Building Confederated Web-based Services with Priv.io

Liang Zhang    Alan Mislove
Northeastern University
Online social networks
Online social networks

- OSNs are popular for content sharing
  - Facebook: 300M photos uploaded per day
Online social networks

- OSNs are popular for content sharing
  - Facebook: 300M photos uploaded per day
- Sharing is "FREE" for users
  - User does NOT pay for content sharing
Online social networks

- OSNs are popular for content sharing
  - Facebook: 300M photos uploaded per day
- Sharing is "FREE" for users
  - User does NOT pay for content sharing
- Who funds the service? -- Advertising
  - Monetizes your content for ads
  - But, we have privacy settings on Facebook!?
Privacy in OSNs

- Privacy control on OSNs
  - Control information flows within the site
  - **CANNOT** keep data private from the provider
Privacy in OSNs

- Privacy control on OSNs
  - Control information flows within the site
  - **CANNOT** keep data private from the provider

- Leads to unintended consequences
  - Easy in, (almost) no way out
    - e.g., hard to migrate data from Facebook to Google+
  - Privacy leakage
    - e.g., Facebook data bug leaked 6 million users' info
  - Big brother is watching YOU...
    - e.g., NSA, GCHQ
Privacy in OSNs

- Privacy control on OSNs
  - Control information flows within the site
  - **CANNOT** keep data private from the provider

- Leads to unintended consequences
  - Easy in, (almost) no way out
    - e.g., hard to migrate data from Facebook to Google+
  - Privacy leakage
    - e.g., Facebook data bug leaked 6 million users' info
  - Big brother is watching YOU...
    - e.g., NSA, GCHQ

- Can we protect user privacy *from the provider?*
Alternative designs
Alternative designs

- Encrypt data uploaded to the provider (e.g., Privly, NOYB)
  - Require additional software installed, low accessibility
  - Transfer cost to OSN providers, not sustainable
Alternative designs

- Encrypt data uploaded to the provider (e.g., Privly, NOYB)
  - Require additional software installed, low accessibility
  - Transfer cost to OSN providers, not sustainable
- User-hosted servers for private data (e.g., Persona, Vis-à-Vis)
  - Expensive for user to host server
Alternative designs

- Encrypt data uploaded to the provider (e.g., Privly, NOYB)
  - Require additional software installed, low accessibility
  - Transfer cost to OSN providers, not sustainable
- User-hosted servers for private data (e.g., Persona, Vis-à-Vis)
  - Expensive for user to host server
- Decentralized system (e.g., PeerSoN, Diaspora)
  - Reliability difficult to achieve
Alternative designs

- Encrypt data uploaded to the provider (e.g., Privly, NOYB)
  - Require additional software installed, low accessibility
  - Transfer cost to OSN providers, not sustainable
- User-hosted servers for private data (e.g., Persona, Vis-à-Vis)
  - Expensive for user to host server
- Decentralized system (e.g., PeerSoN, Diaspora)
  - Reliability difficult to achieve
- Our insight: Leverage cloud computing to host user content
  - Users store encrypted data on cloud provider of choice
  - But, how much would it cost?
Using the Cloud
Using the Cloud

- Storage
- Bandwidth
- Requests
- Computation
# Using the Cloud

<table>
<thead>
<tr>
<th>Service</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td>$0.095/GB/month for storage</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>$0.12/GB for outgoing bandwidth</td>
</tr>
<tr>
<td>Requests</td>
<td>$0.004 per 10,000 GET requests</td>
</tr>
<tr>
<td>Computation</td>
<td>$14.40 per month for a t1.micro instance</td>
</tr>
</tbody>
</table>

* Prices are based on Amazon cloud platform.*
Using the Cloud

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td>$0.095/GB/month for storage</td>
<td>✔️</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>$0.12/GB for outgoing bandwidth</td>
<td>✔️</td>
</tr>
<tr>
<td>Requests</td>
<td>$0.004 per 10,000 GET requests</td>
<td>✔️</td>
</tr>
<tr>
<td>Computation</td>
<td>$14.40 per month for a t1.micro instance</td>
<td>❌</td>
</tr>
</tbody>
</table>

* Prices are based on Amazon cloud platform.
Using the Cloud

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td>$0.095/GB/month for storage</td>
<td>✓</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>$0.12/GB for outgoing bandwidth</td>
<td>✓</td>
</tr>
<tr>
<td>Requests</td>
<td>$0.004 per 10,000 GET requests</td>
<td>✓</td>
</tr>
<tr>
<td>Computation</td>
<td>$14.40 per month for a t1.micro instance</td>
<td>✗</td>
</tr>
</tbody>
</table>

* Prices are based on Amazon cloud platform.

- If we ignore computation,
  - **Cost for 99% users is less than $1**
  - Using real world data (Facebook, Twitter, Flickr)
  - More details in paper
Priv.io
Priv.io

- Goal: low cost platform for web services with strong user privacy
Priv.io

- Goal: low cost platform for web services with strong user privacy
- Key insights:
  - User provides storage, bandwidth via cloud providers
    - Protects privacy, provides control
  - Use users' web browsers for computation
    - Provides cost-efficient computation
Priv.io

- Goal: low cost platform for web services with strong user privacy
- Key insights:
  - User provides storage, bandwidth via cloud providers
    - Protects privacy, provides control
  - Use users' web browsers for computation
    - Provides cost-efficient computation
- Result: Priv.io, a **confederated** service
  - Each user retains control over his/her own data
  - Confederated means users are free to leave
Outline

- Motivation
- Priv.io design
- Security, privacy and limitation
- Evaluation
Sharing on Facebook
Sharing on Facebook
Sharing on Facebook
Sharing on Facebook
Sharing on Priv.io
Sharing on Priv.io
Sharing on Priv.io

priv.io

Priv.io

Liang

Liang’s storage

amazon web services
Sharing on Priv.io

Priv.io

Liang's storage

Alan's storage

Liang

Alan

Dropbox
Priv.io overview

- **Social platform** for building web apps
  - e.g., Google Doc, Facebook, Twitter

- **Architecture**
  - Servers
    - Server side support
    - User-contracted cloud providers
  - Priv.io Core
    - Kernel of the system
  - Priv.io Applications
    - User facing functionality
Servers

- Priv.io server
  - Bootstraps Priv.io
    - Serves static content
  - Uses DNS to map cloud providers
    - e.g., liang.priv.io => liang.priv.io.s3.amazonaws.com
    - Hide users traces
- Cloud providers
  - Assumption: Accessible with REST API
  - Storage
    - Two credentials: owner read/write, friends read
  - Providers today: Amazon, Google, Azure, Dropbox
Priv.io core

- Run applications
  - Ensures security, privacy
- Resource management
  - Access user provided resources
  - Easy encryption/decryption (ABE, AES)
- Content sharing
  - Create, manage friendship
  - Access own, friends storage
- Expose services to applications via API
Priv.io application model
Priv.io application model

Liang

Newsfeed

Core

amazon web services

/app_1/
/app_2/
/newsfeed/

Alan

Newsfeed

Core

alan.priv.io

/newsfeed/
Priv.io application model
Priv.io application model

Priv.io API

Permission
- requestPermissions
User Information
- getUsername, getFriends
Storage
- store, retrieve
Communication
- send, receive, delete
Priv.io application

- Implemented in HTML5
  - Runs in users' browsers
  - Each app gets its own iframe
- Various applications
  - Less social interaction: Google doc
  - More social interaction: Facebook newsfeed
- Hosting applications
  - Applications are served on Priv.io server
  - Access via subdomain, e.g., newsfeed.app.priv.io
Application: Newsfeed
Application: Newsfeed
Application: Newsfeed

Priv.io

newsfeed.app.priv.io

Liang
Application: Newsfeed

```
<table>
<thead>
<tr>
<th>Priv.io</th>
</tr>
</thead>
<tbody>
<tr>
<td>liang.priv.io</td>
</tr>
<tr>
<td>/app_1/</td>
</tr>
<tr>
<td>/newsfeed/</td>
</tr>
<tr>
<td>/app_2/</td>
</tr>
</tbody>
</table>
```

DNS: liang.priv.io.s3.amazonaws.com
Application: Newsfeed

Priv.io

Amazon Webservices

/app_1/

/newsfeed/

feeds/82dda1
liang/37bb2a
alan/91de7d
liang/695a32

Liang

Hello everyone! Leave a comment below so that my future friends can see you.

Alan Mislove
Looks like fun!

Jonathan Schuster
Hello, everyone.

Arash Mostavi
What up?
Application: Newsfeed

Priv.io

amazon web services™

/lapp_1/    /newsfeed/

/comments/37bb2a
liang/37bb2a
alan/91de7d
liang/695a32

/comments/695a32
What's up?

Liang Zhang @liang
Hello everyone! Leave a comment below so that my future friends can see you :)

Alan Mislove @zblowe
Looks like fun!

Jonathan Schuster @schuster
Hello, everyone.

Arash Molavi @barash
what up?

Liang

374d ago
Application: Newsfeed

Priv.io

liang.priv.io

amazon web services

/app_1/

/app_2/

/newsfeed/

feeds/82dda1
liang/37bb2a
alan/91de7d
liang/695a32

comments/37bb2a
Hello everyone! Leave a comment below so that my ...

comments/695a32
What’s up?

alan.priv.io

/newsfeed/

comments/91de7d
Looks like fun!

Liang

Liang Zhang @Liangle6
Hello everyone! Leave a comment below so that my future friends can see you.

Alan Mislove @amislove
Looks like fun!

Jonathan Schuster @jschuster
Hello, everyone.

Arash Molavi @barash
what up?

Alan Mislove @amislove
Looks like fun!
Outline

- Motivation
- Priv.io design
- Security, privacy and limitation
- Evaluation
Security: can app bypass API?
Security: can app bypass API?

- Newsfeed
- devil
- var passwd
- API
Security: can app bypass API?
Privacy: leak user data?
Privacy: leak user data?

Content Security Policy (CSP)

devil.com
Limitations

• No global view
  - e.g., no global search
  - Some can be partially replicated with local view
    - e.g., friend suggestion
• Computation only in browser
  - Don't have background processes
  - Push notification
  - Future research
Outline

- Motivation
- Priv.io design
- Security, privacy and limitation
- Evaluation
Evaluation overview

- Prototype system
  - Supports Amazon SQS and S3
  - Runs latest common web browsers (desktop and mobile)
  - 5,931 lines of JavaScript

- How much overhead from encryption?
  - Microbenchmarks on running time
  - AES: 100K object: under 43ms (desktop), 327ms (mobile)
  - **Provide decent performance**
  - ABE more expensive, but much less frequent
How's user-perceived performance?

- Load 15 Newsfeed items
- Feed loading time:
  - below 515ms (desktop), 5.1s (mobile)
- Comparable to today's OSN services on desktop
How does Priv.io work in practice?

- Deploy within our department for two months
  - 28 graduate students and professors
  - 88 friendships, an average 3.82 friends per user
  - Post 221 items

- It works on today's browsers (desktop and mobile)
Summary

- Confederated platform for building Web-based services
- Leverage:
  - Cloud providers for storage, bandwidth, and messaging
  - User’s Web browser for computation
- In Priv.io, users
  - Retain control of their own data
  - Keep data privately from the service provider
  - Enjoy a highly reliable and available service
- Result:
  - Work with today's web browsers
  - Newsfeed: Facebook alike application
Thank You!

Questions?

https://priv.io/

https://github.com/LeoLiangZhang/Priv.io