

# Securing Secrets and Managing Trust in Modern Computing Applications

Andy Saylor

Dissertation Defense  
04/04/16



University of Colorado **Boulder**

# Proposal Review

How can we **secure**  
and **control** our data?

# Challenges to Privacy and Security (Chapter 3)

Third Parties?



Google

Google



**amazon**  
web services™



Google



**amazon**  
web services™

**facebook®**

How can we **secure**  
and **control** our data?

How can we **secure**  
and **control** our data?

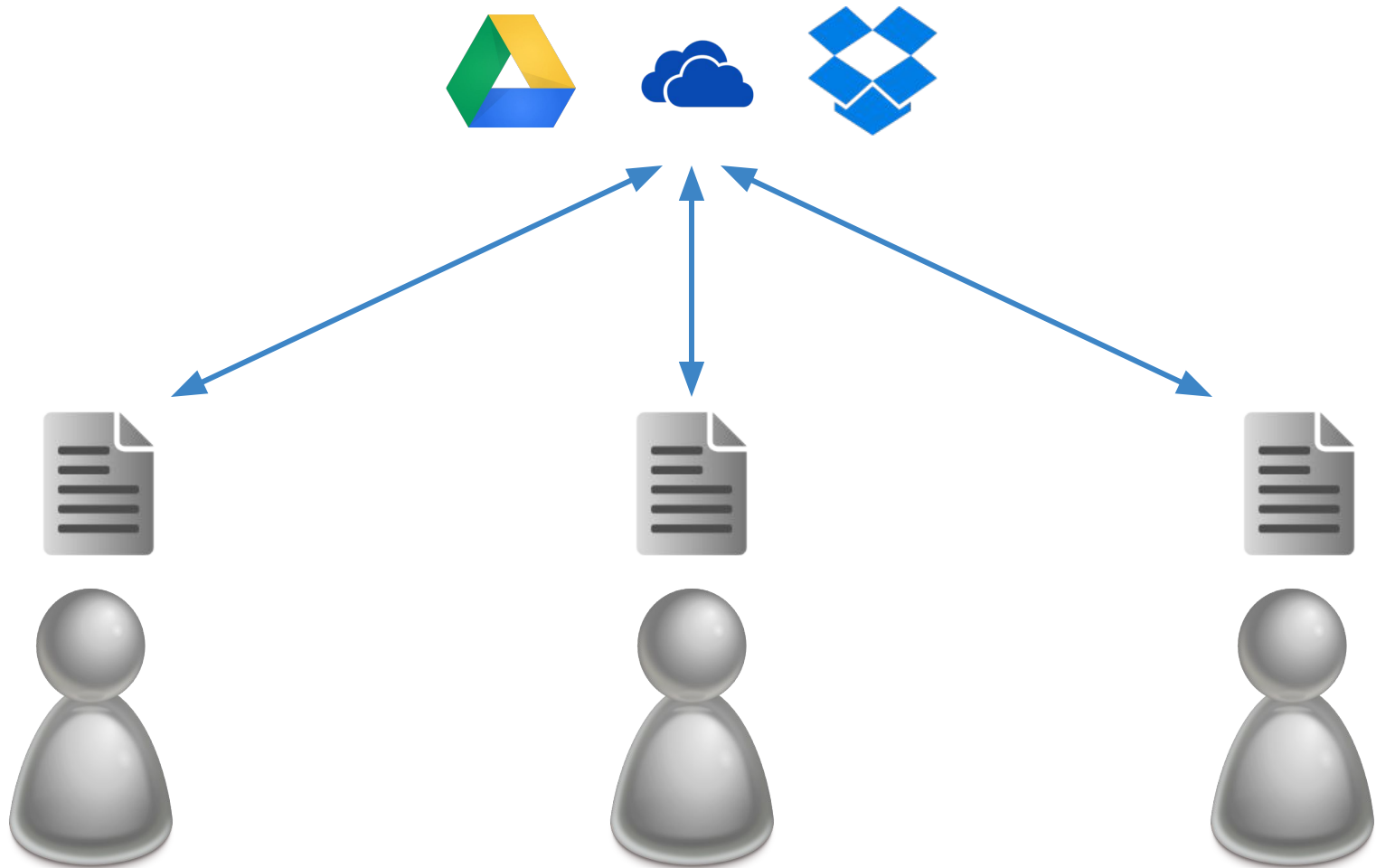
*(even in the presence **third parties**)*

# Modern Use Cases?

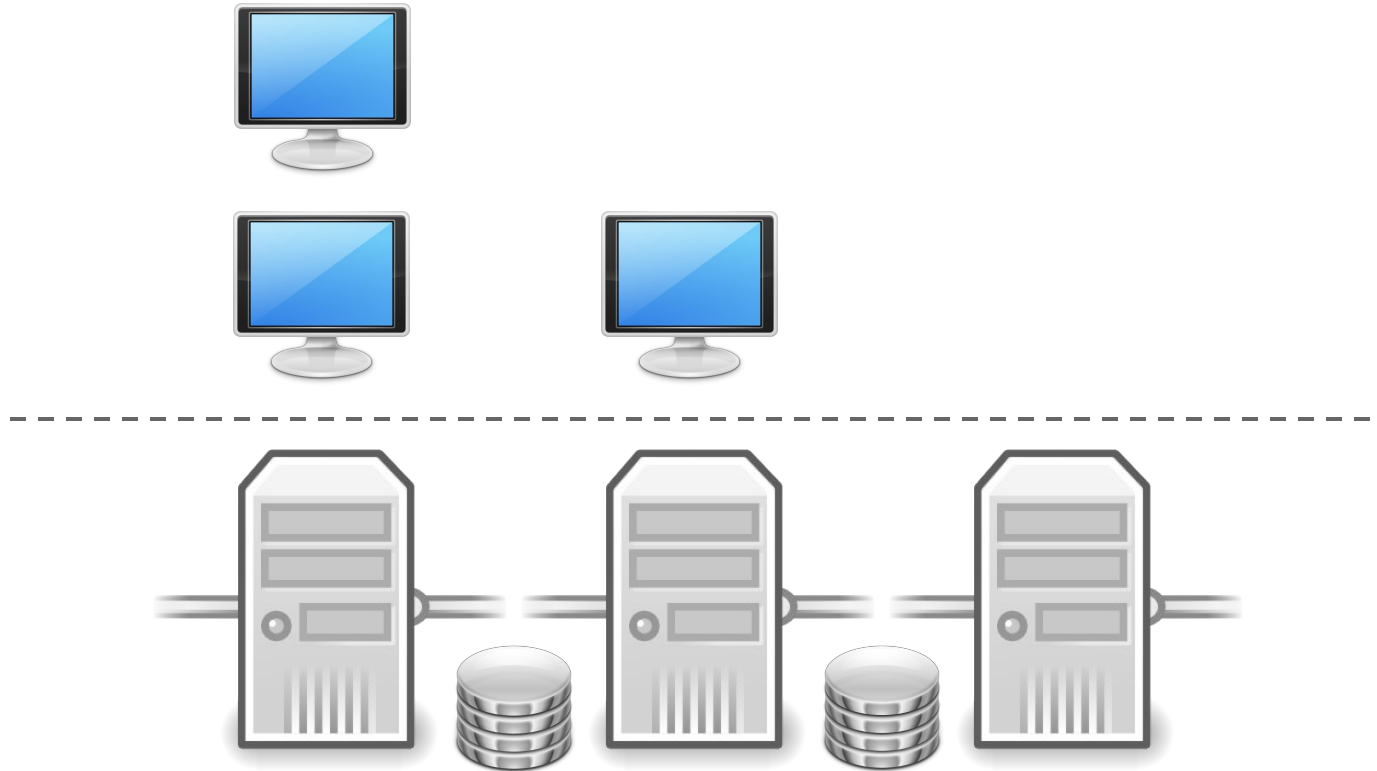
# Multi-Device File Access



# Multi-User File Sharing



# Cloud Infrastructure



How can we **secure**  
and **control** our data?

*(even in the presence **third parties**)*



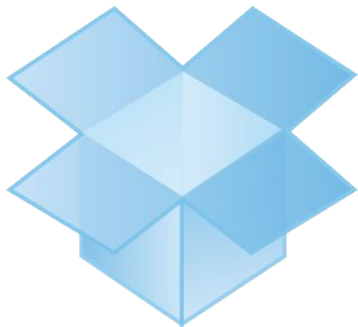
How can we **secure**  
and **control** our data?

*(even in the presence **third parties**)*

*(while also supporting modern **use cases**)*

Client-Side Encryption?

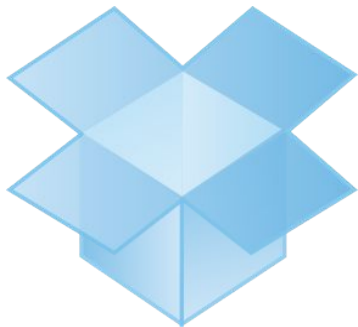
“My Data”

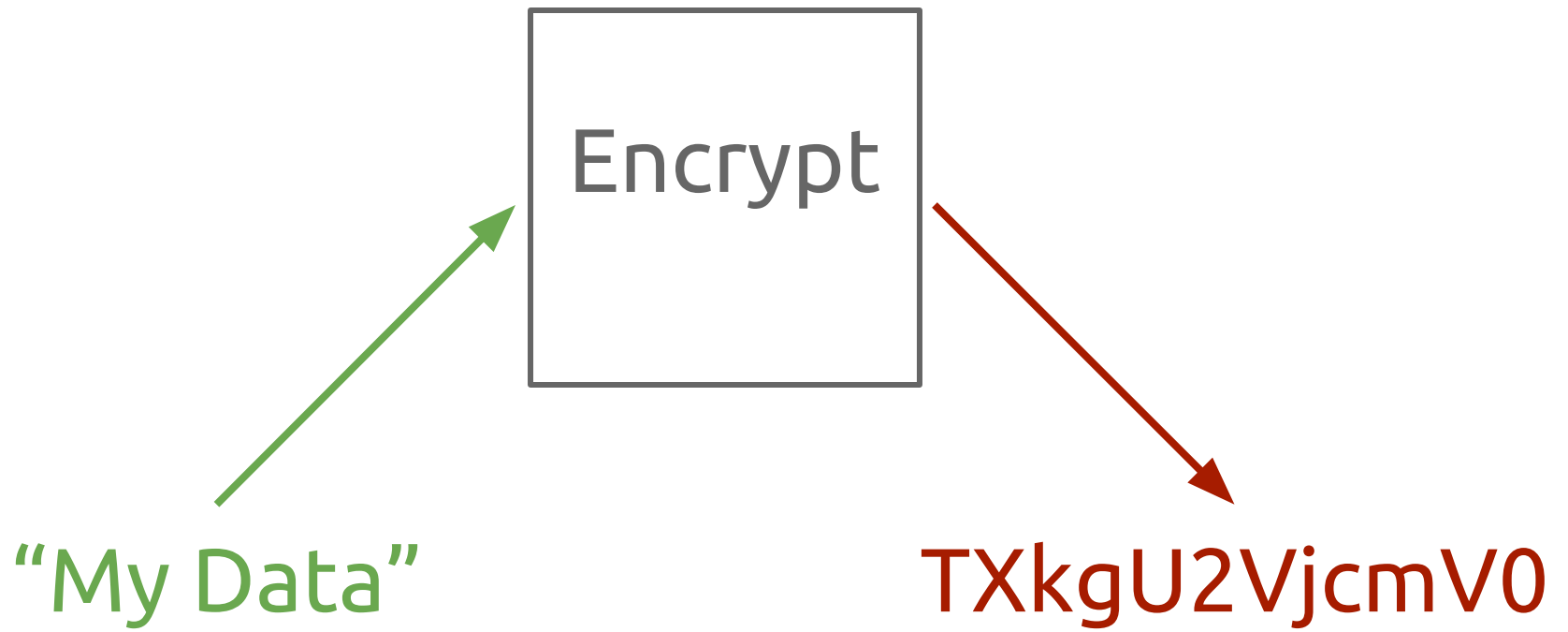


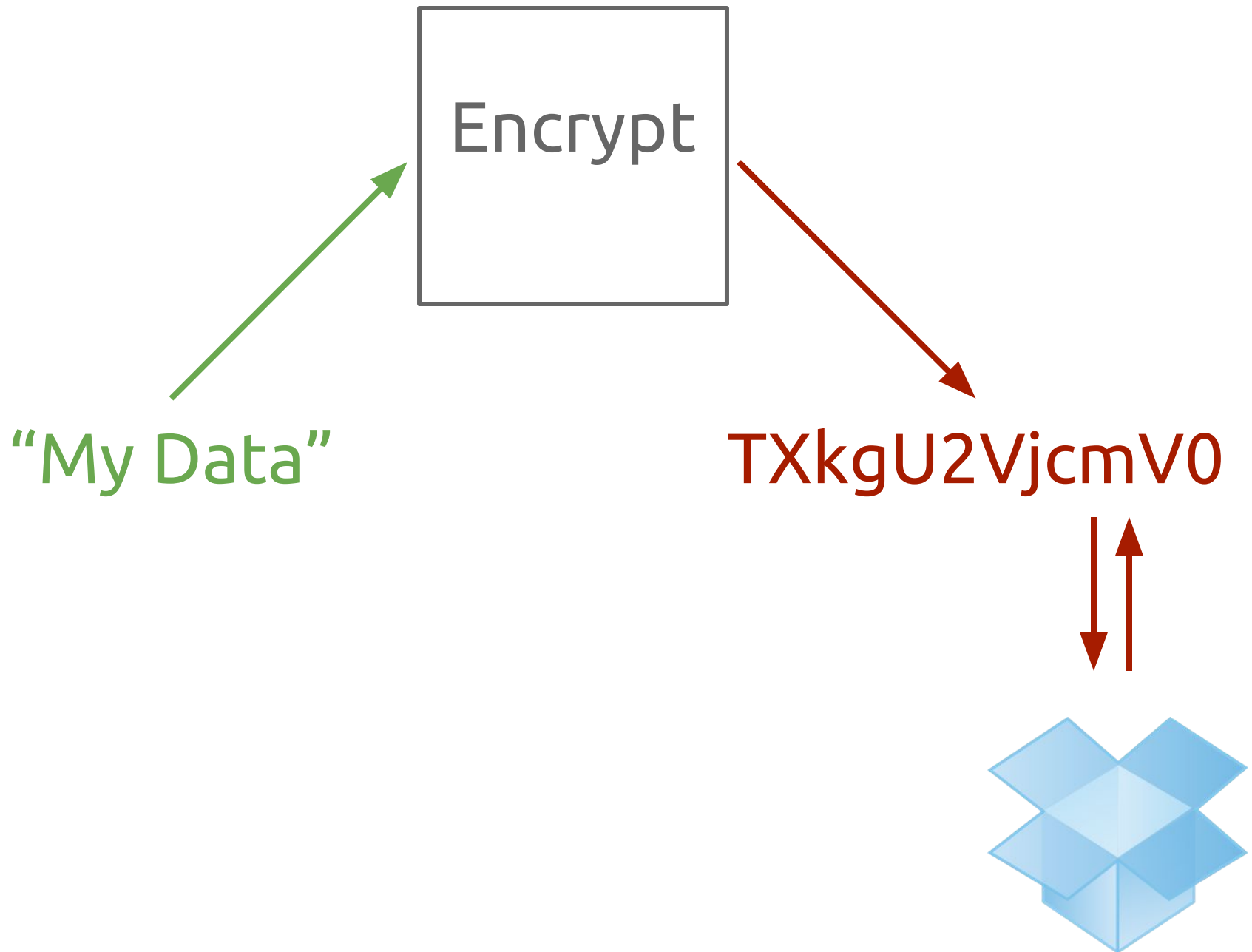


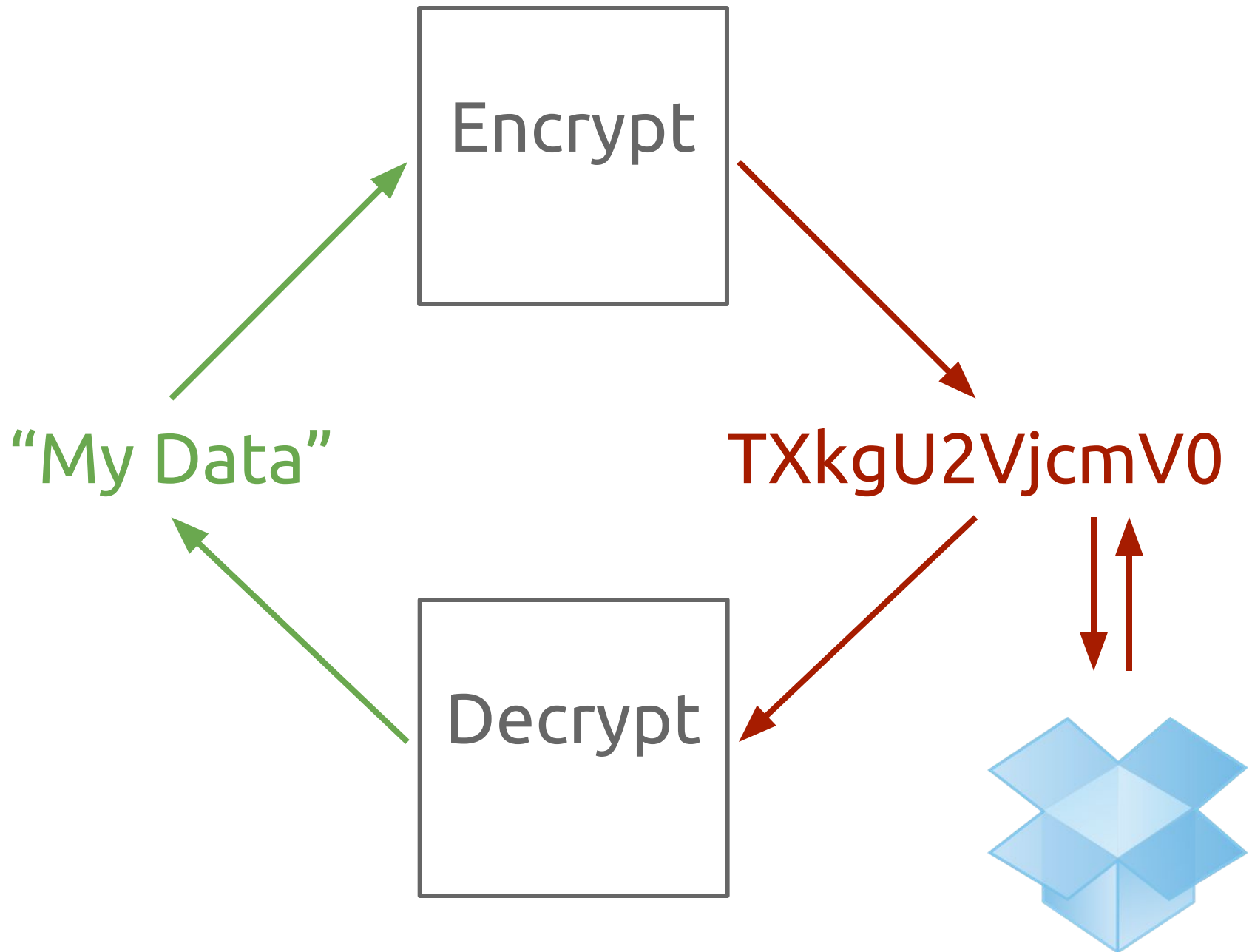
*Cryptography!*

“My Data”









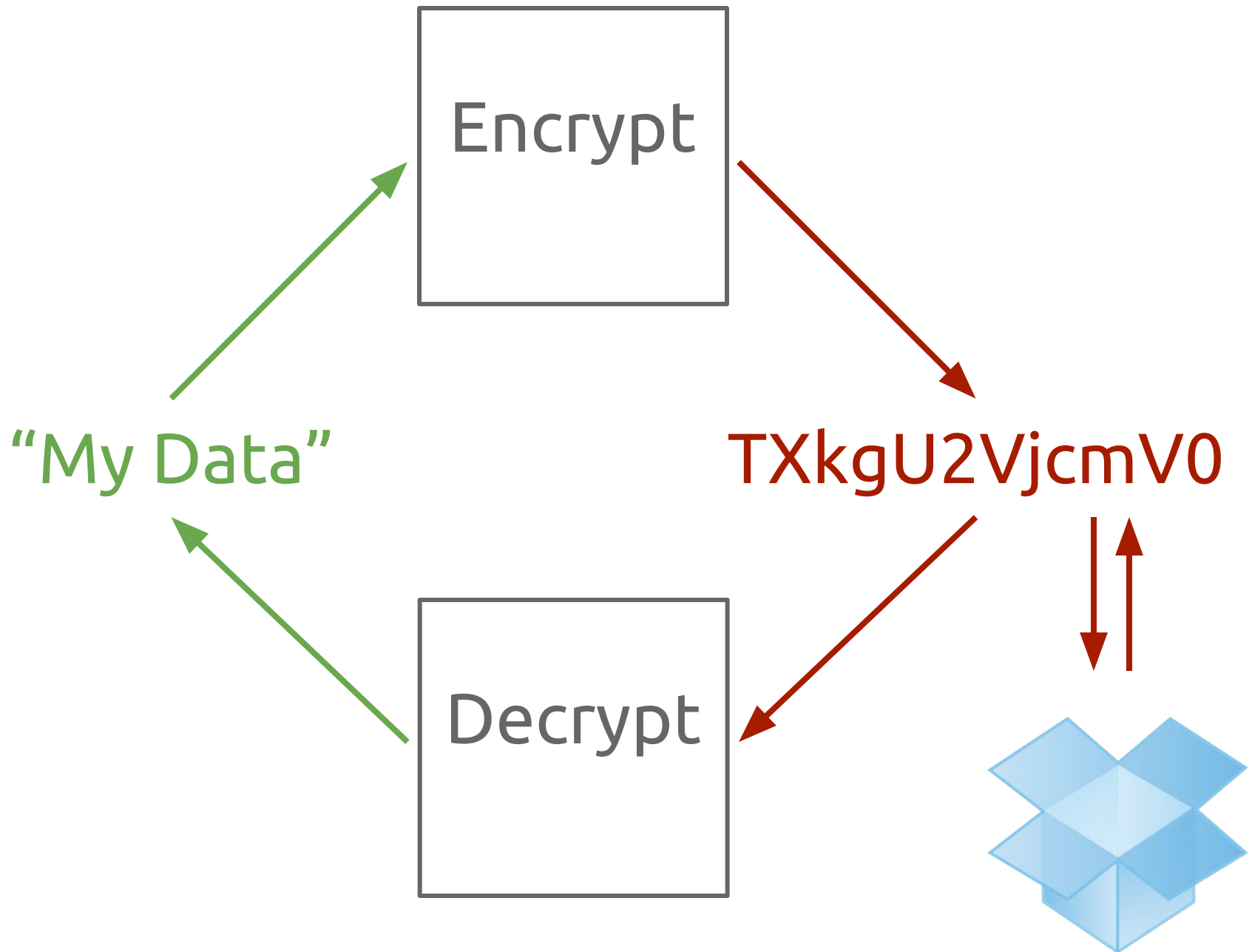


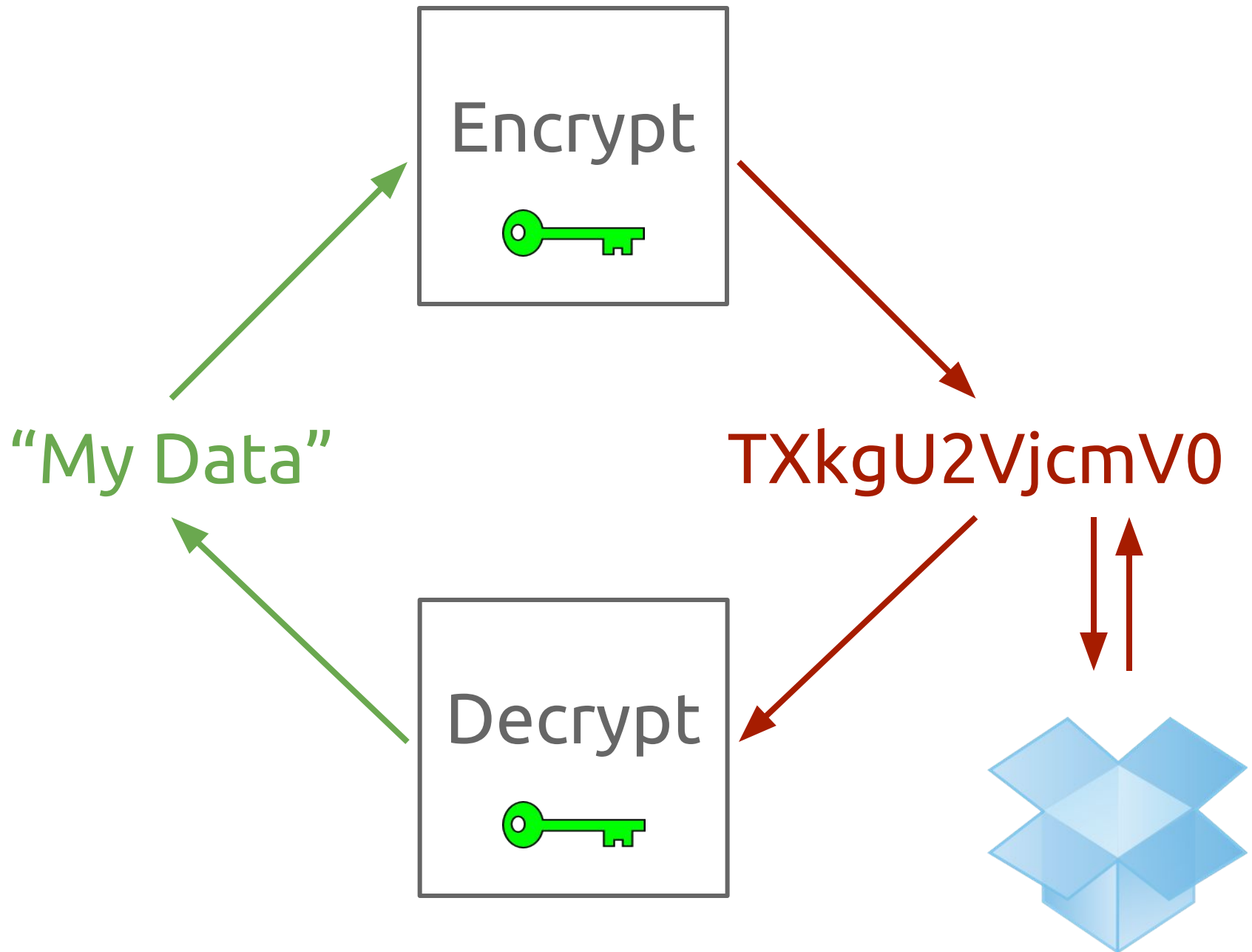


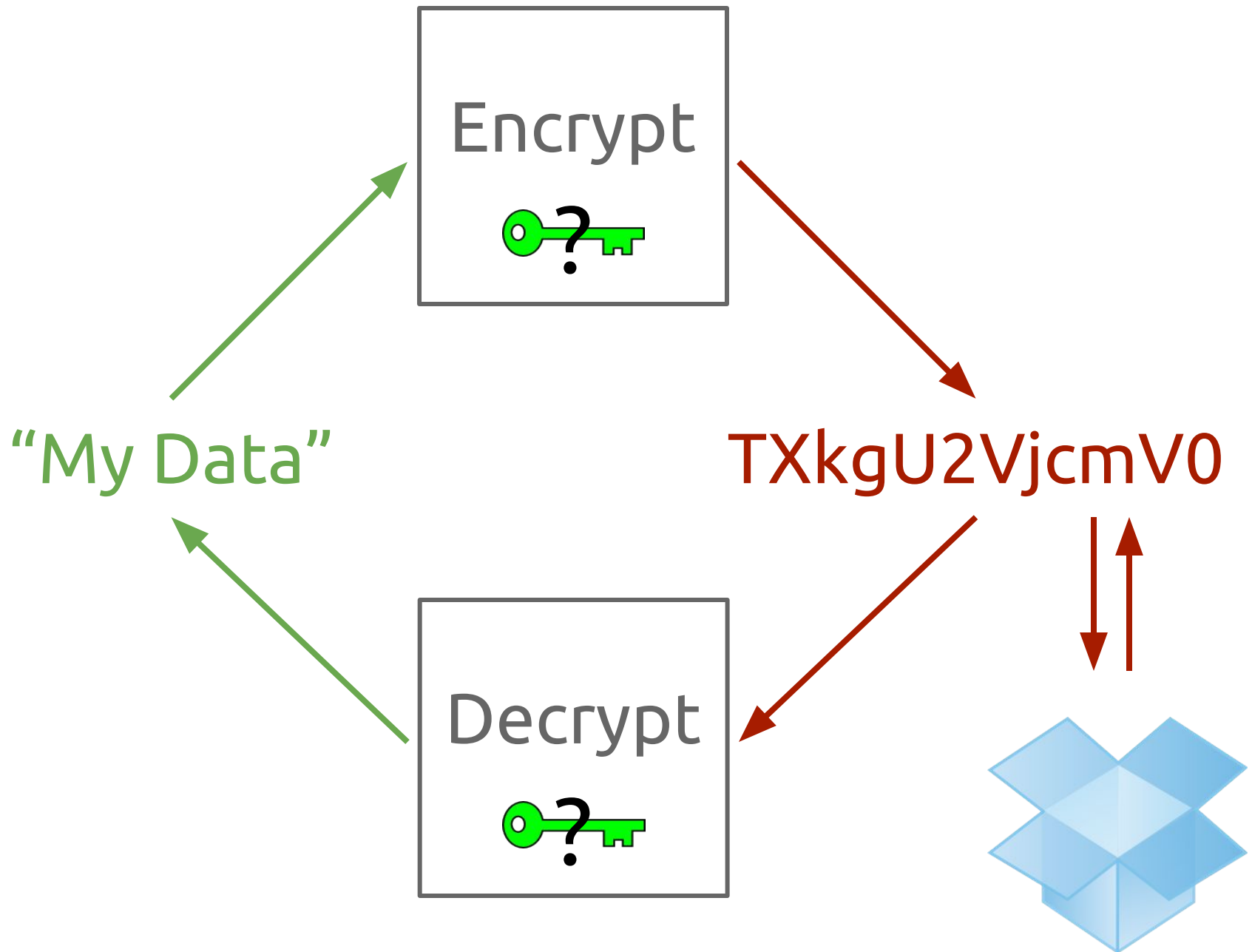
*Cryptography!*



*Cryptography!*



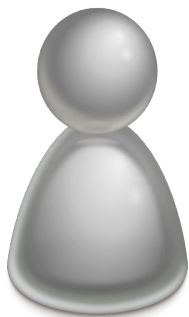




# Related Work

## (Chapter 4)

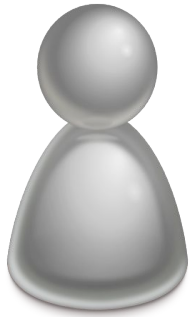
# Secret Storage





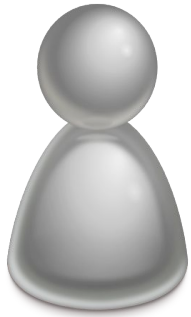


**LastPass** \*\*\*\*



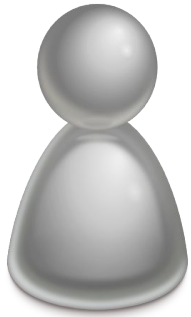
**LastPass** \*\*\*\*

- + Consumer secret storage
- + Encourages use of random passwords
- + Simple browser integration



**LastPass** \*\*\*\*

- + Consumer secret storage
- + Encourages use of random passwords
- + Simple browser integration
- Requires single (semi-)trusted third party



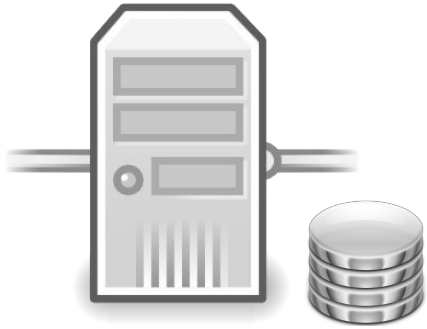
**LastPass** \*\*\*\*

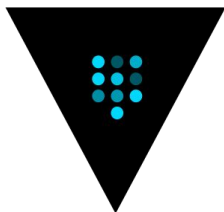
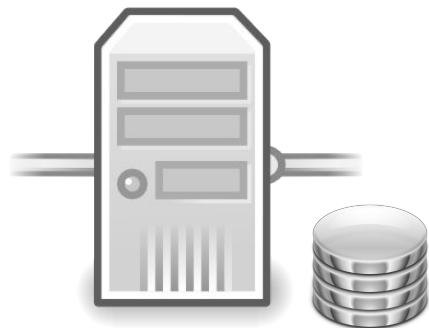
- + Consumer secret storage
- + Encourages use of random passwords
- + Simple browser integration
- Requires single (semi-)trusted third party
- Passwords, not general purpose secrets



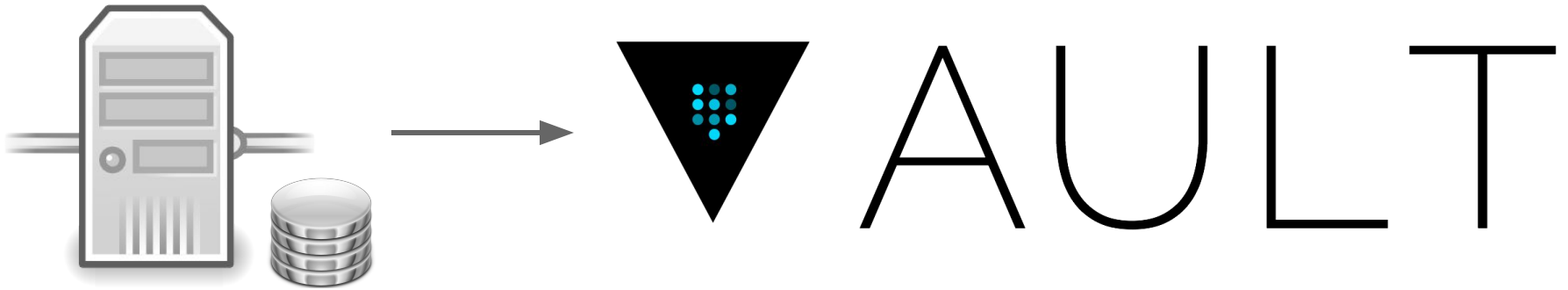
**LastPass** \*\*\*\*

- + Consumer secret storage
- + Encourages use of random passwords
- + Simple browser integration
- Requires single (semi-)trusted third party
- Passwords, not general purpose secrets
- Not designed for automated use cases



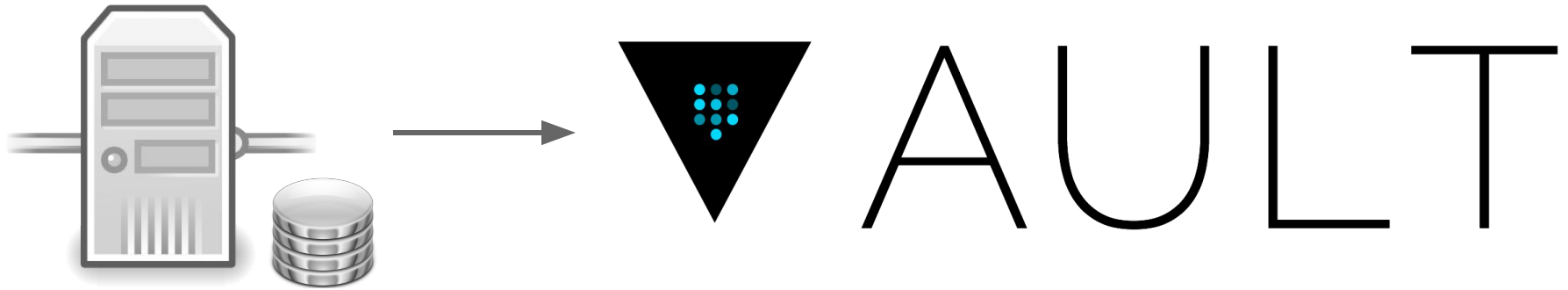


AULT

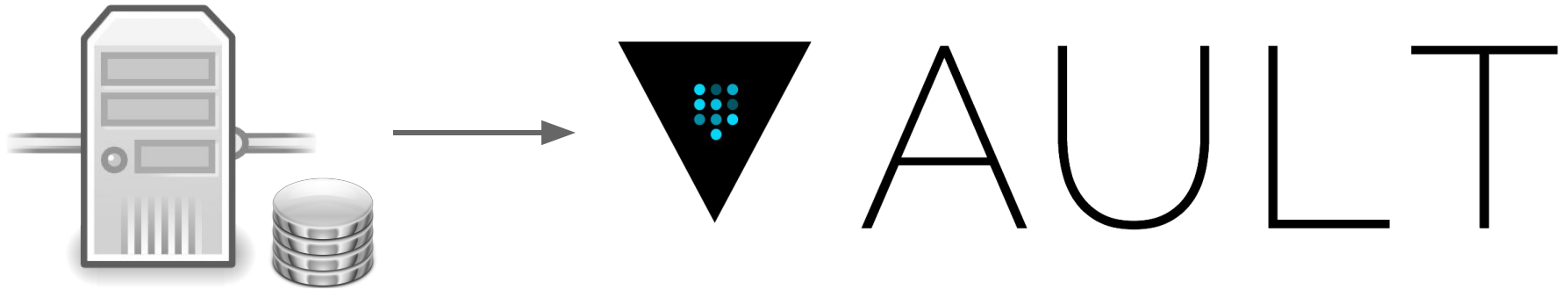


- + Datacenter secret storage
- + Designed for automated use cases
- + Support for auditing, leasing, etc

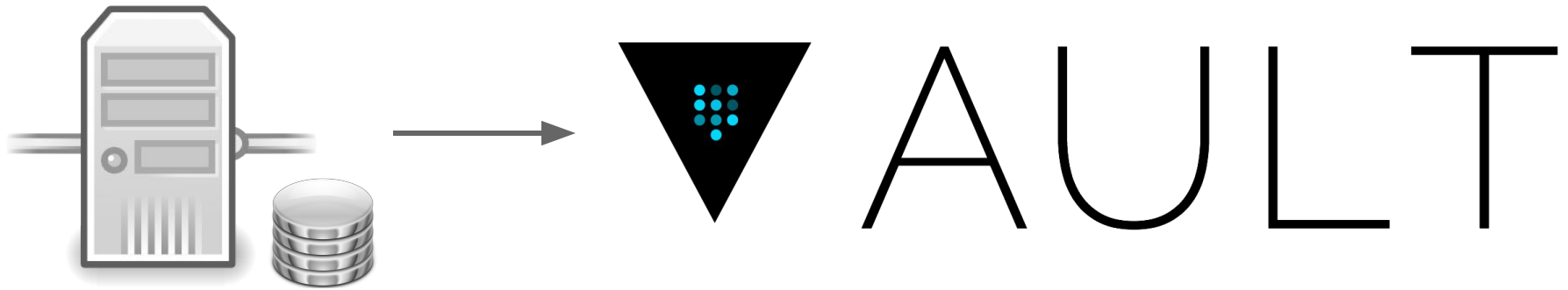




- + Datacenter secret storage
- + Designed for automated use cases
- + Support for auditing, leasing, etc
- Requires single trusted server



- + Datacenter secret storage
- + Designed for automated use cases
- + Support for auditing, leasing, etc
- Requires single trusted server
- Lacks support for out-of-band approval



- + Datacenter secret storage
- + Designed for automated use cases
- + Support for auditing, leasing, etc
- Requires single trusted server
- Lacks support for out-of-band approval
- Designed for single administrative domain

# Goals



- + Quantify and analyze third-party trust exposure inherent in modern applications

- + Quantify and analyze third-party trust exposure inherent in modern applications
- + Provide primitives for minimizing, managing, and monitoring third party trust exposure

- + Quantify and analyze third-party trust exposure inherent in modern applications
- + Provide primitives for minimizing, managing, and monitoring third party trust exposure
- + Use primitives to create security and privacy enhancing systems for modern applications



- + Quantify and analyze third-party trust exposure inherent in modern applications
- + Provide primitives for minimizing, managing, and monitoring third party trust exposure
- + Use primitives to create security and privacy enhancing systems for modern applications

# An Issue of Trust (Chapter 5)

# Analysis Framework

# Analysis Framework

Degree of Trust  
(Capabilities)

# Analysis Framework

Degree of Trust  
(Capabilities)

Types of Violation  
(Attacks)

# Degree of Trust

Storage (S)

Access (R)

Manipulation (W)

Meta-Analysis (M)

# Degree of Trust

## Storage (S)

*Can a third party faithfully store private user data and make it available to the user upon request?*

Access (R)

Manipulation (W)

Meta-Analysis (M)

# Degree of Trust

Storage (S)

Access (R)

*Can a third party read and interpret  
the private user data they store?*

Manipulation (W)

Meta-Analysis (M)



# Degree of Trust

Storage (S)

Access (R)

Manipulation (W)

*Can a third party modify the  
private user data to which they have access?*

Meta-Analysis (M)

# Degree of Trust

Storage (S)

Access (R)

Manipulation (W)

**Meta-Analysis (M)**

*Can a third party gather user metadata  
related to any stored private user data?*

# Types of Violation

Implicit (P)

Compelled (C)

Unintentional (U)

Colluding (L)

# Types of Violation

## Implicit (P)

*Occurs when a third party violates a user's trust in a manner approved by the third party.*

Compelled (C)

Unintentional (U)

Colluding (L)

# Types of Violation

Implicit (P)

Compelled (C)

*Occurs when a third party is compelled by another actor to violate a user's trust.*

Unintentional (U)

Colluding (L)

# Types of Violation

Implicit (P)

Compelled (C)

Unintentional (U)

*Occurs when a third party  
unintentionally violates a user's trust.*

Colluding (L)

# Types of Violation

Implicit (P)

Compelled (C)

Unintentional (U)

**Colluding (L)**

*Occurs when multiple trusted parties collude to gain capabilities beyond what the user intended each to have.*

# Degree of Trust

Storage (S)

Access (R)

Manipulation (W)

Meta-Analysis (M)

# Types of Violation

Implicit (P)

Compelled (C)

Unintentional (U)

Colluding (L)



# Degree of Trust

Storage (S)

Access (R)

Manipulation (W)

Meta-Analysis (M)

# Types of Violation

Implicit (P)

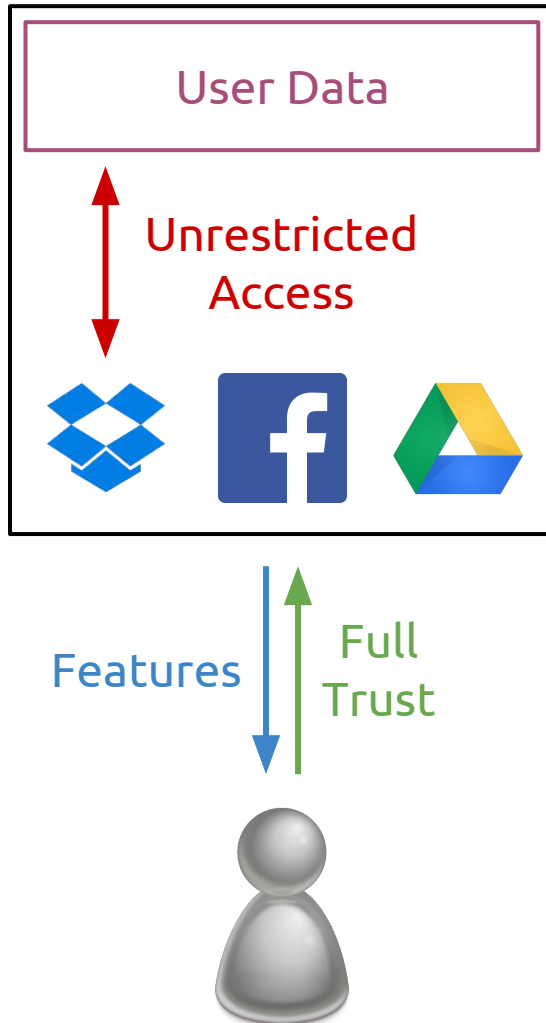
Compelled (C)

Unintentional (U)

Colluding (L)

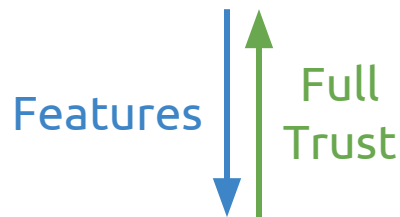
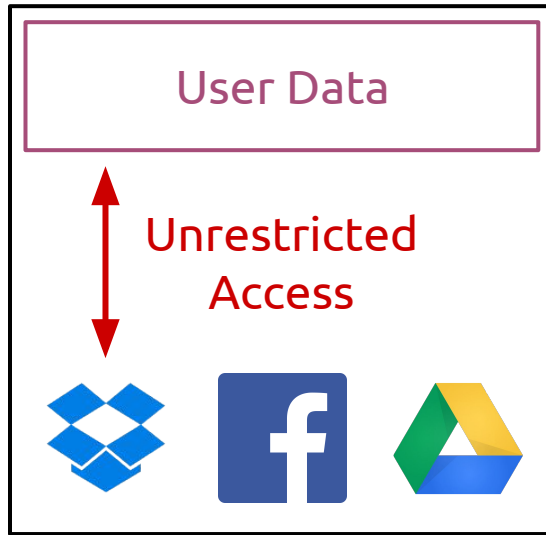
# Traditional Trust Model

Feature Provider



# Traditional Trust Model

Feature Provider



Storage (S)

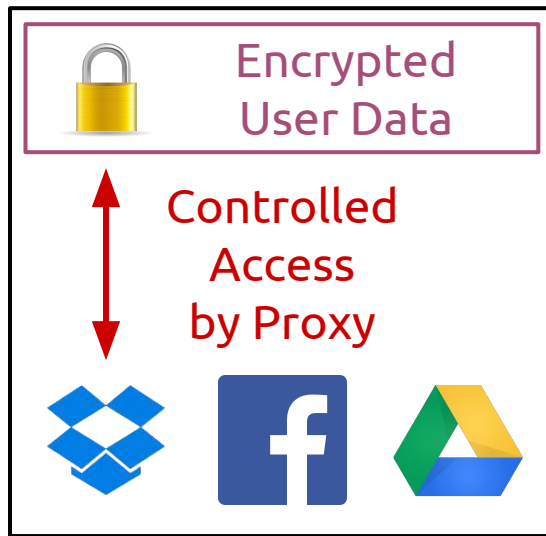
Access (R)

Manipulation (W)

Meta-Analysis (M)

# SSaaS Trust Model

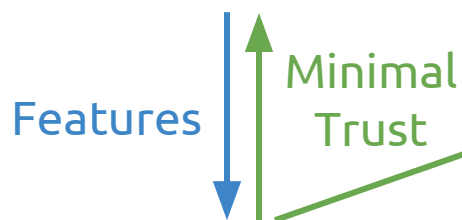
## Feature Provider



## Secret Storage Provider

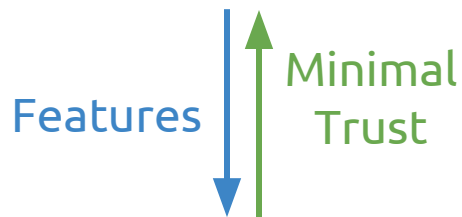
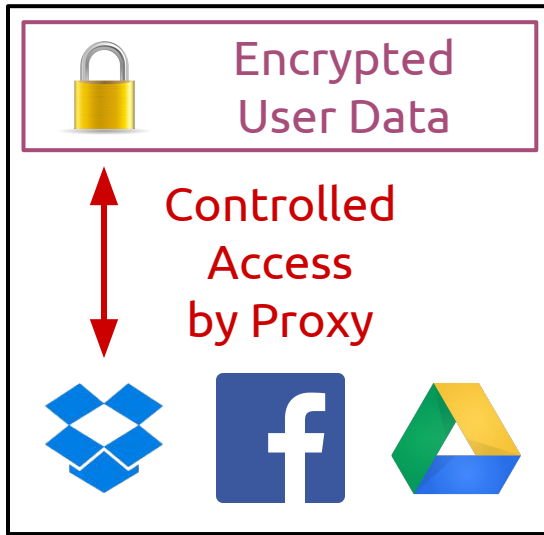


Controlled Access



# SSaaS Trust Model

Feature Provider



Storage (S)

~~Access (R)~~

~~Manipulation (W)~~

Meta-Analysis (M)

# SSaaS Trust Model

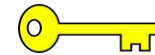
Storage (S)

~~Access (R)~~

~~Manipulation (W)~~

Meta-Analysis (M)

Secret Storage  
Provider

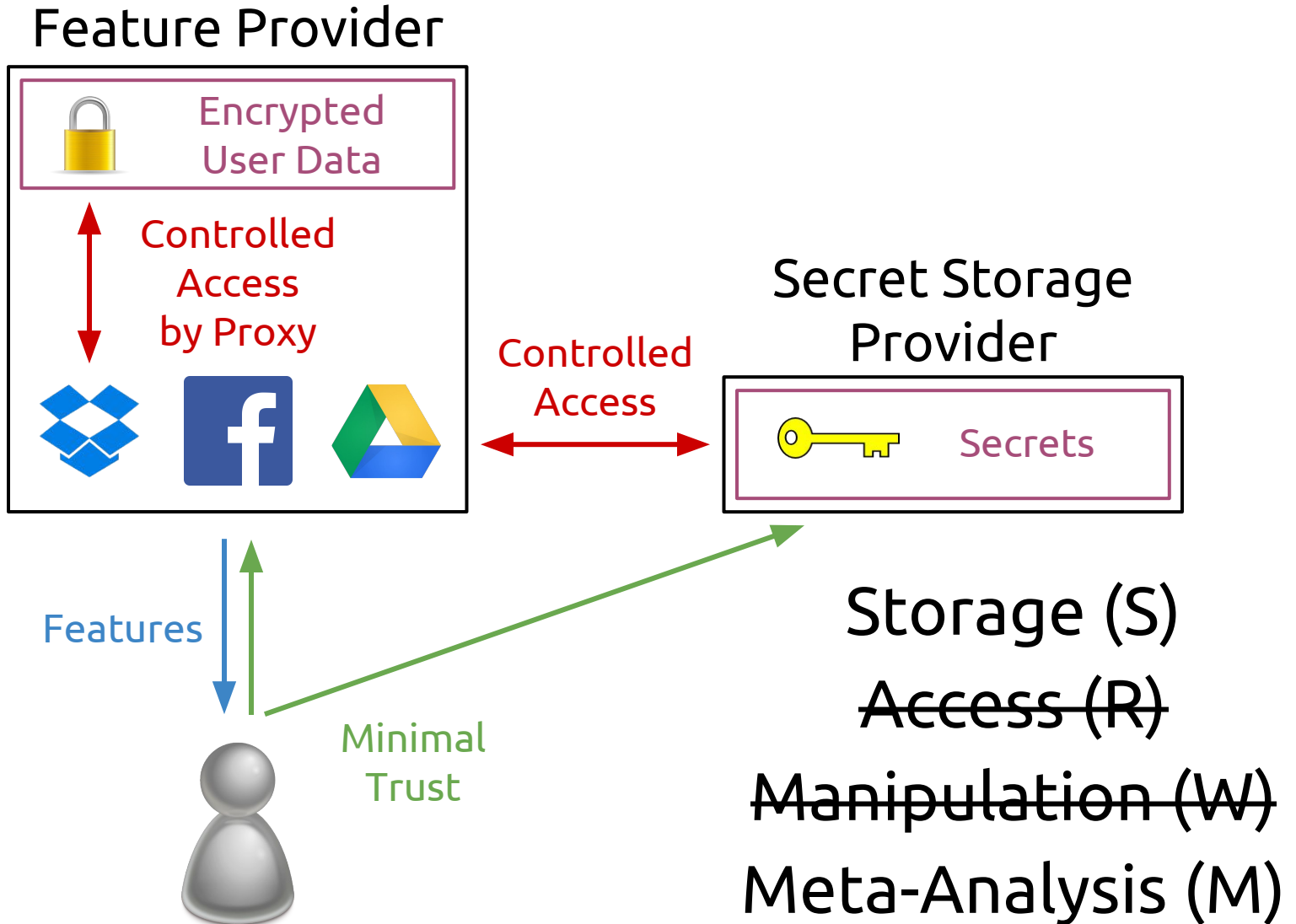


Secrets

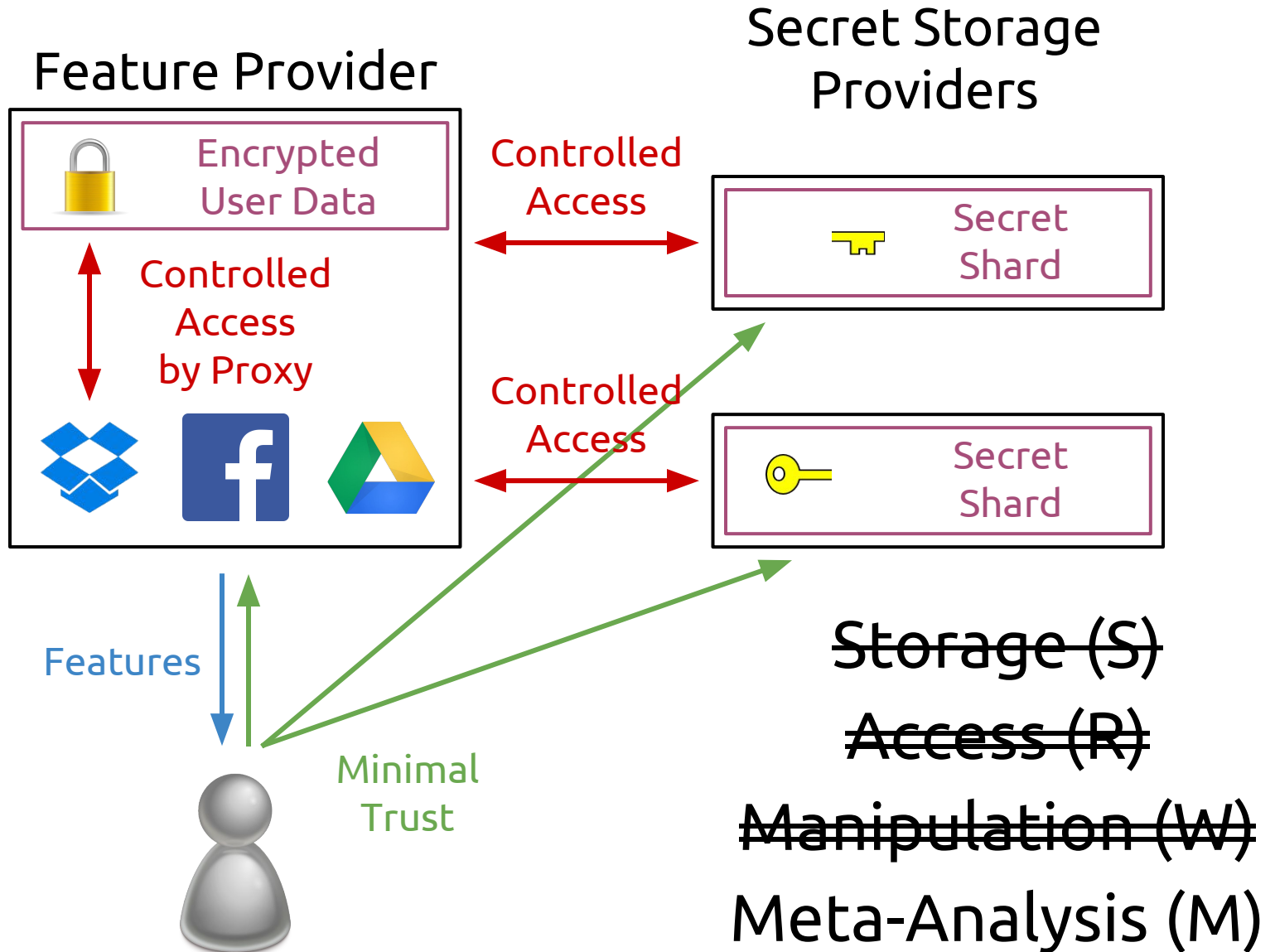


Minimal  
Trust

# SSaaS Trust Model



# SSaaS Trust Model





# Degree of Trust

Storage (S)

Access (R)

Manipulation (W)

Meta-Analysis (M)

# Types of Violation

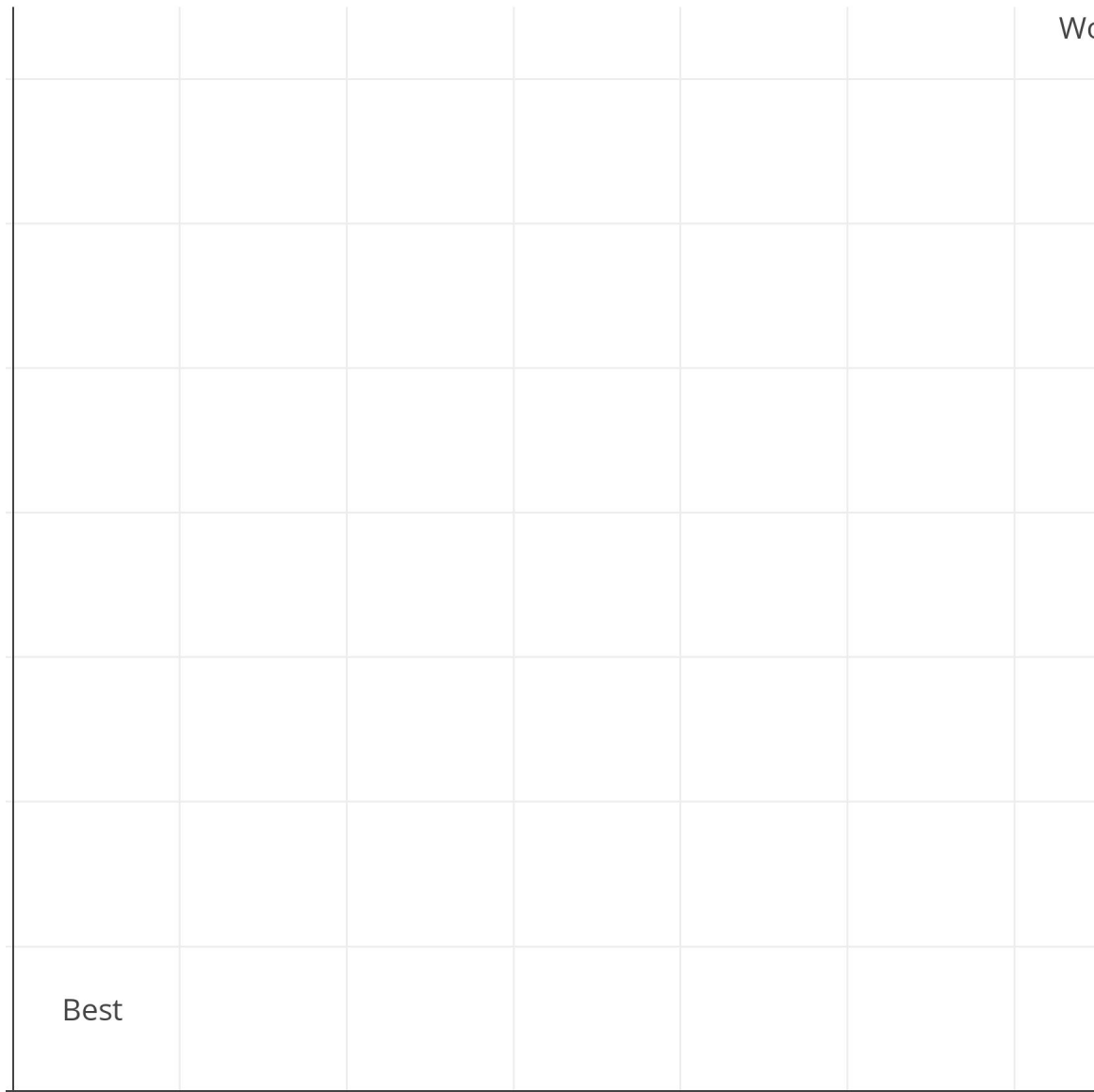
Implicit (P)

Compelled (C)

Unintentional (U)

Colluding (L)

Risk of Violation

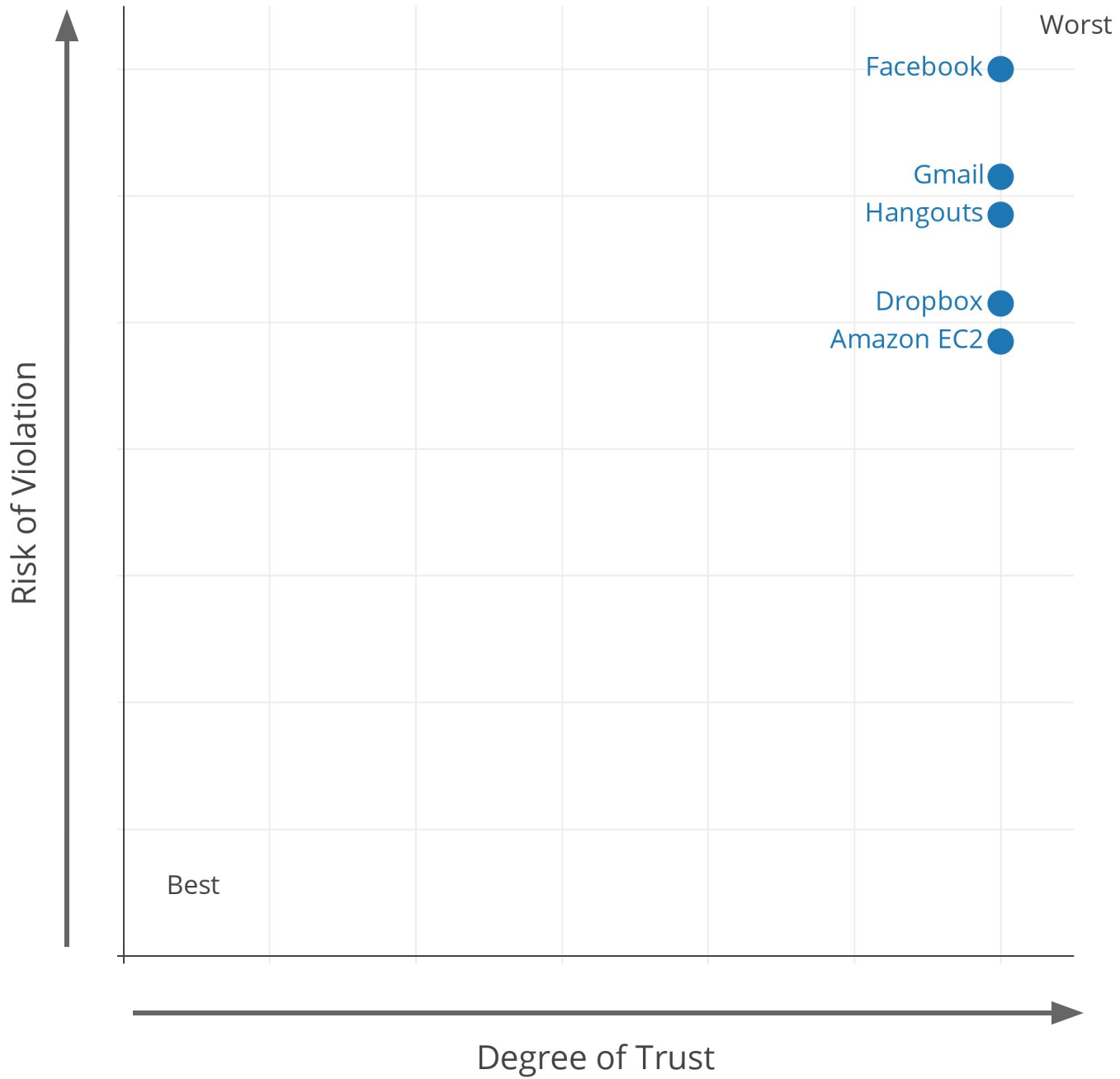


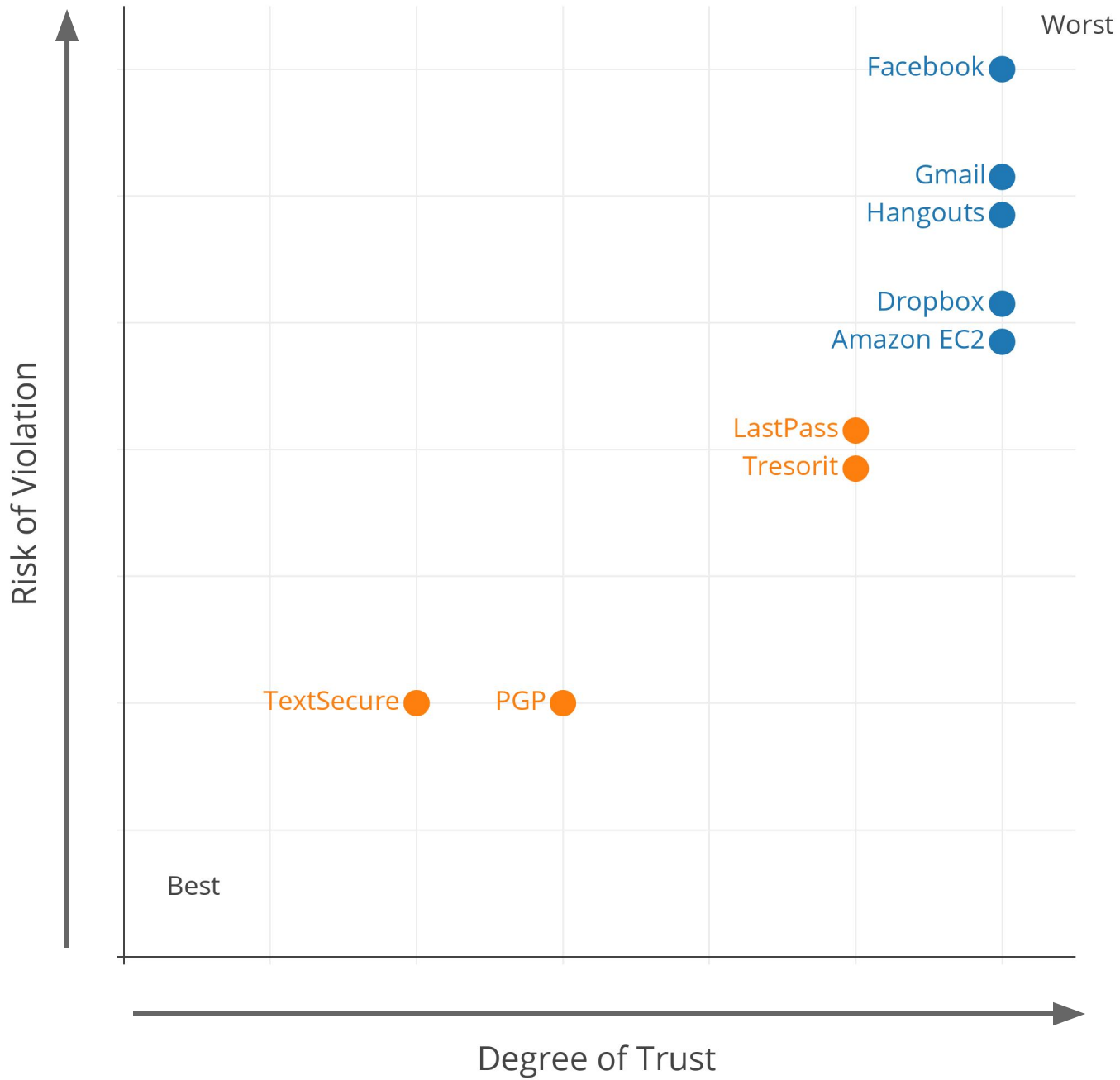
Worst

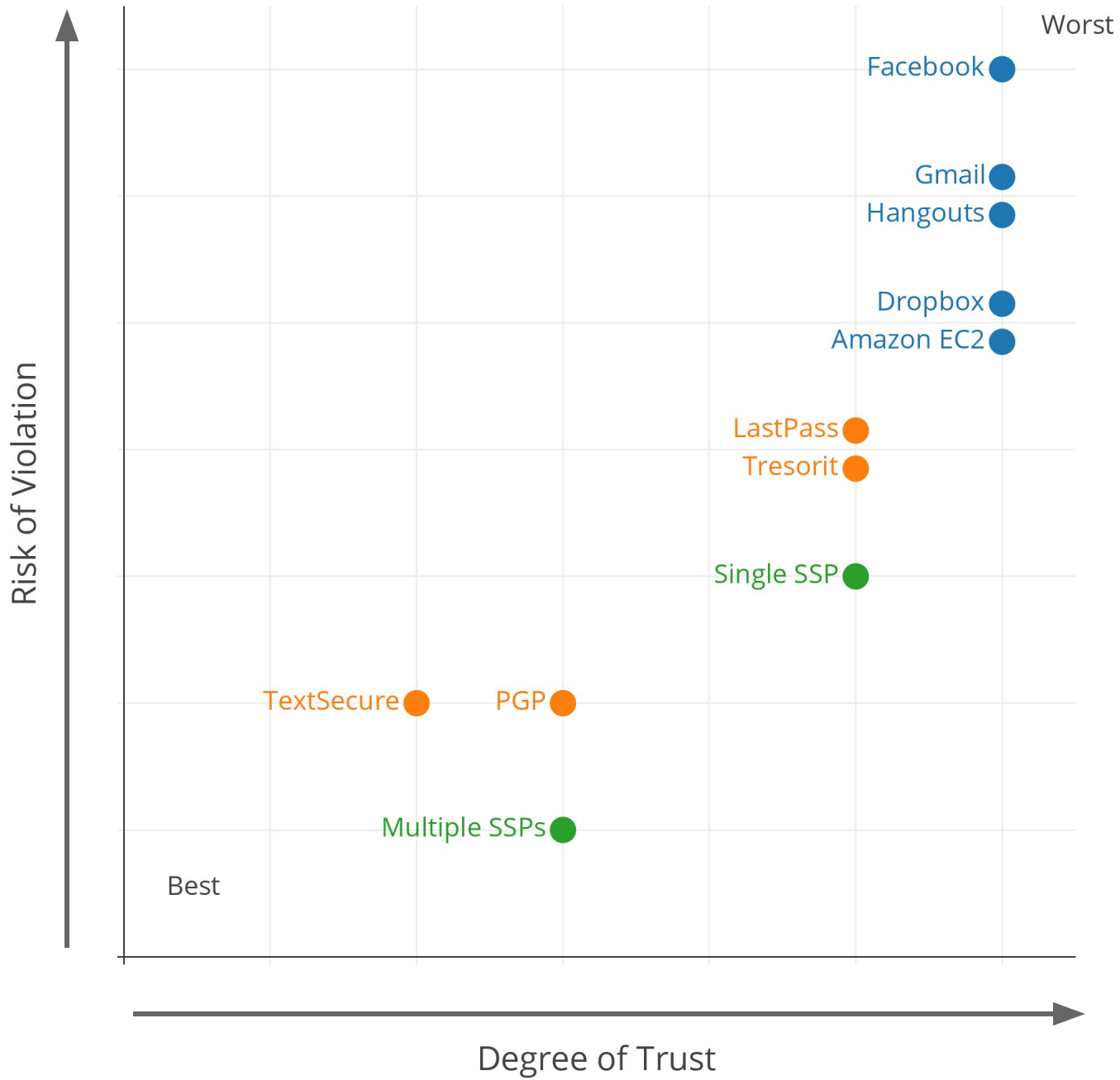
Best

Degree of Trust









- + Quantify and analyze third-party trust exposure inherent in modern applications
- + Provide primitives for minimizing, managing, and monitoring third party trust exposure
- + Use primitives to create security and privacy enhancing systems for modern applications

# Secret Storage as a Service (Chapter 6)





# Centralized Secret Storage

Centralized Secret Storage

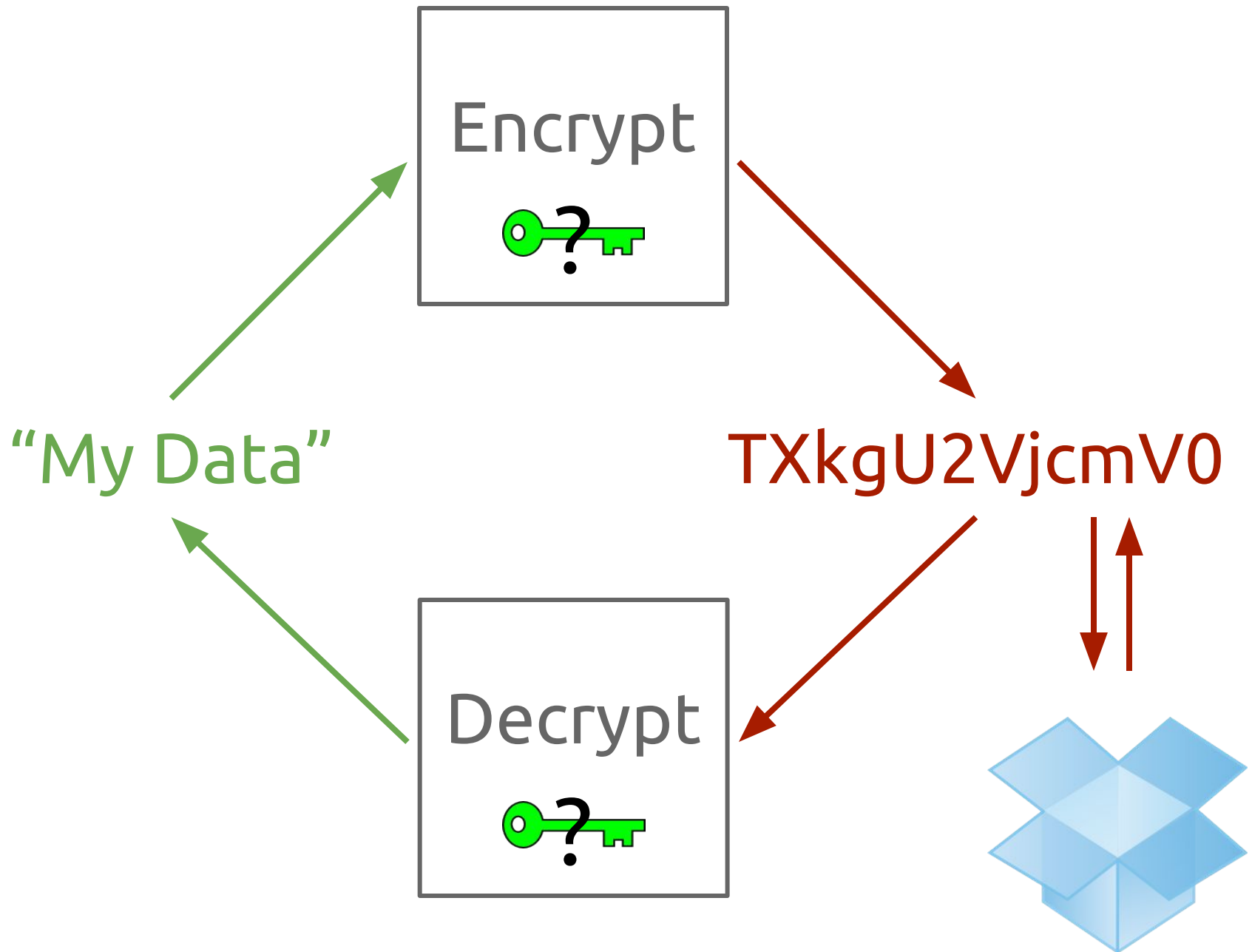
Flexible Access Control

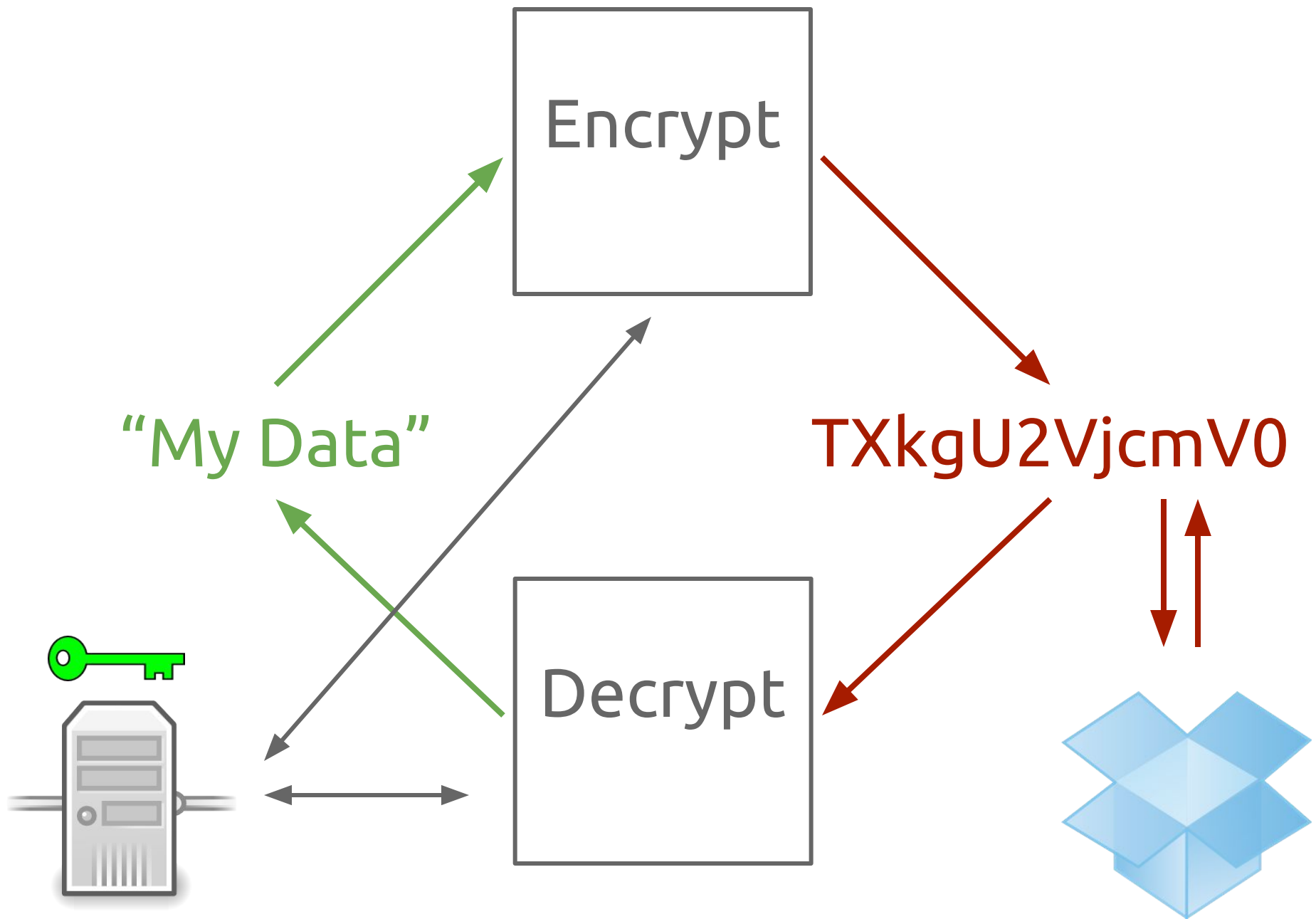
Centralized Secret Storage

Flexible Access Control

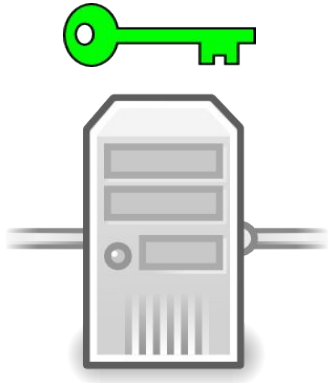
Auditing and Revocation

# SSaaS Architecture

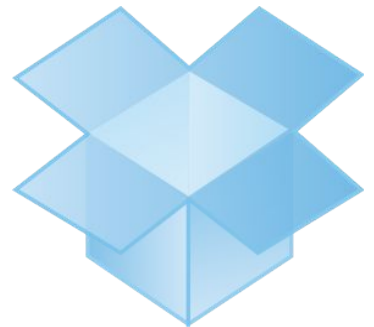




# Secret Storage Provider (SSP)



# Feature Provider (FP)

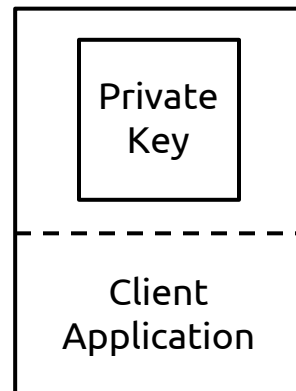


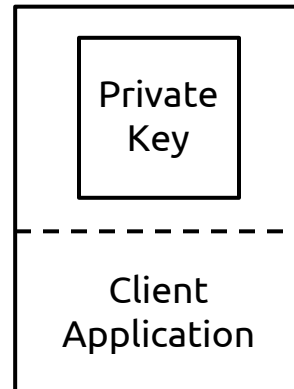
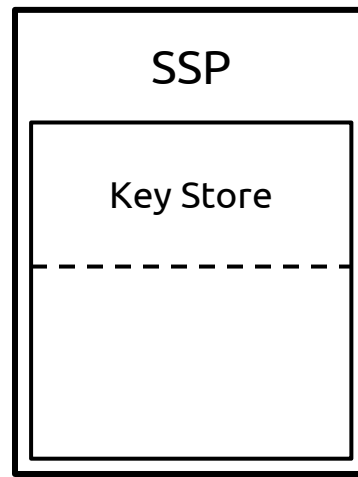


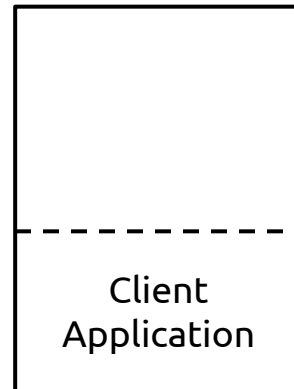
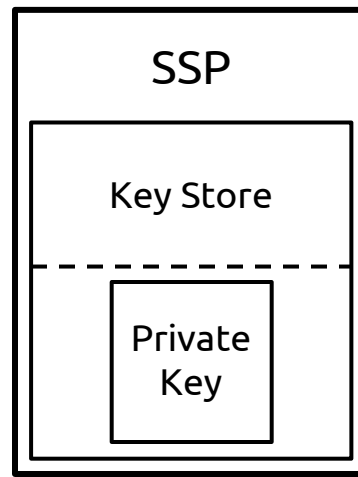
SSaaS

Security & Trust

Single SSP







Should we trust  
a single provider?

Maybe

Incentives aligned with upholding trust



Incentives aligned with upholding trust

Reputation at stake

Incentives aligned with upholding trust

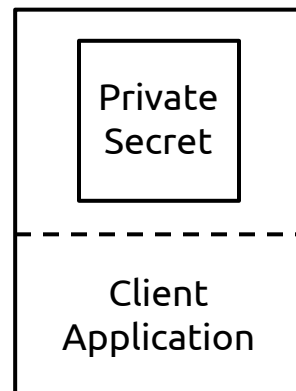
Reputation at stake

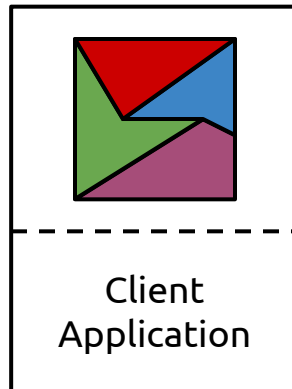
Still a “minimally trusted” entity

Must we trust  
a single provider?

No

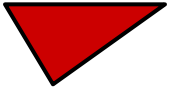
Multiple SSPs



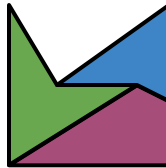


SSP A

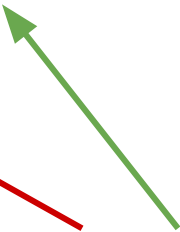
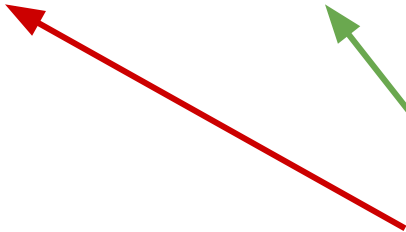
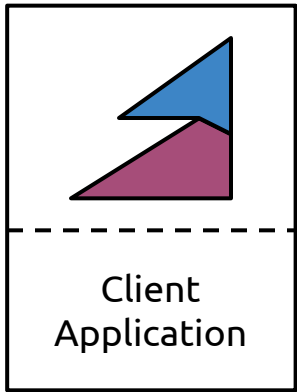
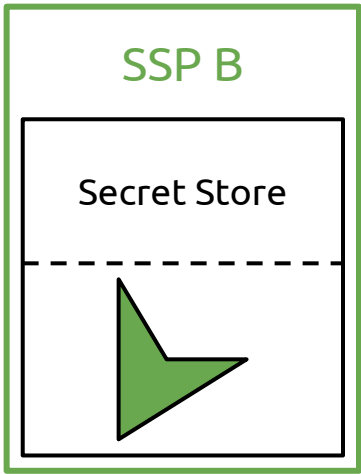
Secret Store

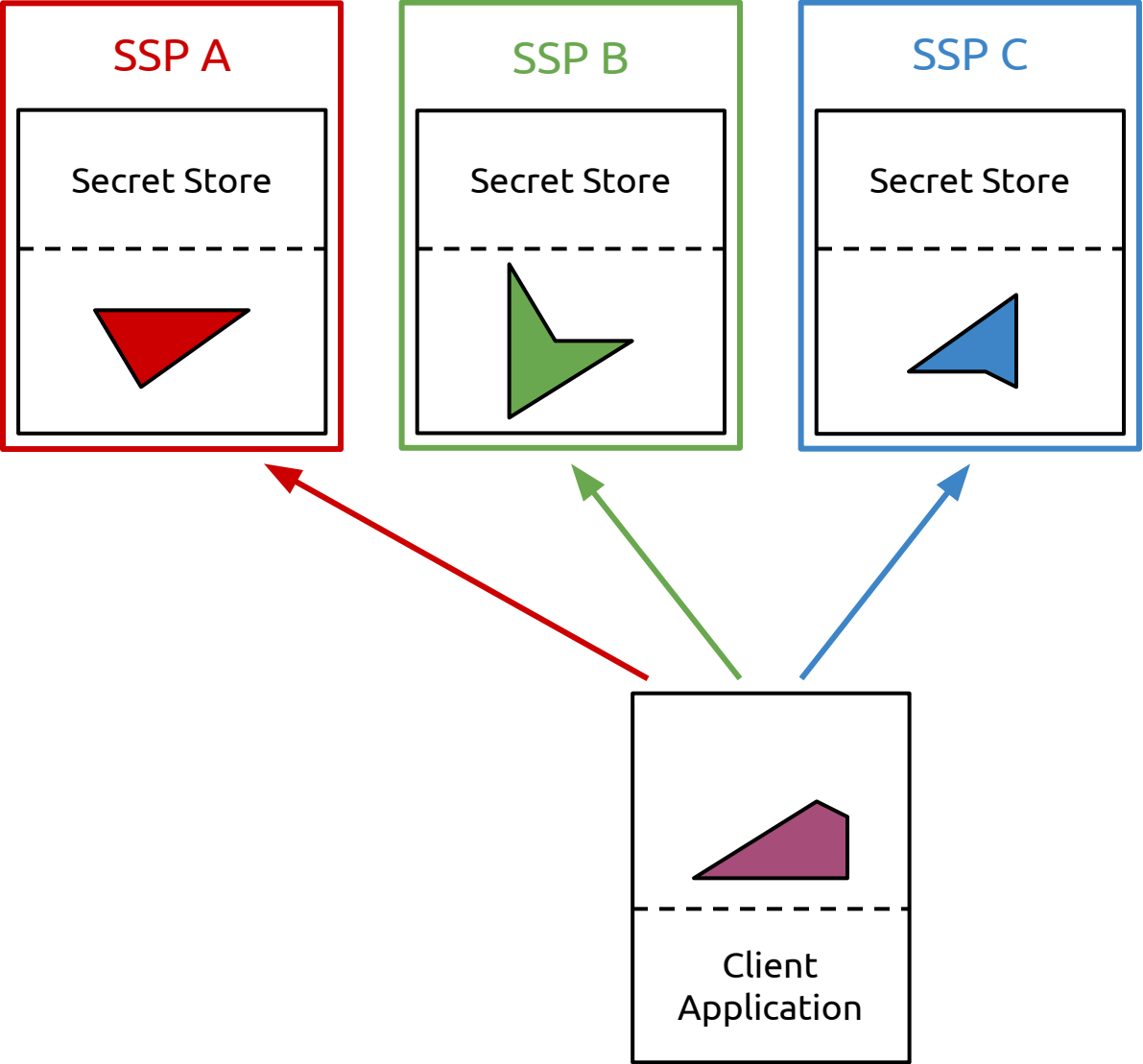


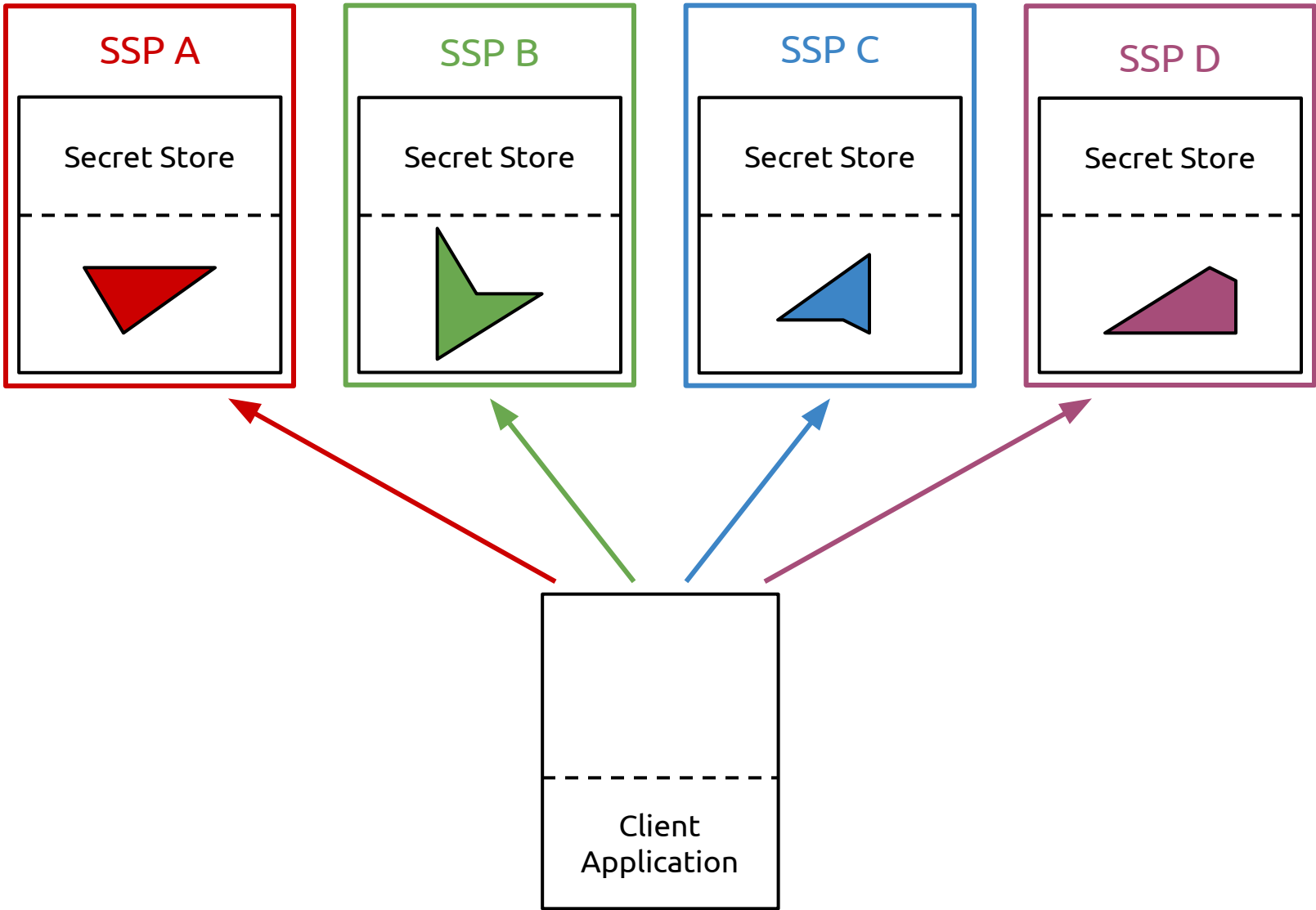
Client  
Application

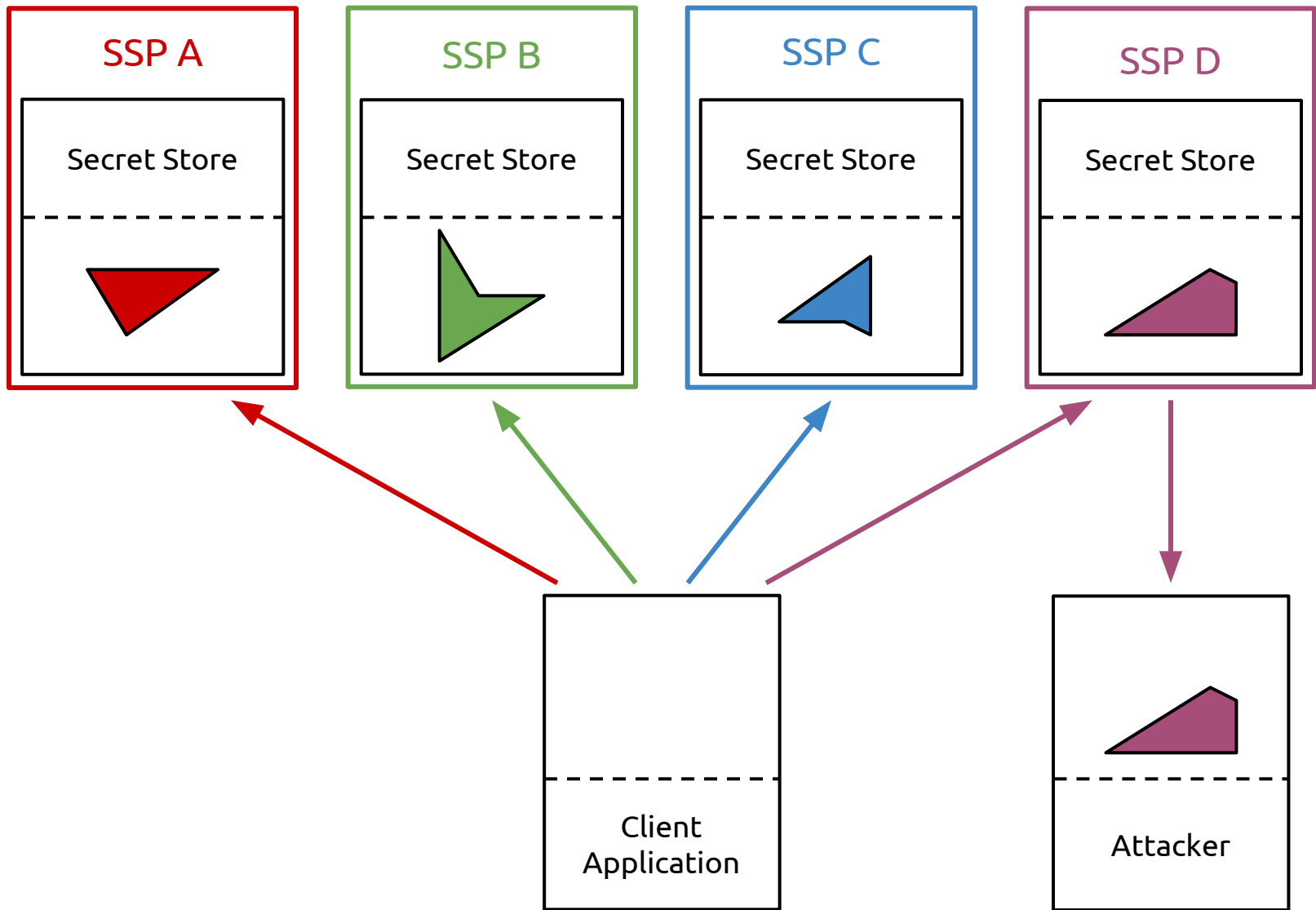


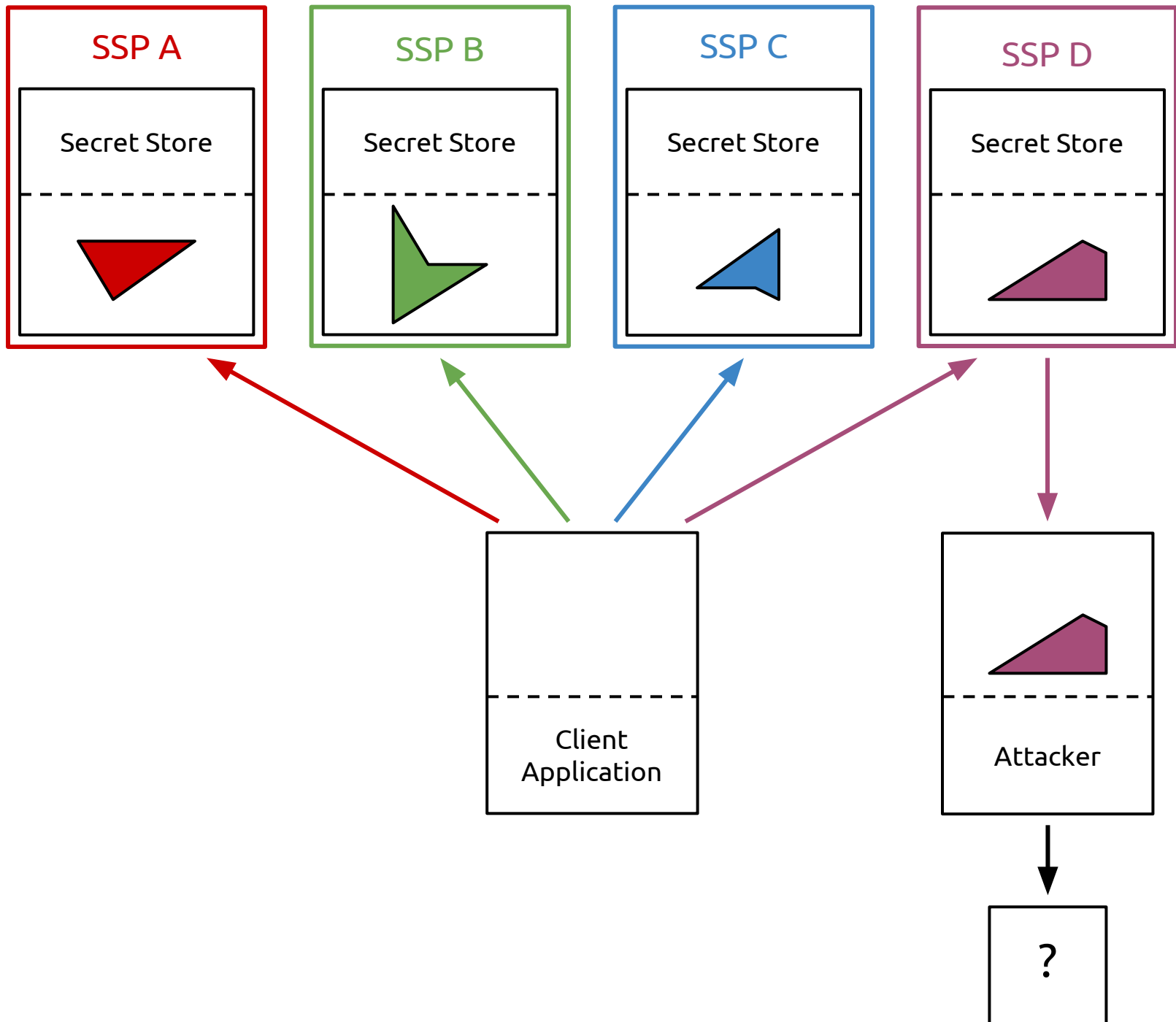












- + Quantify and analyze third-party trust exposure inherent in modern applications
- + Provide primitives for minimizing, managing, and monitoring third party trust exposure
- + Use primitives to create security and privacy enhancing systems for modern applications

# SSaaS Applications (Chapter 7)

**Custos:** A First-Gen SSaaS Prototype  
(Chapter 8)

**Tutamen:** Next-Gen Secret Storage  
(Chapter 9)

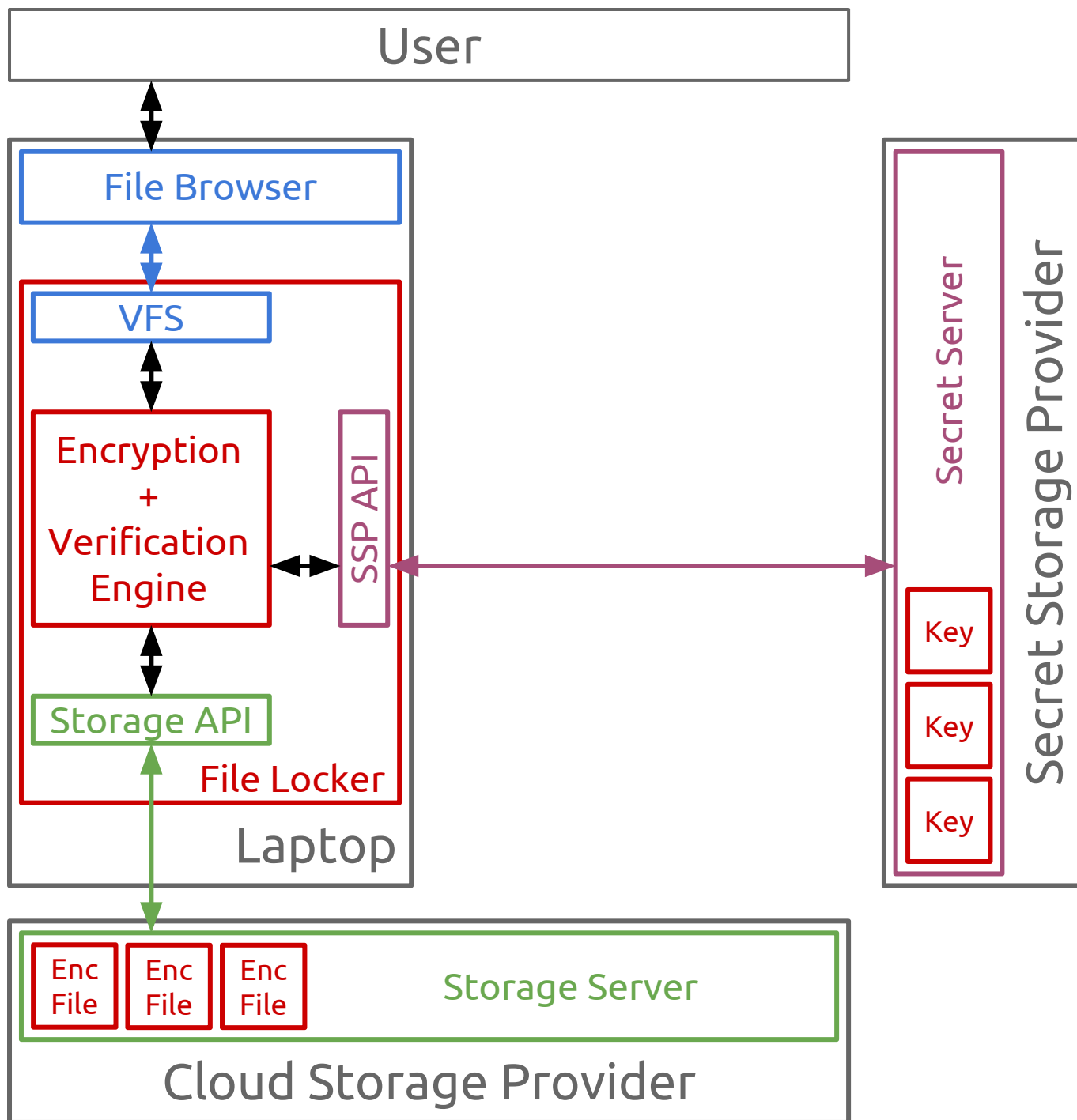
# SSaaS Applications (Chapter 7)

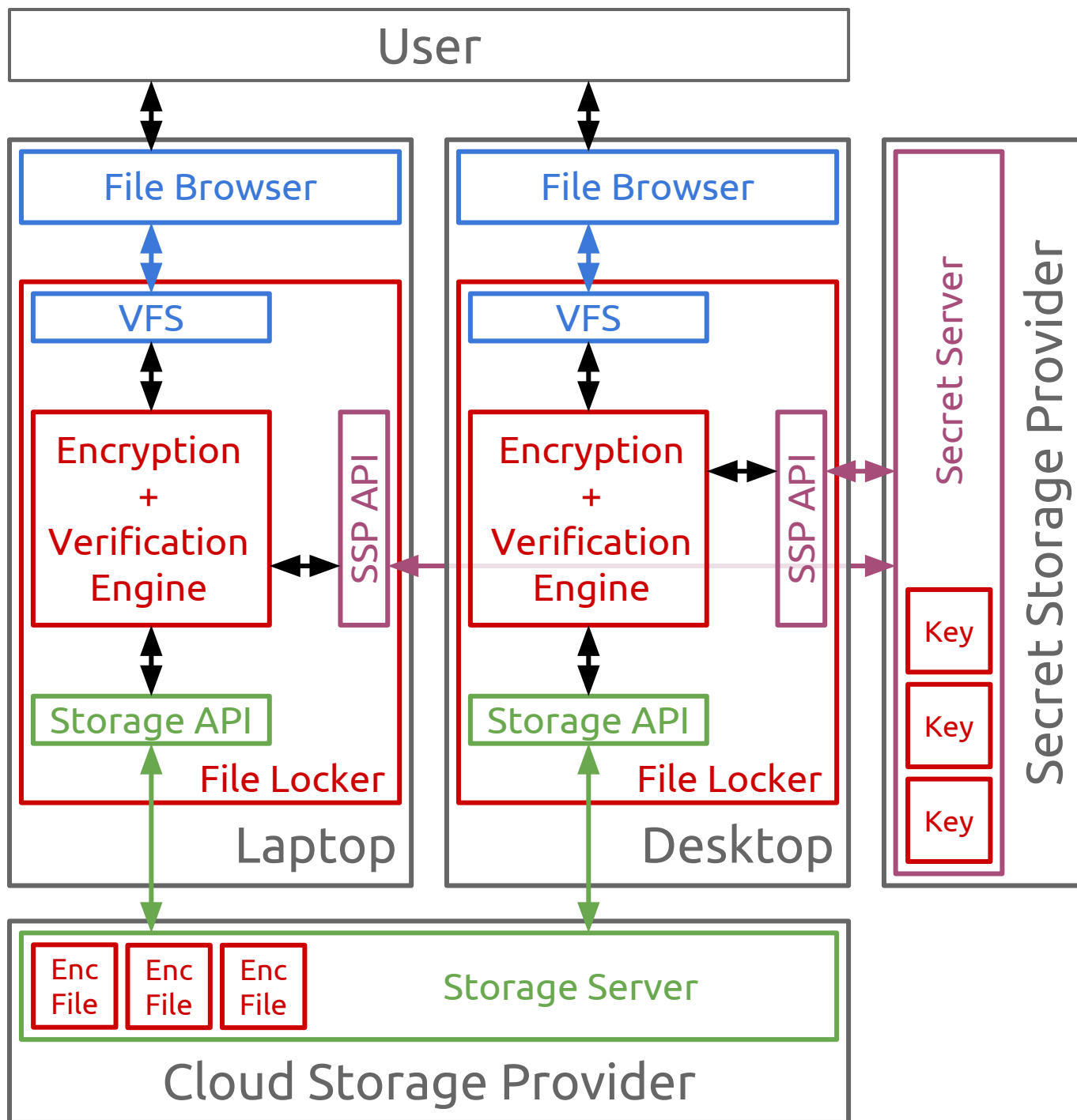
Custos: A First-Gen SSaaS Prototype  
(Chapter 8)

Tutamen: Next-Gen Secret Storage  
(Chapter 9)

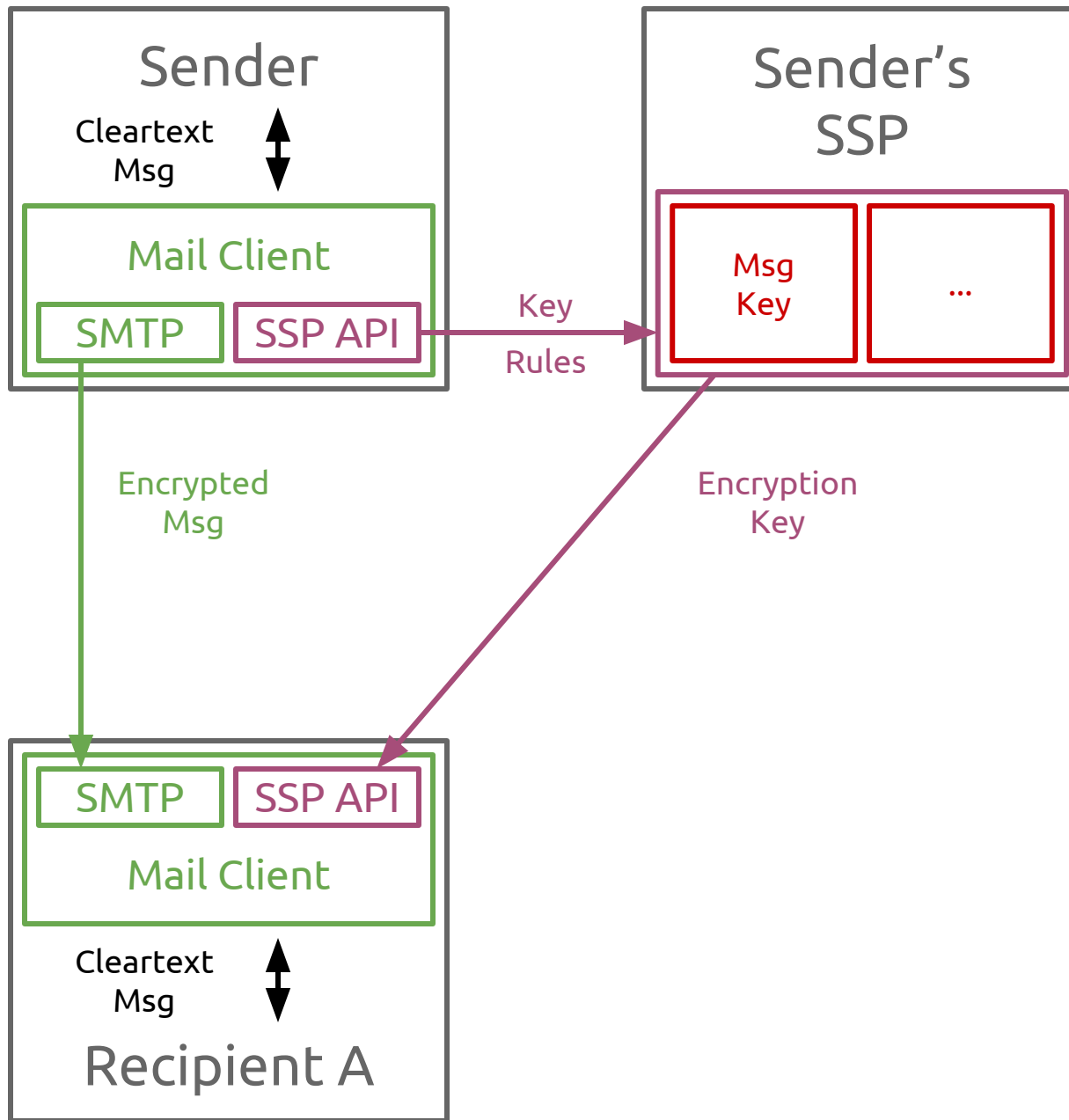


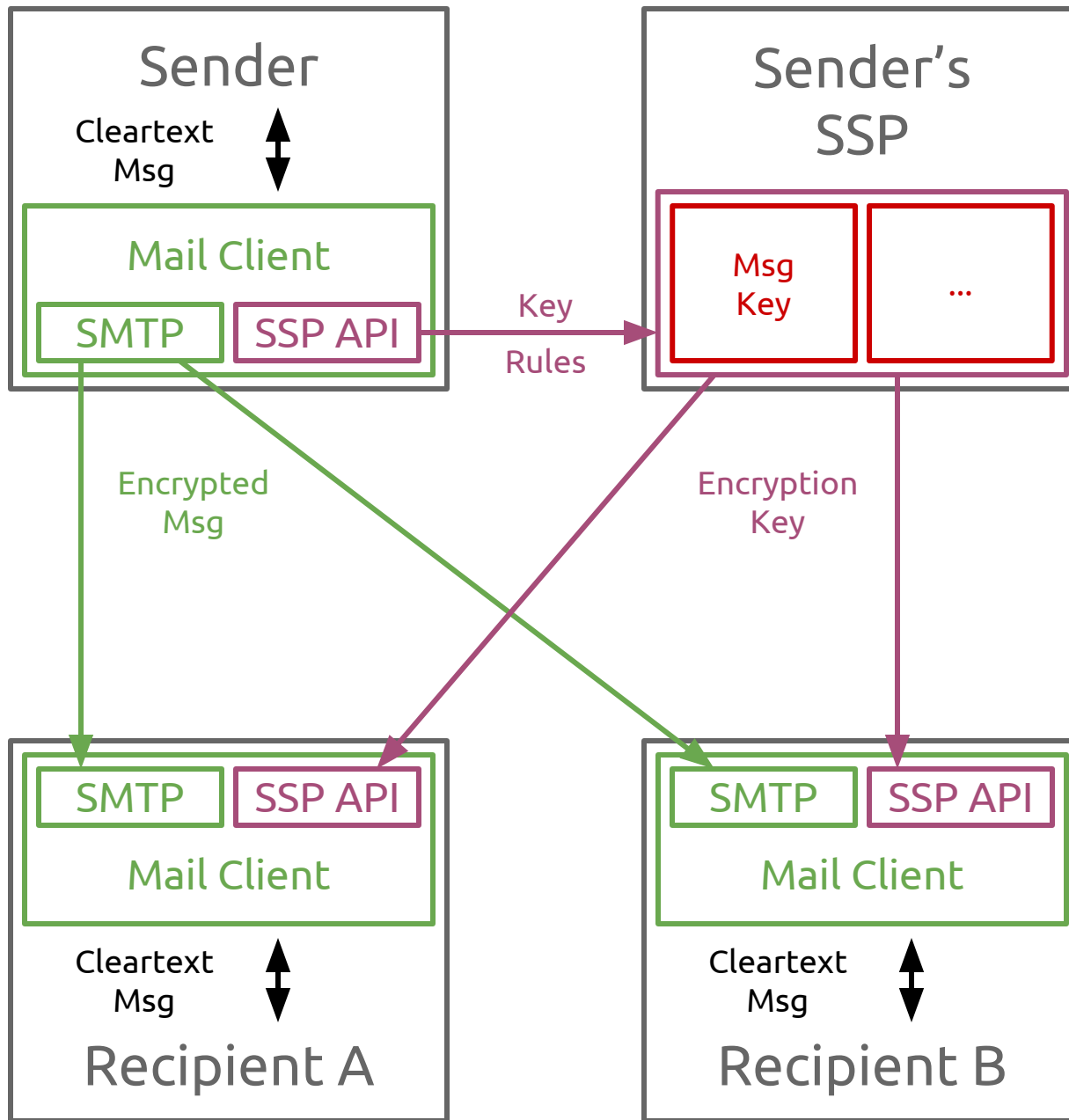
# Storage Applications



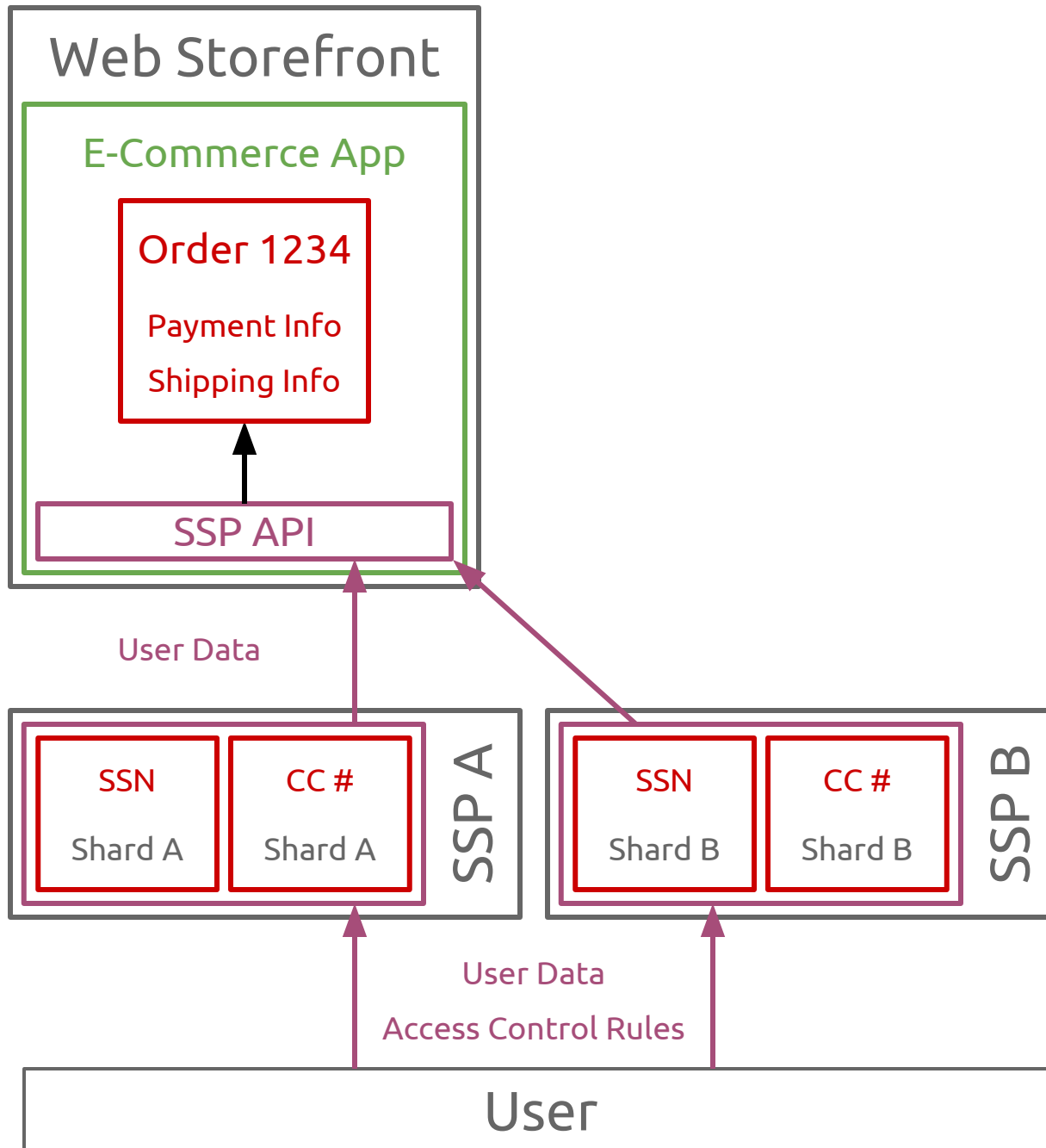


# Communication Applications

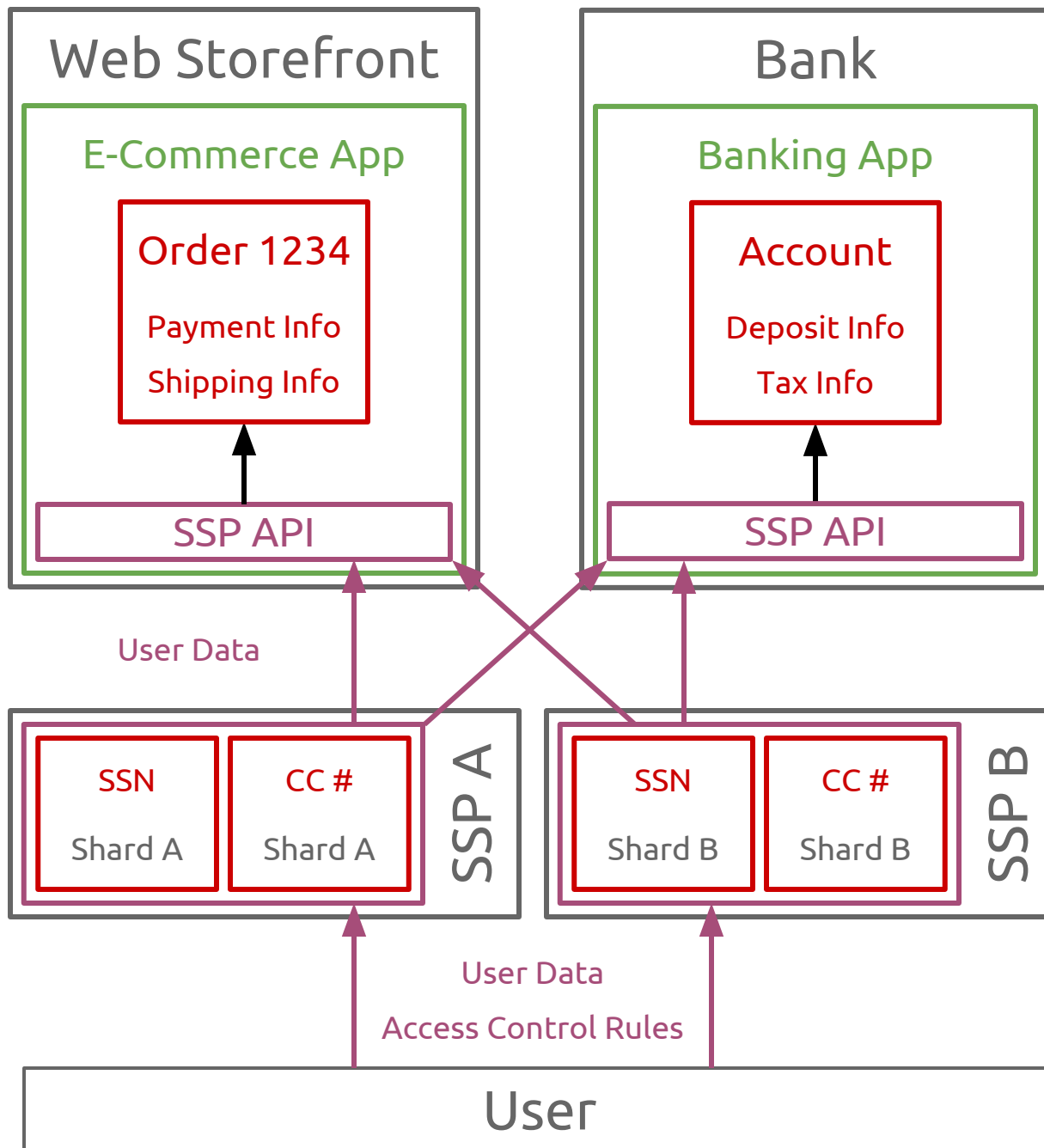




# Personal Data Repository







And so on...

# SSaaS Applications (Chapter 7)

**Custos:** A First-Gen SSaaS Prototype  
(Chapter 8)

**Tutamen:** Next-Gen Secret Storage  
(Chapter 9)

# “Key Storage as a Service” (KSaaS)

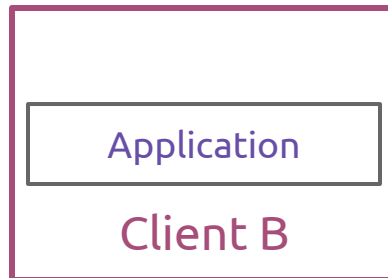
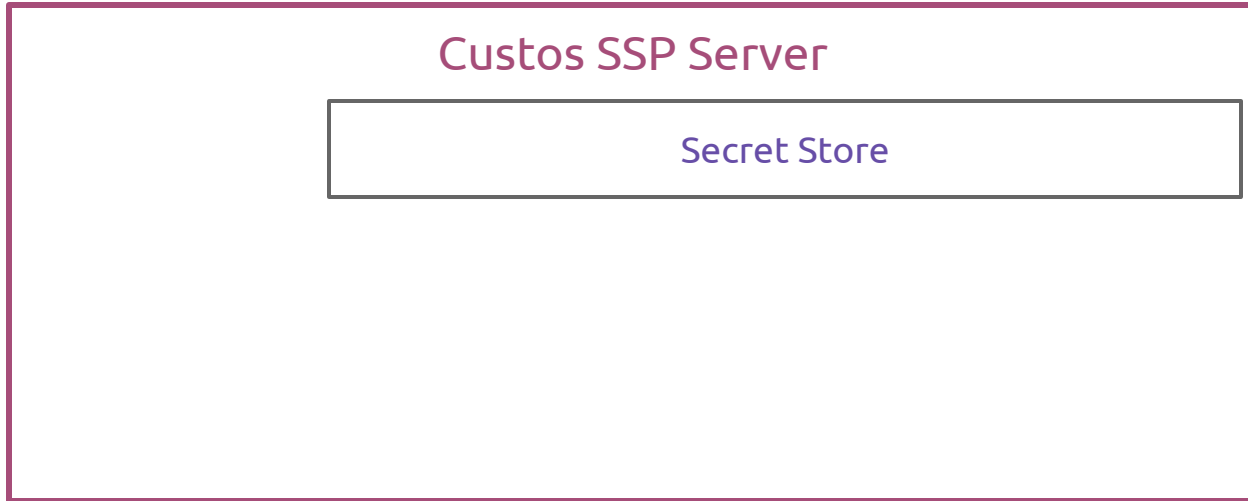
Custos SSP Server

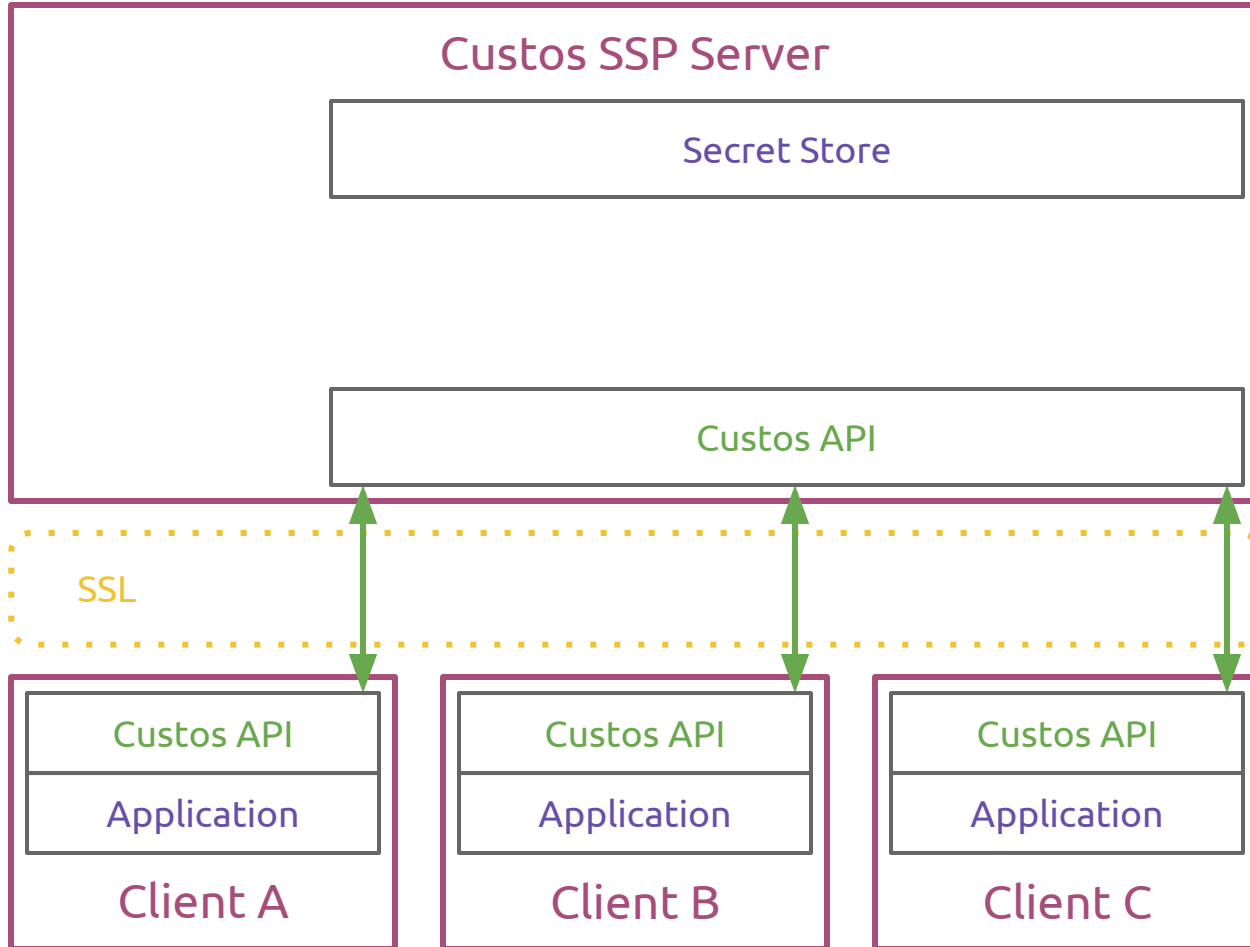
Custos SSP Server

Client A

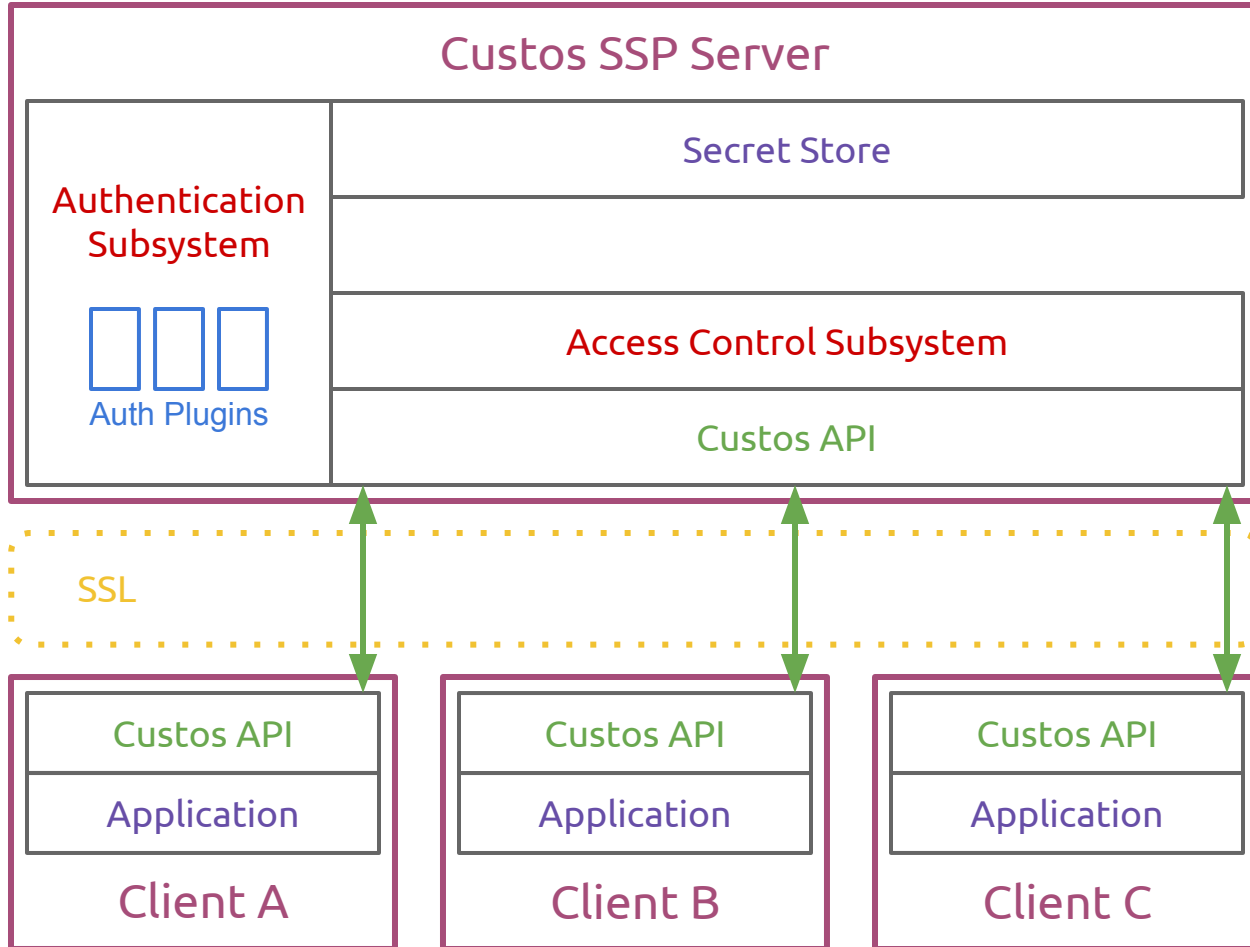
Client B

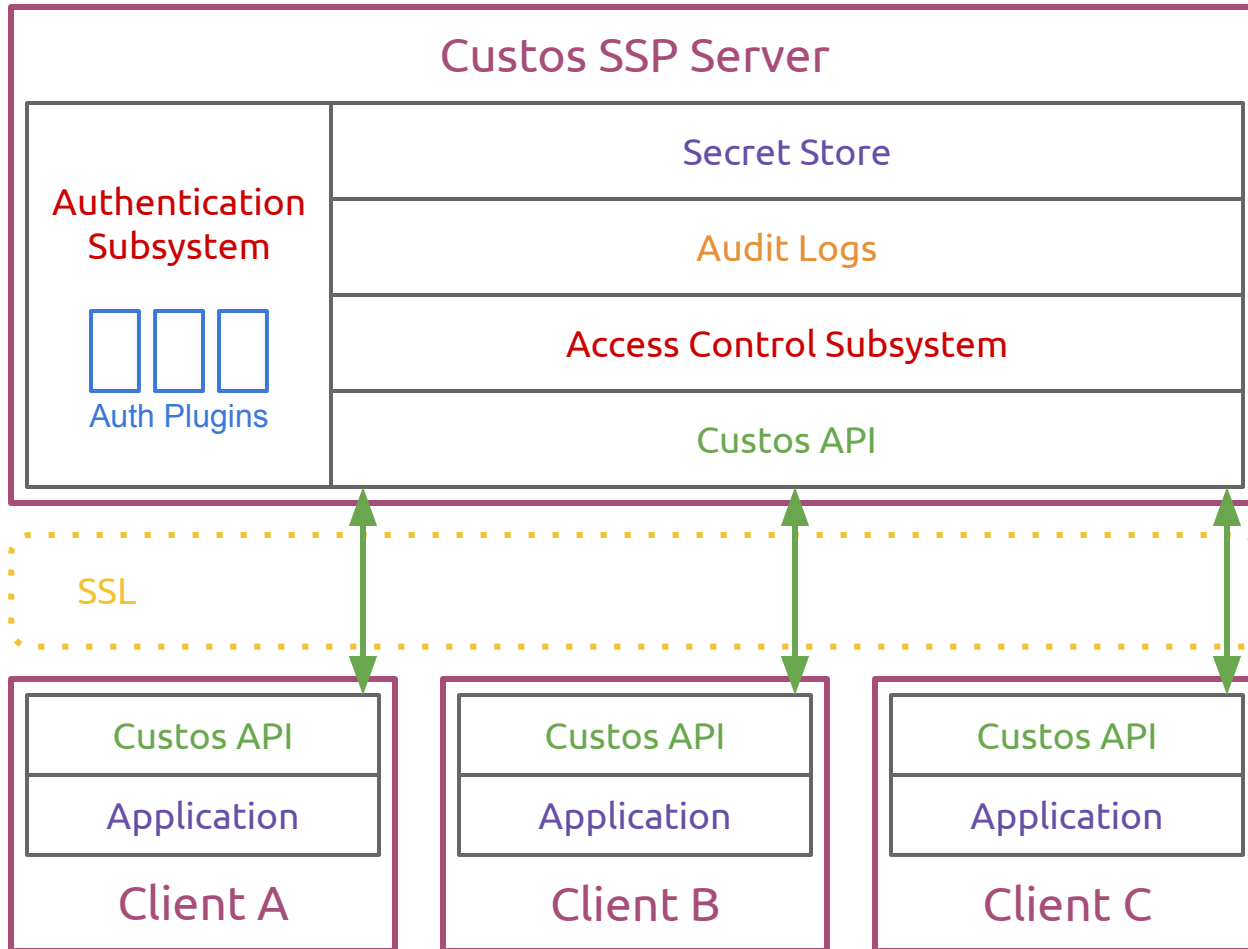
Client C











# Custos Challenges

# Custos Challenges

## 1. Complex Access Control System

# Custos Challenges

1. Complex Access Control System
  - a. Implicit Attributes (IP, ToD)

# Custos Challenges

1. Complex Access Control System
  - a. Implicit Attributes (IP, ToD)
  - b. Explicit Attributes (Passwords, Keys)

# Custos Challenges

1. Complex Access Control System
  - a. Implicit Attributes (IP, ToD)
  - b. Explicit Attributes (Passwords, Keys)
2. Lacks multi-SSP support primitives

# Custos Challenges

1. Complex Access Control System
  - a. Implicit Attributes (IP, ToD)
  - b. Explicit Attributes (Passwords, Keys)
2. Lacks multi-SSP support primitives
  - a. Requires mapping of IDs across SSPs



# Custos Challenges

1. Complex Access Control System
  - a. Implicit Attributes (IP, ToD)
  - b. Explicit Attributes (Passwords, Keys)
2. Lacks multi-SSP support primitives
  - a. Requires mapping of IDs across SSPs
  - b. Leaks explicit attributes

# SSaaS Applications (Chapter 7)

**Custos:** A First-Gen SSaaS Prototype  
(Chapter 8)

**Tutamen:** Next-Gen Secret Storage  
(Chapter 9)



Custos

Custos



# Custos

- Implicit vs Explicit Attr?
- No Multi-SSP Support

# VAULT

- Requires Trusted Server
- Lacks Out-of-Band Support
- Single Admin Domain

## Tutamen Contributions

---

Custos

- Implicit vs Explicit Attr?
- No Multi-SSP Support



- Requires Trusted Server
- Lacks Out-of-Band Support
- Single Admin Domain

+ Loosely Coupled, Internet-Scale Operation

## Tutamen Contributions

---

Custos

- Implicit vs Explicit Attr?
- No Multi-SSP Support

 VAULT

- Requires Trusted Server
- Lacks Out-of-Band Support
- Single Admin Domain



- + Out-of-Band and Automated Authentication
- + Loosely Coupled, Internet-Scale Operation

## Tutamen Contributions

---

Custos

- Implicit vs Explicit Attr?
- No Multi-SSP Support

 VAULT

- Requires Trusted Server
- Lacks Out-of-Band Support
- Single Admin Domain

- + Avoid a Single Trusted Entity
- + Out-of-Band and Automated Authentication
- + Loosely Coupled, Internet-Scale Operation

## Tutamen Contributions

---

Custos



- Implicit vs Explicit Attr?
- No Multi-SSP Support
- Requires Trusted Server
- Lacks Out-of-Band Support
- Single Admin Domain

# Tutamen Architecture



# Storage Server

Storage Server

Access Control Server

Storage Server

Access Control Server

Application

Storage Server

Access Control Server

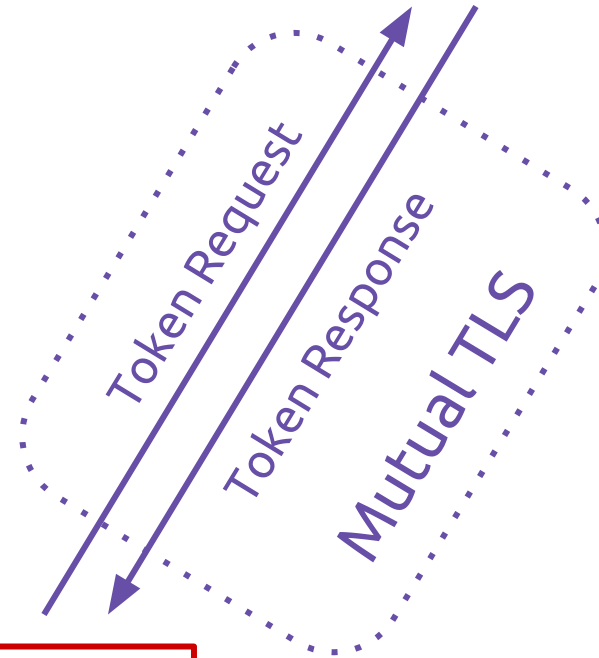
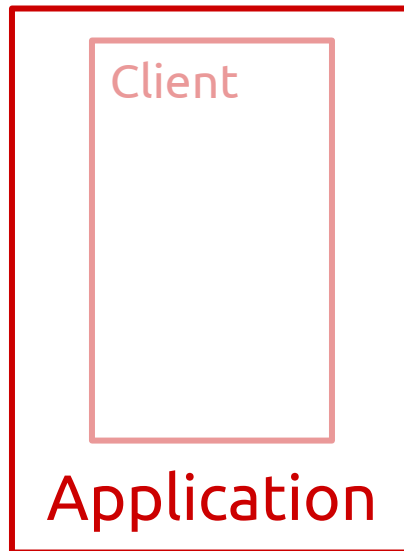
Client

Application



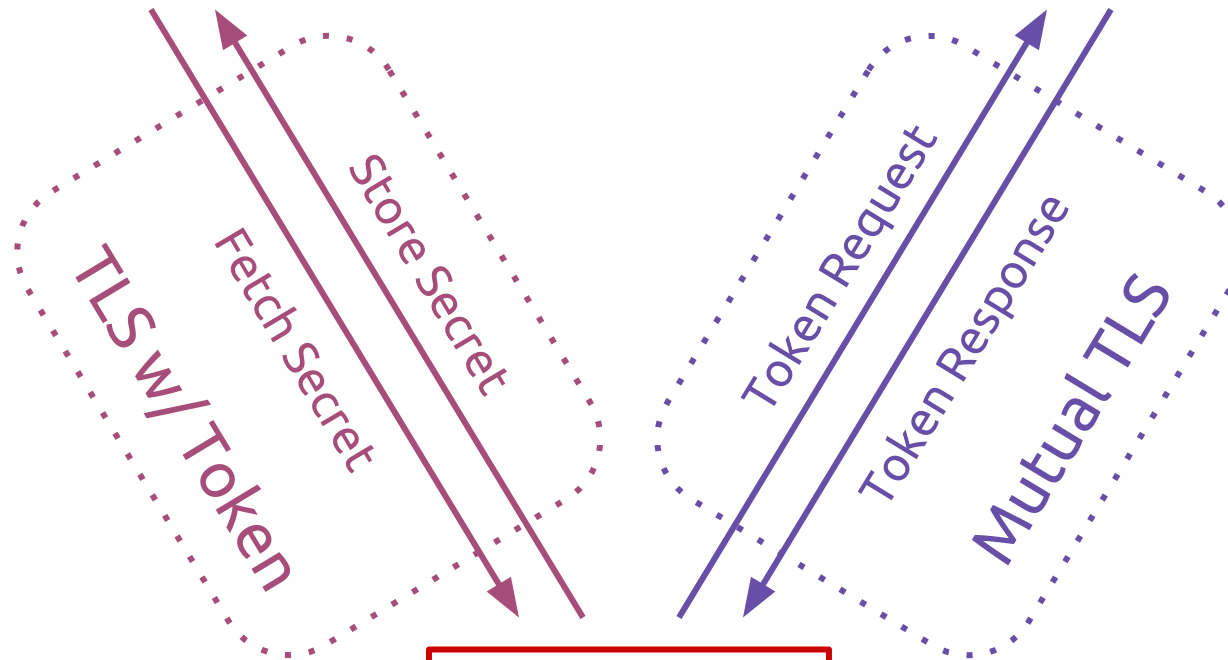
Storage Server

Access Control Server



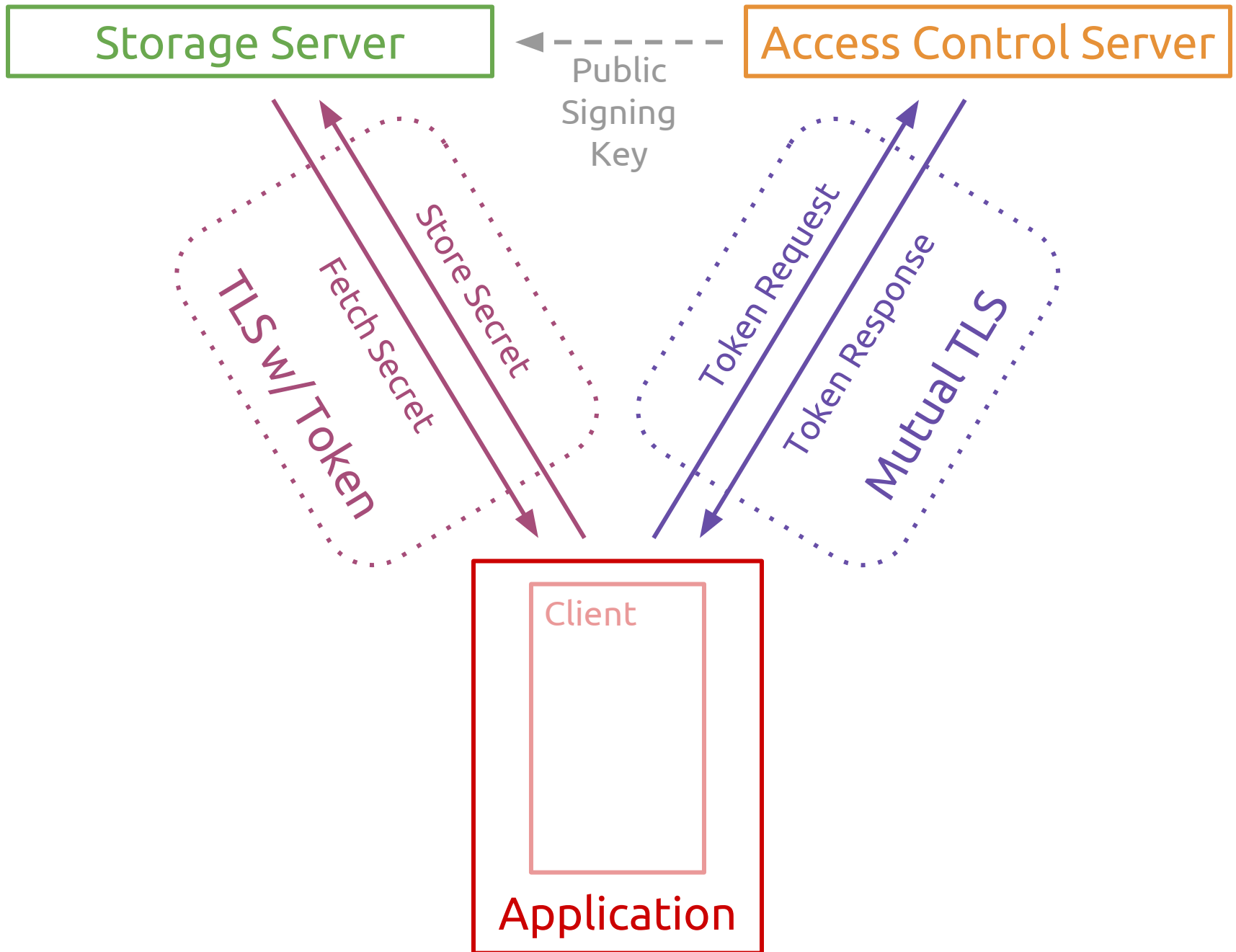
Storage Server

Access Control Server



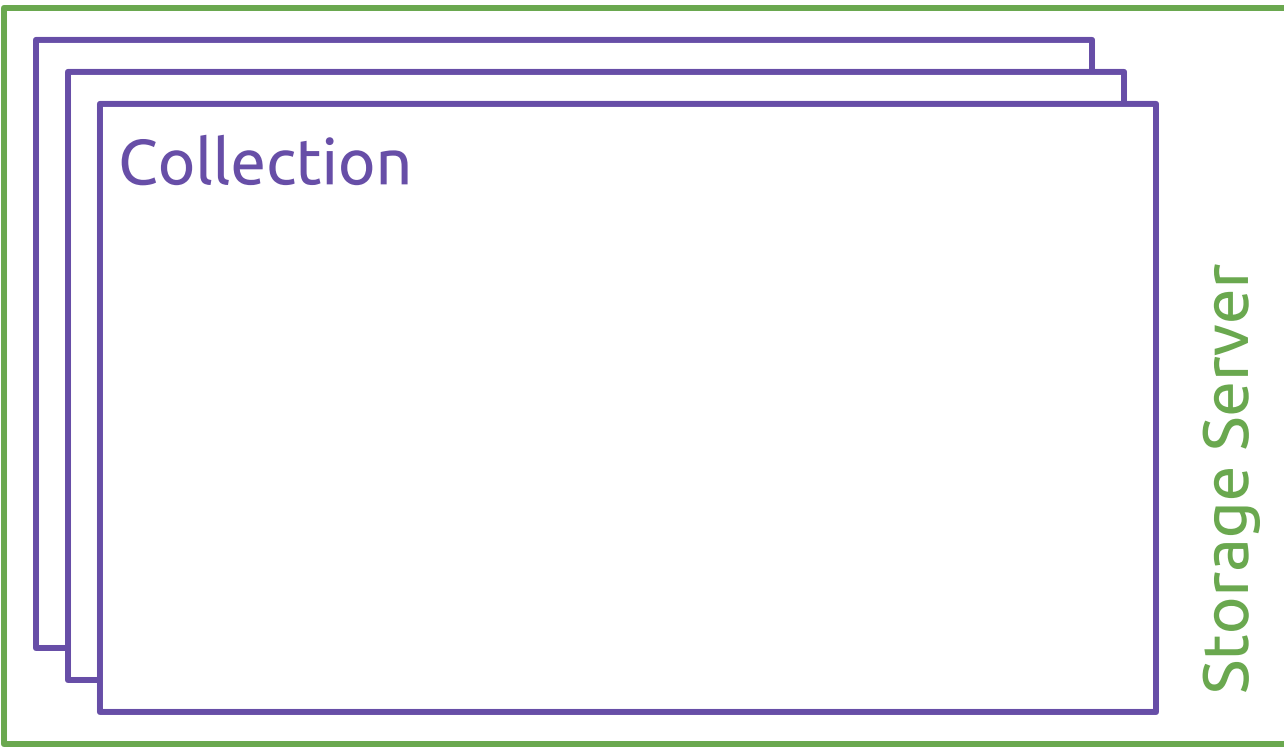
Client

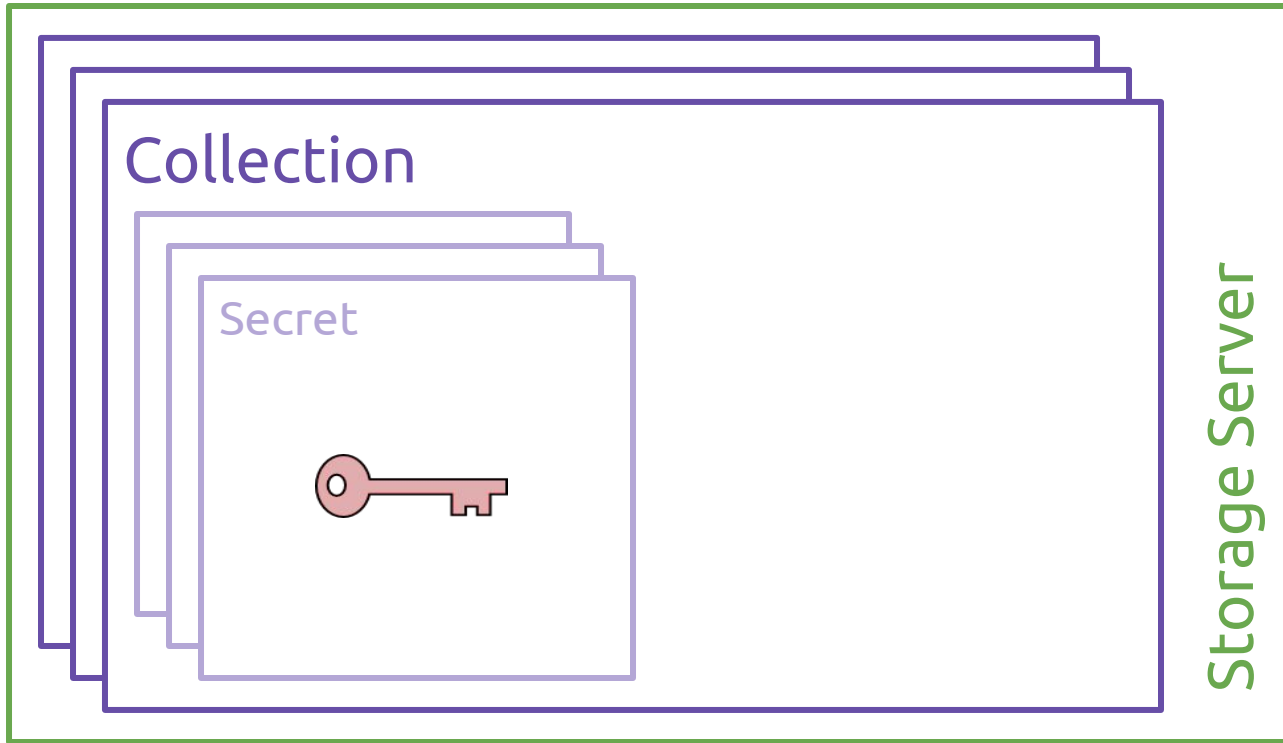
Application

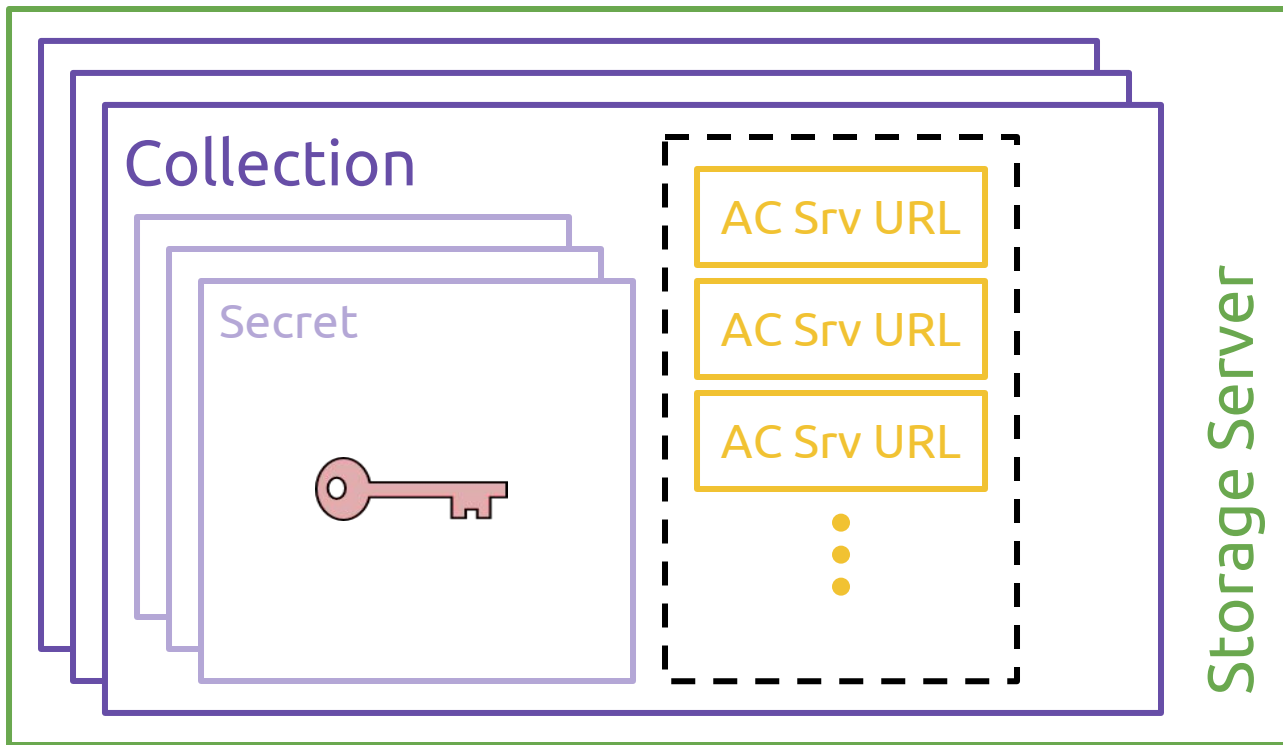




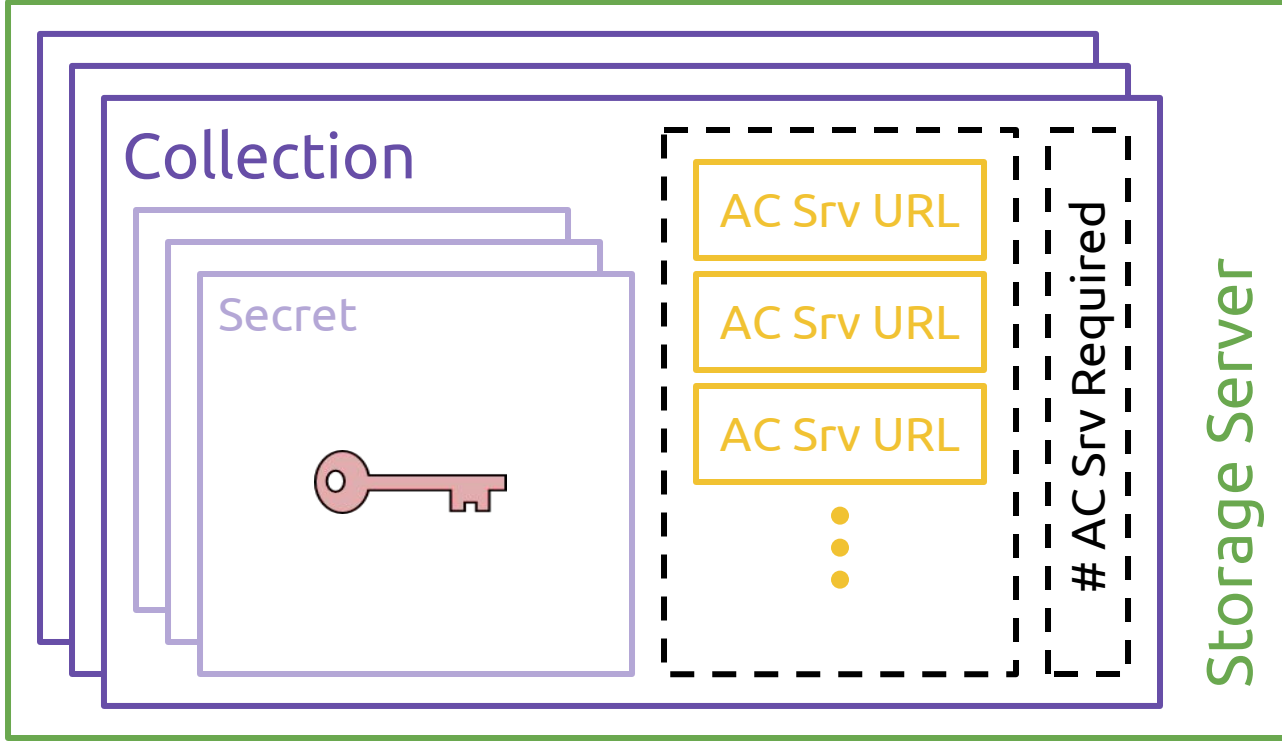
Storage Server













# Access Control Server

# Access Control Server



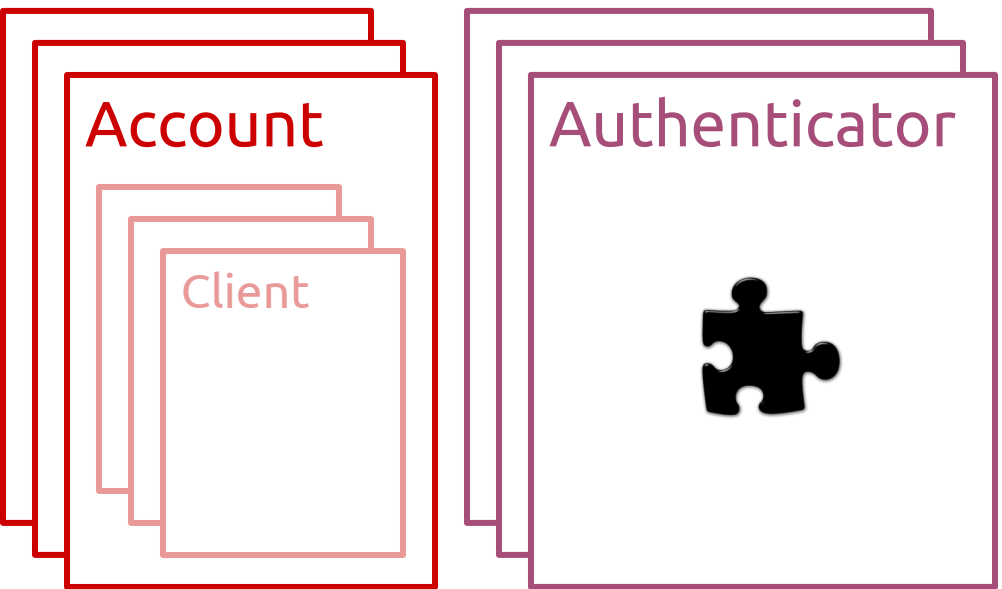
Account

The diagram shows a large, empty rectangular box representing the Access Control Server. To its left is a stack of three overlapping rectangular boxes, each representing an account file. The topmost box is labeled 'Account' in red text. The boxes are outlined in red, and the entire diagram is enclosed within a thin orange border.

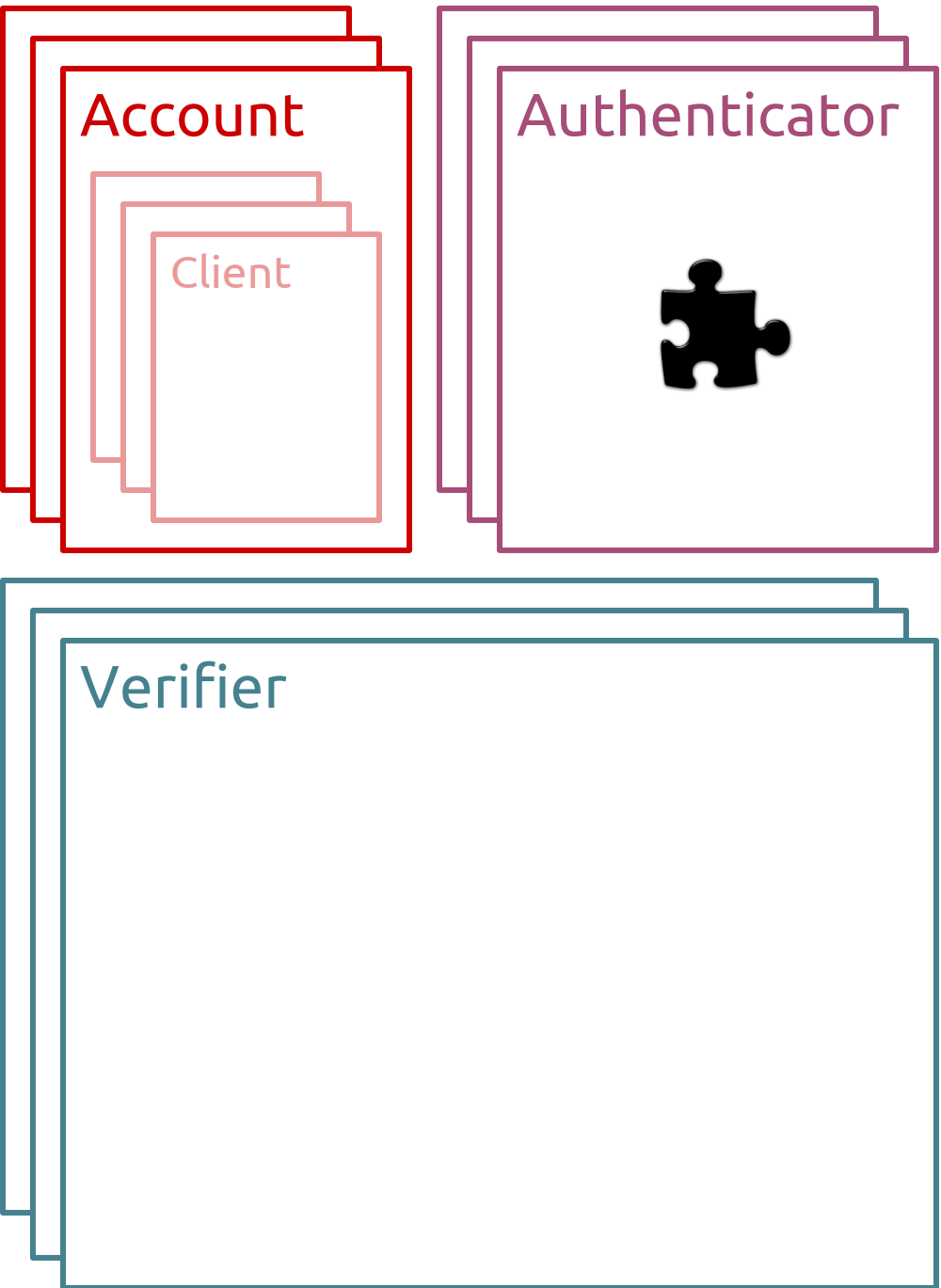
# Access Control Server



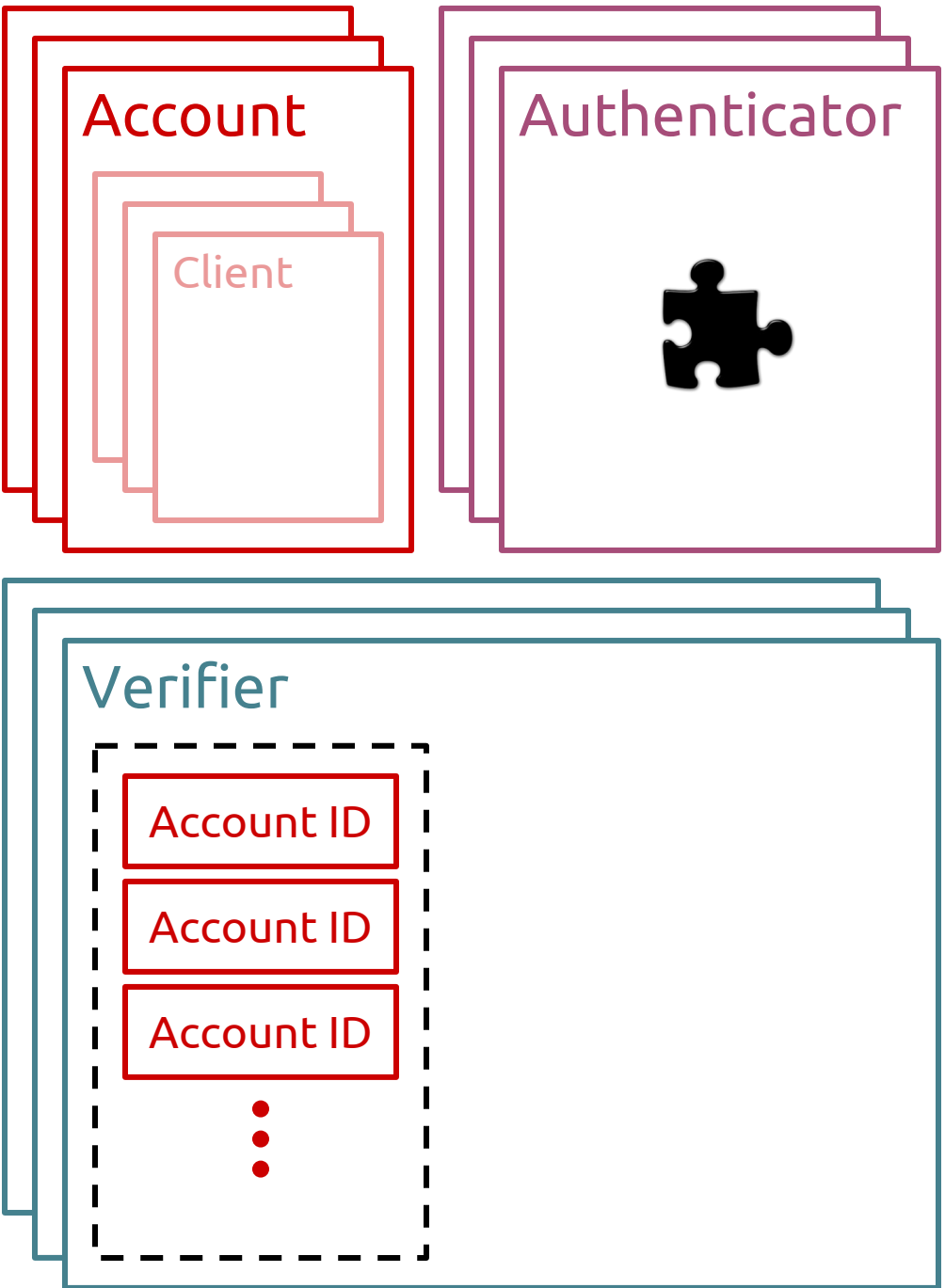
# Access Control Server



# Access Control Server

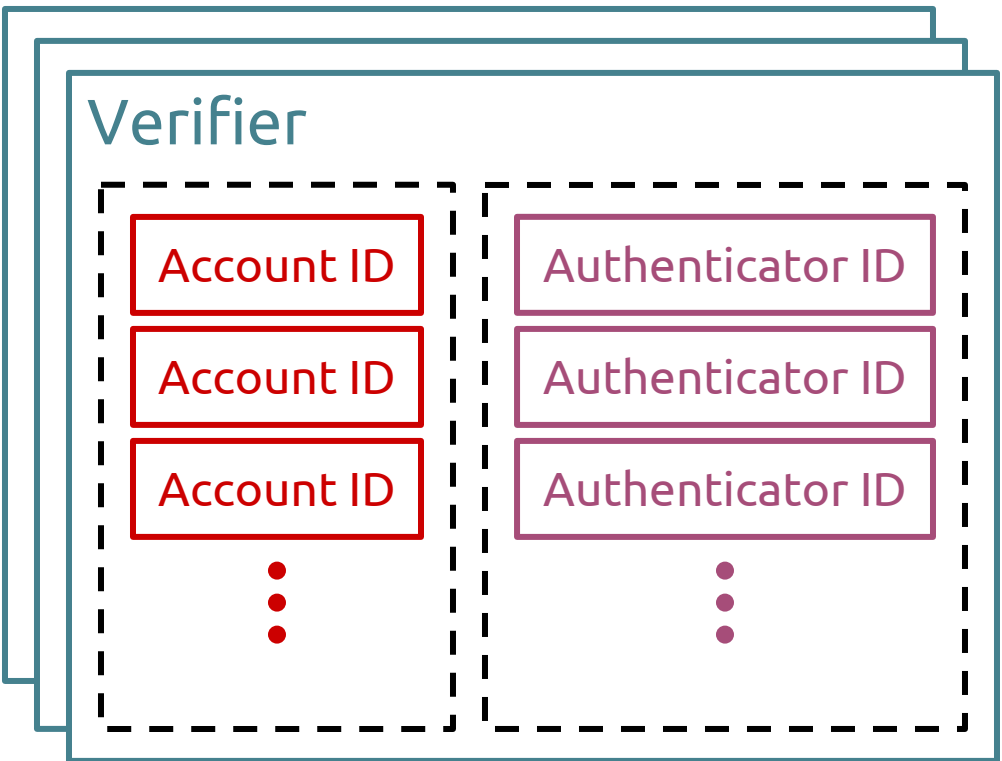
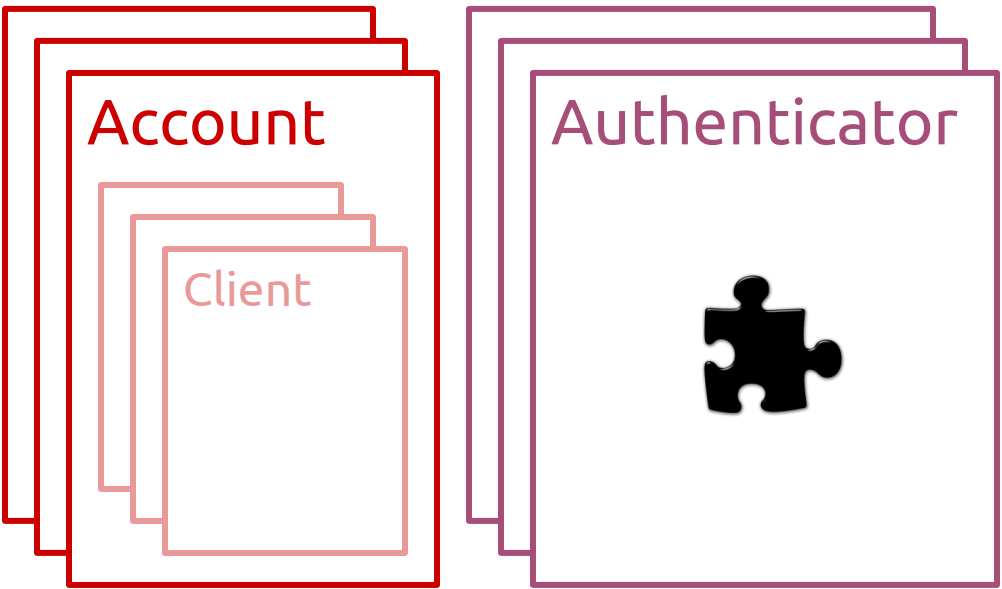


# Access Control Server





# Access Control Server



# Access Control Server

Account

Client

Authenticator



Permissions

Object Type

Object ID

Permission Name

Verifier

Account ID

Account ID

Account ID



Authenticator ID

Authenticator ID

Authenticator ID



# Access Control Server

Account

Client

Authenticator



Permissions

Object Type

Object ID

Verifier

Account ID

Account ID

Account ID



Authenticator ID

Authenticator ID

Authenticator ID



Permission Name

Verifier ID

Verifier ID

Verifier ID



# Tutamen Multi-SSP Operation



Application



Secret

Application

AC Server A

AC Server B

Secret

Application

Storage Server A

Storage Server B

Storage Server C

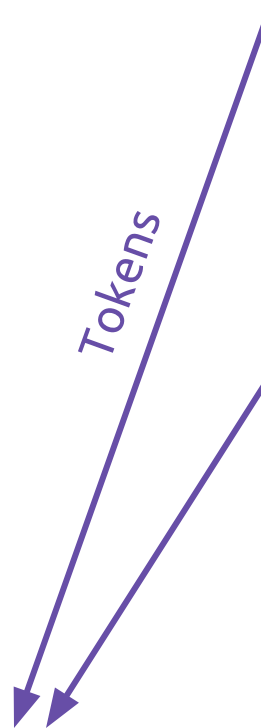
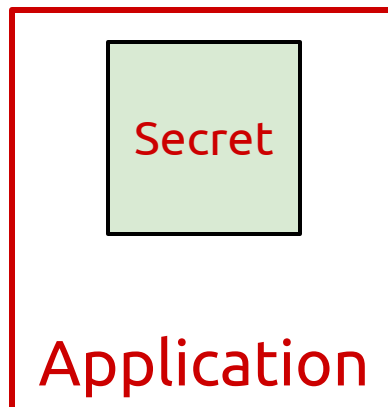
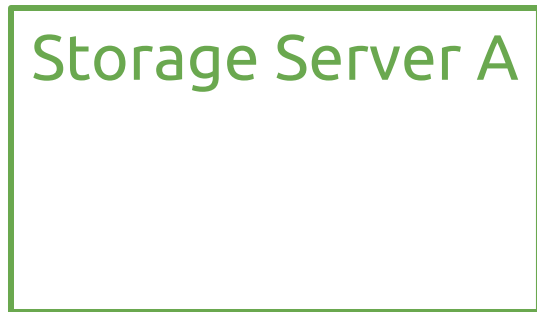
AC Server A

AC Server B

Secret

Application





Storage Server A

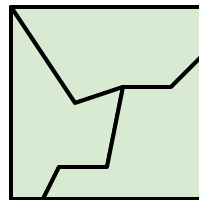
Storage Server B

Storage Server C

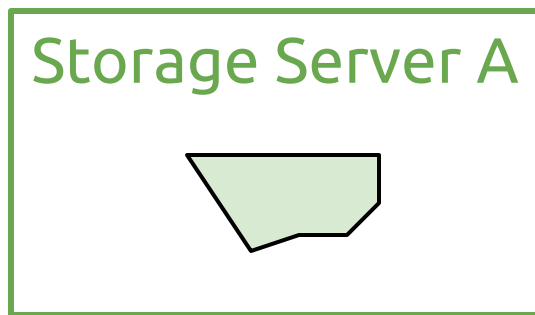
AC Server A

AC Server B

*Tokens*



Application



Public Signing Keys



A dashed gray arrow pointing from AC Server A to Storage Server A. The text "Public Signing Keys" is written in gray above the arrow.



Secret Shards + Tokens



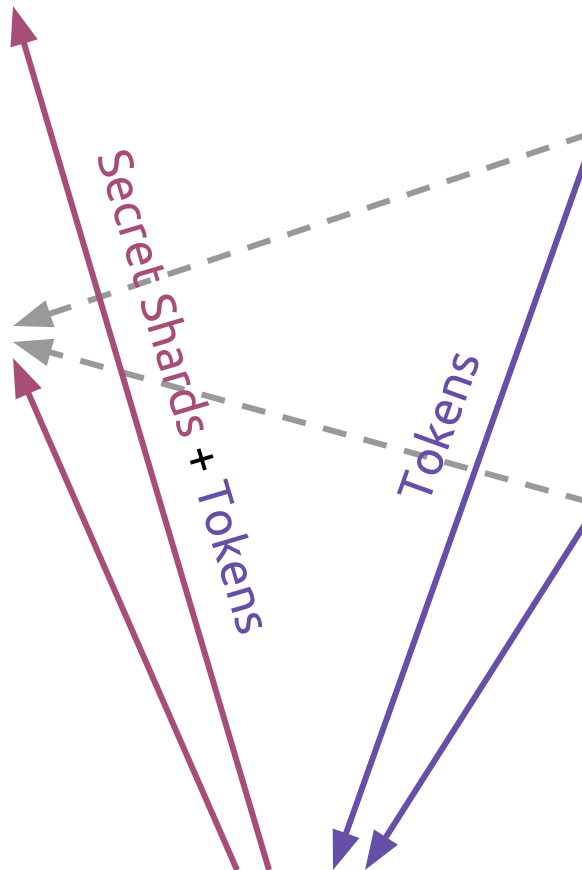
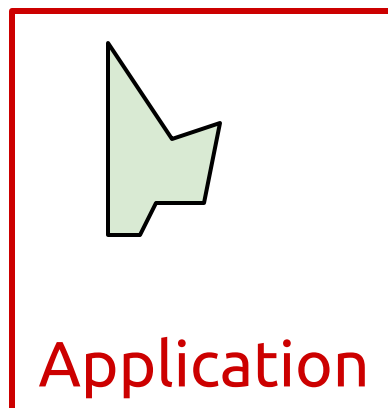
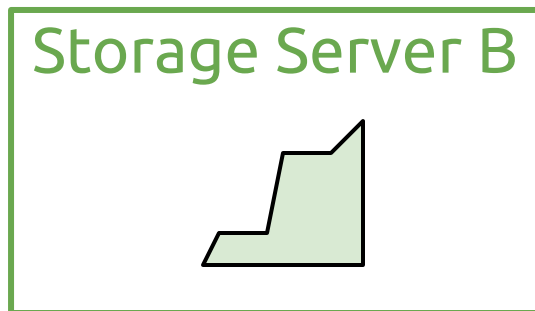
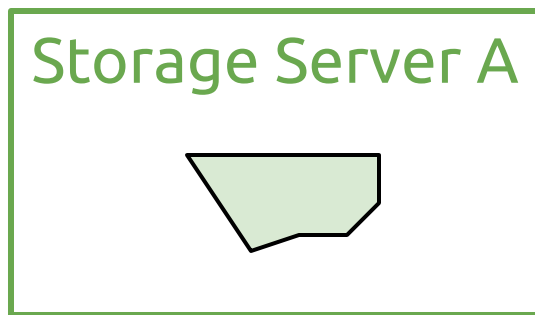
A solid purple arrow pointing from the Application to Storage Server A. The text "Secret Shards + Tokens" is written in purple along the arrow.

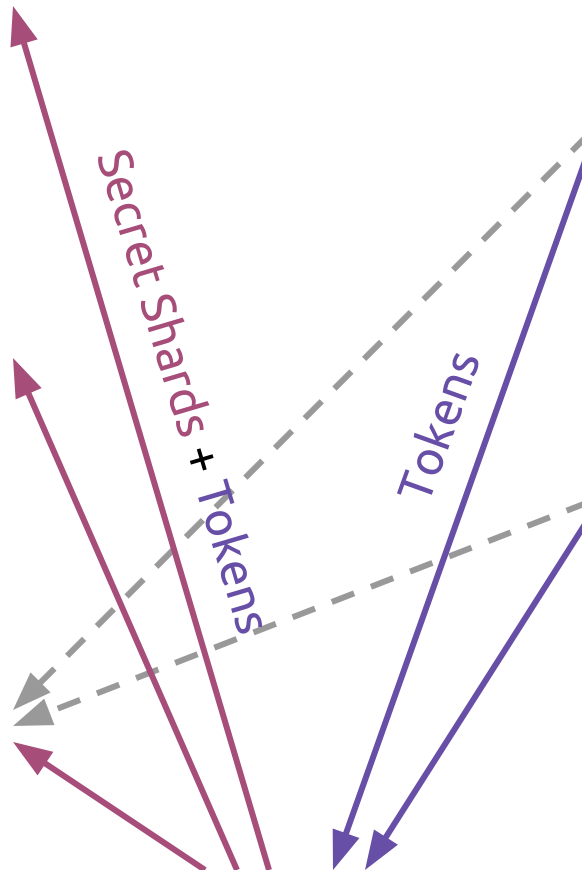
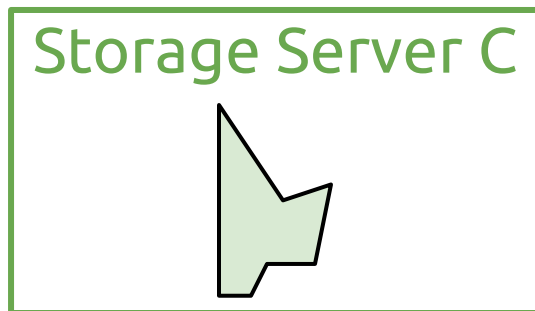
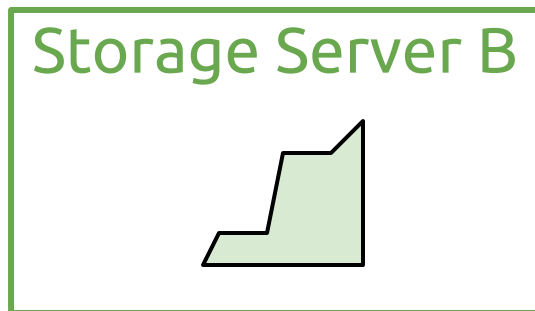
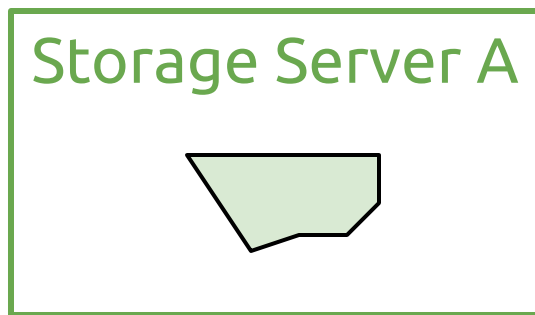
Tokens



A solid purple arrow pointing from AC Server A to the Application. The text "Tokens" is written in purple above the arrow.







# Tutamen Secret Retrieval

# Tutamen Secret Retrieval

## w/ Out of Band Human-in-the-Loop





```
Permissions for Collection cf3529eb13be:  
  { read: [ Verifier a74b2e2d493d ] }
```

Permissions for Collection cf3529eb13be:

```
{ read: [ Verifier a74b2e2d493d ] }
```

Verifier a74b2e2d493d

```
{ Accounts: [ Account cceb832edcdb ] }
```

```
  Authenticators: [ Authenticator 34e85e1bb264 ] }
```

Permissions for Collection cf3529eb13be:

```
{ read: [ Verifier a74b2e2d493d ] }
```

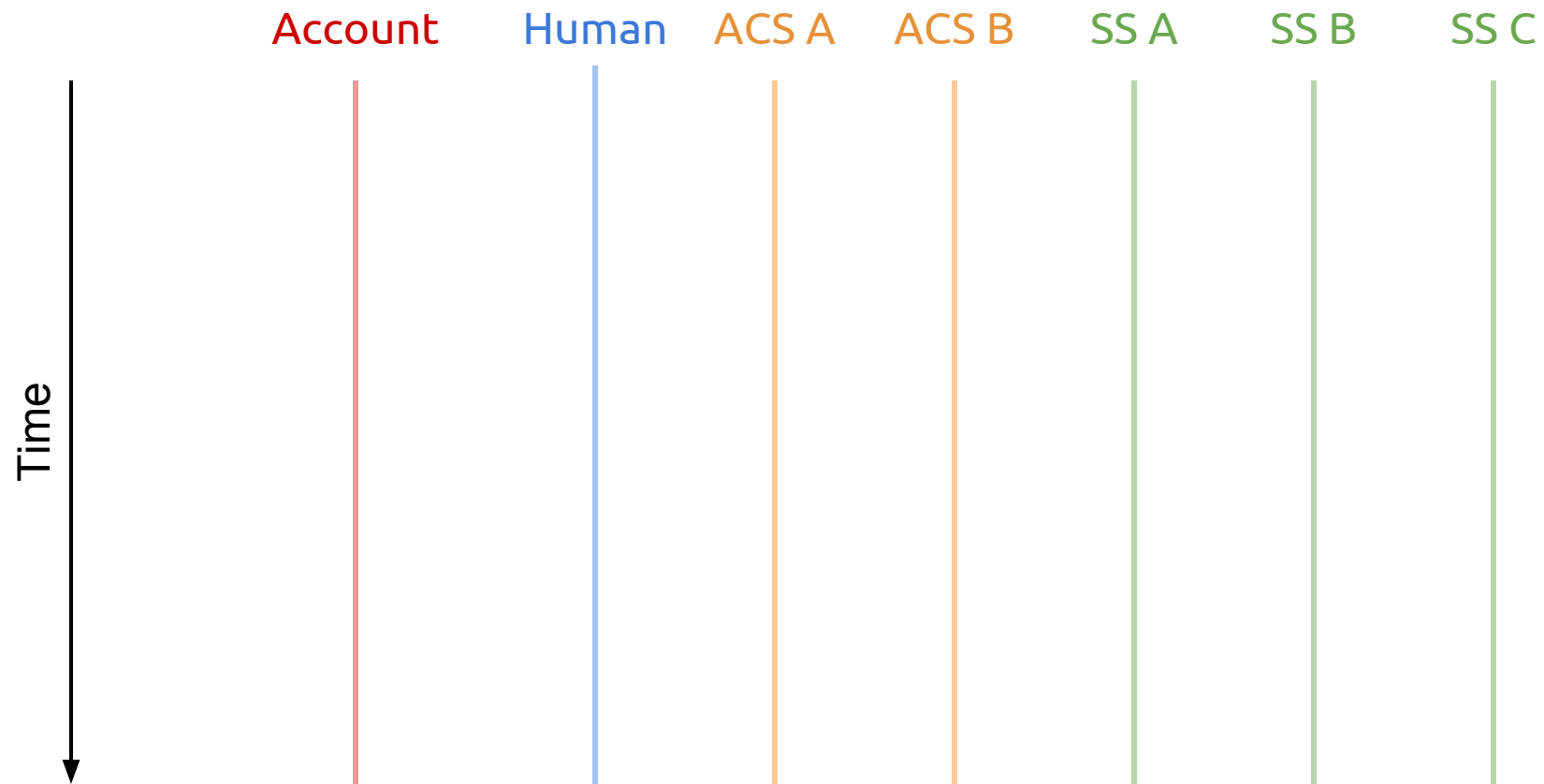
Verifier a74b2e2d493d

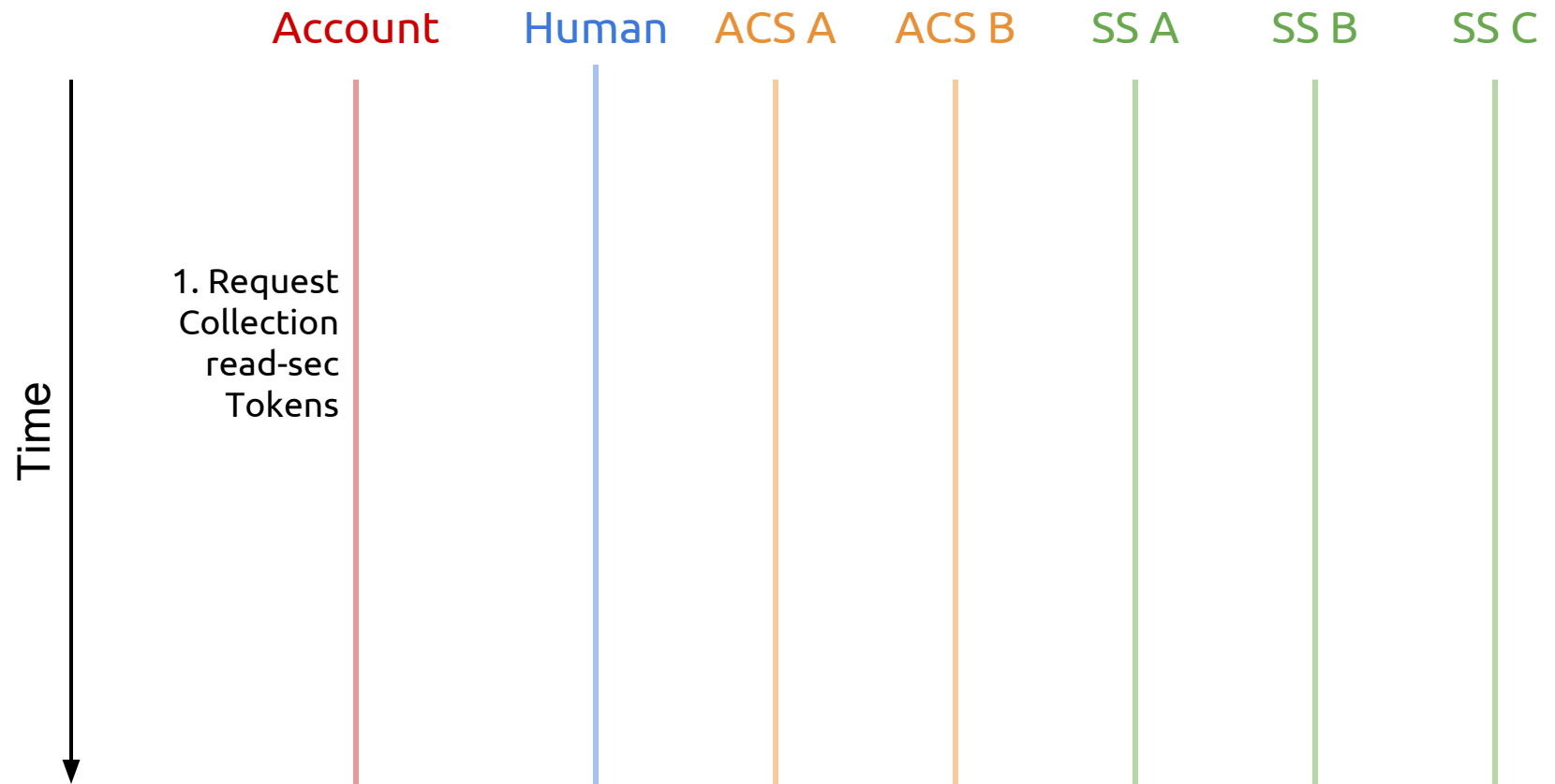
```
{ Accounts: [ Account cceb832edcdb ] }
```

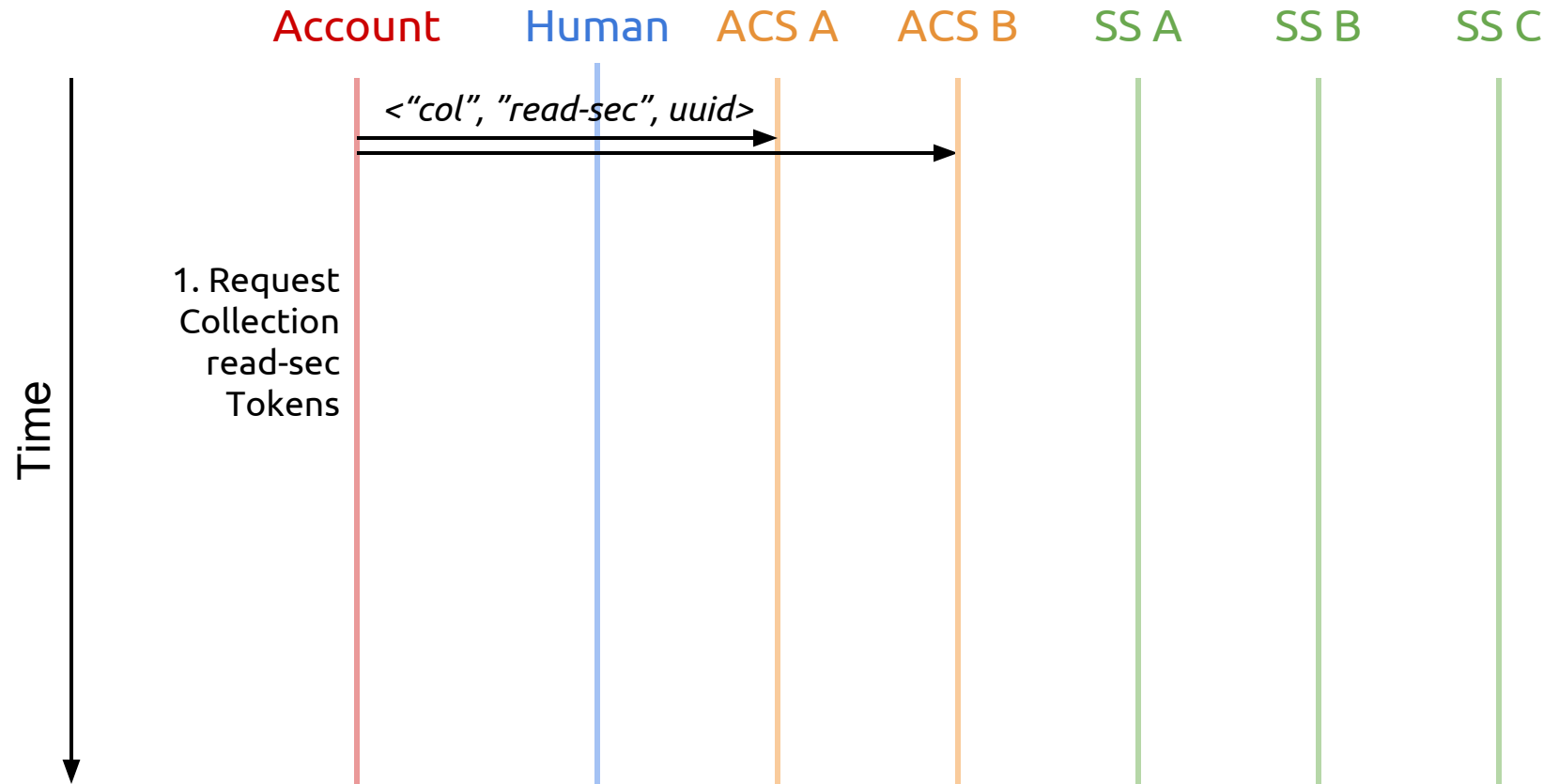
```
  Authenticators: [ Authenticator 34e85e1bb264 ] }
```

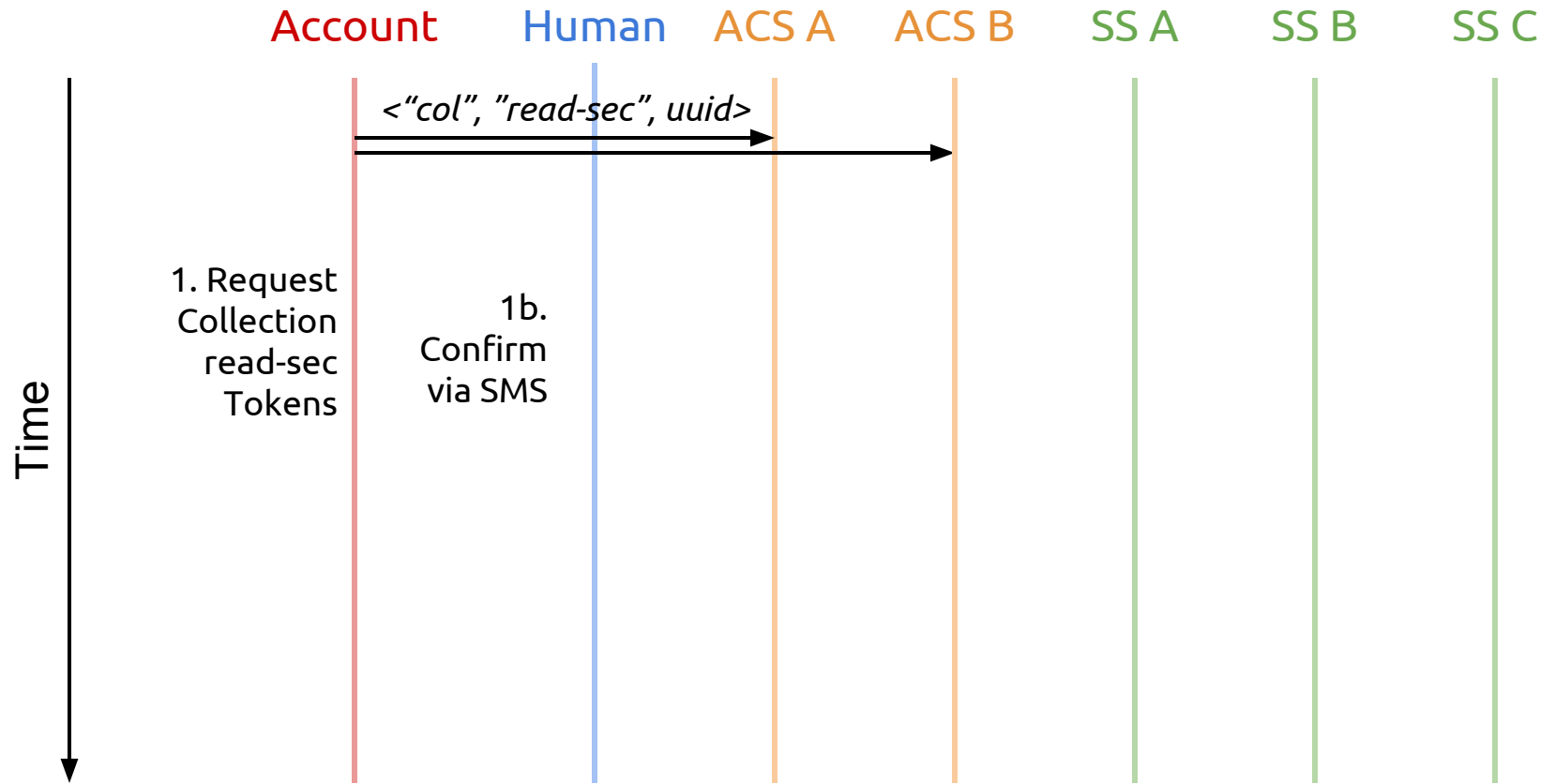
Authenticator 34e85e1bb264

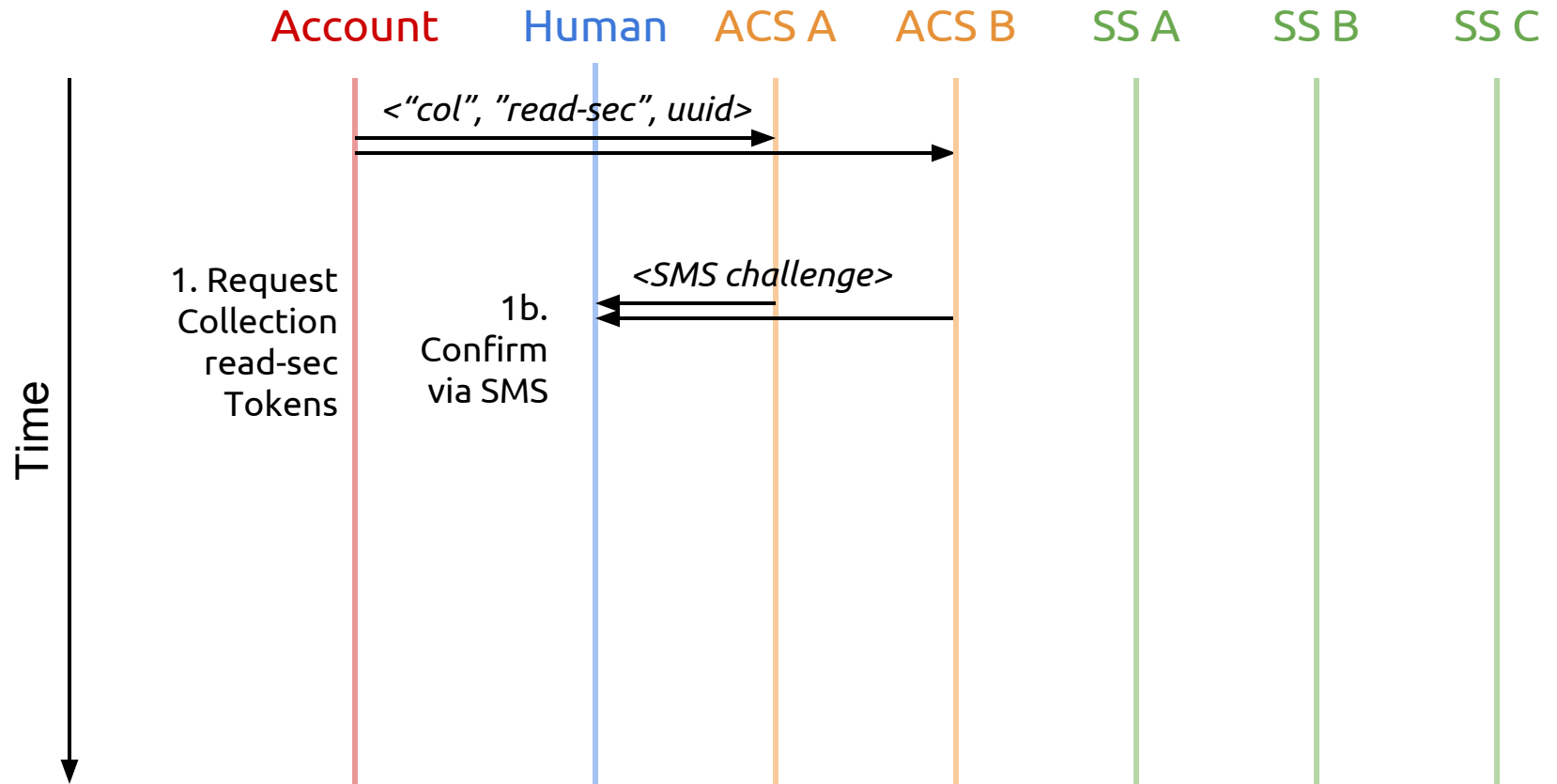
```
{ Plugin: SMS Challenge/Response }
```



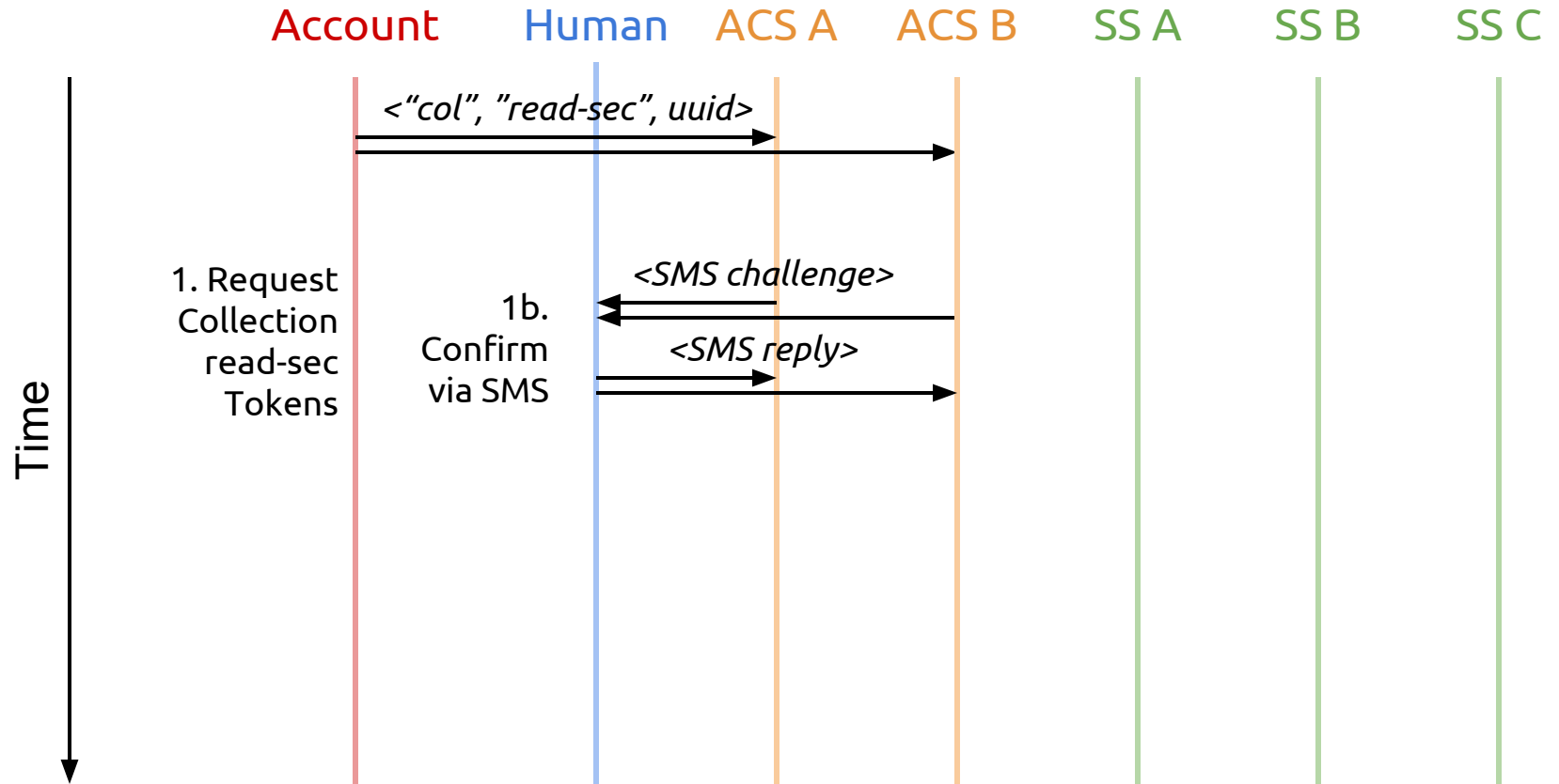


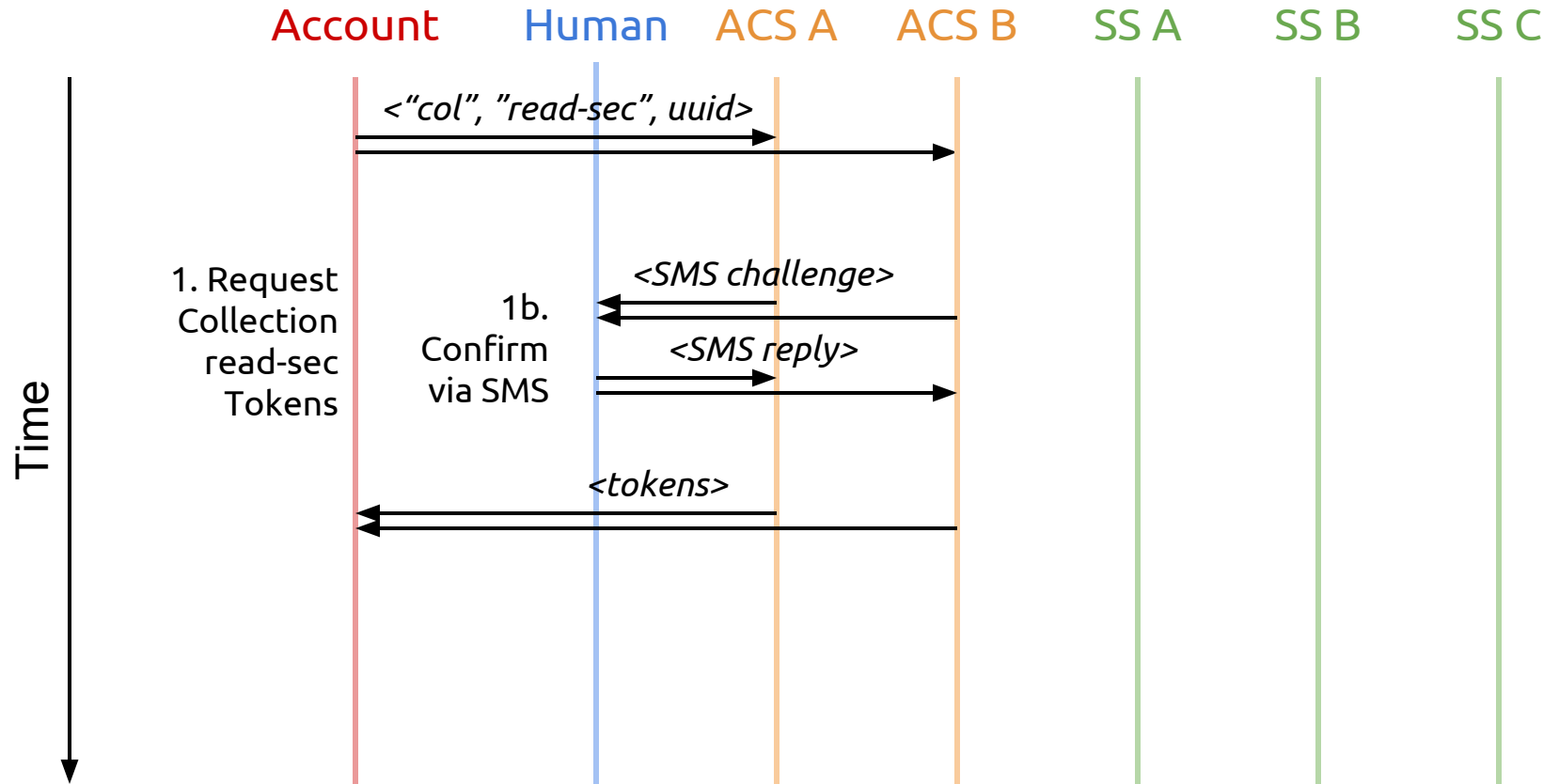


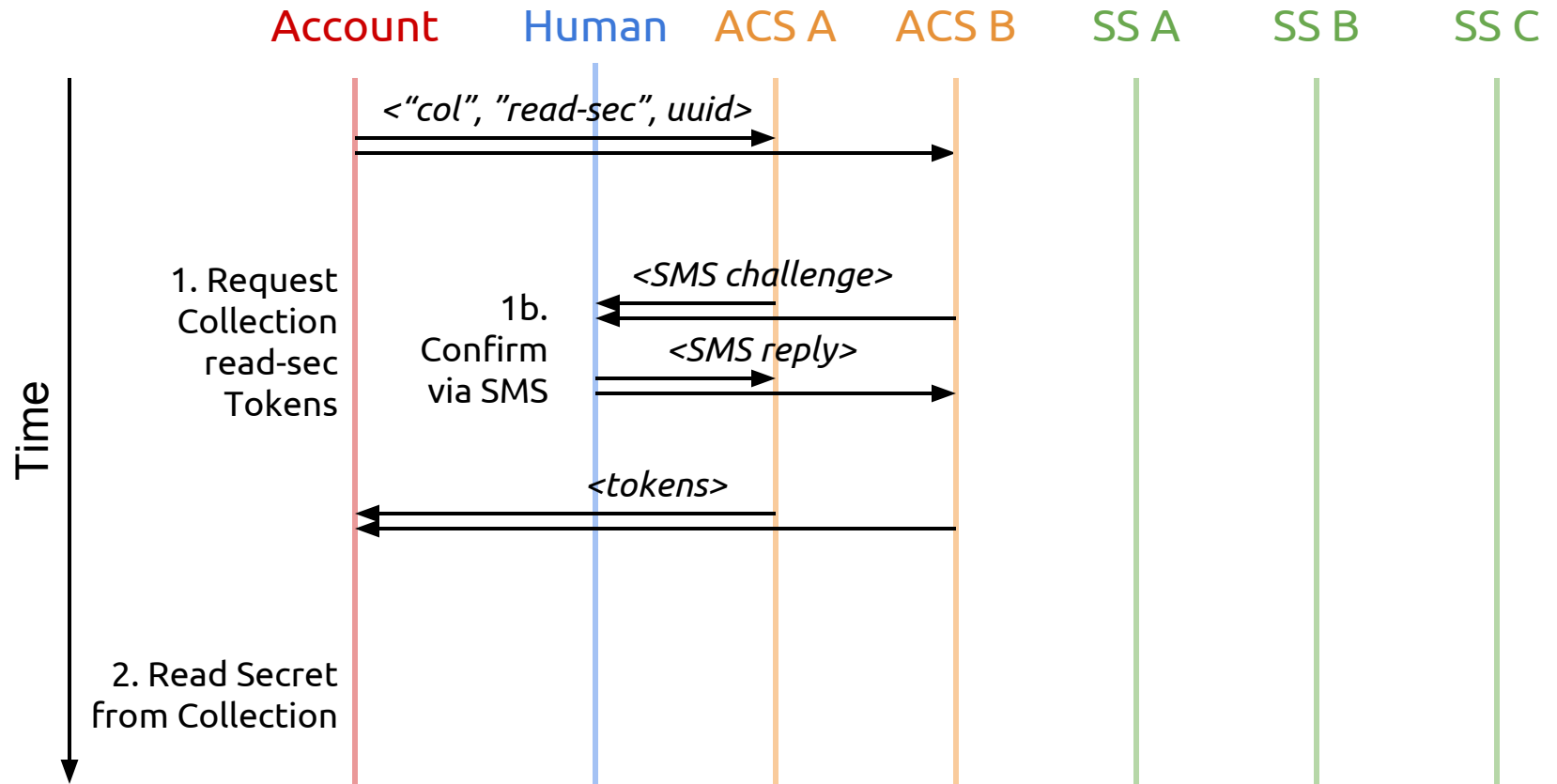


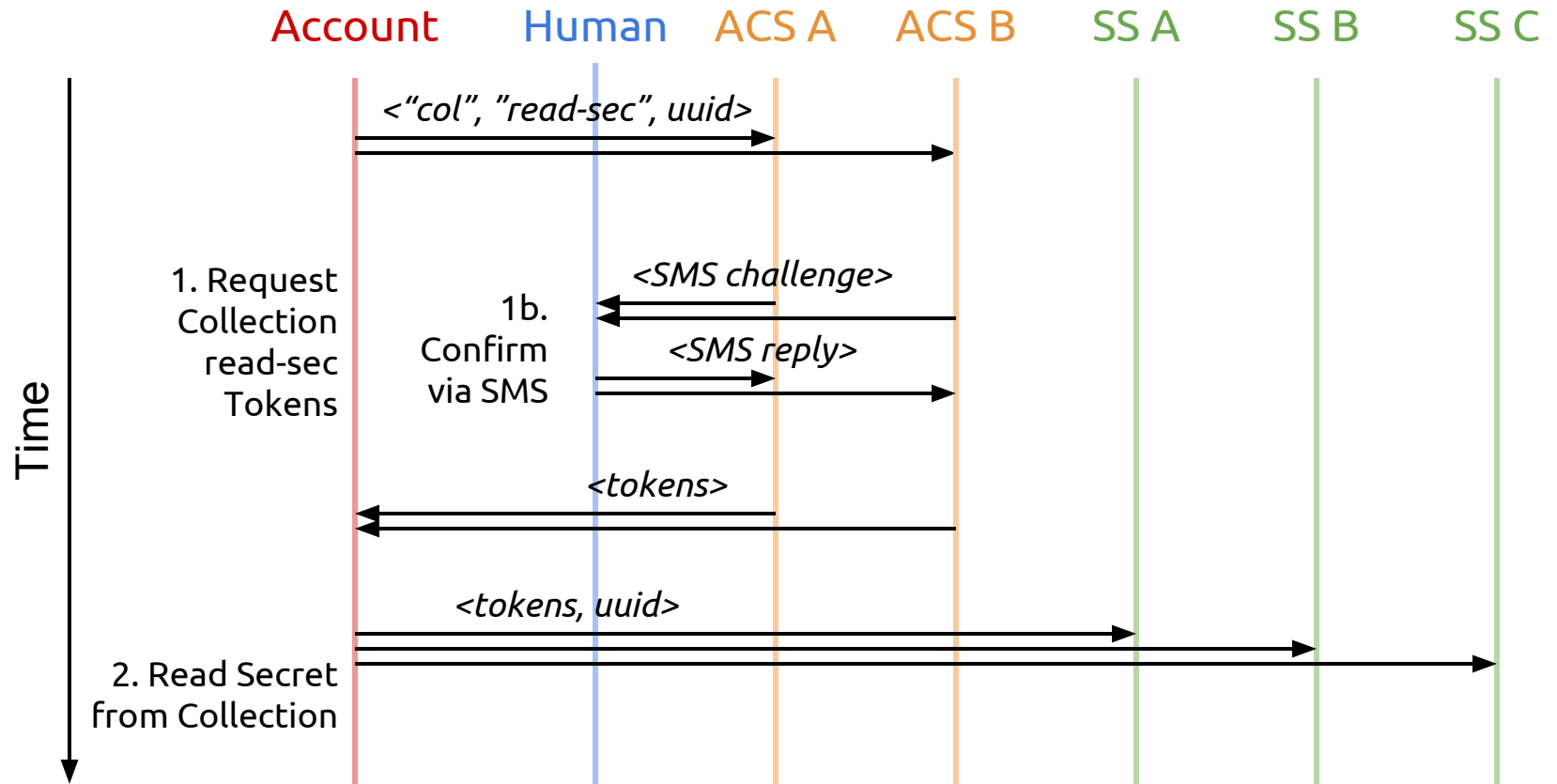


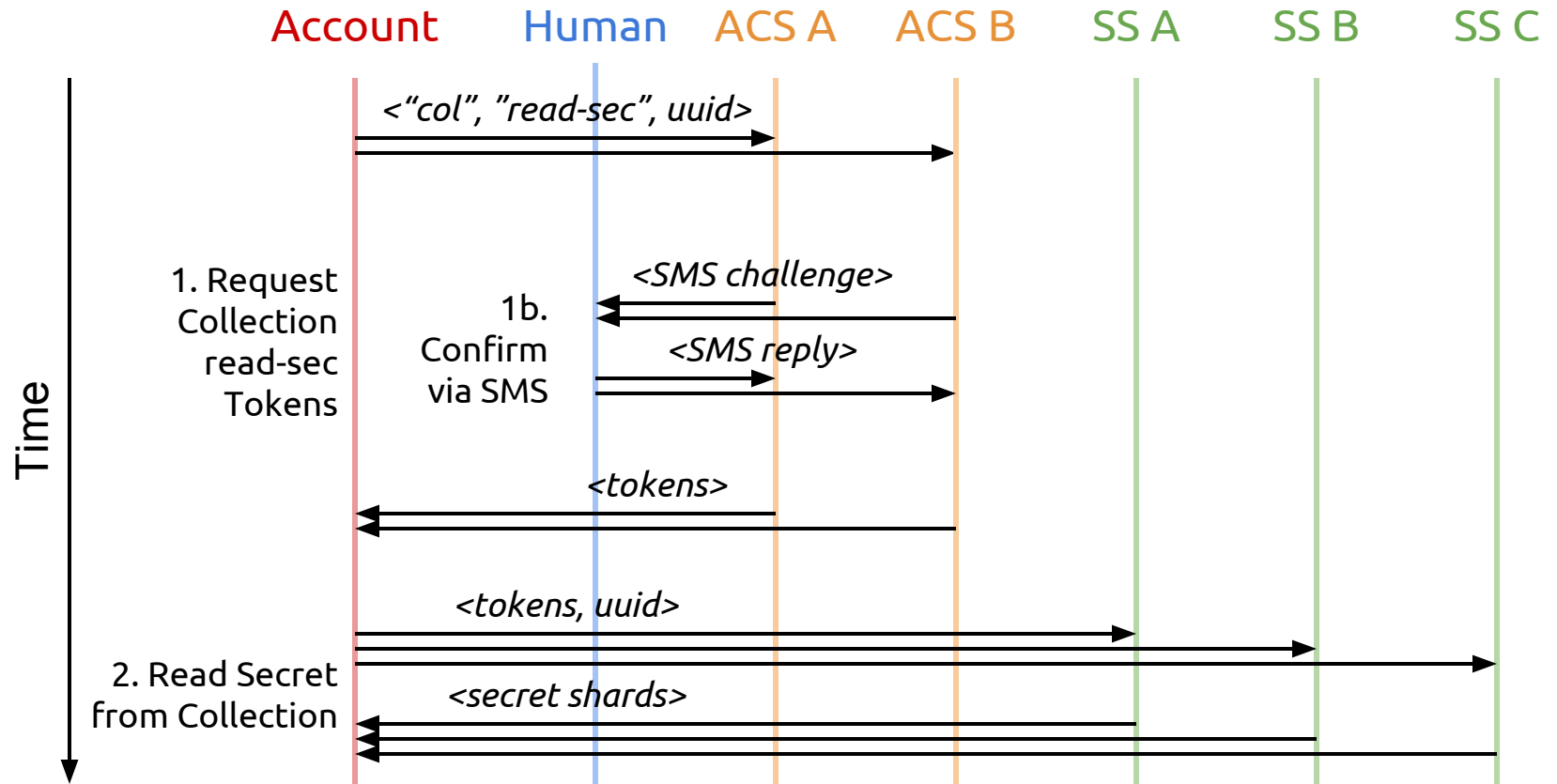












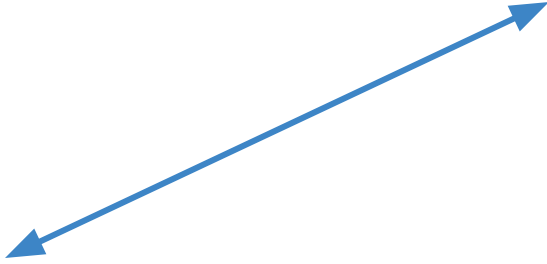
# Tutamen Applications

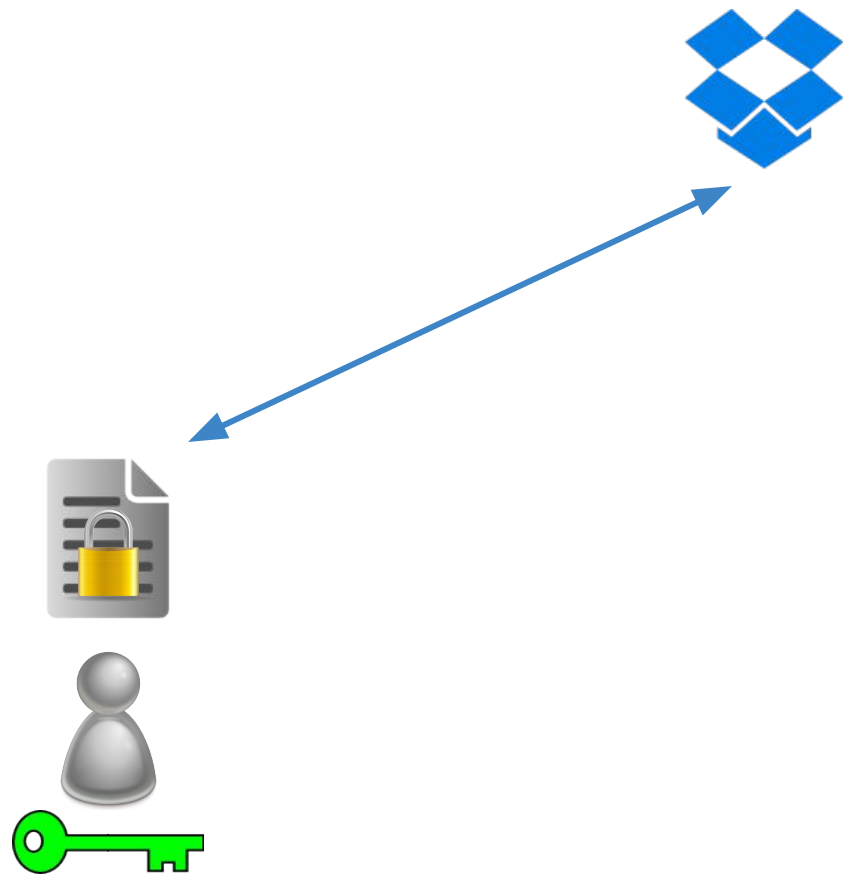
# Fusebox: Tutamen-backed Dropbox Client

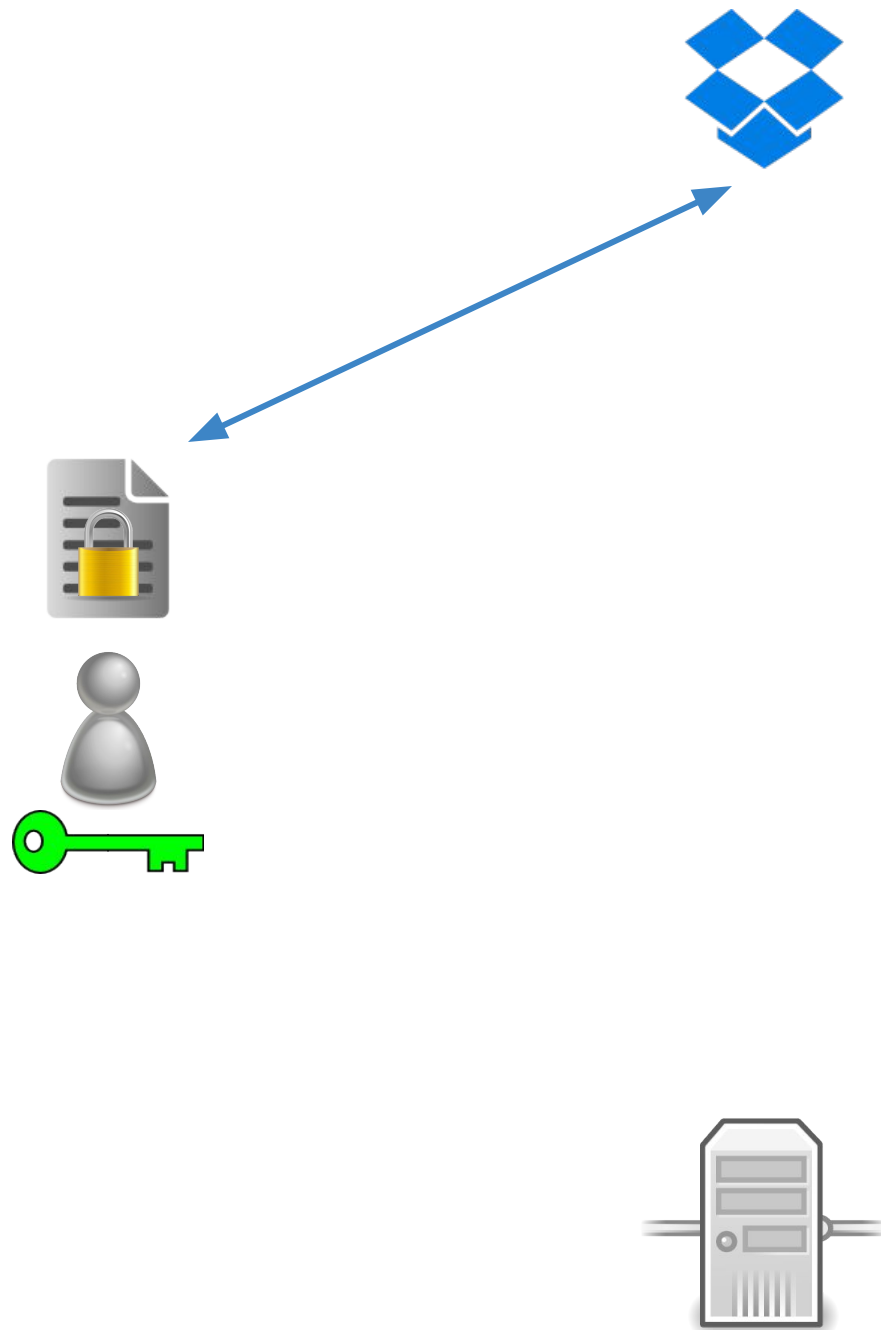
Implementation by Taylor Andrews

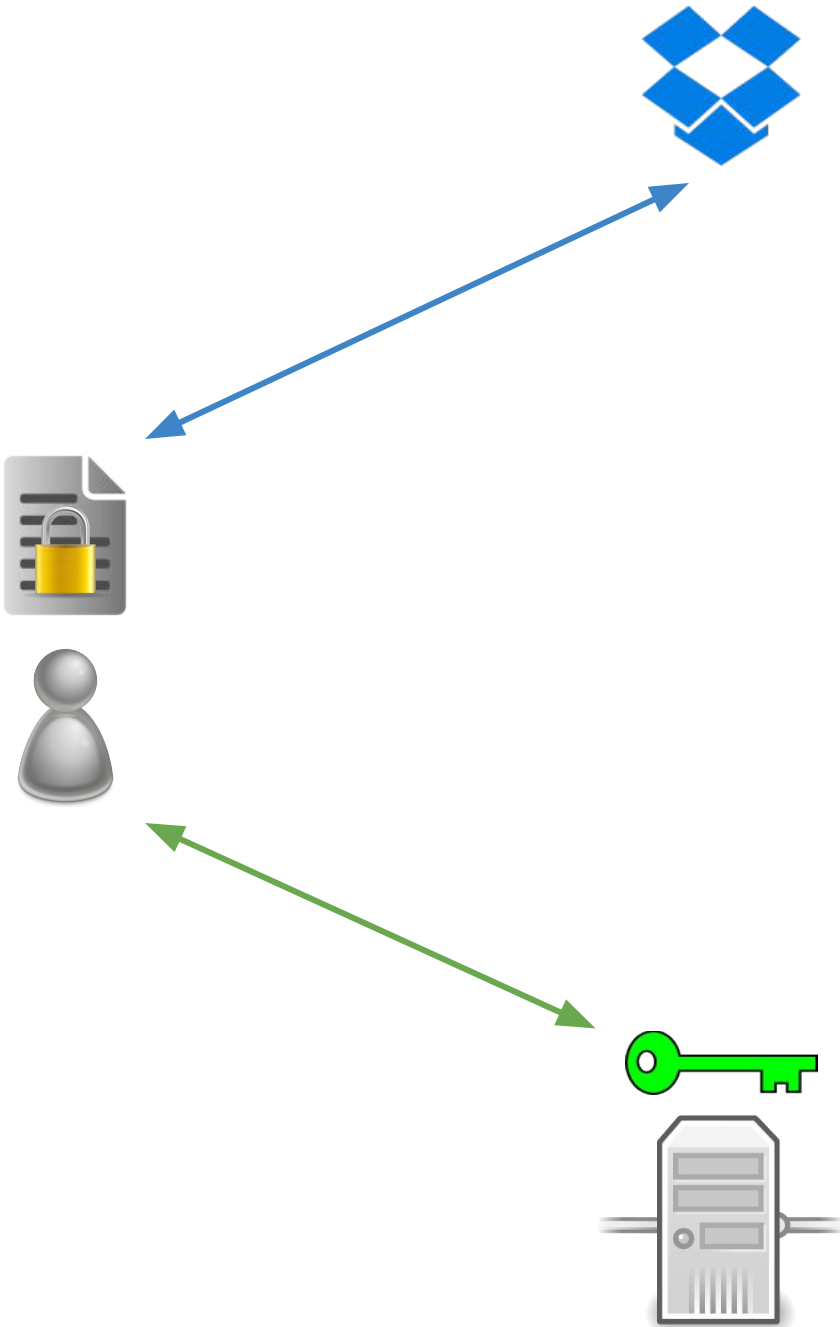


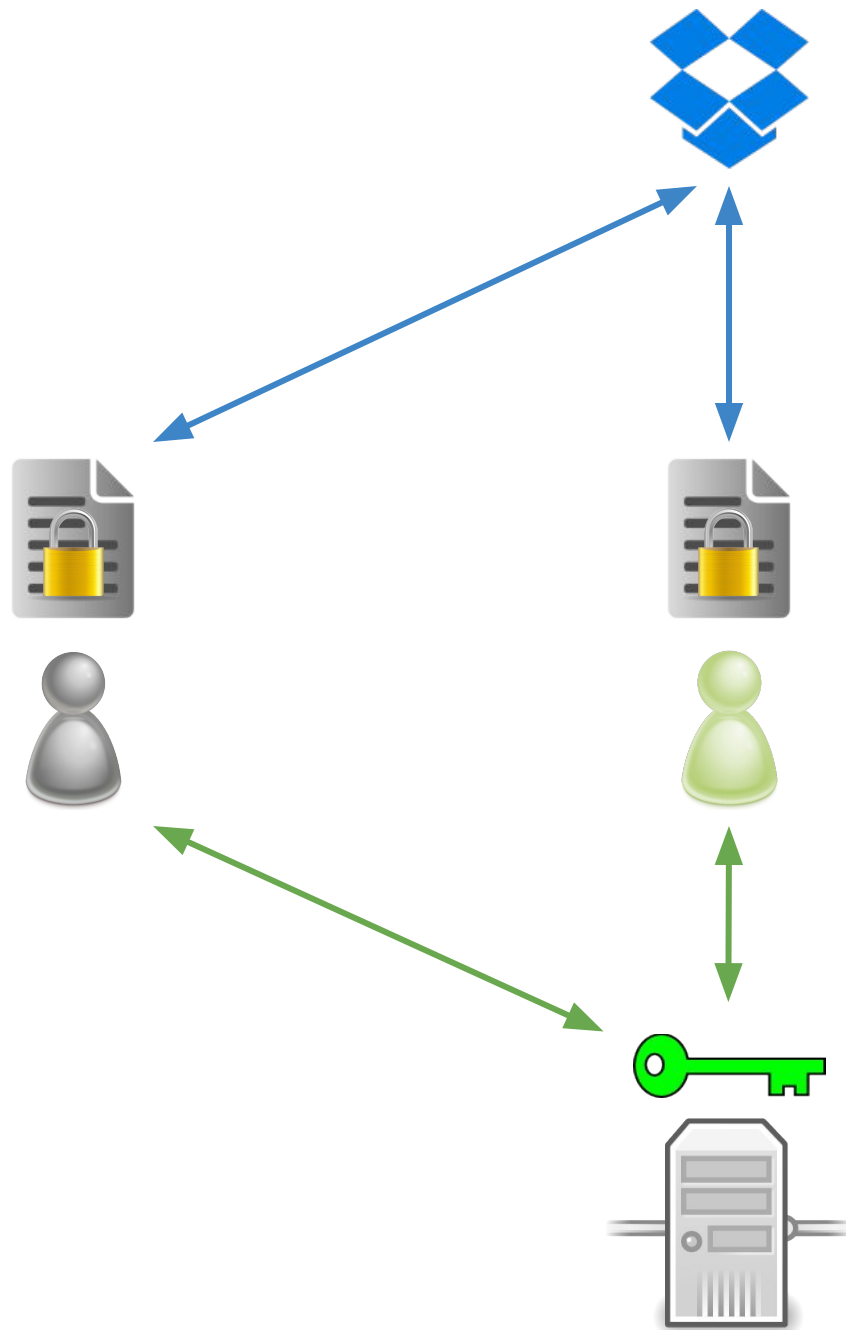


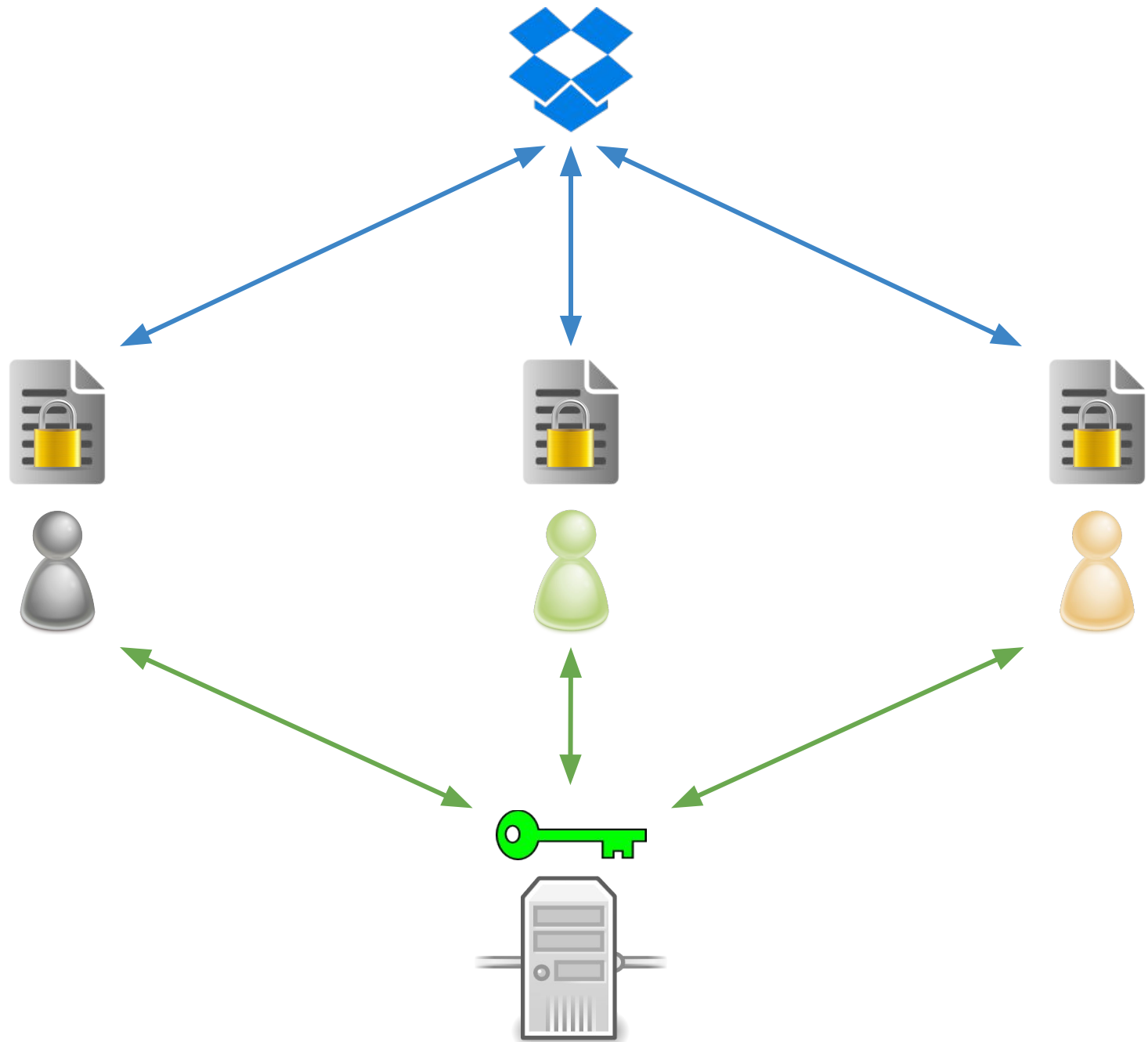


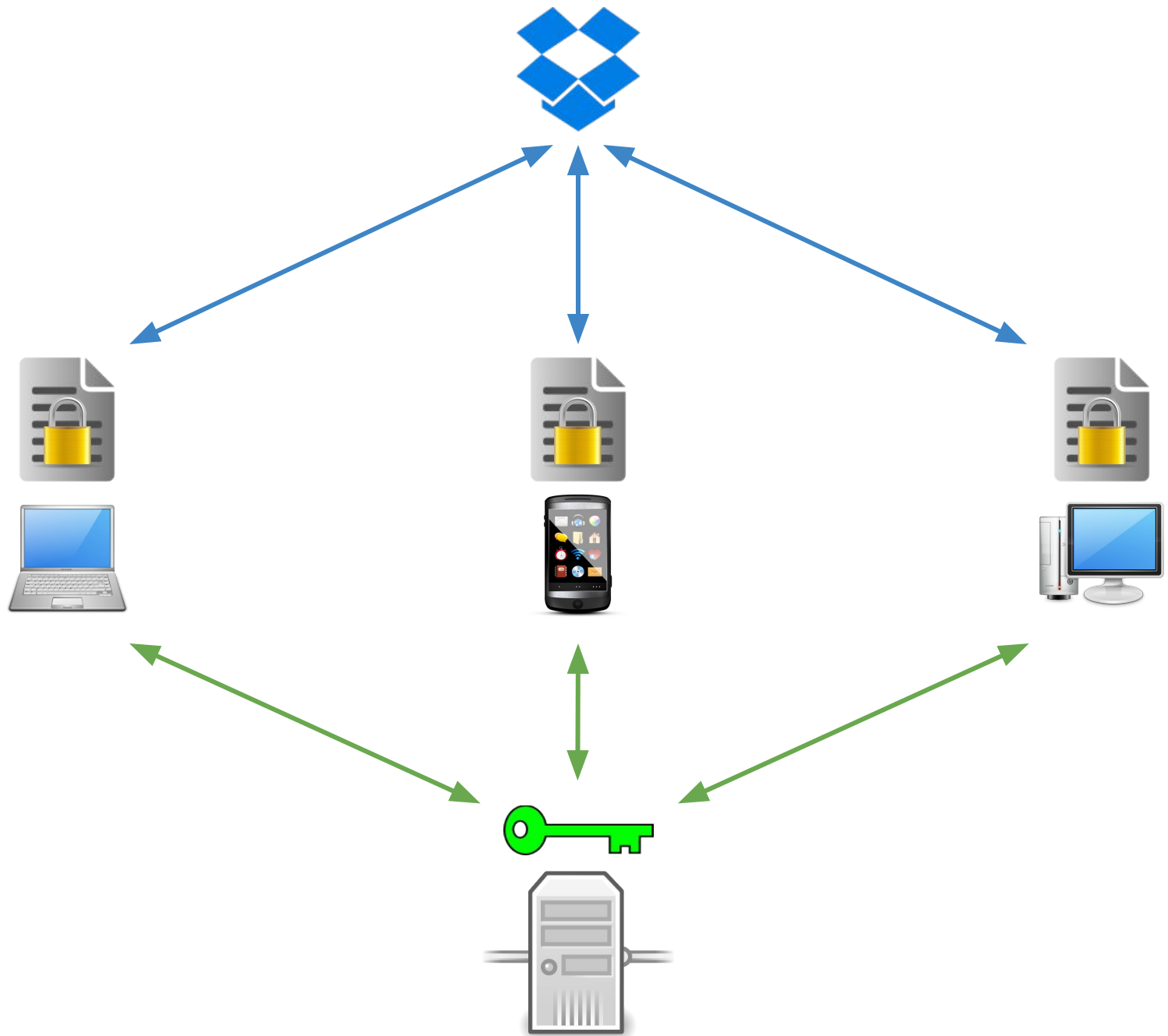












# Tutamen-backed dm-crypt/LUKS FDE







CorrectHorseBatteryStaple





CorrectHorseBatteryStaple





CorrectHorseBatteryStaple





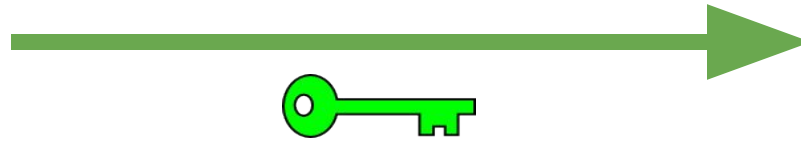
CorrectHorseBatteryStaple





CorrectHorseBatteryStaple

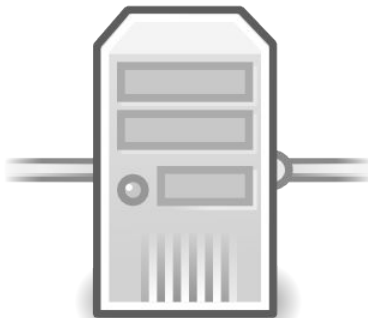


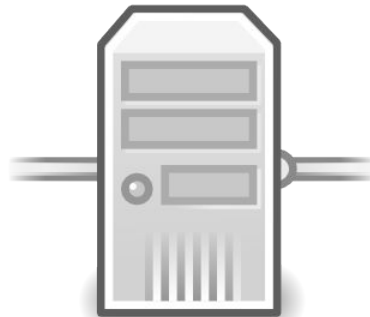








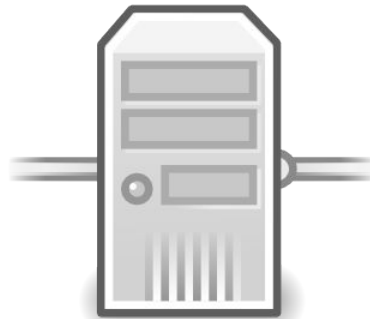




1.2.3.4/24

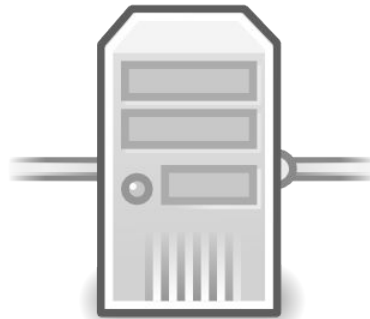


SMS Challenge

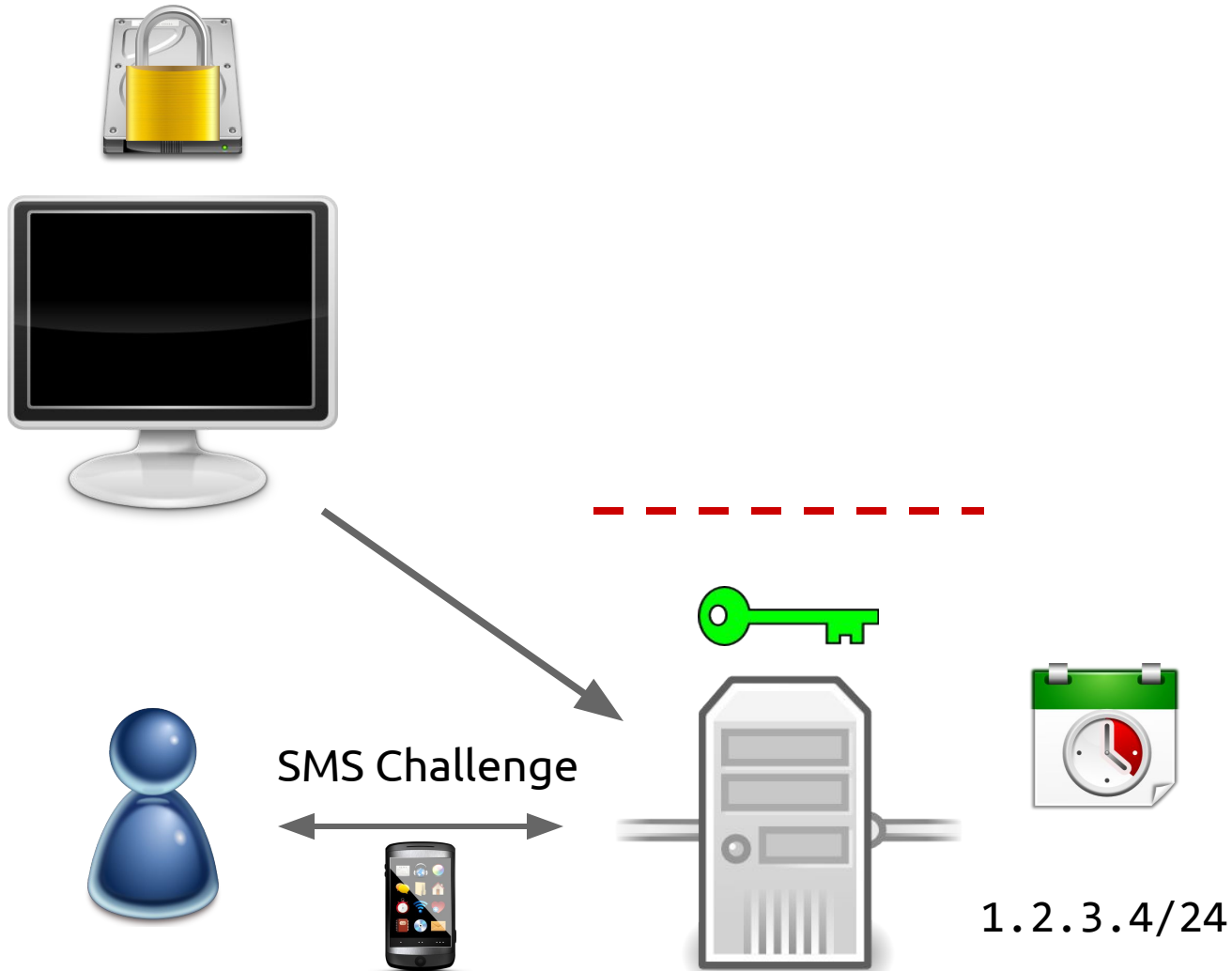


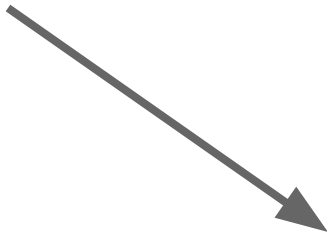


SMS Challenge



1.2.3.4/24

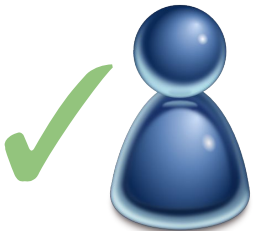


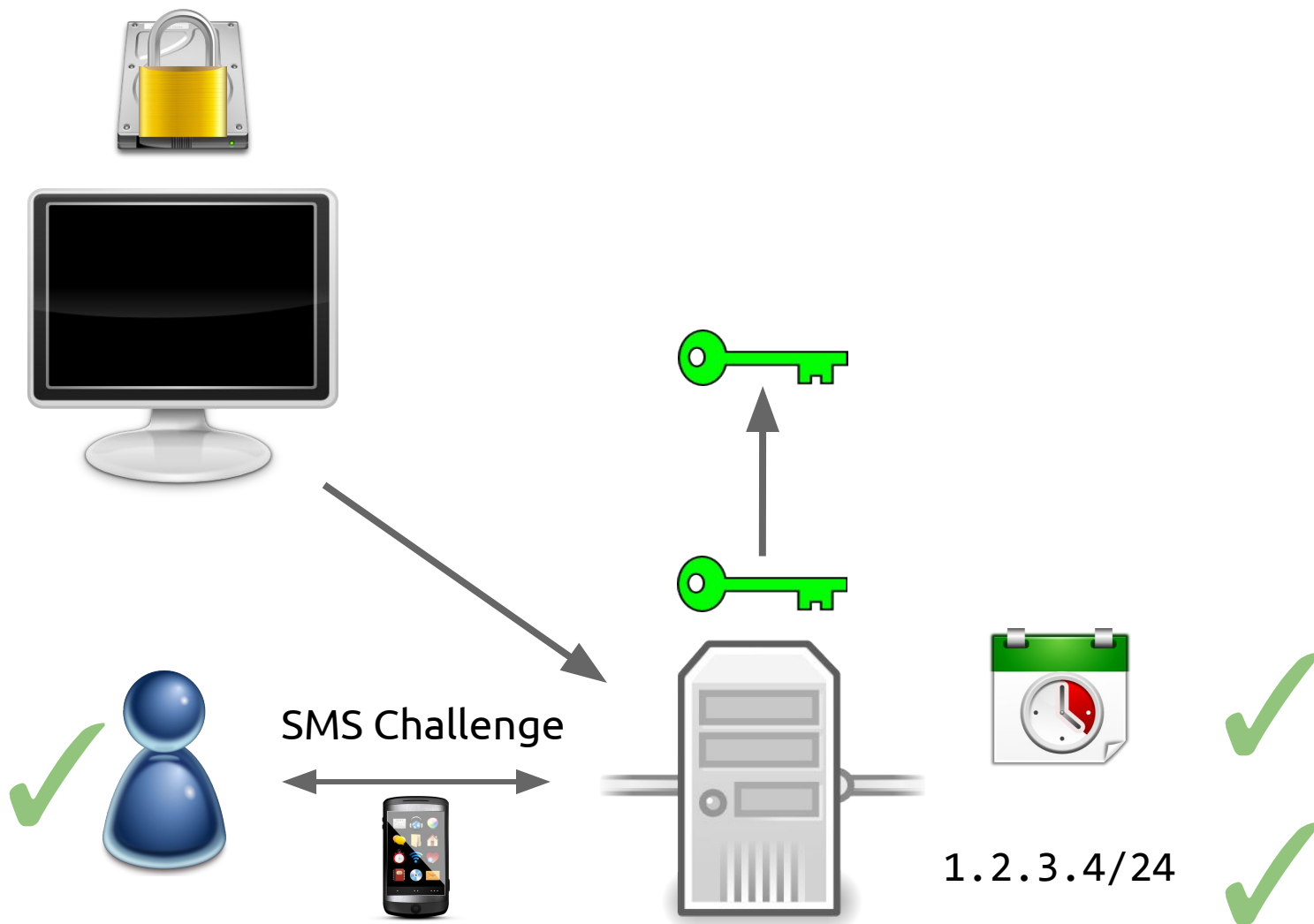


SMS Challenge

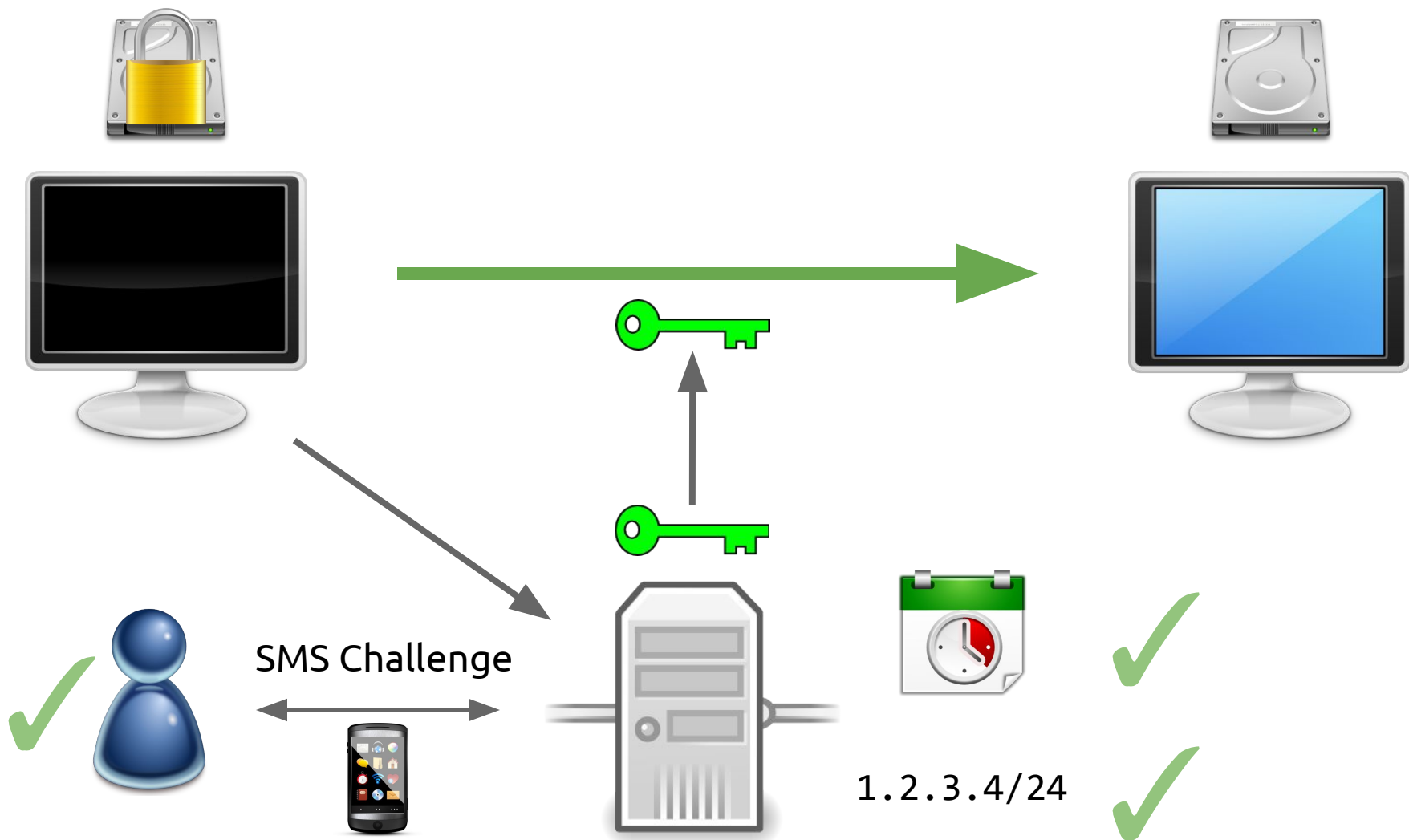


1.2.3.4/24

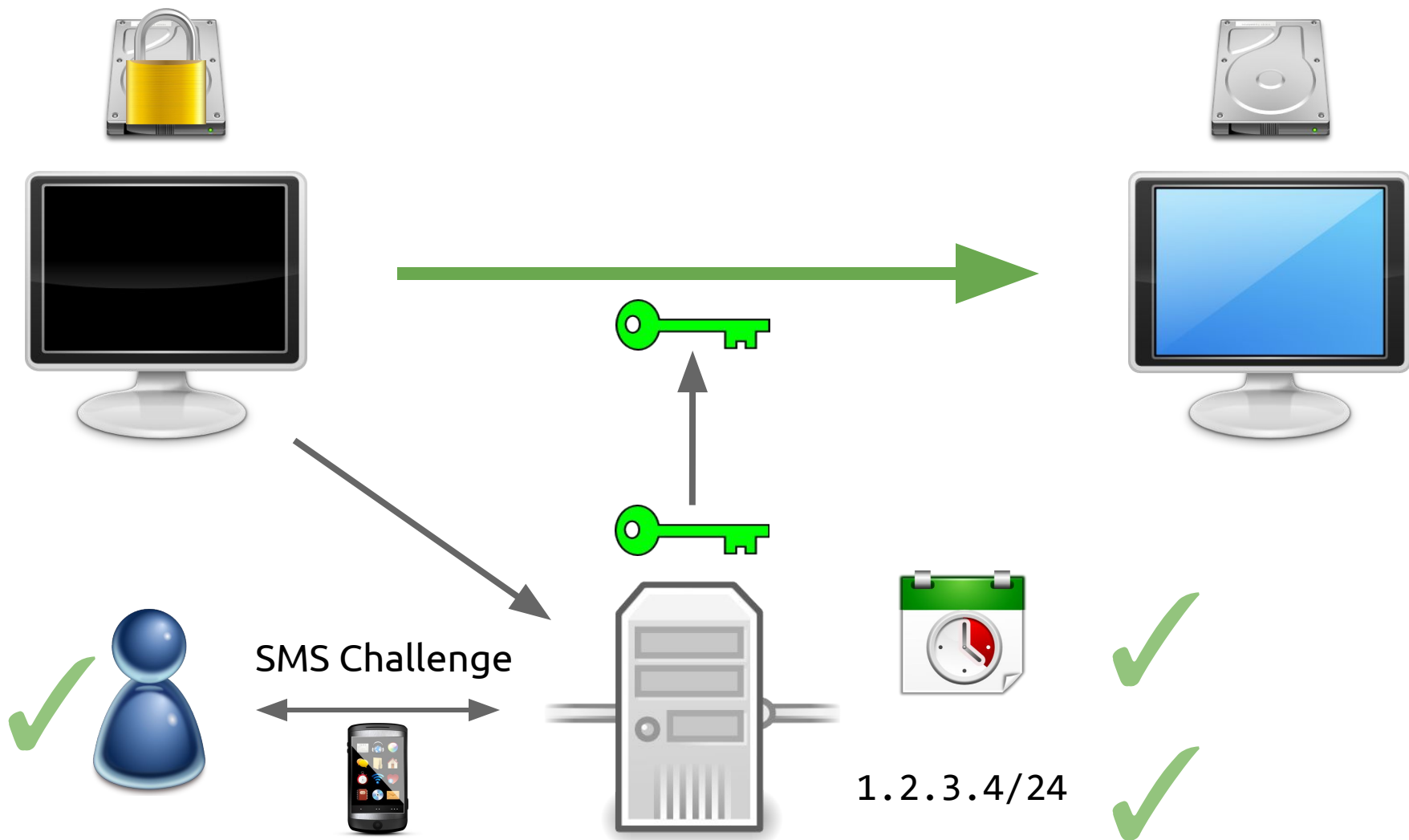


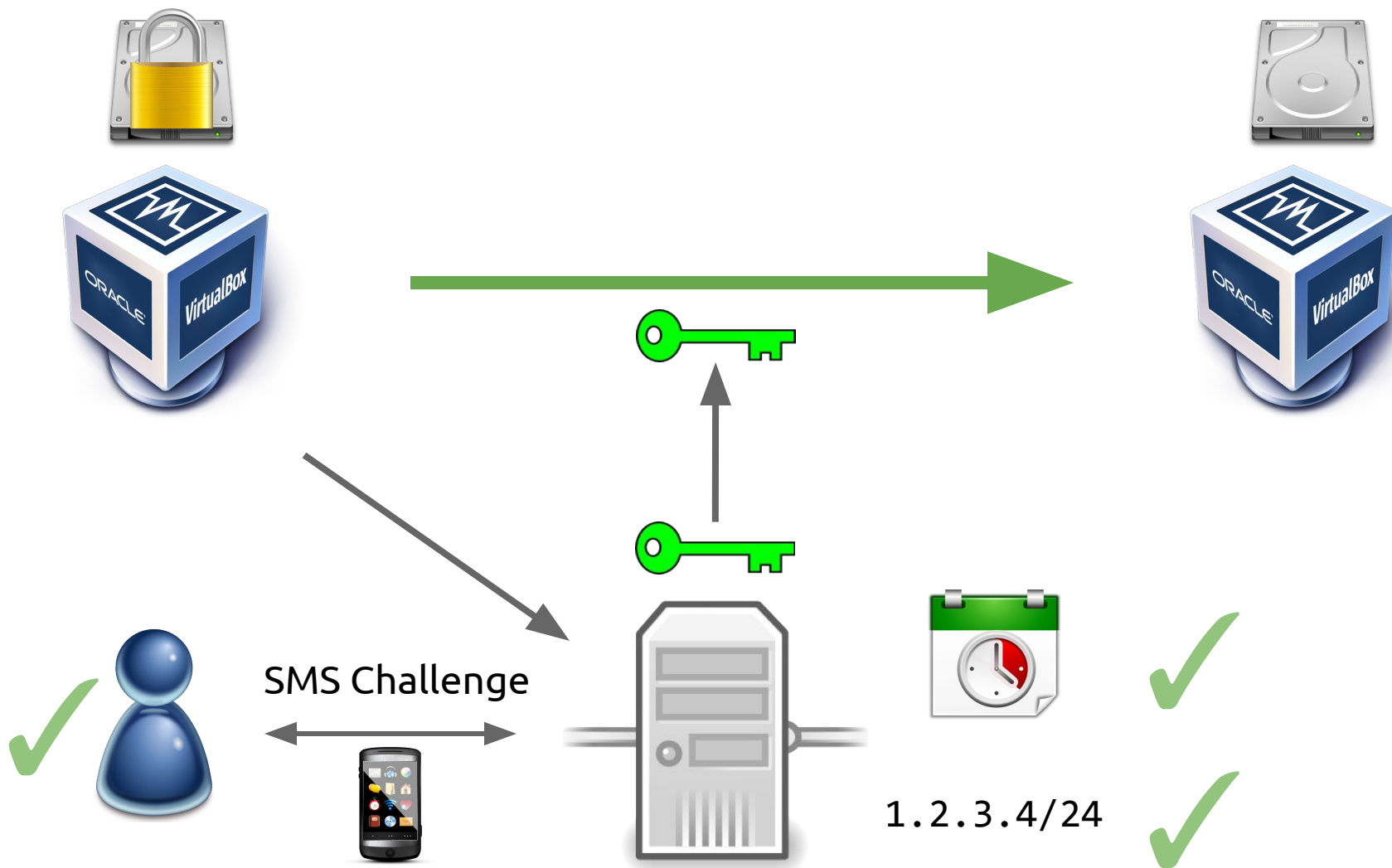




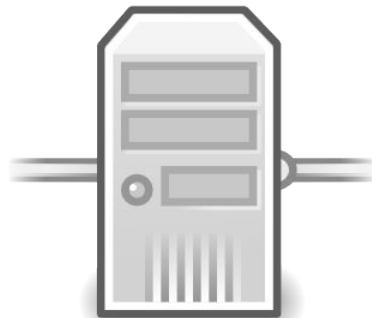


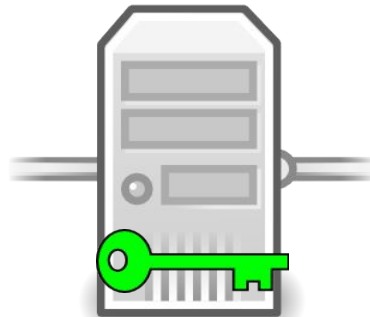
# Tutamen-backed QEMU VM Encryption

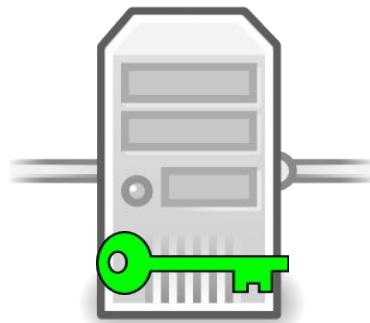




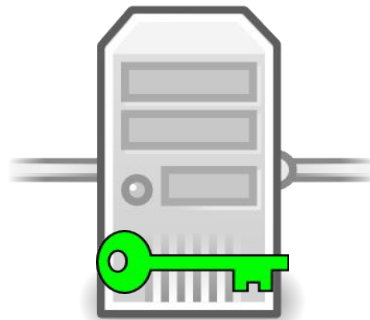
# Tutamen Management Utility











# Policy (Chapter 10)

# Access to Underlying Technology

Access to Underlying Technology

Maximizing Trustworthiness

Access to Underlying Technology

Maximizing Trustworthiness

Minimizing Trust

# Access to Underlying Technology

# Access to Underlying Technology

## LAW & DISORDER / CIVILIZATION & DISCONTENT

### Judge: Apple must help FBI unlock San Bernardino shooter's iPhone

Specifically, Apple must create custom firmware file so FBI can brute force passcode.

by Cyrus Farivar - Feb 16, 2016 7:26pm MST

[Share](#) [Tweet](#) [Email](#) 298



 Karlis Dambrāns

# Access to Underlying Technology

## LAW & DISORDER / CIVILIZATION & DISCONTENT

### Judge: Apple must help FBI unlock San Bernardino shooter's iPhone

Specifically, Apple must create custom software

by Cyrus Farivar - Feb 16, 2016 7:26pm MST



Kārlis Dambrāns

### Apple case creates fervor for encryption bill in Congress






A   Save for Later  Reading List

By Karoun Demirjian February 25  [Follow @karoun](#)



FBI Director James Comey strongly hinted to lawmakers on Thursday that it's time for Congress to begin a serious debate about encryption. (EPA/JIM LO SCALZO)

#### Most Read

- 1 'I am so sick of the Sanders campaign lying about me': Clinton snaps at Greenpeace activist 
- 2 Hillary Clinton is starting to get sick of Bernie Sanders 
- 3 The inside story of how Alabama Gov. Robert Bentley's sex scandal broke wide open 
- 4 After stumbles, Trump seeks to avert damaging loss in Wisconsin 
- 5 Trump could be stripped of his South Carolina delegates. The question is: Would the GOP dare? 



# Access to Underlying Technology

## LAW & DISORDER / CIVILIZATION & DISCONTENT

### Judge: Apple must help FBI unlock San Bernardino shooter's iPhone

Specifically, Apple must create custom

by Cyrus Farivar - Feb 16, 2016 7:26pm MST



Kārlis Dambrāns

### Apple case creates fervor for encryption bill in Congress

A   2  Save for Later  Reading List

By Karoun Demirjian February 25   Follow @karoun



FBI Director James Comey strongly hinted to lawmakers on Thursday that it's time for Congress to begin a serious debate about encryption. (EPA/JIM LO SCALZO)

### Keys Under Doormats:

MANDATING INSECURITY BY REQUIRING GOVERNMENT ACCESS TO ALL DATA AND COMMUNICATIONS

Harold Abelson, Ross Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Whitfield Diffie, John Gilmore, Matthew Green, Susan Landau, Peter G. Neumann, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Michael Specter, Daniel J. Weitzner

#### Abstract

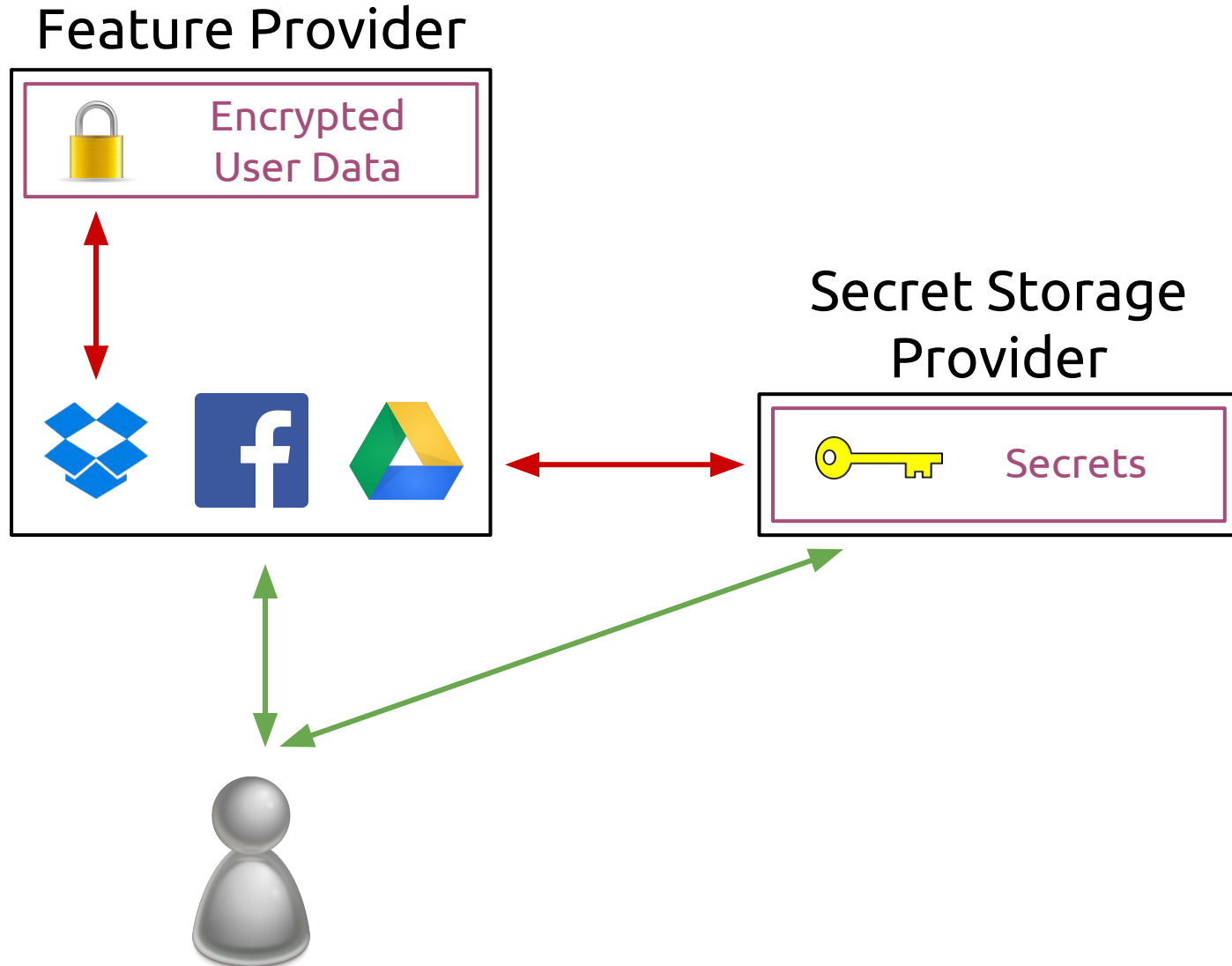
Twenty years ago, law enforcement organizations lobbied to require data and communication services to engineer their products to guarantee law enforcement access to all data. After lengthy debate and vigorous predictions of enforcement channels "going dark," these attempts to regulate the emerging Internet were abandoned. In the intervening years, innovation on the Internet flourished, and law enforcement agencies found new and more effective means of accessing vastly larger quantities of data. Today we are again hearing calls for regulation to mandate the provision of exceptional access mechanisms. In this report, a group of computer scientists and security experts, many of whom participated in a 1997 study of these same topics, has convened to explore the likely effects of imposing extraordinary access mandates.

We have found that the damage that could be caused by law enforcement exceptional access requirements would be even greater today than it would have been 20 years ago. In the wake of the growing economic and social cost of the fundamental insecurity of today's Internet environment, any proposals that alter the security dynamics online should be approached with caution. Exceptional access would force Internet system developers to reverse "forward secrecy" design practices that seek to minimize the impact on user privacy when systems are breached. The complexity of today's Internet environment, with millions of apps and globally connected services, means that new law enforcement requirements are likely to introduce unanticipated, hard to detect security flaws. Beyond these and other technical vulnerabilities, the prospect of globally deployed exceptional access systems raises difficult problems about how such an environment would be governed and how to ensure that such systems would respect human rights and the rule of law.

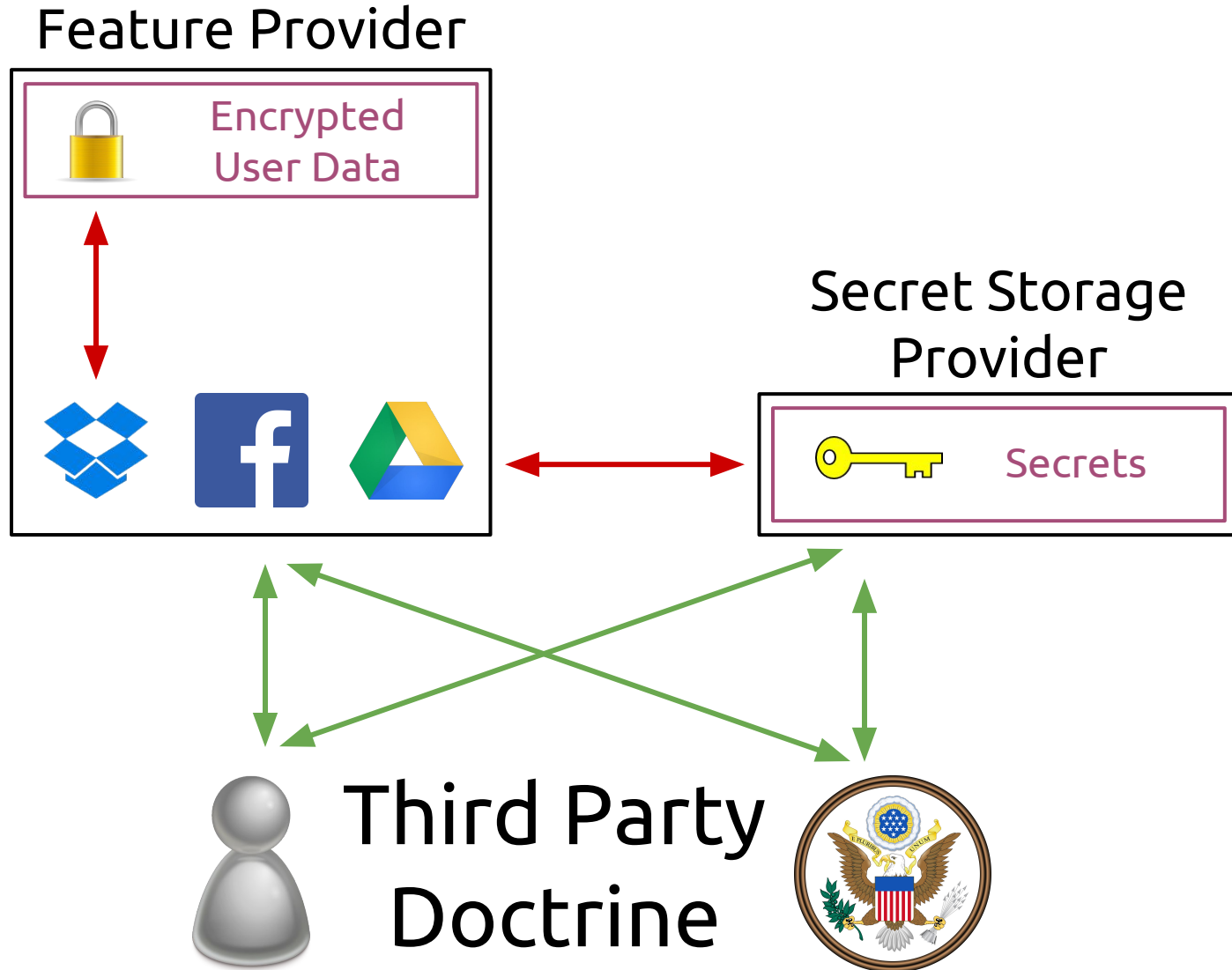
July 7, 2015

# Maximizing Trustworthiness

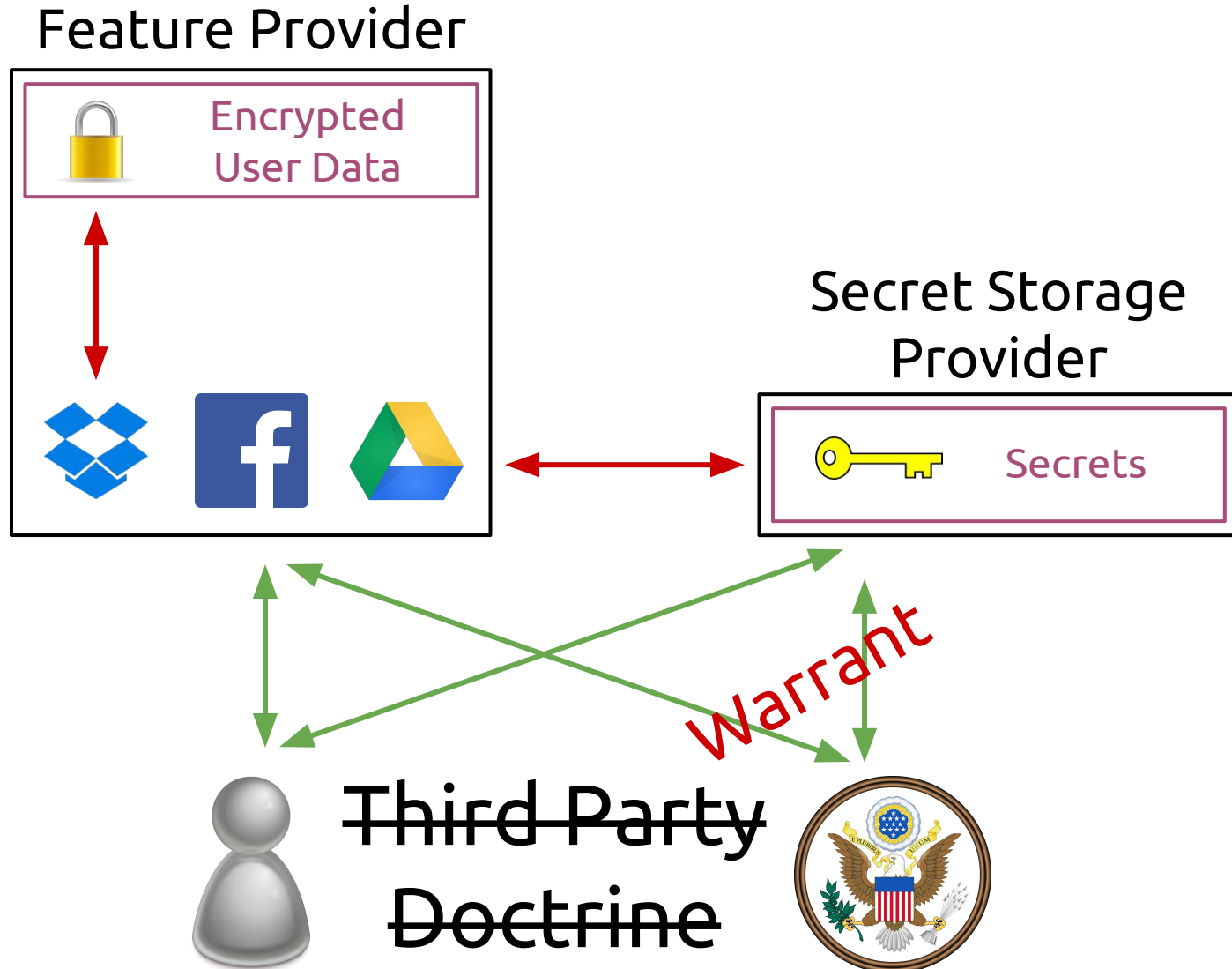
# Maximizing Trustworthiness



# Maximizing Trustworthiness



# Maximizing Trustworthiness



# Maximizing Trustworthiness

# Maximizing Trustworthiness

Secret Storage  
Provider



# Maximizing Trustworthiness

Secret Storage  
Provider



Liability



# Maximizing Trustworthiness

Secret Storage  
Provider



Liability



Insurance

# Maximizing Trustworthiness

\$\$\$ ?



Secrets

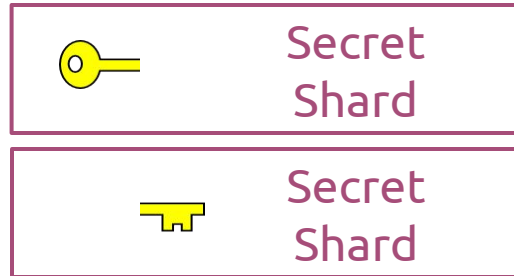
Liability



Insurance

# Maximizing Trustworthiness

\$\$\$ ???



Liability



Insurance

# Minimizing Trust

# Minimizing Trust

Network Working Group  
Request for Comments: 1149

D. Waitzman  
BBN STC  
1 April 1990

A Standard for the Transmission of IP Datagrams on Avian Carriers

## Status of this Memo

This memo describes an experimental method for the encapsulation of IP datagrams in avian carriers. This specification is primarily useful in Metropolitan Area Networks. This is an experimental, not recommended standard. Distribution of this memo is unlimited.

## Overview and Rational

Avian carriers can provide high delay, low throughput, and low altitude service. The connection topology is limited to a single point-to-point path for each carrier, used with standard carriers, but many carriers can be used without significant interference with each other, outside of early spring. This is because of the 3D ether space available to the carriers, in contrast to the 1D ether used by IEEE802.3. The carriers have an intrinsic collision avoidance system, which increases availability. Unlike some network technologies, such as packet radio, communication is not limited to line-of-sight distance. Connection oriented service is available in some cities, usually based upon a central hub topology.

## Frame Format

The IP datagram is printed, on a small scroll of paper, in hexadecimal, with each octet separated by whitestuff and blackstuff. The scroll of paper is wrapped around one leg of the avian carrier. A band of duct tape is used to secure the datagram's edges. The bandwidth is limited to the leg length. The MTU is variable, and paradoxically, generally increases with increased carrier age. A typical MTU is 256 milligrams. Some datagram padding may be needed.

Upon receipt, the duct tape is removed and the paper copy of the datagram is optically scanned into a electronically transmittable form.

## Discussion

Multiple types of service can be provided with a prioritized pecking order. An additional property is built-in worm detection and eradication. Because IP only guarantees best effort delivery, loss of a carrier can be tolerated. With time, the carriers are self-

# Standardization

# Minimizing Trust

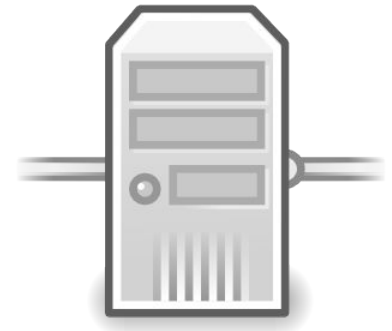
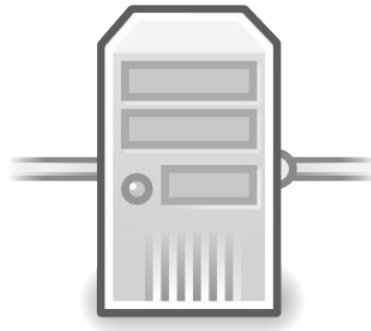
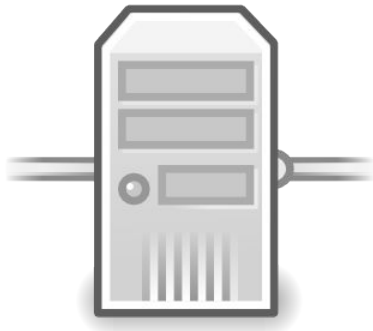
Secret Storage  
Provider



Secret Storage  
Provider



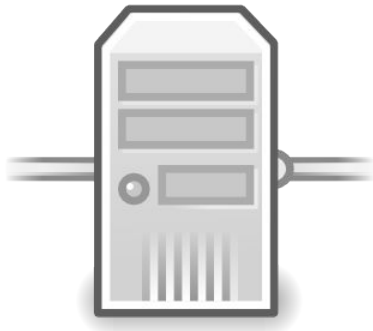
Secret Storage  
Provider



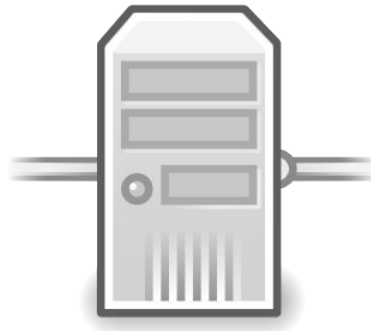
Competition

# Minimizing Trust

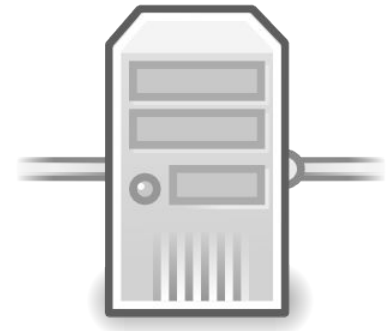
Secret Storage  
Provider



Secret Storage  
Provider

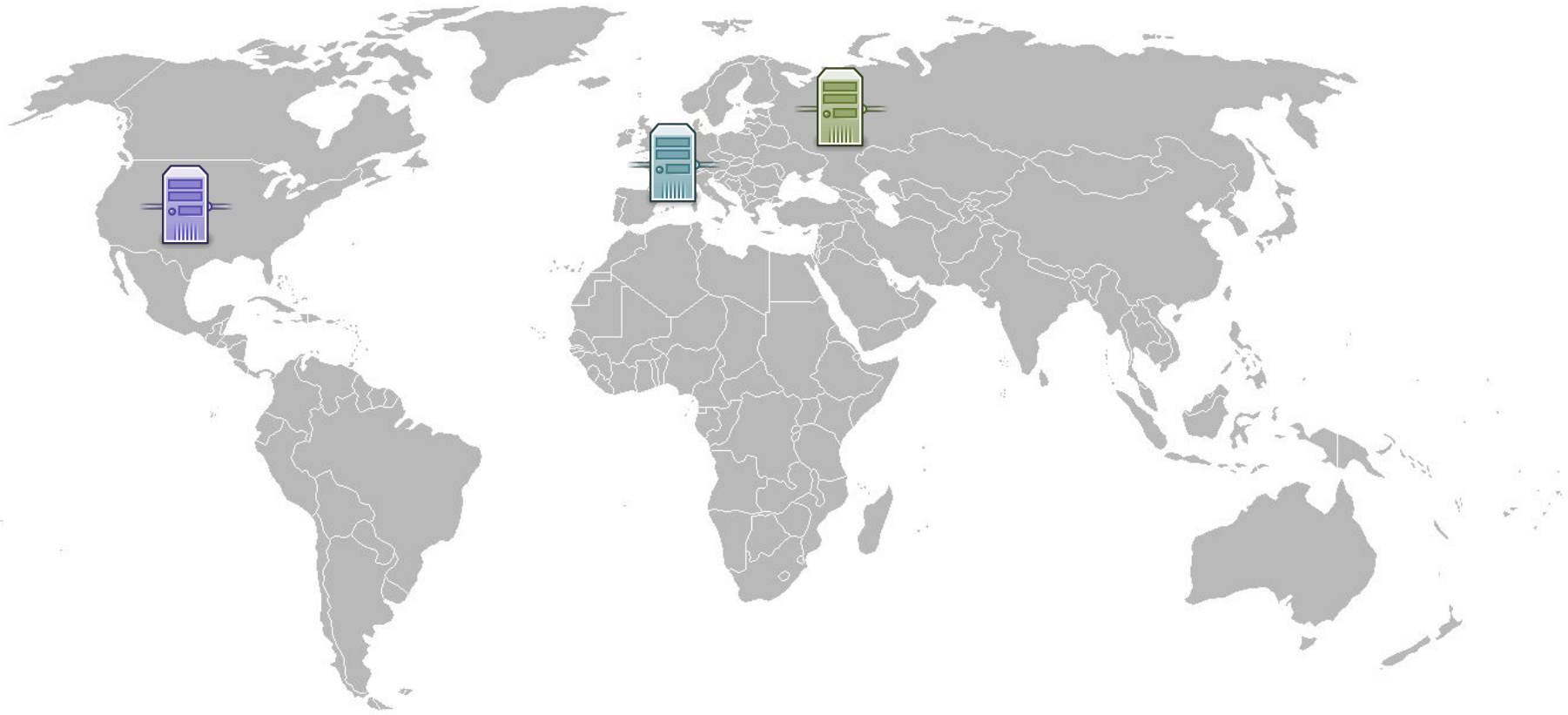


Secret Storage  
Provider



Competition

# Minimizing Trust



Jurisdictional Arbitrage



# Conclusion

How can we **secure**  
and **control** our data?

*(even in the presence **third parties**)*

*(while also supporting modern **use cases**)*

# Secret Storage as a Service

# Secret Storage as a Service

Separation of Trust and Features

# Secret Storage as a Service

Separation of Trust and Features

Avoids Single Trusted Third Party

# Secret Storage as a Service

Separation of Trust and Features

Avoids Single Trusted Third Party

Commoditize Privacy and Security

# Secret Storage as a Service

Supports Common Use Cases

# Secret Storage as a Service

Supports Common Use Cases



Increase End-user Security



# Future Work

Auditing -> Automation

Auditing -> Automation

Performance and Deployment

Auditing -> Automation

Performance and Deployment

Further Policy Exploration

Thank You

Questions?

Extra Slides





# Modern Demands

# Modern Demands



# Third-Party Solutions

Modern Demands



Third-Party Solutions



Security and Privacy Concerns

Modern Demands



Third-Party Solutions



Security and Privacy Concerns



New Solutions?

Application	Storage	Access	Manipulation	Meta-analysis	Score

Degree of Third Party Trust Across Capabilities



Application	Storage	Access	Manipulation	Meta-analysis	Score
Dropbox	Full	Full	Full	Full	12
Facebook	Full	Full	Full	Full	12
Gmail	Full	Full	Full	Full	12
Hangouts	Full	Full	Full	Full	12
Amazon EC2	Full	Full	Full	Full	12

Degree of Third Party Trust Across Capabilities



Application	Storage	Access	Manipulation	Meta-analysis	Score
Dropbox	Full	Full	Full	Full	12
Tresorit	Full	Partial	Partial	Full	10
Facebook	Full	Full	Full	Full	12
Gmail	Full	Full	Full	Full	12
PGP/GPG	Full	None	None	Full	6
Hangouts	Full	Full	Full	Full	12
TextSecure	Full	None	None	Minimal	4
LastPass	Full	Minimal	Full	Full	10
Amazon EC2	Full	Full	Full	Full	12

Degree of Third Party Trust Across Capabilities



Application	Storage	Access	Manipulation	Meta-analysis	Score
Dropbox	Full	Full	Full	Full	12
Tresorit	Full	Partial	Partial	Full	10
Facebook	Full	Full	Full	Full	12
Gmail	Full	Full	Full	Full	12
PGP/GPG	Full	None	None	Full	6
Hangouts	Full	Full	Full	Full	12
TextSecure	Full	None	None	Minimal	4
LastPass	Full	Minimal	Full	Full	10
Amazon EC2	Full	Full	Full	Full	12
Single SSP	Full	Partial	Partial	Full	10
Multiple SSPs	Partial	Minimal	Minimal	Partial	6

Degree of Third Party Trust Across Capabilities





Application	Implicit	Compelled	Unintended	Colluding	Score

Risk of Third Party Trust Violations



Application	Implicit	Compelled	Unintended	Colluding	Score
Dropbox	Disincent.	Known	Disincent.	N/A	5
Facebook	Known	Known	Disincent.	N/A	7
Gmail	Vulnerable	Known	Disincent.	N/A	6
Hangouts	Vulnerable	Known	Disincent.	N/A	6
Amazon EC2	Disincent.	Known	Disincent.	N/A	5

Risk of Third Party Trust Violations



Application	Implicit	Compelled	Unintended	Colluding	Score
Dropbox	Disincent.	Known	Disincent.	N/A	5
Tresorit	Disincent.	Vulnerable	Disincent.	N/A	4
Facebook	Known	Known	Disincent.	N/A	7
Gmail	Vulnerable	Known	Disincent.	N/A	6
PGP/GPG	Disincent.	Disincent.	Minimized	N/A	2
Hangouts	Vulnerable	Known	Disincent.	N/A	6
TextSecure	Disincent.	Disincent.	Minimized	N/A	2
LastPass	Disincent.	Vulnerable	Disincent.	N/A	4
Amazon EC2	Disincent.	Known	Disincent.	N/A	5

## Risk of Third Party Trust Violations

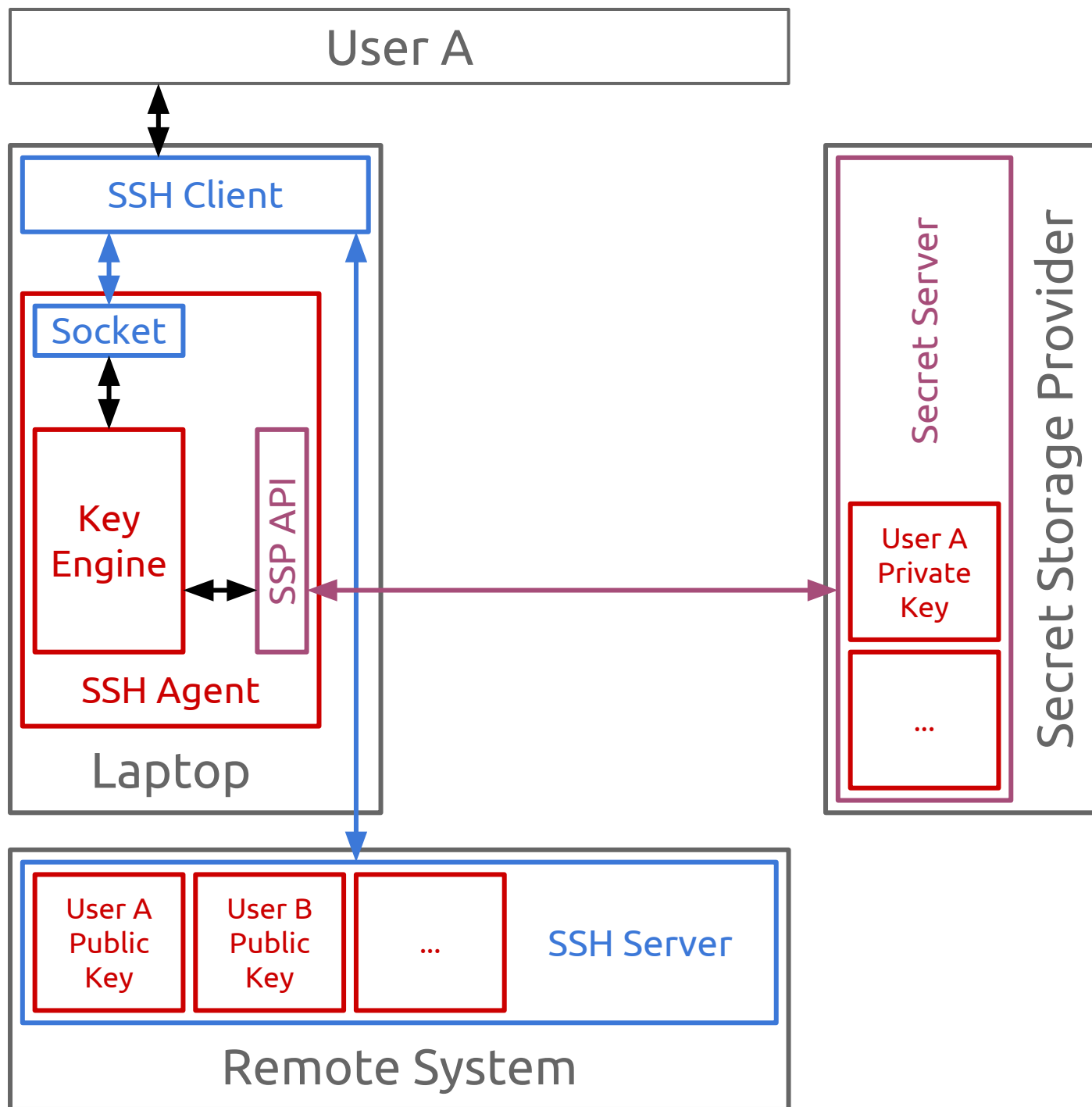


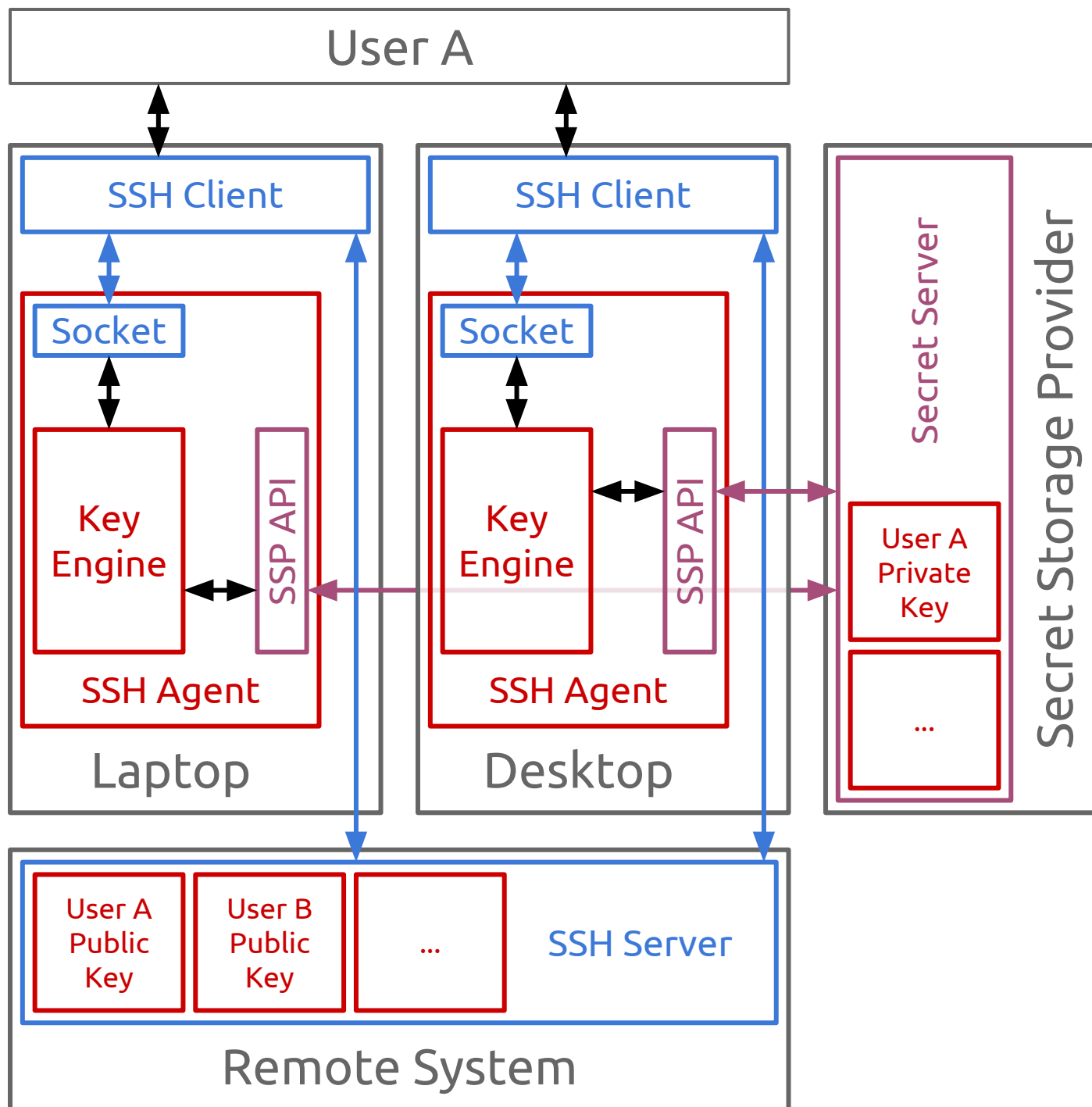
Application	Implicit	Compelled	Unintended	Colluding	Score
Dropbox	Disincent.	Known	Disincent.	N/A	5
Tresorit	Disincent.	Vulnerable	Disincent.	N/A	4
Facebook	Known	Known	Disincent.	N/A	7
Gmail	Vulnerable	Known	Disincent.	N/A	6
PGP/GPG	Disincent.	Disincent.	Minimized	N/A	2
Hangouts	Vulnerable	Known	Disincent.	N/A	6
TextSecure	Disincent.	Disincent.	Minimized	N/A	2
LastPass	Disincent.	Vulnerable	Disincent.	N/A	4
Amazon EC2	Disincent.	Known	Disincent.	N/A	5
Single SSP	Disincent.	Disincent.	Minimized	Disincent.	3
Multiple SSPs	Disincent.	Minimized	Minimized	Minimized	1

### Risk of Third Party Trust Violations



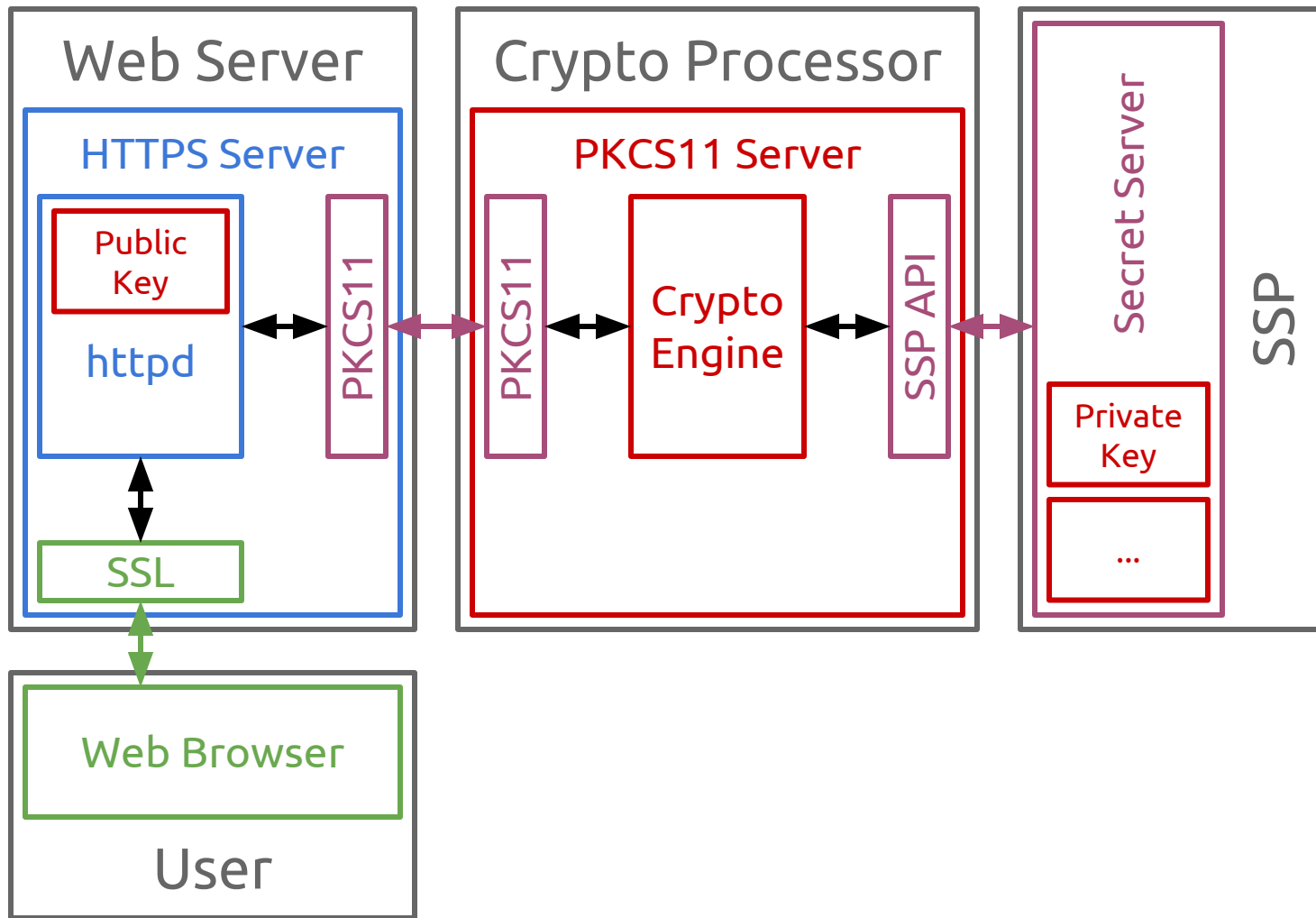
# Authentication Applications



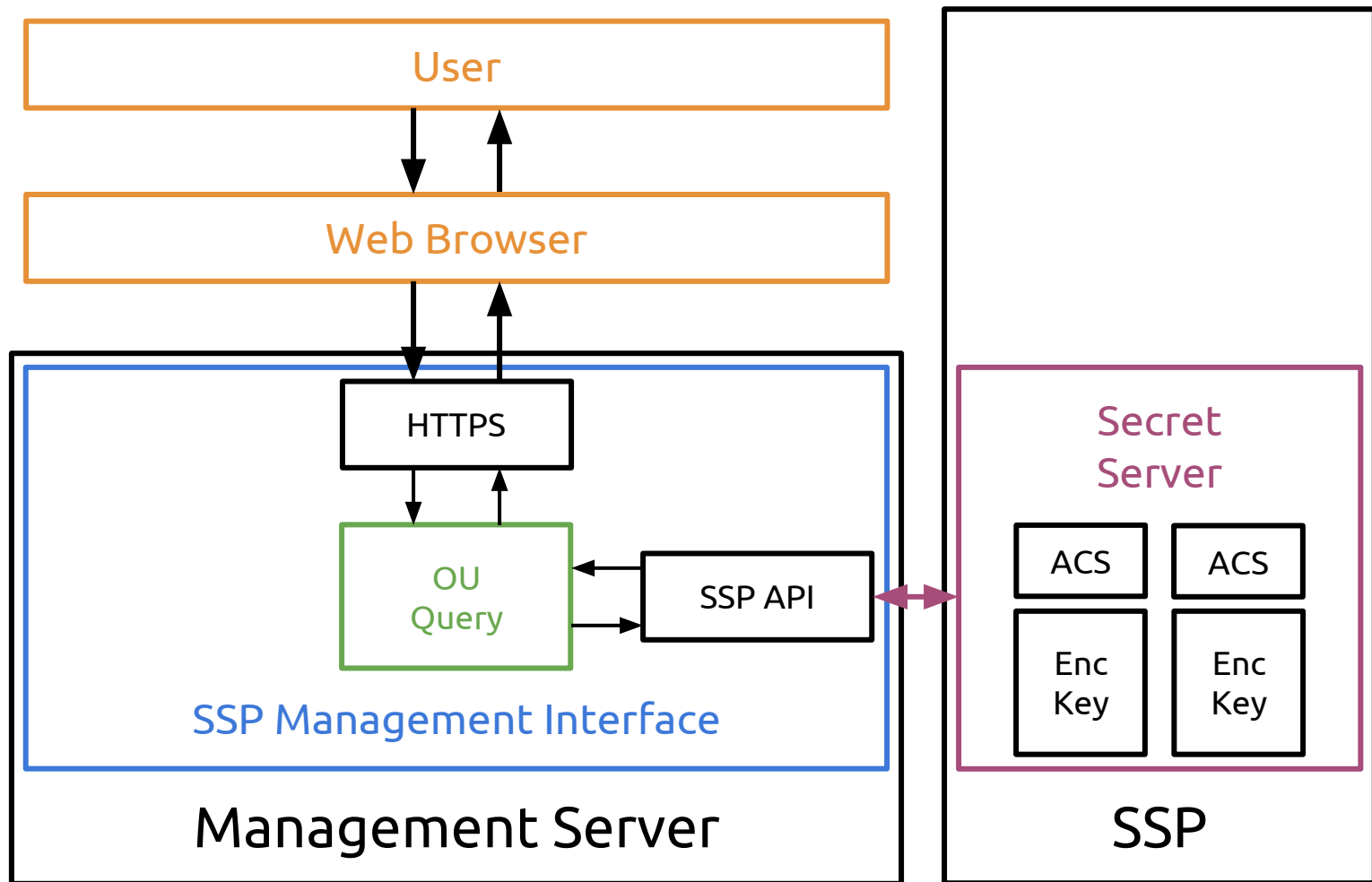


# Crypto Processing Applications

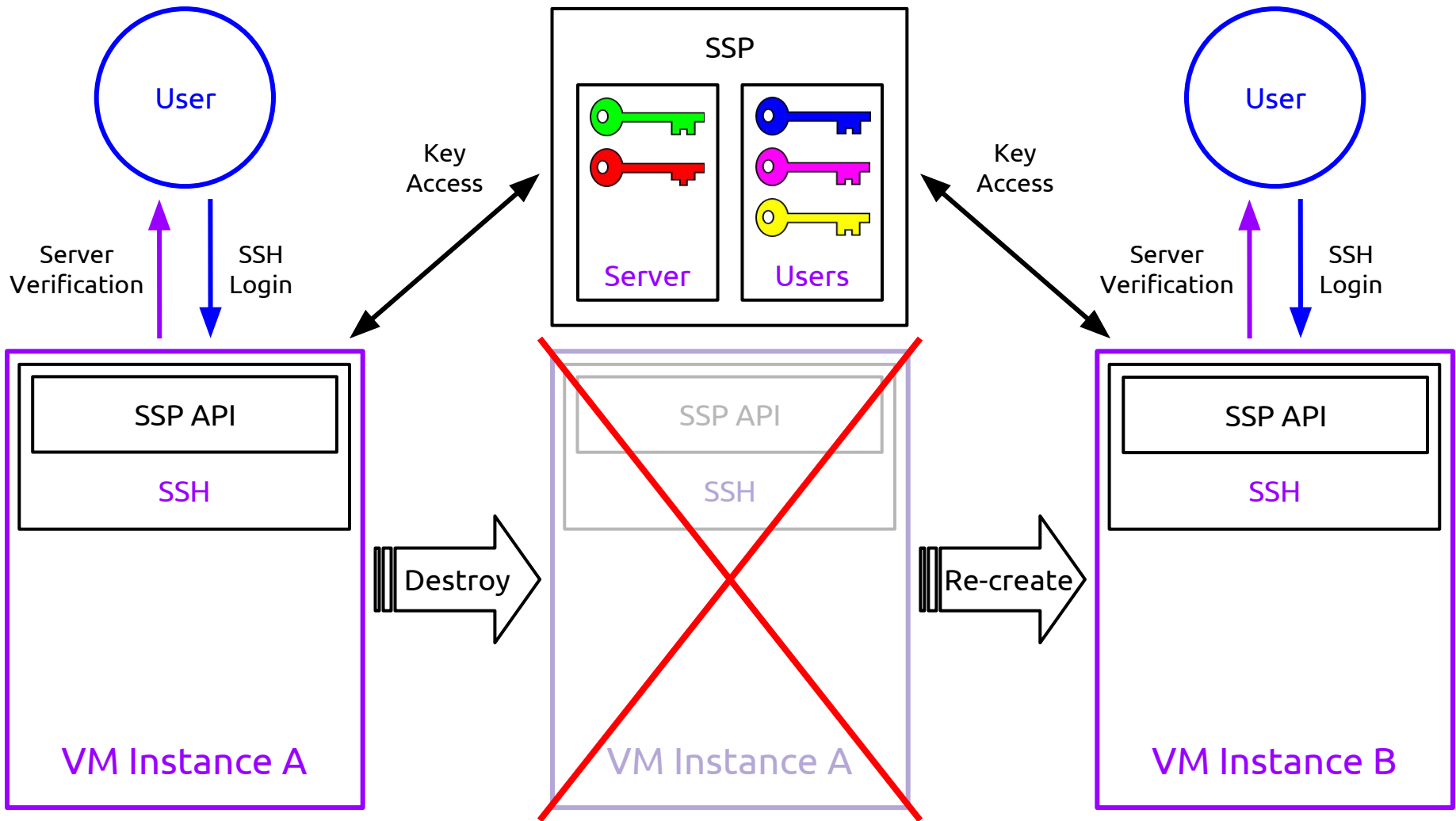




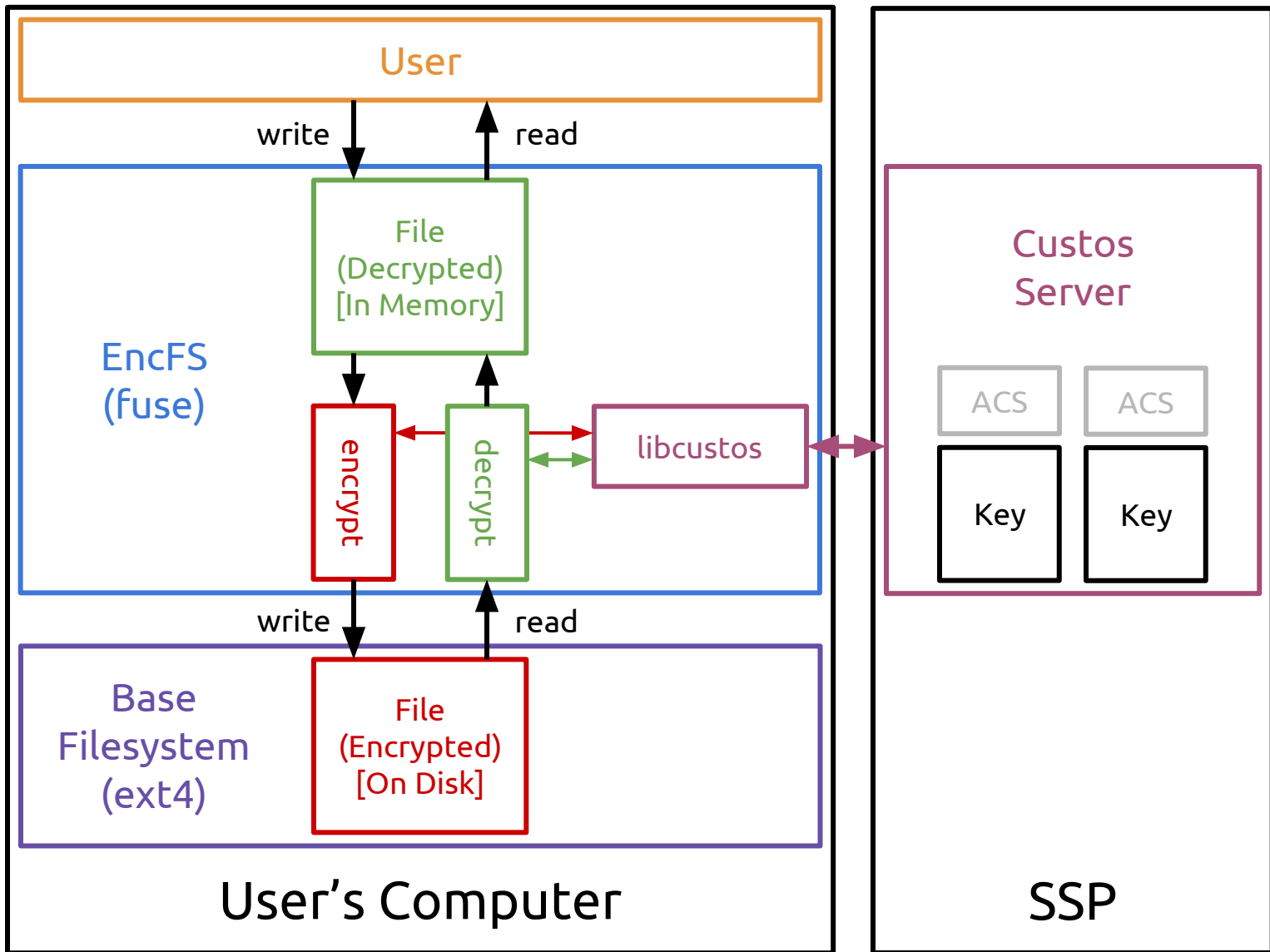
Management Server

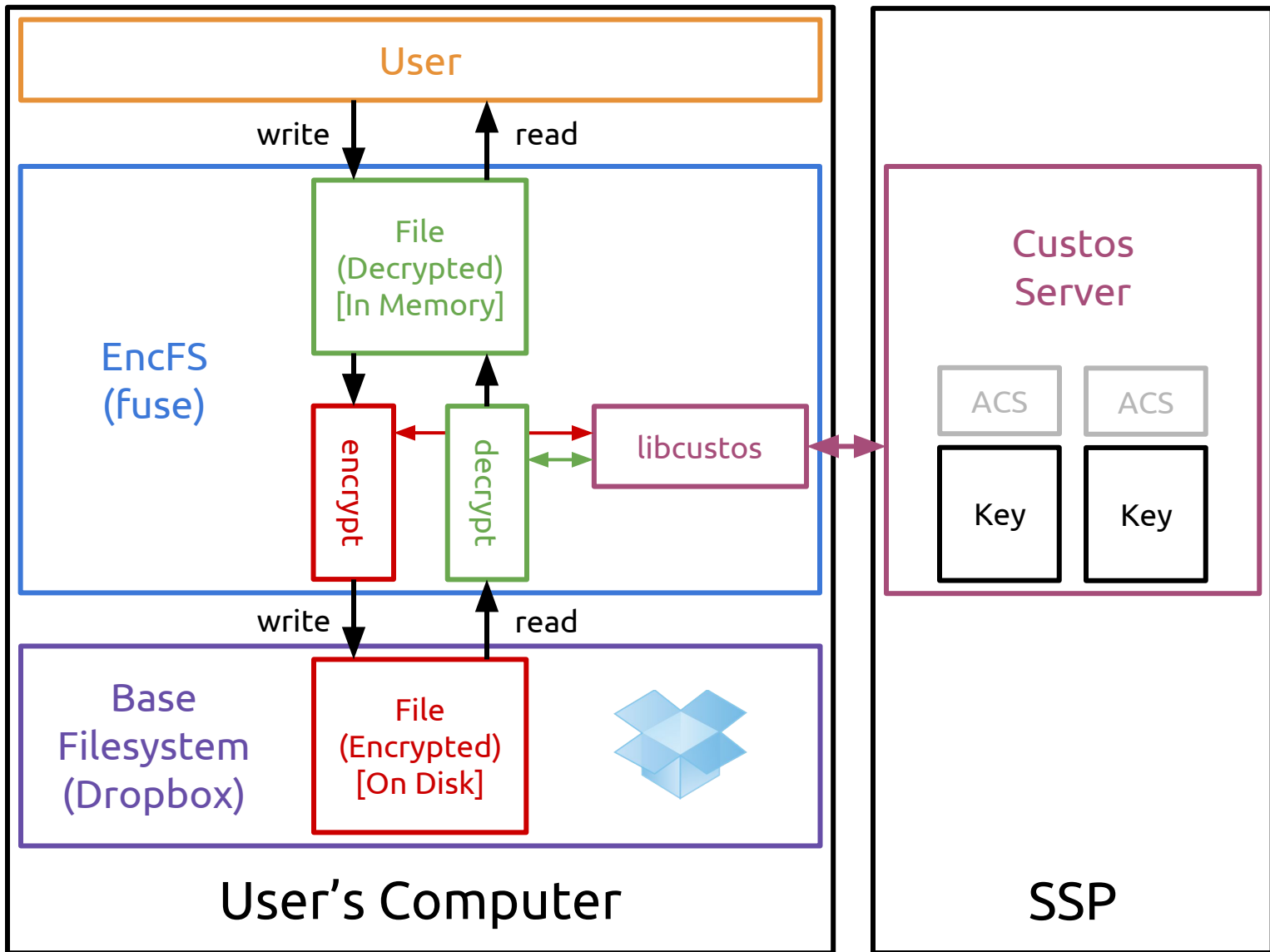


# SSH Server Key Management

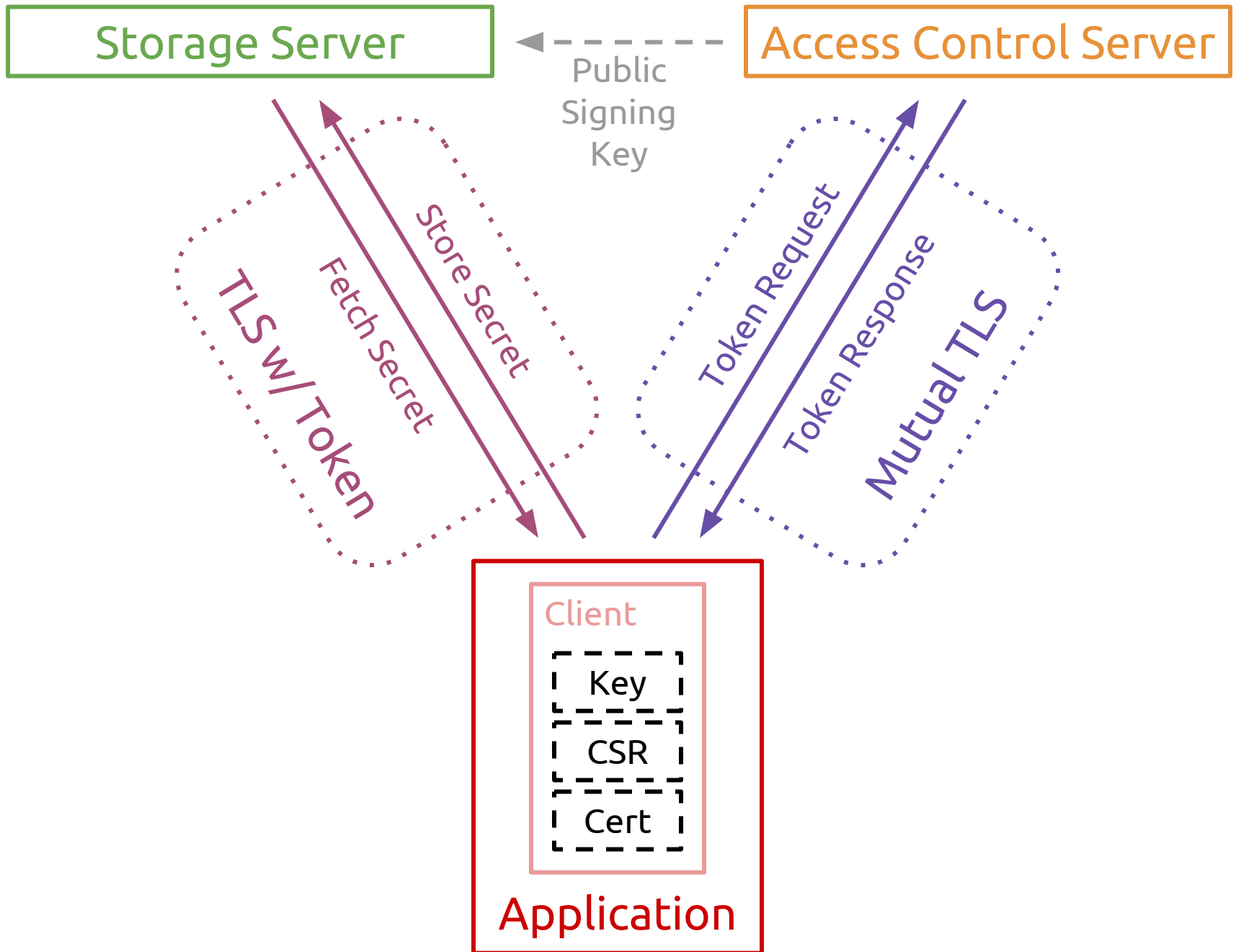


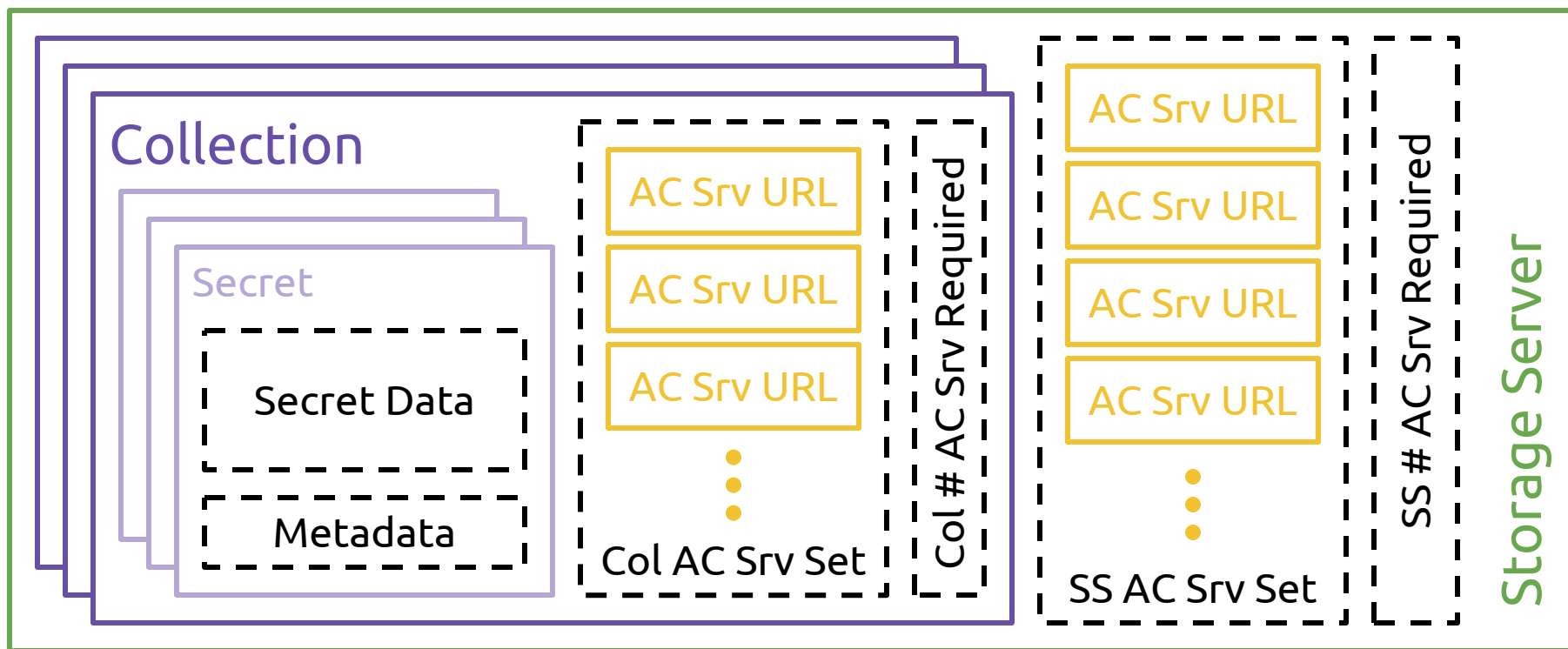
# EncFS: Custos-Backed Encrypted File System











# Access Control Server

Account

Client

CSR

Cert

Authenticator



Auth Plugin

Plugin Data

Permissions

Object Type

Object ID

Verifier

Account ID

Account ID

Account ID



Account Set

Authenticator ID

Authenticator ID

Authenticator ID



Authenticator Set

Permission Name

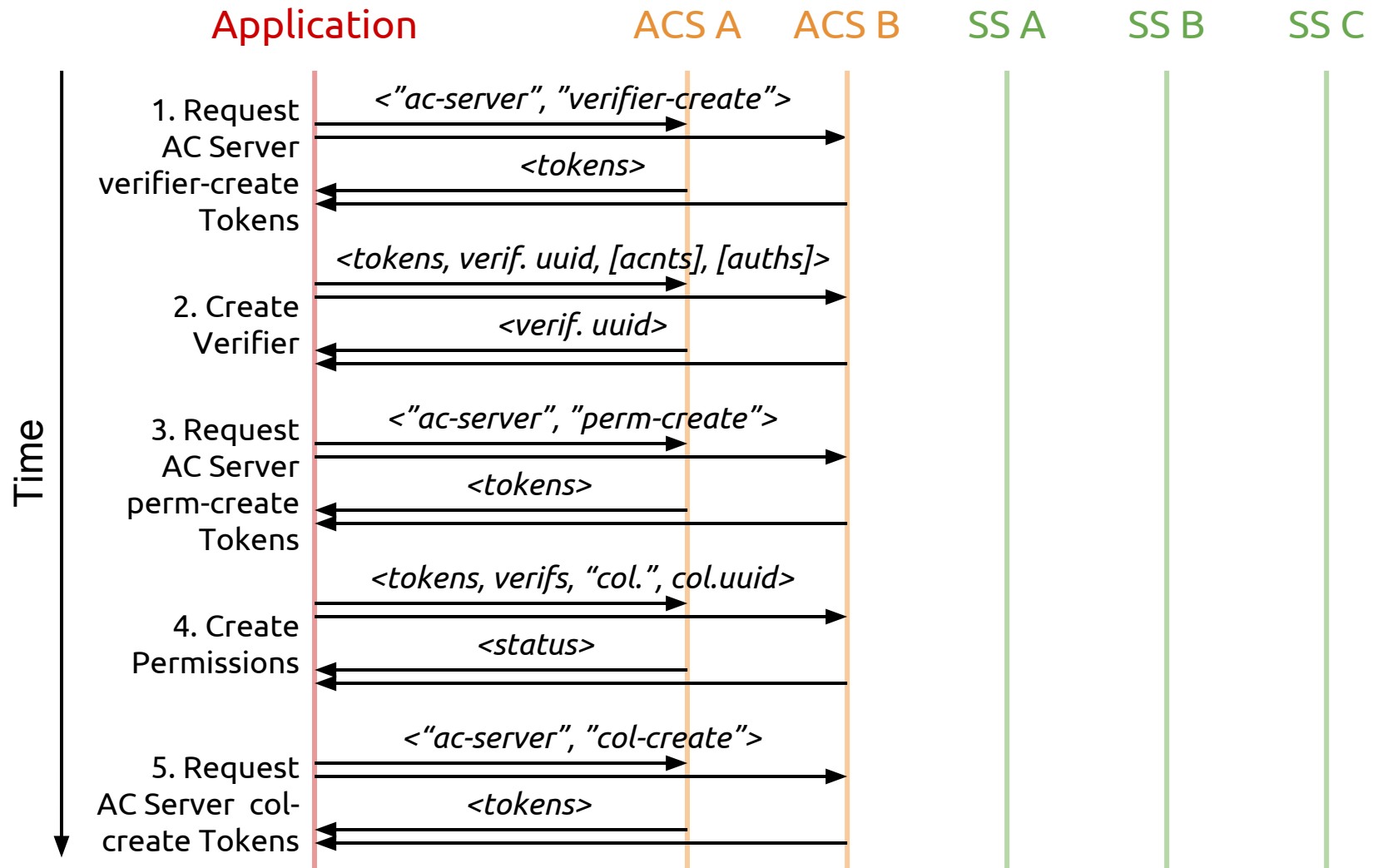
Verifier ID

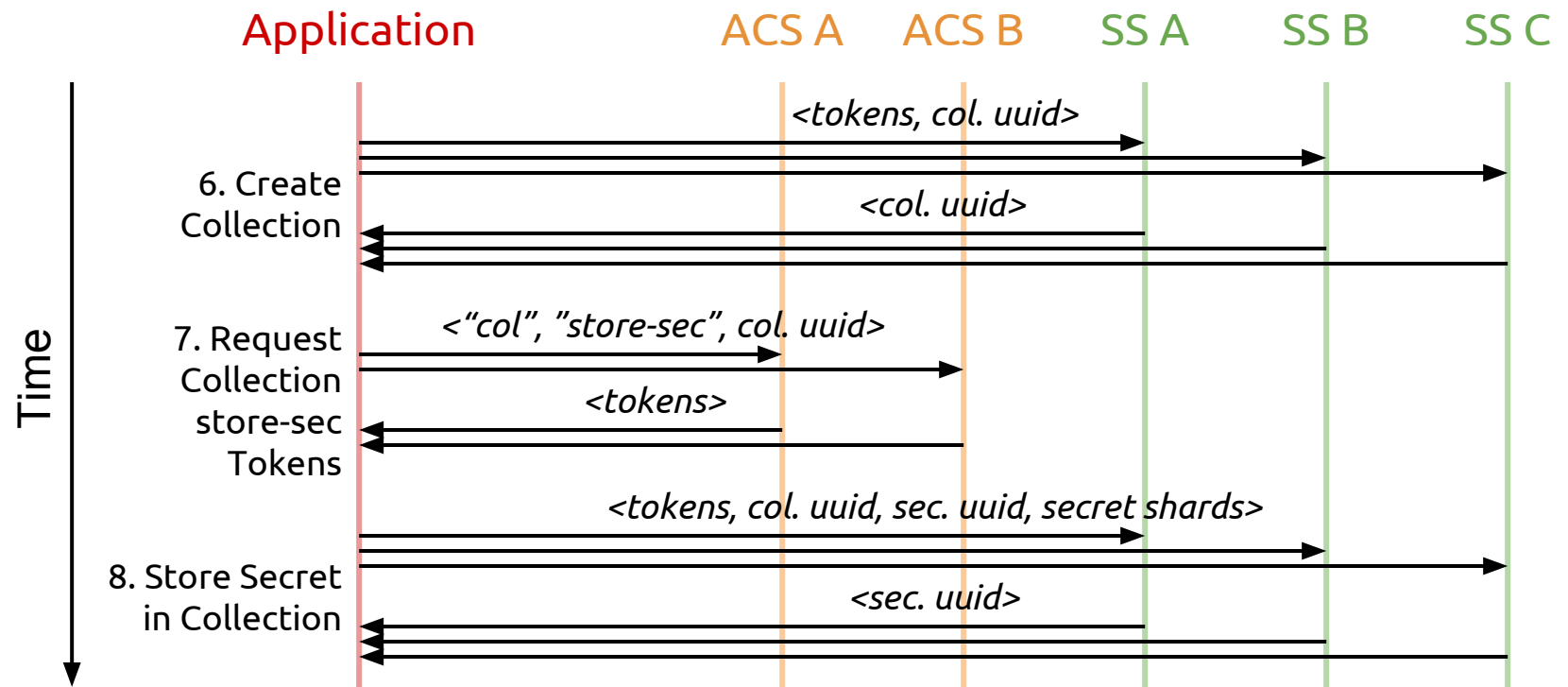
Verifier ID

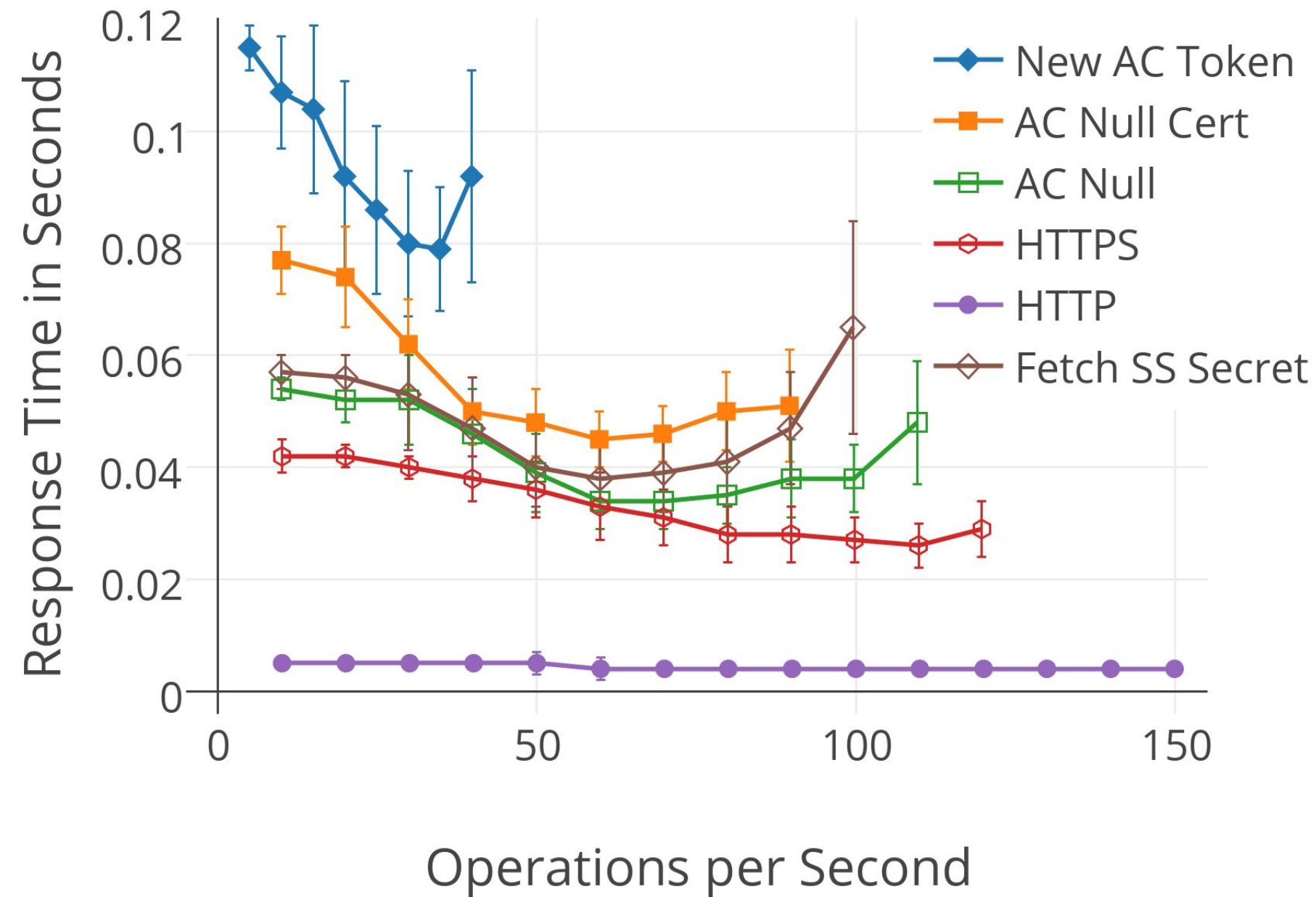
Verifier ID

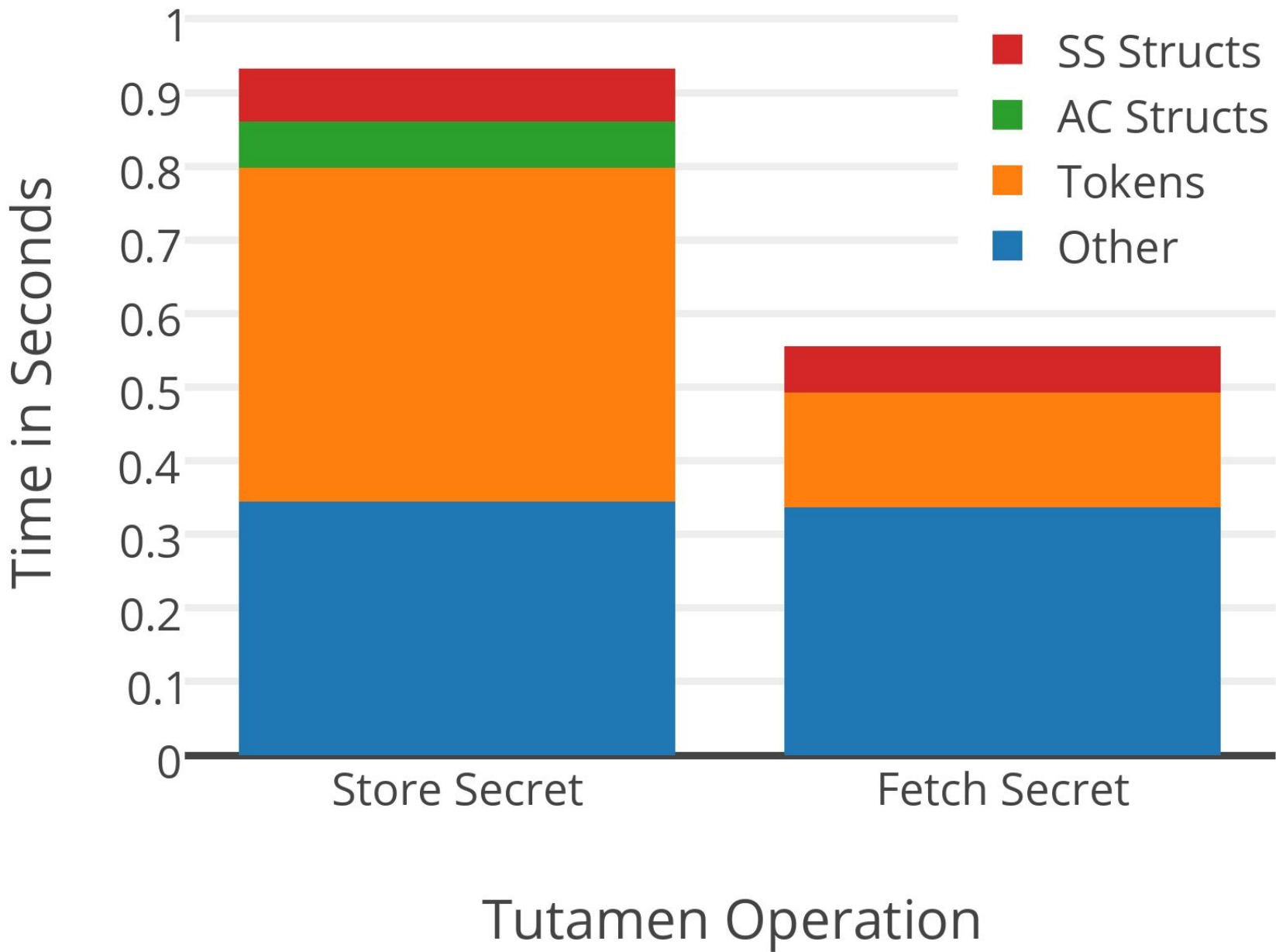


Verifier Set









# Asymmetric Cryptography



Alice

Encryption

Bob's  
Public Key



Bob,  
  
This is my  
super secret  
message.  
  
-Alice



Encrypt



Qm9iLApUa  
GlzIGlzIG15I  
HN1cGVyIH  
NlY3JldCBtZ  
XNzYWdlLgo  
tQWxpY2UK



Decrypt

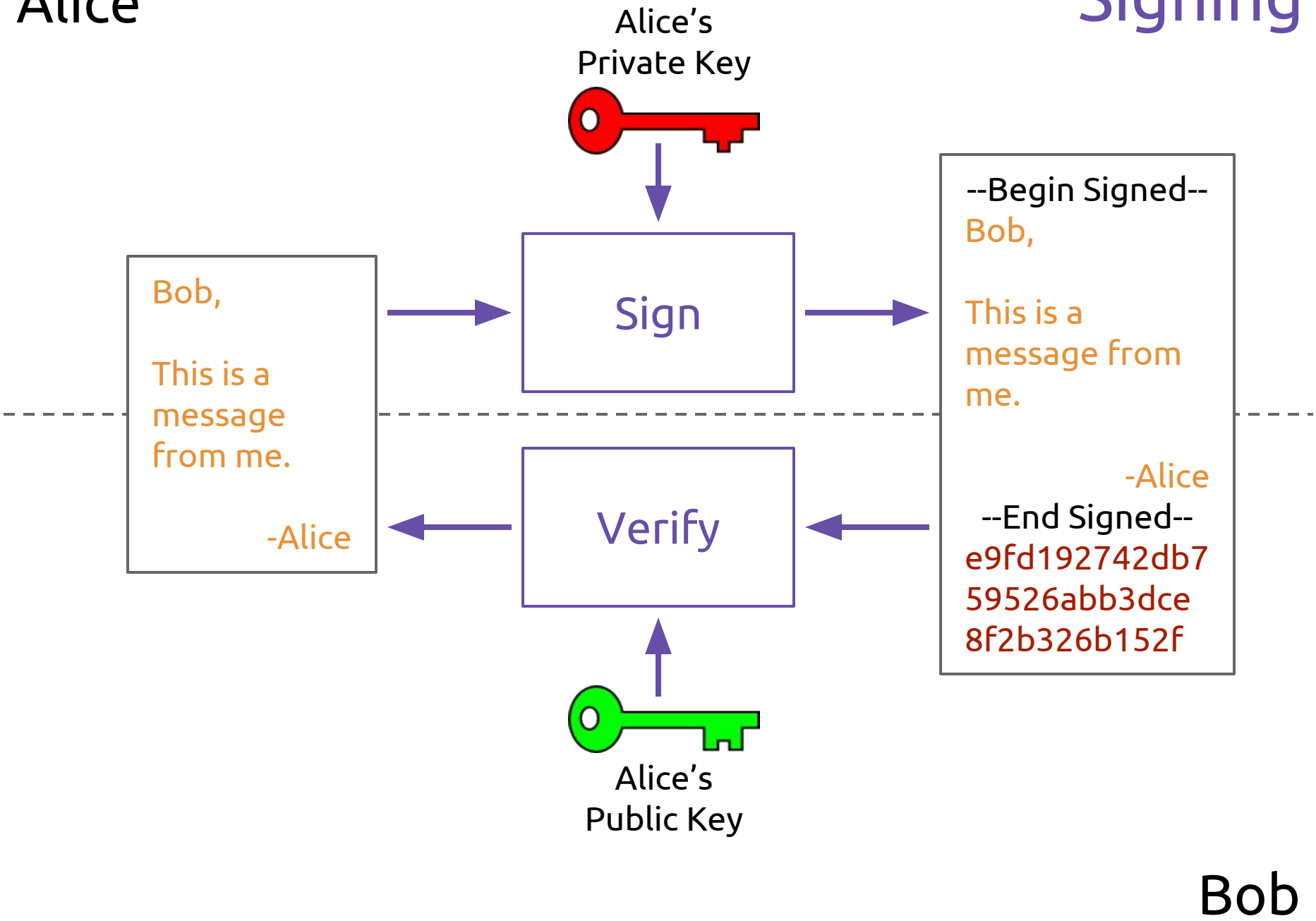


Bob's  
Private Key

Bob

Alice

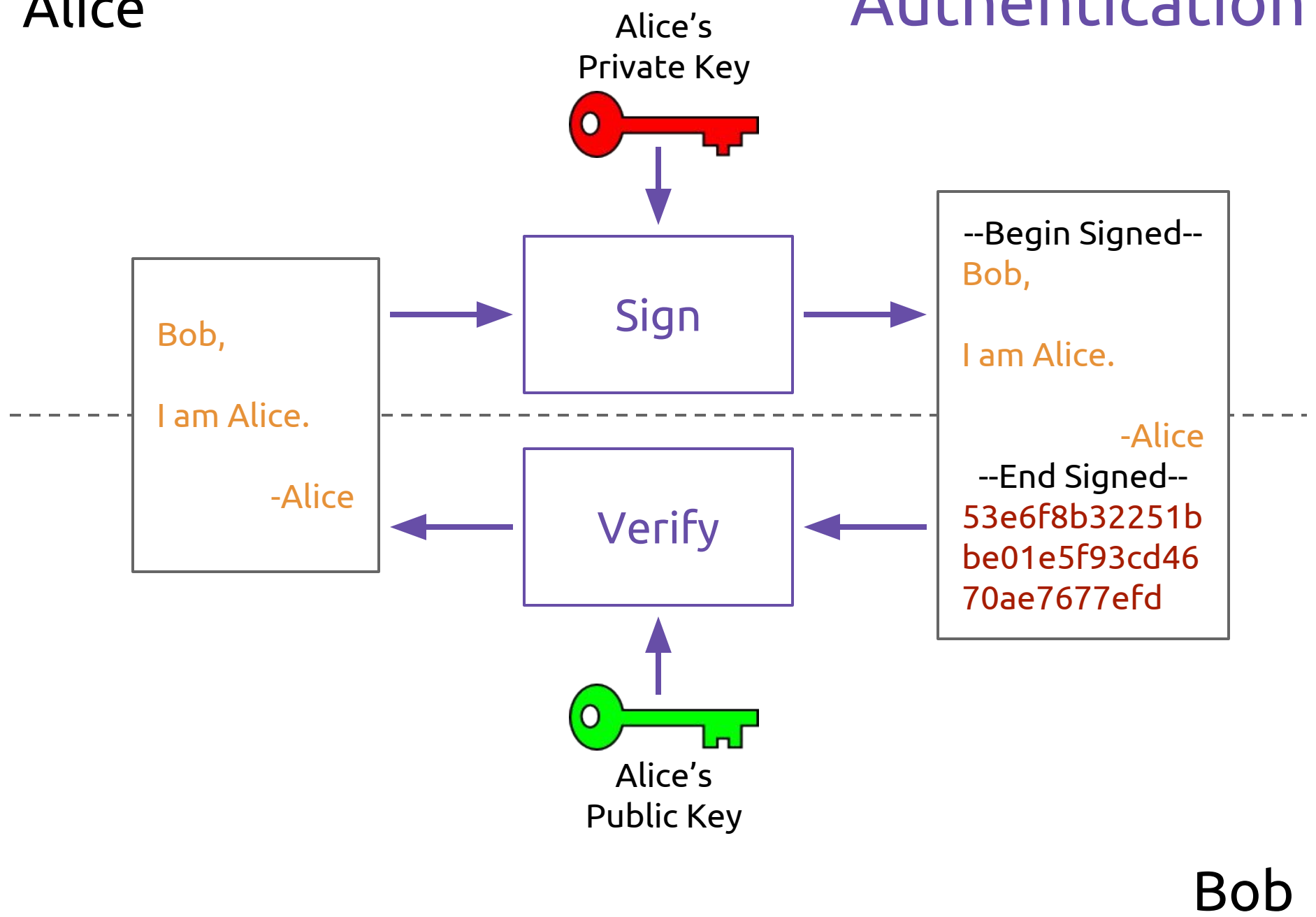
Signing



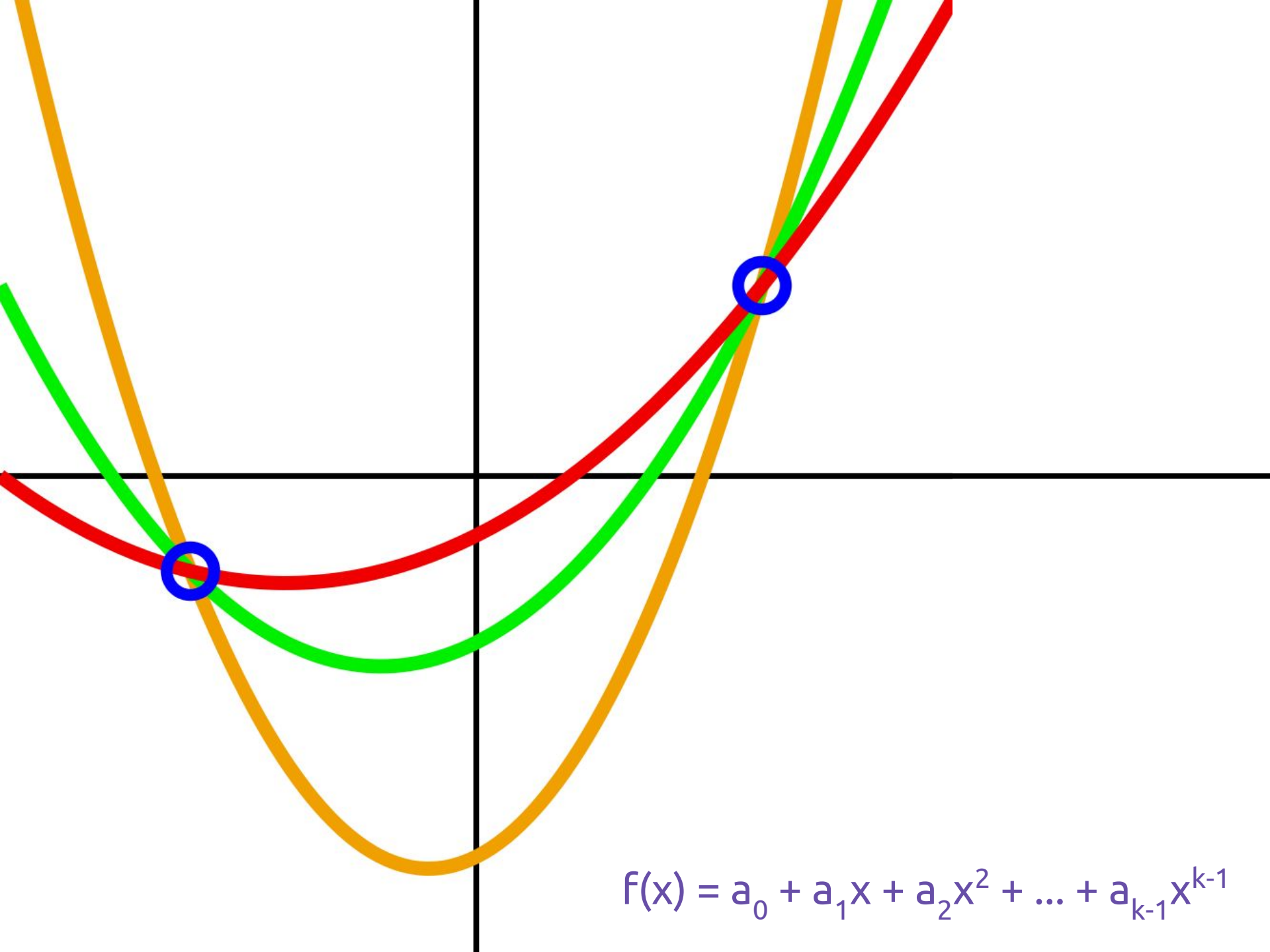
Bob

Alice

# Authentication



# Secret Sharing

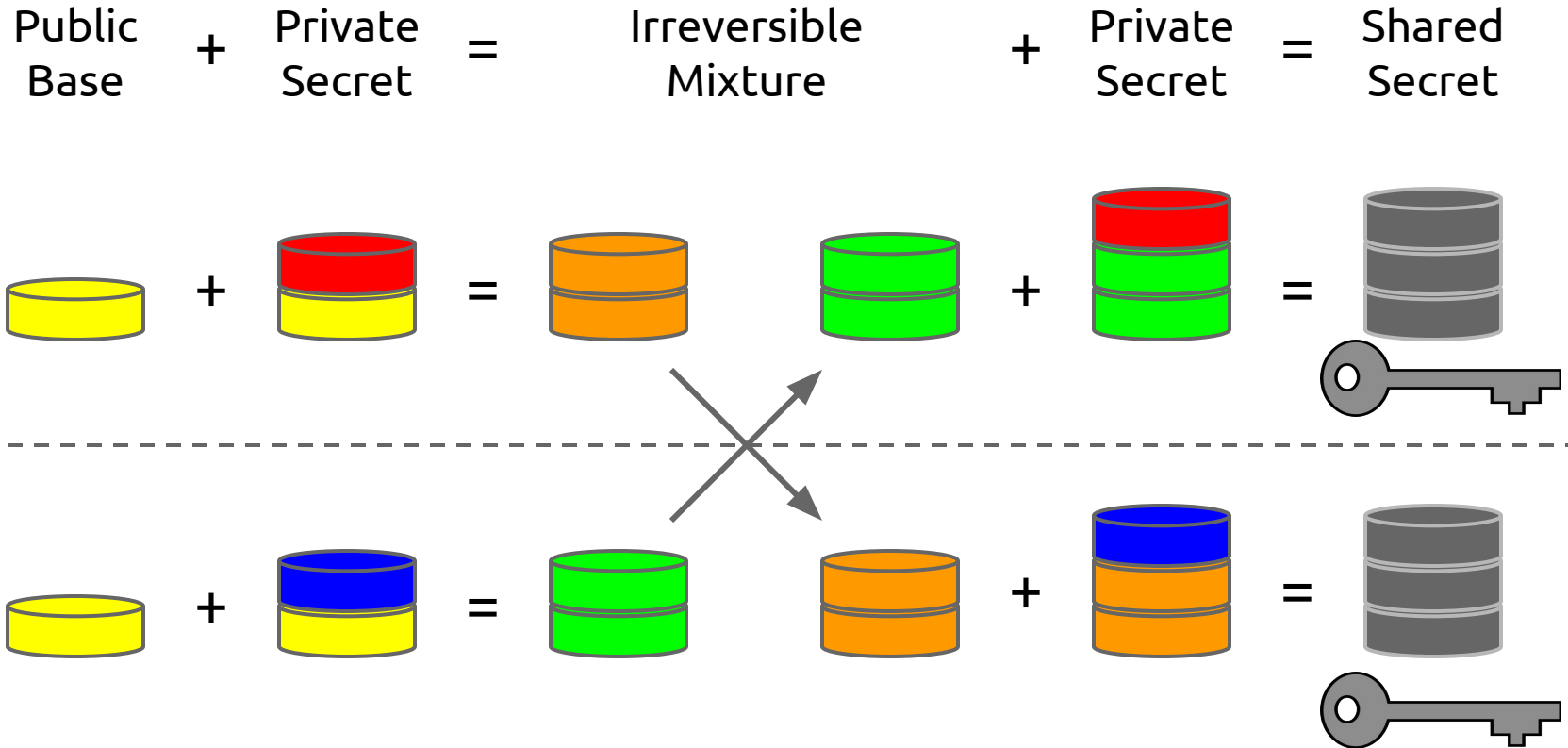


$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1}$$

# Diffie-Hellman

Alice

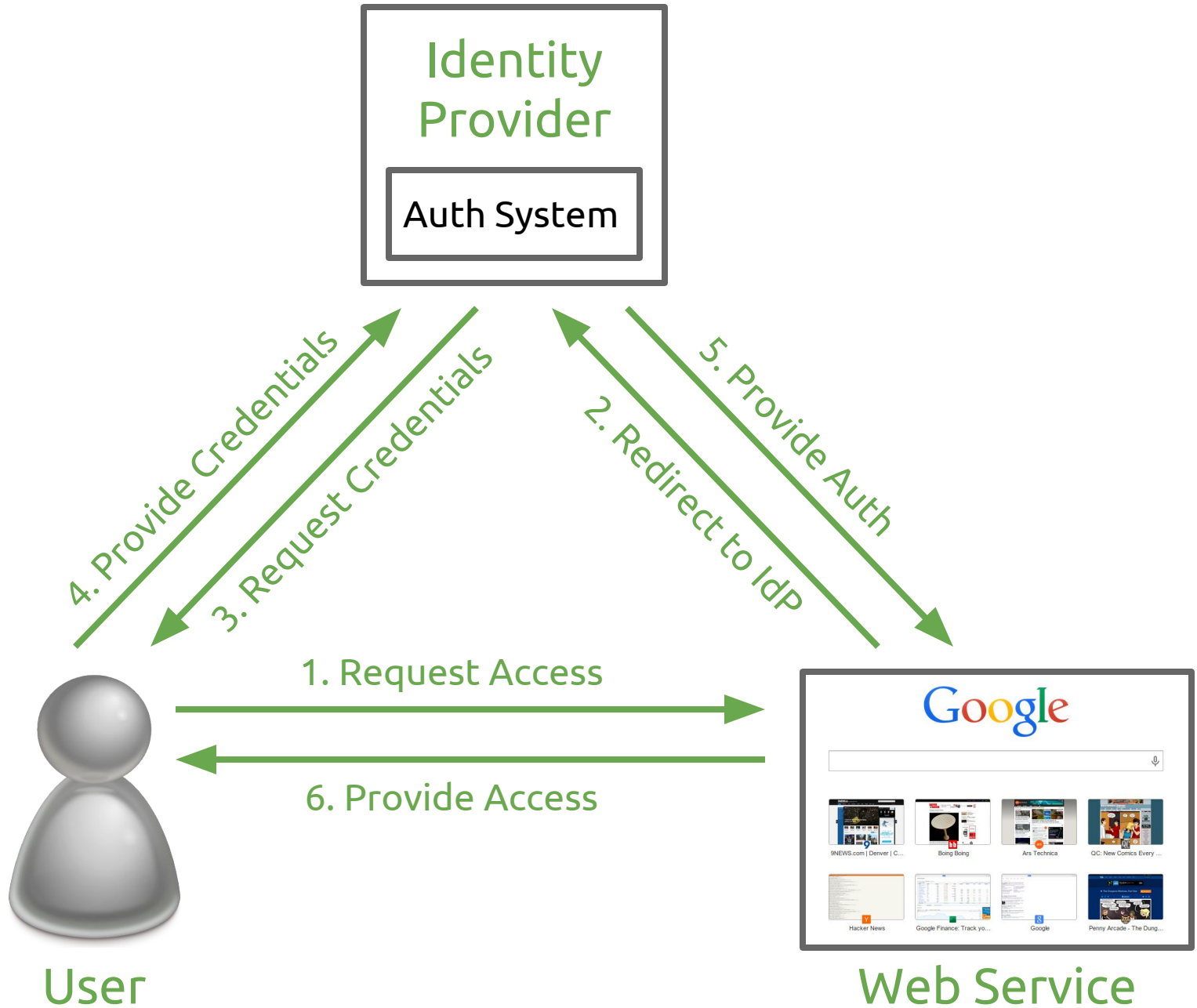
# Key Exchange



Bob

# Federated Access Control





# Revoking Access

# Example: Revoke Shared Access



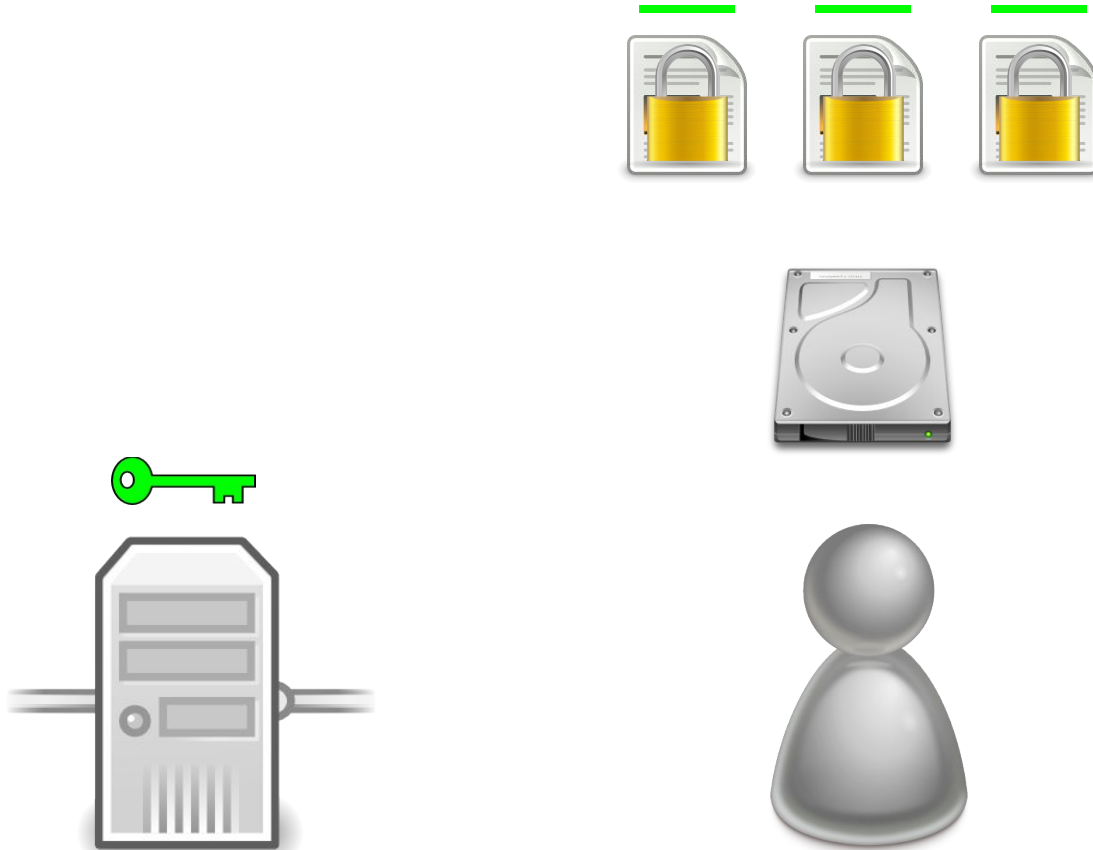
# Example: Revoke Shared Access



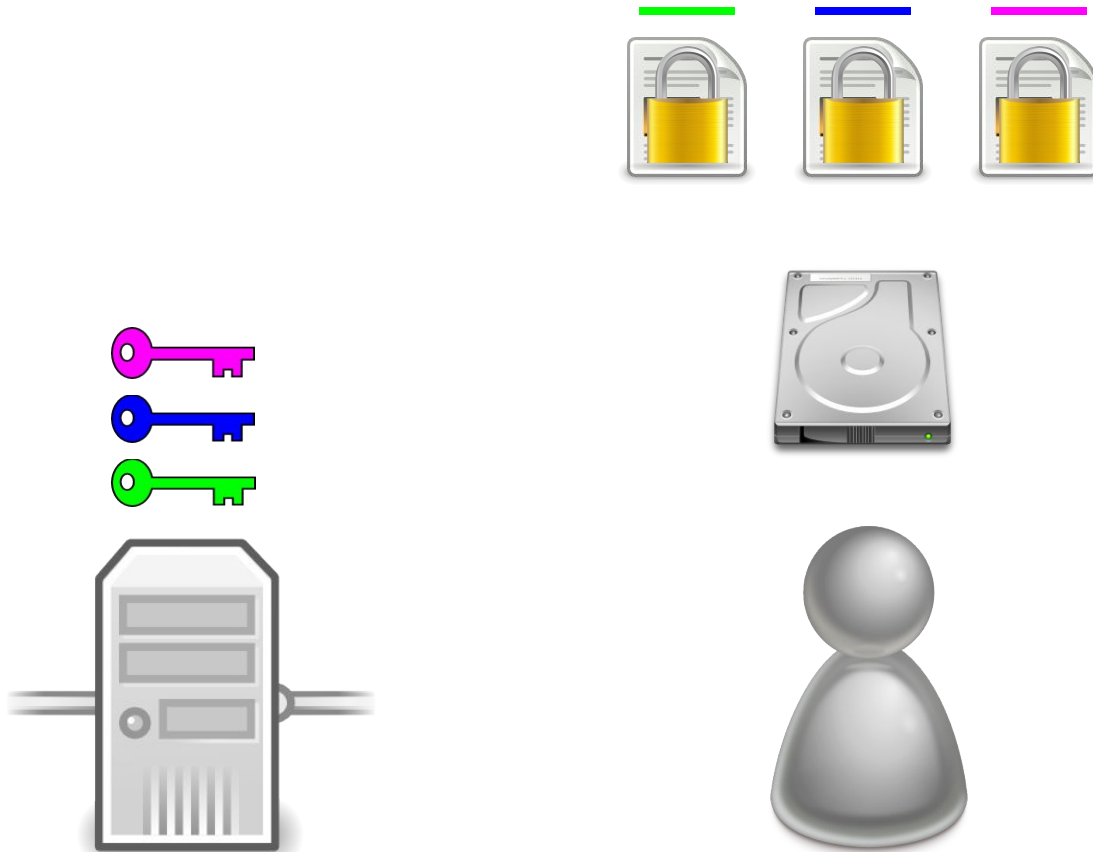
# Example: Revoke Shared Access



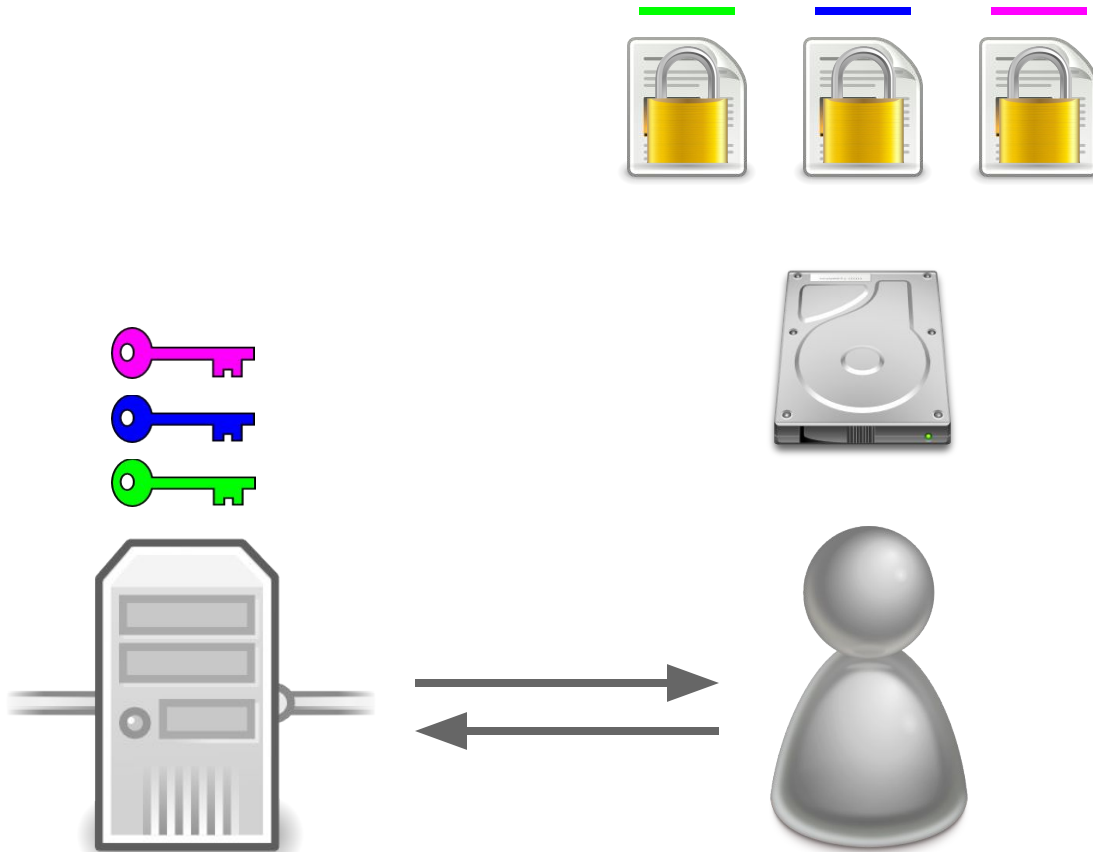
# Example: Revoke Shared Access



# Example: Revoke Shared Access

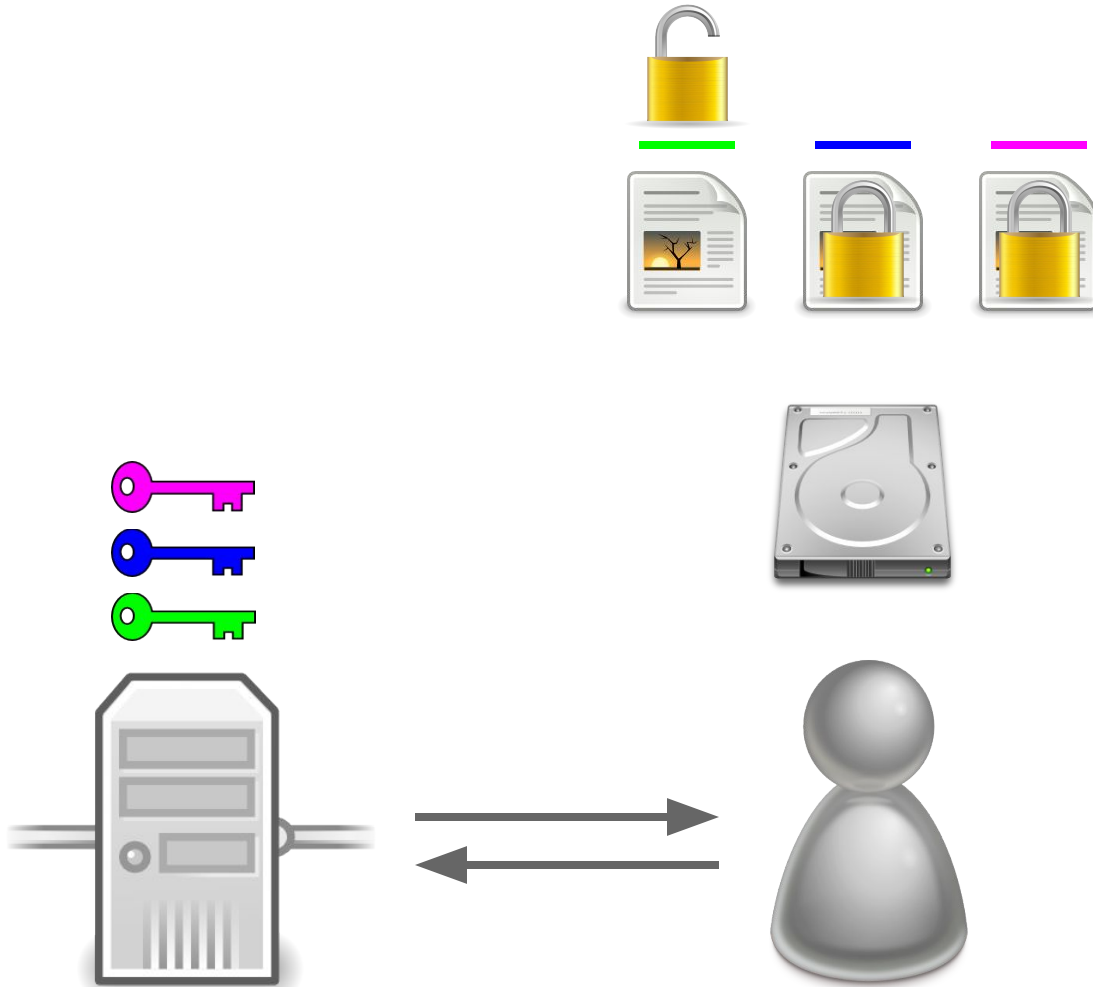


# Example: Revoke Shared Access

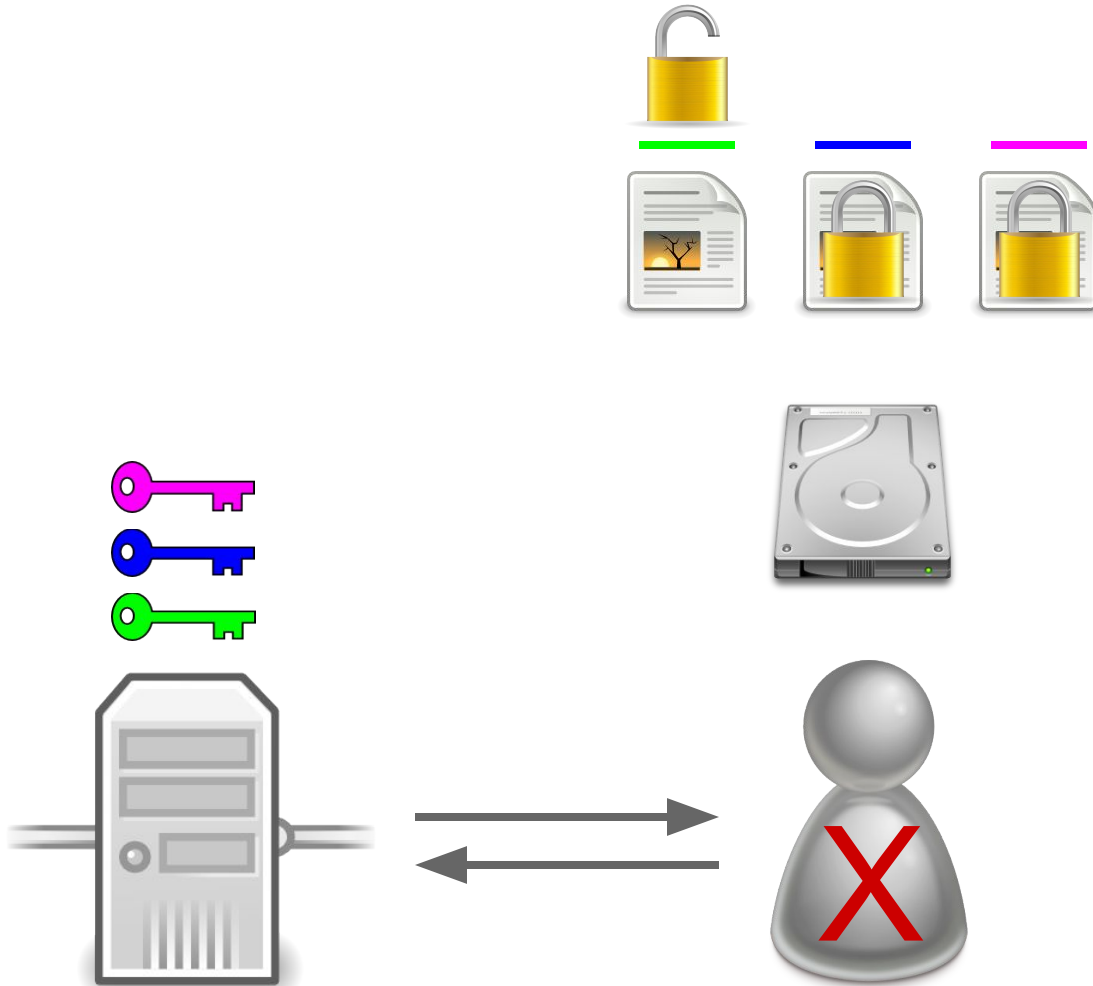




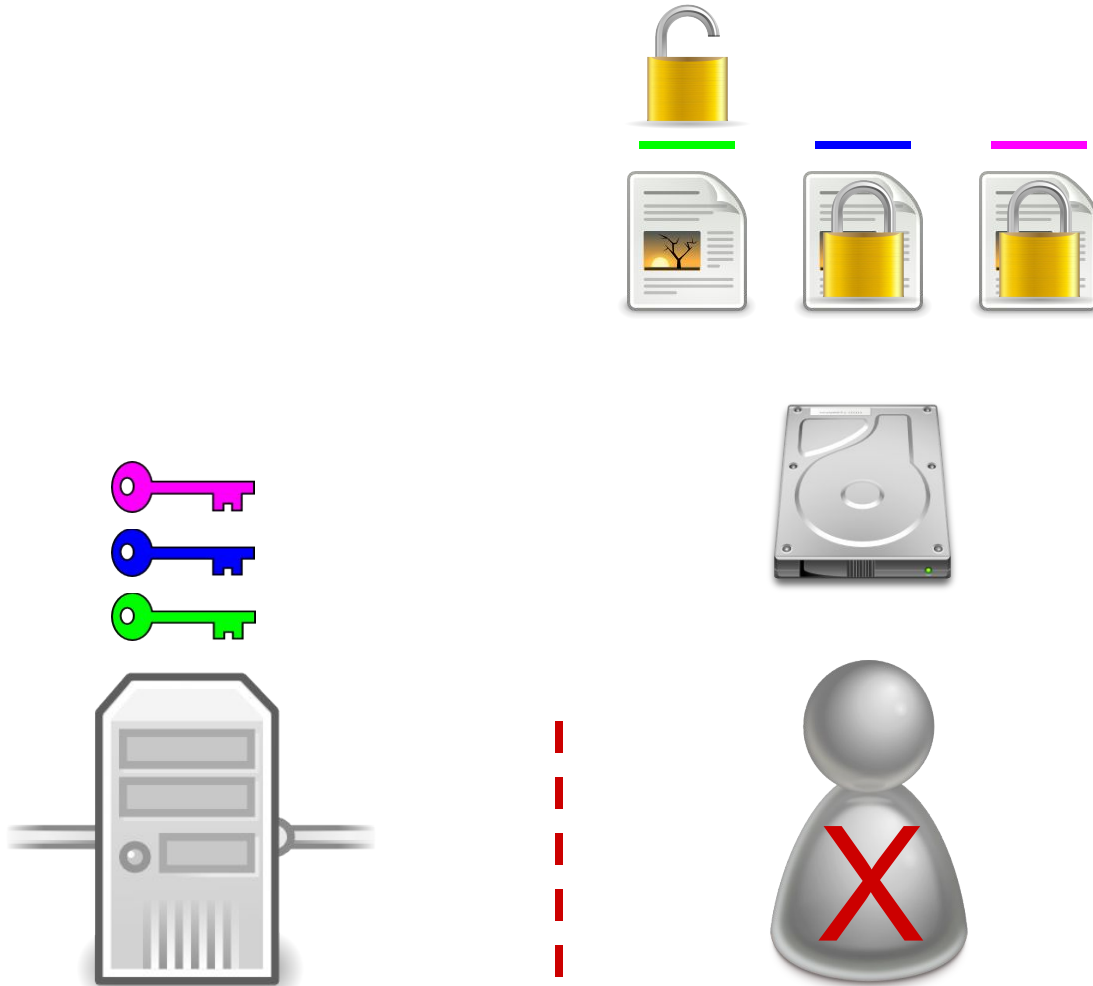
# Example: Revoke Shared Access



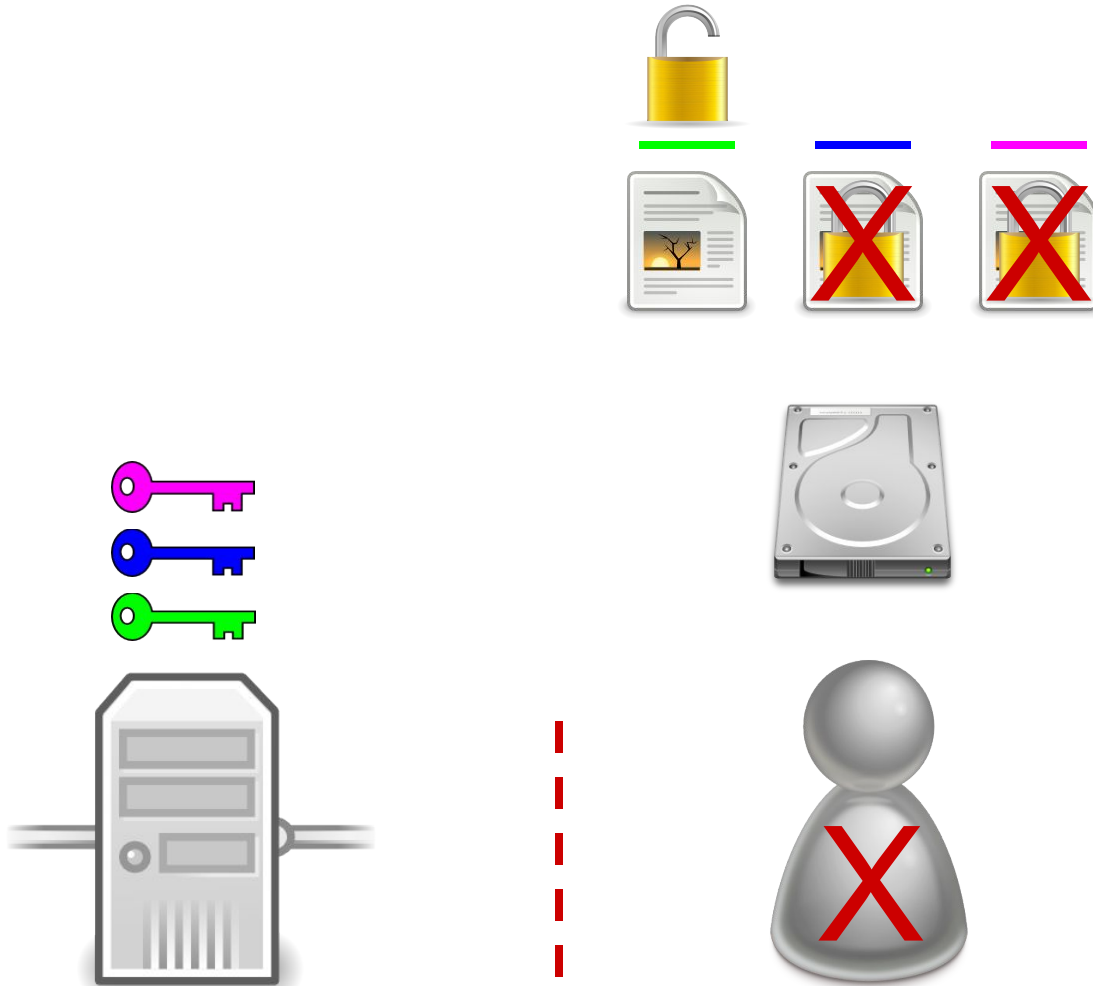
# Example: Revoke Shared Access



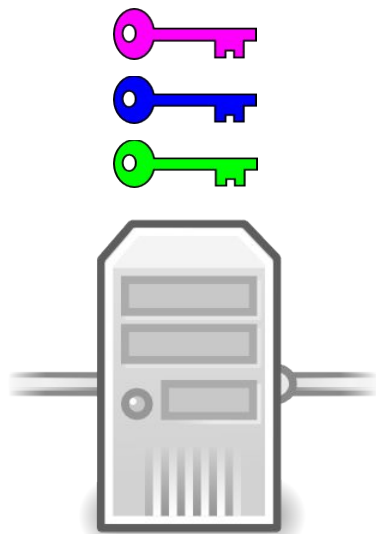
# Example: Revoke Shared Access






# Example: Revoke Shared Access

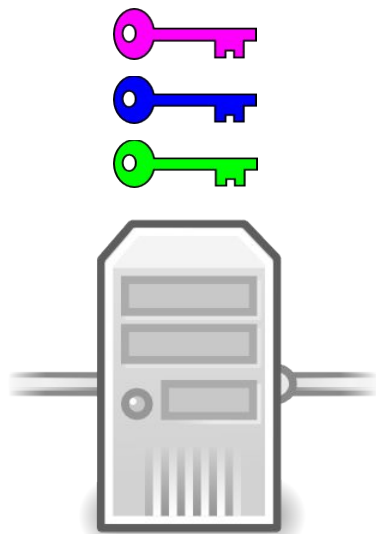





# Example: Revoke Shared Access



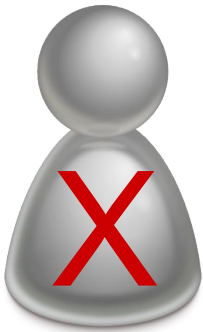
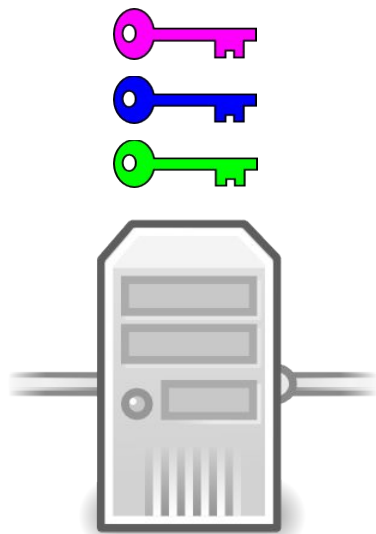
- 140813: Bob Accessed 
- 140906: Bob Accessed 
- 141003: Bob Accessed 



# Example: Revoke Shared Access



140813: Bob Accessed   
140906: Bob Accessed   
141003: Bob Accessed 

# Example: Revoke Shared Access



- 140813: Bob Accessed 
- 140906: Bob Accessed 
- 141003: Bob Accessed 