The Pythia PRF Service
Paper Summary

Ruturaj Kiran Vaidya
University of Kansas

1 Summary

To harden password databases, well known companies sent passwords to PRF service before getting verified. PRF services apply cryptographic function such as HMAC to inputs under a secret key, ensuring the security against offline brute-force attacks. Even though current PRF services are good for the security purpose, they have some issues. Paper proposes "Pythia", which offers flexibility, security and ease of deployment, which were lacking in the previous approaches. "Pythia" is a next-gen PRF service democratizes cryptographic hardening. "Pythia" offers message privacy, while requiring clients to reveal tweak to the server, it offers individual key rotation (fast), enabling forward key security. So, the basis of "Pythia" is a verifiable, partially partially-oblivious PRF ("a two-party protocol that allows secure the secure computation of Fk(w)(t,m), where F is a PRF with server-held key kw and t;m are the input values"[1]), that hides the input message, revealing the portion of it. It also "supports efficient bulk rotation of previously obtained PRF values to new keys"[1]. Authors claim that their system is highly practical, showing highly practical performance on Amazon EC2 instances.

2 Contribution

Paper presents design and implementation of a next-gen PRF system called "Pythia", offering flexibility, security and ease of deployment, over previous approaches. "Pythia" shows highly practical performance on Amazon instances. Authors developed a new password database system with password "onion", "combining palatalized calls to "Pythia" and a conventional key hashing mechanism". It supports "Pythia" key rotations and achieves high security. They also gives two applications using "Pythia", bringing security benefits to a new enterprise password storage system and a new brainwallet system for Bitcoin.

3 Strengths

"Pythia" can be deployed not only within the enterprise, but also as a public multi-tenant web service, solving issues with previous PRF approaches. It prevents offline brute-force attacks. Also, it shows good performance.

4 Limitations

"Pythia" demands more computational effort from the attacker, to be compromised, but I think it is still vulnerable (maybe modern deep learning techniques might help the attacker). There are some obvious performance implications(Also, I think using this in the extreme public cloud environment will raise additional performance concerns). Also it reveals a portion of message (I find this bad).

5 Future Work

Future work is required to reduce the performance implications. Also, I find message revealing bad, more work can be done to avoid that.

References

[1] Adam Everspaugh, Rahul Chatterjee, Samuel Scott, Ari Juels and Thomas Ristenpart *The Pythia PRF Service*, 24th USENIX Security Symposium (USENIX Security 15)