

The Pythia PRF Service

Adam Everspaugh, Rahul Chatterjee,
Sam Scott, Ari Juels, Thomas Ristenpart



Summary

Passwords: **Ubiquitous**, but **vulnerable** to offline attack



password db

New direction: Complete architecture for password storage using a new cryptographic PRF **service**.

Better: no offline attacks, compromise recovery, key management, cryptographic erasure of stolen information





Sign In

Not a member? [Join now](#)

The LinkedIn logo, consisting of the word "Linked" in black and "in" in white inside a blue square, with a trademark symbol.

Not a member? [Join now](#)

LinkedIn © 2015 [User Agreement](#) [Privacy Policy](#) [Community Guidelines](#) [Cookie Policy](#) [Copyright Policy](#) [Guest Controls](#)

Website stores one of:

- pw
- ➔ Hash(pw)
- salt, Hash(salt, pw)
- salt, Hash⁴⁰⁹⁶(salt, pw)

6.5M hashes leaked

90%
recovered 2 weeks

Password Database Compromises

U.S. Edition News Video TV Opinions More... Search CNN

U.S. World Politics Tech Health Entertainment Living Travel Money Sports

Yahoo hacked, 450,000 passwords posted online

By Doug Gross, CNN Updated 9:31 AM ET, Fri July 13, 2012

Wired

home > tech US world opinion sports soccer arts lifestyle fashion bu

Hacking

engadget REVIEWS - FEATURES - GUIDES - VIDEOS - GALLERIES - FORUMS - GAMING - Search Products & Articles

THE ECONOMIC TIMES

al network hacked, some unt details compromised

50 ha

By Doug Gross, CNN Updated 4:34 PM

criminals had compromised accounts and passwords of 38 million users.

The California-headquartered firm said it has informed all the affected users and has reset their passwords.

"Our investigation has confirmed that the attackers obtained access to Adobe IDs and what were at the time valid, encrypted passwords for approximately 38 million active users," an Adobe spokesperson told PTI.

On October 3, Adobe said it faced two attacks from cyber criminals who stole credit card data of 2.9 million customers. Its security team had discovered the sophisticated attacks involving illegal access of customer information and source code of many Adobe products.

Blizzard Entertainment has just posted an "important security update" to its official site. The studio responsible for *World of Warcraft*, *Diablo III*, and *StarCraft* revealed that its security team "found an unauthorized and illegal access" into Blizzard's internal network.

5

Facebook's Password Onion



```
$cur = 'password'  
$cur = md5($cur)  
$salt = randbytes(20)  
$cur = hmac_sha1($cur, $salt)  
$cur = remote_hmac_sha256($cur, $secret)  
$cur = scrypt($cur, $salt)  
$cur = hmac_sha256($cur, $salt)
```

Archeological record of FB's struggles with password security.

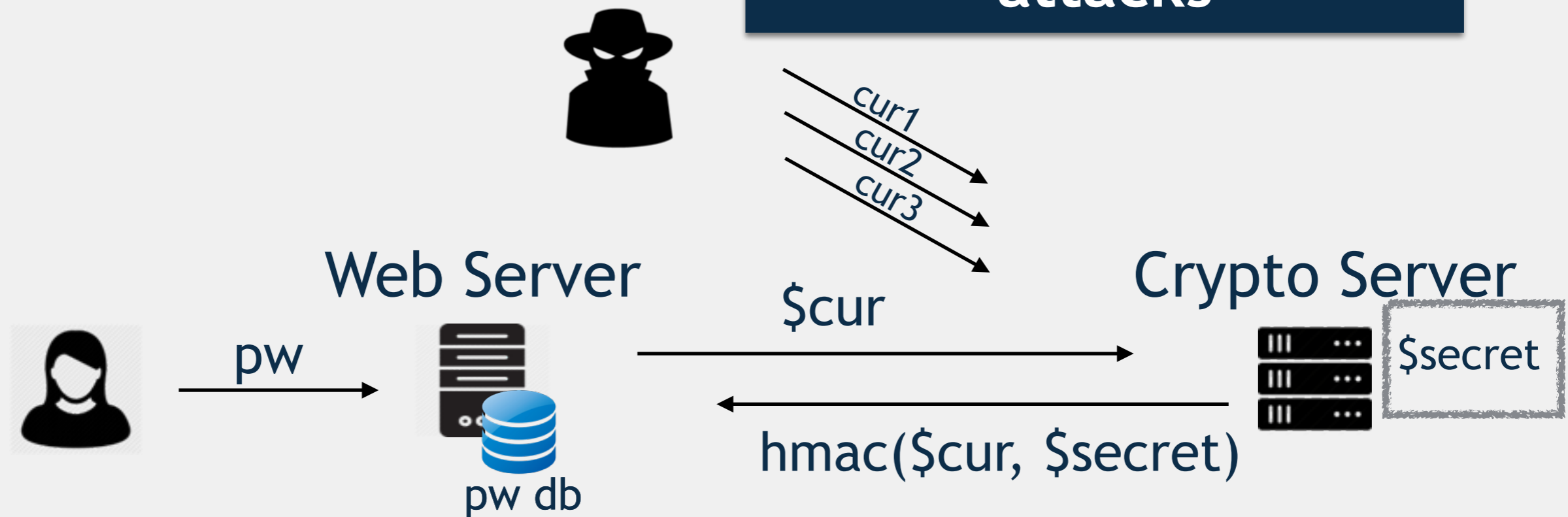
Facebook's Password Onion



```
$cur = 'password'  
$cur = md5($cur)  
$salt = randbytes(20)  
$cur = hmac_sha1($cur, $salt)  
$cur = remote_hmac_sha256($cur, $secret)  
$cur = scrypt($cur, $salt)  
$cur = hmac_sha256($cur, $salt)
```

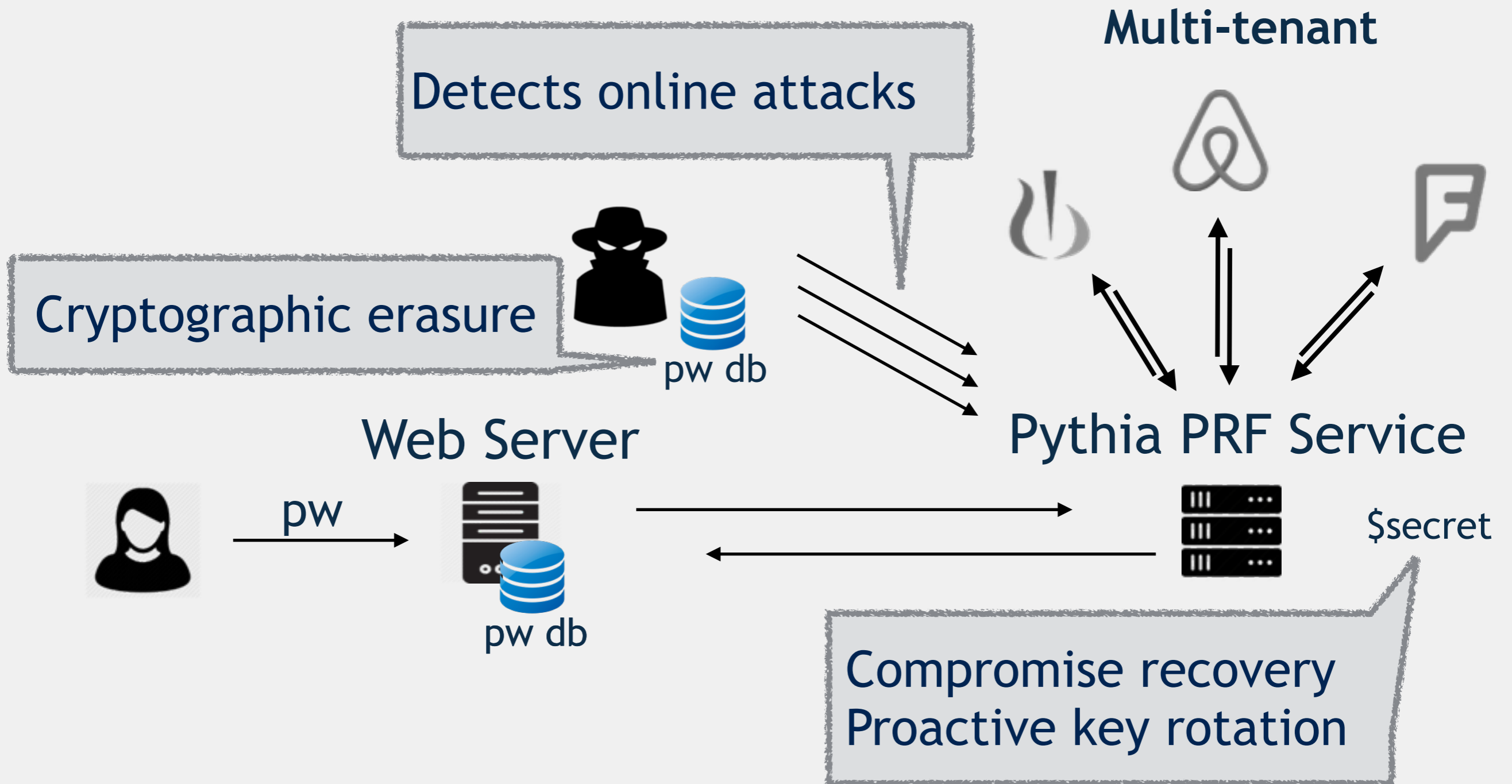
Remote HMAC Distributes Trust

Hard to detect online attacks



How do we rotate \$secret?

Our Approach: Pythia PRF



PRF Query – New User

User



Web Server



Pythia Server



user, pw

```
t := random()  
x := blind(pw)
```

Web Server ID

Blinded PW

query: w, t, x

User ID

```
k := keytable[w]  
y := Fk(t, x)
```

y

Protected PW

```
z := unblind(y)  
store: (user, t, z)
```

Compromise Recovery



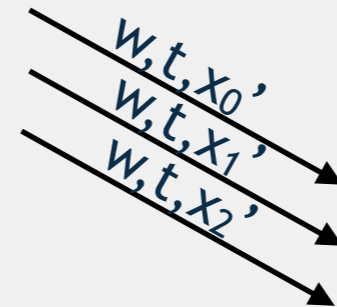
Password db is **useless**

Web Server



~~z_0~~
 ~~z_1~~
...

z_0'
 z_1'
...



Pythia Server

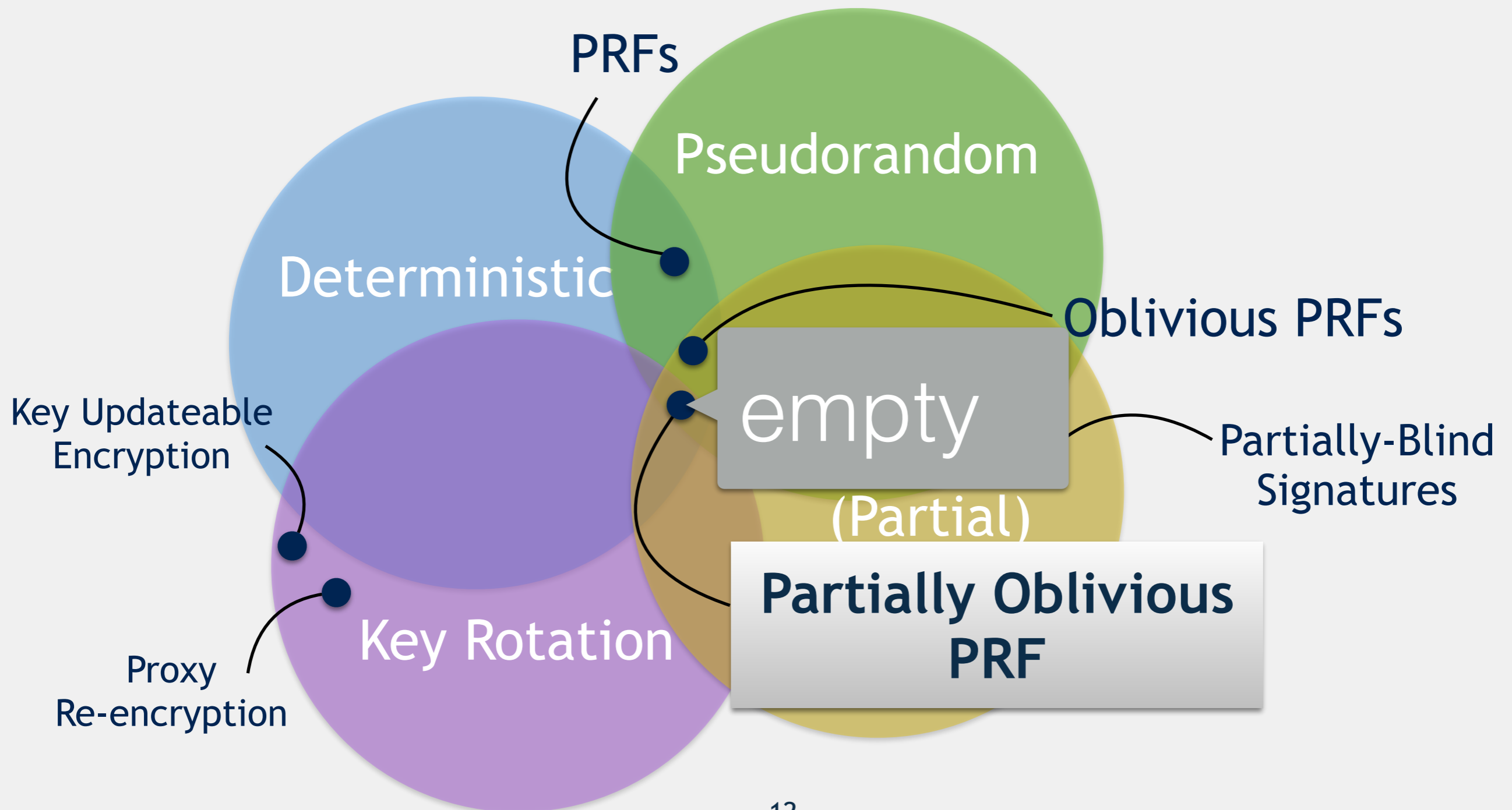


k'

$\Delta_{k \rightarrow k'}$

Doesn't require original password
User password remains unchanged

Existing Crypto Primitives are Insufficient



Partially Obl. PRF Construction

Bilinear Pairing

$$e: G_1 \times G_2 \rightarrow G_T$$

$$e(a^x, b^y) = e(a, b)^{xy}$$

Web Server



$$x := H(\text{pw})^r$$

blind()

PRF Query

w, t, x

$$y := e(H(t), x)^k$$

$F_k(t, x)$

Pythia Server



$$k := \text{keytable}[w]$$

$$z := y^{1/r} = e(H(t), H(\text{pw}))^{k*r*1/r} = e(H(t), H(\text{pw}))^k$$

unblind()

Similar use of pairings: [Sakai, Ohgishi, Kasahara] [Boneh, Waters]

Partially Obl. PRF Construction

Web Server



Pythia Server



Compromise Recovery

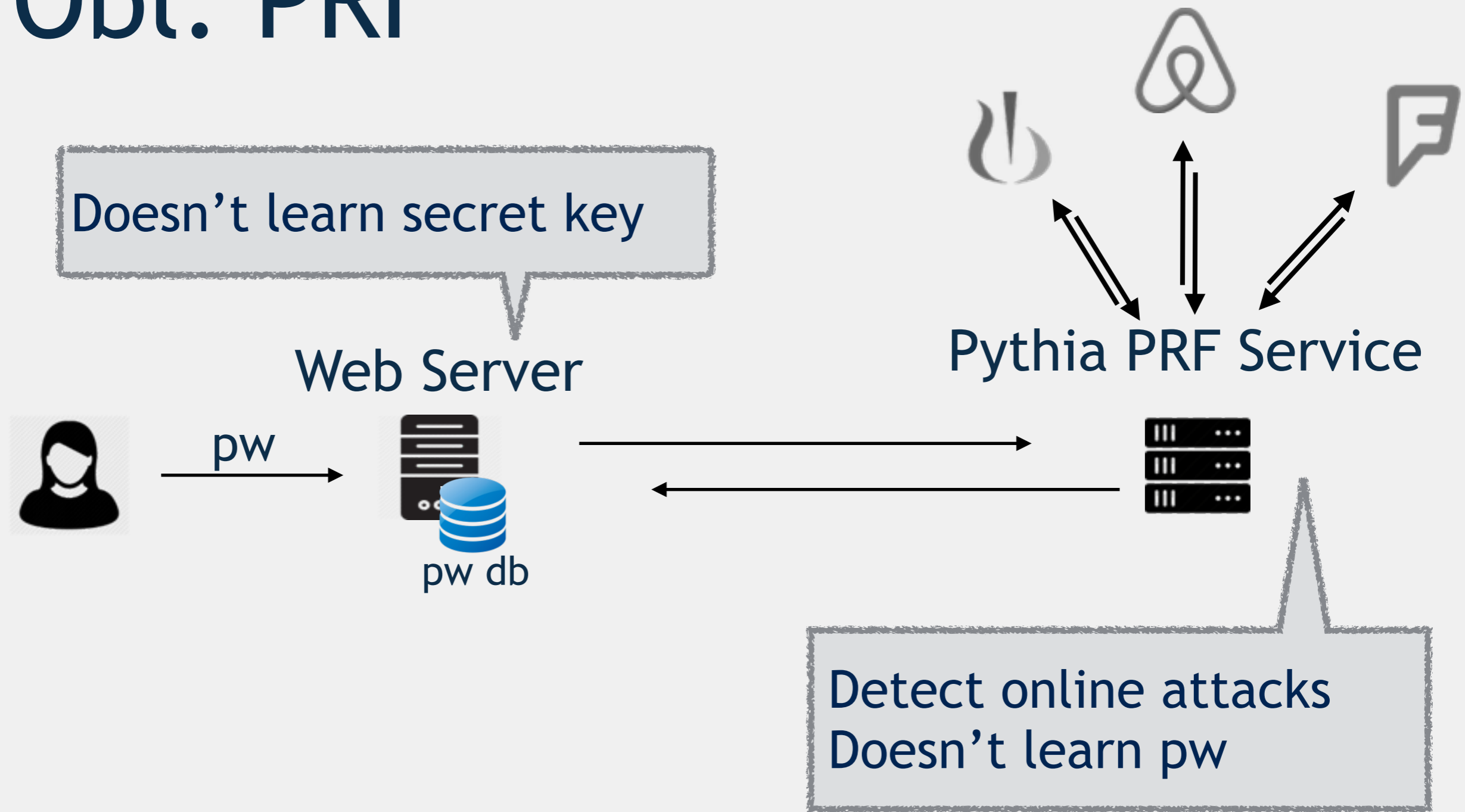
k' / k



$$z' := z^{k'/k} = e(H(t), H(pw))^{k*k'/k} = e(H(t), H(pw))^{k'}$$

update()

Advantages of Partially Obl. PRF



Easy to Deploy

```
def verify(username, pass):  
    (salt,check) = authTableLookup(username)  
    digest = hashpass(salt, pass)  
    ppass = pythia.query(server, w, t, pass)  
    digest = pythia.combine(ppass, digest)  
    return digest == check
```

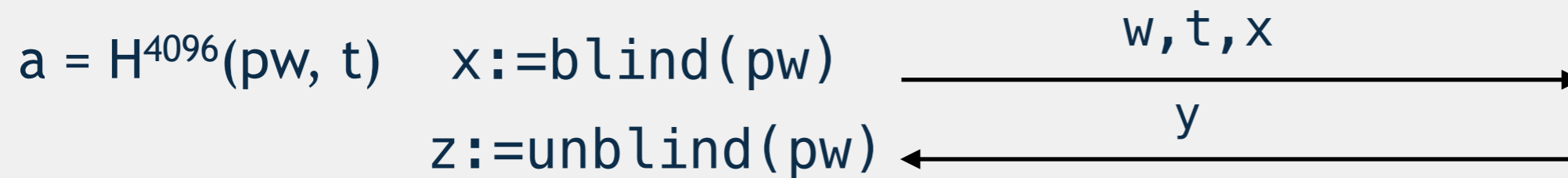
Small change to code base
No impact on user experience

Parallel Password Onion

Web Server



Pythia Server



$\text{result} := z^a$

No performance penalty
Strictly better security – Defense in depth

Pythia Open Source Implementation



nginx



MongoDB



Source code on GitHub

Find links and information at:
<https://pages.cs.wisc.edu/~ace>

Test + Development server: remote-crypto.io

Fast, Scalable PRF Service

PRF Query: 11.8ms (LAN) 96ms (WAN)

Throughput: 1350 connections/sec (8-core EC2 instance)
Within factor of 2 of a TLS query

Storage: $O(1)$ per web server
Supports arbitrary number of users
for each web server

100M Web Server: 18.6 GB (keytable)

Beyond Web Servers

**File Encryption with
remote erasure**



Message-locked Encryption



**Bitcoin
Brainwallet**

Conclusion

Password storage is **broken**: too easy to **crack** with offline attacks



Pythia PRF:

- **prevents** offline attacks, **detects** online attacks
- enables compromise **recovery** via key rotation, and crypto **erasure** via deletion
- democratizes access with a **service** architecture