python-bcrypt

Fred Wenzel

fred@mozilla.com

http://github.com/fwenzel

mozilla

bcrypt

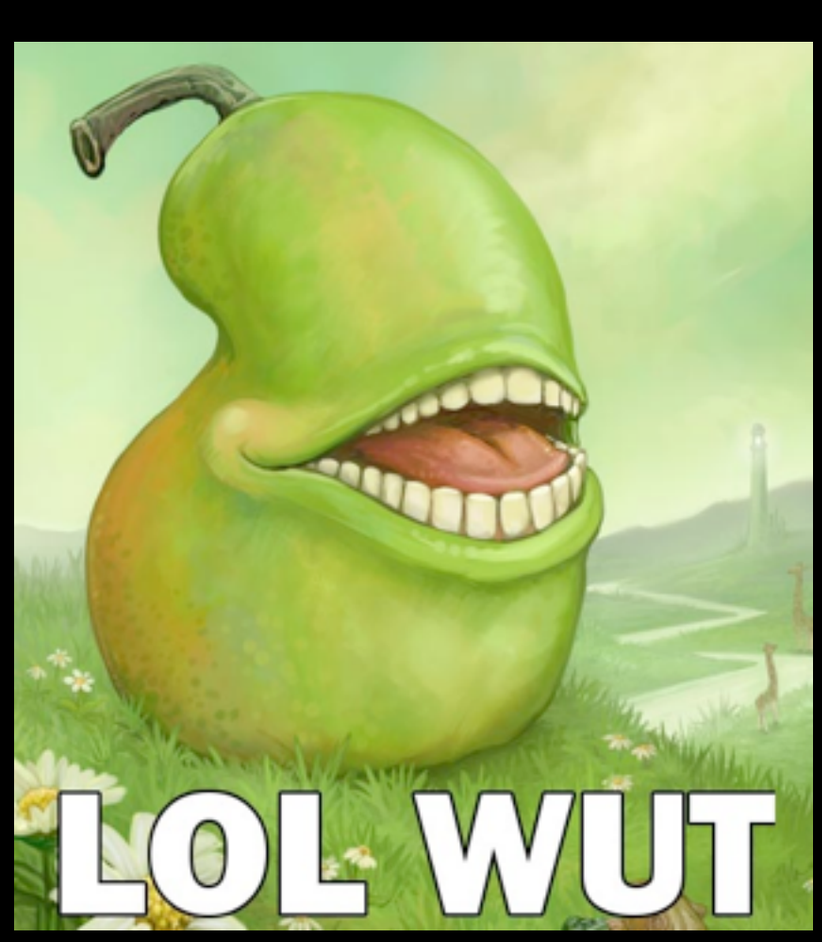
- password hashing algorithm
- intentionally slow
  - mitigates certain attacks
- default: 2^12 (= 4096)
  sequential encryption iterations

reference implementation

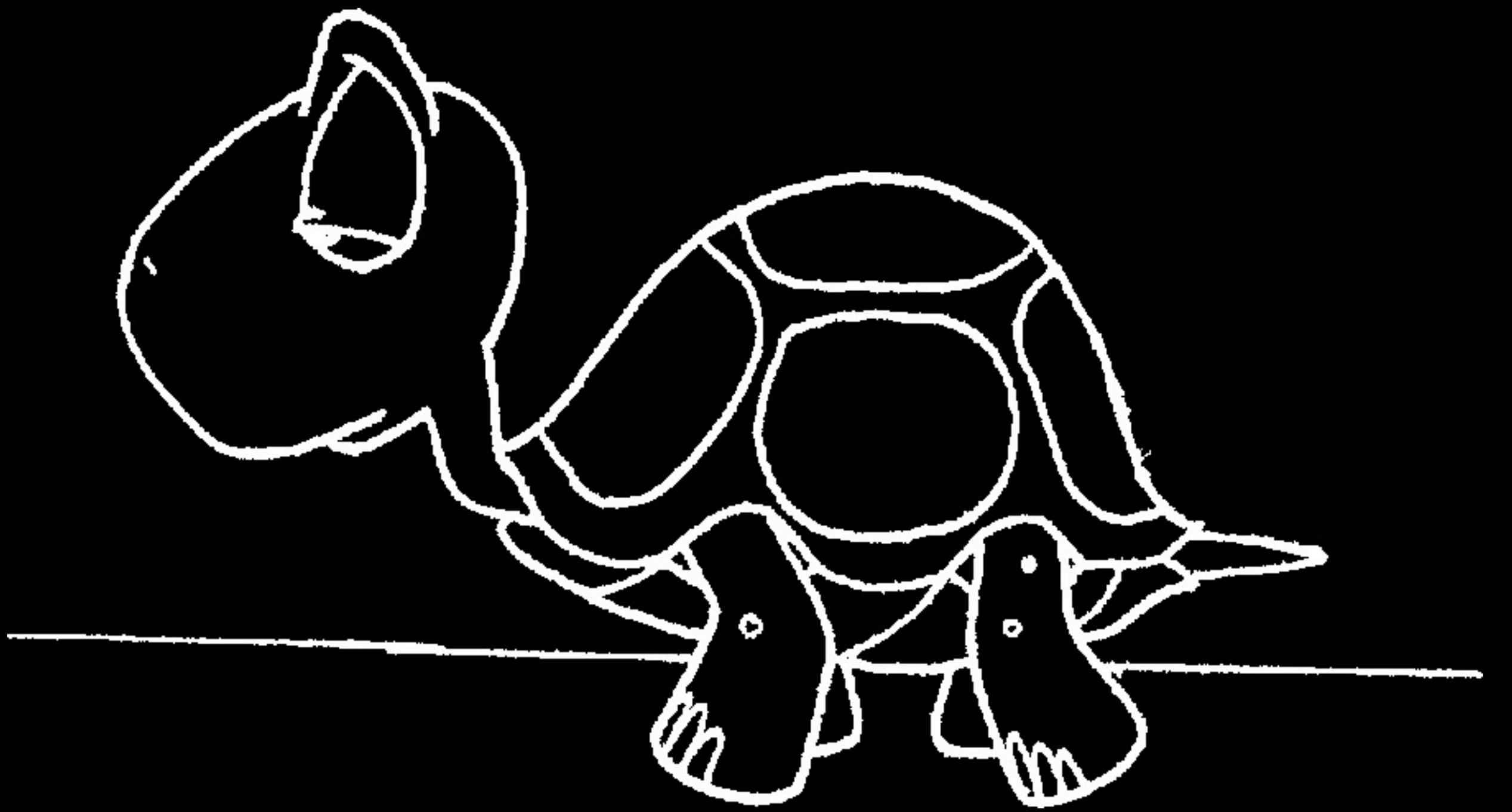
- library written in C
- py-bcrypt: thin Python wrapper around it
- What? Python can do that!
  - -› python-bcrypt

Problem 1

- spec (academic paper) -> code
- different results?
- gdb, ipdb -> replicate every place where reference impl. violates spec
- great success!

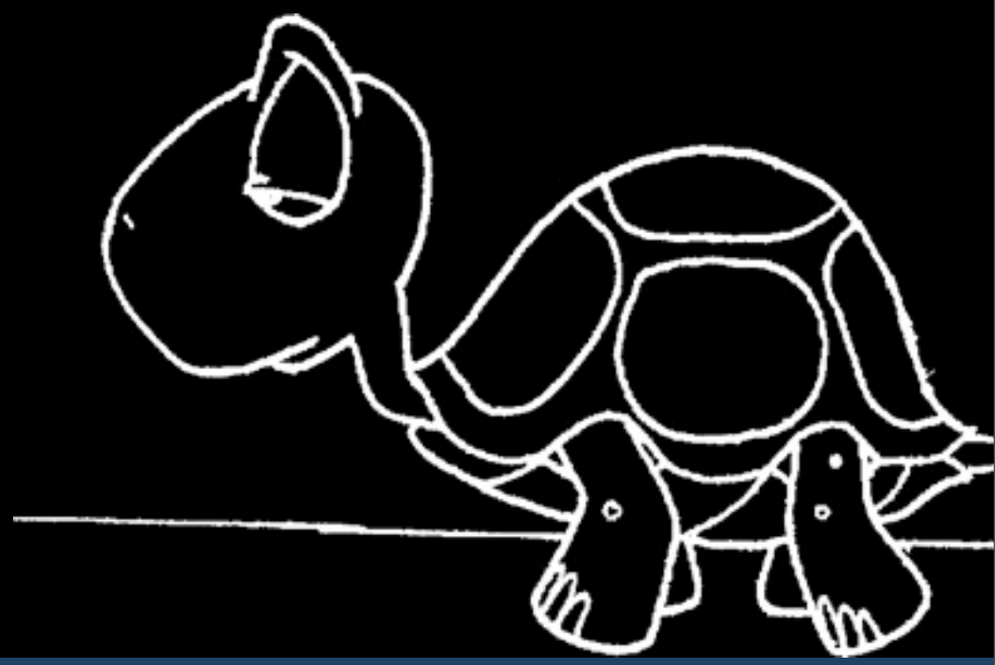


Problem 2



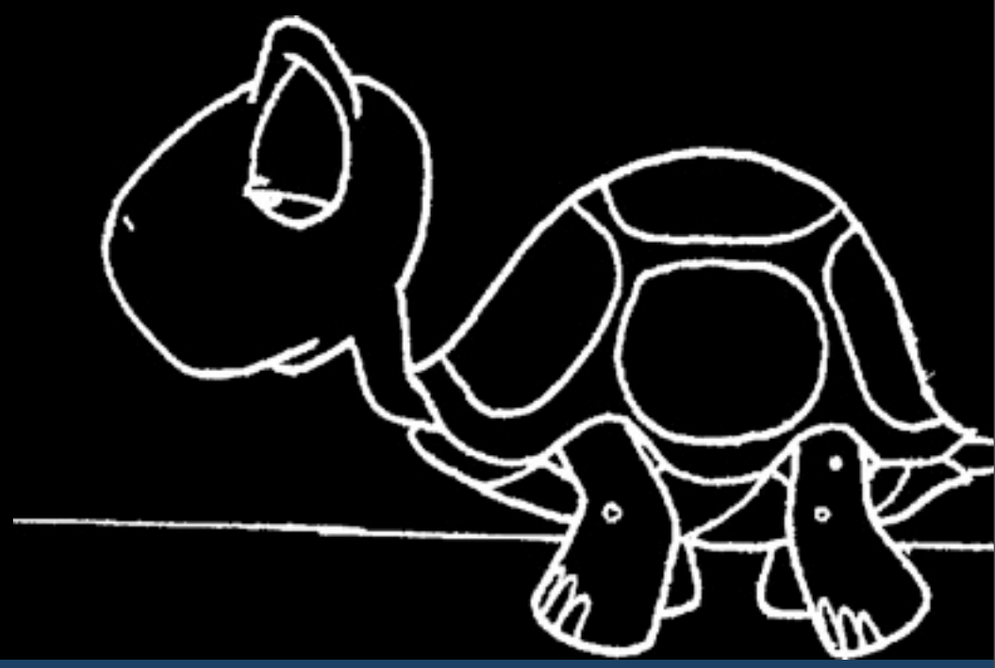
Problem 2

- SLOOOW



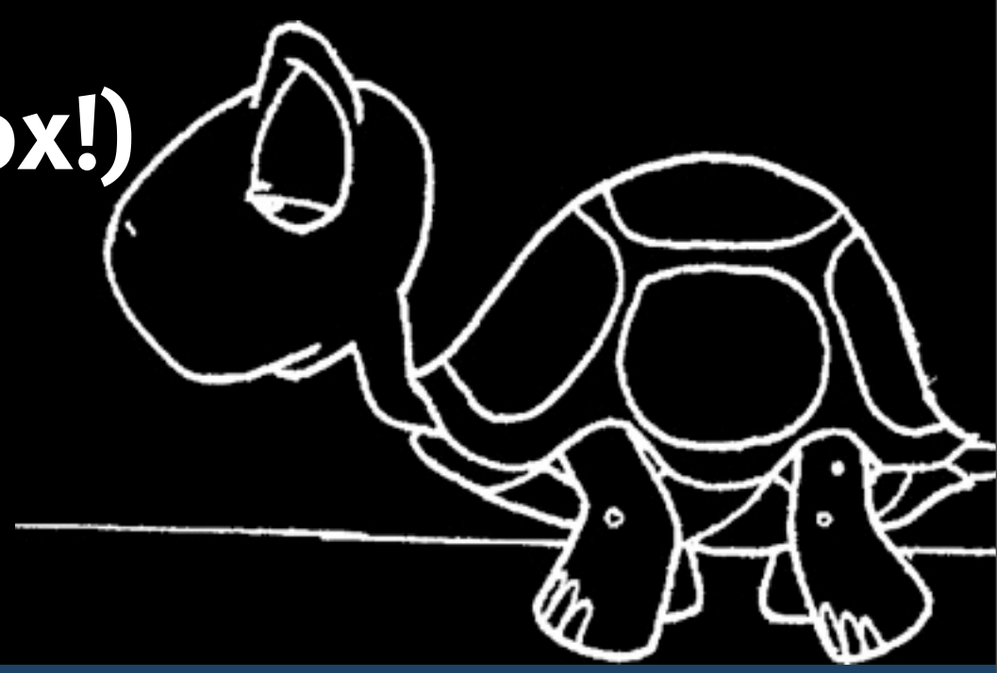
Problem 2

- SLOOOW
- C library: 0.294 CPU seconds



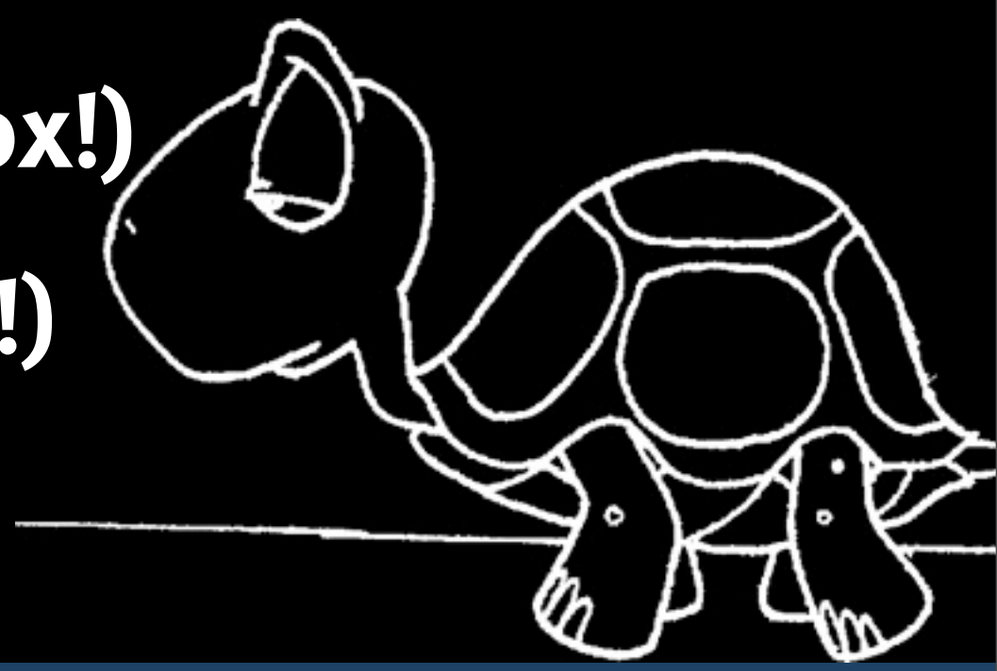
Problem 2

- SLOOOW
- C library: 0.294 CPU seconds
- python-bcrypt:
  98.004 CPU seconds (300x!)



Problem 2

- SLOOOW
- C library: 0.294 CPU seconds
- python-bcrypt:
  98.004 CPU seconds (300x!)
- pypy: 3.365 seconds (10x!)



Lessons learned

- Matching a reference implementation means replicating *it*, not the spec
- yes, including bugs

Lessons learned 2

- If it's (meant to be) slow in C, it'll be ridiculously slow in Python

Lessons learned 2

- If it's (meant to be) slow in C, it'll be ridiculously slow in Python
- pypy is awesome

Thanks!

fred@mozilla.com

github.com/fwenzel/python-bcrypt

pssst: Mozilla is hiring! mozilla.org/careers