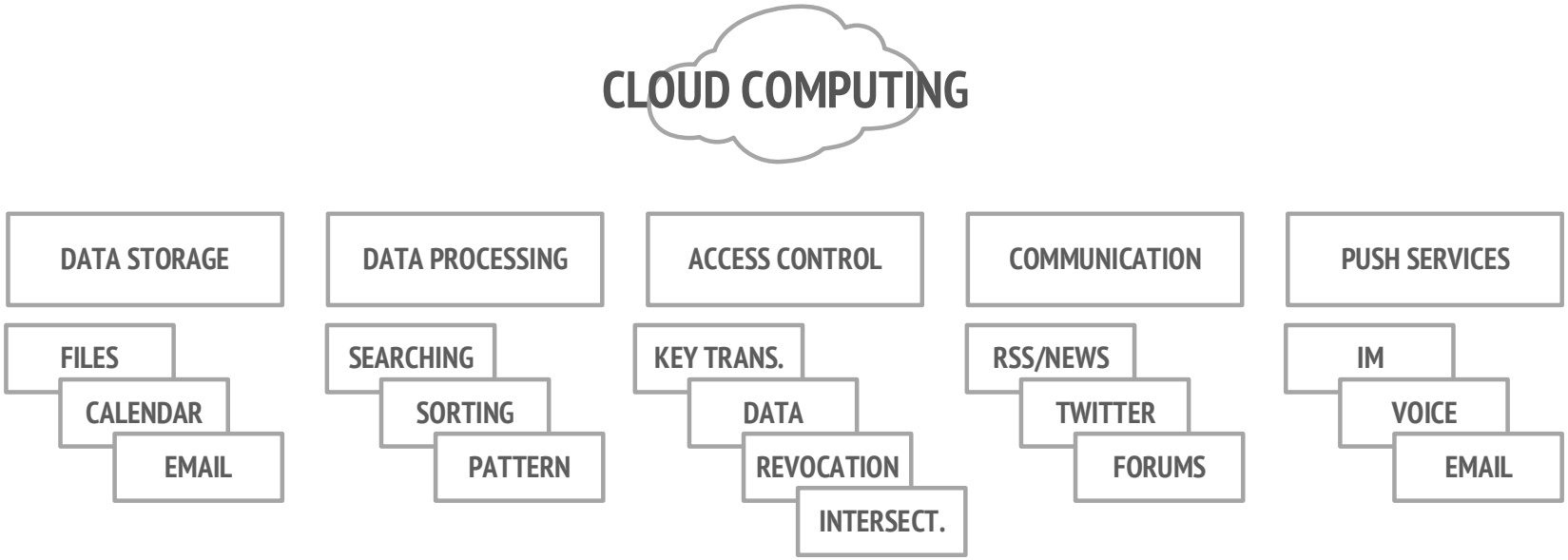
# Cloud Data Security

Mihai Ordean
Designing and Managing Secure Systems
University of Birmingham

# Overview

- Device security
  - Is code on the device vulnerable to exploits ? (e.g. buffer overflows)
  - Is the code authenticated ? (i.e. has not been tampered with)
- Local data security
  - Is the stored data is accessible to everyone? (e.g. encrypted)
  - Is the stored data authenticated?
- **Cloud data security**
  - How is data stored in the cloud?
  - Who has access to data stored in the cloud?
- Metadata security
  - What does metadata reveal about stored data?
  - Can we tamper the metadata?

# What is cloud computing?



# Types of cloud

- Public cloud: anyone can share the cloud resource with you
- Private cloud: the cloud is owned/maintained by your own organization
- Community cloud: a group gets together to jointly own/manage a cloud infrastructure
- Hybrid cloud: some mix of public/private e.g., private cloud does data processing, public cloud does bulk storage

# Cloud computation requirements



# Motivation for using cloud computing

- Simplicity
- Cost
- Security
- Resilience
- Flexibility
- Pace of innovation

# Security issues (generic)

- Prevent
- Deter
- Detect
- Respond
- Recover
- …

- Confidentiality, integrity, availability
  - Authorisation, management
  - Physical security, personnel security (e.g. employee vetting)
  - Privacy management

# Cloud Security Alliance

- The Cloud Security Alliance (CSA) is a not-for-profit organization with a mission to “promote the use of best practices for providing security assurance within Cloud Computing, and to provide education on the uses of Cloud Computing to help secure all other forms of computing”.

- The CSA has over 80,000 individual members worldwide. CSA gained significant reputability in 2011 when the White House selected the CSA Summit as the venue for announcing the federal government’s cloud computing strategy.

# Top 12 cloud security issues (latest)

1. Data breaches
2. Insufficient identity, credential and access management
3. Insecure interfaces and APIs
4. System vulnerabilities
5. Account hijacking
6. Malicious insiders
7. Advanced Persistent Threats
8. Data loss
9. Insufficient due diligence
10. Abuse and nefarious use of cloud services
11. Denial of service
12. Shared technology issues

# Data breaches

- **2018: Facebook security breach: Up to 50m accounts attacked.** The company said attackers were able to exploit a vulnerability in a feature known as “View As” to gain control of people's accounts.
- **2017:** A breach exposed the social security numbers and other data. **Credit firm Equifax says 143m Americans' social security numbers exposed in hack.**
- In mid-2015**, BitDefender**, an antivirus firm, had an **undisclosed number of customer usernames and passwords stolen** due to a security vulnerability in its public cloud application hosted on AWS. The hacker responsible demanded a ransom of $15,000.
- The 2015 Anthem breach of more than **80 million customer records began with stolen credentials** on the corporate network. A third-party cloud service was used to transfer the huge data store from the company's network to the public cloud where it could be downloaded by the hackers.
- British telecom provider **TalkTalk reported multiple security incidents in 2014 and 2015**, which resulted in the theft of four million customers' personal information.

A data breach is an incident in which sensitive, protected or confidential information is released, viewed, stolen or used by an individual who is not authorized to do so.

# Weak identity, credential and access management

- **Attackers Scrape GitHub for Cloud Service Credentials,** Hijack Account to Mine Virtual Currency "Cloud service provider credentials included in a GitHub project were discovered and misused within 36 hours of the project going live.“

- **Praetorian Launches Cloud-based Password Cracking Service** "Praetorian, an Austin, Texas-based provider of information security solutions, has launched a new cloud-based platform that leverages the computing power of Amazon AWS in order to crack password hashes in a simple fashion."

- Data breaches and enabling of attacks can occur because of a lack of scalable identity access management systems, **failure to use multi-factor authentication, weak password use**, and a lack of ongoing automated rotation of cryptographic keys, passwords and certificates.

- **Credentials and cryptographic keys must not be embedded in source code or distributed in public facing repositories such as GitHub**, because there is a significant chance of discovery and misuse. Keys need to be appropriately secured and a well-secured public key infrastructure (PKI) is needed to ensure key-management activities are carried out.

# Insecure APIs

- **The IRS Breach and the Importance of Adaptive API Security.** In mid-2015, the US Internal Revenue Service (IRS) exposed over 300,000 records via a vulnerable API ("Get Transcript API").

Cloud computing providers expose a set of software user interfaces (UIs) or application programming interfaces (APIs) that customers use to manage and interact with cloud services. Provisioning, management, orchestration and monitoring are all performed with these interfaces. The security and availability of general cloud services is dependent on the security of these basic APIs.

# System & application vulnerabilities

- **Magnified Losses, Amplified Need for Cyber-Attack Preparedness** "Heartbleed and Shellshock proved that even open source applications, which were believed more secure than their commercial counterparts... , were vulnerable to threats. They particularly affected systems running Linux, which is concerning given that **67.7% of websites** use UNIX, on which the former (Linux) is based."
- **Verizon 2015 Data Breach Investigations Report** "The Shellshock bug in Bash was 2014's second tumultuous OSS vulnerability event, quickly eclipsing Heartbleed due to many more successful attacks."
- **Cyberthreat Defense Report (2014)** "75% of attacks use publicly known vulnerabilities in commercial software that could be prevented by regular patching."

System vulnerabilities are exploitable bugs in programs that attackers can use to infiltrate a computer system for the purpose of stealing data, taking control of the system or disrupting service operations. Vulnerabilities within the components of the operating system kernel, system libraries and application tools put the security of all services and data at significant risk.

# System & application vulnerabilities

- **Shellshock** is a family of security bugs in the Unix Bash shell, the first of which was disclosed in September 2014. Many Internet-facing services, such as some web server deployments, use Bash to process requests, allowing an attacker to cause vulnerable versions of Bash to execute arbitrary commands. This can allow an attacker to gain unauthorized access to a computer system.

```
env x='() { :;}; echo vulnerable’ \
    bash -c "echo this is a test“
```

- **Heartbleed** is a security bug in the OpenSSL cryptography library, which is a widely used implementation of the Transport Layer Security (TLS) protocol. It was introduced into the software in 2012 and publicly disclosed in April 2014. Heartbleed results from improper input validation (due to a missing bounds check) in the implementation of the TLS heartbeat extension. The vulnerability allows a situation in which more data can be read than should be allowed; in particular, secret keys may be read by an attacker.

# Account hijacking

- In June 2014, **Code Spaces' Amazon AWS account was compromised** when it failed to protect the administrative console with multifactor authentication. All the company's assets were destroyed, putting it out of business.

- In April 2010, **Amazon experienced a cross-site scripting (XSS)** bug that allowed attackers to hijack credentials from the site. In 2009, numerous Amazon systems were hijacked to run Zeus botnet nodes.

Account or service hijacking is not new. Attack methods such as phishing, fraud and exploitation of software vulnerabilities still achieve results. Credentials and passwords are often reused, which amplifies the impact of such attacks. Cloud solutions add a new threat to the landscape. If an attacker gains access to your credentials, they can eavesdrop on your activities and transactions, manipulate data, return falsified information and redirect your clients to illegitimate sites. Your account or service instances may become a new base for attackers. From here, they may leverage the power of your reputation to launch subsequent attacks.

# Malicious insiders

- **Insider Threats to Cloud Computing** "Overall, the 'inside job' is responsible for most cloud computing security woes. Enterprises have to become proactive in finding solutions to their security threats to protect their sensitive information."
- **Cloud's Privileged Identity Gap Intensifies Insider Threats** "Organizations need to rein in shared accounts and do a better job tracking user activity across cloud architectures."
- **David Barksdale accessed users' accounts, violating the privacy of at least four minors** during his employment as Systems Engineer at Google. He boasted about it with friends. He was fired in July 2010 after his actions were reported to the company.

"A malicious insider threat to an organization is a current or former employee, contractor, or other business partner who has or had authorized access to an organization's network, system, or data and intentionally exceeded or misused that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information or information systems."

# Advanced Persistent Threats (APTs)

- **Carbanak**: How Would You Have Stopped a $1 Billion APT Attack? "... Carbanak, an APT attack against financial institutions around the world, may well be considered the largest cyberheist to date. ... Unlike the usual cybercriminal method of stealing consumer credentials or compromising individual online banking sessions with malware, the brazen Carbanak gang targeted banks' internal systems and operations, resulting in a multichannel robbery that averaged $8 million per bank."
- **Current Trends in the APT World** "The alleged Chinese Cyber-Espionage with its APTs caused the theft of "hundreds of terabytes of data from at least 141 organizations across a diverse set of industries beginning as early as 2006."
- **Stuxnet** specifically targets programmable logic controllers (PLCs), which allow the automation of electromechanical processes such as those used to control machinery on factory assembly lines, amusement rides, or centrifuges for separating nuclear material.

Advanced Persistent Threats (APTs) infiltrate systems to establish a long-lived foothold in the target platform, usually with political or commercial motives. APTs pursue their goals stealthily over extended periods of time. Once in place, APTs can move laterally through datacentre networks and blend in with normal network traffic to achieve their objectives.

# Data loss

- In June 2014, **Code Spaces**, an online hosting and code publishing provider, was hacked, leading to the compromise and complete destruction of most customer data. The company was ultimately unable to recover from this attack and went out of business
- In April 2011, **Amazon EC2** suffered a crash that led to significant data loss for many customers.

Data stored in the cloud can be lost for reasons other than malicious attacks. An accidental deletion by the cloud service provider, or worse, a physical catastrophe such as a fire or earthquake, can lead to the permanent loss of customer data unless the provider or cloud consumer takes adequate measures to back up data, following best practices in business continuity and disaster recovery as well as daily data backup and possibly off-site storage.

Furthermore, the burden of avoiding data loss does not fall solely on the provider's shoulders. If a customer encrypts his or her data before uploading it to the cloud but loses the encryption key, the data will be lost as well.

# Insufficient due diligence

- **Contract and Financial Viability** In 2013, Nirvanix, a cloud storage specialist that hosted data for IBM, Dell and its own customers, filed for Chapter 11 bankruptcy and shuttered its operations. Customers were given less than two weeks to move their data to another service. This caused huge problems for customers. For example:
  - Film and TV production studio Relativity Media was using Nirvanix's cloud as a hub through which employees in its global locations could collaborate and share massive digital files to accelerate production.
- **Non-Compliance:** Healthcare and financial services must retain data to meet government compliance regulations. If the data is lost, these services become non-compliant.

Companies should develop a good roadmap and checklist for due diligence when evaluating technologies and cloud service providers (CSPs). An organization that rushes to adopt cloud technologies and choose CSPs without performing due diligence exposes itself to a myriad of commercial, financial, technical, legal and compliance risks that jeopardize its success.

# Denial of service

- As **Cloud Use Grows, So Will Rate of DDoS Attacks** "Cloud providers face increasing number of DDoS attacks, [similar to those that] private data centers already deal with today"
- **Feedly Knocked Offline** by **DDoS Attack** Following Evernote and Deezer Attacks "In what looks like a series of coordinated cyber-attacks by a criminal gang, three major cloud-based services have all been knocked offline in recent days. News aggregator Feedly, note-taking app Evernote and music streaming service Deezer have all come under attack from criminals in the last few days leading to all three suffering service outages.

Denial-of-service (DoS) attacks are attacks meant to prevent users of a service from being able to access their data or their applications. By forcing the targeted cloud service to consume inordinate amounts of finite system resources such as processor power, memory, disk space or network bandwidth, the attackers cause an intolerable system slowdown and leaves all legitimate service users unable to access the service.

# Abuse and nefarious use of cloud services

MIRAI - the DDoS That Almost Broke the Internet "The attackers were able to generate more than 300 Gbps of traffic. Their own network only had access to 1/100th of that amount of traffic."

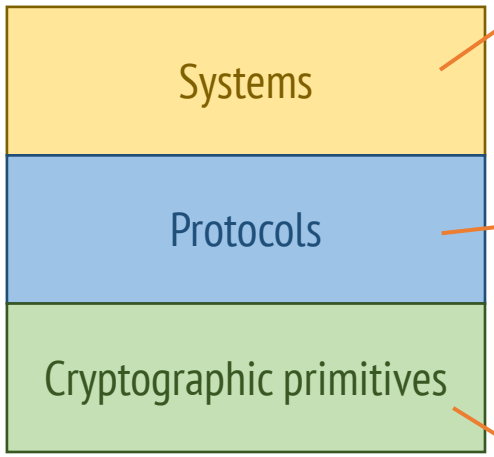
- Poorly secured cloud service deployments, free cloud service trials and fraudulent account sign-ups via payment instrument fraud expose cloud computing models IaaS, PaaS, and SaaS to malicious attacks.
- Malicious actors may leverage cloud computing resources to target users, organizations or other cloud providers. Examples of misuse of cloud service-based resources include launching DDoS attacks, email spam and phishing campaigns; "mining" for digital currency; large-scale automated click fraud; brute-force compute attacks of stolen credential databases; and hosting of malicious or pirated content.

# Shared technology issues

- **Cross-VM Side Channels and Their Use to Extract Private Keys** "...construction of an access-driven side-channel attack by which a malicious virtual machine (VM) extracts fine-grained information from a victim VM running on the same physical computer."
- **Understanding the VENOM Vulnerability** "The unchecked buffer vulnerability (CVE-2015-3456) occurs in the code for QEMU's virtual floppy disk controller. A successful buffer overflow attack exploiting this vulnerability can enable an attacker to execute his or her code in the hypervisor's security context and escape from the guest operating system to gain control over the entire host."

Cloud service providers deliver their services scalably by sharing infrastructure, platforms or applications. Underlying components (e.g., CPU caches, GPUs, etc.) that comprise the infrastructure supporting cloud services deployment may not have been designed to offer strong isolation properties for a multitenant architecture (IaaS), re-deployable platforms (PaaS) or multicustomer applications (SaaS). This can lead to shared technology vulnerabilities that can potentially be exploited in all delivery models. A defense-in-depth strategy is recommended and should include compute, storage, network, application and user security enforcement and monitoring, whether the service model is IaaS, PaaS, or SaaS. A single vulnerability or misconfiguration can lead to a compromise across an entire provider's cloud.

# The “security stack”




**Goals:** send encrypted messages (Signal), access network from remote location securely (IP-SEC)
**Systems examples:**
Secure messaging, secure file storage, Secure DB,…

**Goals:** Authenticate parties, generate session keys, establish end-to-end enc. channels…
**Cryptographic protocol examples:**
TLS, SSH, …

**Primitive examples:**
Symmetric crypto: AES, 3-DES, …
Public crypto: RSA, DH, …
Hash functions: SHA(1-3) , RC4, …
Digital signatures
Random number generators
Private information retrieval
….

# Cloud computing attacker models

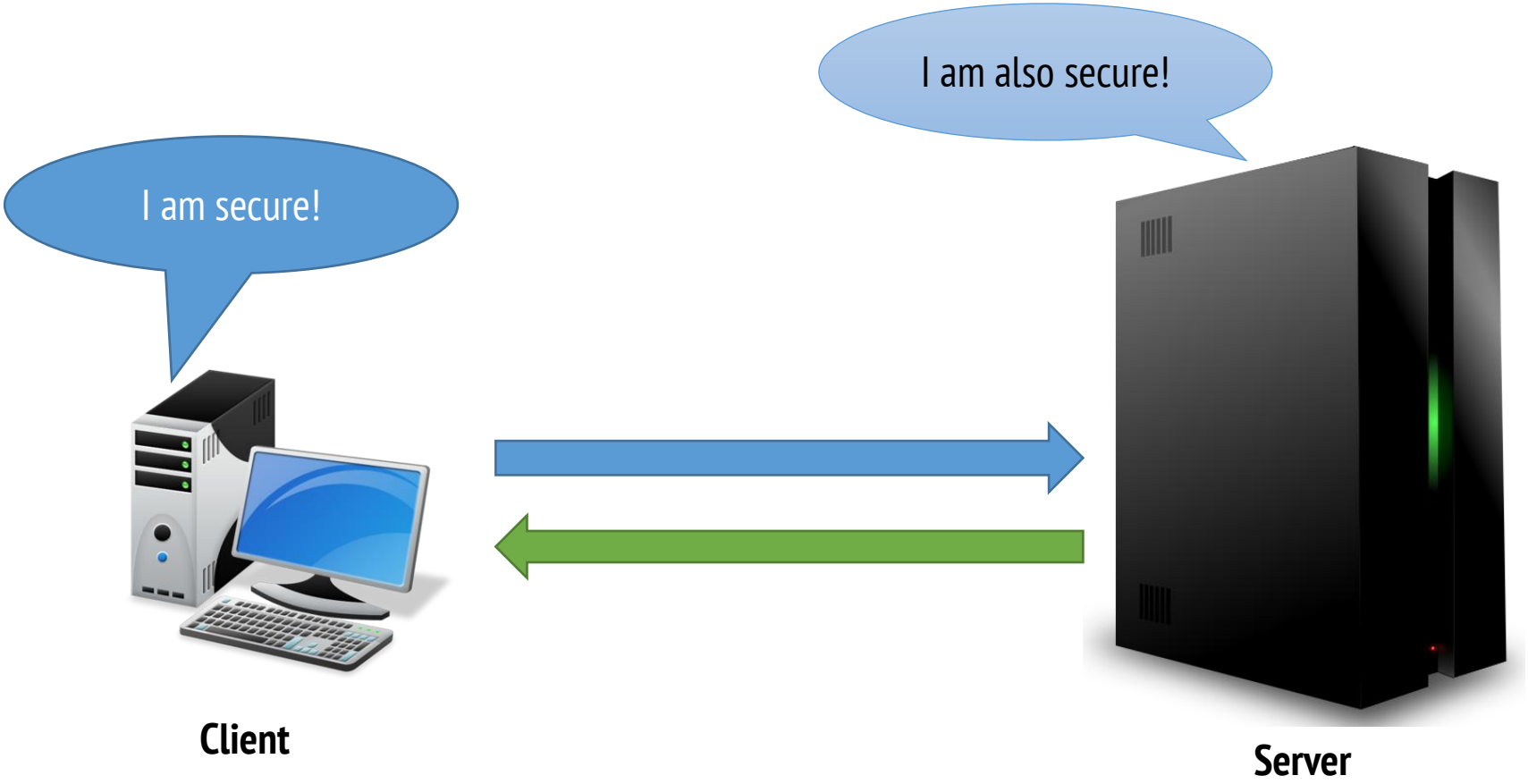
# Cloud computing attacker models

An **attacker model** is a description of capabilities specifying the kind of access an attacker has to a system when attempting to "break“ it.

# Attacker models



# Attacker models



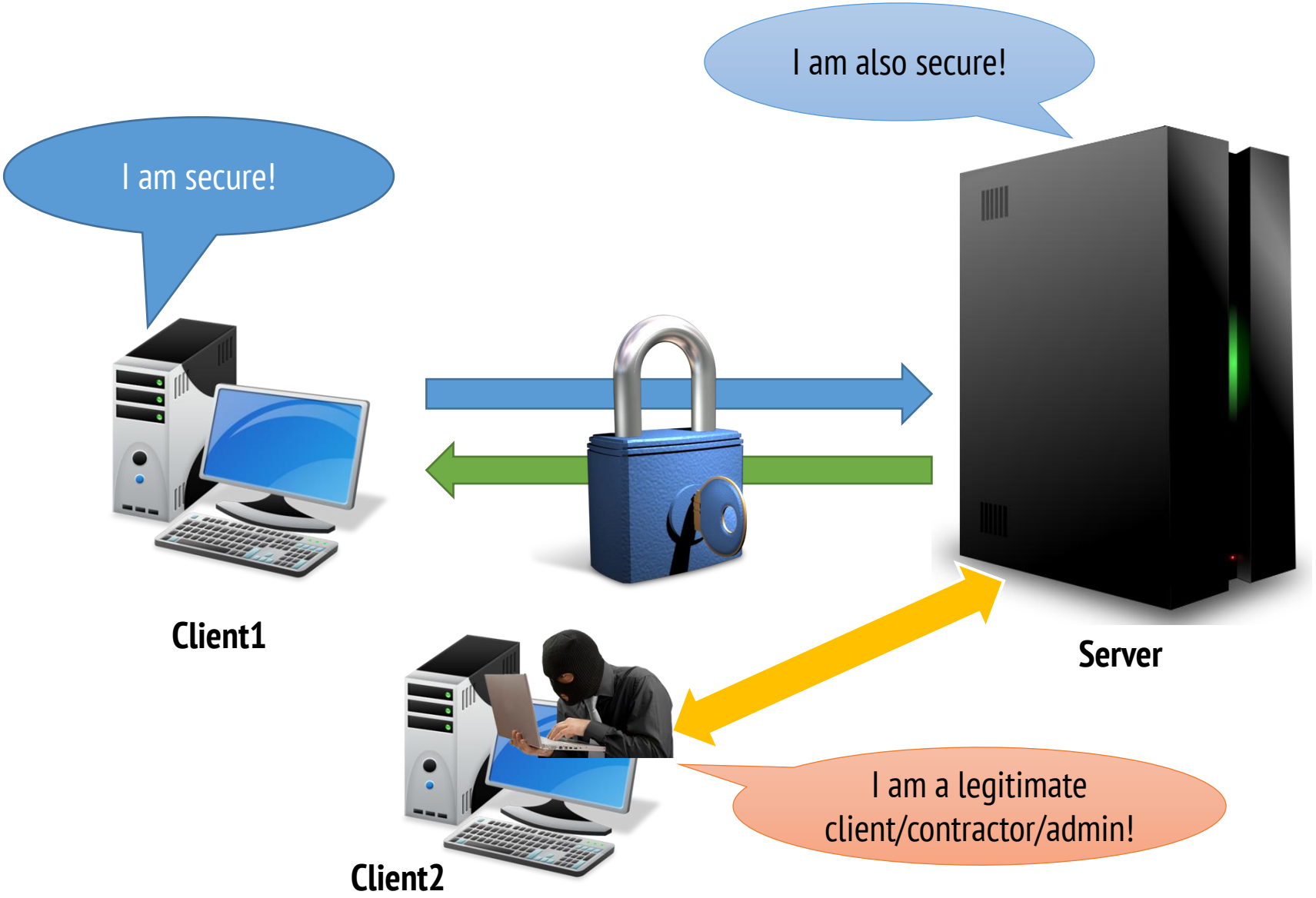
# Attacker models



# Attacker models



# Attacker models



# Cloud computing attacker models

Adversarial models

1. Honest
2. Honest-but-curious
3. Malicious-but-cautious
4. Malicious

# Honest-but-curious attacker model

Most common used attacker model in cloud computing

Assumes:

- The adversary will obey the protocol.
- Will try to access data if possible.
- It is a **passive** attacker.

# Malicious-but-cautious attacker model

Is a more powerful attacker than the honest but curious attacker.

Assumes:

- Does not necessarily try to obey the protocol, but will try to remain undetected.
- Attacker will tamper with data.
- Is cautious and tried not to leave verifiable evidence of its misbehaviour.
- Is an **active** and **covert** attacker.

# Malicious attacker model

The most powerful attacker. Sometimes (usually when referring to data in transit) is described as a Yolev-Dao model attacker model.

Assumes:

- Complete control over networking and resources.
- Attacker will go to any lengths to access, tamper or recover data.

# The “Confidentiality from the cloud provider” goal

# Confidentiality from the cloud provider Holy grail?

- Currently, the price of using the cloud is that you have to divulge your data to the cloud provider.
- The business model of many cloud providers, such as Google, is to exploit that data.
- Many designs for new architectures focus on the idea that it might be possible to get the benefits of cloud computing without paying that price.

# Is confidentiality from the cloud provider enough?

The malicious-but-cautious attacker model is appropriate for the cloud service provider:

- Malicious, because the CSP will cheat if it can
- Cautious, because it wants to keep you as a customer

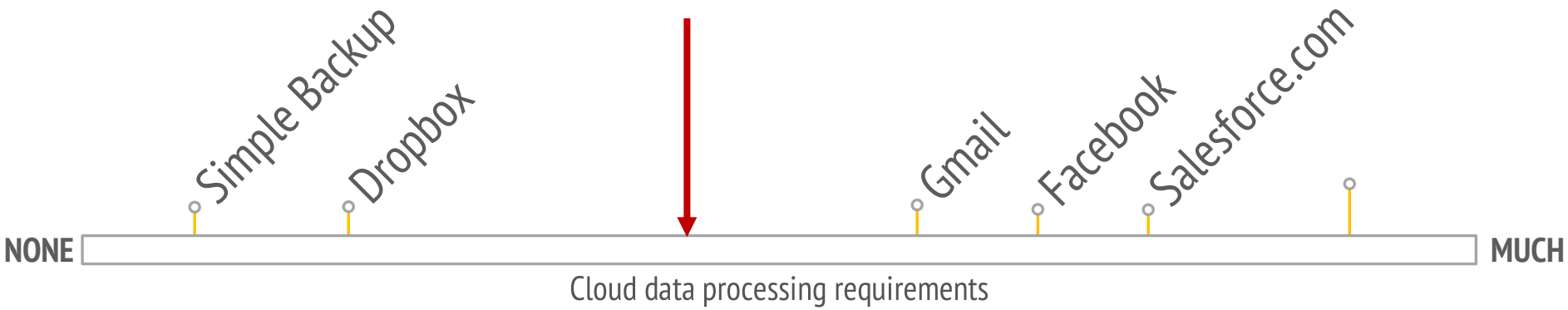
# What issues are addressed by “confidentiality-from-the-cloud-provider”?

1. Data breaches
2. Insufficient identity, credential and access management
3. Insecure interfaces and APIs
4. System vulnerabilities
5. Account hijacking
6. Malicious insiders
7. Advanced Persistent Threats
8. Data loss
9. Insufficient due diligence
10. Denial of service
11. Shared technology issues

# Confidentiality from the cloud provider

- Avoiding having to trust cloud provider means:
  - You don't have to trust its employees or subcontractors
  - You don't lose data confidentiality if cloud provider gets hacked

# Cloud computation requirements

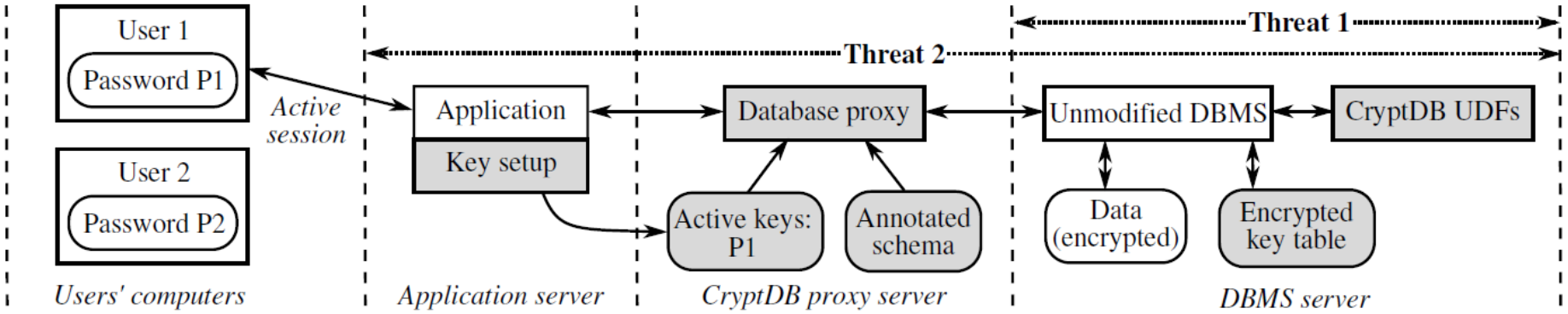


# CryptDB

## A secure database system

R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: protecting confidentiality with encrypted query processing," in Proceedings of the 23rd ACM Symposium on Operating Systems Principles, 2011.

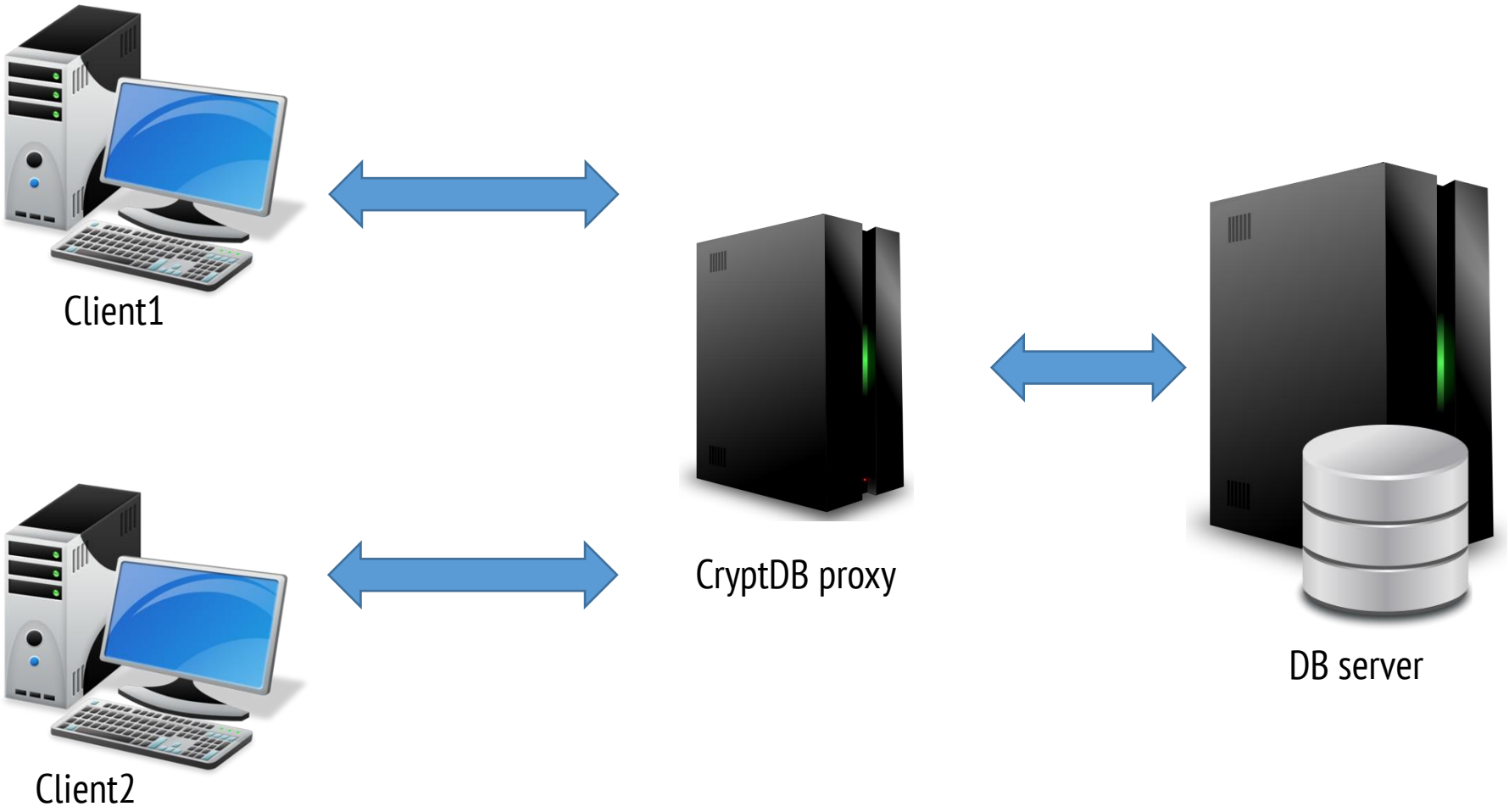
# CryptDB: A secure database system



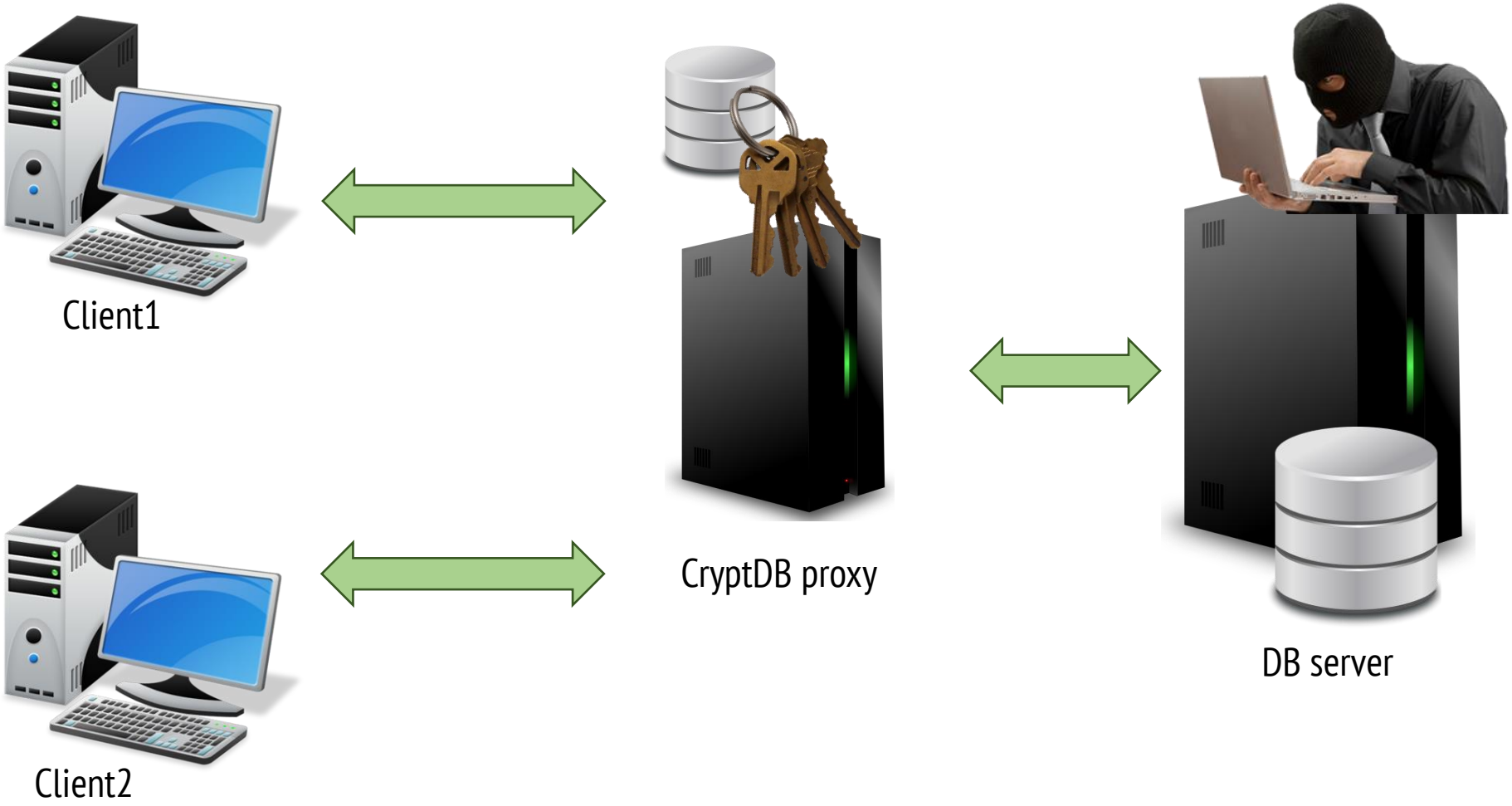
# CryptDB: A secure database system

- CryptDB
  - Is a secure middleware that protects databases deployed on cloud servers
  - Requires a proxy server that is placed in a trusted zone to handle cryptographic key translation
  - Stored data is encrypted in multiple formats to permit database operations
  - Can dynamically adjust data encryption to accommodate for new database operations
  - Runs on unmodified database servers
- Threats
  - Threat 1 – Curious administrator with full access to the DB server
  - Threat 2 – If CryptDB proxy is compromised the adversary cannot obtain data belonging to users that are not logged in

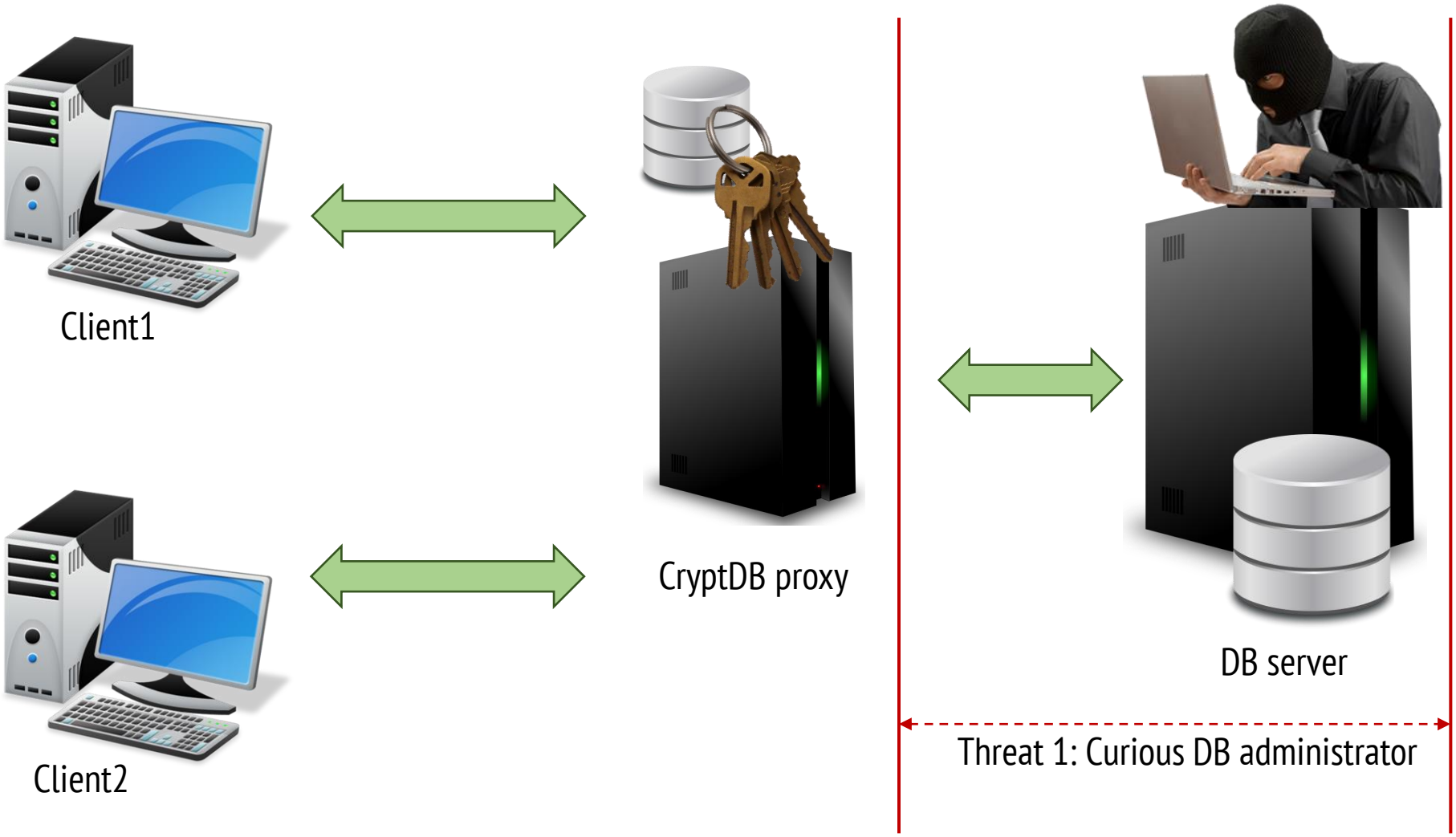
# CryptDB



# CryptDB



# CryptDB


Client1
Client2
CryptDB proxy
DB server
Threat 1: Curious DB administrator

# CryptDB: DB functionality

Naïve approach:

- Store keys in the CryptDB proxy.
- Encrypt database with keys.

Usage

- Download DB from server to CryptDB proxy
- Decrypt
- Perform DB operation (e.g. sort)
- Send result to user

# CryptDB: DB functionality

Naïve approach:

- Store keys in the CryptDB proxy.
- Encrypt whole database with one key.

Usage

- Download DB from server to CryptDB proxy
- Decrypt
- Perform DB operation (e.g. sort)
- Send result to user

**NOT EFFICIENT !**

# CryptDB: DB functionality

A more efficient approach?

- Store keys in the CryptDB proxy.
- **Encrypt each record of the DB with a key.**

Usage

- Download DB from server to CryptDB proxy
- Decrypt
- Perform DB operation (e.g. sort)
- Send result to user

# CryptDB: DB functionality

A more efficient approach?

- Store keys in the CryptDB proxy.
- **Encrypt each record of the DB with a key.**

Usage

- Download DB from server to CryptDB proxy
- Decrypt
- Perform DB operation (e.g. sort)
- Send result to user

- **EQUALLY NOT EFFICIENT !**

# CryptDB: DB functionality

CryptDB approach:

- Store keys in the CryptDB proxy.
- **Encrypt each record of the DB with a key.**
- **Use metadata information to specify how different DB components are to be used.**
- **Use different encryption methods to "augment" the usage.**

# Threat 1: Curious DB administrator

Crypto that enables functionality:

- Hide everything (RND): encrypt using non-deterministic encryption
- Equality comparison (DET): encrypt using deterministic encryption
- Sort functionality (order-preserving encryption OPE): make sure that if **x<y** then **enc(x)<enc(y)**.
- Join functionality (**equality-join** and **range-join):** use same key for multiple DB columns.

# Threat 1: Curious DB administrator

***Employees***

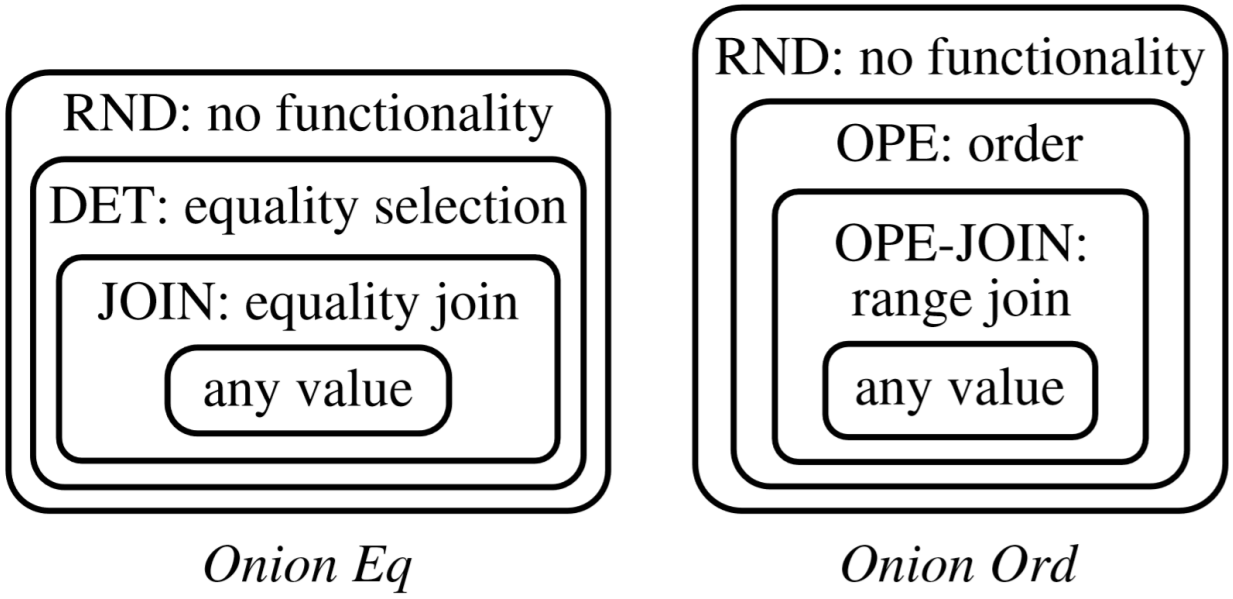
| *ID* | *Name* |
|---|---|
| 23 | Alice |




***Table1***

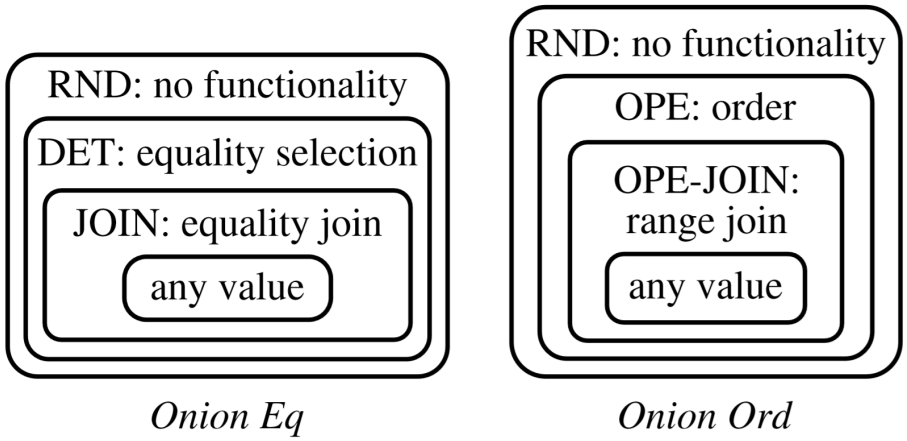
| *C1-IV* | *C1-Eq* | *C1-Ord* | *C1-Add* | *C2-IV* | *C2-Eq* | *C2-Ord* | *C2-Search* |
|---|---|---|---|---|---|---|---|
| x27c3 | x2b82 | xcb94 | xc2e4 | x8a13 | xd1e3 | x7eb1 | x29b0 |

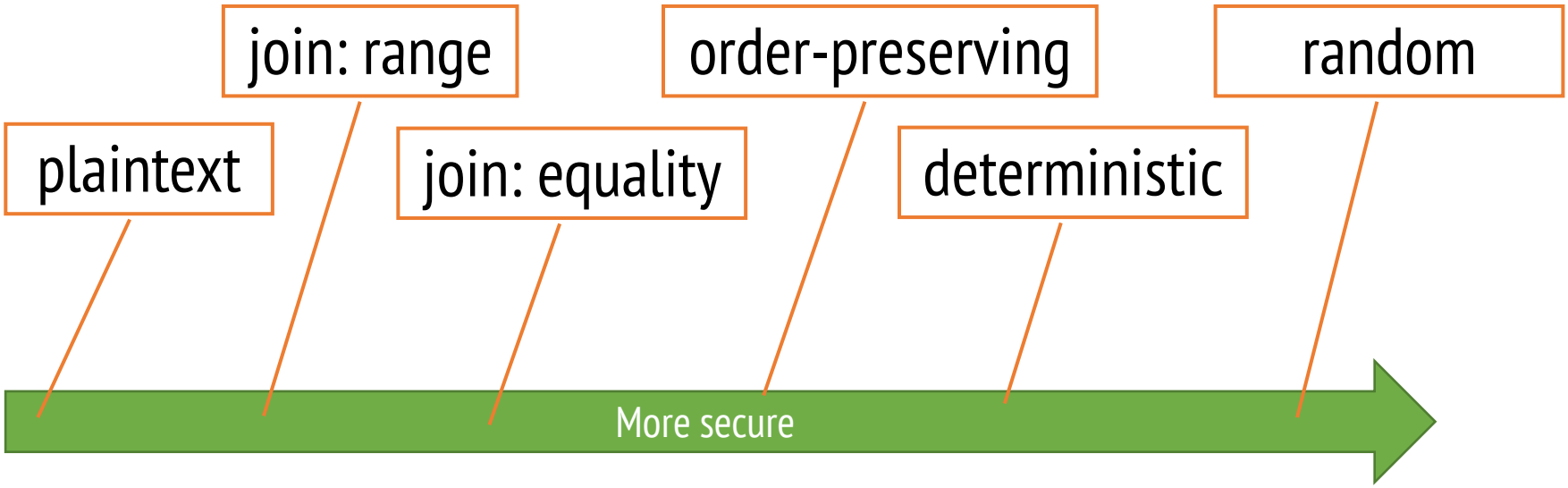
# CryptDB: DB functionality




Encrypted DB record in CryptDB

# Issues


RND: no functionality
DET: equality selection
JOIN: equality join
any value
Onion Eq
RND: no functionality
OPE: order
OPE-JOIN: range join
any value
Onion Ord

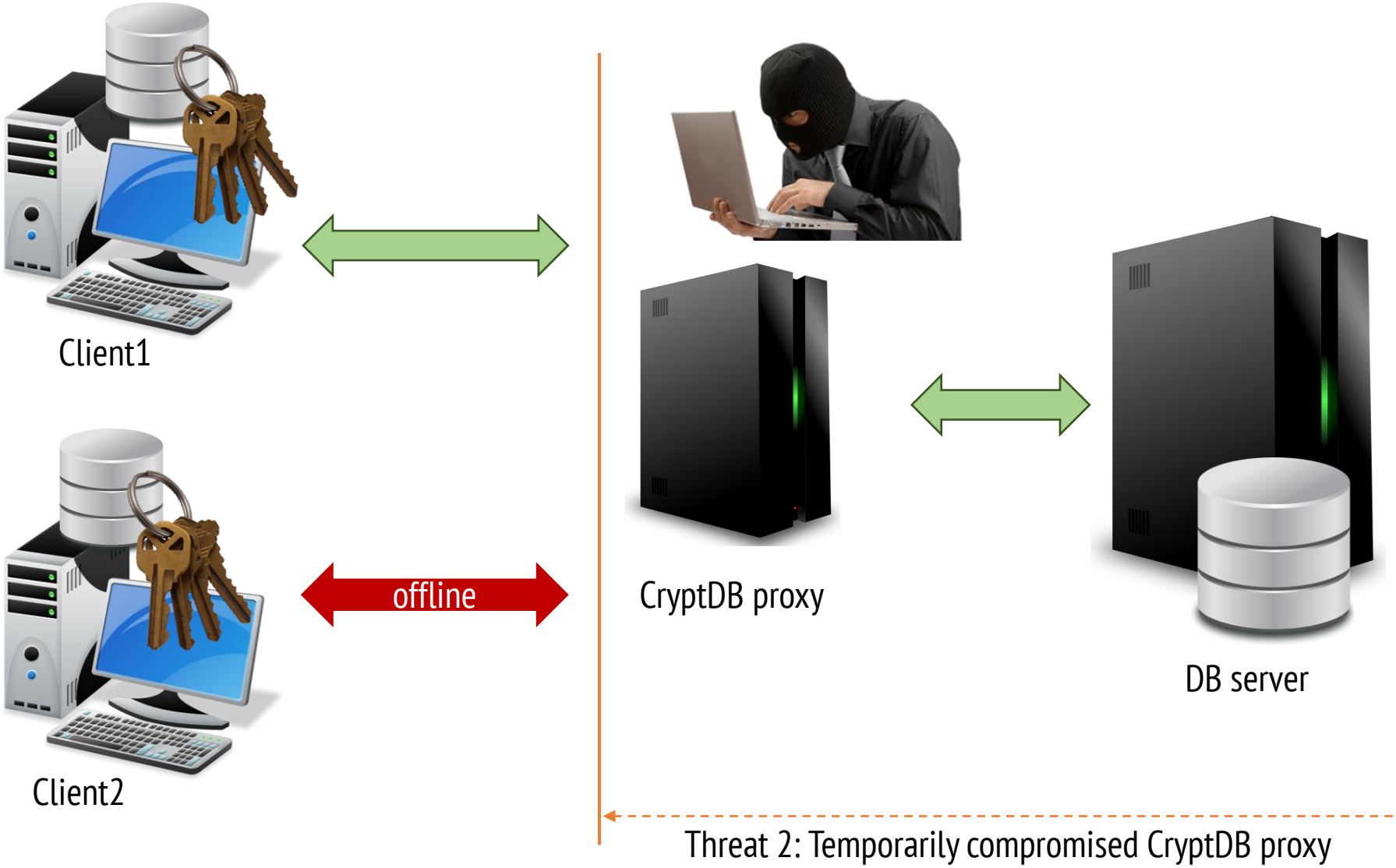

plaintext
join: range
join: equality
order-preserving
deterministic
random
More secure

# Issues

- Not all records are equally secure.
- Over time all records will use default to using the weakest security allowed.

- **FAIL TO SAFE is not respected.**

# Threat 2: Tmp. compromised CryptDB proxy

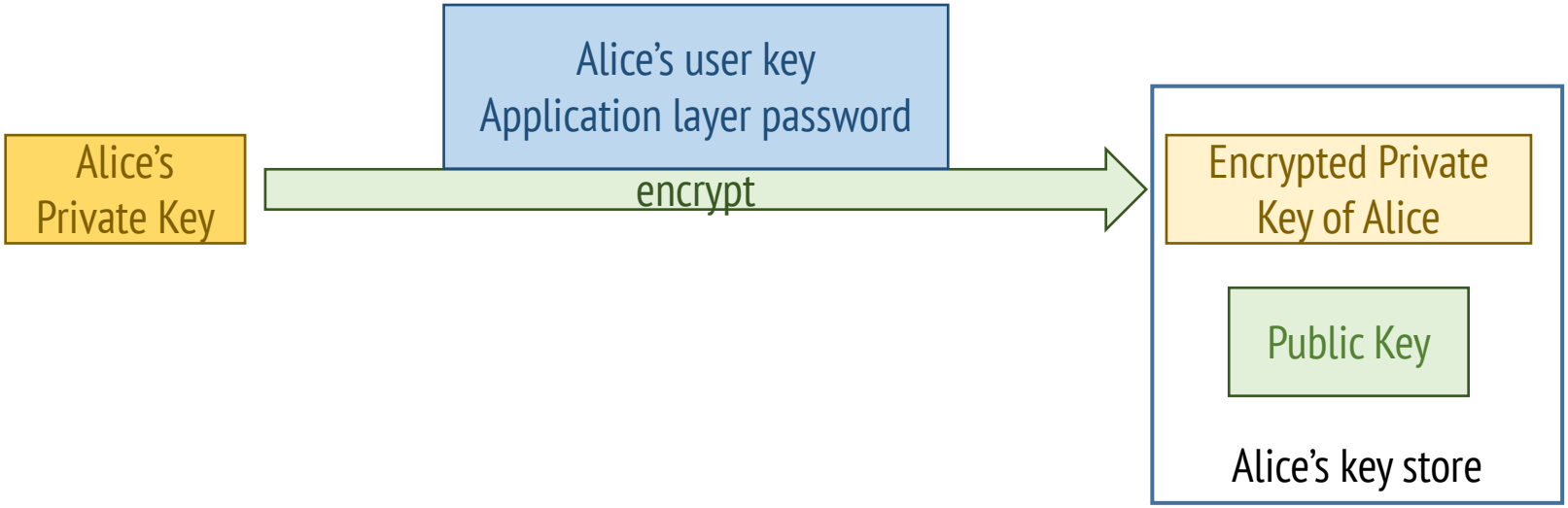


# Key Chaining

- CryptDB proxy must be able to perform operations on behalf of the user
- User enables the CryptDB proxy to access its data
- CryptDB proxy is trusted to discard key after the operation is successfully completed

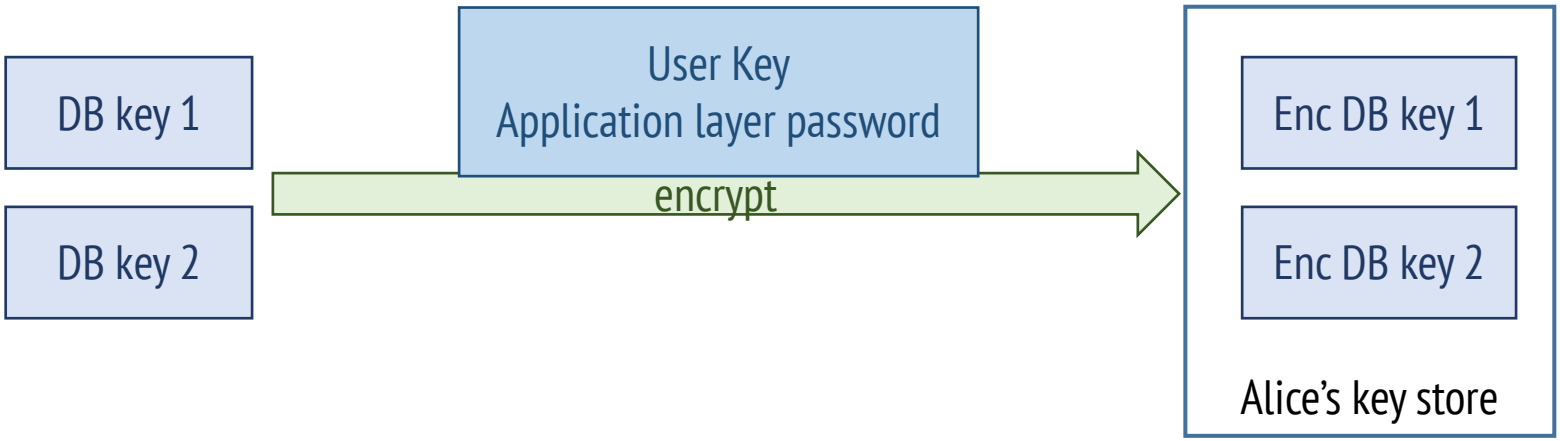
# Key Chaining

## User keys derivation



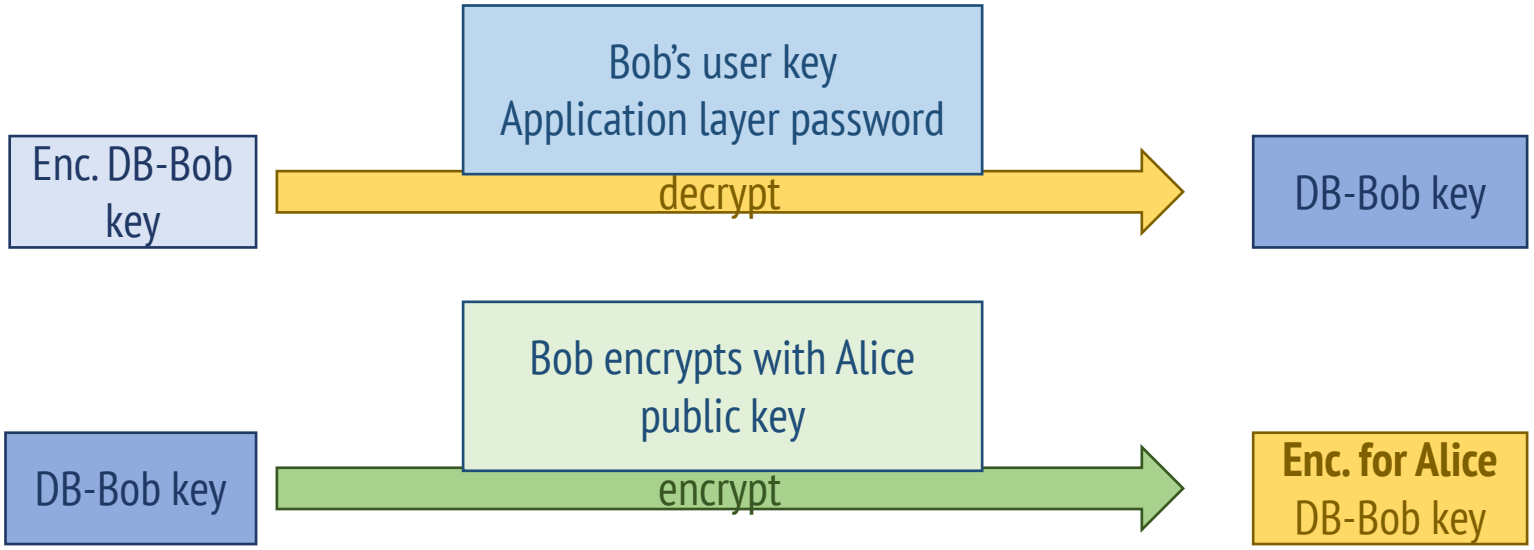
# Key Chaining

DB access



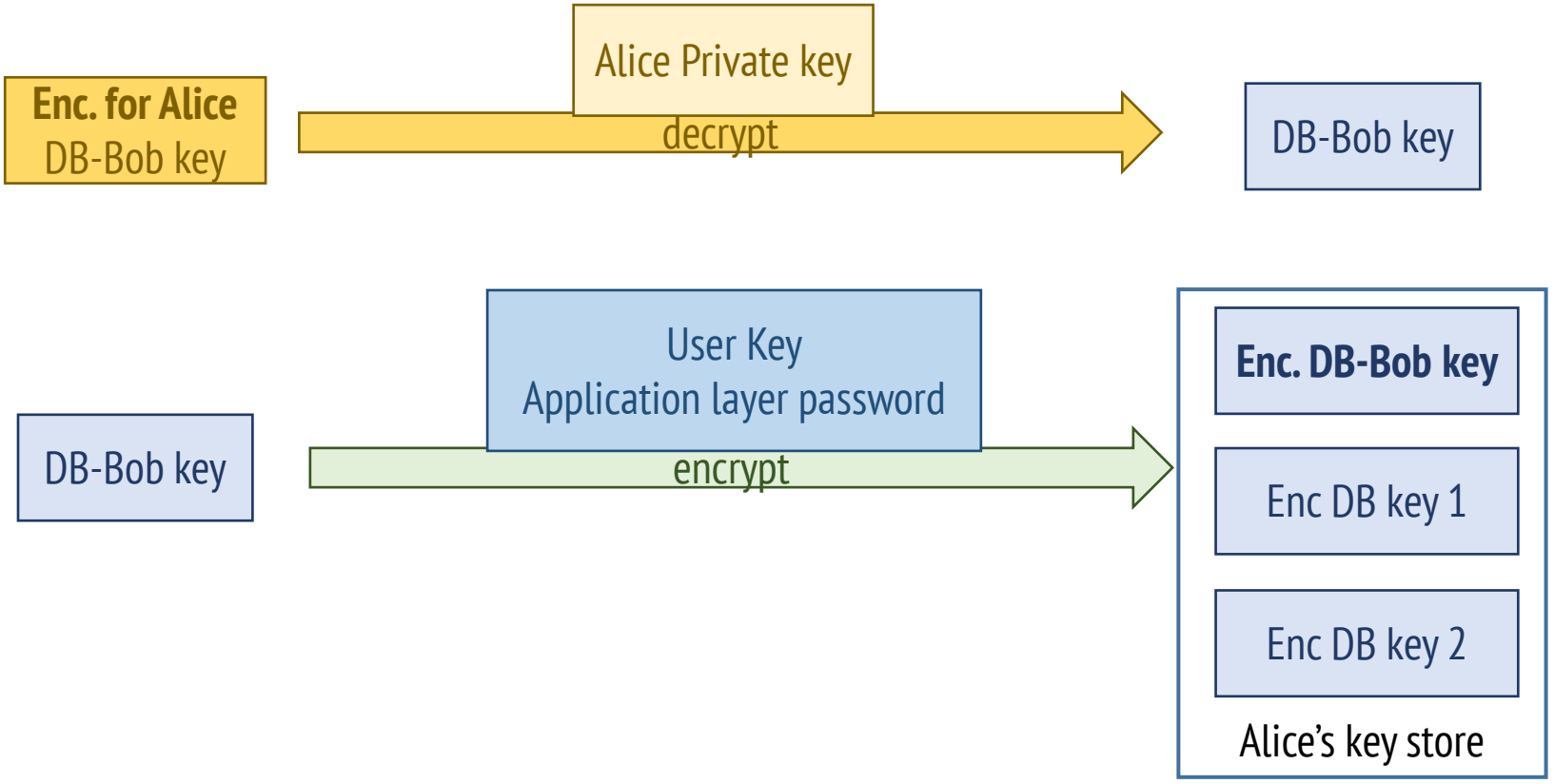
# Key Chaining

Bob wants to share DB-Bob with Alice

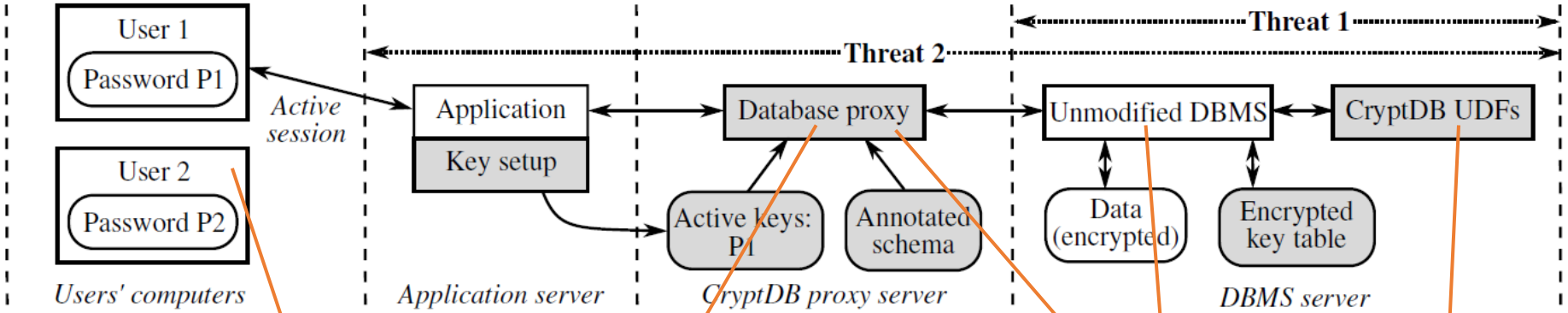


# Key Chaining

Alice adds key to her store



# CryptDB: A secure database system




The use of CryptDB does not affect users and application developers

CryptDB proxy:

- Intermediates requests between applications and the standard DB server
- Is placed in a trusted (threat 1) semi-trusted (threat2) zone
- Threat 1: Stores encryption keys to the DBMS
- Threat 2: Stores encryption keys that are chained to user passwords

Adapts regular SQL queries to encrypted SQL queries compatible with standard DB servers.

Standard database server e.g. MySQL

User defined functions for special crypto operations

# ConfiChair

An application management system

# Conference system functionality

1. Conference chair sets up her conference
2. Chair "opens" the conference for submissions
3. Authors submit papers to the conference
4. Chair "closes" the conference for submissions
5. Chair assigns (say) 3 PC members (aka reviewers) to each paper, and opens the "reviewing phase"
6. PC members login, see their assignments, download the papers, write their reviews
7. Once all the reviewers have reviewed a paper, the discussion phase begins for the paper. Reviewers read each other's reviews and comment on them, aiming to reach accept/reject consensus.
8. Chair takes the accept/reject decision, usually following the recommendation of the reviewers.

# Why do it in the cloud?

Advantages

- Cloud provider takes responsibility for all aspects of organising the service: storage, software, backup, availability, security, version management, ...
- And does it all for free!

- Problem
  - Cloud provider has full unmediated access to all the data.

# EasyChair

- A **SaaS** system consisting of authors, reviewers, & chairs




"We believe that since 2006 we have become number one conference management system in the number of conferences, users and submissions. All together EasyChair proudly hosted 49,000 conferences and 1M users."

# The confidentiality problem

- EasyChair managers have direct access to **the submission and reviewing profiles** of 1M users across 49k conferences, including submission rate, acceptance rate, reviewer profile (fair/unfair, thorough/scant, prompt/late).
- EasyChair could [in principle, if it wished] **offer profiling services to appointment panels, awarding bodies, recruitment agencies.**
- The data could become a **target for hackers/crackers**.
- Similar situation for other SaaS systems.

# ConfiChair: An application management system

- A secure application management system
  - Cryptographically protects applications and evaluations from the cloud provider
  - Uses data re-encryption to enable the administrative role user to provide differentiated access control
  - Uses an encrypted Key Purse to manage keys
  - Uses Stanford JavaScript Crypto Library (SJCL) and HTML5 to provide in-browser encryption
- Security properties
  - Secrecy properties – ConfiChair doesn't know the content of applications, evaluations, rankings, discussions, notifications.
  - Unlinkability property - ConfiChair does not know that a particular evaluator R evaluated an application submitted by a particular author A.

M. Arapinis, S. Bursuc, and M. Ryan, "Privacy supporting cloud computing: ConfiChair, a case study," in Principles of Security and Trust, 2012, pp. 89–108.

# Security properties

## Secrecy property:

ConfiChair does not know the content of:

- Papers
- Reviews
- Scores
- Discussions
- Decisions

## Unlinkability property:

- ConfiChair does not know that a particular reviewer R reviewed a paper written by a particular author A.

# Functionality

- Initialisation
  - Setup the cryptographic keys for a new conference
- Registration
  - Authors register their papers
- Evaluation/Review
  - Reviewers are given access to author's papers and can write reviews.
- Discussion
  - Reviewers are allowed to communicate with one-another for the purposes of agreeing on the features/short comes of the paper
- Notification
  - Authors are notified of their success

# Functionality

There are 3 user roles:

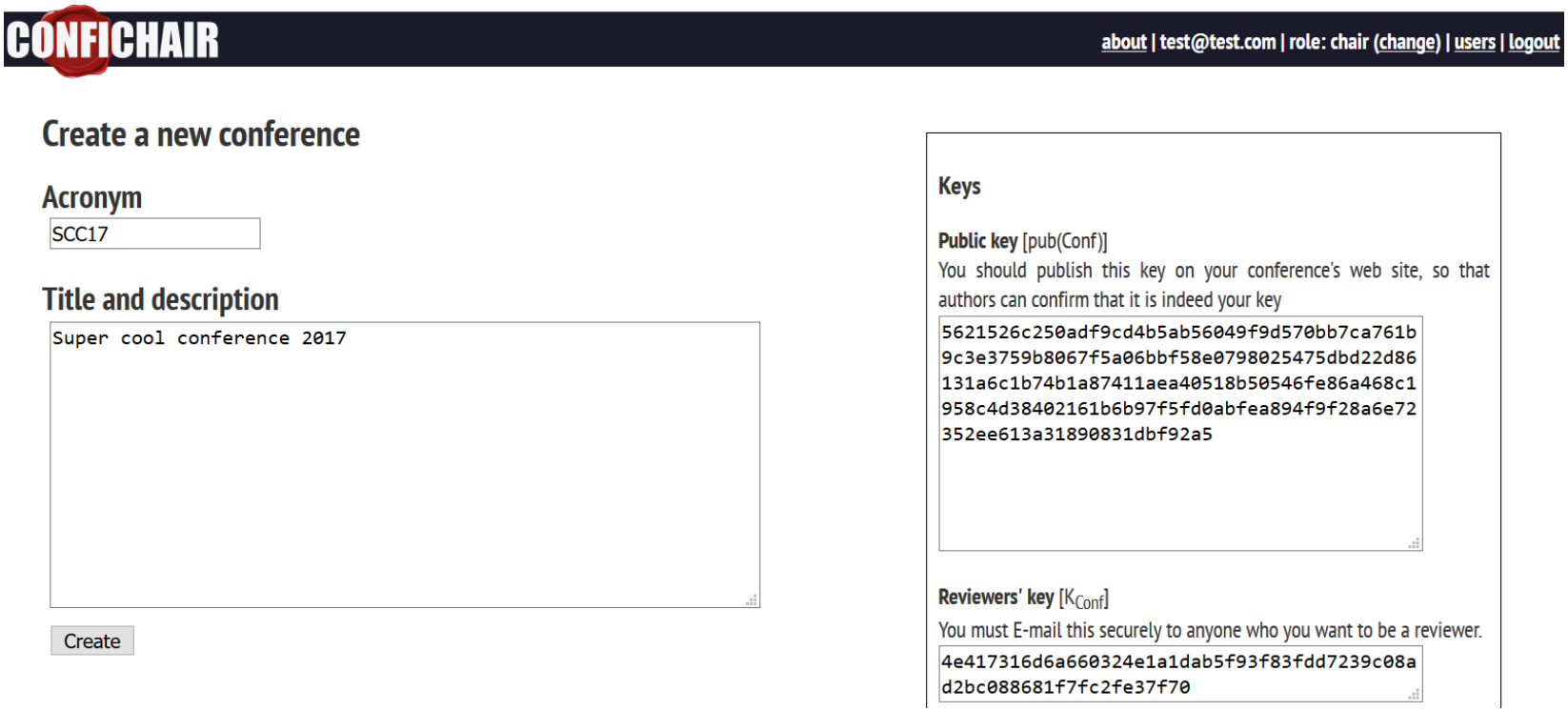
- Chair
- Author
- Reviewer

All users have a keypurse which stores their cryptographic material encrypted with a password derived key

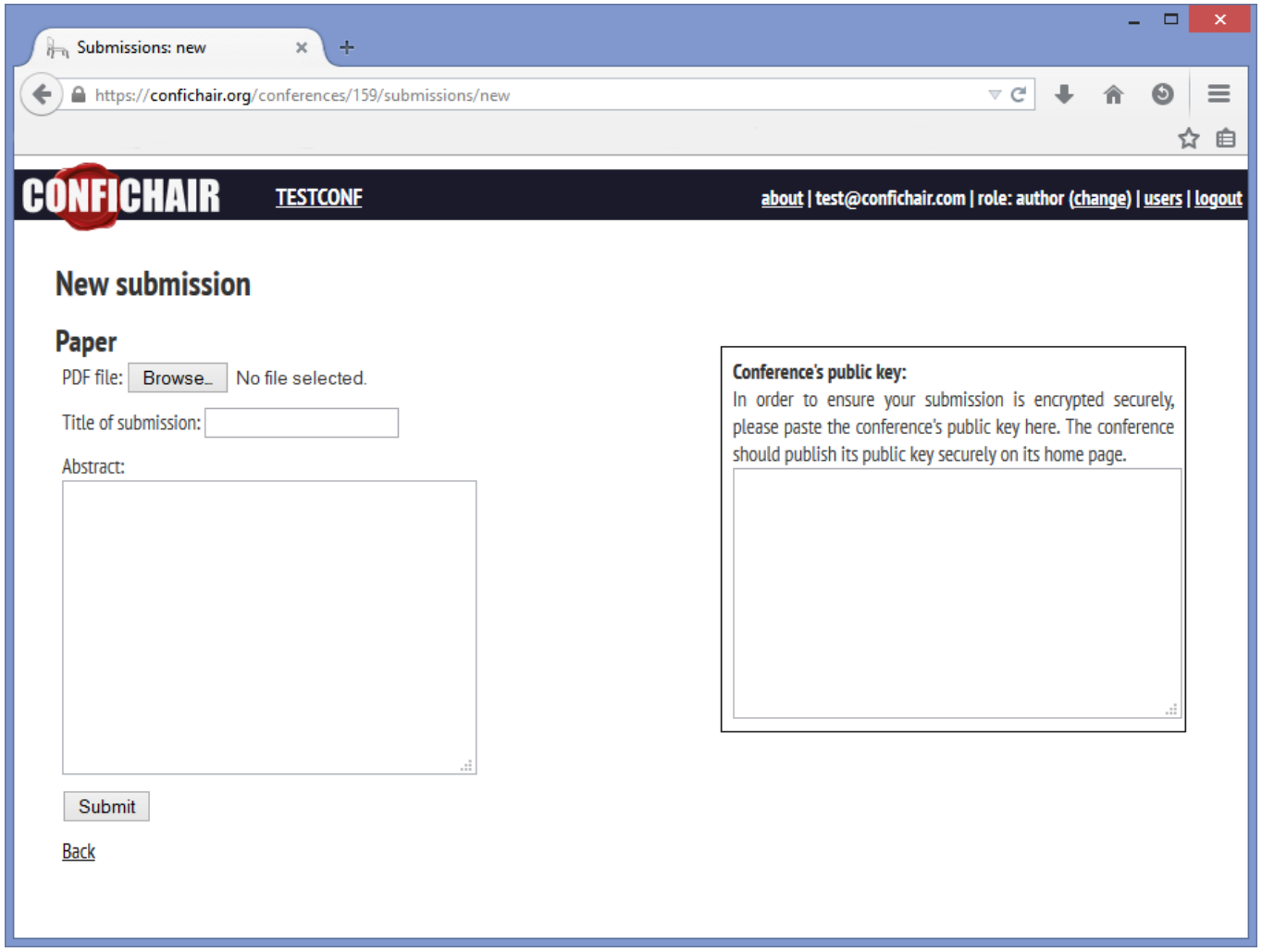
# Keypurse contents

- Chairs
  - Public/private conference key pair
  - A symmetric conference key
- Authors
  - Unique keys for each paper submitted
- Reviewers
  - Pre-shared symmetric conference keys
- Cloud
  - No un-encrypted cryptographic material

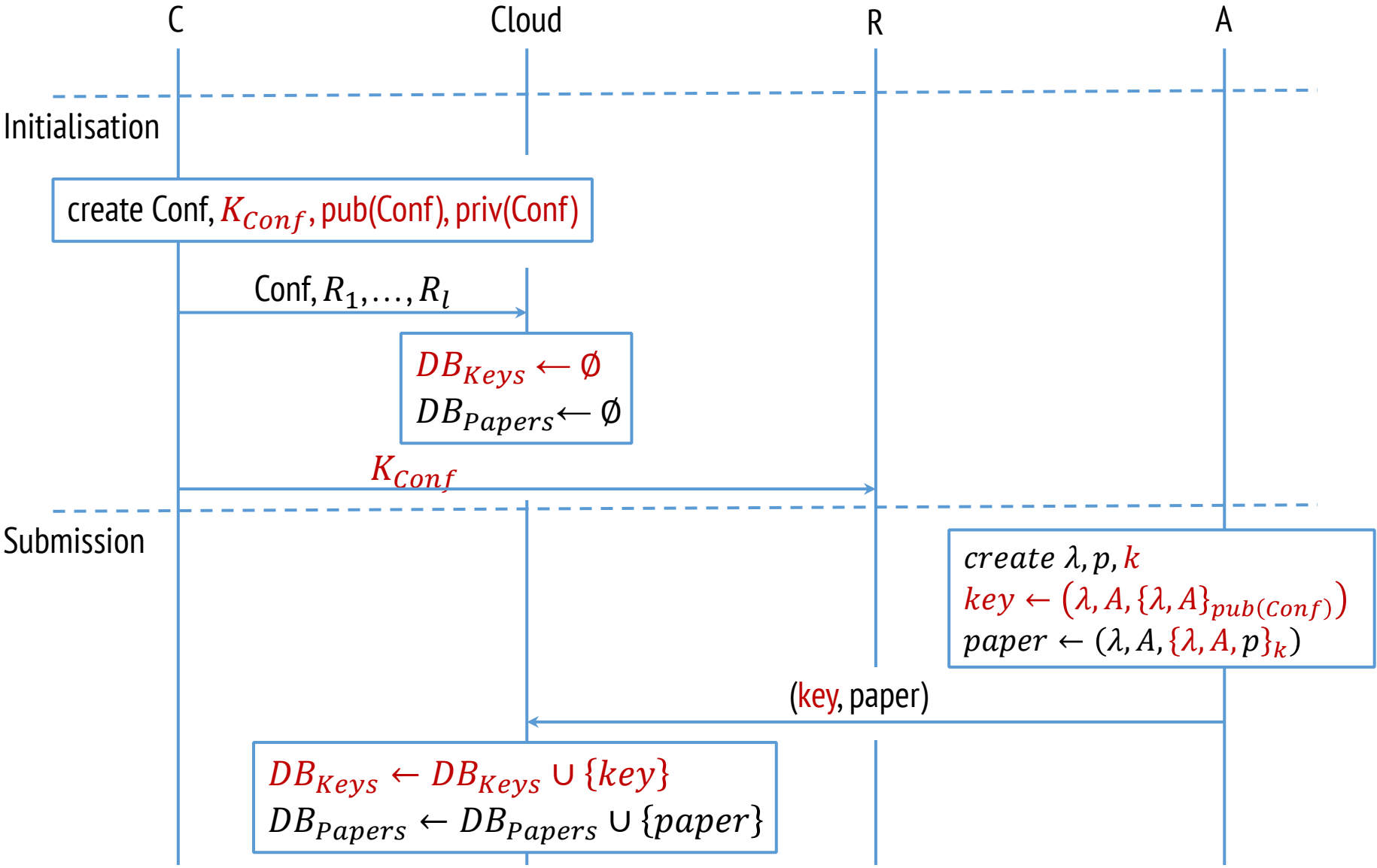
# ConfiChair: An application management system



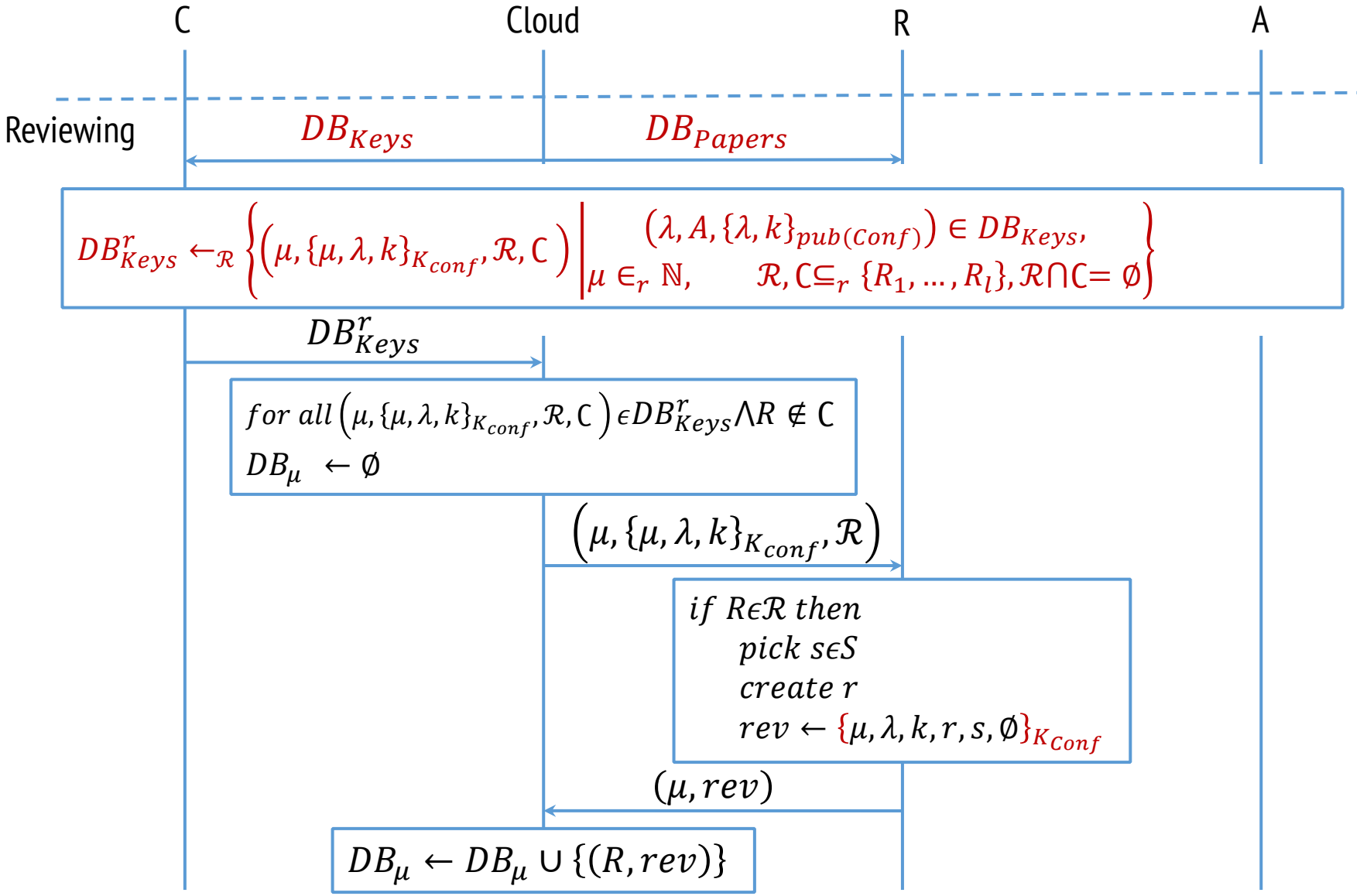
# ConfiChair: An application management system



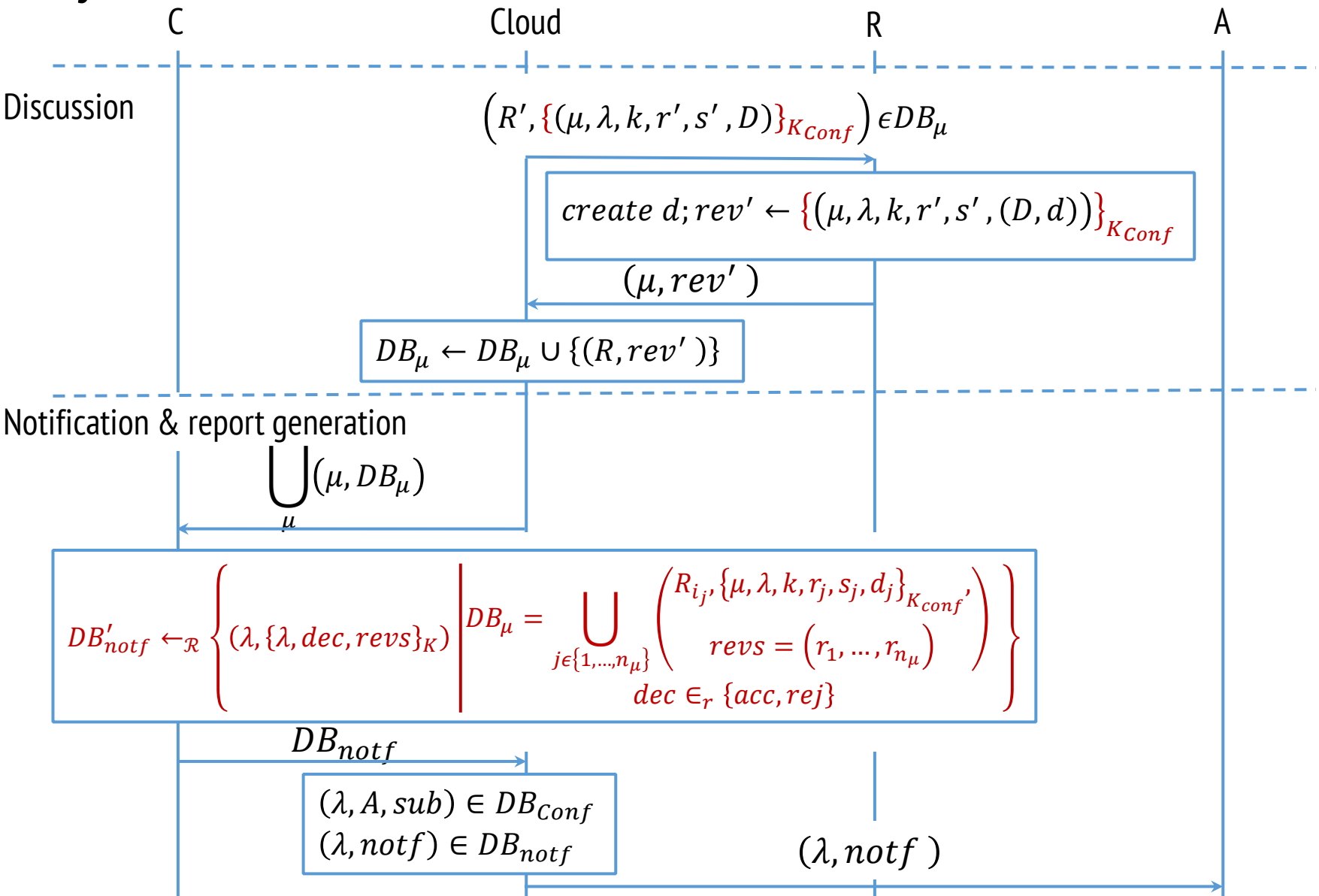
# ConfiChair: An application management system



# ConfiChair: An application management system



# ConfiChair: An application management system



# What is protected? What is the cost?

Not protected

- Accessed files
- Number of reviews
- “Intensity of the debate”

Usability

- Chair needs to re-encrypt all papers to allow reviewers to access them.
- Chair needs to re-encrypt all reviews to allow authors to see them

# But wait, you said unlinkability!

- “ConfiChair does not know that a particular reviewer R reviewed a paper written by a particular author A.”
- This property is difficult to achieve because the CSP can simply follow the ciphertexts (without needing to decrypt them):
  1. Author Alice uploaded an encrypted paper
  2. Reviewer Richard later downloaded the same encrypted paper
  3. Therefore, Richard reviewed Alice’s paper
- The way we achieved the desired unlinkability property is not very good. We simply coded it so that **papers are divided into bundles of 10**, and when a reviewer asks to download one paper, he actually downloads the group of 10 that it belongs to. **This is not really secure**. It does not really satisfy the unlikability property.

# Conclusions

- We need to understand the security challenges against cloud based applications.
- Cloud based applications should be studied with multiple attacker models in mind.
- There are security and financial benefits in designing applications  under the “confidentiality-from-the-cloud-provider” model.