA verification routines, injection environment variables, patching hardware ID, re-enabling the REGISTER option and other commandline options etc, etc.

What does the 'Plugins' button do? Also this is quite obvious, you can create DLLs that can find data in security.dll (a memory copy of this is provided so you can search patterns in it). Plugins can be useful because they save a lot of time when creating (for example) hooks or patching a lot of data.

Tab 7: EVLog

This is the quickest to explain tab. You drag and drop a file and you get all set environment

variables. Right click the list to copy (parts) of the list entries and you can click the 'Dump' button to save a full list of all set variables.

Tab 8: Misc
Currently this is the tab with the most options:

Selected File shows the currently dropped file.

Current Symmetric can retrieve the currently used symmetric key from the selected file. This is useful if you don't remember your key or if the program has a strange HWID behaviour (like RadioBoss).

Generate Checksum can be used in various ways. First you can simply generate (salted) checksums from a symmetric key and secondly it can look if the checksum exists in the currently selected file (click the Find button and if the checkbox gets checked you're good to go). This is really useful in Armadillo v7.40 and above because salted checksums are not same when the symmetric is the same. Obviously you don't have to retrieve the salt manually, this can be done by clicking the small 'R' button.

VerifySym is a really useful feature of this tab. You can (1) give it a list of valid symmetrics and let it check them all, (2) verify one symmetric, (3) retrieve the needed values and (4, not yet implemented) check if a symmetric is valid for a certain executable (hit the 'A' checkbox). Important to know is that you need a _certs.bin file. This file is dumped by the Certs tab (Armadillo v5.00 and higher).

ArmaSectionDeleter is another useful feature of this tool. It is comparable with 'Armadillo Reducer' from FOFF team. The only difference is that it's sometimes more, sometimes less buggy. Use is quite obvious, hit Get to analyse a file, then hit Del to delete sections (The tool will try to automatically detect sections to remove, but this is not fail-proof, therefore watch out).

Date Tool is a small tool that can be used to convert dates from and to Armadillo format. The use if obvious, but you should know that dates are considered decimal, not hex.

FixClock Key is another useful feature, it can retrieve the Project ID of a file and generate FixClock keys from it (useful if you need more trial time for example).

License Removal is not yet implemented and subject for removal. It was supposed to delete registry entries.

Final words
It's is good to know one thing before you start using this tool: it will not improve your reversing skill.
Another thing you should know is that there are some 'hidden' GUI features. The OnTop checkbox can be useful and another thing is that dragging & dropping a file in a tab that does not support file dragging & dropping will show you a small menu with possible drop locations you can click so it's dropped on a good tab. This means that you do **not** have to switch tabs the first time you run the tool, simply drag & drop a file.