Smartcard standards <https://www.cl.cam.ac.uk/~mgk25/project-ideas/smartcard-standards/>

<https://www.youtube.com/watch?v=IhButZJIdcg>

<https://www.youtube.com/watch?v=QJyxUvMGLr0>

Generic smartcard info including price

<http://www.cardwerk.com/smartcards/smartcard_technology.aspx>

MIFARE classic flaws:

<https://link.springer.com/content/pdf/10.1007%2F978-3-540-85893-5_20.pdf>

<https://www.cs.bham.ac.uk/~garciaf/publications/Security_Flaw_in_MIFARE_Classic.pdf>

Java Card sdk release notes: <https://docs.oracle.com/javacard/3.0.5/devnotes/index.html>

Java Card SDK specifications: <https://docs.oracle.com/javacard/3.0.5/JCSRN/JCSRN.pdf>

Java Card SDK technical documents <http://www.oracle.com/technetwork/java/embedded/javacard/documentation/javacard-docs-1970421.html>

Dev kit user guide: <https://docs.oracle.com/javacard/3.0.5/guide/running_the_samples_in_eclipse.htm#JCUGC-GUID-5499F923-FDCC-4A55-AC2C-9AE11B33FD73>

1 hour presentation on Java Card development. Good comments on coding practices. <https://www.youtube.com/watch?v=khgT5dwKvOo>

<https://www.openscdp.org/ocf/PGuide.pdf>

OpenCard demos <http://opencard.cvs.sourceforge.net/viewvc/opencard/apps/stockbroker-demo/src/demos/samples/>

Full description of how to develop javacard applet <http://www.oracle.com/technetwork/java/javacard/applet-136808.html>

Claim of PCSC-OCF bridge - <https://books.google.co.uk/books?id=AUMhjfVezWkC&pg=PT314&lpg=PT314&dq=opencard+unix&source=bl&ots=B6OIMRO2b-&sig=PXhujoEENwoL2dUIcFtiDhTx-P8&hl=en&sa=X&ved=0ahUKEwiNlNv3raXXAhWE2RoKHbf1CzoQ6AEINTAD#v=onepage&q=opencard%20unix&f=false>

Claim to the contrary and that they should be compatible <https://community.oracle.com/thread/1755737?start=0&tstart=0>

Review of SCL3711 data sheet only mentions PC/SC

Other person with pyscard problem I had <https://stackoverflow.com/questions/31140601/establishcontextexception-error-in-python>

JC garbage management <https://askra.de/software/jcdocs/app-notes-2.2.2/garbagecoll.html>

A clear view of opacity <https://www.secureidnews.com/news-item/a-clear-view-of-opacity/>

Opacity specifications <https://www.securetechalliance.org/resources/pdf/OPACITY_Protocol_3.7.pdf>

Some standards. Look at 5.8.1. <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf>

Python asn1 docs <https://media.readthedocs.org/pdf/python-asn1/latest/python-asn1.pdf>

(also get other BER sources e.g. Wikipedia)

EC curve used (secp256r1) defined p15 here <http://www.secg.org/SEC2-Ver-1.0.pdf>

CMAC implementation I used as a reference for my own <https://pastebin.com/JQ9xQ5vK>

CMAC test values: <https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Standards-and-Guidelines/documents/examples/AES_CMAC.pdf> (btw typo in example 1, block #0, inBlock should equal k2)

JCMathLib GitHub <https://github.com/OpenCryptoProject/JCMathLib>

Modular inverse Java implementation <http://www.geeksforgeeks.org/multiplicative-inverse-under-modulo-m/>

Based my EC Point multiplication on this <https://stackoverflow.com/questions/15727147/scalar-multiplication-of-point-over-elliptic-curve>

Egcd implementation <https://en.wikibooks.org/wiki/Algorithm_Implementation/Mathematics/Extended_Euclidean_algorithm>