Talek: Private Group Messaging with Hidden Access Patterns

Raymond Cheng
Will Scott, Elisaweta Masserova, Irene Zhang, Vipul Goyal, Arvind Krishnamurthy, Tom Anderson, Bryan Parno
## Group Messaging

<table>
<thead>
<tr>
<th>Chat</th>
<th>Newsfeed</th>
<th>Calendar</th>
<th>Game</th>
<th>IoT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: message</td>
<td>1: image</td>
<td>1: new event</td>
<td>1: playerA move</td>
<td>1: config lights</td>
</tr>
<tr>
<td>2: message</td>
<td>2: tweet</td>
<td>2: delete event</td>
<td>2: playerB move</td>
<td>2: security video</td>
</tr>
</tbody>
</table>
Encryption protects the content...

<table>
<thead>
<tr>
<th>Chat</th>
<th>Newsfeed</th>
<th>Calendar</th>
<th>Game</th>
<th>IoT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:message</td>
<td>1: image</td>
<td>1: new event</td>
<td>1: playerA move</td>
<td>1: config lights</td>
</tr>
<tr>
<td>2:message</td>
<td>2: tweet</td>
<td>2: delete event</td>
<td>2: playerB move</td>
<td>2: security video</td>
</tr>
</tbody>
</table>

![Diagram showing encryption protection for various applications](image-url)
... but communication patterns are exposed
Private group messaging system for sharing data through untrusted clouds

- Hide both contents and communication patterns
- Made practical using **oblivious logging** and **private notifications**
- System with 3-4 orders of magnitude better performance than closest related work
Security Goal: Indistinguishability

Any two access sequences from a client look indistinguishable to the adversary
Security Goal: Indistinguishability

Any two access sequences from a client look indistinguishable to the adversary.
## Talek Related Work

<table>
<thead>
<tr>
<th>System</th>
<th>Security Goal</th>
<th>Threat Model</th>
<th>Technique</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Talek</td>
<td>indistinguishability</td>
<td>≥1</td>
<td>IT-PIR</td>
<td>pub/sub</td>
</tr>
<tr>
<td>Pynchon Gate</td>
<td>k-anonymity</td>
<td>≥1</td>
<td>mixnet/IT-PIR</td>
<td>email</td>
</tr>
<tr>
<td>Riffle</td>
<td>k-anonymity</td>
<td>≥1</td>
<td>mixnet/IT-PIR</td>
<td>file-sharing</td>
</tr>
<tr>
<td>Riposte</td>
<td>k-anonymity</td>
<td>≥1</td>
<td>IT-PIR</td>
<td>broadcast</td>
</tr>
<tr>
<td>Dissent</td>
<td>k-anonymity</td>
<td>≥1</td>
<td>DC-nets</td>
<td>broadcast</td>
</tr>
<tr>
<td>Vuvuzela</td>
<td>differential privacy</td>
<td>≥1</td>
<td>mixnet</td>
<td>1-1 messaging</td>
</tr>
<tr>
<td>DP5</td>
<td>indistinguishability</td>
<td>≥1</td>
<td>IT-PIR</td>
<td>chat presence</td>
</tr>
<tr>
<td>Popcorn</td>
<td>indistinguishability</td>
<td>≥1</td>
<td>C-PIR/IT-PIR</td>
<td>video streaming</td>
</tr>
<tr>
<td>Pung</td>
<td>indistinguishability</td>
<td>0</td>
<td>C-PIR</td>
<td>key-value store</td>
</tr>
<tr>
<td>ORAM</td>
<td>indistinguishability</td>
<td>0</td>
<td>ORAM</td>
<td>storage</td>
</tr>
</tbody>
</table>

**Weaker Security Goal**

**Application Specific**

**Prohibitively Expensive**
Security Goal: Indistinguishability
Any two access sequences from a client look indistinguishable to the adversary

Systems Goals:
• Mobile-friendly: 1 message per request/response
• Efficient: Thousands of online users sending a message every 5 seconds
• General Purpose: messaging and newsfeeds
• Low latency: ~5-10s
Anytrust Threat Model

- Application configured with >1 independent clouds
- Clouds logging everything about users

At least 1 non-colluding
Talek Threat Model

Anytrust: At least 1 non-colluding

Mutually distrusting users

Trusted groups
Oblivious logging enables servers to operate on noise, while delivering group messaging functionality.
Talek Overview

- Leader
- Follower
- Follower

GetUpdates

Write

Read

Client

Application

libtalek

Topic 1
Topic 2
Topic 3

write queue
write queue
read queue

publish()
subscribe()
1. How do we bound the cost of a PIR operation?
2. How do publishers write in a way that looks random?
3. How do subscribers find messages on the server?
4. How do we deal with write conflicts?
5. How do we keep all servers consistent?
Comparison to Previous Work

![Bar chart comparing throughput (requests/s) for different operations and system configurations. The chart compares Talek, Pung, and Riposte.](chart.png)
Comparison to Previous Work

Pung (OSDI 2016):
- Stronger threat model
- Uses computational PIR
Comparison to Previous Work

Riposte (Oakland 2015):
- Same threat model
- Anonymous writes by “PIR in reverse”
Scaling Clients

![Graph showing throughput vs. number of clients for different client counts: n=32K, n=131K, n=524K.](image)
https://github.com/privacylab/talek
me@raymondcheng.net