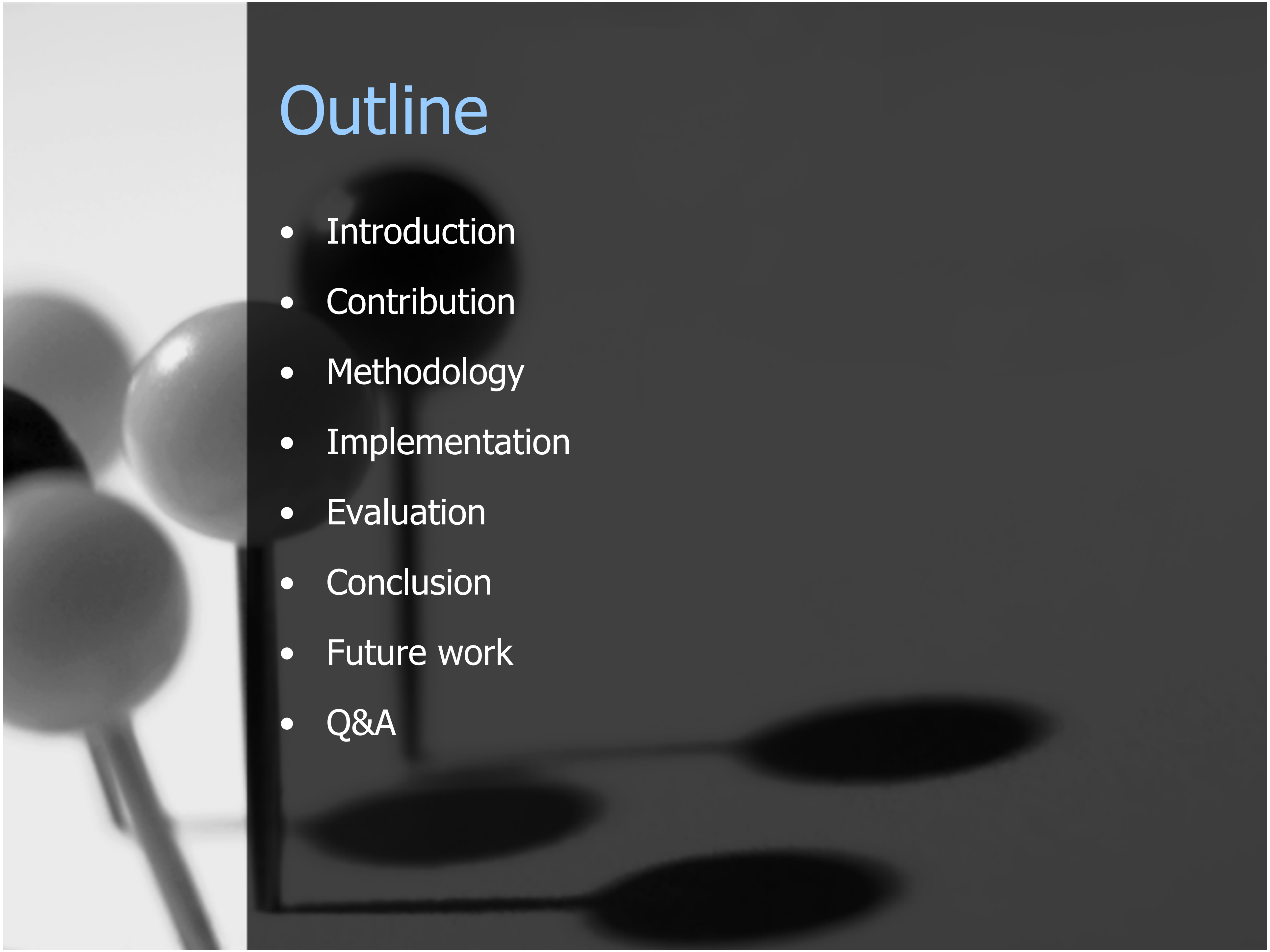
Distributed and Secure Social Network Mobile Application with Id-based crypto and Cloud storage service

Tien Dat Le



Outline

- Introduction
- Contribution
- Methodology
- Implementation
- Evaluation
- Conclusion
- Future work
- Q&A

OSN background

- Online social network (OSN): a system where
  - Users are main entities with profiles
  - Users can create links to others users and resources
  - Users can navigate the social network by browsing the profiles and resources link

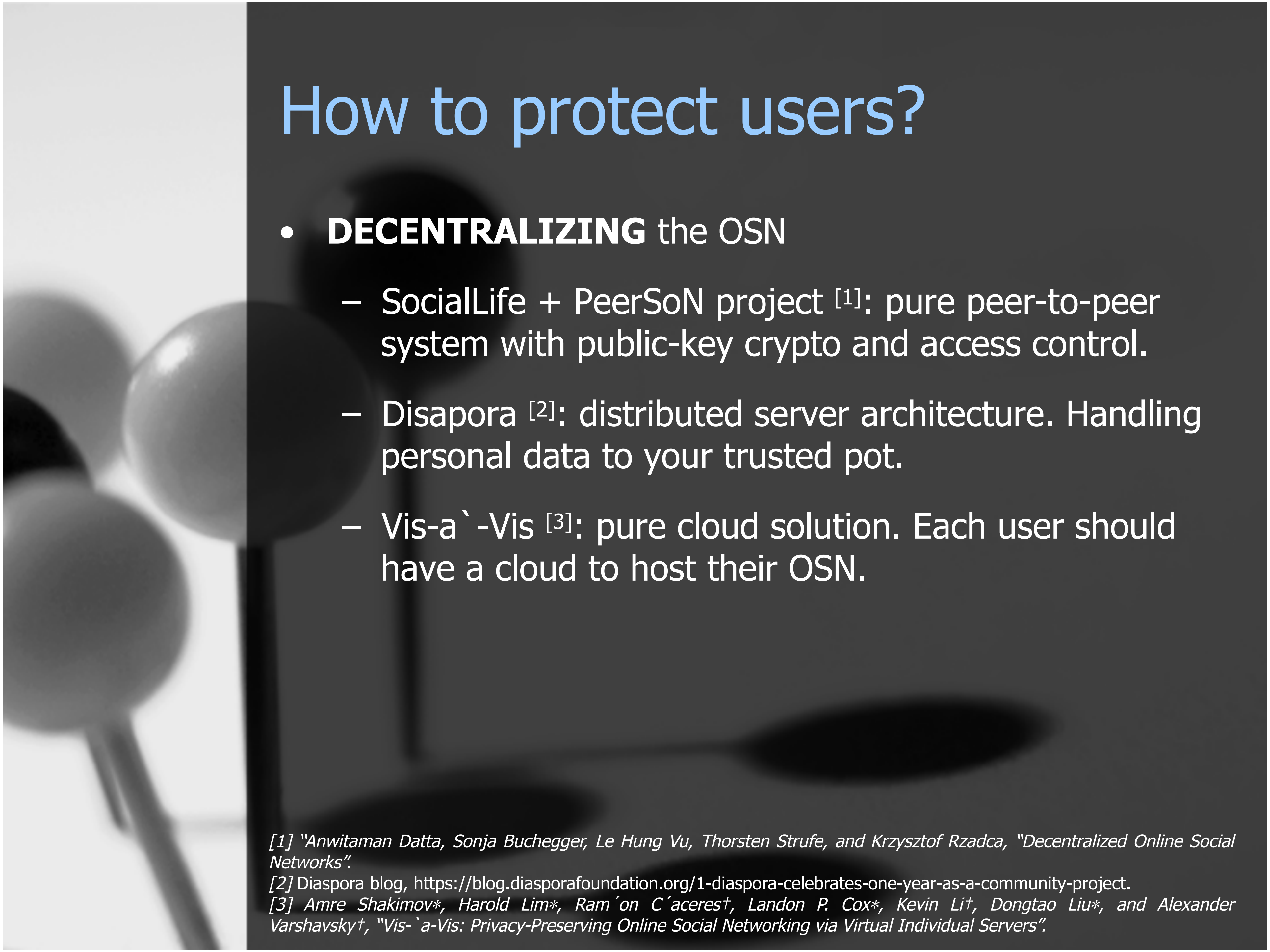
Loss of control, Lack of privacy

- OSN providers monetize by selling your privacy to the marketer [1]
- The trending worsen: constantly change in privacy policies to diminish the right of users to control their data [2]
- Privacy violation by manipulating the data flow to users [3]

[1] "Facebook, MySpace confront privacy loophole," The Wall Street Journal, May 2010.

[2] https://www.eff.org/deeplinks/2010/04/facebook-timeline.

[3] http://www.latimes.com/nation/nationnow/la-na-nn-facebook-research-20140629-story.html



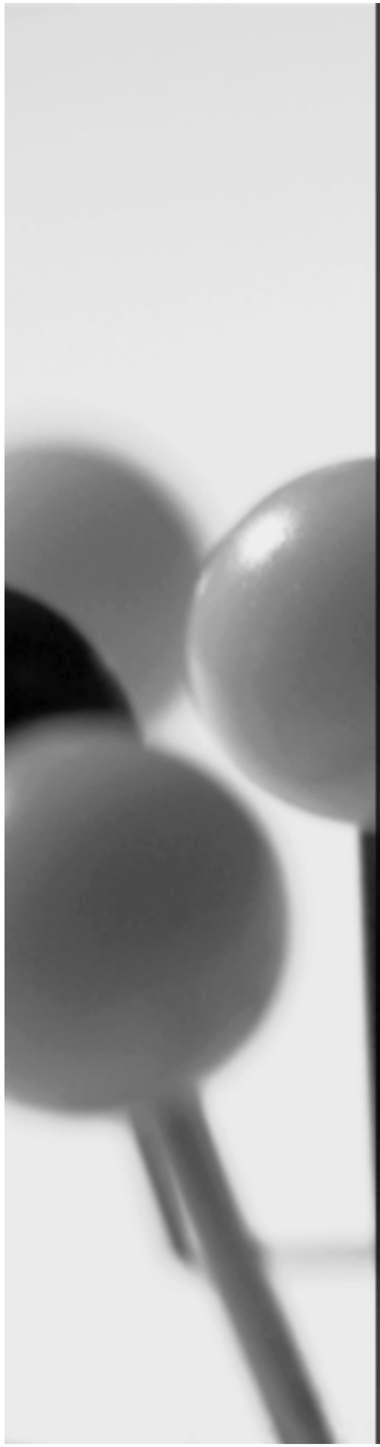
How to protect users?

- **DECENTRALIZING** the OSN
  - SocialLife + PeerSoN project [1]: pure peer-to-peer system with public-key crypto and access control.
  - Disapora [2]: distributed server architecture. Handling personal data to your trusted pot.
  - Vis-a`-Vis [3]: pure cloud solution. Each user should have a cloud to host their OSN.

[1] "Anwitaman Datta, Sonja Buchegger, Le Hung Vu, Thorsten Strufe, and Krzysztof Rzadca, "Decentralized Online Social Networks".

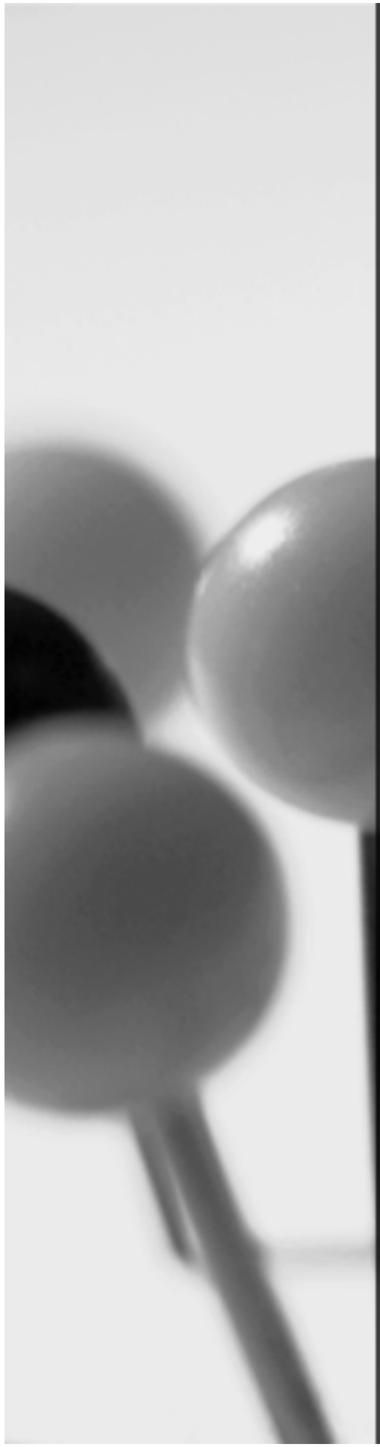
[2] Diaspora blog, https://blog.diasporafoundation.org/1-diaspora-celebrates-one-year-as-a-community-project.

[3] Amre Shakimov∗, Harold Lim∗, Ram´on C´aceres†, Landon P. Cox∗, Kevin Li†, Dongtao Liu∗, and Alexander Varshavsky†, "Vis-`a-Vis: Privacy-Preserving Online Social Networking via Virtual Individual Servers".



Challenges still remain

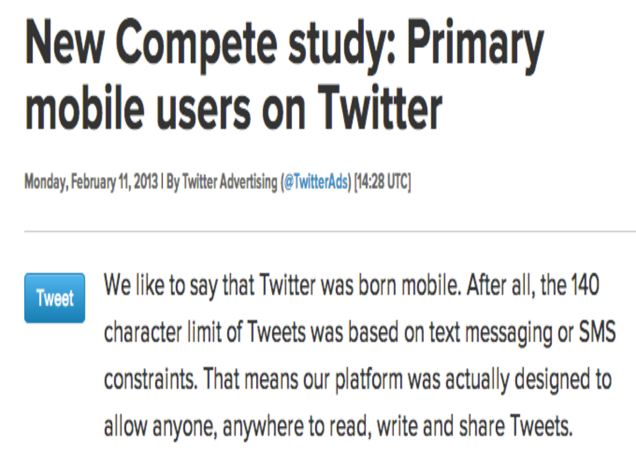
- SocialLife + PeerSoN project:
  - requires users to manage trust and certificates by their own
- Disapora:
  - What if your trusted pod server is evil?
  - A few dominated pods could become centralization.
- Vis-a`-Vis:
  - COST for their own cloud.



Mobile trending – more challenges

- Twitter have revealed a new complete study for mobile users as they are the main users [1]

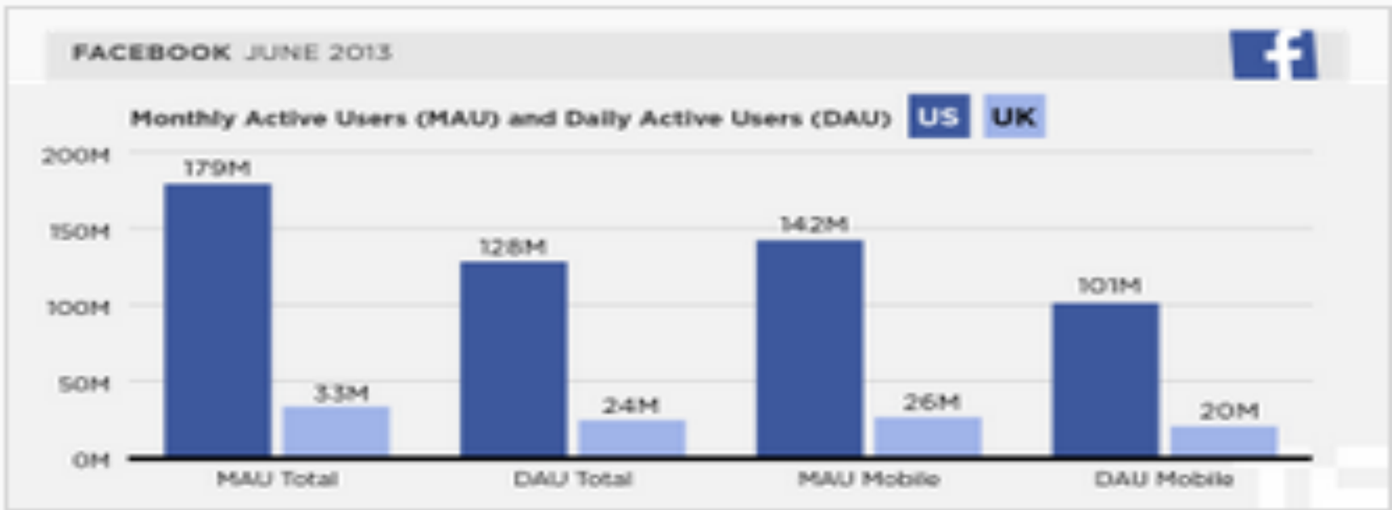
- Due toTech crunch, facebook reveals 78% of US users are mobile [2]



New Compete study: Primary mobile users on Twitter

Monday, February 11, 2013 | By Twitter Advertising (@TwitterAds) [14:28 UTC]

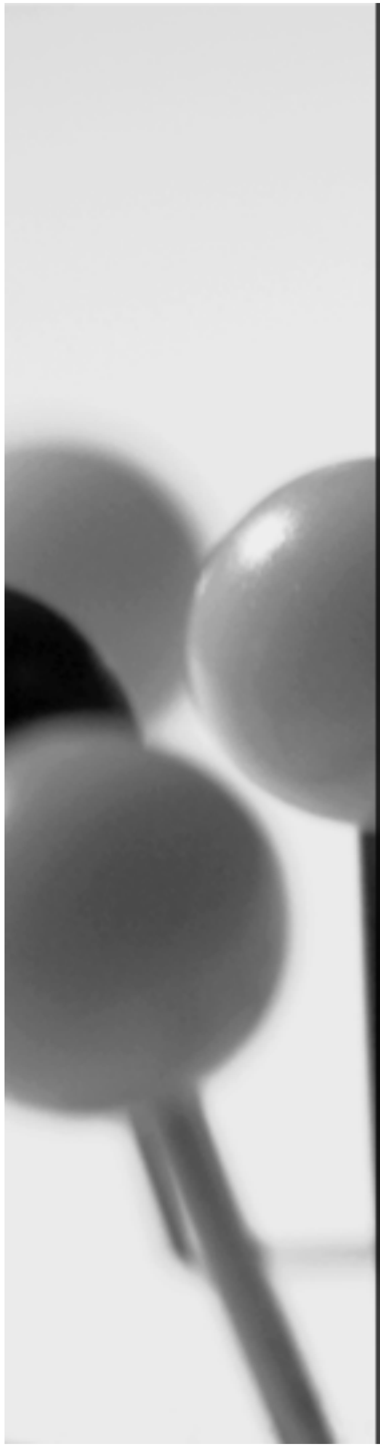
We like to say that Twitter was born mobile. After all, the 140 character limit of Tweets was based on text messaging or SMS constraints. That means our platform was actually designed to allow anyone, anywhere to read, write and share Tweets.



FACEBOOK JUNE 2013

Monthly Active Users (MAU) and Daily Active Users (DAU)

[1] https://blog.twitter.com/2013/new-compete-study-primary-mobile-users-on-twitter
[2] http://techcrunch.com/2013/08/13/facebook-mobile-user-count/

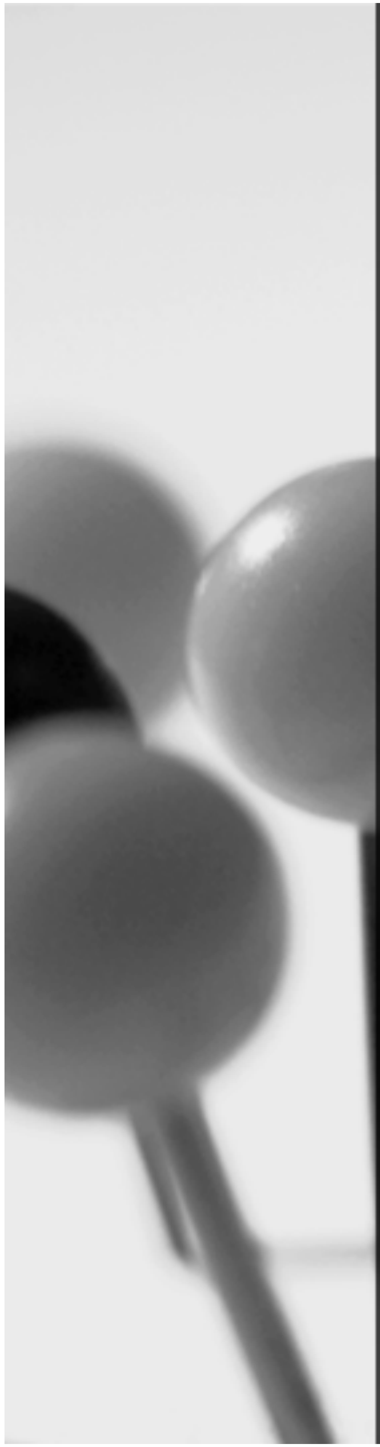


Why mobile trending challenge

- Energy consumption and computing resources constrains
- Less resources to contribute to the p2p network
- No security tools to manage your certificate
- More chances to lose top private secret keys with the mobile

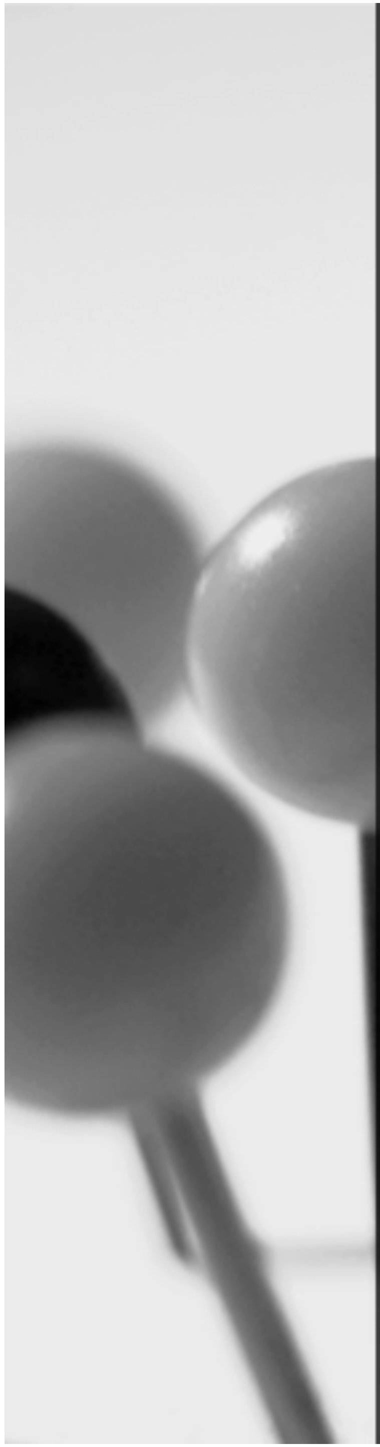
That means:

- Not likely to prefer computing redundancy security scheme for strangers
- Serving hot-spot contents killing your battery
- Small trusted p2p networks is not enough resources



How to solve problems

- The resources constrains problems:
  - Separating storage service, security provider service and communication service
  - Outsourcing heavy-weight resources demand activities to cloud computing
  - Only keep core activities on the mobile p2p network
- Security problems:
  - Top secrets should be kept separately on different trusted provider
  - Only session keys on the mobile

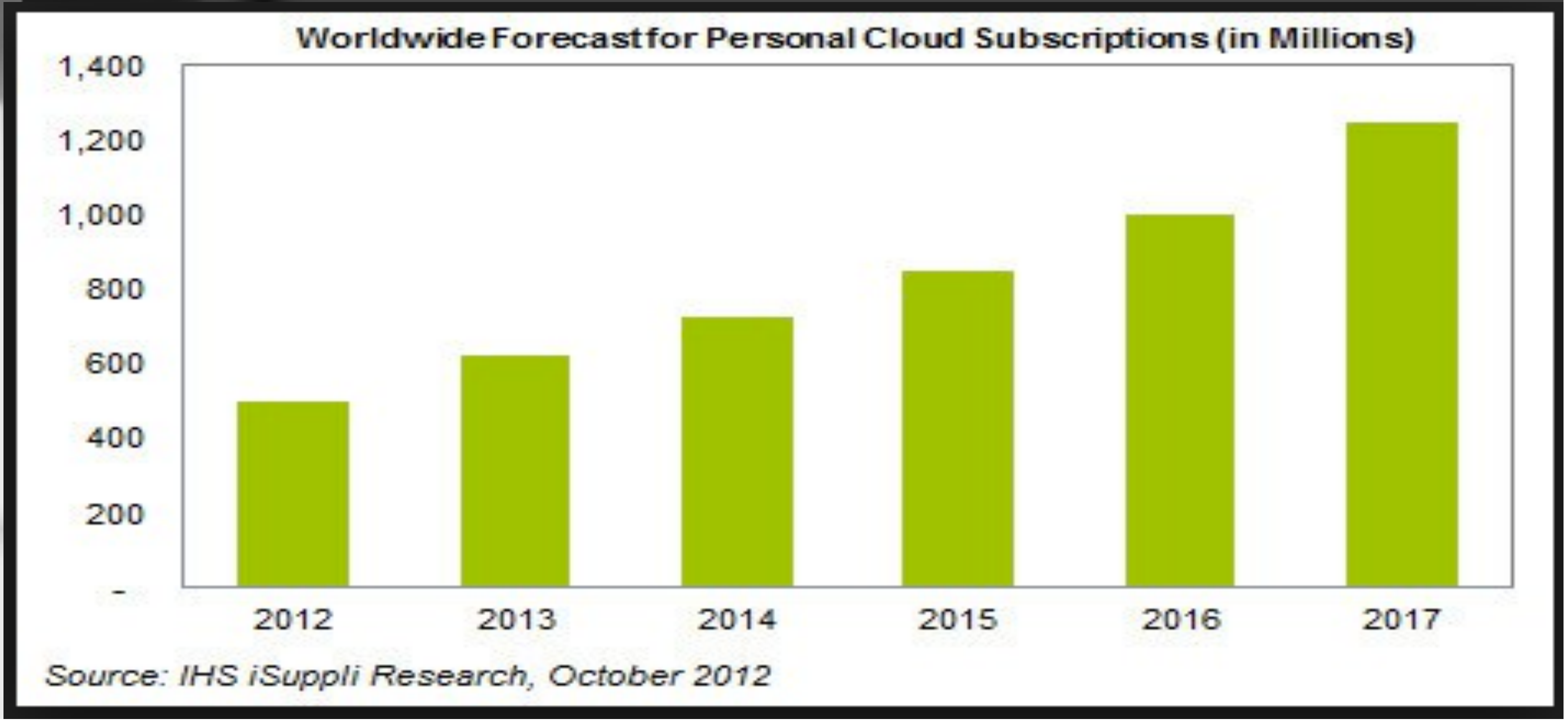


Our approach

- Lightweight communication system suitable for mobile.
- Put access control burden to semi-trusted CLOUD STORAGE SERVICE
- Deploying IDENTITY-BASED CRYPTO for privacy
- By separating storage service and security service, you could:
  - Encrypted data to sensor it from cloud provider
  - Exploit access control list of data storage service to distribute session key and mitigate key-escrow problem of identity-based crypto.

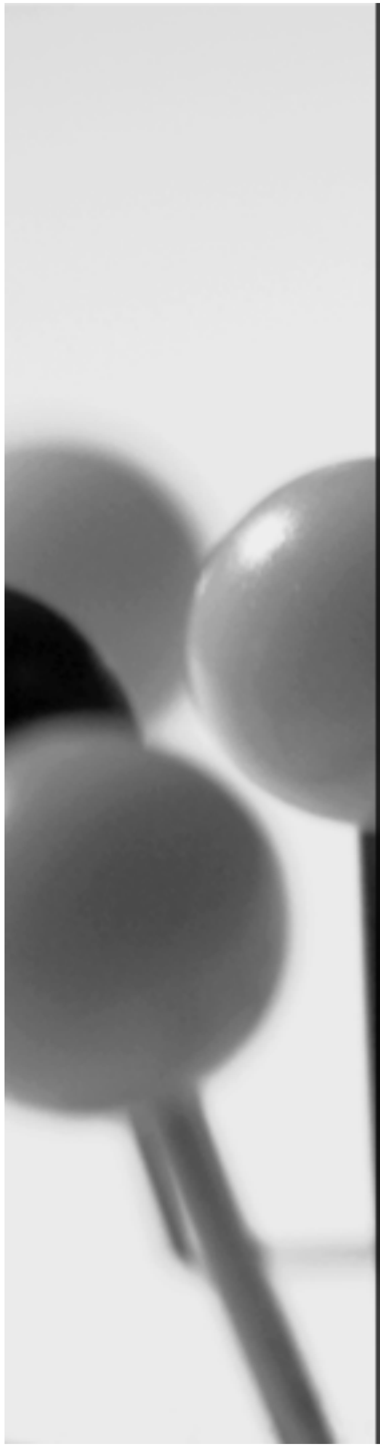
Why cloud storage service?

- FREE (For personal usage is enough)
- Familiar with people
- Trending technology
- High availability
- Multi-platform API support
- Access control support


Worldwide Forecast for Personal Cloud Subscriptions (in Millions)

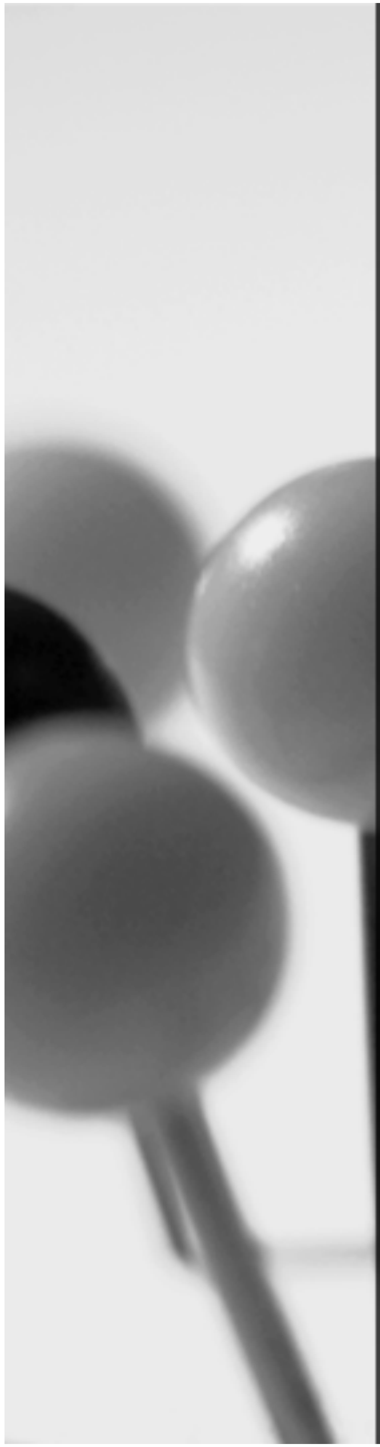
Source: IHS iSuppli Research, October 2012

Why identity-based crypto?



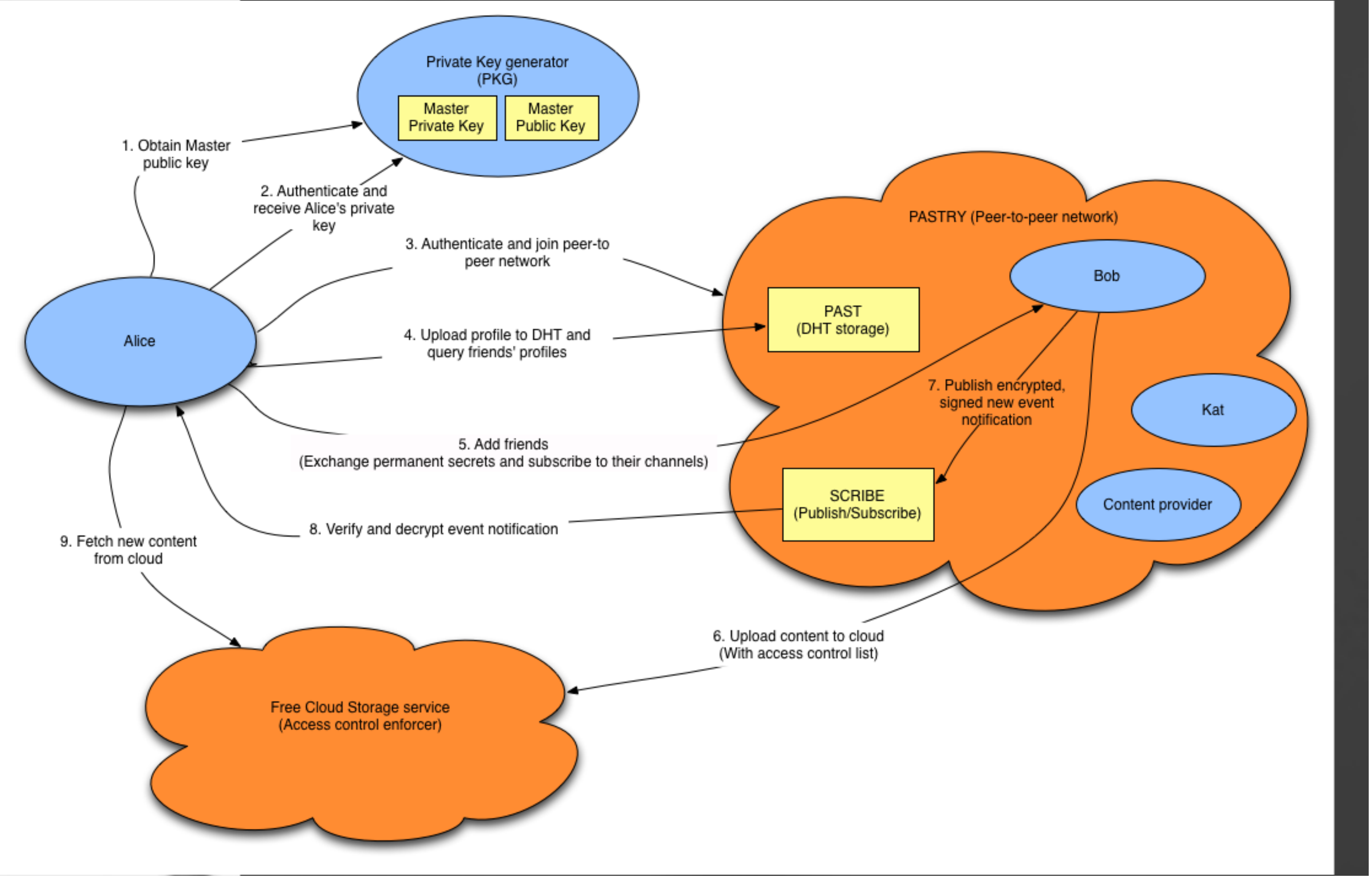
- Simplizing CA infrastructure to private-key generator server
- Remove the burden of managing certificate
- Support online identity naturally
- More flexible than public key crypto in BROADCAST encryption (Group key revoking easier)

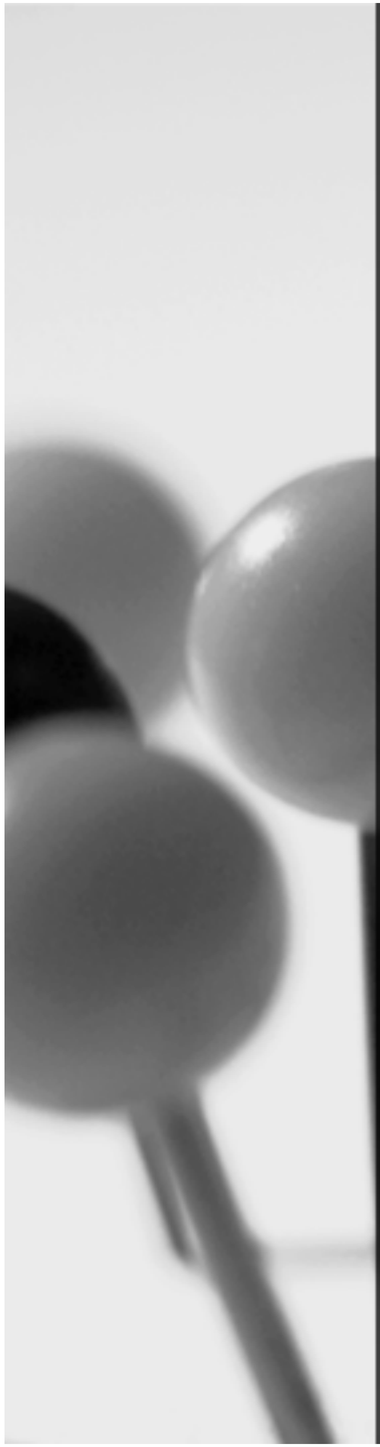
Concept components



- Deploying structure peer-to-peer PASTRY network as backbone for peer communication:
  - SCRIBE publish/subscribe system for multicasting notification and event.
  - PAST DHT storage system for profile searching and indexing.
- Define cloud storage interface for the architecture for data storage and data control access (proof-of-concept version work for GoogleDrive)
- Deploy Identity-based cryptography for authentication and key distribution scheme with JPBC library for java

Application model





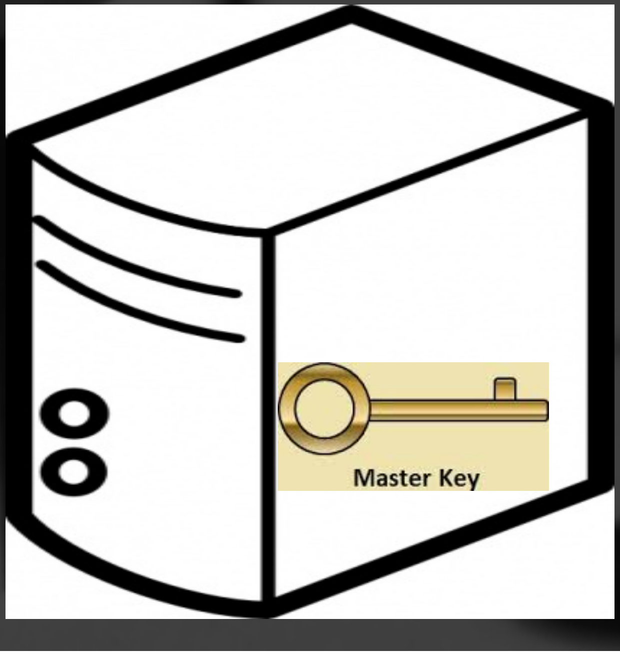
Private Key Generator

- Based on rfc5408 for Identity-Based encryption architecture
  - Toolkit to generate ASN1 encode certificate for Master Secret Key – Public Key pair.
  - Servers to deploy the private key generator extraction algorithm
  - Our first prototype supports:
    - Cécile Delerablée Identity-based broadcast encryption [1]
    - Kenneth and Jacob Efficient Identity-based Signatures Secure in the Standard Model [2]

[1] C. Delerable, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys".
[2] hKenneth G. Paterson, Jacob C. N. Schuldt, "Efficient Identity-based signatures secure in the standard model".

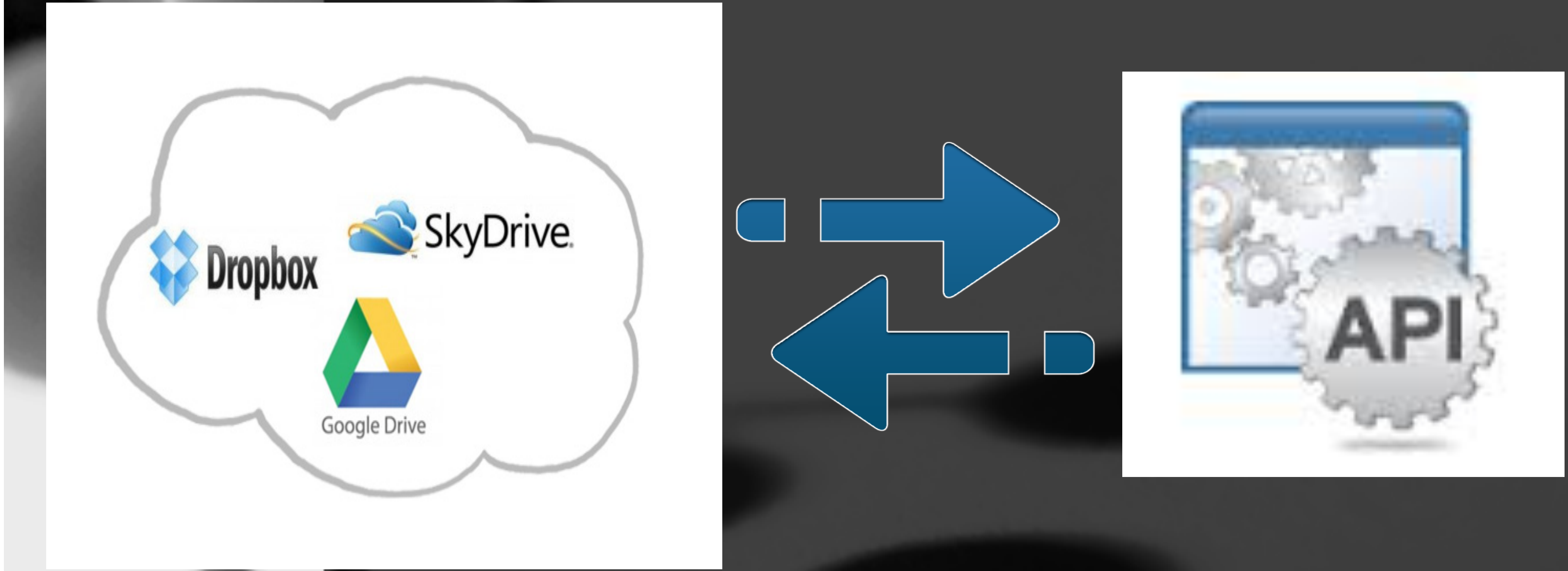
Who hosts the PKGs ?

- Diaspora proved that there is many trusted third party likely to host your PKG like for the Diaspora pods
- Since users could not trust them totally, Encapsulated Session Keys is put on the cloud storage with access control



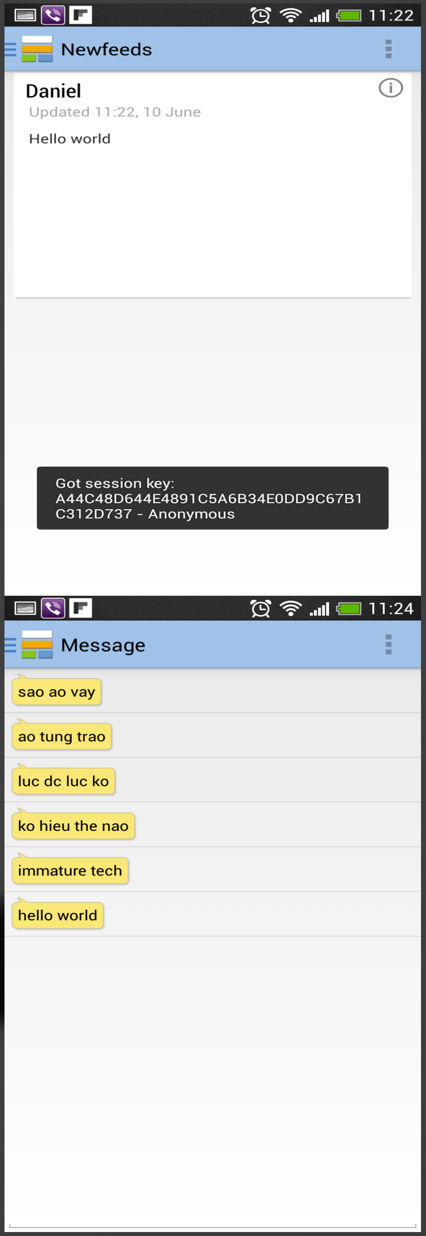
How to add new cloud provider

- We define an api interface that cloud storage should support to work with our apps
- Adding new clouds = mapping cloud's API to our API



First prototype

- Our DOSN prototype support
  - Find your friend profile with id
  - Posting status
  - Sending message (Online, offline)
  - Follow your friends
  - Follow people
  - Follow #hashtag
  - Change session key at will
  - Encrypted your contents with session key



Daniel
Updated 11:22, 10 June
Hello world

Got session key:
A44C48D644E4891C5A6B34E0DD9C67B1
C312D737 - Anonymous

sao ao vay

ao tung trao

luc dc luc ko

ko hieu the nao

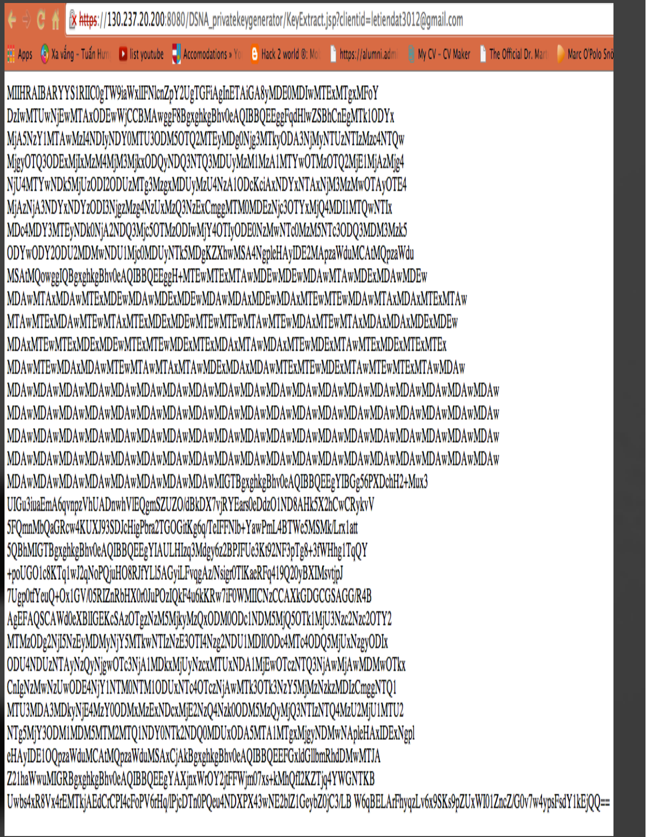
immature tech

hello world

Java web server as PKG

- Id-based master key-pair generator make it easy to setup and manage the PKG
- All you need is a java web-server

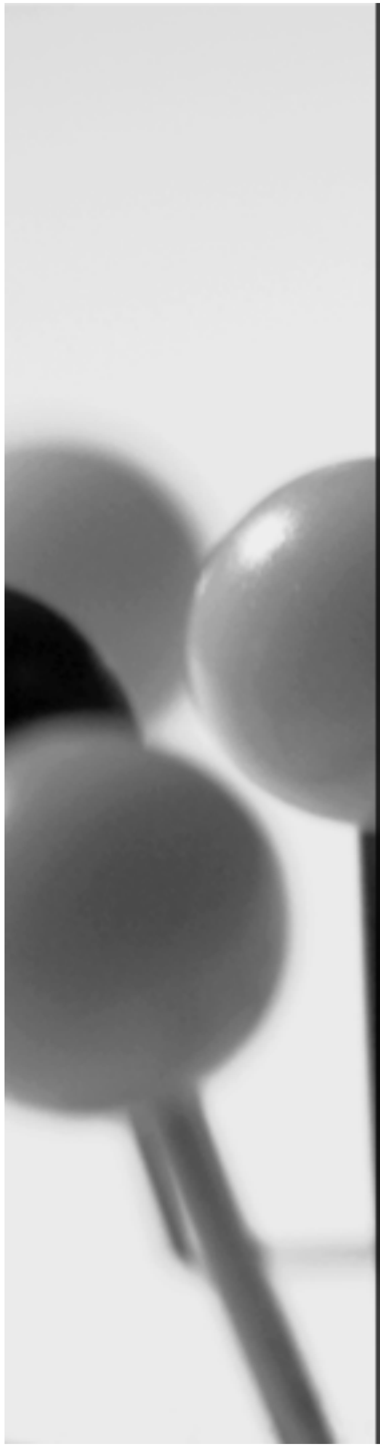
=

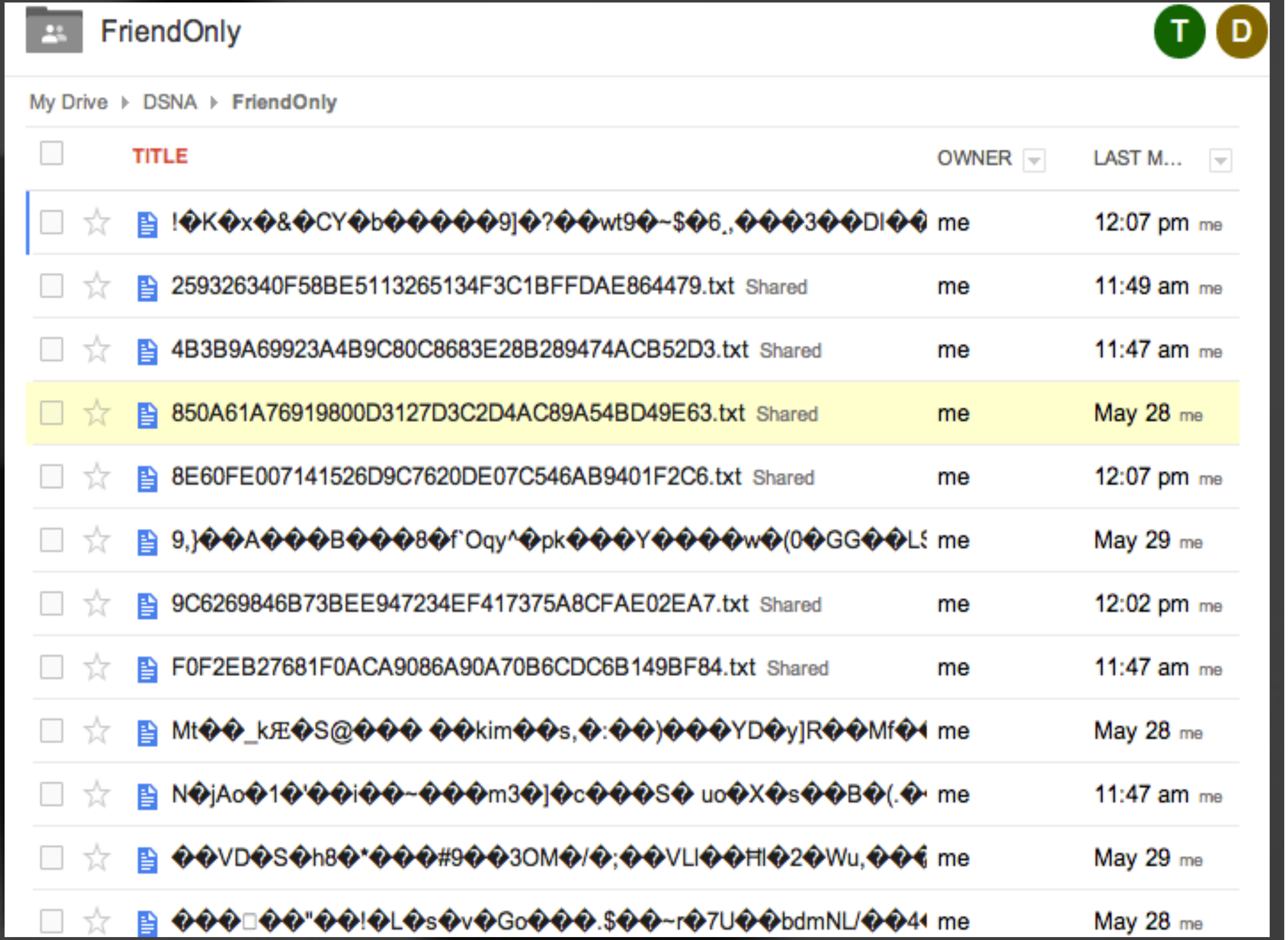


https://130.237.20.200:8080/DSNA_privatekeygenerator/KeyExtract.jsp?clientid=letiendat3012@gmail.com

```
MIIHRAIBARYYS1RIIC0gTW9iaWxlIFNlcnZpY2UgTGFiAgInETAiGA8yMDE0MDIwMTExMTgxMFoY
DzIwMTUwNjEwMTAxODEwWjCCBMAwggF8BgxghkgBhv0eAQIBBQEEggFqdHlwZSBhCnEgMTk1ODYx
MjA5NzY1MTAwMzI4NDIyNDY0MTU3ODM5OTQ2MTEyMDg0Njg3MTkyODA3NjMyNTUzNTIzMzc4NTQw
MjgyOTQ3ODExMjIxMzM4MjM3MjkxODQyNDQ3NTQ3MDUyMzM1MzA1MTYwOTMzOTQ2MjE1MjAzMjg4
NjU4MTYwNDk5MjUzODI2ODUzMTg3MzgxMDUyMzU4NzA1ODcKciAxNDYxNTAxNjM3MzMwOTAyOTE4
MjAzNjA3NDYxNDYzODI3NjgzMzg4NzUxMzQ3NzExCmggMTM0MDEzNjc3OTYxMjQ4MDI1MTQwNTIx
MDc4MDY3MTEyNDk0NjA2NDQ3Mjc5OTMzODIwMjY4OTIyODE0NzMwNTc0MzM5NTc3ODQ3MDM3Mzk5
ODYwODY2ODU2MDMwNDU1Mjc0MDUyNTk5MDgKZXhwMSA4NgpleHAyIDE2MApzaWduMCAtMQpzaWdu
MSAtMQowggIQBgxghkgBhv0eAQIBBQEEggH+MTEwMTExMTAwMDEwMDEwMDAwMTAwMDExMDAwMDEw
MDAwMTAxMDAwMTExMDEwMDAwMDExMDEwMDAwMDAxMDEwMDAxMTEwMTEwMDAwMTAxMDAxMTExMTAw
MTAwMTExMDAwMTEwMTAxMTExMDExMDEwMTEwMTEwMTAwMTEwMDAxMTEwMTAxMDAxMDAxMDExMDEw
MDAxMTEwMTExMDExMDEwMTExMTEwMDExMTExMDAxMTAwMDAxMTEwMDExMTAwMTExMDExMTExMTEx
MDAwMTEwMDAxMDAwMTEwMTAwMTAxMTAwMDExMDAxMDAwMTExMTEwMDExMTAwMTEwMTExMTAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMIGTBgxghkgBhv0eAQIBBQEEgYIBGg56PXDchH2+Mux3
UIGu3iuaEmA6qvnpzVhUADnwhVlEQgmSZUZO/dBkDX7vjRYEars0eDdzO1ND8AHk5X2hCwCRykvV
5FQmnMbQaGRcw4KUXJ93SDJcHigPbra2TGOGitKg6q/TeIFFNlb+YawPmL4BTWe5MSMk/Lrx1att
5QBhMIGTBgxghkgBhv0eAQIBBQEEgYIAULHIzq3Mdgy6z2BPJFUc3Kt92NF3pTg8+3fWHhg1TqQY
+poUGO1c8KTq1wJ2qNoPQjuHO8RJfYLl5AGyiLFvqgAz/Nsigr0TlKacRFq419Q20yBXIMsvtjpJ
7Ugp0tfYeuQ+Ox1GV/05RIZnRbHX0r0JuPOzIQkF4u6kKRw7iF0WMIICNzCCAXkGDGCGSAGG/R4B
AgEFAQSCAWd0eXBlIGEKcSAzOTgzNzM5MjkyMzQxODM0ODc1NDM5MjQ5OTk1MjU3Nzc2Nzc2OTY2
MTMzODg2NjI5NzEyMDMyNjY5MTkwNTIzNzE3OTI4Nzg2NDU1MDI0ODc4MTc4ODQ5MjUxNzgyODIx
ODU4NDUzNTAyNzQyNjgwOTc3NjA1MDkxMjUyNzcxMTUxNDA1MjEwOTczNTQ3NjAwMjAwMDMwOTkx
CnIgNzMwNzUwODE4NjY1NTM0NTM1ODUxNTc4OTczNjAwMTk3OTk3NzY5MjMzNzkzMDIzCmggNTQ1
MTU3MDA3MDkyNjE4MzY0ODMxMzExNDcxMjE2NzQ4Nzk0ODM5MzQyMjQ3NTIzNTQ4MzU2MjU1MTU2
NTg5MjY3ODM1MDM5MTM2MTQ1NDY0NTk2NDQ0MDUxODA5MTA1MTgxMjgyNDMwNApleHAxIDExNgpl
eHAyIDE1OQpzaWduMCAtMQpzaWduMSAxCjAkBgxghkgBhv0eAQIBBQEEFGxldGllbmRhdDMwMTJA
Z21haWwuMIGRBgxghkgBhv0eAQIBBQEEgYAXjnxWrOY2jtFFWjm07xs+kMhQfI2KZTjq4YWGNTKB
Uwbs4xR8Vx4rEMTkjAEdCrCPI4cFoPV6rHq/lPjcDTn0PQeu4NDXPX43wNE2blZ1GeybZ0jC3/LB W6qBELArFhyqzLv6x9SKs9pZUxWI01ZncZ/G0v7w4ypsFsdY1kEjQQ==
```

How your cloud drive look like



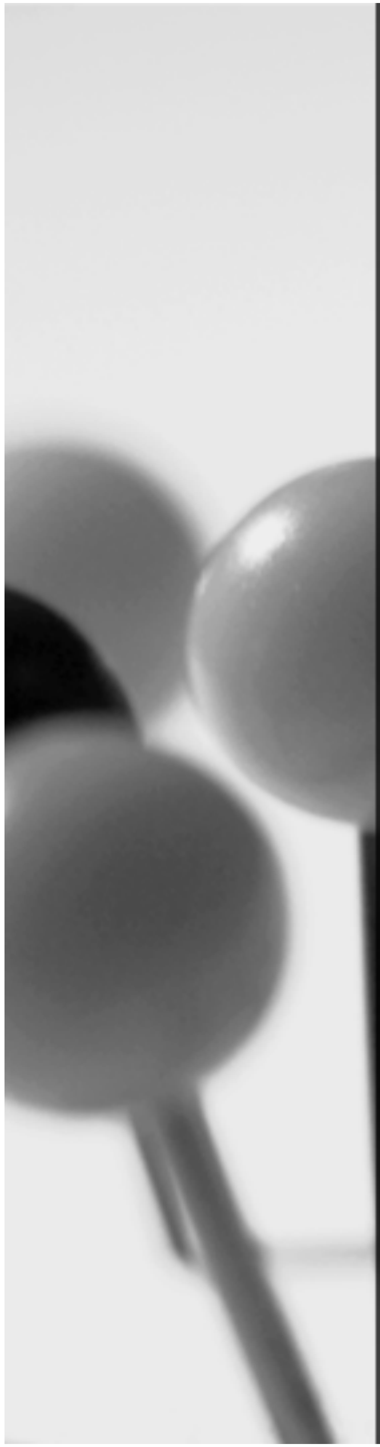


FriendOnly

My Drive ▸ DSNA ▸ FriendOnly

□		TITLE	OWNER	LAST M...
□	☆	!�K�x�&�CY�b�������9]�?��wt9�~$�6,,���3��DI��	me	12:07 pm me
□	☆	259326340F58BE5113265134F3C1BFFDAE864479.txt Shared	me	11:49 am me
□	☆	4B3B9A69923A4B9C80C8683E28B289474ACB52D3.txt Shared	me	11:47 am me
□	☆	850A61A76919800D3127D3C2D4AC89A54BD49E63.txt Shared	me	May 28 me
□	☆	8E60FE007141526D9C7620DE07C546AB9401F2C6.txt Shared	me	12:07 pm me
□	☆	9,}��A����B����8�f`Oqy^�pk����Y�����w�(0�GG��L	me	May 29 me
□	☆	9C6269846B73BEE947234EF417375A8CFAE02EA7.txt Shared	me	12:02 pm me
□	☆	F0F2EB27681F0ACA9086A90A70B6CDC6B149BF84.txt Shared	me	11:47 am me
□	☆	Mt��_kЯE�S@���� ��kim��s,�:��)����YD�y]R��Mf�	me	May 28 me
□	☆	N�jAo�1�'��i��~���m3�]�c����S� uo�X�s��B�(.�	me	11:47 am me
□	☆	��VD�S�h8�*����#9���3OM�/�;��VLI��Ħl�2�Wu,���	me	May 29 me
□	☆	���□��"��!�L�s�v�Go����.$��~r�7U��bdmNL/��4	me	May 28 me

Evaluation

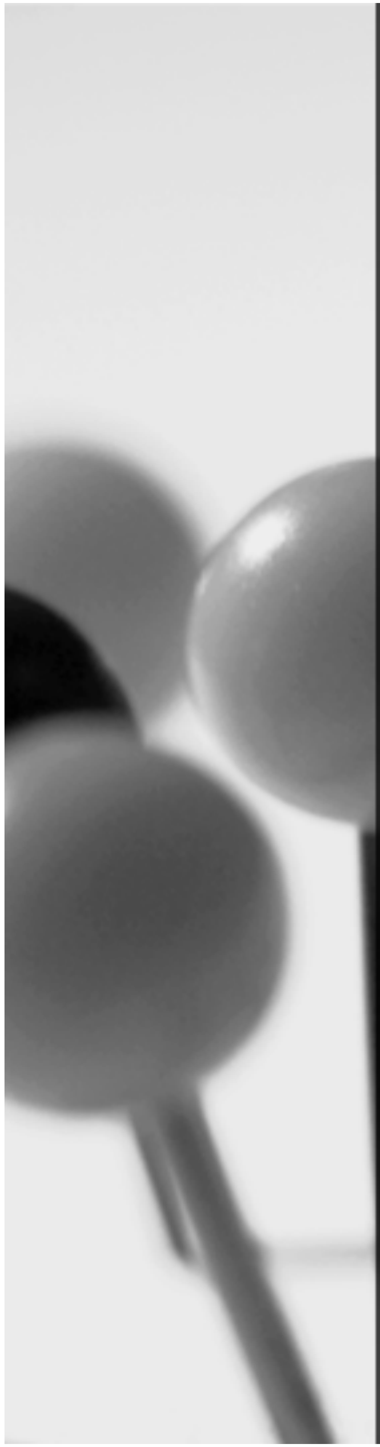


Security

- Depends on which information attackers possess
  - Only encrypted contents: 128-bits security level
  - Encrypted contents and encapsulated session keys: 80-bits security level

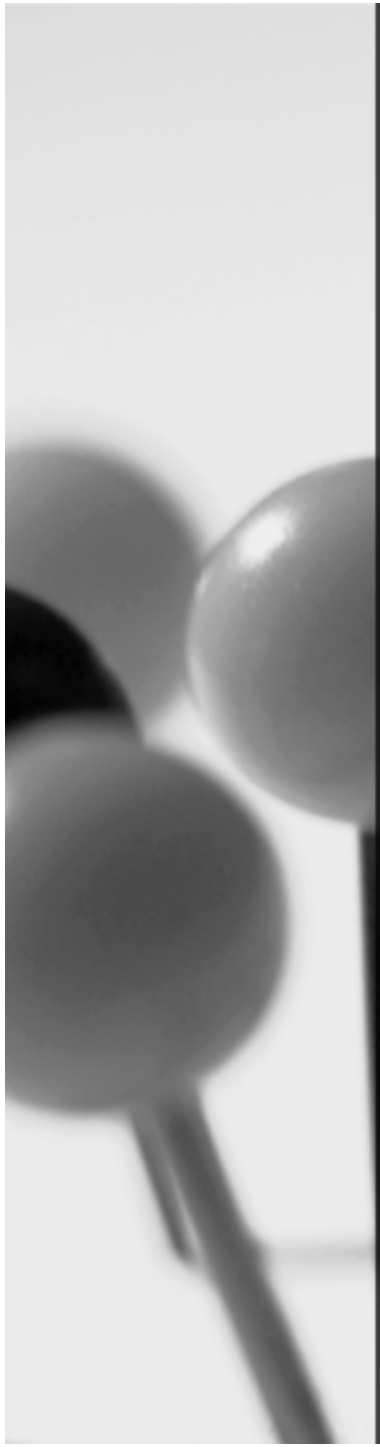
Attack From	Security level	Possible combinations	Time required to break [1]
A storage provider	128-bits	3.4×10^{38}	1.02×10^{18} years
Colluded storage providers	80-bits	1.209×10^{24}	3552 years
Hacker have access to the encrypted contents	128-bits	3.4×10^{38}	1.02×10^{18} years
Hacker have access to encapsulated session keys and encrypted contents	80-bits	1.209×10^{24}	3552 years

[1] *"How secure is AES against brute force attacks? | EE Times," EETimes. [Online] Available: http://www.eetimes.com/document.asp?doc_id=1279619*



Efficiency

- Computation:
  - Symmetric encryption/decryption: averagely 10ms per 3KB item
  - Session key distribution process time: 1053 ms per key on average (due to identity-based cryptography cost)
- Energy:
  - 8% of battery for processing 1,000 encrypted items of size 3KB received.
  - 8% of battery for processing 1,000 plain items of size 3KB received.
  - 16% of battery for distributing 1,000 session keys



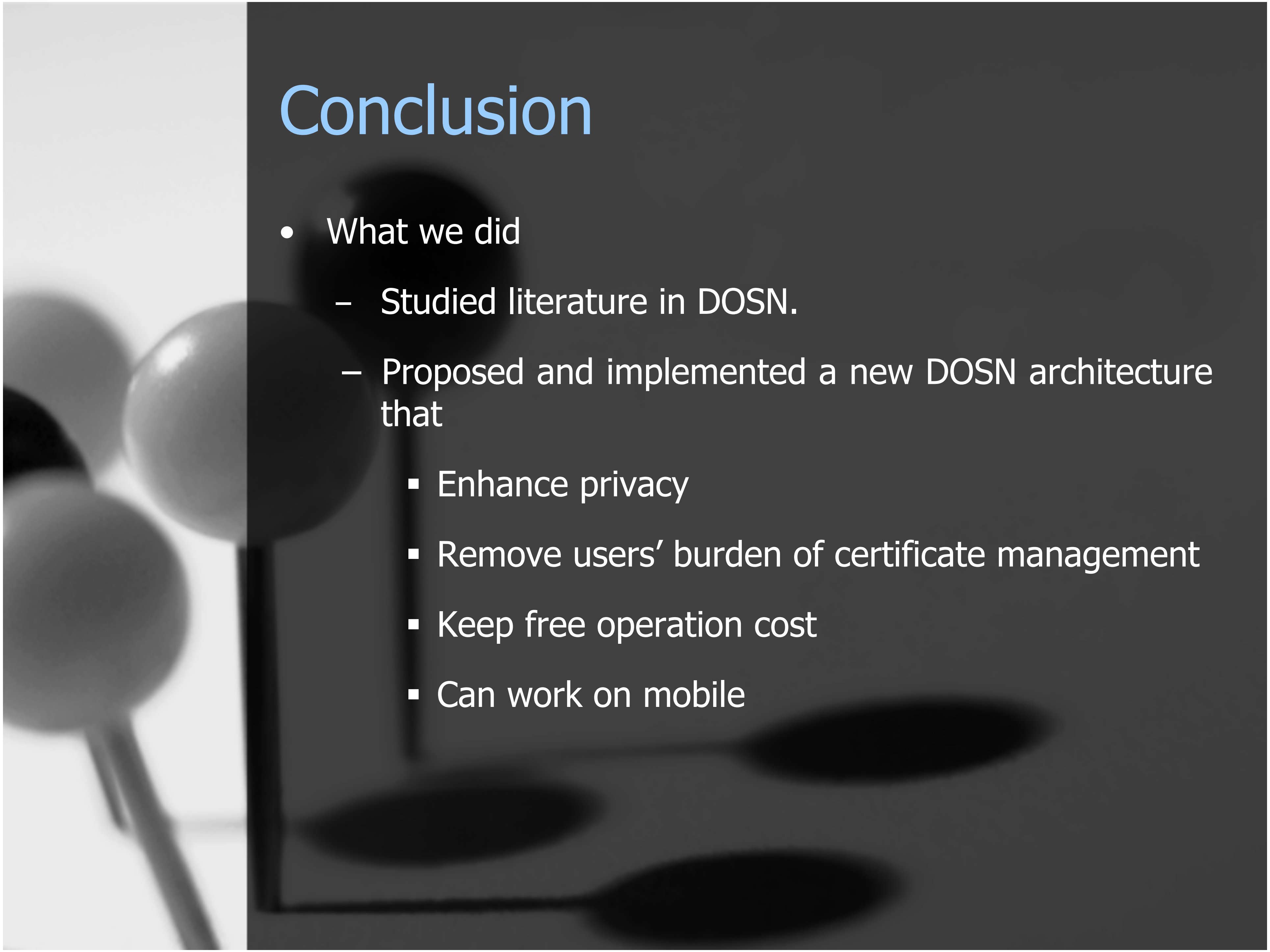
Estimated overhead cost of security scheme

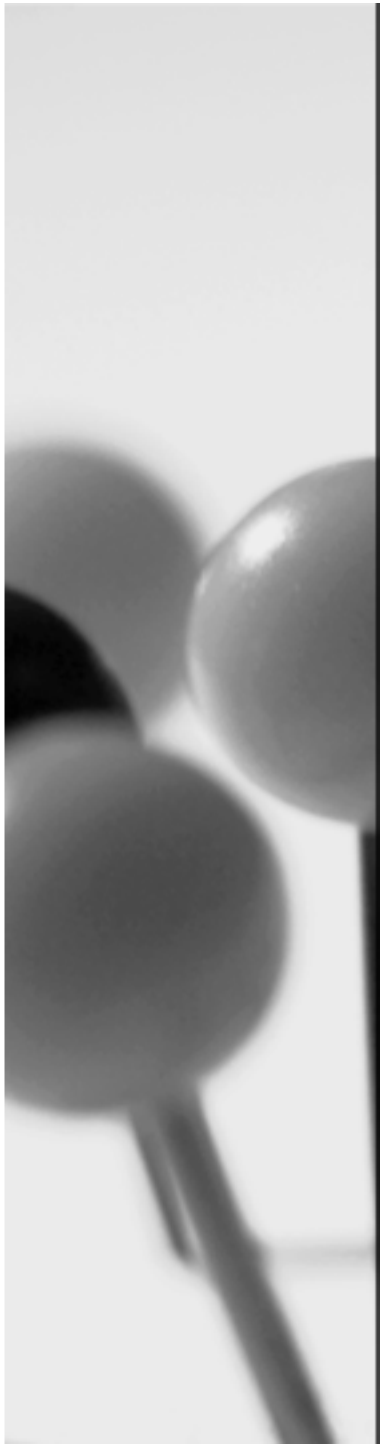
- Using a model based on statistics[1] where an user have: averagely 300 friends, 200 of them publishing content daily (3 items averagely each). Given that active users force to change session keys:
  - Daily: 210.6 seconds and 3% battery for distributing session keys per day.
  - 3-days basis: 70.2 seconds and 1% battery for distributing session keys per day.
  - Weekly: 30.1 seconds and 0.5% battery for distributing session keys per day.
- Symmetric encryption cost could be ignore for small sizes item

[1] MarketingCharts. [Online]. Available: http://www.marketingcharts.com/wp/online/18-24-year-olds-on-facebook-boast-an-average-of-510-friends-28353/.

Conclusion

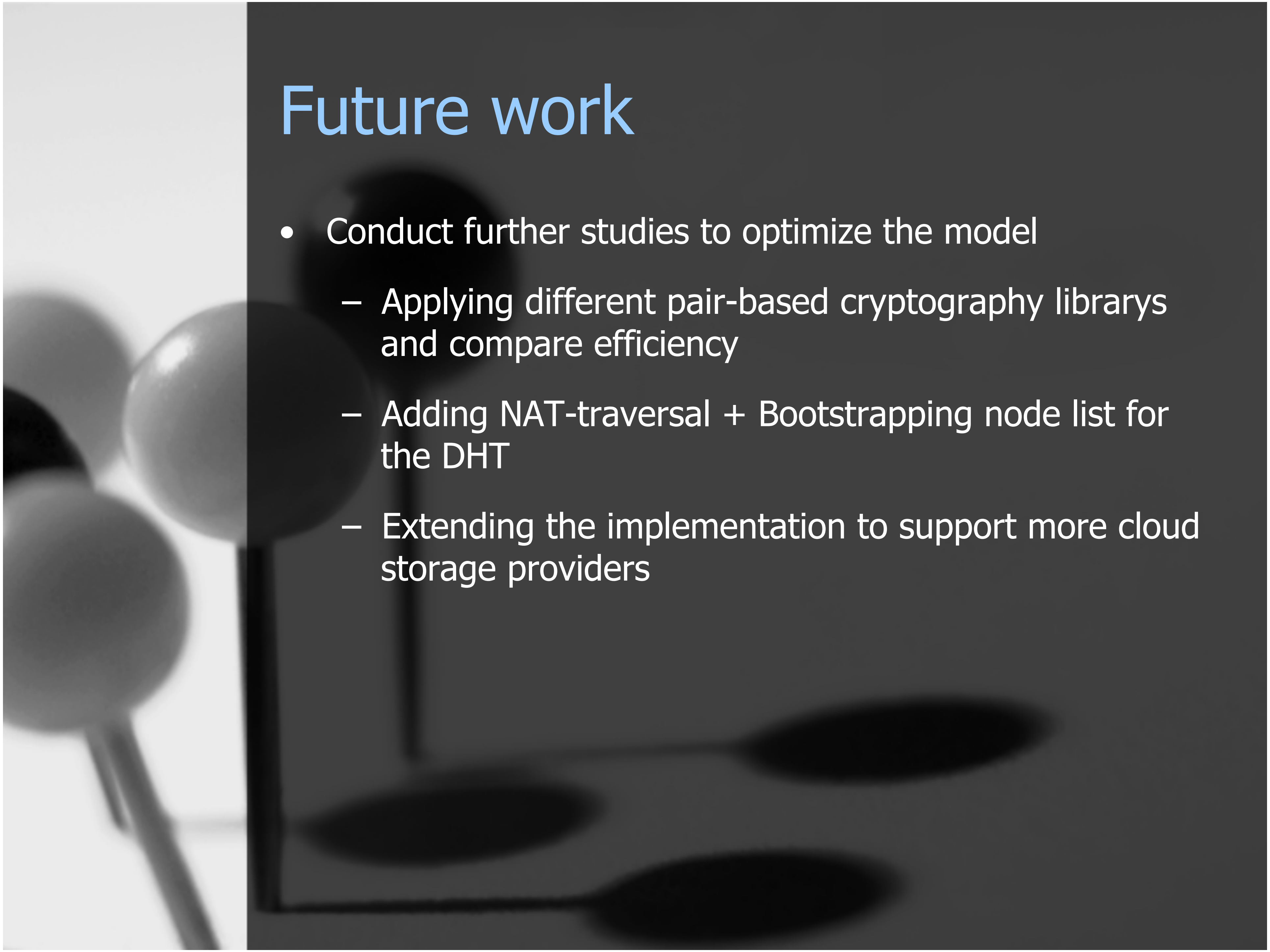
- What we did
  - Studied literature in DOSN.
  - Proposed and implemented a new DOSN architecture that
    - Enhance privacy
    - Remove users' burden of certificate management
    - Keep free operation cost
    - Can work on mobile





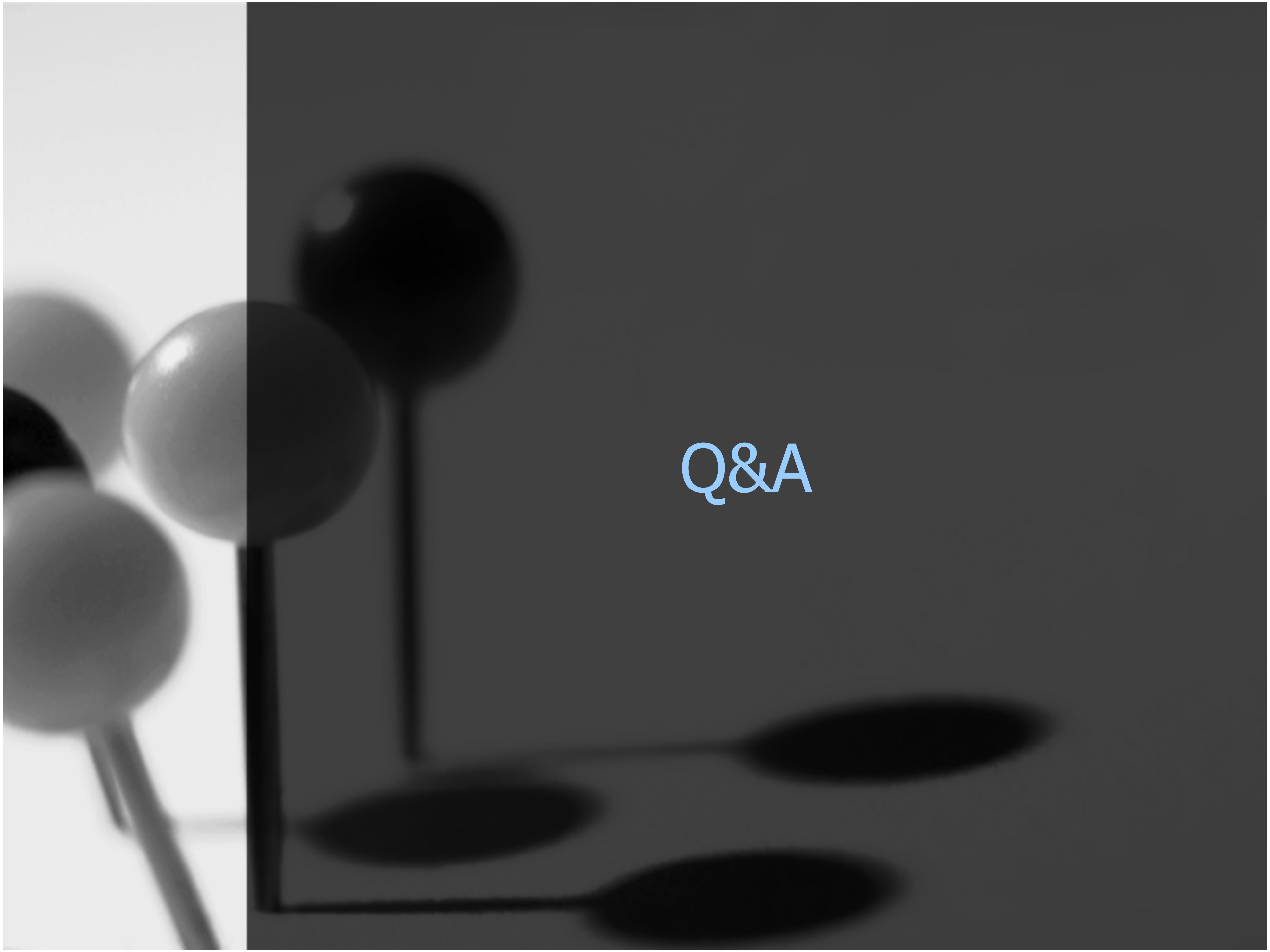
Conclusion

- What we archived
  - First proof-of-concept prototype:
    - Enhance privacy by separating session keys and encrypted contents in different clouds
    - Provide global trust and remove certificate management effort with id-based crypto
    - Have free operation cost
    - Show adequate computation and energy efficiency to work on mobile devices

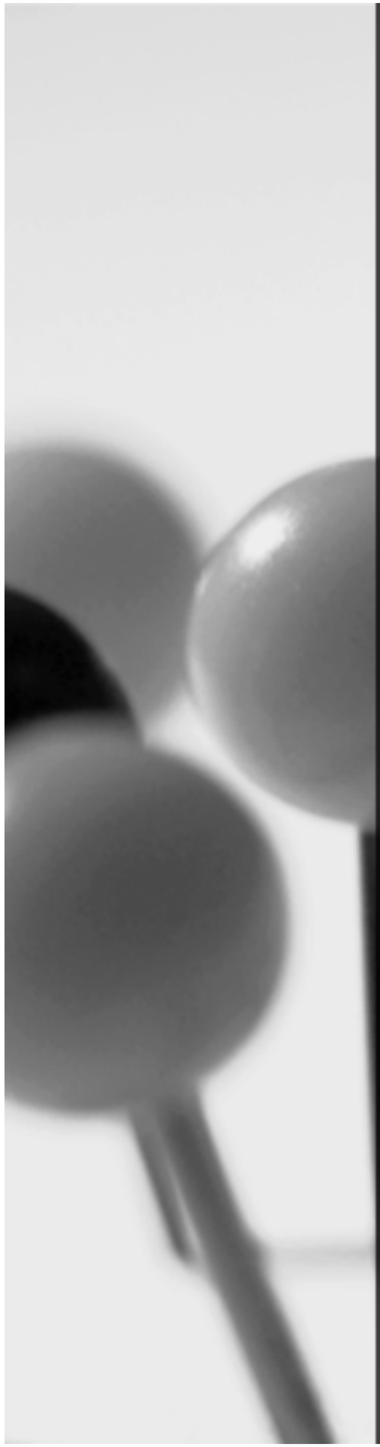


Future work

- Conduct further studies to optimize the model
  - Applying different pair-based cryptography librarys and compare efficiency
  - Adding NAT-traversal + Bootstrapping node list for the DHT
  - Extending the implementation to support more cloud storage providers



Q&A



References

[DN07] *Danah Boyd and Nicole B. Ellison. Social network sites: Definition, history and scholarship. Journal of Computer-Mediated Communication, 13(1), 2007*

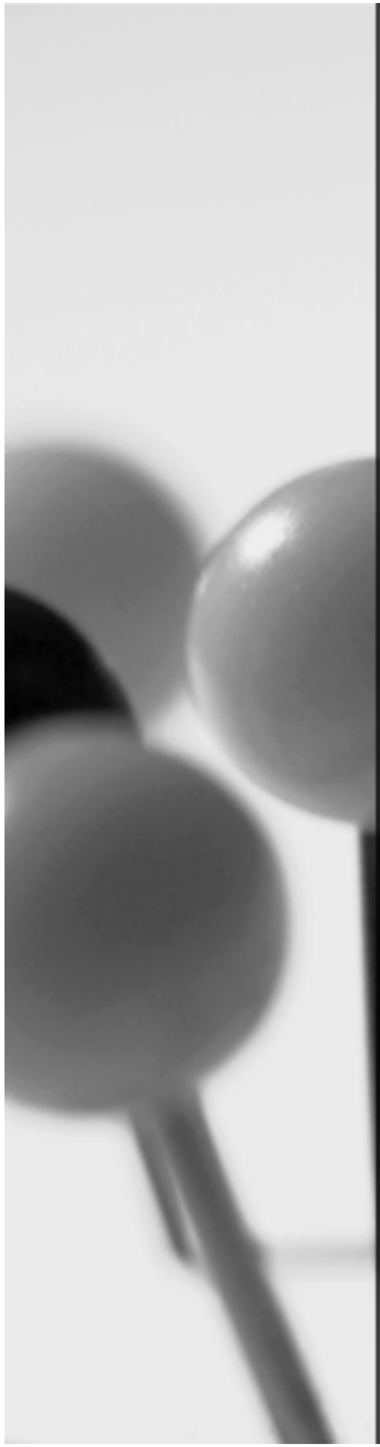
[SV10] *E. Steel and J. E. Vascellaro, "Facebook, MySpace confront privacy loophole," The Wall Street Journal, May 2010*

[KO10] *K. Opsahl, "Facebook's eroding privacy policy: A timeline," https://www.eff.org/deeplinks/2010/04/facebook-timeline. Accessed 29/5/2014*

[AS10] *Anwitaman Datta, Sonja Buchegger, Le Hung Vu, Thorsten Strufe, and Krzysztof Rzadca, "Decentralized Online Social Networks". In Handbook of Social Network Technologies and Applications, 2010, pp 349-378*

[DB14] Diaspora blog,

https://blog.diasporafoundation.org/1-diaspora-celebrates-one-year-as-a-community-project. Accessed 29/5/2014



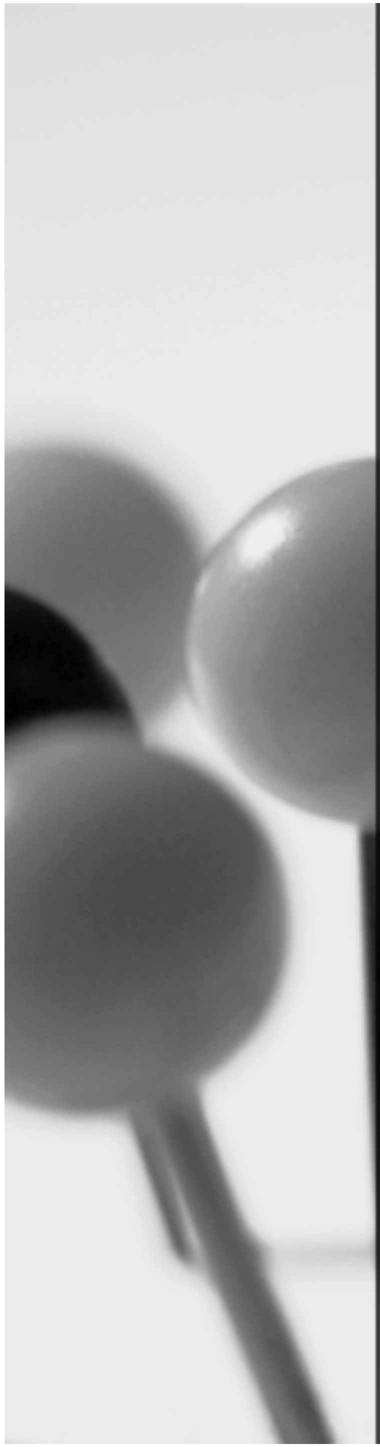
References

[AHRL] *Amre Shakimov∗, Harold Lim∗, Ram´on C´aceres†, Landon P. Cox∗, Kevin Li†, Dongtao Liu∗, and Alexander Varshavsky†, "Vis-`a-Vis: Privacy-Preserving Online Social Networking via Virtual Individual Servers"*

[PMUT13] *New compete study: Primary mobile users on Twitter, https://blog.twitter.com/2013/new-compete-study-primary-mobile-users-on-twitter, Accessed 1/6/2014*

[FMUC13] *Facebook mobile user count, http://techcrunch.com/2013/08/13/facebook-mobile-user-count/, Acessed 30/5/2014*

[CD07] *C. Delerable, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in Pairing-Based Cryptography Pairing 2007, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4575, pp. 39–59.*



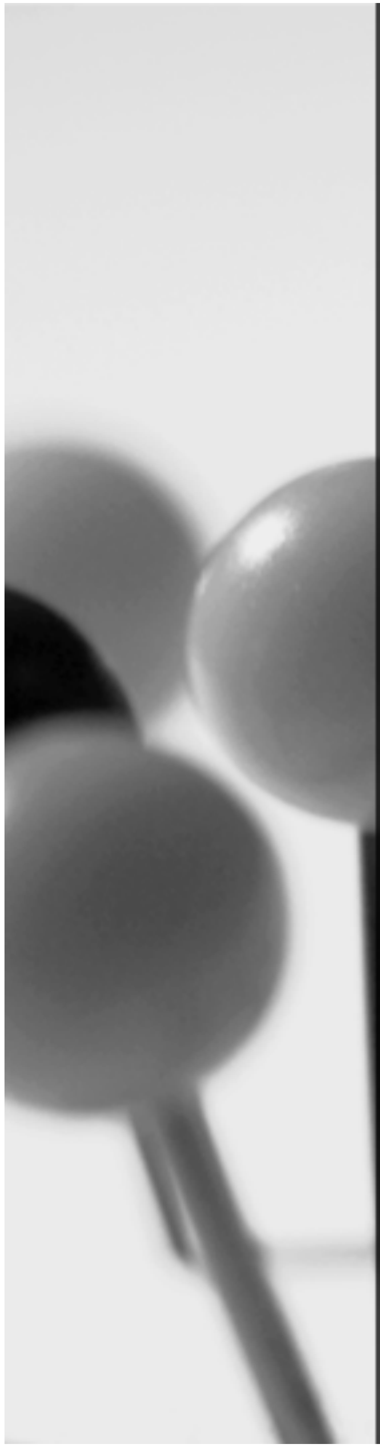
References

[PS06] *Kenneth G. Paterson, Jacob C. N. Schuldt, "Efficient Identity-based signatures secure in the standard model", Information Security and Privacy Lecture Notes in Computer Science Volume 4058, 2006, pp 207-222*

[FTUE14] *"Facebook tinkered with users' emotions in experiment." [Online]. Available: http://www.latimes.com/nation/nationnow/la-na-nn-facebook-research-20140629-story.html. [Accessed: 11-Jul-2014*

[MC14] *"18-24-Year-Olds on Facebook Boast an Average of 510 Friends," MarketingCharts. [Online]. Available: http://www.marketingcharts.com/wp/online/18-24-year-olds-on-facebook-boast-an-average-of-510-friends-28353/. [Accessed: 22-Jun-2014].*

[EE14] *"How secure is AES against brute force attacks? | EE Times," EETimes. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1279619. [Accessed: 17-Jul-2014].*



Appendix

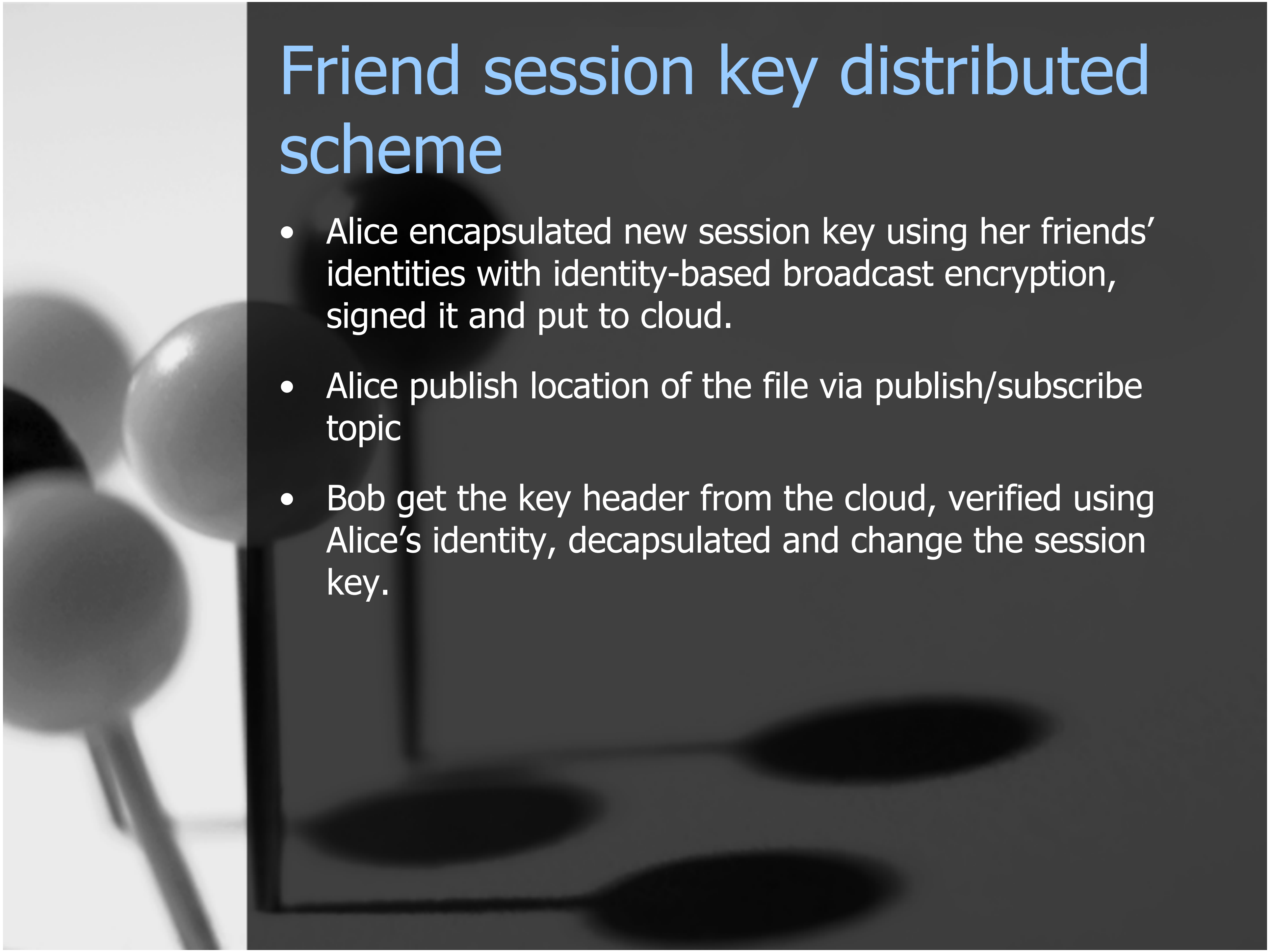
- Source code:

https://github.com/kekkaishivn/DSNA-Application

- Private key generator server:

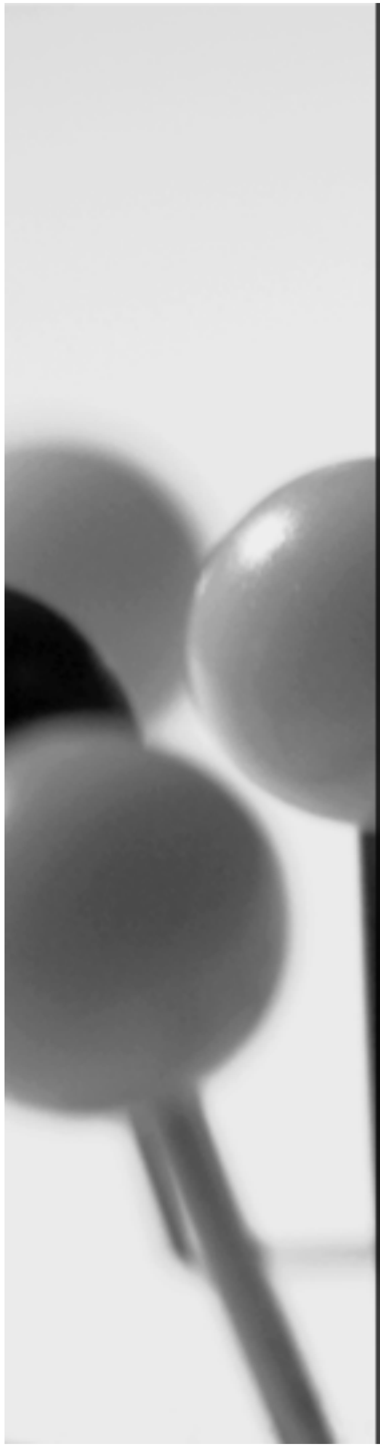
https://130.237.20.200:8080/DSNA_privatekeygenerator/SystemPublic.txt

https://130.237.20.200:8080/DSNA_privatekeygenerator/KeyExtract.jsp?clientid=letiendat3012@gmail.com



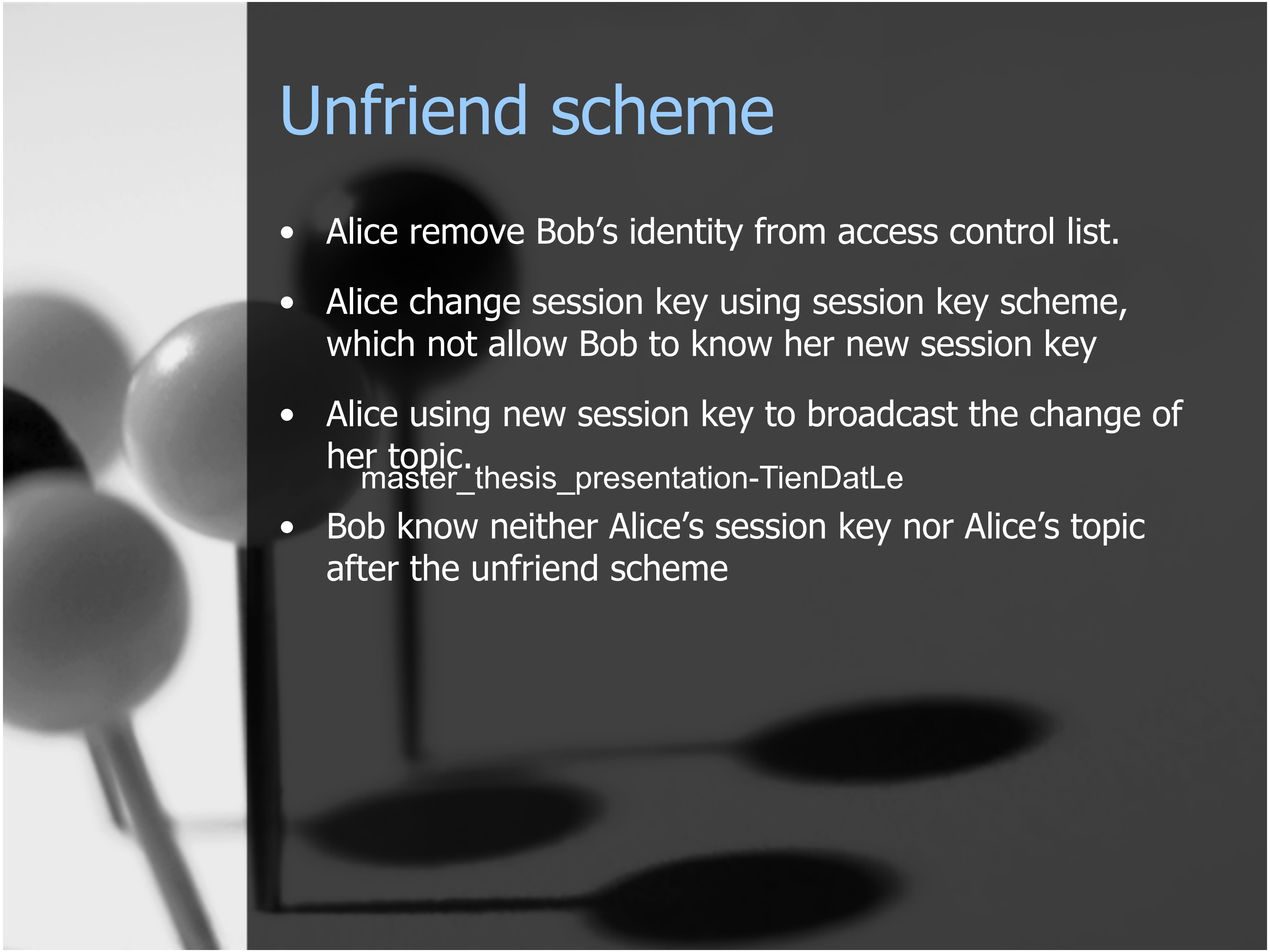
Friend session key distributed scheme

- Alice encapsulated new session key using her friends' identities with identity-based broadcast encryption, signed it and put to cloud.
- Alice publish location of the file via publish/subscribe topic
- Bob get the key header from the cloud, verified using Alice's identity, decapsulated and change the session key.



Friend authenticated scheme

- Alice get Bob’s profile from DHT using Bob’s identity (gmail address).
- Alice get Bob’s To-Send-Friend-Request topic and send friend request via the topic.
- Bob get Alice’s friend request with Alice identity. He create an file with a nonce and his profile encrypted key in cloud; encrypt the file by Alice identity and send file location to Alice.
- Alice get the file from Bob, create an confirmation with a nonce+1 and her profile encrypted key in cloud; encrypt the file by Bob identity and send file location to Bob.
- Alice and Bob decrypt their profile to get their To-Subscribe-Topic. They subscribe each other topics and become friend.



Unfriend scheme

- Alice remove Bob's identity from access control list.
- Alice change session key using session key scheme, which not allow Bob to know her new session key
- Alice using new session key to broadcast the change of her topic.

- Bob know neither Alice's session key nor Alice's topic after the unfriend scheme

Cloud interface

```
public interface CloudStorageService {
	public List<String> initializeDSNAFolders() throws
UserRecoverableAuthIOException, IOException;
	public String uploadContentToFriendOnlyFolder(String title, String type, String
description, InputStream content) throws UserRecoverableAuthIOException, IOException;
	public String uploadContentToPublicFolder(String title, String type, String
description, InputStream content) throws UserRecoverableAuthIOException, IOException;
	public List<String> addPermission(String fileId, List<String> userIds, String
type, String role) throws UserRecoverableAuthIOException, IOException;
	public void removePermission(String fileId, String permissionId) throws
UserRecoverableAuthIOException, IOException;
	public void removePermission(String fileId, String userId, String permission)
throws UserRecoverableAuthIOException, IOException;
	public String createFolder(String title, String description, String parentId)
	throws UserRecoverableAuthIOException, IOException;
	public String createFile(String title, String type, String description, String
parentId, InputStream content) throws UserRecoverableAuthIOException, IOException;
	public void getFile(String fileId, Continuation<InputStream, Exception> action);
}
```