Checking Passwords on Leaky Computers: A Side Channel Analysis of Chrome’s Password Leak Detection Protocol

Andrew Kwong, Walter Wang, Jason Kim, Jonathan Berger, Daniel Genkin, Eyal Ronen, Hovav Shacham, Riad Wahby, Yuval Yarom
Why Check for Compromised Passwords?

Credential Stuffing

Enabled by Default!

Change your password
A data breach on a site or app exposed your password. Chrome recommends changing your password on vault.bitwarden.com now.

OK
Chrome’s Password Leak Detection

Is Password Leak Detection secure against side-channel attacks?  
No!
The Protocol in a Nutshell

**Memory-hard Hashing:** Scrypt

**PSI Step 1:** Hash-to-curve

**PSI Step 2:** Blind with Mod. Exp.

**PSI Step 3:** Server Computes

\[(P^a)^a^{-1} = P\]

**PSI Step 4:** Unblind with Mod. Inv.

*PSI:* Private Set Intersection
The Protocol in a Nutshell

Memory-hard Hashing: Scrypt

PSI Step 1: Hash-to-curve

*PSI: Private Set Intersection

(\(P^a\))^{a^{-1}} = P

PSI Step 4: Unblind with Mod. Inv.
Let the Attacks Begin!

**PSI**: Private Set Intersection

**PSI** Step 1: Hash-to-curve

**PSI** Step 4: Unblind with Mod. Inv.

Native Adversary

---

Memory-hard Hashing: Scrypt
function scrypt(inp)
    $X = \text{init}(\text{inp})$
    for $i = 0$ to $N - 1$
        $j = \text{int}(X)$
        $\text{temp} = X \oplus V[j]$
        $X = \text{mix}(\text{temp})$

rockyou.txt  14M

 hunter2

 qwerty

123456
What About Browser-based Adversaries?

Memory-hard Hashing: Scrypt

PSI Step 1: Hash-to-curve

$Q$

PSI Step 4: Unblind with Mod. Inv.

$(P^a)^{a^{-1}} = P$

Native Adversary

Browser Adversary

*PSI: Private Set Intersection
Attacking Input-dependent Loop Iterations

```
function hash2curve(hash)
    point = RandomOracleSHA256(hash)
    while !OnCurve(point) do
        point = RandomOracleSHA256(point)
    end
```

66% Recovery
5 Traces
5 sec.

Offline: Make Dictionary

```
(hunter2, 3)
(12345678, 4)
(qwerty, 10)
```

rockyou.txt

14M
What Can a Malicious Server Do?

*PSI*: Private Set Intersection

\[(P^a)^{a^{-1}} = P\]

**Memory-hard Hashing:**
- Scrypt

**PSI Step 1:** Hash-to-curve

**PSI Step 4:** Unblind with Mod. Inv.

**Native Adversary**

**Browser Adversary**

**Native + Server Adversary**
Leaking the Blinded Point

\[
\mathbf{QP}^{\mathbf{a}} = \mathbf{P}
\]

\[
(Q^a)^{a^{-1}} = Q
\]

- hunter2 $\rightarrow$ (-1, 7)
- qwerty $\rightarrow$ (5, 0)
- abc123 $\rightarrow$ (-2, -8)

\(a^{-1}\) takes 34 ms.

\(a^{-1}\)
The Final Picture

foo = arr[secret];
Constant-time Implementations!

Thank you for listening!

Jason Kim
nosajmik@gatech.edu

Modular Inversion
Hash-to-curve
Scrypt

Native
Browser
Server