[PV204] Profile code performancence - SatoChipApplet

Martin Knotek, Lenka Svetlovská, Jiří Týma

1 Introduction

In this document we will try to summerize the work we have done on our SatoChip applet, it's iprovements and performance measurements.

2 Code improvements

- Created a set of unit tests, based on already present tests, which were however in a different project. This is a major update, since the test can be easily run with Gradle, debugged and extended.
- Moved a lot of code from *Setup()* method to constructor, variable guarding the *Setup()* transfered to *byte*.
- Investigated a few function we were not sure about their details, like the *getData()* checking the lenght of incoming data or *keyImport()* funciton and how it transferes and stores keys.

3 Code performance

We measured all the instructions we had tests for using the time measurement available in the *transmit()* method in *CardManager*. On top of that we measured three chosen functions using *JCProfiler* to get a more precise measurement. See figure 1 for a table containing all the measurements.

Category	Instruction	Byte	Tested	Time (ms)	JCProfiler time (ms)
Applet initialization	SETUP	0x2A	y	11 (already done), 618 (if first setup)	
Keys' use and management	IMPORT_KEY	0x32	y	55-124	49
	GET_PUBLIC_FROM_PRIVATE	0x35	y	78	
External authentication	CREATE_PIN	0x40	y	227	
	VERIFY_PIN	0x42	y	28	
	CHANGE_PIN	0x44	y	47	
Objects' use and management	CREATE_OBJ	0x5A	y	0-1	
	DELETE_OBJ	0x52	y	1	
	WRITE_OBJ	0x54	y	1	
Status information	GET_STATUS	0x3C	y	15	
HD wallet	BIP32_IMPORT_SEED	0x6C	y	9452	9404
	BIP32_GET_AUTHENTIKEY	0x73	y	329	
	SIGN_SHORT_MESSAGE	0x72	y	16 (empty), 180 (non-empty)	non empty 155

Figure 1: Performance measurements results.