Blindfold: A System to “See No Evil” in Content Discovery

Ryan S. Peterson  Bernard Wong  Emin Gün Sirer
Cornell University
Content Discovery Landscape

- Mininova forced to remove all but a small white-list of torrents
- Pirate Bay found guilty of assistance to copyright infringement
- YouTube sued by Viacom for a billion dollars
Blindfold Goals

- Implement a publicly searchable content discovery service
  - Support non-authenticated clients

- Protect the server from the content

- Empower storage operators to operate as utility providers
Related Work

- Encrypted database search
  - Prevents the database from knowing the contents of the data while providing a search primitive over the encrypted data
  - Requires clients to have out-of-band access to the shared secret

- Not applicable to public content discovery services
Decouple the content discovery service into two components
- Index server
- Content server

Map keywords to content in a way that a human can navigate, but is difficult to automate
Implement a mapping function $f(x) \rightarrow y$

Deconstruct the function $f(x)$ into

- $f(x) = f''(\text{captcha}(f'(x)))$
- $f'(x) \rightarrow z; \text{captcha}(z) \rightarrow z'; f''(z') \rightarrow y$

Store $f'(*)$ and $f''(*)$ on two key-value servers

- Optionally in different administrative domains
Keywords: 
- Ubuntu 
- Linux 
- 9.10

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h^\alpha(\text{Fedora}) )</td>
<td>( \text{kbpsh}, h^\alpha(...) )</td>
</tr>
<tr>
<td>( h^\alpha(\text{Slackware}) )</td>
<td>( \text{3m573}, h^\alpha(...) )</td>
</tr>
<tr>
<td>( h^\alpha(\text{Debian}) )</td>
<td>( \text{vzpk2z}, h^\alpha(...) )</td>
</tr>
</tbody>
</table>

- Finding: looking for content

Index server

Content server
Upload New Content

Keywords:
• Ubuntu
• Linux
• 9.10

\[ \Phi = \text{hmac}_{\text{Ubuntu}}(\text{finding}) \]
(used to detect tempering of key-value mapping)
Content Search

3. Verify $\phi$
4. Solve captcha
7. Compute $h^{\alpha^{-1}}(\phi)$
8. Decrypt content using $h^{\alpha^{-1}}(\phi)$
Problem: Key Squatting

- A malicious user can squat on a keyword by inserting an unsolvable captcha into the index server.

- **Solution 1**
  - Index server verifies the source of captchas

- **Solution 2**
  - Store a separate user-generated captcha for each content object under the same keyword
  - Users must solve a captcha for each matching content
Fast, efficient, and non-intrusive technique for accessing data through explicit keyword search without revealing the key or the value to the server

- Enables key-value storage services to be completely blind to the content
- Provides plausible deniability to the storage operators