

# SEIFPASS: A SECURE PASSWORD MANAGER

**MIT PRIMES**

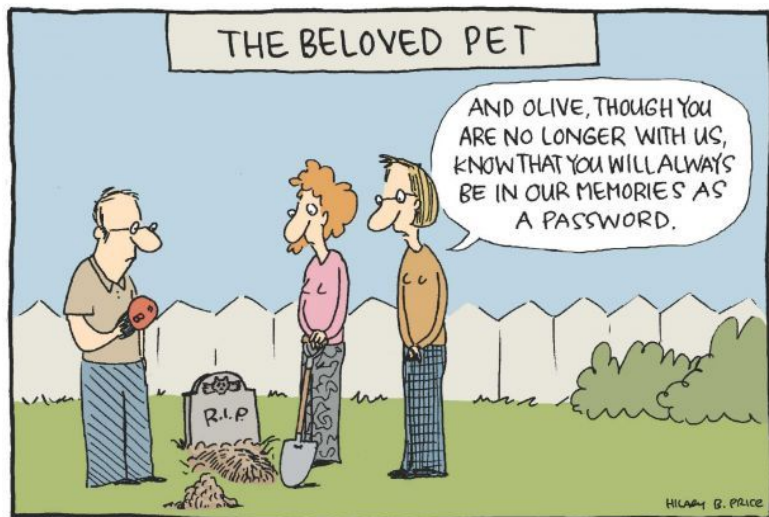
**By: Cristian Gutu**

**Computer Science**

**Mentor: Albert Kwon**

# PROBLEM

- Passwords: **common**, but **vulnerable** to offline attacks



Web Server



User	Password
Alice	password123
Bob	MyPetsName
...	...

# HASH

Web Server



User	Password
Alice	hash(password123)
Bob	hash(MyPetsName)
...	...



Attacker



Common passwords	Computed hash
password123	hash(password123)
MyPetsName	hash(MyPetsName)
...	...

# SALTED HASH

Web Server



User	Hashed Password, Salt
Alice	hash(password123 + 4abcd@!#), 4abcd@!#
Bob	hash(MyPetsName + abcd%), abcd%
...	...

Attacker



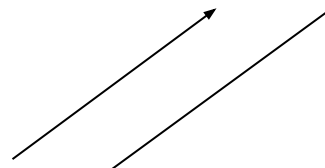
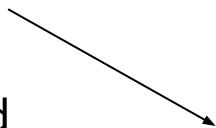
Common passwords	Computed hash
password123	hash(password123 + 4abcd@!#)
MyPetsName	hash(MyPetsName + abcd%)
...	...

# SOLUTION

- Architecture for password hardening using a remote cryptographic **service**

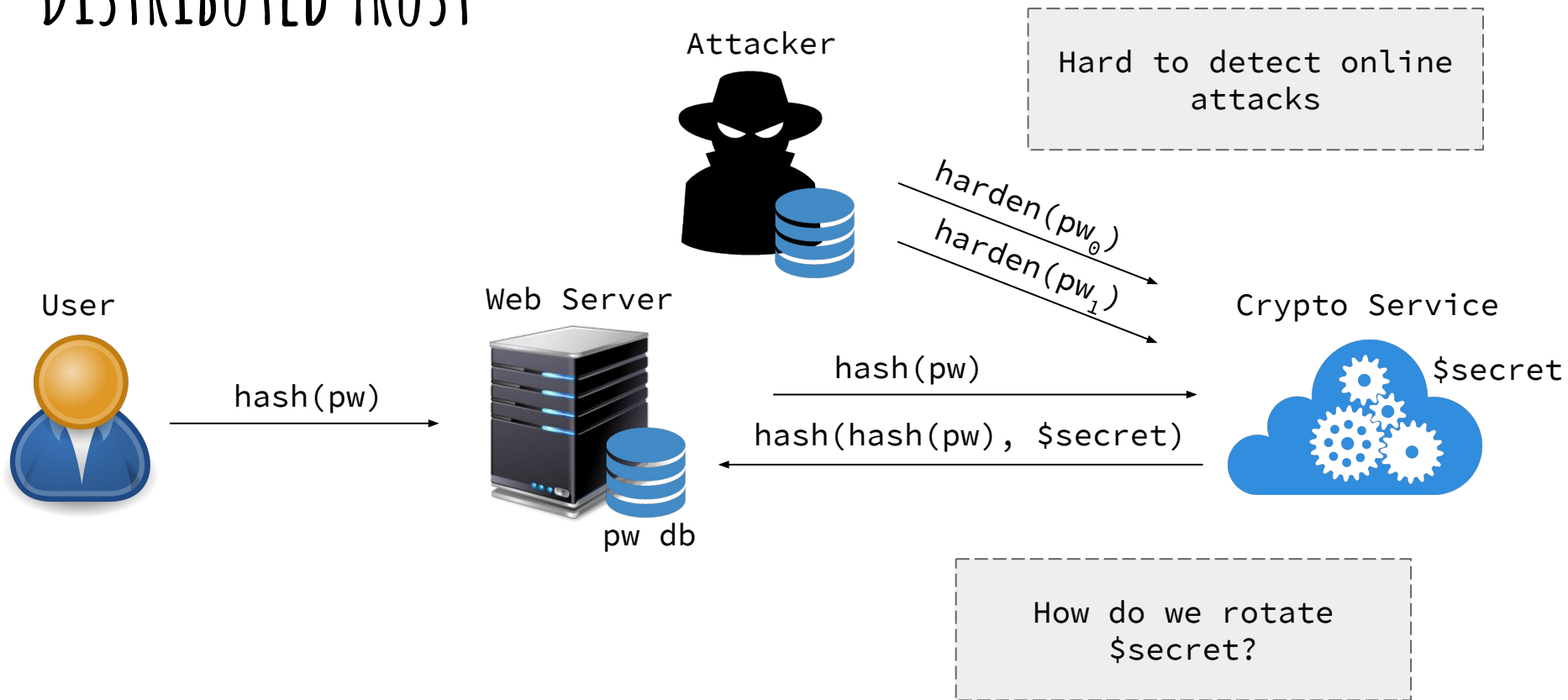
Benefits:

- No offline attacks

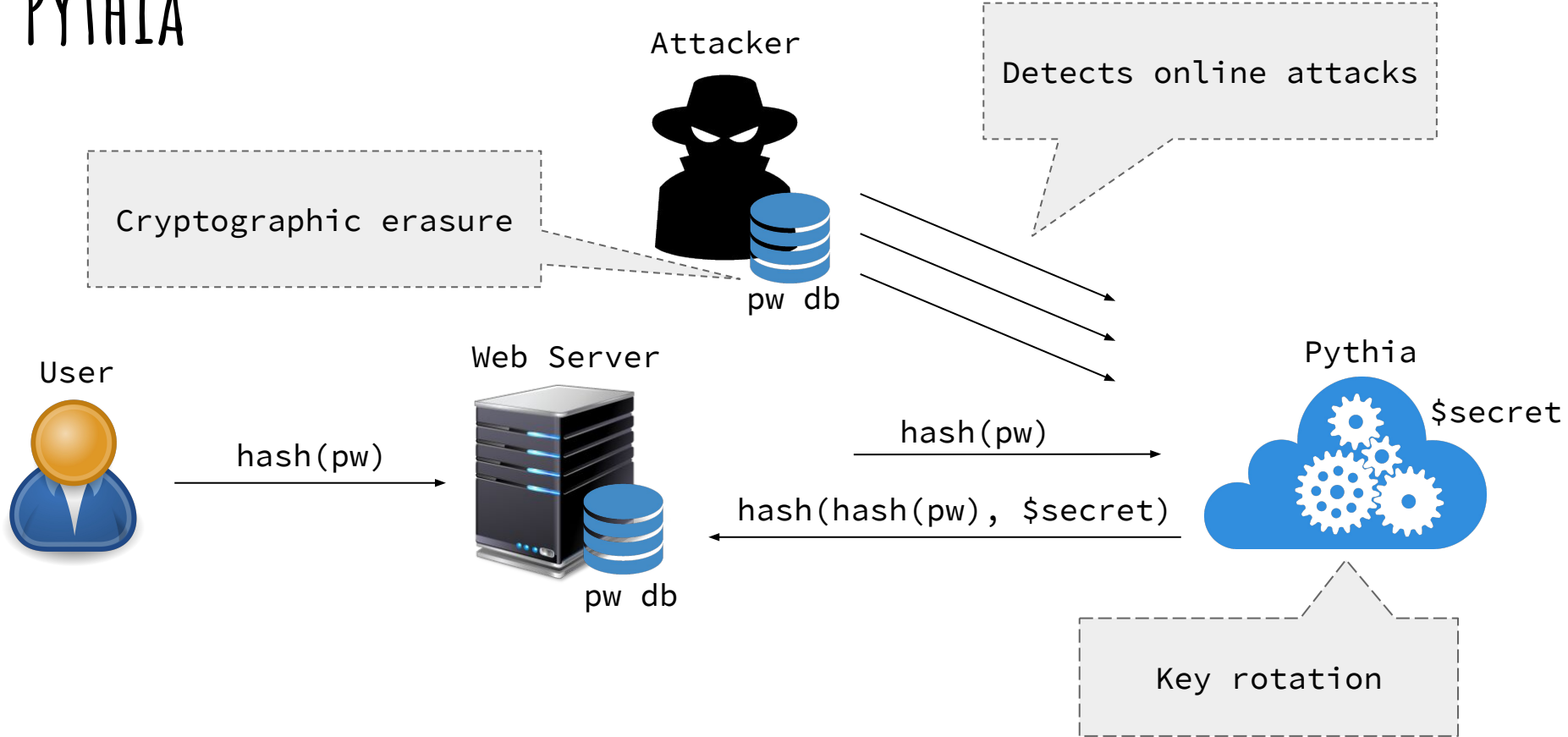


Hardening: encrypting password with secret key stored only on crypto service

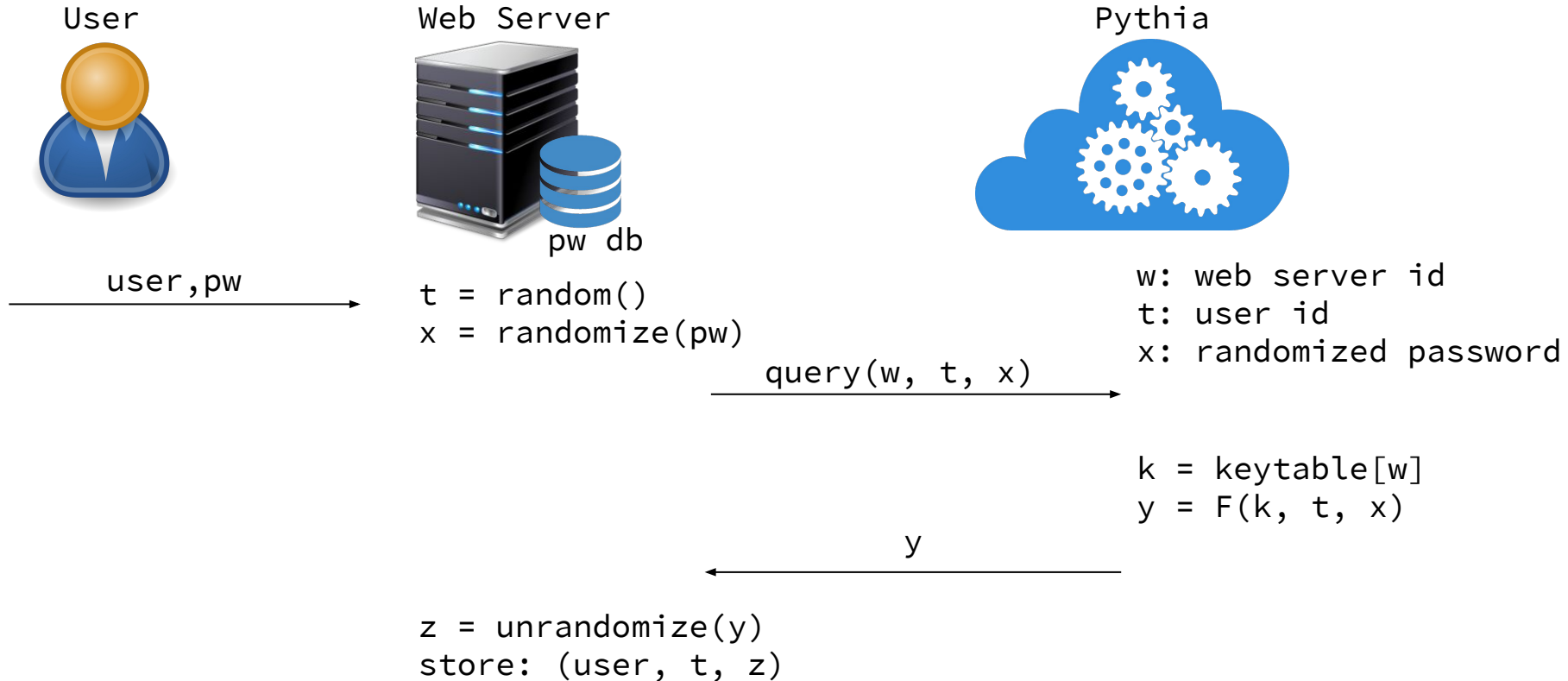
# DISTRIBUTED TRUST



# PYTHIA

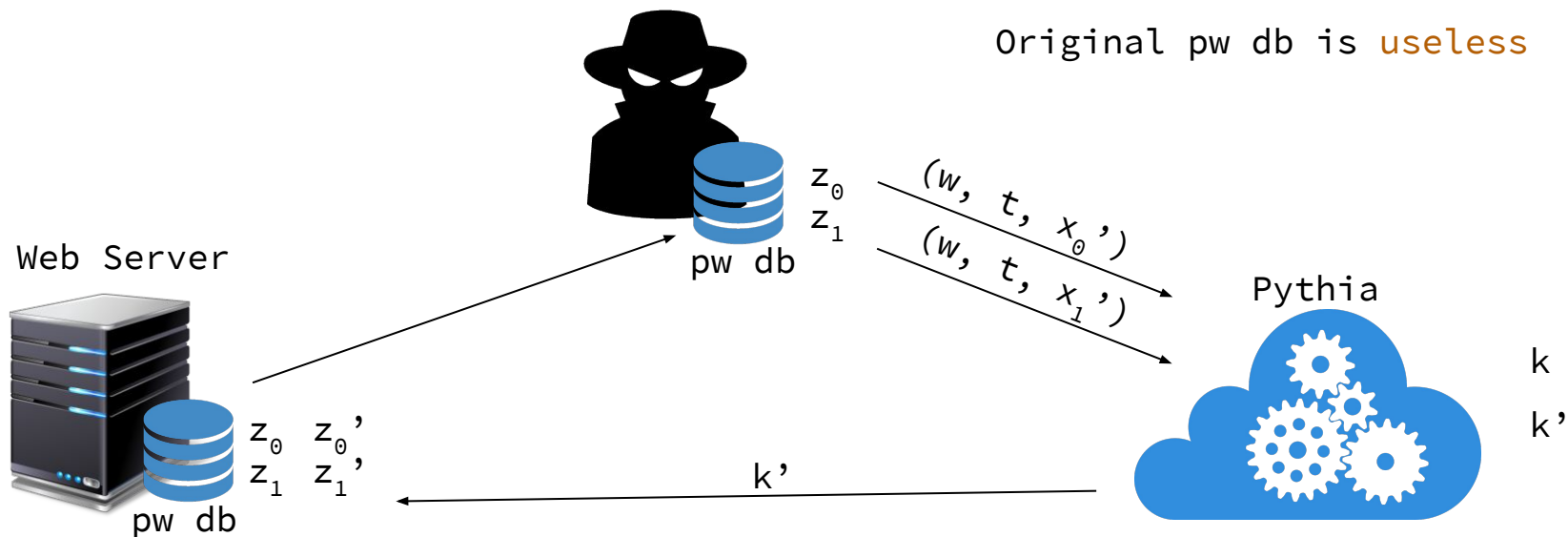


# PYTHIA - NEW USER





# COMPROMISE RECOVERY



No need for original password

User password does not change

## Key:

$z_i$  : passwords

$w$ : web server id

$t$ : user id

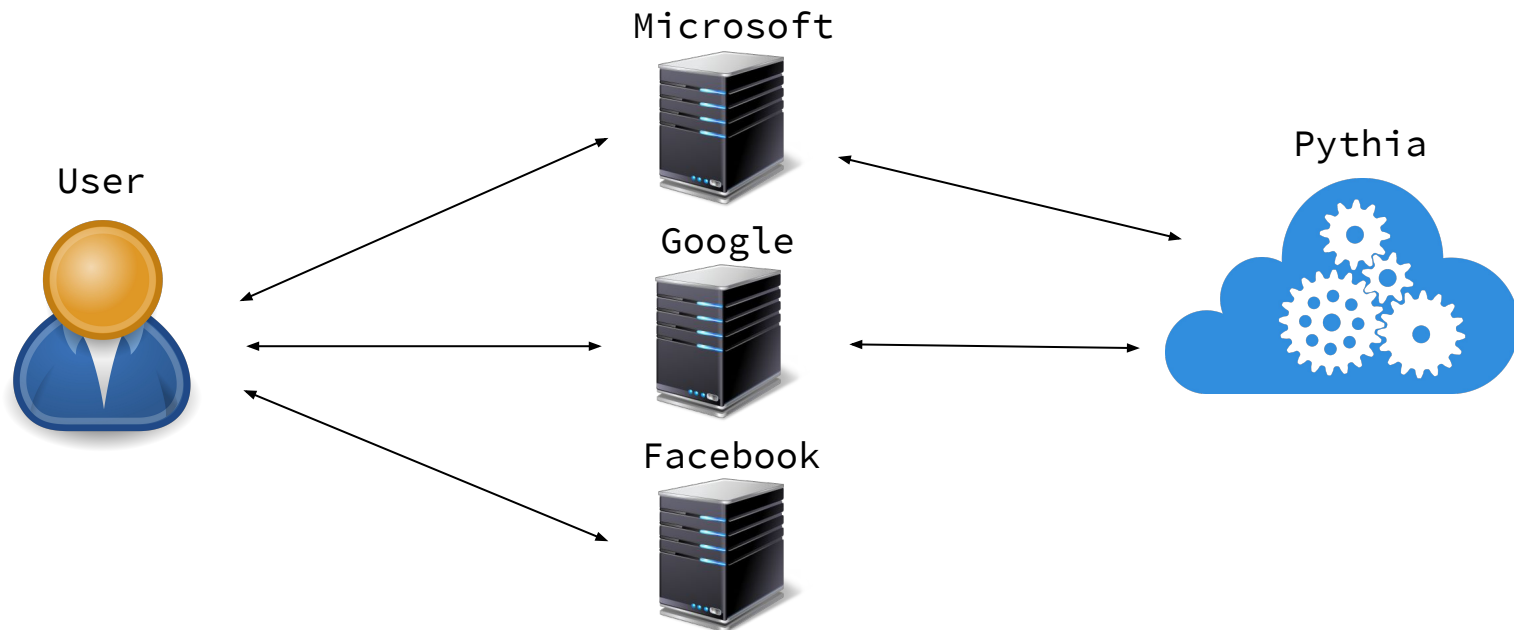
$x$ : randomized password

$k$ : pythia instance for this web

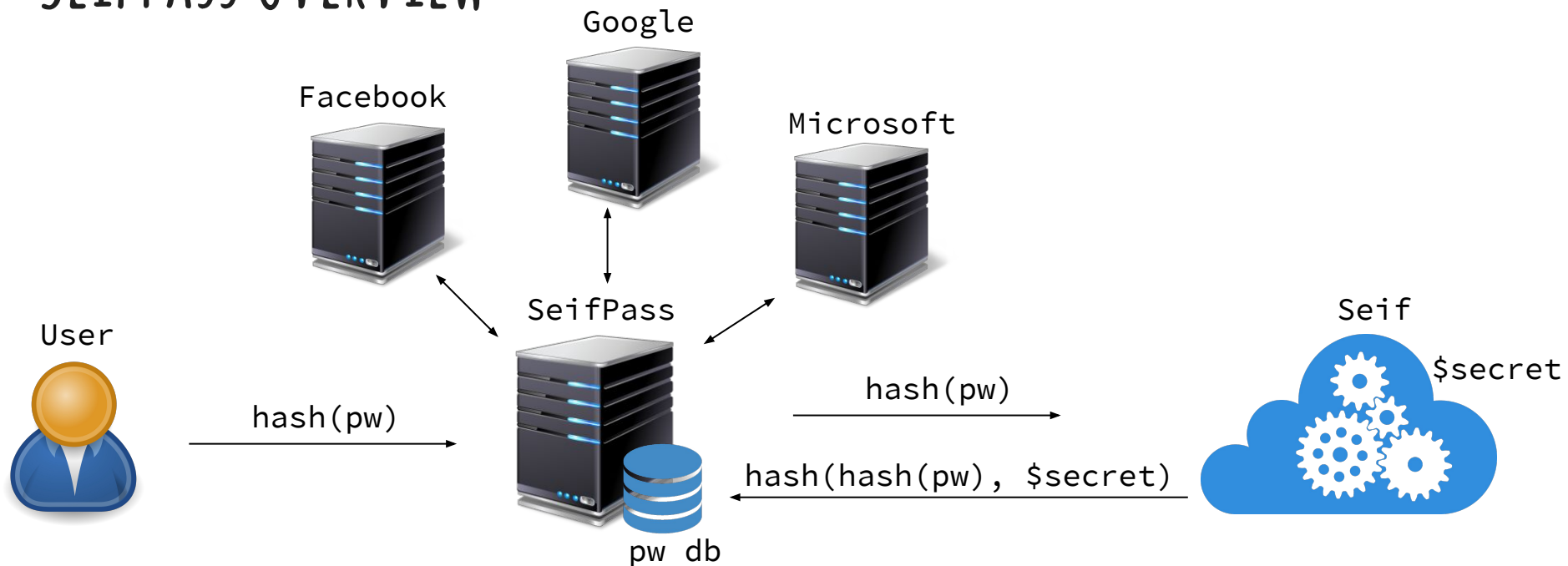
server

# LIMITATIONS OF PYTHIA

- May be hard to merge in large code base
- No guarantee applications will implement it



# SEIFPASS OVERVIEW



User does not have to depend on application to secure password

# SEIF - NODE.JS PYTHIA WRAPPER

- Pythia Service in Node.js
- Easy to use
- <https://github.com/naitsirc/seif>

## Dependencies:

- Relic Crypto Library
- pyrelic (python wrapper for Relic)
- Node.js
- Express.js (web application framework)
- mongoDB



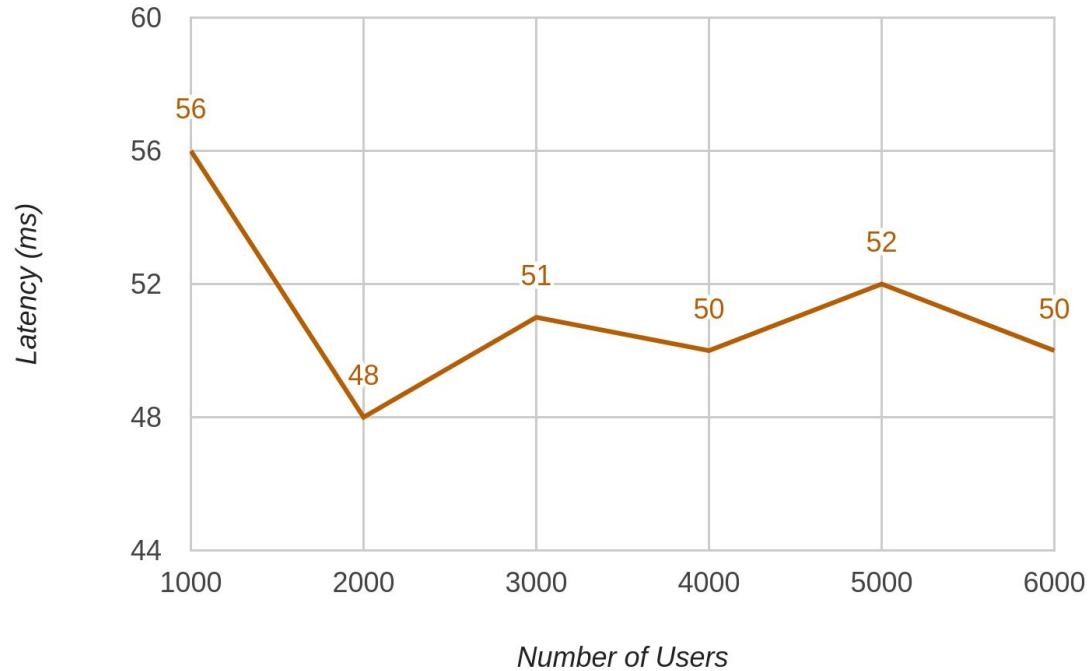
express



mongoDB®

# SCALABILITY STUDY

## New Password Latency vs Number of Users



# SEIFPASS - SIGN UP

The image shows a 'Sign up' form with a blue header bar containing the text 'Sign up'. Below the header, there are three input fields, each with a label above it: 'Username' with the value 'Cristian', 'Email address' with the value 'Email', and 'Password' with the value 'Password'. Below the 'Password' field is a blue button with the text 'Sign up'. At the bottom of the form is a link labeled 'Sign in'.

# SEIFPASS - SIGN IN

Sign in

**Username**

**Password**

[Sign in](#)

[Create account](#)

# SEIFPASS - NEW PASSWORD

SeifPass

Delete Log out

New password

**Account**

Google

**Generated password**

CfSrb1gE

Generate

Edit

Save

Close



# SEIFPASS - PASSWORD MANAGER

The screenshot displays the SeifPass password manager interface. At the top, a dark header bar contains the application name 'SeifPass' on the left and three action buttons: '+ New', '× Delete', and '↶ Log out'. Below the header, there are two entries, each with a blue header bar and a white content area. The first entry is for 'Google' and the second is for 'Facebook'. Both entries show a 'Password:' label followed by a long alphanumeric string.

**SeifPass** + New × Delete ↶ Log out

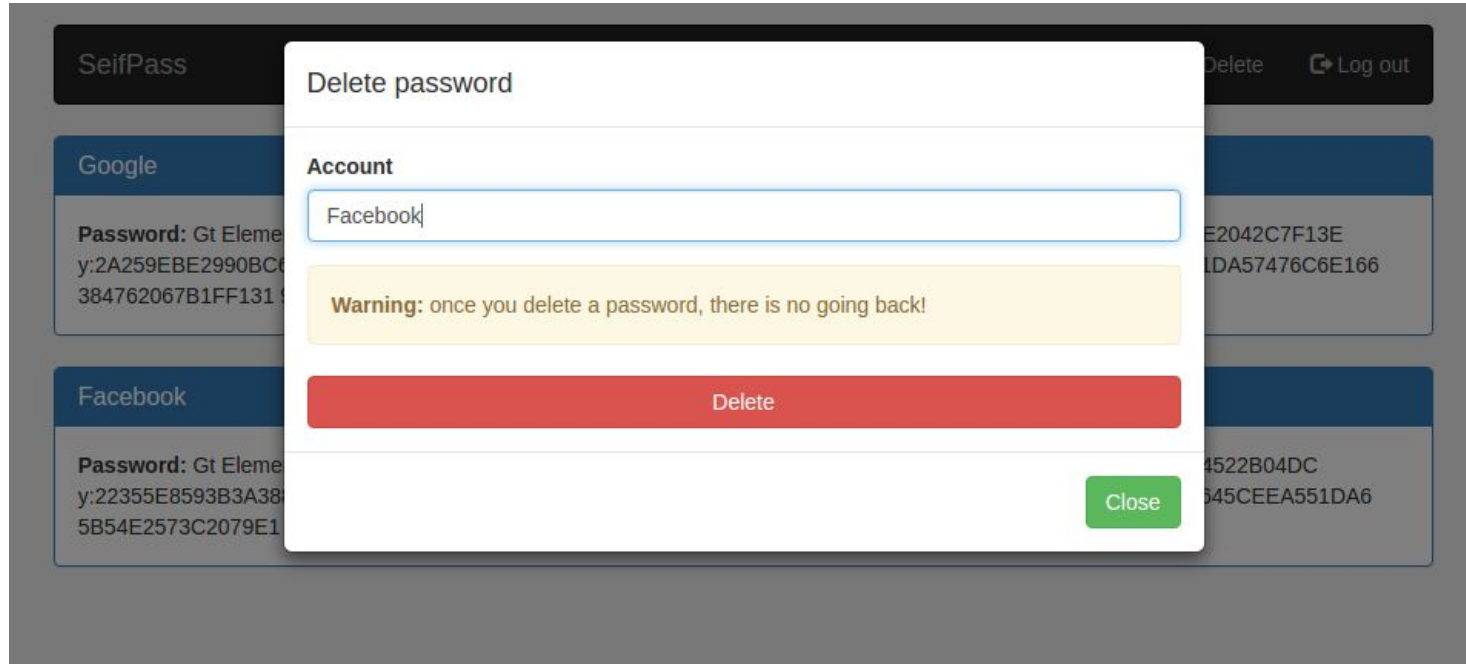
**Google**

**Password:** Gt Element Not Normalized x:5ED5BF854962C807 DF5DC269E9DB0FAD C7ACA210C6ECD5CB 9F510E2042C7F13E y:2A259EBE2990BC62 7D2A2274A4191683 8965A966C9146D4D 3FC1FD2B87DD2EAD z:2A68E8E841B0894A 231DA57476C6E166 384762067B1FF131 98F68BE6EC6F7916

**Facebook**

**Password:** Gt Element Not Normalized x:180919B0C9A28AE BBCFEBC96C8FF13 F8FD826752191F27 5DB44DB4522B04DC y:22355E8593B3A388 6957FDE1268CE850 5FBDBA4B9066283F 3230B2770F77953E z:51A8DDCB600801AD 8F8645CEEA551DA6 5B54E2573C2079E1 998B6F8C00095ED6

# SEIFPASS - DELETE PASSWORD



# ACKNOWLEDGMENTS

Thank you to Albert Kwon for mentoring  
and project suggestion!

Thank you Srimi Devadas for the Computer  
Science Program!

Thank you MIT PRIMES for the  
opportunity!

Thank you to my parents for the support!