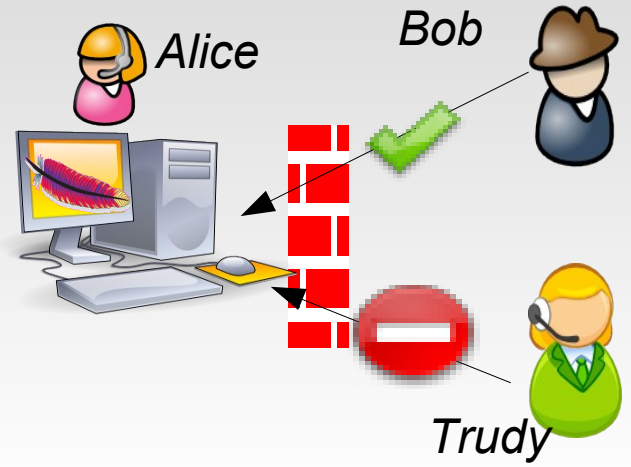
Authenticating Out-of-Band Communication Over Social Links

Anirudh Ramachandran and Nick Feamster
School of Computer Science, Georgia Tech
{avr,feamster}@cc.gatech.edu

Motivating Application: Secured Web Server

- Alice wishes to set up a *secure* web service to share her photos *only* with her friends. She must



1. Distribute the URL of the service to her friends
   - *What if the server is unreachable or its IP / location changes (e.g., DHCP)?*
2. Create and distribute credentials for the service to each friend
   - Email/IM: *What if she wants to add or revoke friends?*

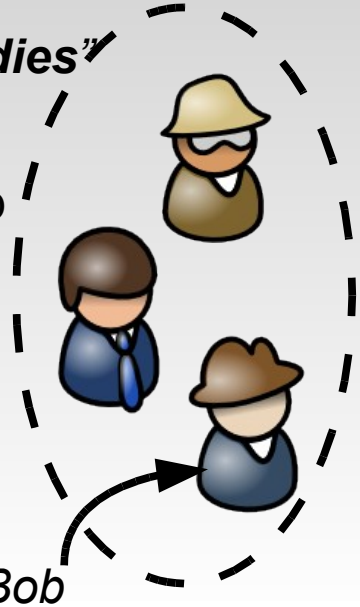
How can Online Social Networks help?

- Social networks store and manage a user's friends
  - Expresses real-life relationships online
- Security based on social relationships is exactly what many applications need
- *Challenge*: How can we leverage relationships on OSNs for securing inter-app communication?

Securing a Web server using OSNs

1. 
Log in
facebook
Alice

2. 
Create friend group



3. 
Publish service details
to group

Group Name:*Buddies*
Permissions: *Members Only*
Alice's server IP: *100.1.1.1*
Alice's password: "*insecure*"
Status: *Online*

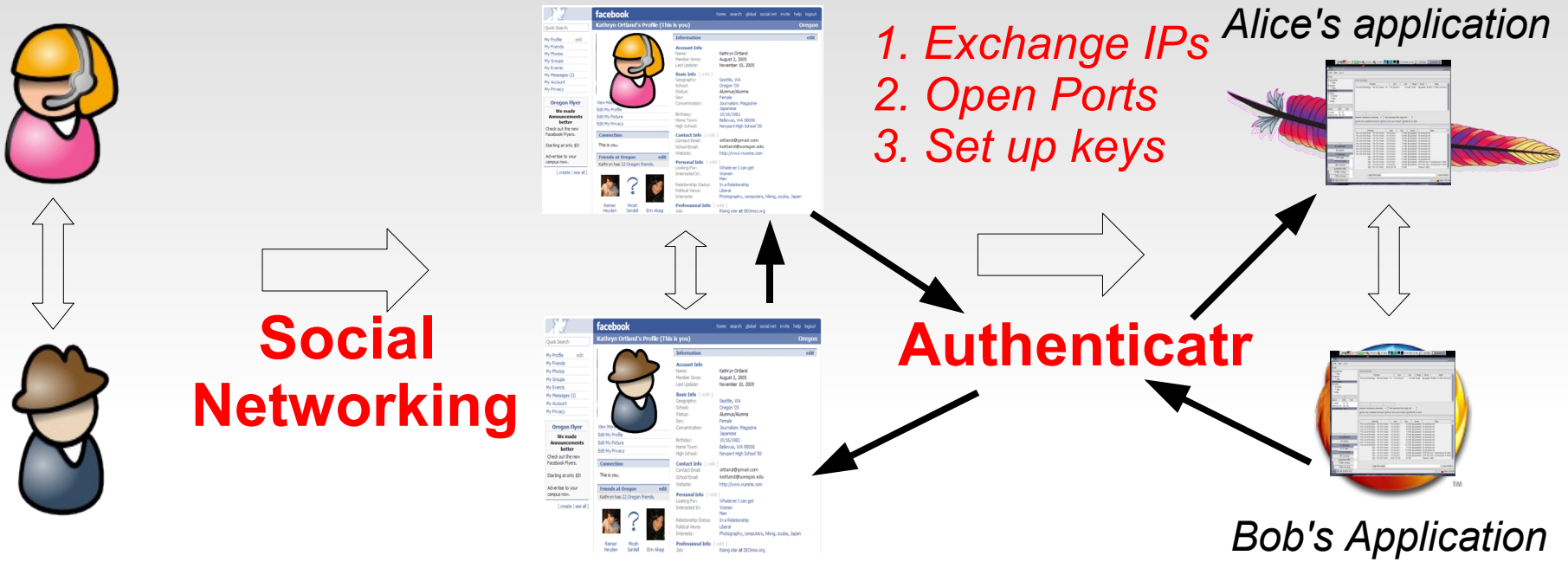
4. 
Get Alice's server credentials
facebook

5. 
Present credentials to Alice's server


Alice's Server

If only applications could do this automatically...

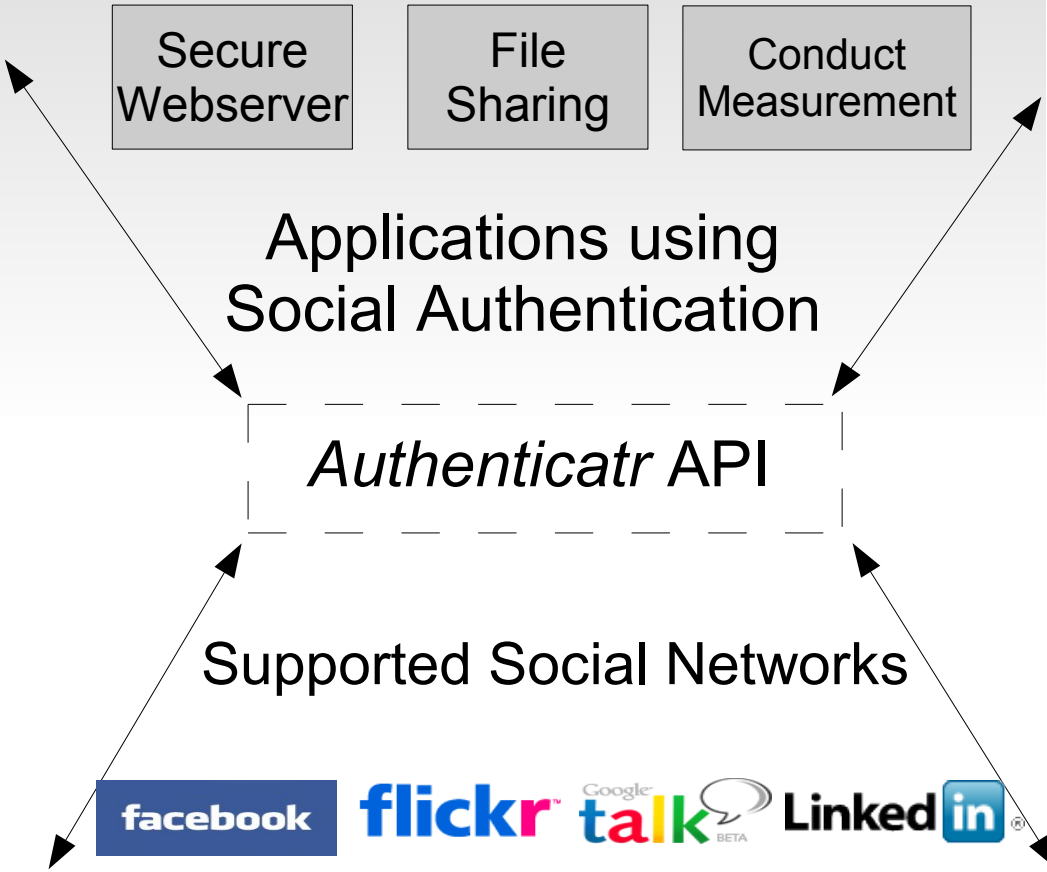
Our Contribution: *Authenticatr*



- Requirements for the social networ*k*
  - The social network must be *authenticated*
  - It must support *basic messaging* between friends

Design Overview

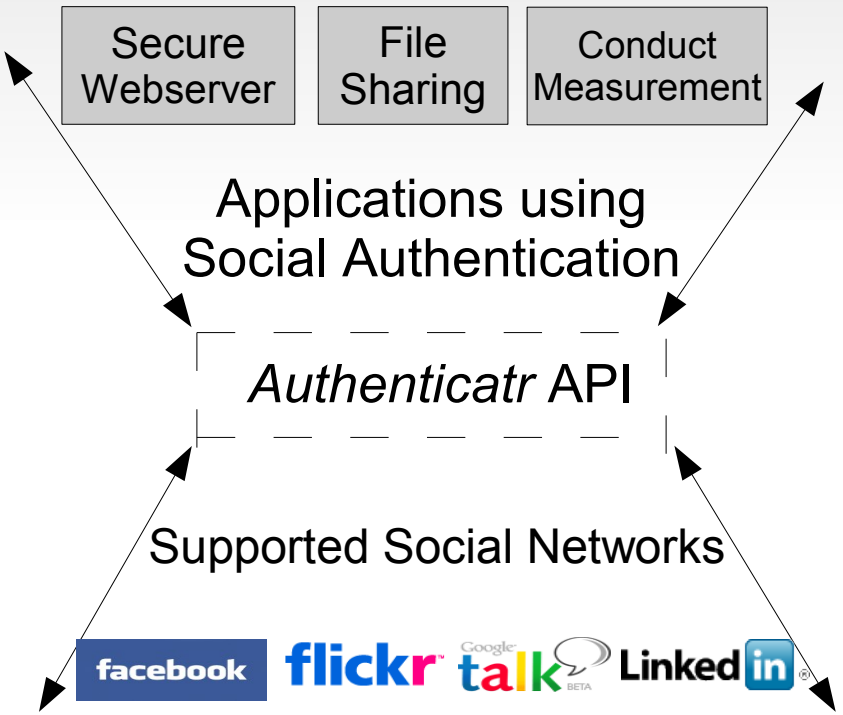
- Three components
  - A set of applications that can use *social context* for authentication
  - A set of social communication protocols
  - An API that exports a uniform interface to all applications.



"Hourglass" design

Outline

- Motivating Application #2: P2P file sharing
- Authenticatr API
- More Applications!
- Ongoing and Future Work
- Related Work
- Summary



Motivating Application #2: P2P sharing

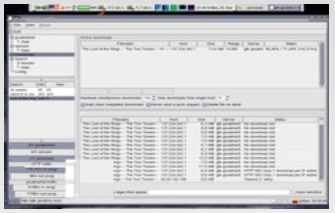
- Alice wishes to *securely* share large files with some of her friends
  - ✗ *Send it via email or IM*: file size limits; Alice must initiate each file transfer; friends cannot be added or removed
  - ✗ *Share it on a P2P network (e.g., Gnutella)*: No security (or Alice must password-protect the files, and distribute the file names and keys to each friend)
- All of peer discovery, secure communication, and scalability are difficult to achieve

P2P Filesharing with Authenticatr

Alice

Present credentials

facebook



Log In

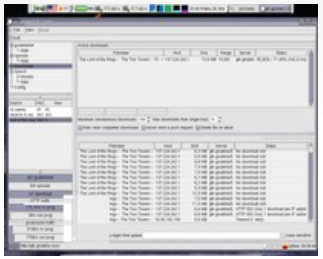
facebook

Retrieve list of friends

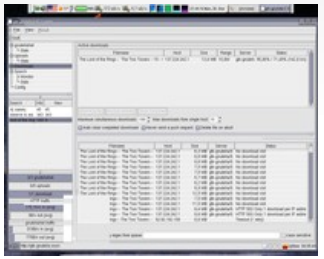
P2P client over
Authenticatr

Choose files to share

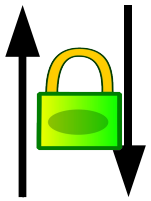
Choose friends



Wait for connections



Exchange IPs, password, etc.
using social messaging



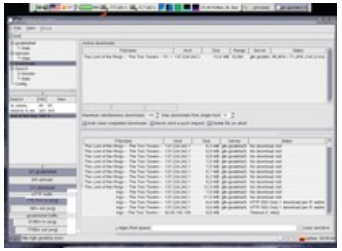
facebook

Present credentials
Initiate direct connection

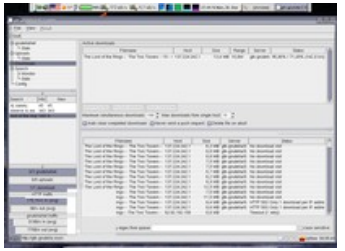
Bob

Present credentials

facebook



Bob's P2P client



Authenticatr API

	Goal	Function Prototype
Authentication API	Attempt to log onto network `n`, returning a session handle	`session* login (network *n, credential *cred)`
Communication API	Send an opaque message `msg` to friend `f` using session `s`	`send (session *s, friend *f, message *msg)`
	Receive opaque message `msg` from friend `f` over session `s`	`recv (session *s, friend *f, message **msg)`
Utility functions	Get the list of friends of user `f` from session `s` as the list `l`	`get_friends (session *s, friend *f, friend **l)`

More Applications!

- Alice wants to *conduct a network measurement* from Bob's computer
  1. Alice's application logs in and inspects Bob's profile to see if his application is active
  2. Alice's app sends a message to Bob such as "`ping google.com`"
  3. Bob's app picks up the message, conducts the experiment, and sends the result back as another message
- May be used for *root-causing network disruptions*

Alice's application:

```
s = login (facebook,
                cred_alice);
get_friends_list (s, NULL,
              &friend_list);
 send (s, friend_list[1],
       "ping google.com");
```

Periodically:

```
  recv (s, friend_list[1],
           &meas_response);
```

Bob's application:

```
 recv (s, friend_list[2],
              &meas_request);
  // Perform measurement
  send (s, friend_list[2],
              meas_response);
```

Application: Key exchange

- Alice and Bob want to *negotiate a shared secret*

1. Alice and Bob set up *Diffie-Hellman* parameters in a set of messages over the social network of choice
2. Using D-H, a key can be established in one more roundtrip

Practical Considerations

- *Changes to host applications*: mainly user input
  - Retrieves user/pass from social network instead of prompting the user
- *Session Multiplexing*: many application instances must use one social network session
  - Each message passed on the social network contains identifying tags (similar to an object broker)

Ongoing and Future Work

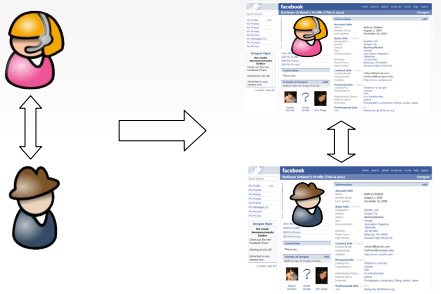
- Two applications: secured web service and a P2P filesharing service
- Two social networks: Google Talk and Facebook
- Challenges:
  - Facebook does not provide a way for desktop applications to send or receive messages
    - Using *notifications* as a hack
    - Can only get “unread” notifications
  - Message ordering/timestamping, locking
- Discussion topic: Wishlist for OSN APIs?

Related Work

- *OpenSocial*: Attempts to unify social networks for *web-based* applications
  - Authenticatr unifies social networks for *desktop* apps; also can work across IM, mailing lists, etc.
- *Lockr:* Attempts to reuse social relationships from one DB/service on other services for access control
- *SocialGraph*: Similar goal, except it uses publicly declared relationships (no security)
  - Authenticatr does not try to combine two social networks; provides a uniform interface for each (to apps)
- *FriendStore*, *Pownce*: Share files within friend networks
  - Authenticatr extends and generalizes this idea

Summary

- Many desktop applications could benefit from secure communication
  - Many, however, forsake it for usability
- Social networking channels offer a secure messaging path to initiate authentication



  - Implements real-world trust relationships online
- *Authenticatr* allows desktop applications to use these social channels for authentication

Thanks!

- Coffee, anyone?