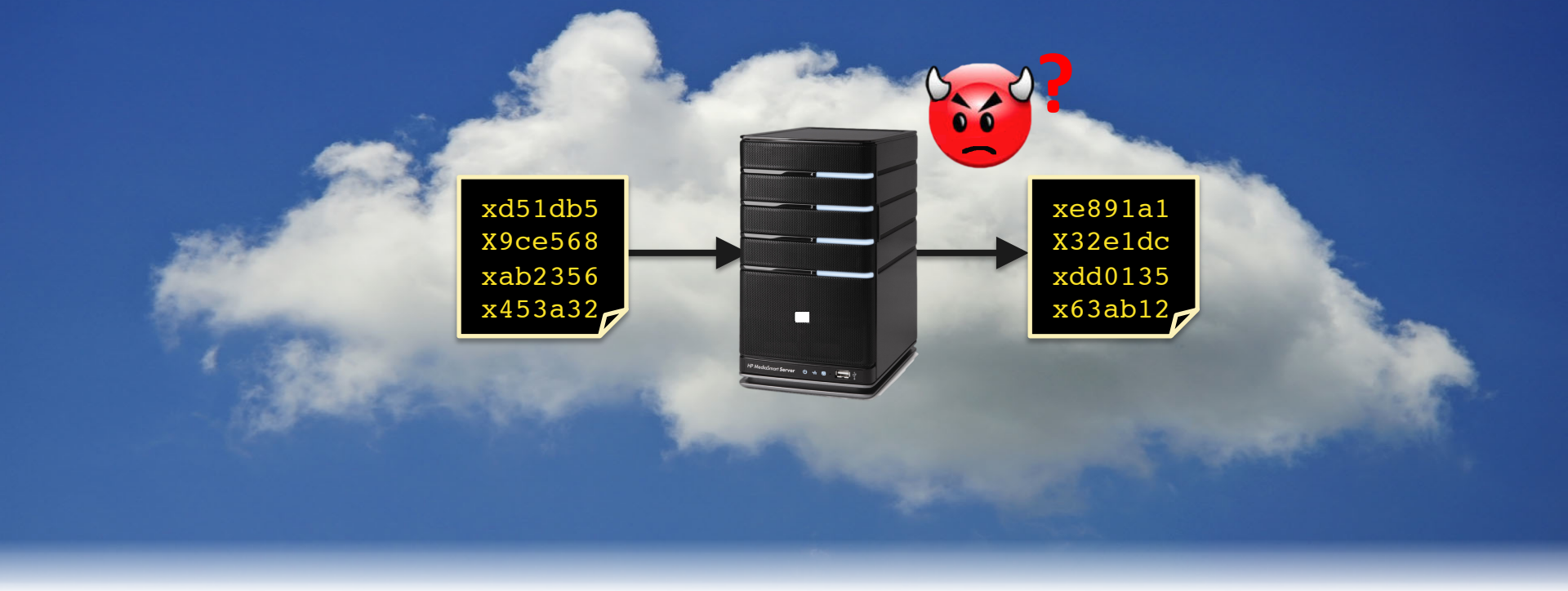


Building systems that compute on encrypted data Mylar

Raluca Ada Popa

Based on joint works with: Catherine Redfield, Nickolai Zeldovich, Hari Balakrishan, Steven Valdez, Jonas Helfer, Frans M. Kaashoek, Andrew Blumberg, Frank H. Li, Shafi Goldwasser, Yael Kalai, Vinod Vaikuntanathan, Raphael Bost, Justine Sherry, Sylvia Ratnasamy, Chang Lan, Ion Stoica, Wenting Li, Rachit Agarwal

Security experts warn of increasing data breaches and privacy risks

LivingSocial Hacked — More Than 50 Million Customer Names, Emails, Birthdates and Encrypted Passwords Accessed (Intern

APRIL 26, 2013 AT 1:15 PM PT

LivingSocial, the daily d
owned in part by Ama

e agents were

Suggested Content

Some Victims of Online Hacking Edge Into the Lig

Compromise of confidential data is prevalent

WordPress firm Automattic suffers root-level hack

dn't Trust Facebook with
mployee's Revelations

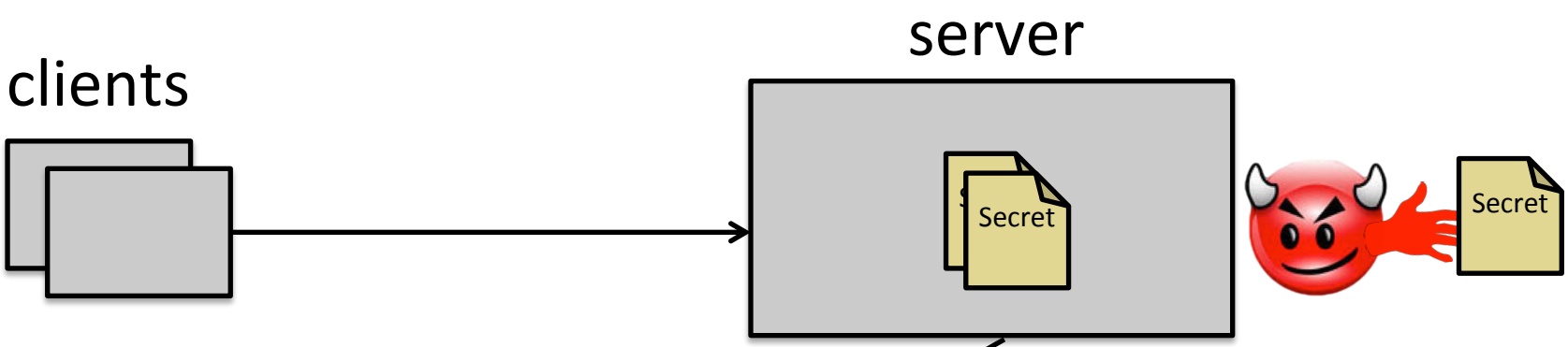
Privacy, security still top cloud concerns

Asia Cloud Forum editors | November 13, 2013

Asia Cloud Forum

An online survey of Microsoft partners has revealed that traditional concerns about

Problem setup



no computation

storage

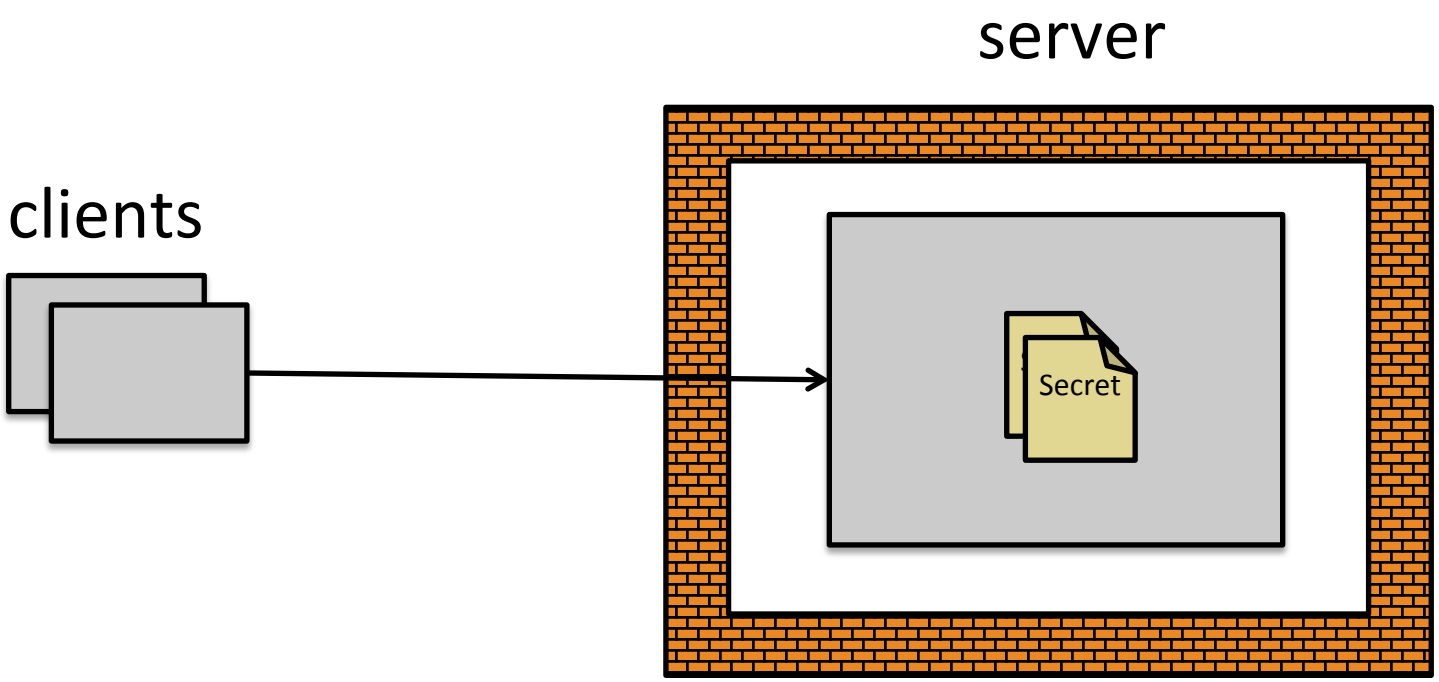
encryption

computation

databases, web applications, mobile applications, machine learning, etc.

??

Current systems strategy



Prevent attackers from breaking into servers

Lots of existing techniques

- Checks at the operating-system level
- Language-based enforcement of a security policy
- Static or dynamic analysis of application code
- Checks at the network level
- Trusted hardware

…

Data still leaks even with these mechanisms

because

attackers eventually break in!

Attacker examples

Attacker	Reason they succeed:
hackers 	software is complex
cloud employees 	**insiders: legitimate server access!**
government 	e.g., physical access

increasingly many companies store data on external clouds

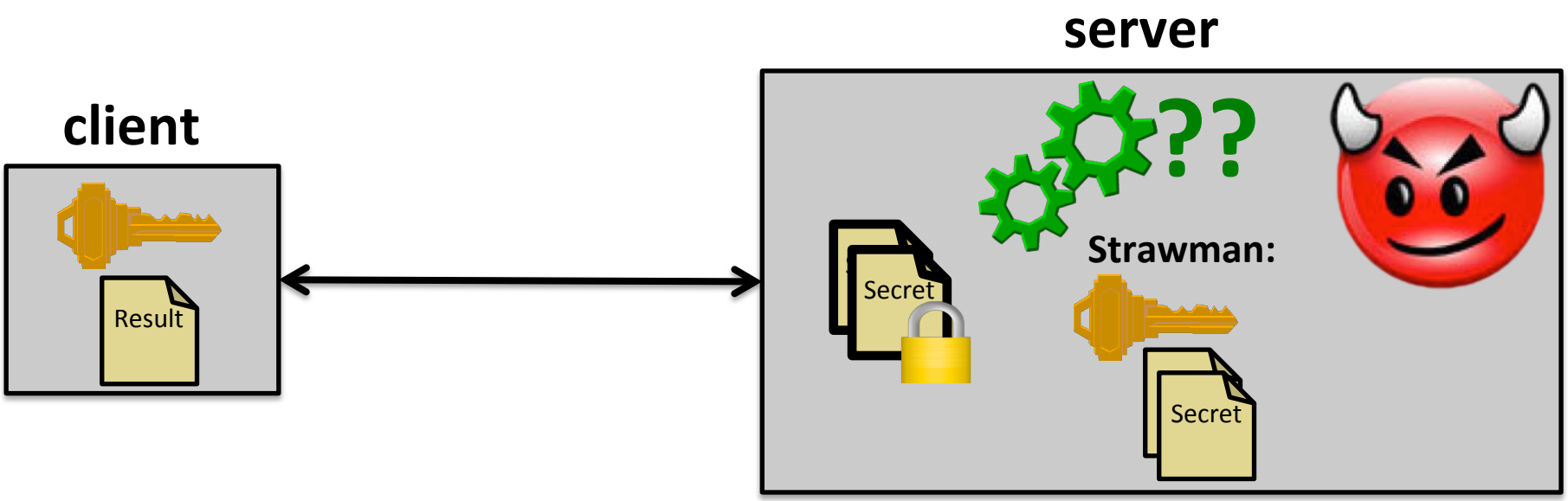
accessed private data according to 

My work:

Systems that protect confidentiality even against attackers with access to all server data

New way of building systems

Servers store, process and *compute on encrypted data in a practical way*

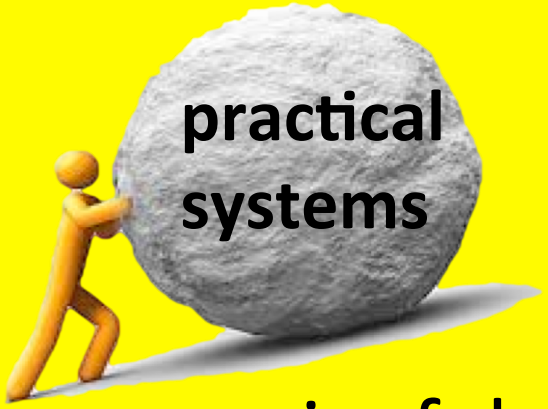


Computing on encrypted data in cryptography
[Rivest-Adleman-Dertouzos'78]

Fully homomorphic encryption (**FHE**) [Gentry'09]

prohibitively slow, e.g., slowdown X 1,000,000,000

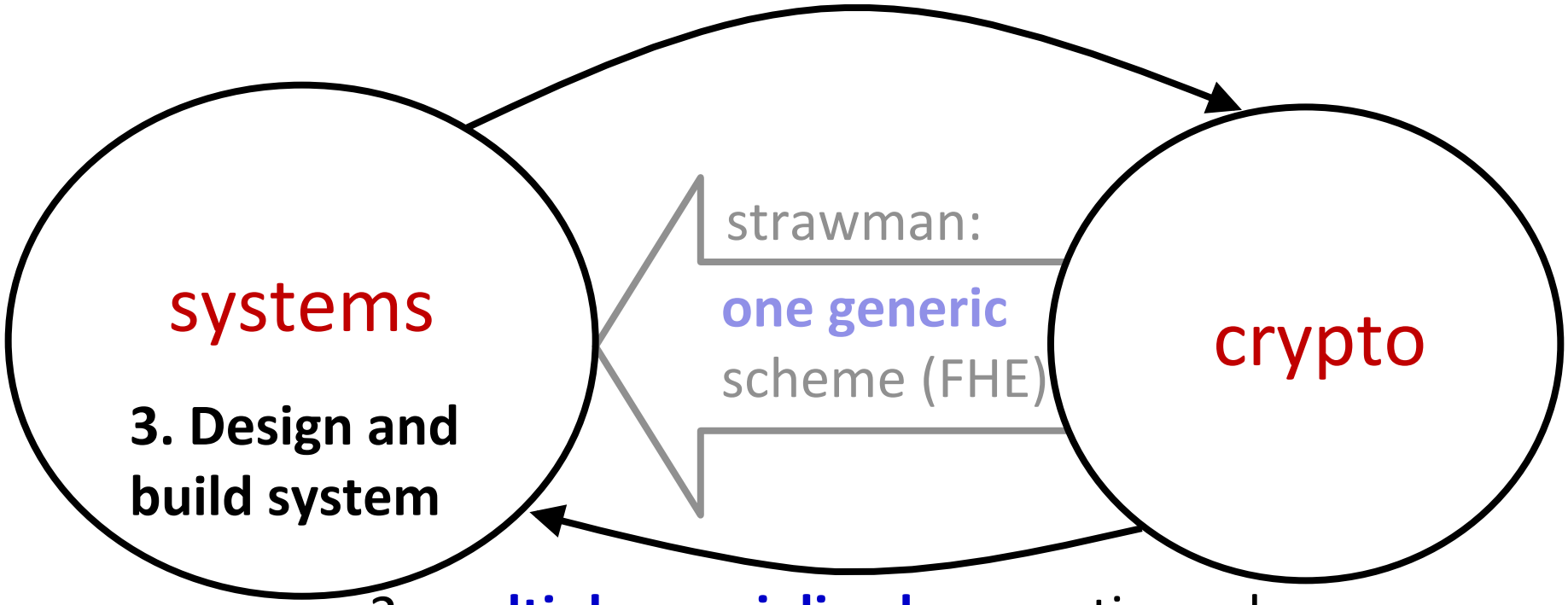
My work: **practical systems**



real-world performance + large class of real applications + meaningful security

Combine systems and cryptography

1. identify core operations needed



2. **multiple specialized** encryption schemes

New schemes:

- comparison, join for CryptDB
- multi-key search for Mylar

What systems can we build with this approach?

A lot of systems!

Project name		System
CryptDB	[SOSP'11][CACM'12] [Oakland'13]	SQL databases
Mylar	[NSDI'14]	web applications
PrivStats	[CCS'11]	mobile applications
VPriv	[Usenix Security'09]	

Coming up…

machine learning classification [NDSS'15]

deep packet inspection **BlindBox**

network middleboxes **PrivMB**

compressed & encrypted key-value store

A lot of systems!

Project name		System
CryptDB	[SOSP'11][CACM'12] [Oakland'13]	SQL databases
Mylar	[NSDI'14]	web applications
PrivStats	[CCS'11]	mobile applications
VPriv	[Usenix Security'09]	

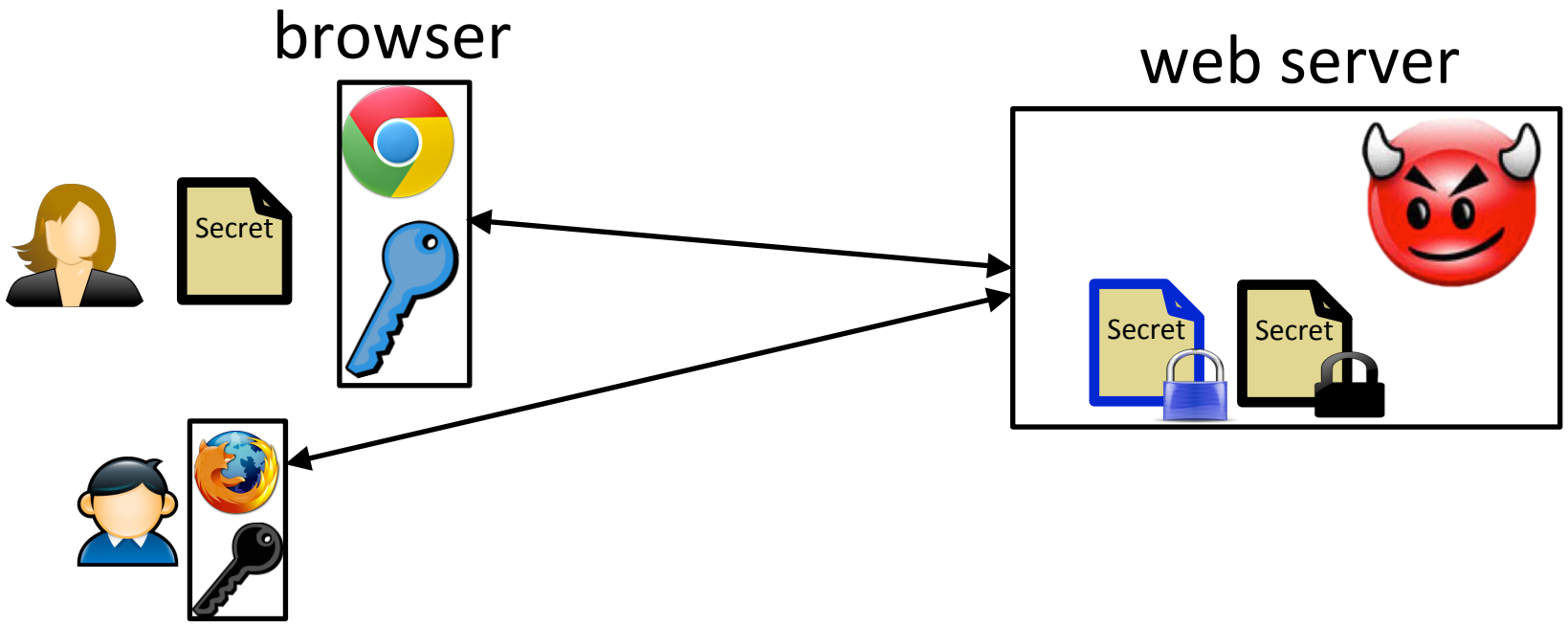
Mylar

[R. A. Popa, E. Stark, J. Helfer, S. Valdez, N. Zeldovich, M. F. Kaashoek, H. Balakrishnan - NDSI'14]

A web framework that protects confidentiality against fully compromised servers

Servers store data encrypted

Plaintext data exists only in browsers



Related work

- File systems: CFS, NCryptfs, SiRiUS, Plutus
- Encrypted databases: CryptDB, Monomi
- Browser encryption: Christodorescu'08, Cryptocat

Far from sufficient for real web apps

Challenges

- Active adversaries (e.g., corrupt webpage)
- Enabling functionality with encryption:
  - data sharing
  - computation

Mylar

- Active adversaries (e.g., corrupt webpage)
- Enabling functionality with encryption:
  - data sharing
  - computation

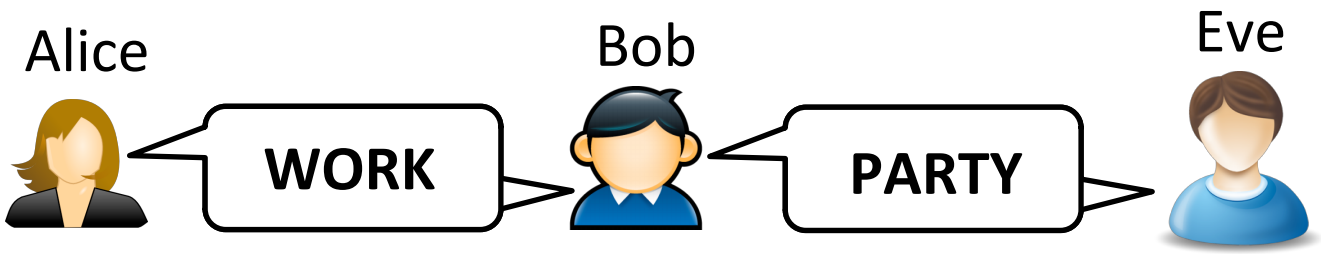
webpage code verification

principal graph & certification

client-side web framework

new encryption scheme:
multi-key search

Example: Chat application



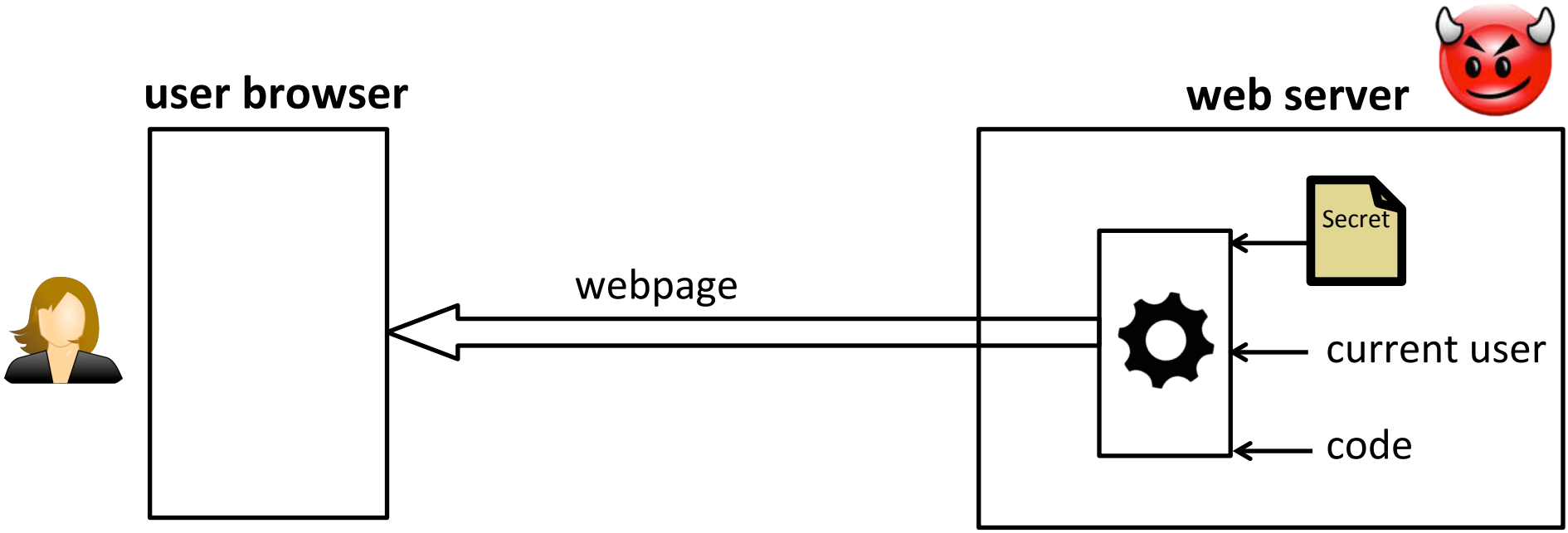
TODO:

- [ ] Server cannot see messages
- [ ] Users share chat rooms securely
- [ ] Format messages, generate html page
- [ ] Search

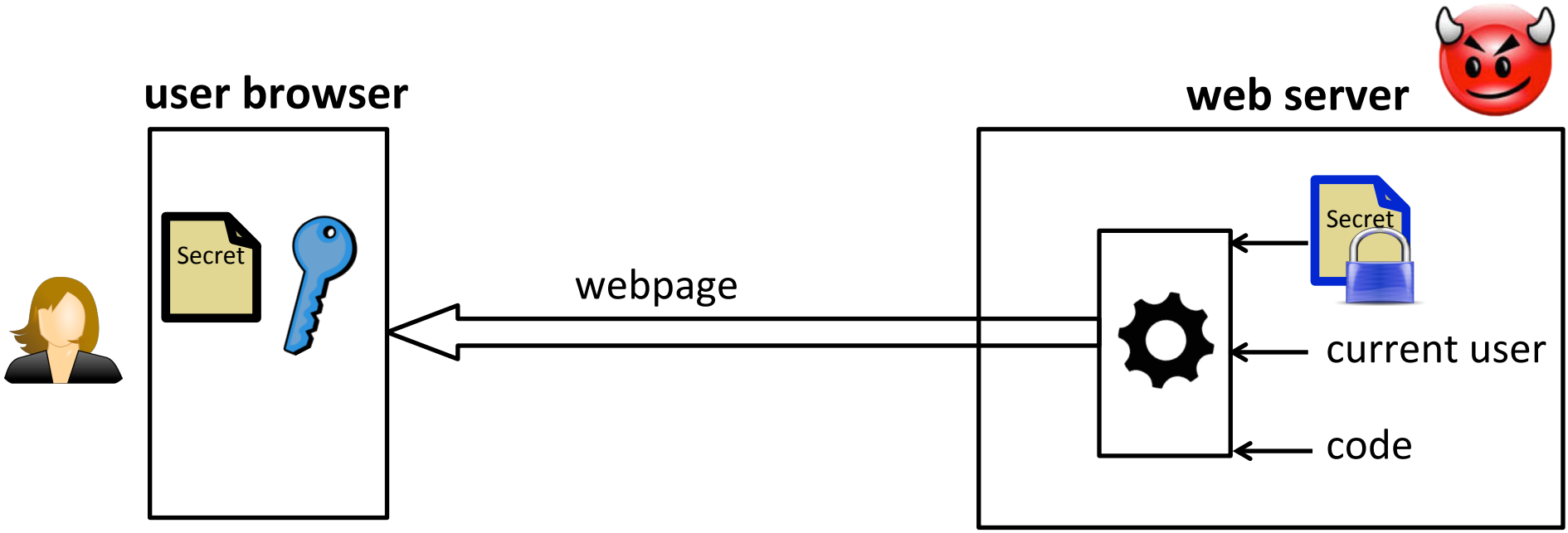
How to organize a web application framework for encryption?

Start: common web framework

e.g., Django, Ruby on Rails



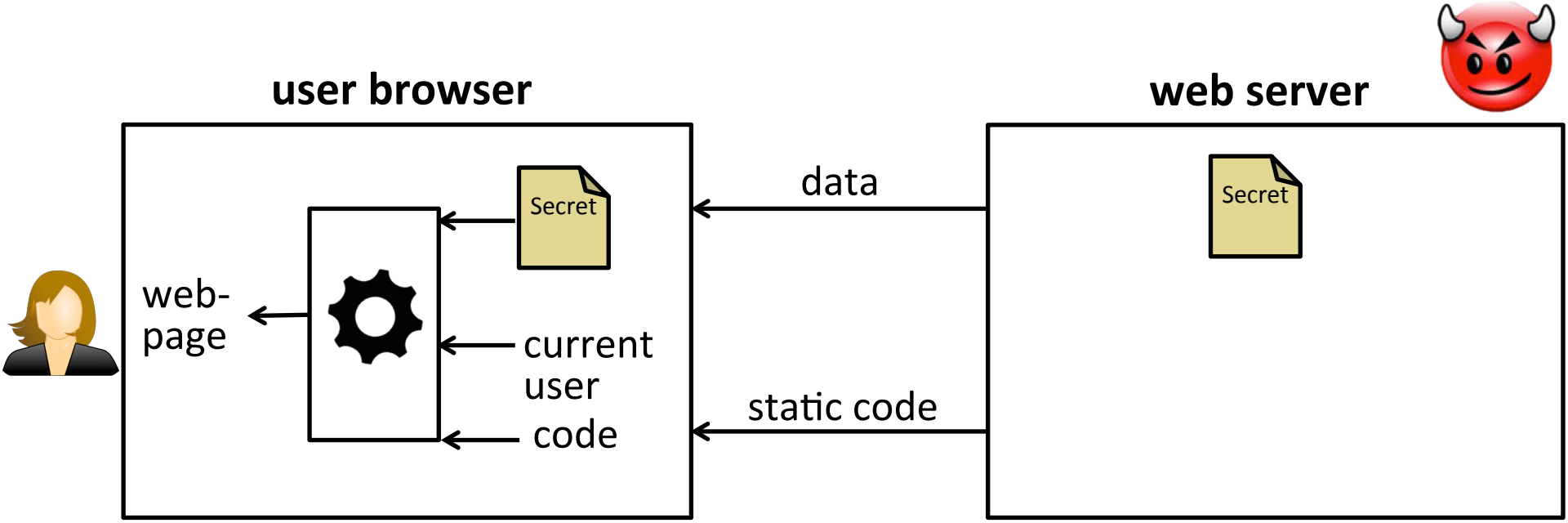
Add encryption



- server's computation is restricted by encryption
- easy to tamper with webpage

Client-side web framework

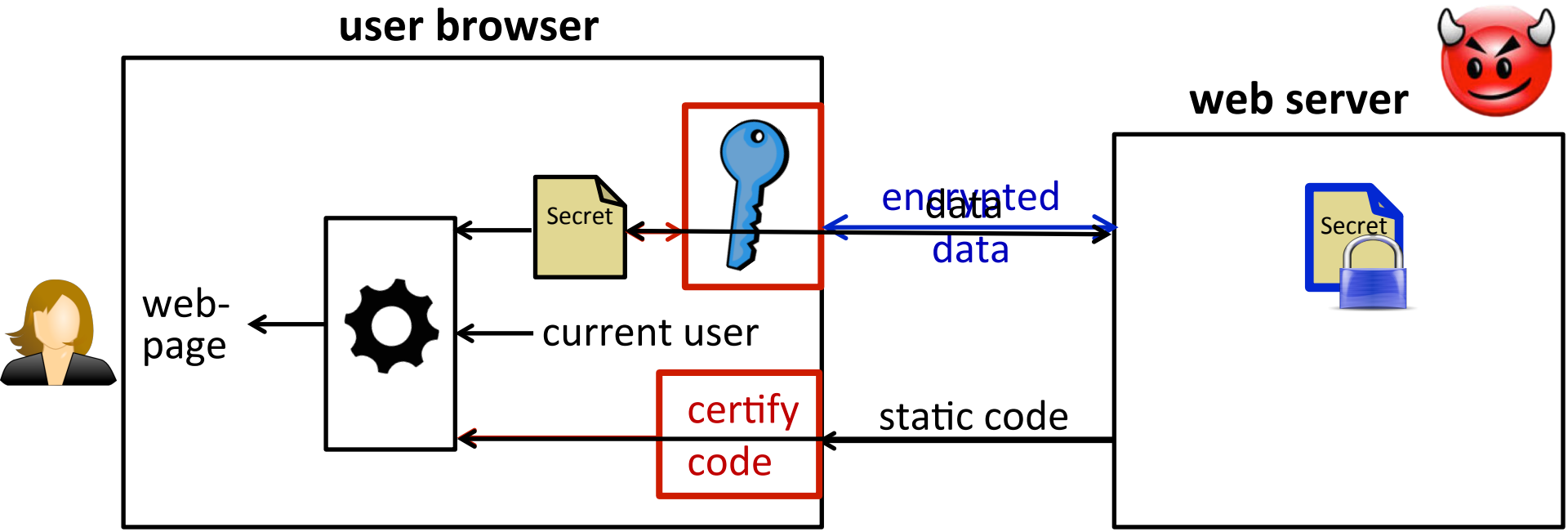
e.g., AJAX programming, Meteor



Data and code separate

Generate webpage at client, compute in browser

Mylar



Certify code (trusted developer)

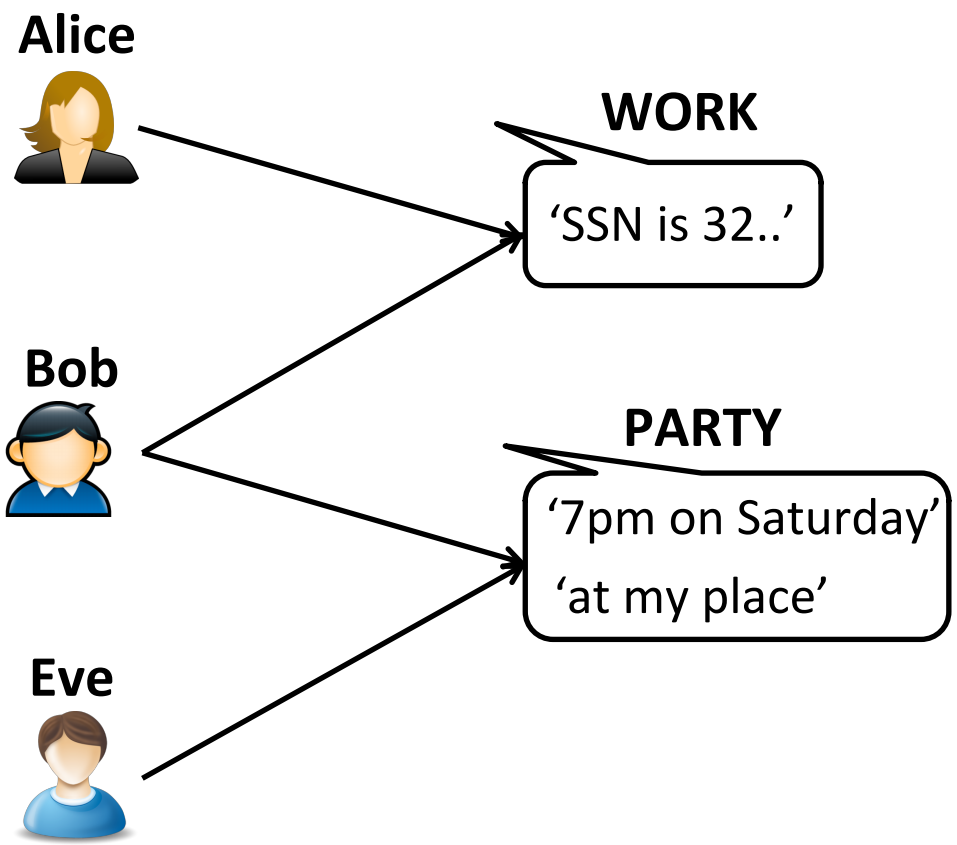
Intercept and encrypt/decrypt data

Chat application

- [x] Server cannot see messages
- [x] Format messages, generate html page
- [ ] Users share chat rooms securely
- [ ] Search

Data sharing

Developer specifies access control via the principal graph



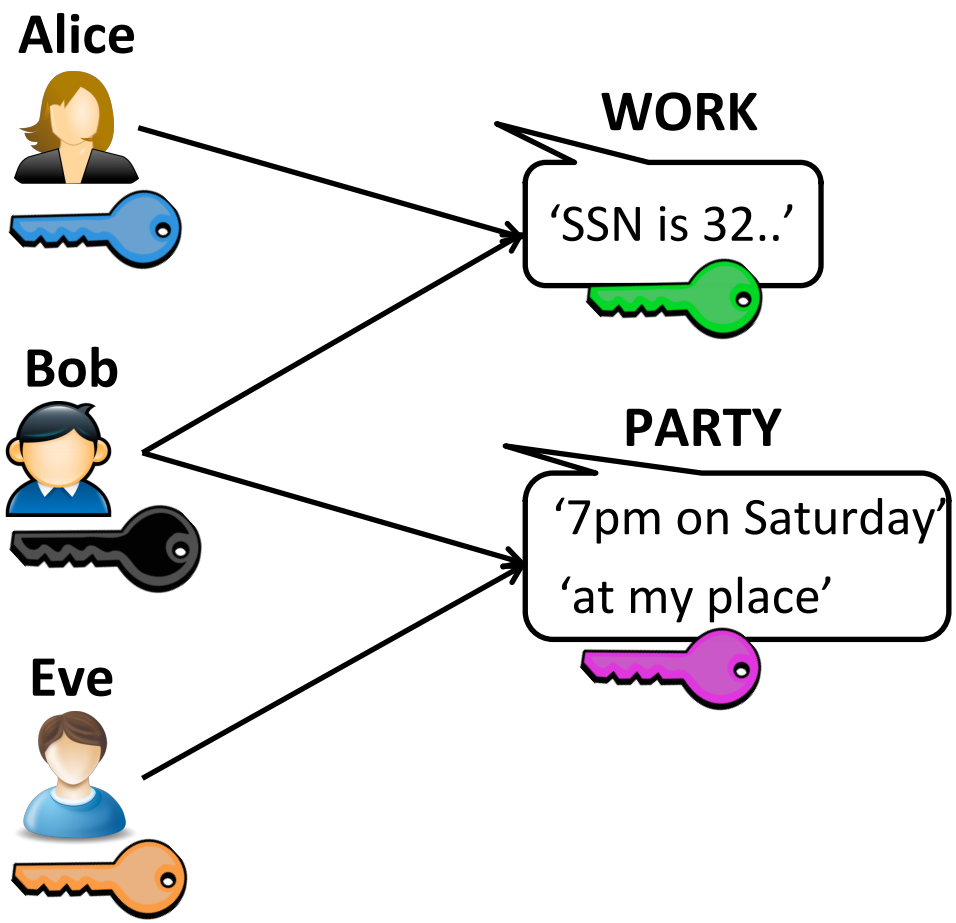
In Alice's browser:

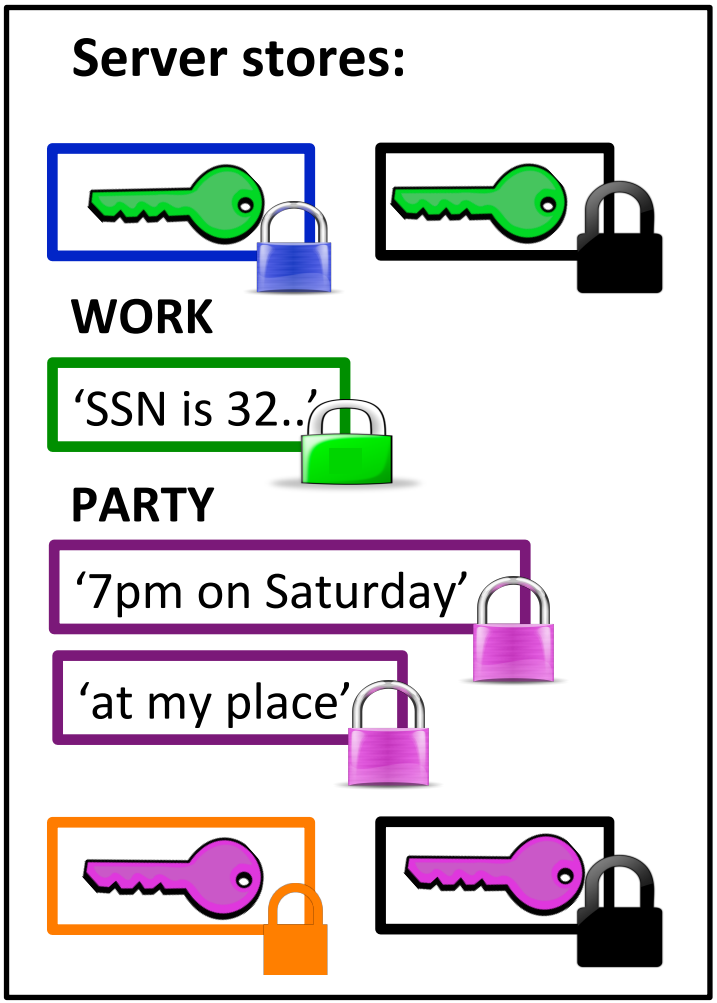
```
function create_chat(chattitle):
chat_princ = princ_create(chattitle,
                          princ_current());

function invite_user(username):
chat_princ.add_access(
                princ_lookup(username));

Messages.encrypted(
                {"message": chatprinc});

function send_message(msg):
Messages.insert({message: msg,
                chat: cur_chat.id,
                chatprinc: chat_princ});
```

Server database:

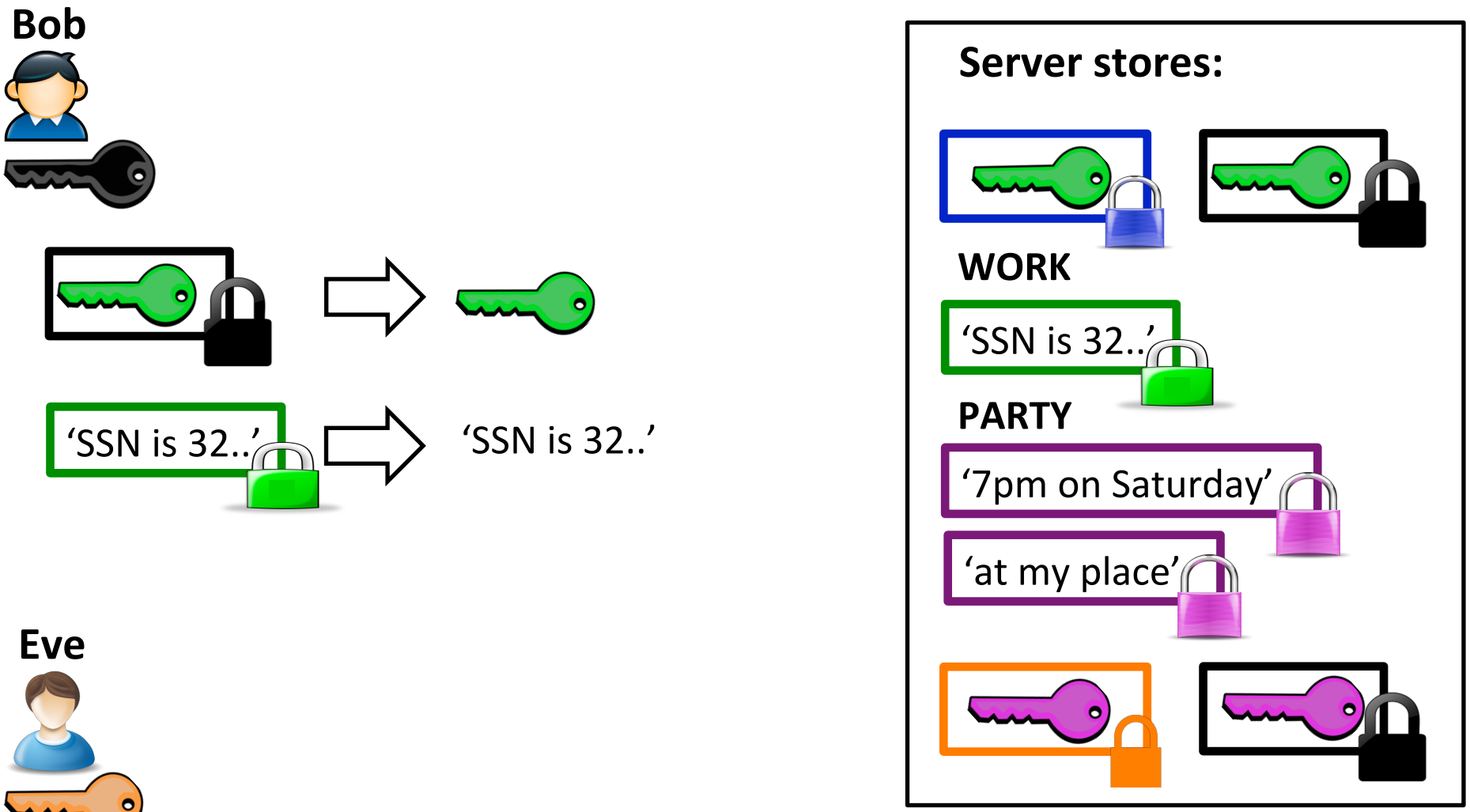
message	chat	chatprinc
'SSN is 32..'	WORK	WORK princ

Enforce access with key chains



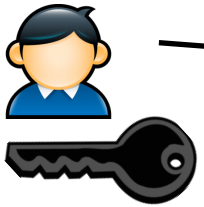


Get access to shared data



Problem: attacker gives incorrect key

Bob



Encrypt message for **WORK**

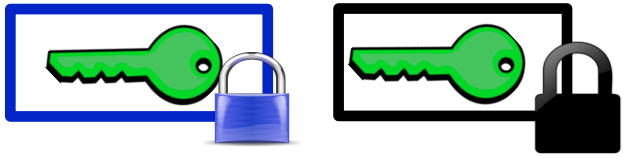
Want key

Receive key

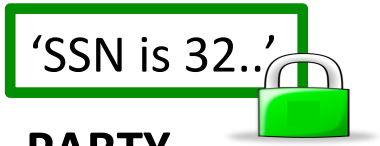
Eve

Has access to  !!!

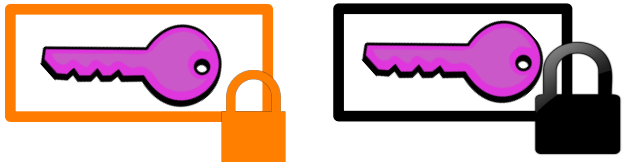
Server stores:



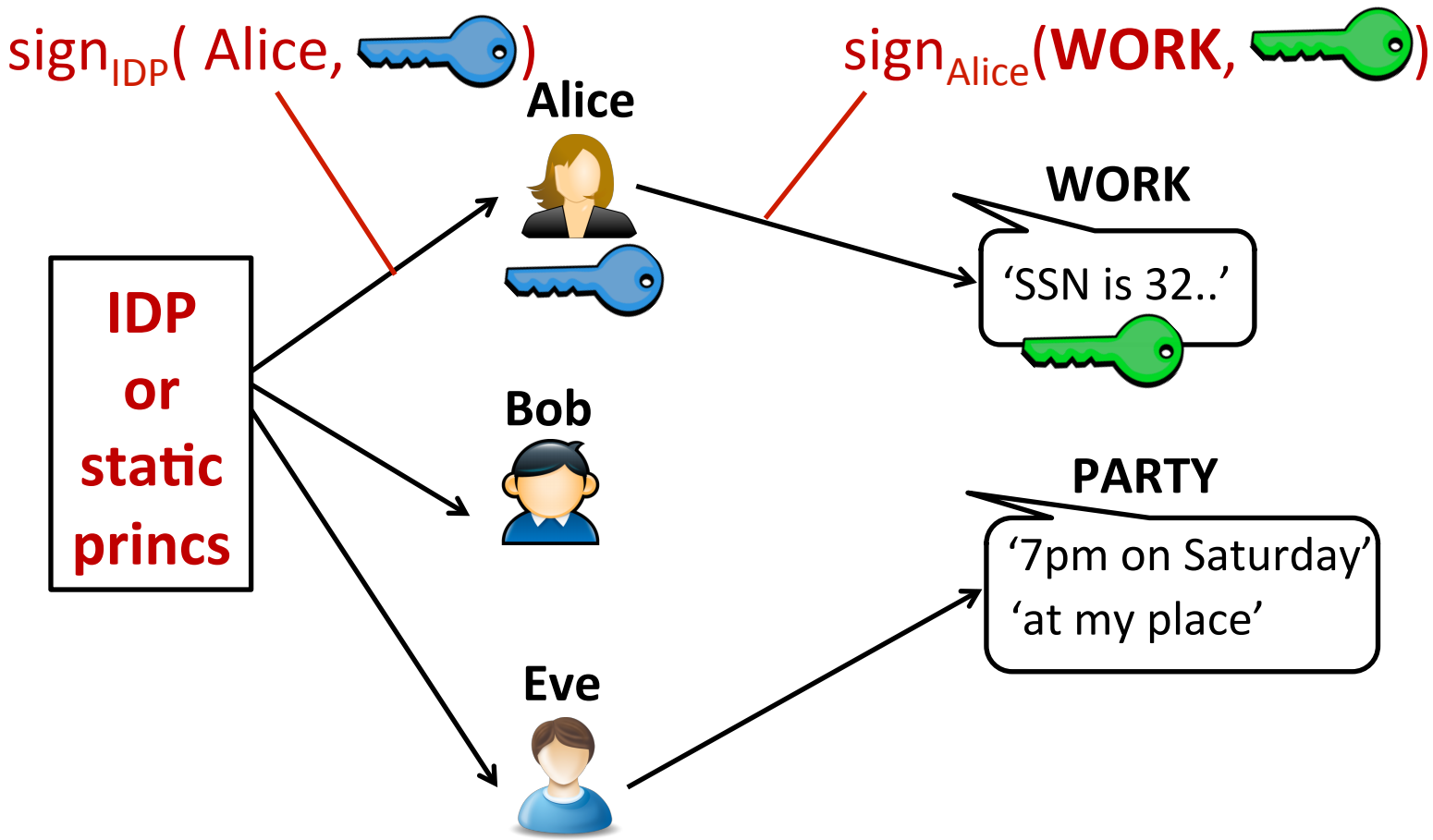
WORK



PARTY



Solution: Certification graph



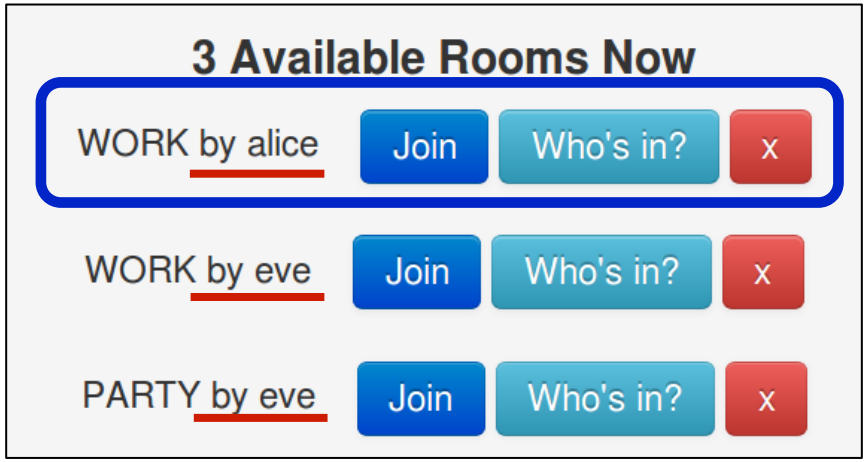
How does Bob's browser know

1. that it needs to check a signature from Alice?
2. Alice's PK? IDP: invoked once per user account creation

Choosing the certification path

1. Principals have human meaningful names
2. Developer displays entire path
3. User chooses path

with Mylar



without Mylar

No other change to user experience!

Chat application

- [x] Server cannot see messages
- [x] Format messages, generate html page
- [x] Users share chat rooms securely
- [ ] Search

Challenge: multi-key

Server:

Bob

WORK

'SSN is 32..'

PARTY

'7pm on Saturday'

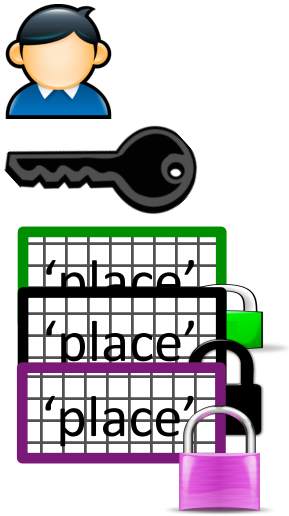
'at my place'

Strawman: use single-key search scheme

[Kamara et al.'12]

Server:

Bob



WORK

'SSN is 32..'

PARTY

'7pm on Saturday'

'at my place'

Match!

Slow: pay overhead for each key

New cryptosystem: multi-key search

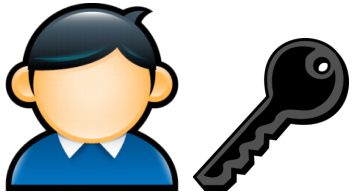
Server adjusts encryption from one key to another

Based on elliptic curves

API:

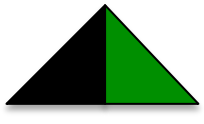
- Setup
- Keygen
- Encrypt
- Token
- Delta
- Adjust
- Match

Delta

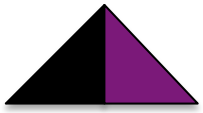


In Bob's browser:

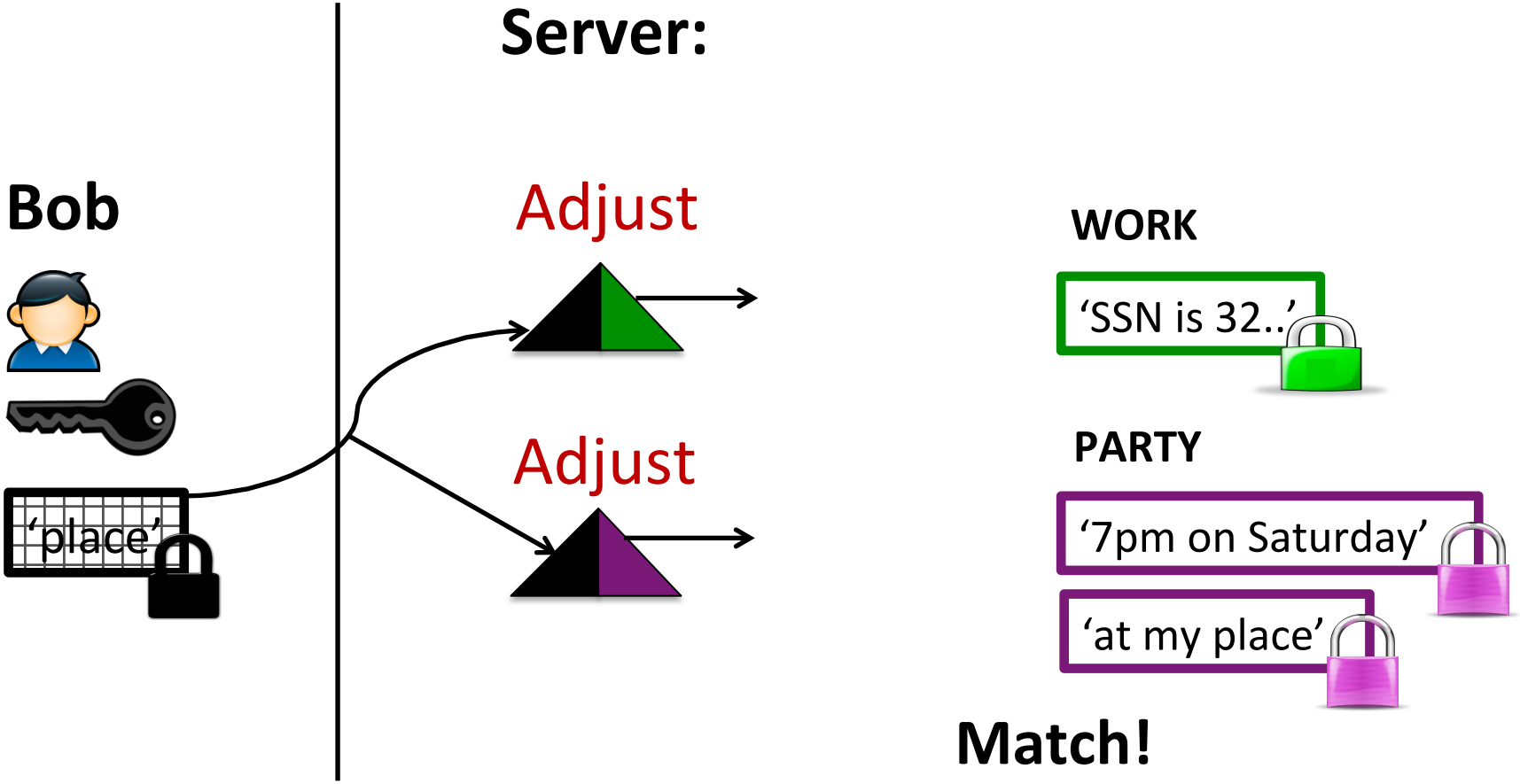
WORK:

Delta( , ) ⇨ 

PARTY:

Delta( , ) ⇨ 

Adjust


Server:
Bob
'place'
Adjust
Adjust
WORK
'SSN is 32..'
PARTY
'7pm on Saturday'
'at my place'
Match!

Chat application

- ☑ Server cannot see messages
- ☑ Format messages, generate html page
- ☑ Users share chat rooms securely
- ☑ Search

Confidentiality guarantees

Protects user A's data confidentiality against

- full server compromise
- compromise of any user machine, except for users with legitimate access to user A's data

assuming

- developer's client-side code does not leak data

Does not protect against side channels or access patterns, and does not hide metadata

Implementation

- On top of Meteor, but design is not limited to Meteor
- 9000 LoC: Javascript and C++

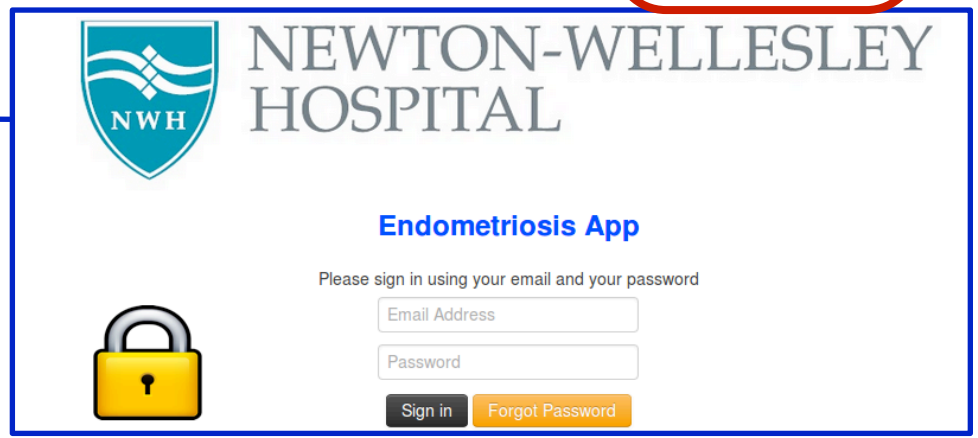
Evaluation

- How much developer effort does porting apps require?
- What is the performance overhead?

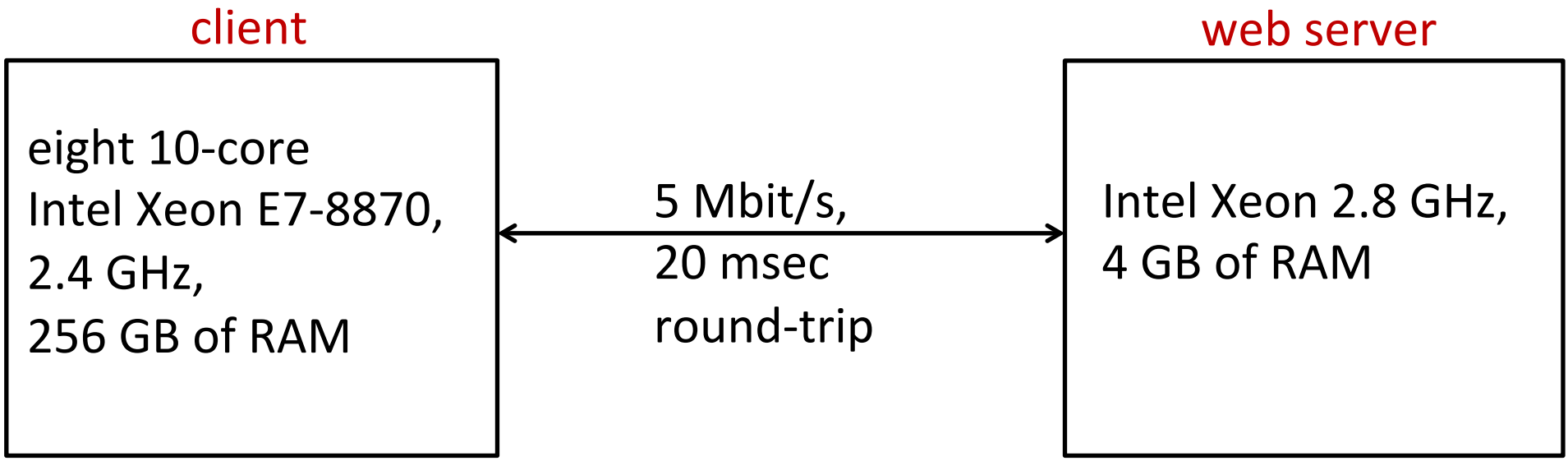
Applications

≈36 LoC

Applications	Fields secured	LoC added	LoC total	Existed before
kChat	chat messages	45	793	Yes
endometriosis	medical fields	28	3659	Yes
class submit	grades, homework, feedback	40	8410	Yes
photo sharing	photos, thumbnails, ..	32	610	No
forum	post body, title, ..	39	912	No
calendar	event body, title, ...	30	798	No

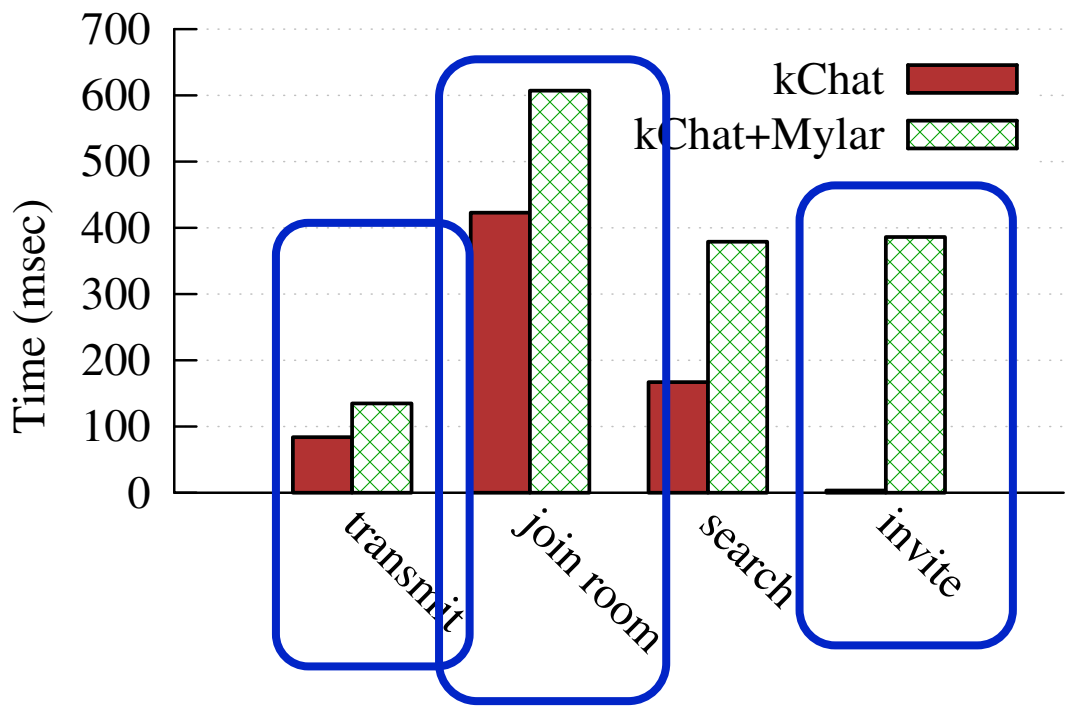


Experimental setup



kChat performance

Latency:



Throughput: 17% reduction

Mylar

- A web platform that protects confidentiality against full server compromise
  - Secures real applications with few LoC
  - Modest overhead

webpage code verification

principal graph & certification

new encryption scheme: multi-key search

http://css.csail.mit.edu/mylar/

Systems that compute on encrypted data

System	Project name
databases	**CryptDB** [SOSP'11][CACM'12] [Oakland'13]
web applications	**Mylar** [NSDI'14]
mobile applications	**PrivStats** [CCS'11] **VPriv** [Usenix Security'09]

Coming up

- machine learning classification over encrypted data [NDSS'15]
- network middleboxes **BlindBox & PrivMB**
  - intrusion detection, firewall, NAT over encrypted traffic!
- big data analytics over encrypted data
  - support complex computations
  - compress encrypted data!

Next challenge: Securing Internet-of-things