Cloud Computing Security

Nanyang Technical University, Singapore
6th February 2012

Mark D. Ryan
University of Birmingham

Cloud computing

Infrastructure as a service (IaaS)
Virtual machines running an operating system with storage.
Examples: Amazon web services, EC2 and S3.

Platform as a service (PaaS)
A specific computing platform and language.
Example: Google App Engine.

Software as a service (SaaS)
A specific software service.
Examples: Facebook, Google Docs, EasyChair,
iCloud, Dropbox, Gmail.

Inexorable rise of cloud computing

Motivation for cloud computing

- simplicity
- cost
- security
- resilience
- flexibility
- pace of innovation

Worldwide cloud services revenue is forecast to reach $68.3 billion in 2010 and projected to reach $148.8 billion by 2014, accordin g to a report by consultancy firm KPMG.

Consequences

- Google Apps is a packaged version of Google Docs, Mail, Calendar and other products under a custom domain name.
- In January 2011, 3,000 businesses were moving to Google Apps each day. . .
- . . . and three million have moved since its debut in 2007.

Google Apps

Security: main obstacle to deployment

Availability and integrity - no problem

Google claims “no scheduled downtime” (thanks to distributed back end) and guarantees 99.9% availability.

Confidentiality - big problem

- Confidentiality violation by the cloud provider. . .
  - Google’s business model involves mining its users’ data.
- . . . and its individual employees:
  - In July 2010, Google dismissed an engineer for accessing user accounts.
- . . . and third-party adversaries
  - In December 2010, Google reported a “highly sophisticated and targetted attack” designed to steal information about users from Gmail.

The Cloud Security Alliance highlights loss of data confidentiality, malicious insiders, technology vulnerabilities and service hijacking as four of its seven top threats of cloud computing.

CSA cloud security alliance℠

Cloud Security Alliance top threats

1. Abuse and nefarious use of cloud computing
2. Insecure interfaces and APIs
3. Malicious insiders
4. Shared technology issues
5. Data loss or leakage
6. Account or service hijacking
7. Unknown risk profile

Can we avoid trusting cloud service providers?

This need to trust is a problem that's behind many of CSA's threats.

Approaches:

- Fully homomorphic encryption
- Searchable encryption
- **Client-side key translation** (this talk)
- Hardware-based security

Fully homomorphic encryption

Won't *ever* be fast enough?

- Key and ciphertext sizes way too big.
  (currently measured in Gb)
- It's public-key crypto without a "data key" / "session key".
  (Normally, we never encrypt data with a public key.)

Useful enough?

- Given $L = [\{x_1\}_{pk}, \{x_2\}_{pk},\ \ldots,\ \{x_n\}_{pk}]$ and a predicate P,
  - can compute $[\{P(x_1)\}_{pk}, \{P(x_2)\}_{pk},\ \ldots,\ \{P(x_n)\}_{pk}]$
  - but cannot compute $[\{x\}_{pk} \in L \mid P(x)]$
    $\rightarrow$ "spam tagging" but not "spam filtering"

Confichair: Privacy-supporting EasyChair

Myrto Arapinis, Sergiu Bursuc, **Mark D. Ryan**
School of Computer Science, University of Birmingham

EasyChair: the little Facebook

Year	#confs
2002	2
2003	3
2004	7
2005	66
2006	276
2007	629
2008	1312
2009	2183
2010	3305
2011	$\geq$ 4517

2012: 15,262 conferences, 573,304 users

EasyChair data about Mark Ryan, 2005-2011

EasyChair data about Mark Ryan, 2005-2011

EasyChair admins can see or derive, e.g., the following data: Mark Ryan reviewed papers by Anwitaman Datta (CALCO'11), Wang Wenqiang (ESOP'11), Dinh Tien Tuan Anh (ESORICS'10), Stefano Braghin (FM'10) and others.

EasyChair data about Mark Ryan, 2005-2011

EasyChair admins can see or derive, e.g., the following data: Mark Ryan reviewed papers by Anwitaman Datta (CALCO'11), Wang Wenqiang (ESOP'11), Dinh Tien Tuan Anh (ESORICS'10), Stefano Braghin (FM'10) and others. Recommended *reject* for all of them.

EasyChair data about Mark Ryan, 2005-2011

EasyChair admins can see or derive, e.g., the following data: Mark Ryan reviewed papers by Anwitaman Datta (CALCO'11), Wang Wenqiang (ESOP'11), Dinh Tien Tuan Anh (ESORICS'10), Stefano Braghin (FM'10) and others. Recommended *reject* for all of them.

Mark Ryan's papers were reviewed by Liu Xin (S&P'10), Li Chenliang (TRUST'09), Rajesh Sharma (POPL'08), Quach Vinh Thanh (CAV'09), etc.

EasyChair data about Mark Ryan, 2005-2011

EasyChair admins can see or derive, e.g., the following data: Mark Ryan reviewed papers by Anwitaman Datta (CALCO'11), Wang Wenqiang (ESOP'11), Dinh Tien Tuan Anh (ESORICS'10), Stefano Braghin (FM'10) and others. Recommended *reject* for all of them.

Mark Ryan's papers were reviewed by Liu Xin (S&P'10), Li Chenliang (TRUST'09), Rajesh Sharma (POPL'08), Quach Vinh Thanh (CAV'09), etc. They all recommended *accept*.

EasyChair data about Mark Ryan, 2005-2011

EasyChair admins can see or derive, e.g., the following data: Mark Ryan reviewed papers by Anwitaman Datta (CALCO'11), Wang Wenqiang (ESOP'11), Dinh Tien Tuan Anh (ESORICS'10), Stefano Braghin (FM'10) and others. Recommended *reject* for all of them.

Mark Ryan's papers were reviewed by Liu Xin (S&P'10), Li Chenliang (TRUST'09), Rajesh Sharma (POPL'08), Quach Vinh Thanh (CAV'09), etc. They all recommended *accept*.

number of papers submitted	25
number of papers accepted	17
Acceptance rate	0.68
number of papers reviewed	107
number of times recommended accept	24
Recomendation agr. w. outcome	28%

EasyChair data about Mark Ryan, 2005-2011

EasyChair admins can see or derive, e.g., the following data: Mark Ryan reviewed papers by Anwitaman Datta (CALCO'11), Wang Wenqiang (ESOP'11), Dinh Tien Tuan Anh (ESORICS'10), Stefano Braghin (FM'10) and others. Recommended *reject* for all of them.

Mark Ryan's papers were reviewed by Liu Xin (S&P'10), Li Chenliang (TRUST'09), Rajesh Sharma (POPL'08), Quach Vinh Thanh (CAV'09), etc. They all recommended *accept*.

number of papers submitted	25
number of papers accepted	17
Acceptance rate	0.68
number of papers reviewed	107
number of times recommended accept	24
Recomendation agr. w. outcome	28%
Probability CSF 2012 re-invites him	0.2
Prob. will win ACM Turing award	$2^{-11.2}$

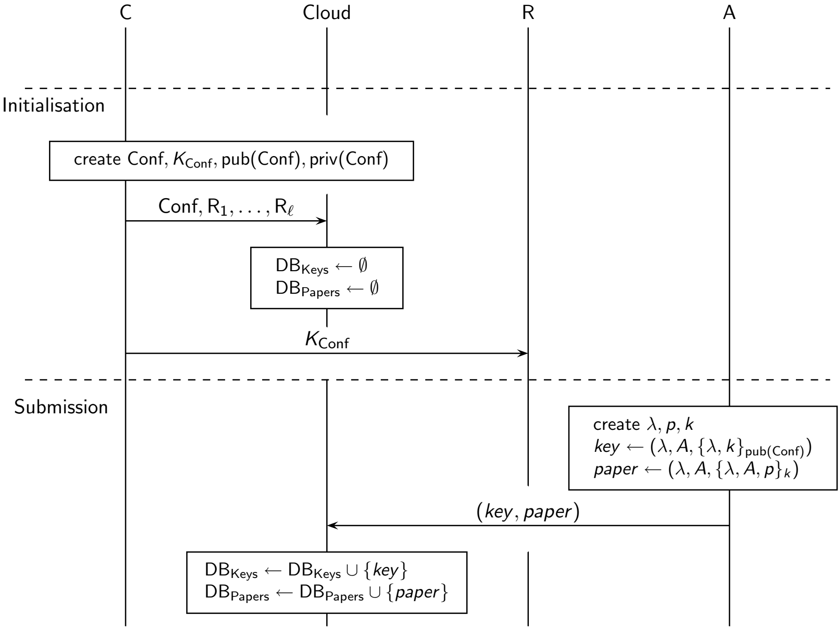
The service provider can

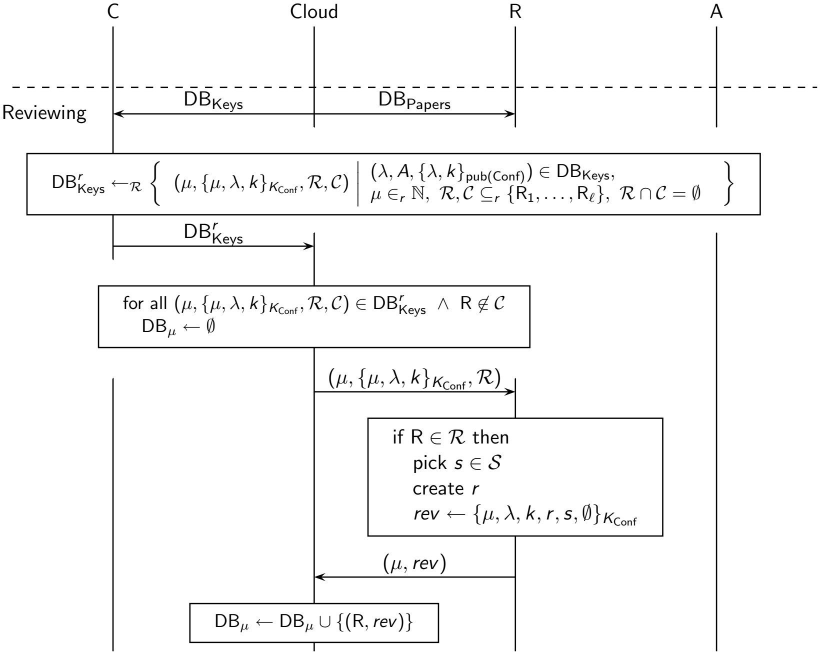
- Read and modify submissions and reviews.
- Impersonate participants.
- Write fake reviews.
- Delete accounts, deny service.
- Serve targetted advertisements using users' data.
- Publish submission/reviewing profiles for funding/awards committees etc.
- Sell services e.g. employment references for academics.

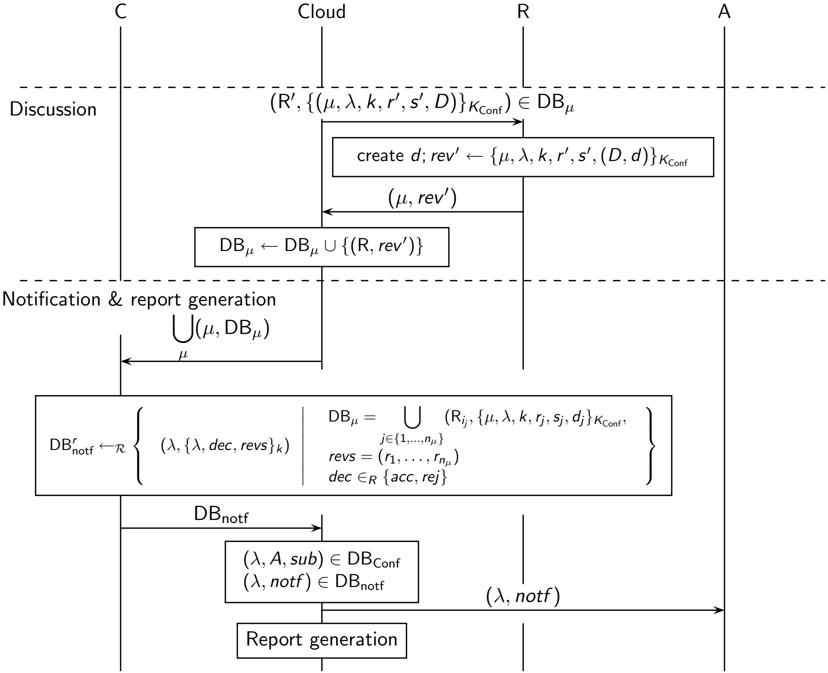
A third-party attacker can use server vulnerabilities to

- obtain submissions and reviews.
- modify or delete data, impersonate participants, write fake reviews.

Confichair


C
Cloud
R
A
Initialisation
create Conf, K_{Conf}, pub(Conf), priv(Conf)
Conf, $R_1, \ldots, R_\ell$
$DB_{\mathsf{Keys}} \leftarrow \emptyset$
$DB_{\mathsf{Papers}} \leftarrow \emptyset$
K_{Conf}
Submission
create λ, p, k
$key \leftarrow (\lambda, A, \{\lambda, k\}_{\mathsf{pub(Conf)}})$
$paper \leftarrow (\lambda, A, \{\lambda, A, p\}_k)$
$(key, paper)$
$DB_{\mathsf{Keys}} \leftarrow DB_{\mathsf{Keys}} \cup \{key\}$
$DB_{\mathsf{Papers}} \leftarrow DB_{\mathsf{Papers}} \cup \{paper\}$


C
Cloud
R
A
Reviewing
$\mathsf{DB}_{\mathsf{Keys}}$
$\mathsf{DB}_{\mathsf{Papers}}$
$\mathsf{DB}^r_{\mathsf{Keys}} \leftarrow_{\mathcal{R}} \left\{ (\mu, \{\mu, \lambda, k\}_{K_{\mathsf{Conf}}}, \mathcal{R}, \mathcal{C}) \;\middle|\; \begin{array}{l} (\lambda, A, \{\lambda, k\}_{\mathsf{pub(Conf)}}) \in \mathsf{DB}_{\mathsf{Keys}}, \\ \mu \in_r \mathbb{N},\ \mathcal{R}, \mathcal{C} \subseteq_r \{\mathsf{R}_1, \ldots, \mathsf{R}_\ell\},\ \mathcal{R} \cap \mathcal{C} = \emptyset \end{array} \right\}$
$\mathsf{DB}^r_{\mathsf{Keys}}$
for all $(\mu, \{\mu, \lambda, k\}_{K_{\mathsf{Conf}}}, \mathcal{R}, \mathcal{C}) \in \mathsf{DB}^r_{\mathsf{Keys}} \ \wedge\ \mathsf{R} \notin \mathcal{C}$
$\mathsf{DB}_\mu \leftarrow \emptyset$
$(\mu, \{\mu, \lambda, k\}_{K_{\mathsf{Conf}}}, \mathcal{R})$
if $\mathsf{R} \in \mathcal{R}$ then
pick $s \in \mathcal{S}$
create r
$rev \leftarrow \{\mu, \lambda, k, r, s, \emptyset\}_{K_{\mathsf{Conf}}}$
(μ, rev)
$\mathsf{DB}_\mu \leftarrow \mathsf{DB}_\mu \cup \{(\mathsf{R}, rev)\}$


C
Cloud
R
A
Discussion
$(\mathsf{R}', \{(\mu, \lambda, k, r', s', D)\}_{K_{\mathsf{Conf}}}) \in \mathsf{DB}_\mu$
create d; $rev' \leftarrow \{\mu, \lambda, k, r', s', (D, d)\}_{K_{\mathsf{Conf}}}$
(μ, rev')
$\mathsf{DB}_\mu \leftarrow \mathsf{DB}_\mu \cup \{(\mathsf{R}, rev')\}$
Notification & report generation
$\bigcup_\mu (\mu, \mathsf{DB}_\mu)$
$\mathsf{DB}^r_{\mathsf{notf}} \leftarrow_{\mathcal{R}} \left\{ (\lambda, \{\lambda, dec, revs\}_k) \;\middle|\; \begin{array}{l} \mathsf{DB}_\mu = \bigcup_{j \in \{1,\ldots,n_\mu\}} (\mathsf{R}_{i_j}, \{\mu, \lambda, k, r_j, s_j, d_j\}_{K_{\mathsf{Conf}}}, \\ revs = (r_1, \ldots, r_{n_\mu}) \\ dec \in_R \{acc, rej\} \end{array} \right\}$
$\mathsf{DB}_{\mathsf{notf}}$
$(\lambda, A, sub) \in \mathsf{DB}_{\mathsf{Conf}}$
$(\lambda, notf) \in \mathsf{DB}_{\mathsf{notf}}$
$(\lambda, notf)$
Report generation

Formal verification

Formal model

Term algebra $\mathcal{T}(\Sigma, \mathcal{N} \cup \mathcal{X})$

$$\begin{array}{rcl}
\mathcal{X} & = & x, y, z, \ldots \\
\mathcal{N} & = & a, b, c, k_1, k_2, \ldots \\
\Sigma & = & \{\mathsf{senc}(\_,\_,\_), \mathsf{sdec}(\_,\_), \mathsf{pub}(\_), \mathsf{aenc}(\_,\_,\_), \mathsf{adec}(\_,\_), \\
 & & \quad \langle \_,\_ \rangle, \mathsf{proj}_1(\_), \mathsf{proj}_2(\_)\}
\end{array}$$

Process calculus ProVerif [Blanchet'2001]

$$\begin{array}{rl}
P, Q, R ::= & 0 \\
 & P \mid Q \\
 & !P \\
 & \text{new } n; P \\
 & \text{let } M = D \text{ in } P \text{ else } Q \\
 & \text{in}(c, M); P \\
 & \text{out}(c, M); P
\end{array}$$

Operational semantics

Term rewriting

$$\begin{array}{rclcrcl}
\mathsf{sdec}(x, \mathsf{senc}(x, y, z)) & \to & z & \qquad & \mathsf{proj}_1(\langle x, y\rangle) & \to & x \\
\mathsf{adec}(x, \mathsf{aenc}(\mathsf{pub}(x), y, z)) & \to & z & \qquad & \mathsf{proj}_2(\langle x, y\rangle) & \to & y
\end{array}$$

Process reduction

$$\begin{array}{l}
\mathrm{out}(c, M).P \mid \mathrm{in}(c, x).Q \to P \mid Q\{^{M}/_{x}\} \\
\mathrm{let}\ M = D\ \mathrm{in}\ P\ \mathrm{else}\ Q \to P\sigma,\ \text{if } D \Downarrow N\ \&\ \sigma = \mu(M, N) \\
\mathrm{let}\ M = D\ \mathrm{in}\ P\ \mathrm{else}\ Q \to Q,\ \text{otherwise}
\end{array}$$

Observational equivalence

Observation $P \Downarrow c$:

$$\exists C[\_] \exists Q, \exists M. \qquad P \rightarrow^* C[\mathrm{out}(c, M).Q]$$

Largest equivalence relation s.t. $P \sim Q$ implies

$$\begin{array}{lrcl} 1. & P \Downarrow c & \Longrightarrow & Q \Downarrow c \\ 2. & P \rightarrow^* P' & \Longrightarrow & \exists Q'.\ Q \rightarrow^* Q' \ \&\ P' \sim Q' \\ 3. & \forall C[\_]. & & C[P] \sim C[Q] \end{array}$$

Secrecy in conference systems

$$\begin{array}{rccc} \text{Papers:} & P_{\text{conf}} & \rightsquigarrow & P^{\mathbf{P}}_{\text{conf}}[\_] \\[1em] \text{Reviews:} & P_{\text{conf}} & \rightsquigarrow & P^{\mathbf{R}}_{\text{conf}}[\_] \\[1em] \text{Scores:} & P_{\text{conf}} & \rightsquigarrow & P^{\mathbf{S}}_{\text{conf}}[\_] \end{array}$$

- Secrecy of papers: $P^{\mathbf{P}}_{\text{conf}}[\text{pap}] \sim P^{\mathbf{P}}_{\text{conf}}[\text{pap'}]$
- Secrecy of reviews: $P^{\mathbf{R}}_{\text{conf}}[\text{rev}] \sim P^{\mathbf{R}}_{\text{conf}}[\text{rev'}]$
- Secrecy of scores: $P^{\mathbf{S}}_{\text{conf}}[\text{sc}] \sim P^{\mathbf{S}}_{\text{conf}}[\text{sc'}]$

Unlinkability in conference systems

Author-Reviewer:

$$P^{\mathbf{AR}}_{\mathsf{conf}}(\mathsf{a}, \mathsf{ra}) | P^{\mathbf{AR}}_{\mathsf{conf}}(\mathsf{b}, \mathsf{rb}) \sim P^{\mathbf{AR}}_{\mathsf{conf}}(\mathsf{a}, \mathsf{rb}) | P^{\mathbf{AR}}_{\mathsf{conf}}(\mathsf{b}, \mathsf{ra})$$

Summary

"Conficháir"

- C does not know p, r and s
- C knows A and R, but does not know the link $A \longleftrightarrow R$

We formalised the properties, and verified them with ProVerif.

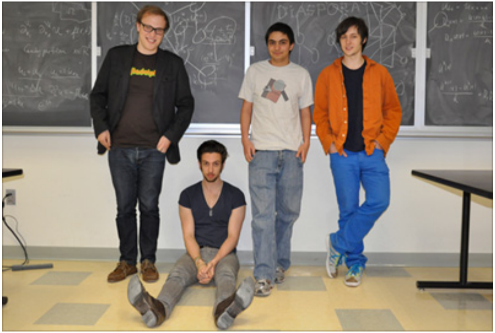
Prototype implementation by Matt Roberts and Joshua Phillips
`confichair.markryan.eu` *(temporary location)*

What about social networking?

Diaspora

- A free personal web server that implements a distributed social networking service, providing a decentralised alternative to social networks like Facebook.

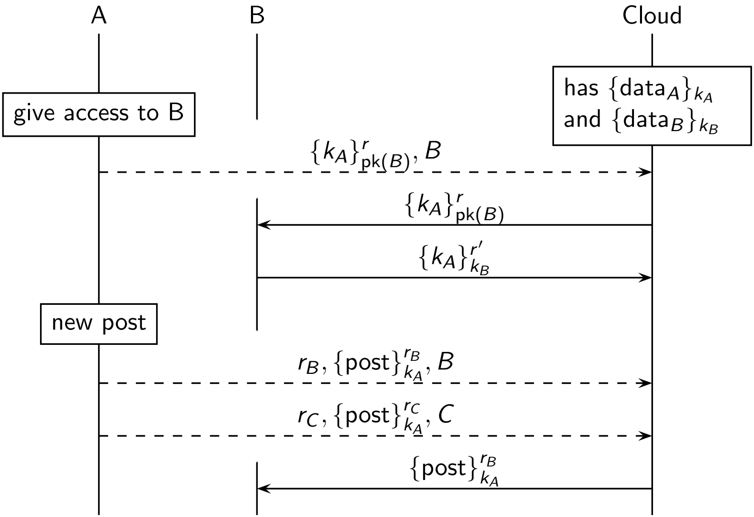
- Works by letting users set up their own server (or “pod”) to host content. Pods interact to share status updates and photographs. A pod can be hosted on a traditional web host, a cloud-based pod server, an ISP or with a friend.



Distributed social networks

- Wikipedia lists 39 of them, in various stages of development (development, pre-alpha, alpha, beta, production).
- Some of the listed ones are now defunct.

Centralised encrypted social networking


A
B
Cloud
has $\{\text{data}_A\}_{k_A}$
and $\{\text{data}_B\}_{k_B}$
give access to B
$\{k_A\}^r_{\text{pk}(B)}, B$
$\{k_A\}^r_{\text{pk}(B)}$
$\{k_A\}^{r'}_{k_B}$
new post
$r_B, \{\text{post}\}^{r_B}_{k_A}, B$
$r_C, \{\text{post}\}^{r_C}_{k_A}, C$
$\{\text{post}\}^{r_B}_{k_A}$

Survey

Questions

1. Do you want confidentiality of mail?
   Yes, but not if it means I have to maintain a PGP key ring and worry about key certificates.

Questions

1. Do you want confidentiality of mail?
   Yes, but not if it means I have to maintain a PGP key ring and worry about key certificates.

2. Do you want confidentiality of Facebook?
   That's even worse. There are geeky solutions where the data is distributed among all the users.

Questions

1. Do you want confidentiality of mail?
   Yes, but not if it means I have to maintain a PGP key ring and worry about key certificates.

2. Do you want confidentiality of Facebook?
   That's even worse. There are geeky solutions where the data is distributed among all the users.

3. Do you want confidentiality of conference mgt?
   The cost:

   - Authors need to paste a public key from the CFP into their browser.
   - Reviewers need to paste a symm. key from their mail into their browser.
   - Chair needs to use their browser to set up the two keys, and publish them by CFP and mail respectively.
   - Chair needs to perform decrypt-encryptions and mixes, which is done by clicking a few buttons in her browser.