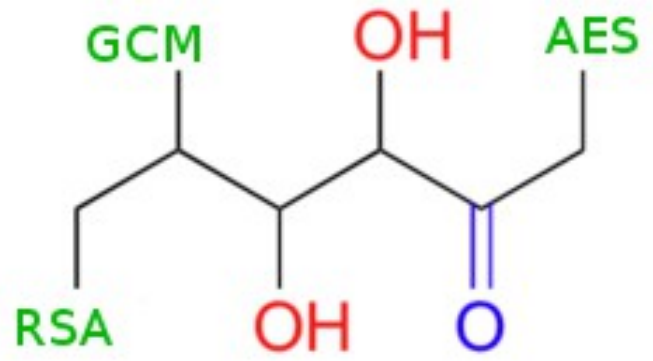
Kryptose™
“Simple & Secure”

Alexander Guziel (asg252)
Antonio Marcedone (am2623)
Jeff Tian (yt336)
Jonathan Shi (js2845)



Our mission…. (System purpose)

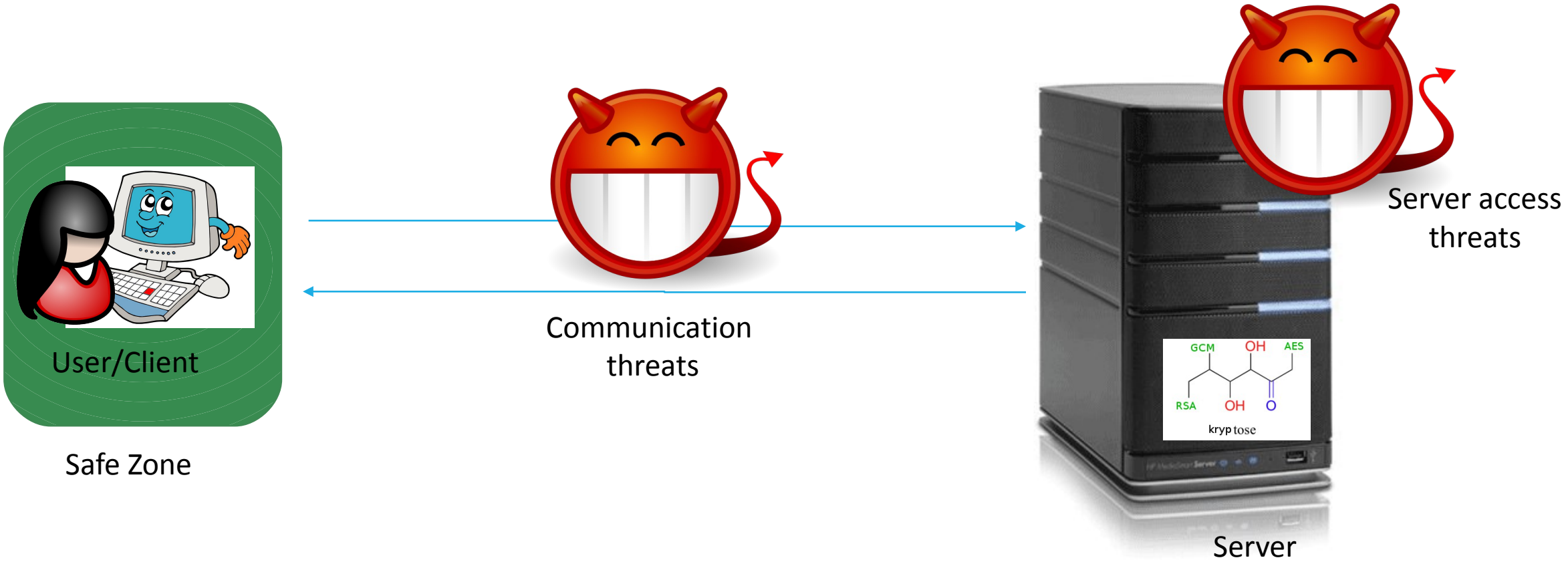
- Memorizing different and high entropy passwords enhances security but it is hard
- As a result, users often choose weak passwords

Our mission…. (System purpose)

- Memorizing different and high entropy passwords enhances security but it is hard
- As a result, users often choose weak passwords

- Our system solves the problem, by securely storing all these passwords online
- Users need to remember only one Master Password

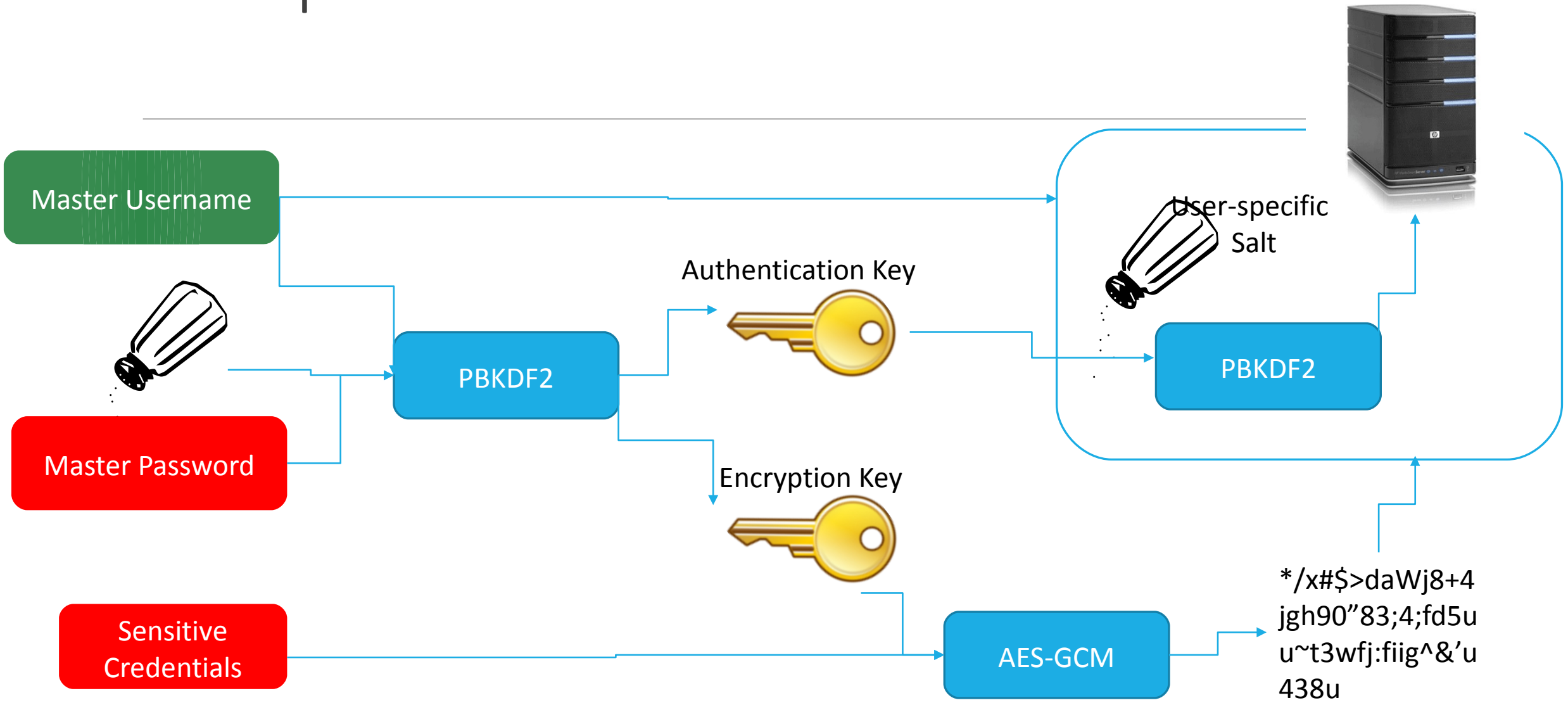
Threat model



Security Goals

- Nobody except the user itself (not even Kryptose™'s sys admin) should be able to learn or tamper with a user's sensitive credentials.
- The system shall prevent other principals from modifying the user's master password or from deleting the accounts.
- A user's Master Password should not be disclosed to any principal
- The server logs should be readable and modifiable only by the system administrator

One password to rule them all



Other features

- We use TLS1.2 with perfect forward secrecy.
- The client authenticates the server with a Certificate by our own CA (no hostname verification, but we only sign certificates for ourselves).
- The server collects Tamperproof Logs
- Encryption using AES in GCM mode.
- Reference Monitor

Testing

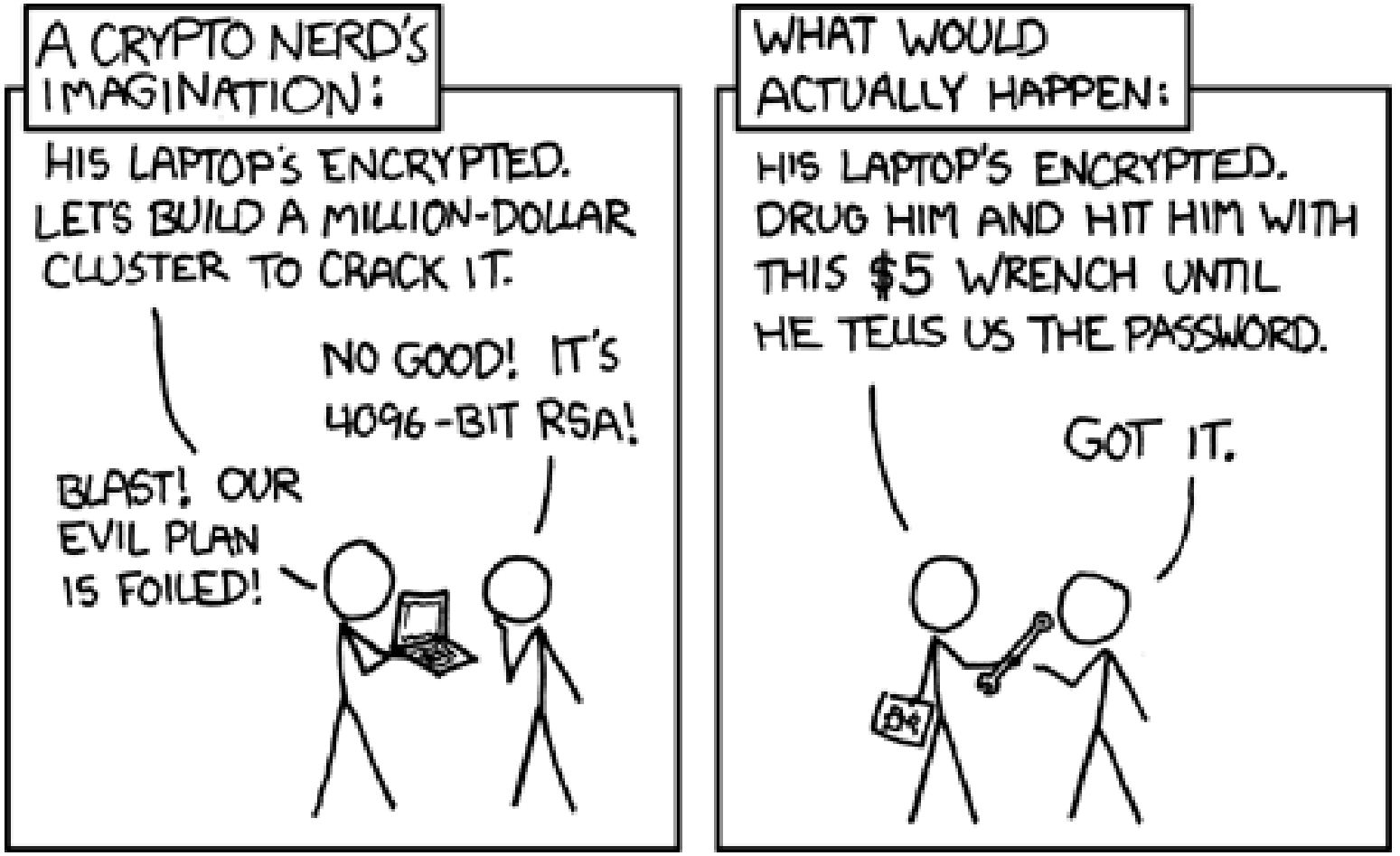
- Junit tests for the cryptographic functionalities and the reference monitor
  (>70% coverage, with manual inspection of the missing parts)
- Integration and unusual context testing for the SSL communication code
  (weak SSL versions are not accepted, certificates are actually checked, we only accept our own certificates and exclude the trust anchors from the OS store)
- Extensive usage testing for the rest of the code and program logic

Limitations….

Technical Limitations….

- Java DOES NOT guarantee to clear memory.
- No CRL check during SSL communication
- No hostname validation (not a problem)
- PBKDF2 is (slightly) vulnerable to attacks with specialized hardware (a new standard is on its way)
- No offline functionality
- No protection against attacks to Availability
- Server logs can leak access patterns and usage habits to a malicious sys admin

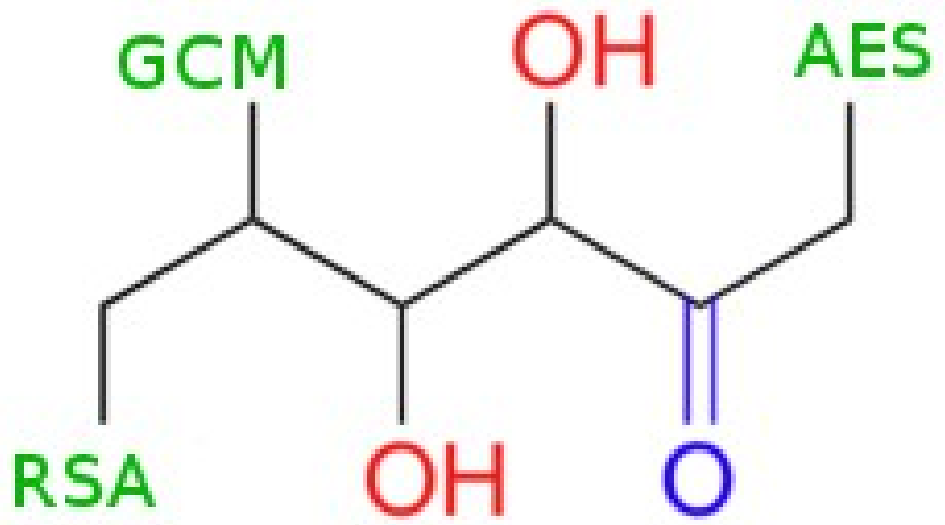
More Limitations…..



Security Elements

- Authentication: The use of an authentication key (derived from a Master Password only known to the user) ensures that only the intended principals can access their own credentials.
  Logs are protected by a key known to the administrator and not stored in the system.
- Authorization: A reference monitor (UserTable.java) analyzes all requests before they are handled to guarantee.
- Audit: Tamperproof server logs contain all the requests received by the server, with data on the principals making those requests.
- Confidentiality/Integrity : Stored credentials are encrypted and can only be read by the user.

Thanks!



kryptose