Structure of an NFC tag

```
SECTOR 0:
           [  UID  ] [ATQASAK] [ free space      ]
  0000000: 0102 0304 c508 0400 6263 6465 6667 6869  ........bcdefghi
  0000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
  0000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
  0000030: ffff ffff ffff ff07 8069 d3f7 d3f7 d3f7  .........i......
           [ key A      ] [access ] [ key B      ]
SECTOR {1..15}:
  0000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
  0000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
  0000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
  0000070: ffff ffff ffff ff07 8069 d3f7 d3f7 d3f7  .........i......
           [ key A      ] [access ] [ key B      ]
```

DarkSide of the Chip (or: god dammit libnfc)

So you want to crack the unknown keys of your Mifare tag? Awesome, this will explain how. This only applies to the Mifare Classic 1K/4K.

Hopefully whoever programmed the key left some defaults, which means we can skip `mfcuk` and go right to `mfoc`:

```
# Mifare Classic 1K, key A|B, store keys to memory
proxmark> hf mf chk *1 ? t
```

If it found any keys, it'll list them and you can skip the next step.

Pray to the demo gods that the tag you're trying to crack is an old tag. The *DarkSide* attack is an exploit released in 2009, and implemented as `mfcuk`, which makes use of a predictable RNG in some Mifare tags. This has been patched in newer tags, so it's usually a guessing game if this will work on a random tag you find in the field.

This can either be done via `libnfc + mfcuk + mfoc` and a regular NFC reader, like the *ACR122U* USB libnfc readers, or by using a *proxmark3*.

```
# Using mfcuk:
$ ./mfcuk -C -R 0:A -s 250 -S 250 -v 3

# Using proxmark3 implementation:
proxmark> hf mf mifare
```

If either of these work, move on to `mfoc`; if not.... eh, better luck next time. Try picking the locks instead.

If you've found a key, use either mfoc or the proxmark3 nested attack:

```
# Using mfoc:
$ ./mfoc -O fullcard.dmp -k XXXXXXXXXXXX

# Using proxmark 3 implementation
proxmark> hf mf nested <A|B> <key (XXXXXXXXXXXX)>
```

Beating my Real Fake Door access system

There are three states of the little display:

```
  State      |  Meaning
 ----------- | ------------
  blank      | No detected Mifare tag on the reader
  *red*      | No access, but it's a valid Mifare tag at least
  *green*    | Success! Tag has the correct permissions/access controls
  ~mystery~  | SUPER SECRET MYSTERY TAGS. Ask @sysrq for an example
```

The provided *Hacker Passport stamps* (which are NFC tags!) are correctly set up for *GREEN* access. Your goal is to crack the tag, clone it to one of the little blue blank tags sitting around, and figure out what to do to also access the secret state.

There's some useful data on the stamp tags too, it might even get you points. Definitely worth getting.