Key Management and Distribution

Cloud-Computing Security

Ruben Niederhagen

May 10, 2013

“Key management is the hardest part of cryptography and often the Achilles’ heel of an otherwise secure system.”

Bruce Schneier

1. Terminology

2. Symmetric Key Distribution
   - Using Symmetric Encryption
   - Using Asymmetric Encryption

3. Public Key Distribution
   - X.509 Certificates
   - Public Key Infrastructure
   - Risks

4. Key Management Strategies in the Cloud

Measures of Security

Core Principles of Information Security

- Confidentiality
- Integrity
- Availability

Measures of Security

Parkerian Hexad by Donn B. Parker in 2002

- Confidentiality
- Integrity
- Availability

Measures of Security

Parkerian Hexad by Donn B. Parker in 2002

- Confidentiality
- Integrity
- Availability
- Possession or Control
- Utility
- Authenticity

Measures of Security

Parkerian Hexad by Donn B. Parker in 2002

- Confidentiality
- Integrity
- Availability
- Possession or Control
- Utility
- Authenticity
  - Non-Repudiation
  - Plausible Deniability

Measures of Security

Parkerian Hexad by Donn B. Parker in 2002

- Confidentiality
- Integrity
- Availability
- Possession or Control
- Utility
- Authenticity
  - Non-Repudiation
  - Plausible Deniability

Anonymity, (Perfect) Forward Secrecy, Trust, ...

Building Blocks

Symmetric Encryption

Using the same shared key for encryption and decryption:

$$C = E(K, P) \qquad P = E(K, C)$$

Examples: Twofish, Serpent, AES (Rijndael), Blowfish, 3DES, ...

Building Blocks

Symmetric Encryption

Using the same shared key for encryption and decryption:

$$C = E(K, P) \qquad P = E(K, C)$$

Examples: Twofish, Serpent, AES (Rijndael), Blowfish, 3DES, ...

Asymmetric Encryption

Using public key for encryption and private key for decryption:

$$C = E(K_{pub}, P) \qquad P = E(K_{priv}, C)$$

Examples: RSA, McEliece, ElGamal, ...

Why do we need key distribution?

Secure communication requires shared, “a priori” knowledge.

How do we achieve this knowledge?

By using some kind of key-distribution scheme!

Key Distribution Using Symmetric Encryption

Given parties A and B, there are several alternatives for key distribution

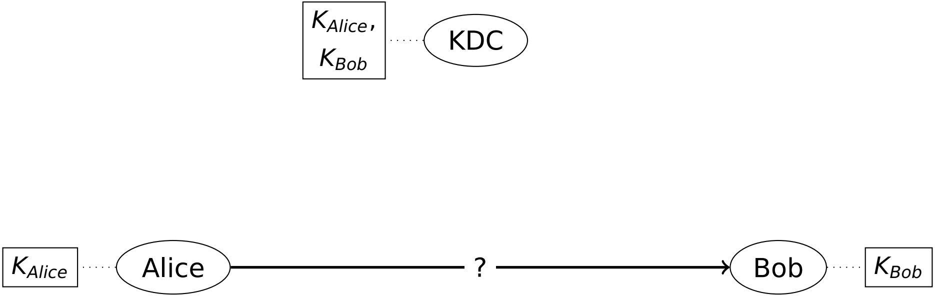
1. A can select key and physically deliver it to B.
2. A third party can select and physically deliver the key to A and B.
3. If A and B have communicated previously, they can use the previous key to encrypt a new key.
4. If A and B have secure communication channel with a third party C, C can relay the key between A and B.

Key Distribution Using Symmetric Encryption

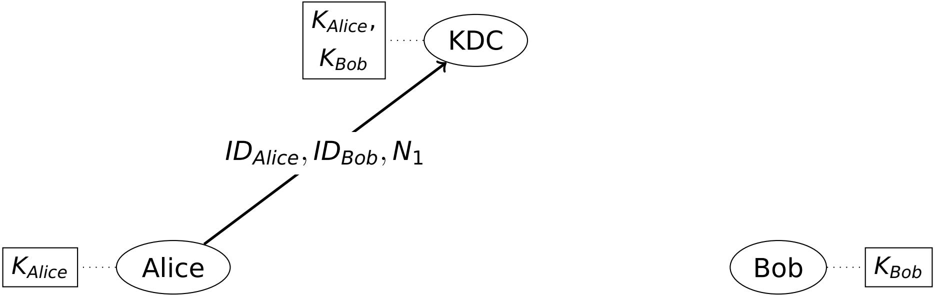
Key Hierarchy

- typically have a hierarchy of keys
- session key
  - temporary key
  - used for encryption of data between users for one logical session
  - discarded after usage
- master key
  - longterm key
  - used to encrypt session keys
  - shared by user and key distribution center

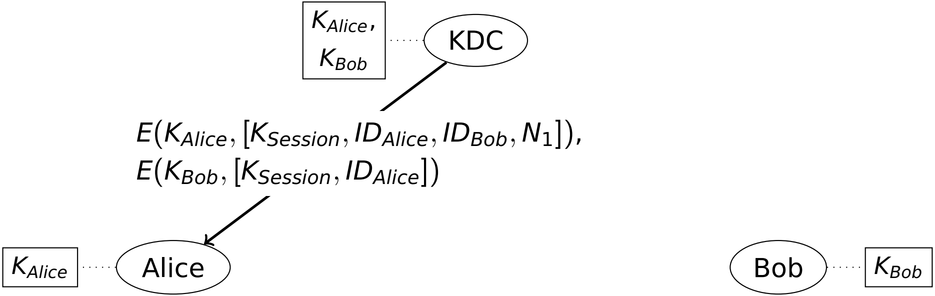
Key Distribution Center (KDC)



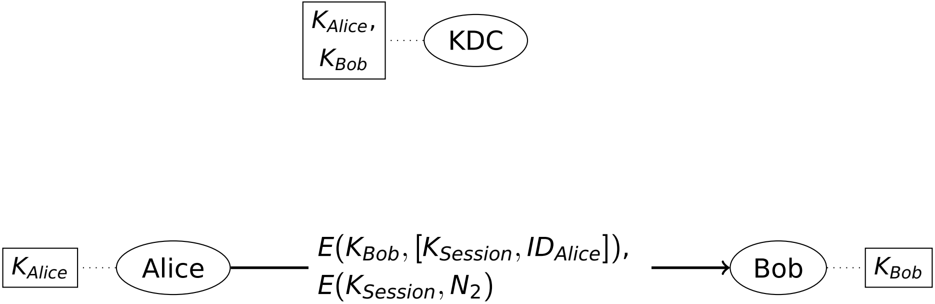
Key Distribution Center (KDC)



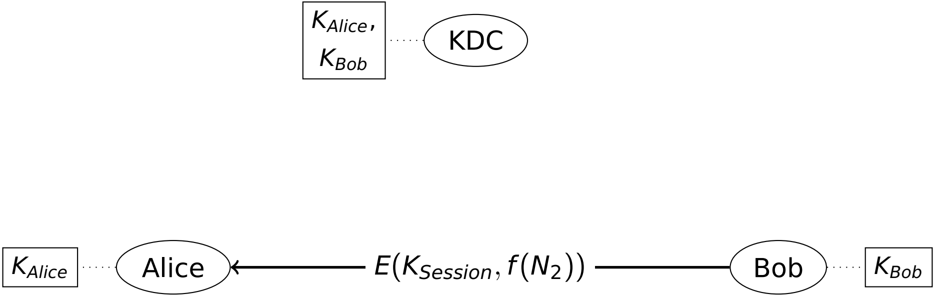
Key Distribution Center (KDC)



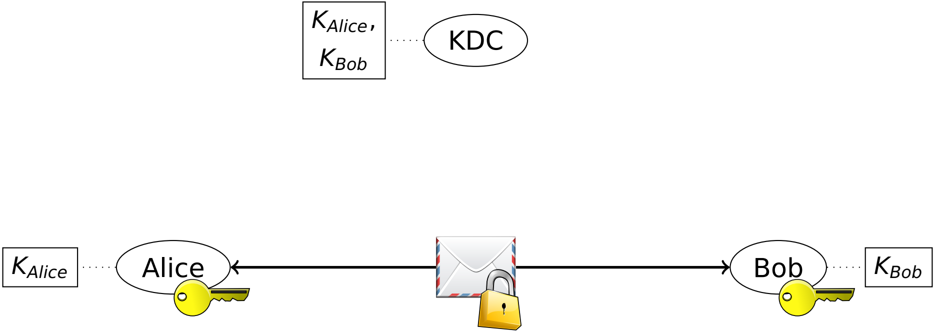
Key Distribution Center (KDC)



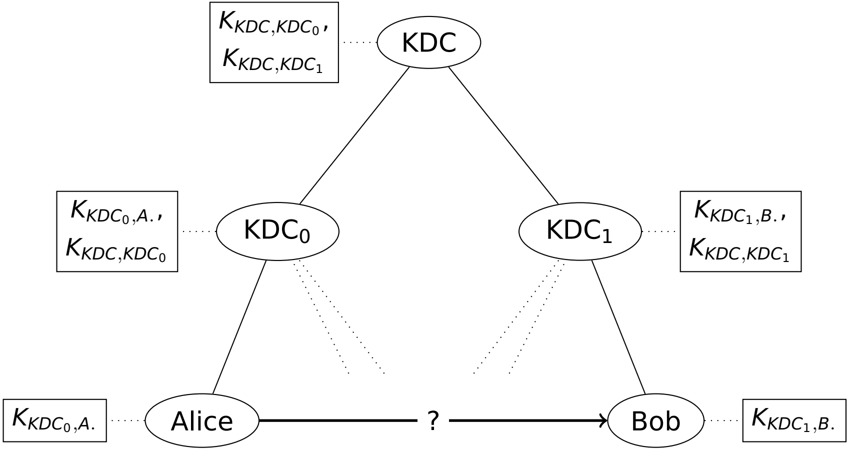
Key Distribution Center (KDC)



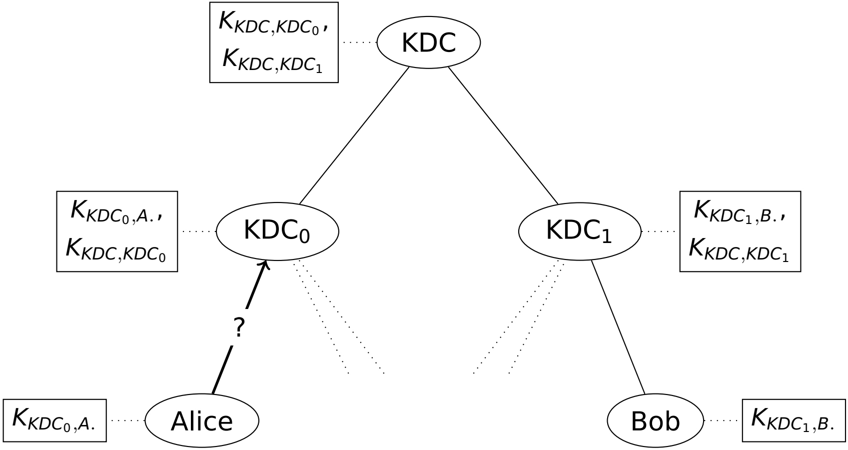
Key Distribution Center (KDC)



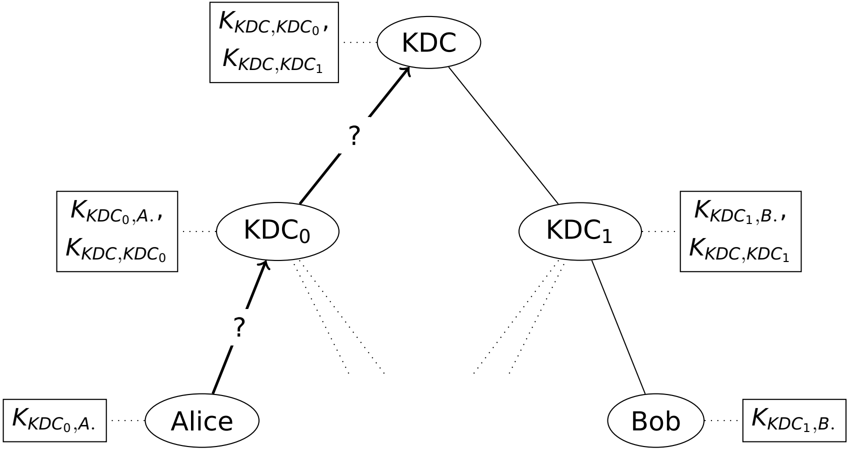
Hierarchical Key Control



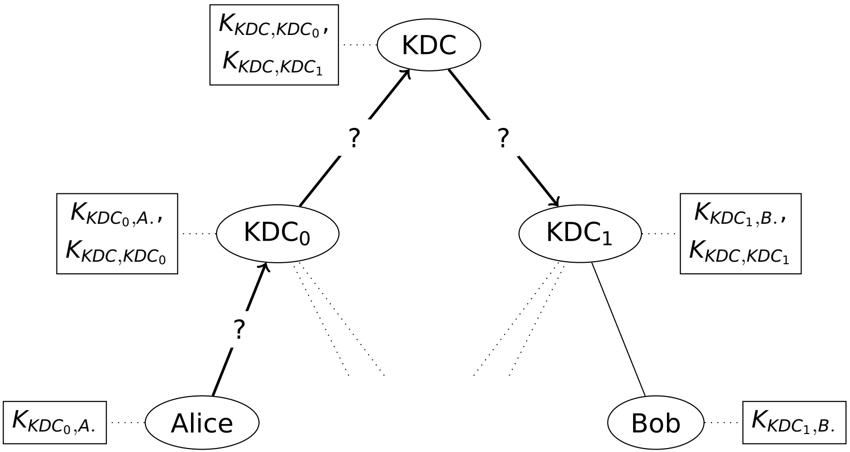
Hierarchical Key Control



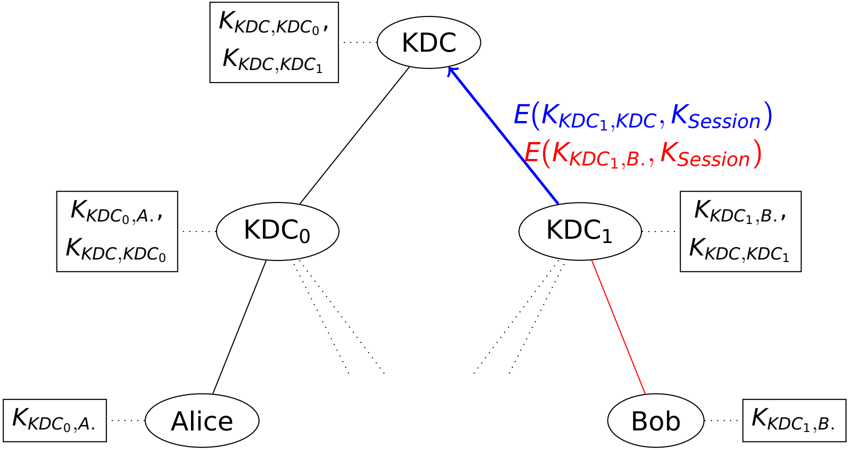
Hierarchical Key Control



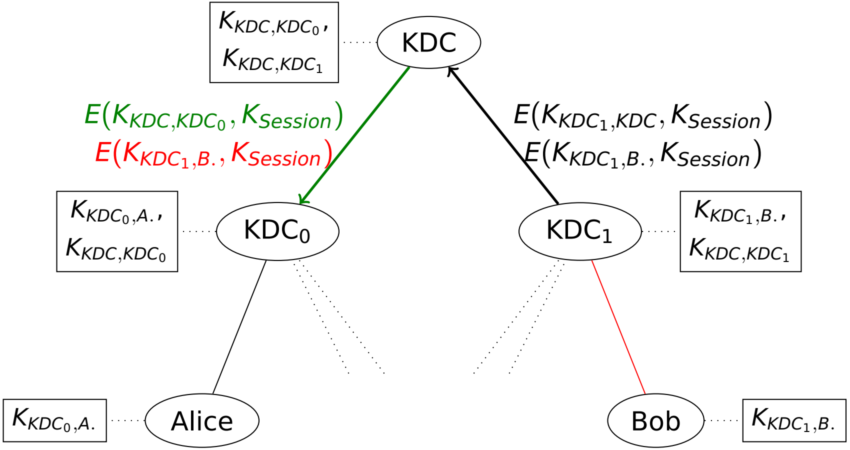
Hierarchical Key Control



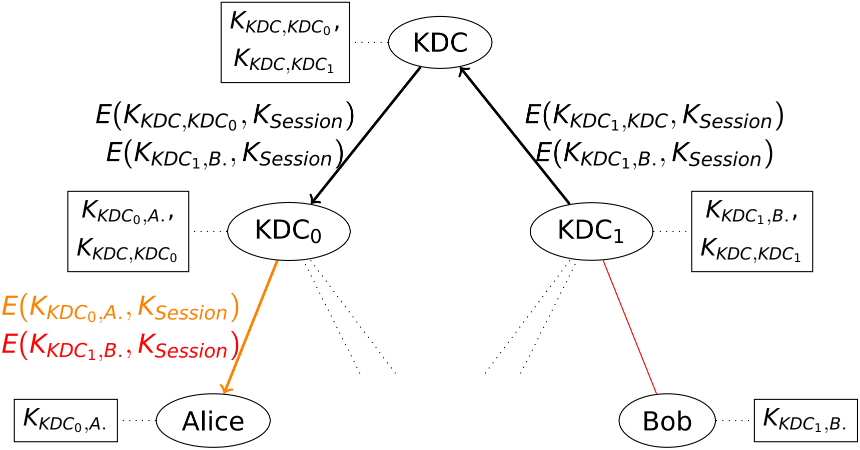
Hierarchical Key Control



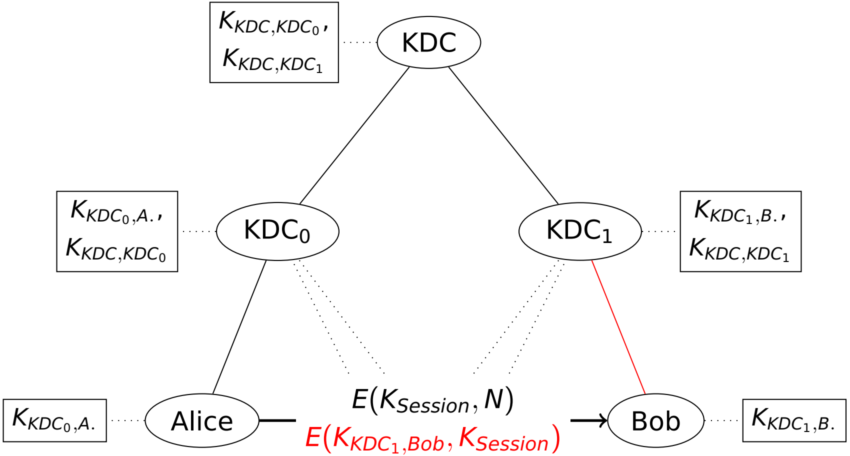
Hierarchical Key Control



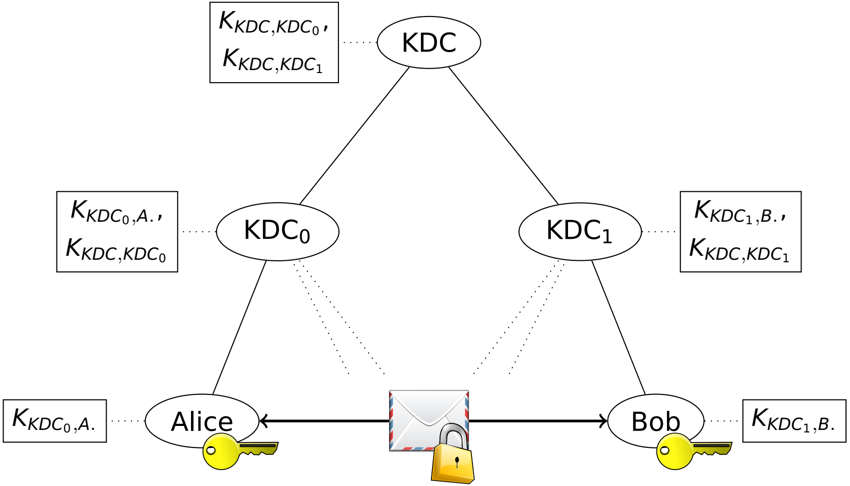
Hierarchical Key Control



Hierarchical Key Control



Hierarchical Key Control



Session-Key Lifetime

Make Lifetime as Short as Possible

- benefit:
  - reduced attack surface
  - less information compromised in case encryption is broken
- disadvantage:
  - requires to obtain keys more often
  - requires more time

Session-Key Lifetime

Make Lifetime as Short as Possible

- benefit:
  - reduced attack surface
  - less information compromised in case encryption is broken
- disadvantage:
  - requires to obtain keys more often
  - requires more time

Connection-Oriented Protocols

- naturally choice: one key per connection
- re-key if connection is maintained for too long

Session-Key Lifetime

Make Lifetime as Short as Possible

- benefit:
  - reduced attack surface
  - less information compromised in case encryption is broken
- disadvantage:
  - requires to obtain keys more often
  - requires more time

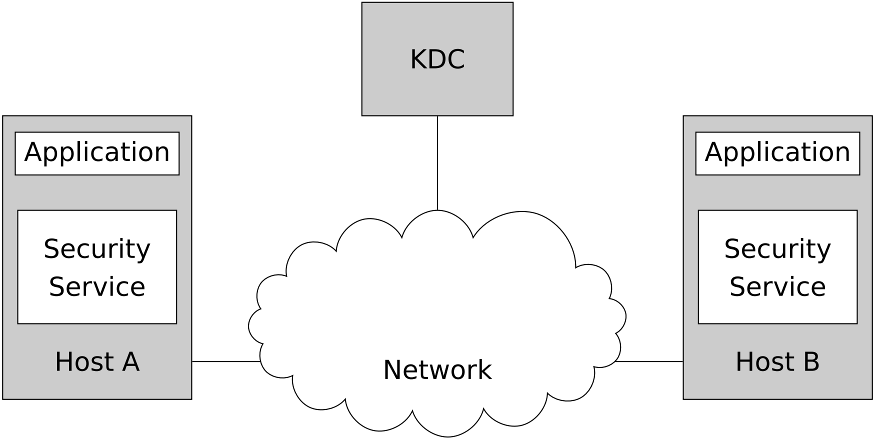
Connection-Oriented Protocols

- naturally choice: one key per connection
- re-key if connection is maintained for too long

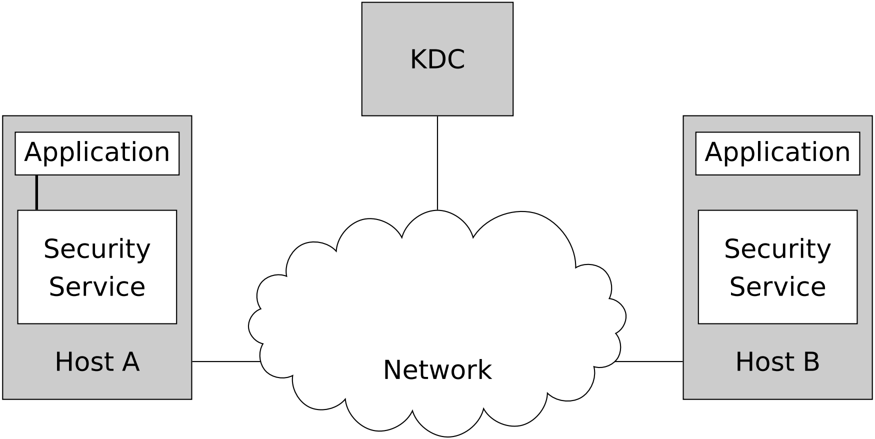
Connectionless Protocols

- naturally choice: one key per exchange
- re-key after a certain amount of time

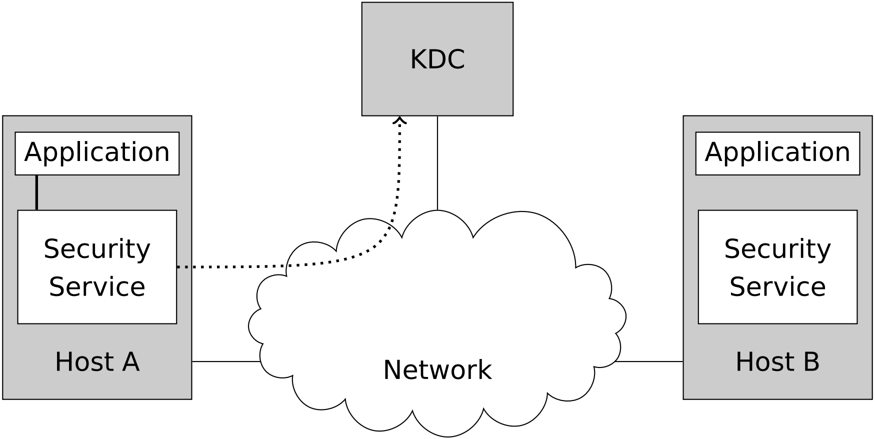
Transparent Key Control Scheme



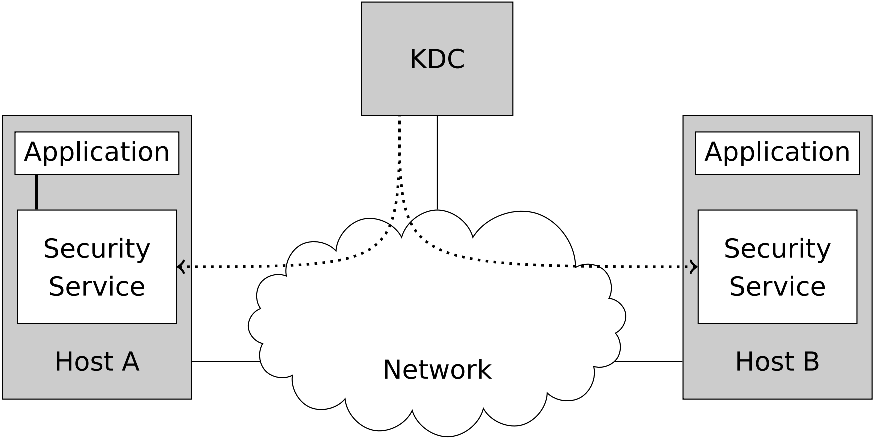
Transparent Key Control Scheme



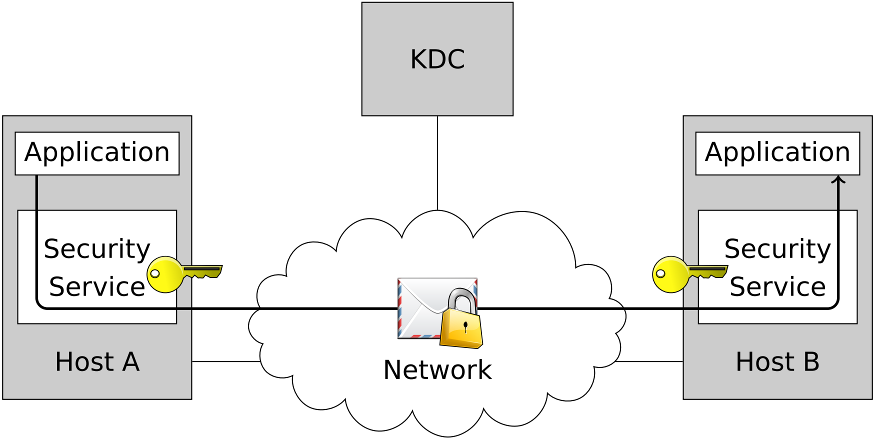
Transparent Key Control Scheme



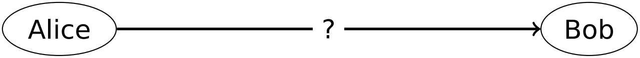
Transparent Key Control Scheme



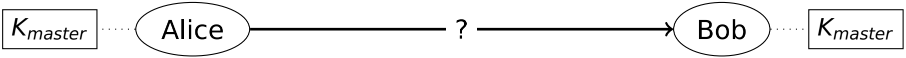
Transparent Key Control Scheme



Decentralized Key Control

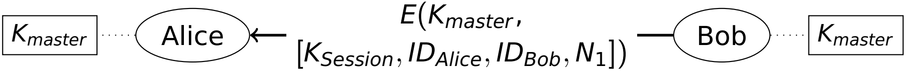


Decentralized Key Control



Decentralized Key Control

Decentralized Key Control



Decentralized Key Control

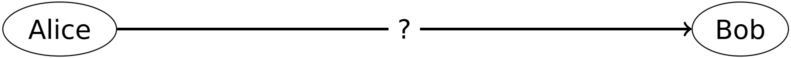
Requires $n(n-1)/2$ master keys for n nodes.

Key Distribution Using Asymmetric Encryption

Features of Asymmetric Encryption Schemes

- typically slower than symmetric schemes
- can be used to encrypt symmetric keys for distribution
- public key can be distributed openly

Simple Secret Key Distribution



Simple Secret Key Distribution

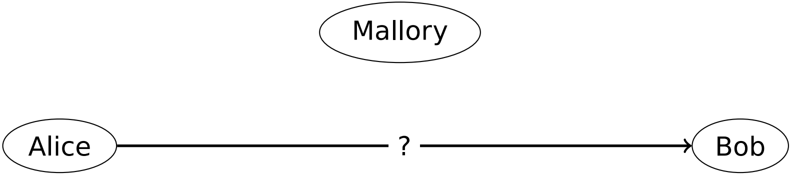
$$\text{Alice} \xrightarrow{\quad K_{pub\ Alice},\, ID_{Alice} \quad} \text{Bob}$$

Simple Secret Key Distribution

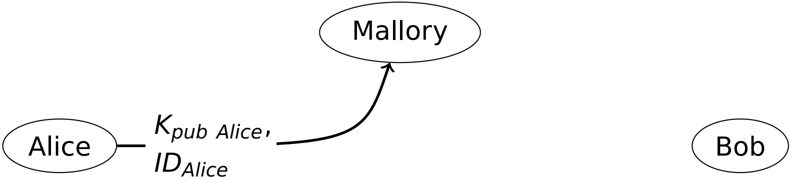
$$\text{Alice} \xleftarrow{\quad E(K_{pub\ Alice}, K_{Session}) \quad} \text{Bob}$$

Simple Secret Key Distribution

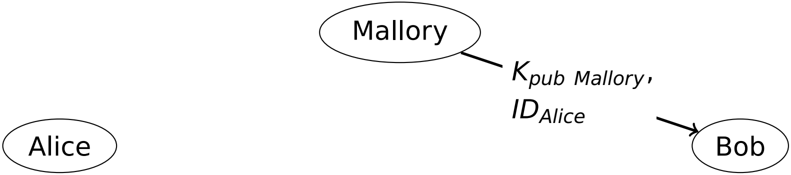
Simple Secret Key Distribution



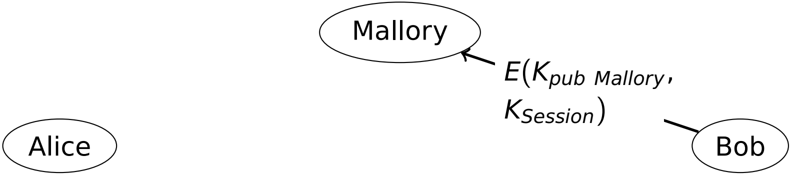
Simple Secret Key Distribution



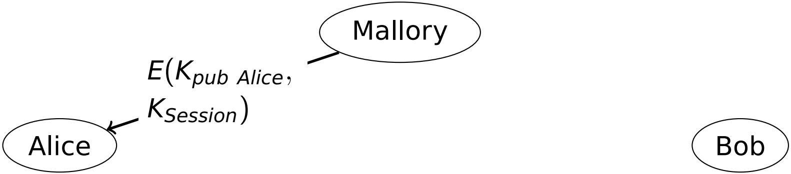
Simple Secret Key Distribution



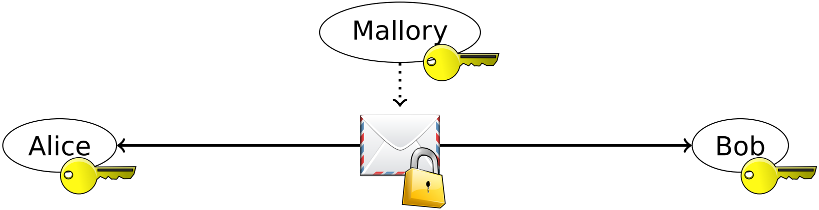
Simple Secret Key Distribution



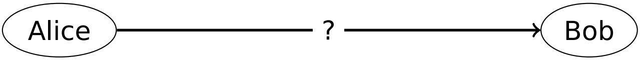
Simple Secret Key Distribution



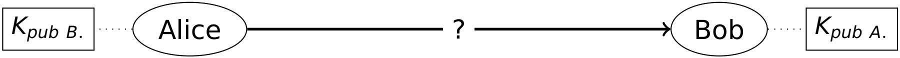
Simple Secret Key Distribution



Secret Key Distribution with Confidentiality and Authentication



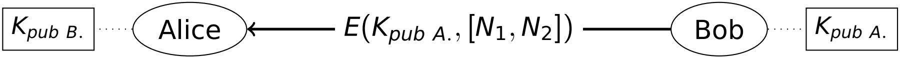
Secret Key Distribution with Confidentiality and Authentication



Secret Key Distribution with Confidentiality and Authentication

$K_{pub\ B.}$ ⋯ Alice $\xrightarrow{E(K_{pub\ B.}, [ID_{A.}, N_1])}$ Bob ⋯ $K_{pub\ A.}$

Secret Key Distribution with Confidentiality and Authentication



Secret Key Distribution with Confidentiality and Authentication

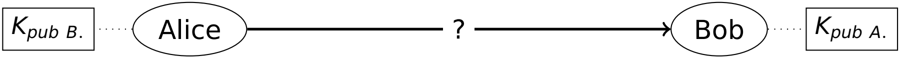
$\boxed{K_{pub\ B.}}$ ⋯ Alice $\xrightarrow{E(K_{pub\ B.}, [N_2, K_{Session}])}$ Bob ⋯ $\boxed{K_{pub\ A.}}$

Secret Key Distribution with Confidentiality and Authentication

Secret Key Distribution with Confidentiality and Authentication

Use signatures!

Secret Key Distribution with Confidentiality and Authentication



Secret Key Distribution with Confidentiality and Authentication

$K_{pub\ B.}$ ······ Alice ——

$E(K_{pub\ Bob}, K_{Session})$,
N, ID_{Alice},
$E(K_{priv\ Alice}, f([...]))$

$\longrightarrow$ Bob ······ $K_{pub\ A.}$

Secret Key Distribution with Confidentiality and Authentication

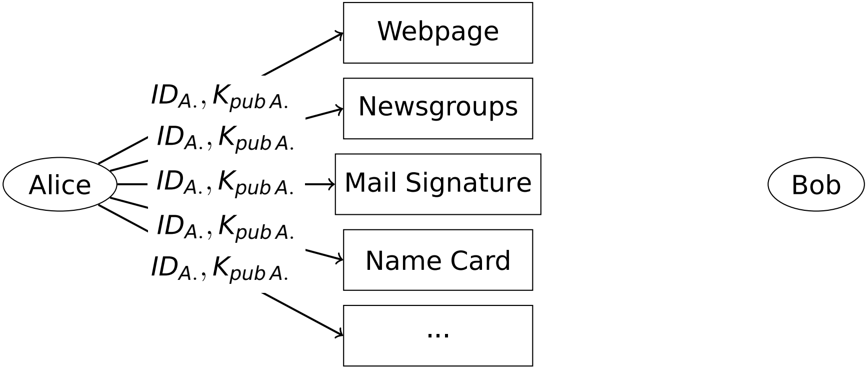
Secret Key Distribution with Confidentiality and Authentication

How do we distribute the public keys?

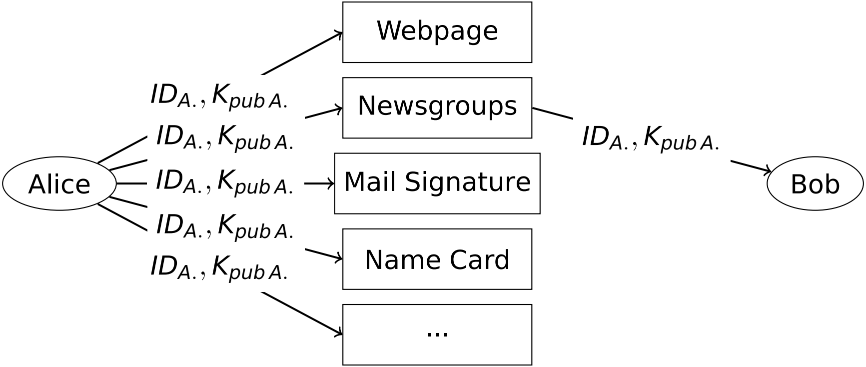
Public Key Distribution

General Schemes:

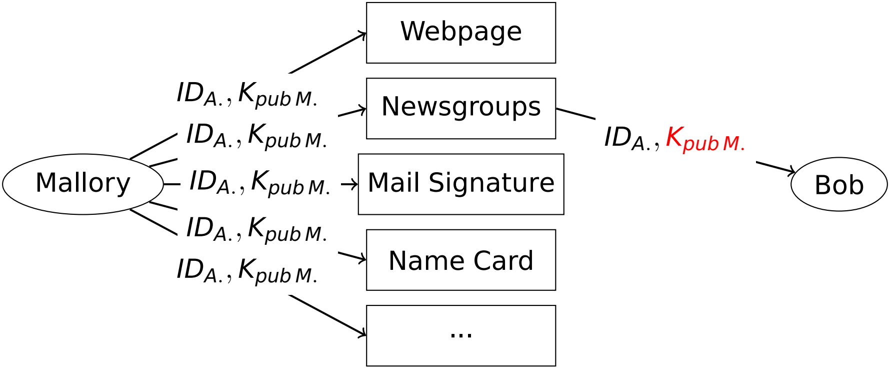
- public announcement
- publicly available directory
- public-key authority
- public-key certificates


Webpage
Newsgroups
Mail Signature
Name Card
...
Alice
Bob
$ID_{A.}, K_{pub\,A.}$
$ID_{A.}, K_{pub\,A.}$
$ID_{A.}, K_{pub\,A.}$
$ID_{A.}, K_{pub\,A.}$
$ID_{A.}, K_{pub\,A.}$

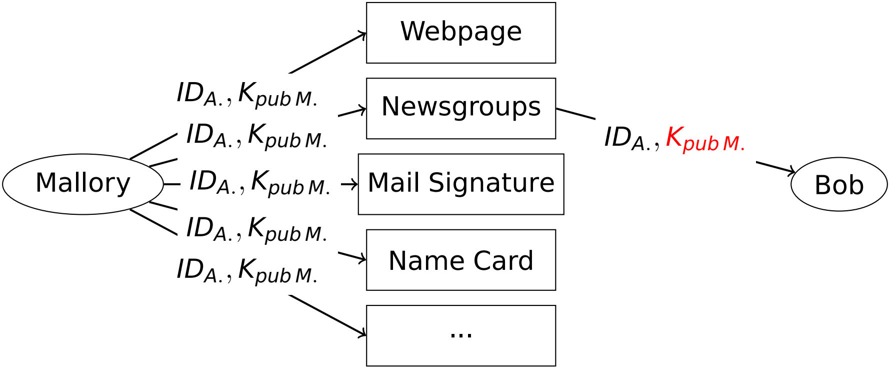
Public Announcement



Public Announcement



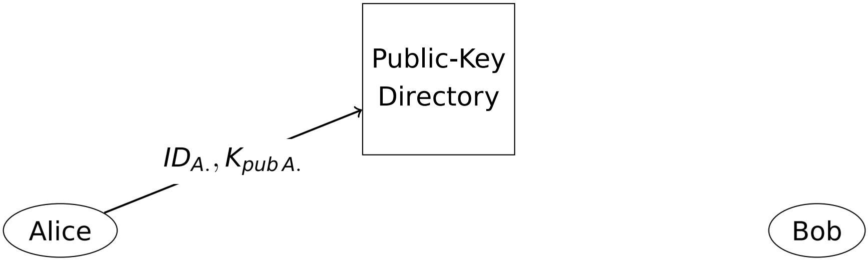
Public Announcement



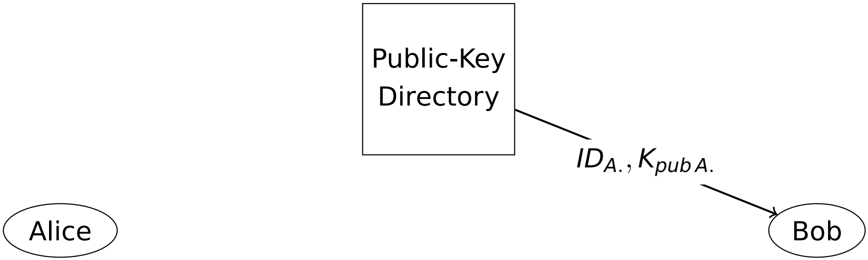
Easy to forge!

Publicly Available Directory

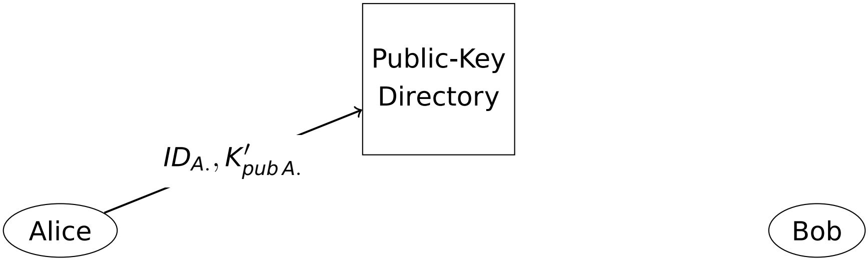
Publicly Available Directory



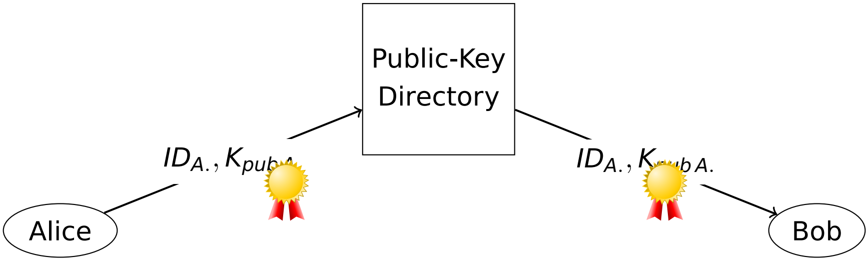
Publicly Available Directory



Publicly Available Directory

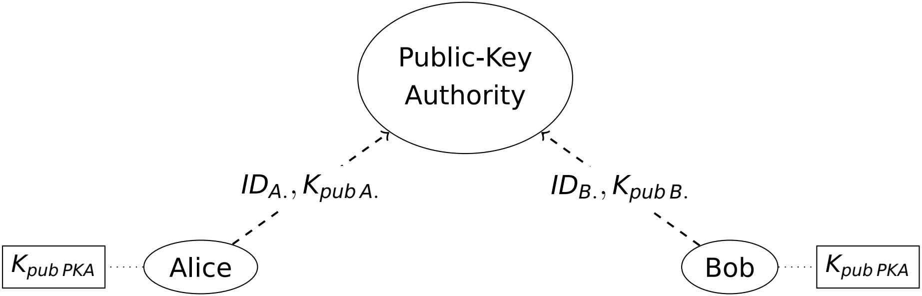


Publicly Available Directory

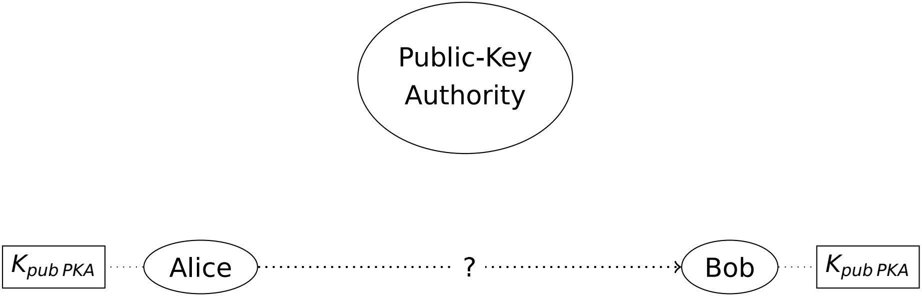


Communication with public-key directory must be authenticated, acknowledged, and protected against replay attacks!

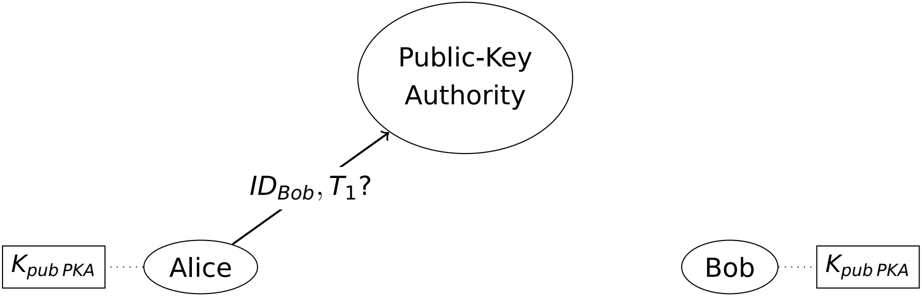
Public-Key Authority



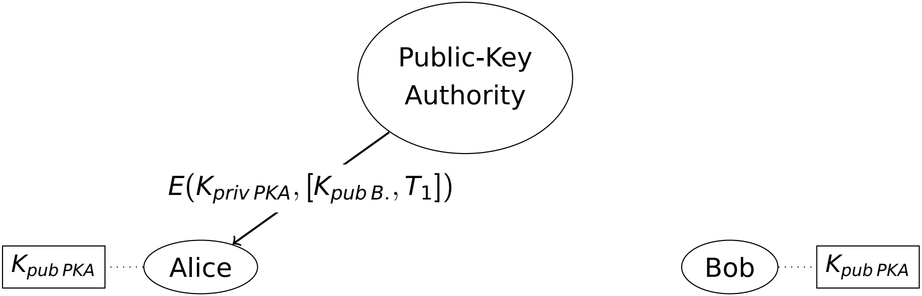
Public-Key Authority



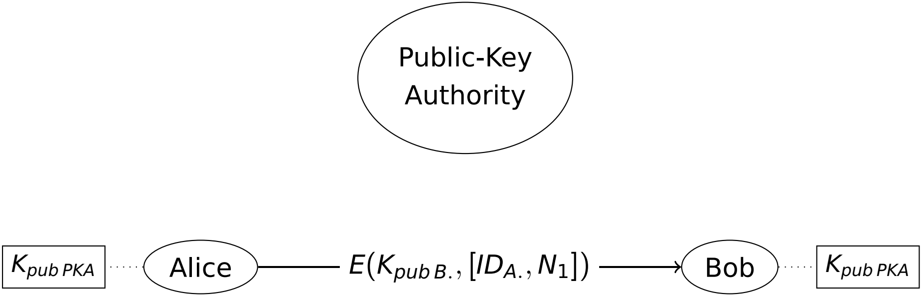
Public-Key Authority



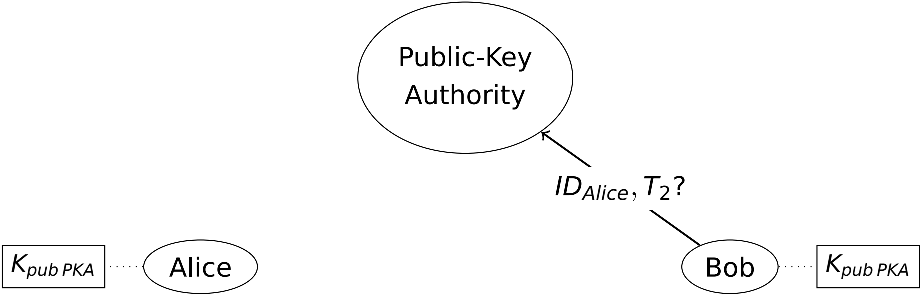
Public-Key Authority



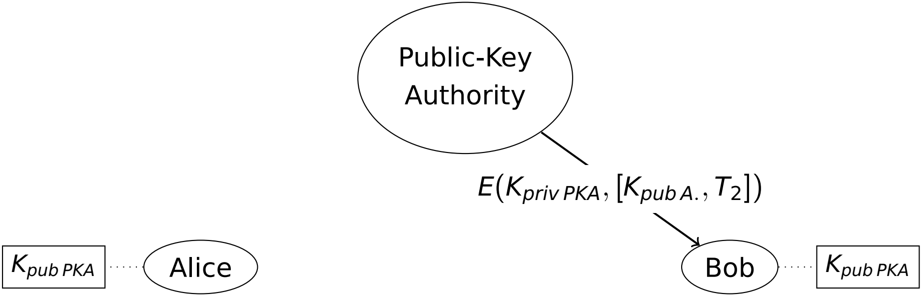
Public-Key Authority



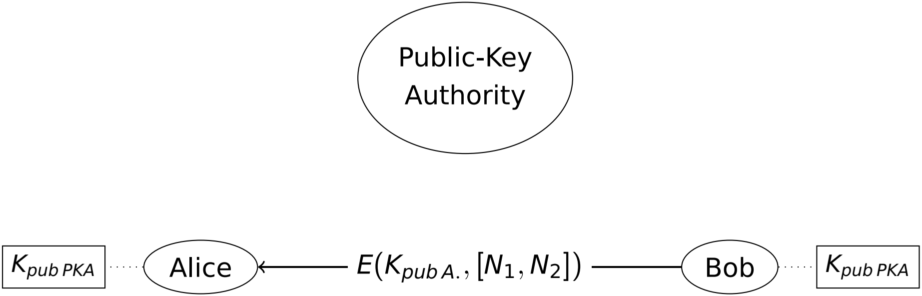
Public-Key Authority



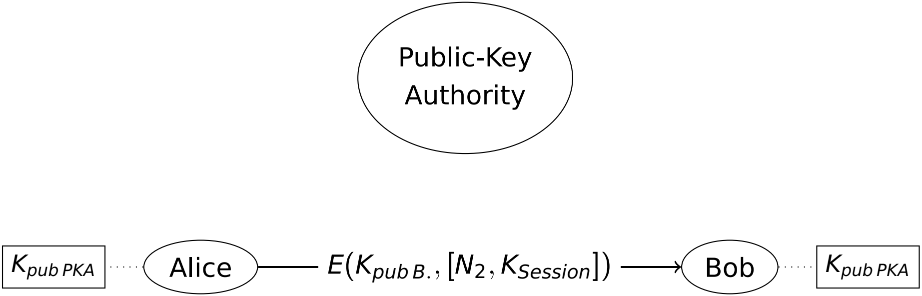
Public-Key Authority



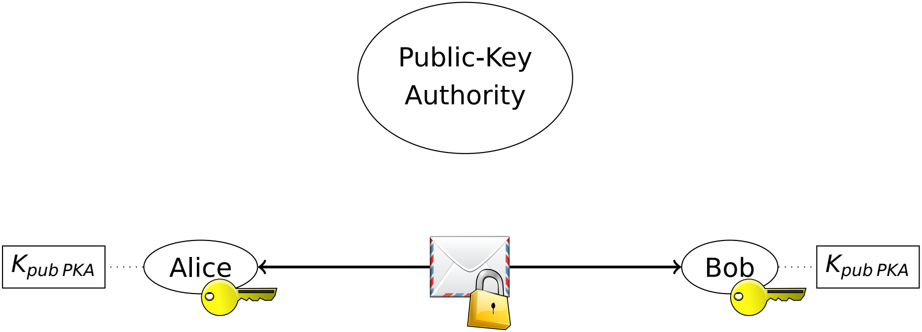
Public-Key Authority



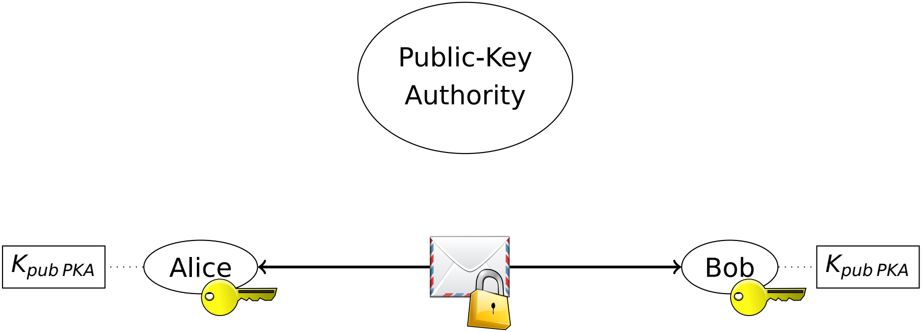
Public-Key Authority



Public-Key Authority



Public-Key Authority

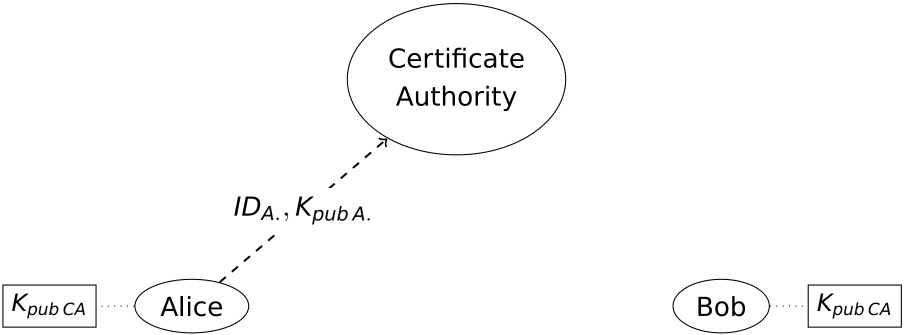


Long latencies due to communication with PKA!

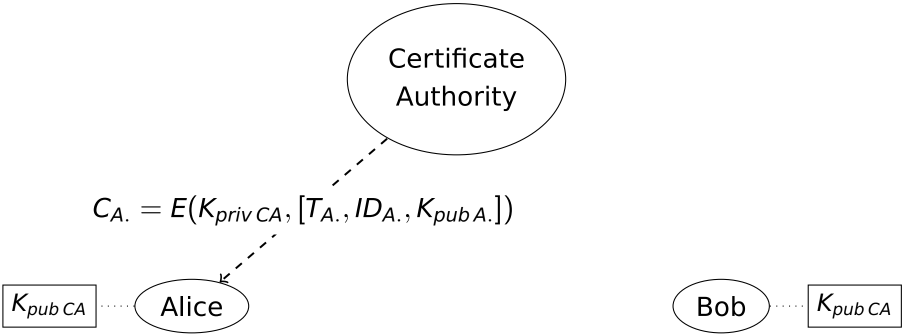
- Alice and Bob may cache public keys.
- Use certificates...

Public-Key Certificates

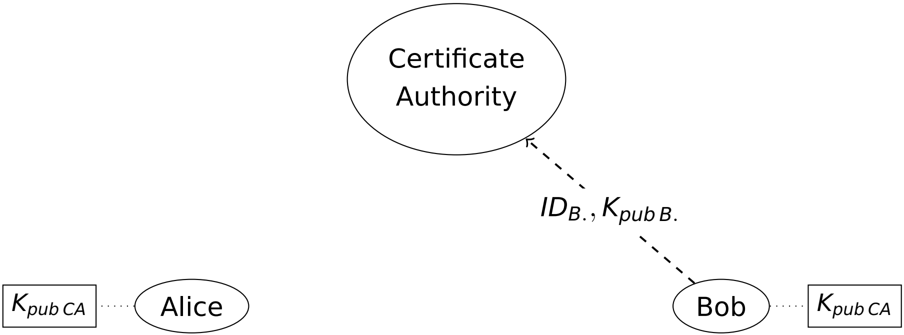
Public-Key Certificates



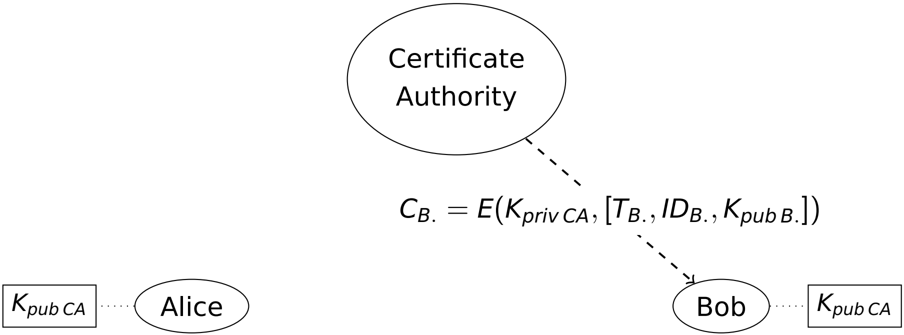
Public-Key Certificates



Public-Key Certificates

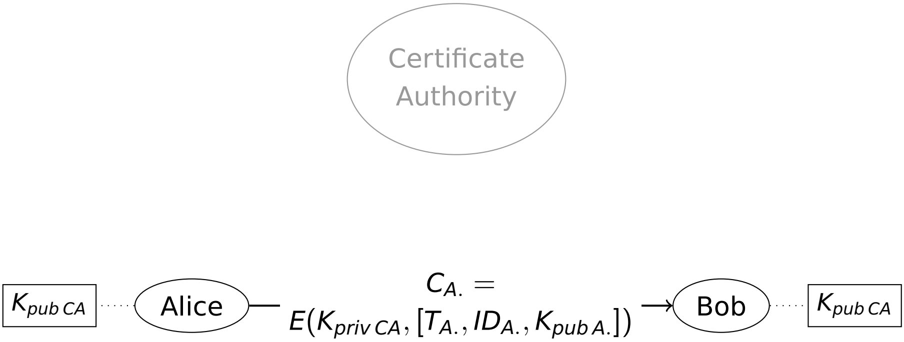


Public-Key Certificates

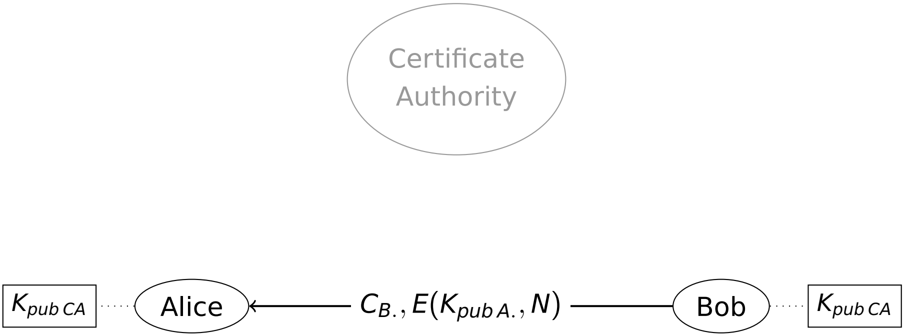


Public-Key Certificates

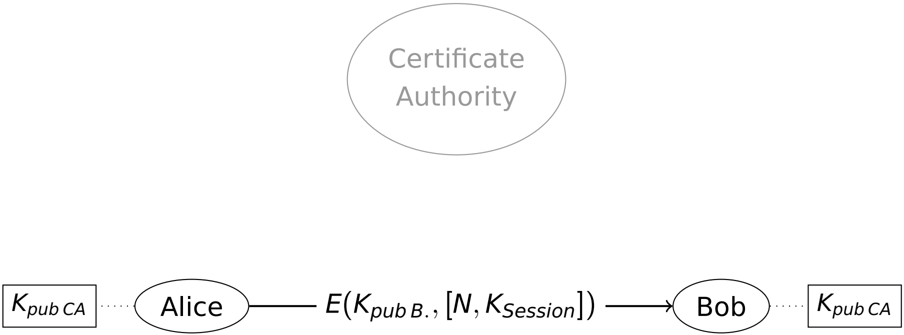
Public-Key Certificates



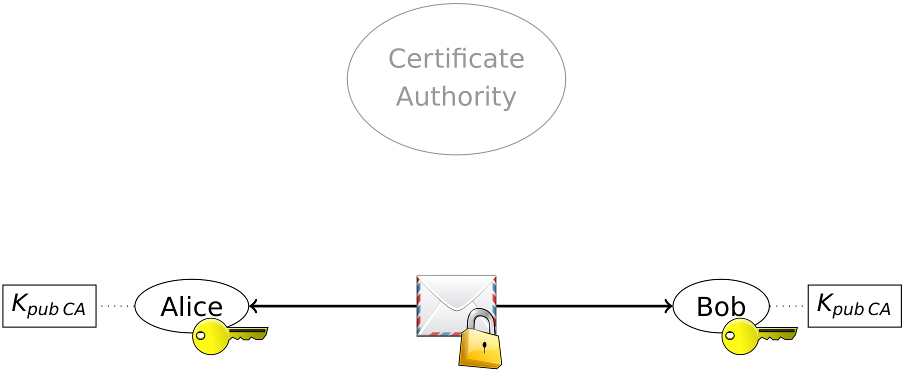
Public-Key Certificates



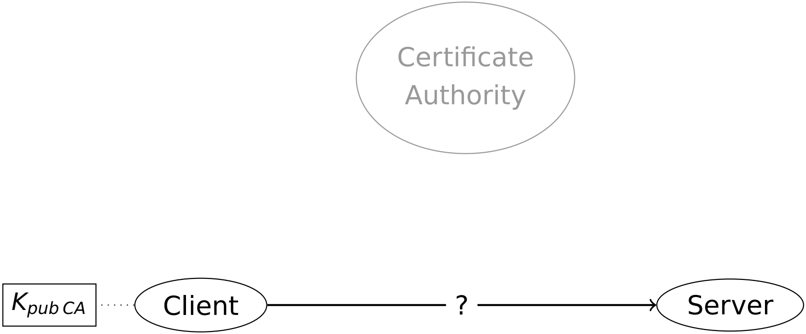
Public-Key Certificates



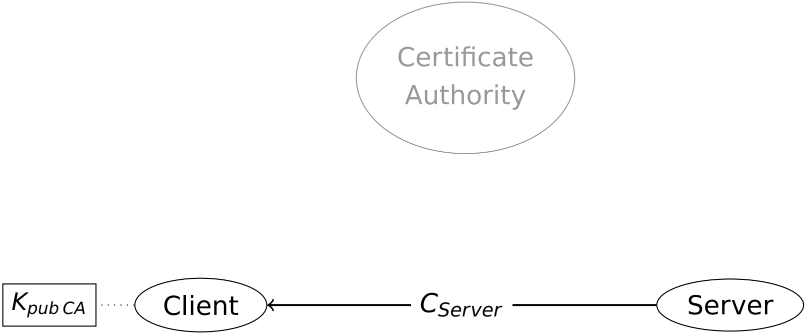
Public-Key Certificates



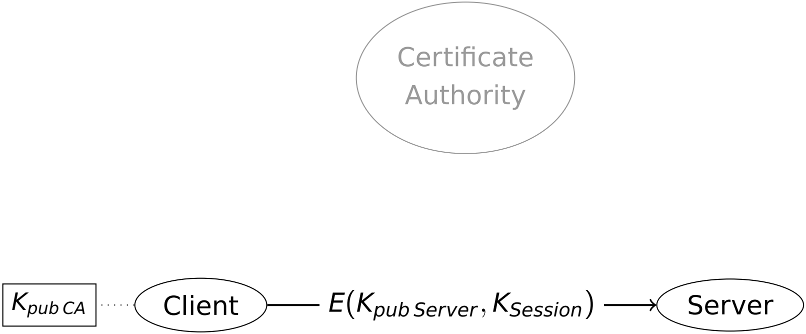
Server Authentication



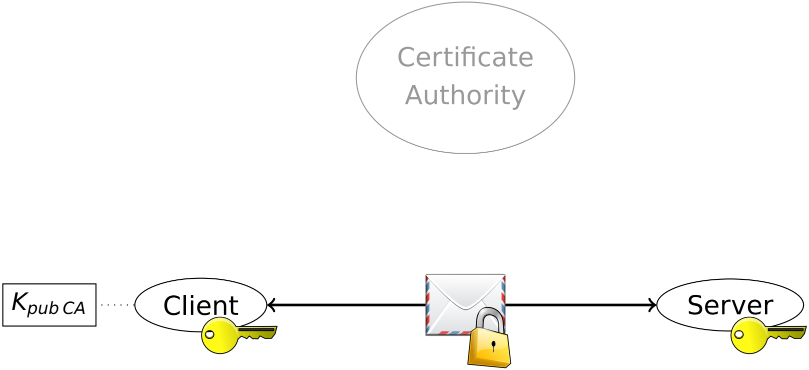
Server Authentication



Server Authentication



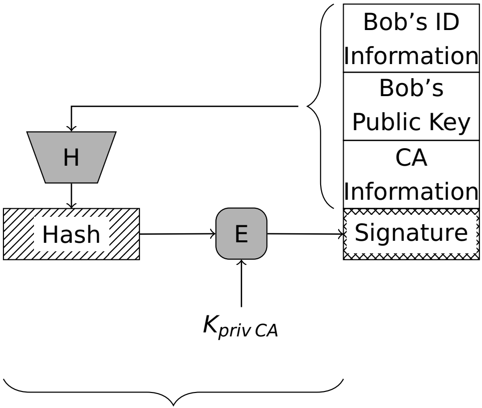
Server Authentication



X.509 Certificates

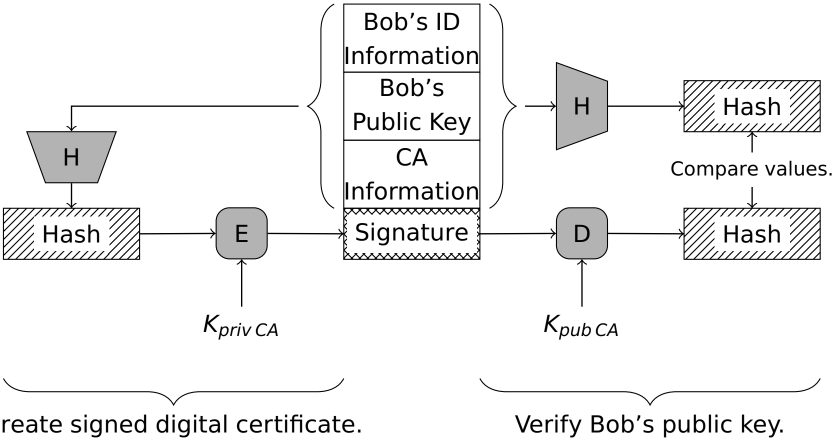
Bob's ID Information
Bob's Public Key
CA Information

X.509 Certificates

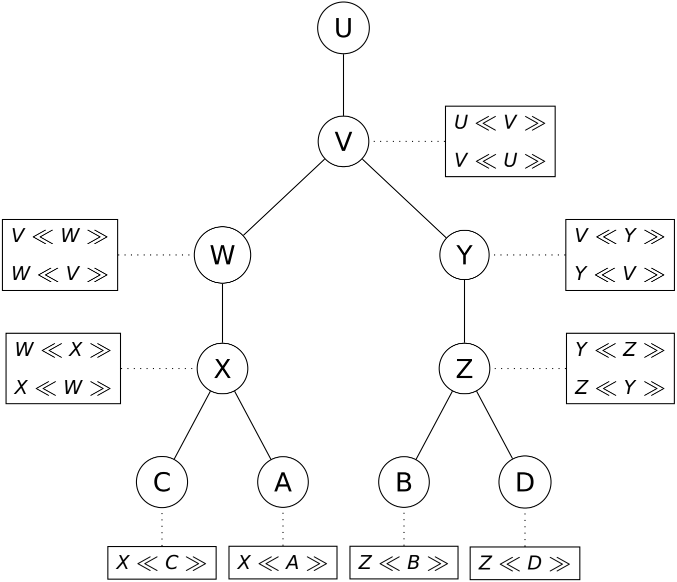


Create signed digital certificate.

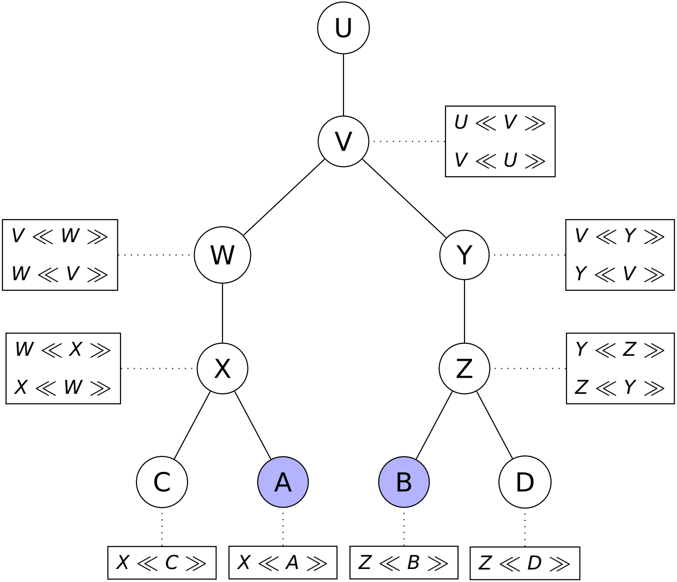
X.509 Certificates



X.509 Hierarchy



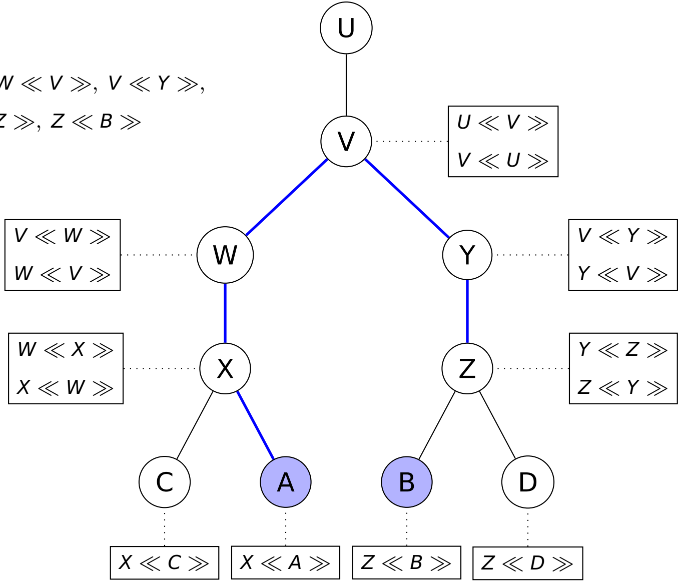
X.509 Hierarchy



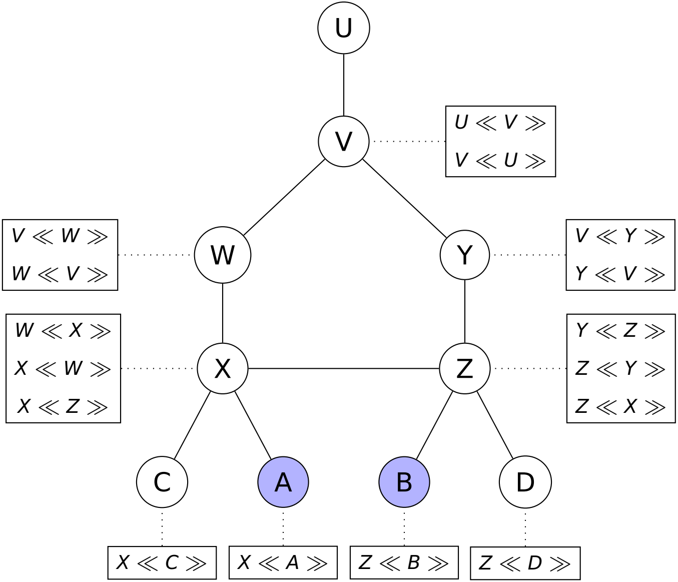
X.509 Hierarchy

$X \ll W \gg$, $W \ll V \gg$, $V \ll Y \gg$,

$Y \ll Z \gg$, $Z \ll B \gg$

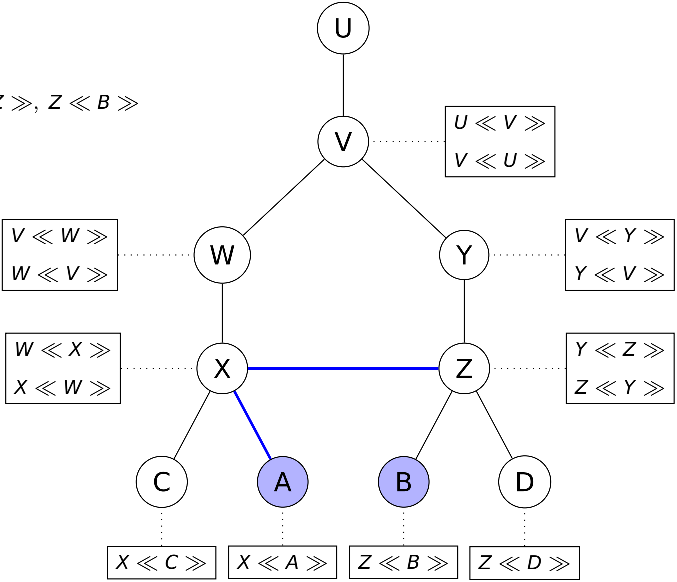


X.509 Hierarchy



X.509 Hierarchy

$X \ll Z \gg, Z \ll B \gg$



X.509 Example

```
Certificate:
   Data:
       Version: 1 (0x0)
       Serial Number: 7829 (0x1e95)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, ...
       Validity
           Not Before: Jul  9 16:04:02 1998 GMT
           Not After : Jul  9 16:04:02 1999 GMT
       Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala, ...
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:...
               Exponent: 65537 (0x10001)
   Signature Algorithm: md5WithRSAEncryption
       93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:...
```

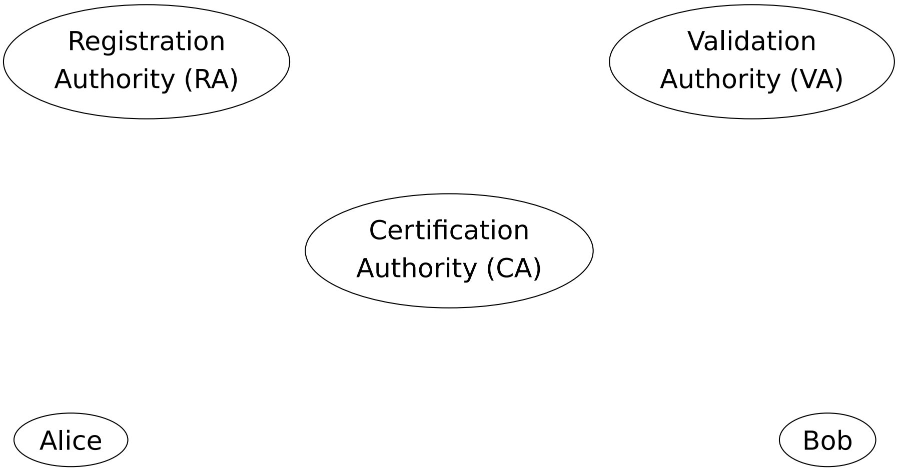
Server Certificate

X.509 Example

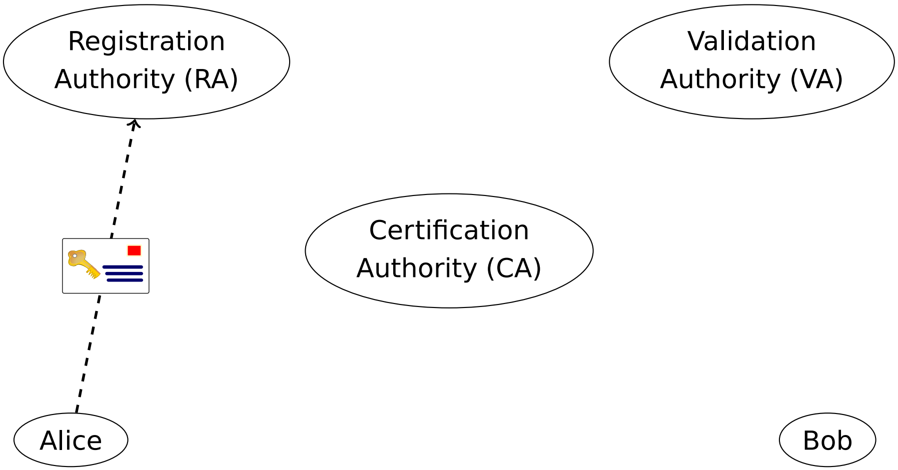
Sub-CA Certificate

```
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number: 1 (0x1)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, ...
       Validity
           Not Before: Jul  9 16:04:02 1998 GMT
           Not After : Jul  9 16:04:02 1999 GMT
       Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala, ...
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:...
               Exponent: 65537 (0x10001)
       X509v3 extensions:
           X509v3 Basic Constraints:  critical
               CA:TRUE
   Signature Algorithm: md5WithRSAEncryption
       93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:...
```

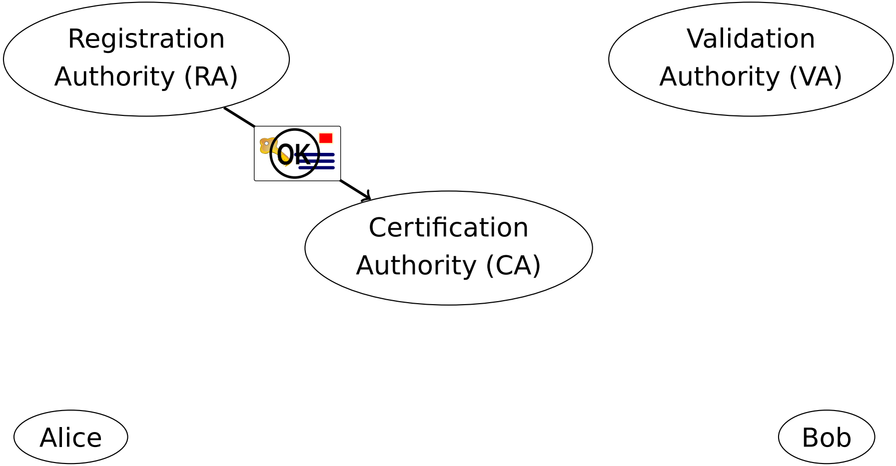
Public Key Infrastructure



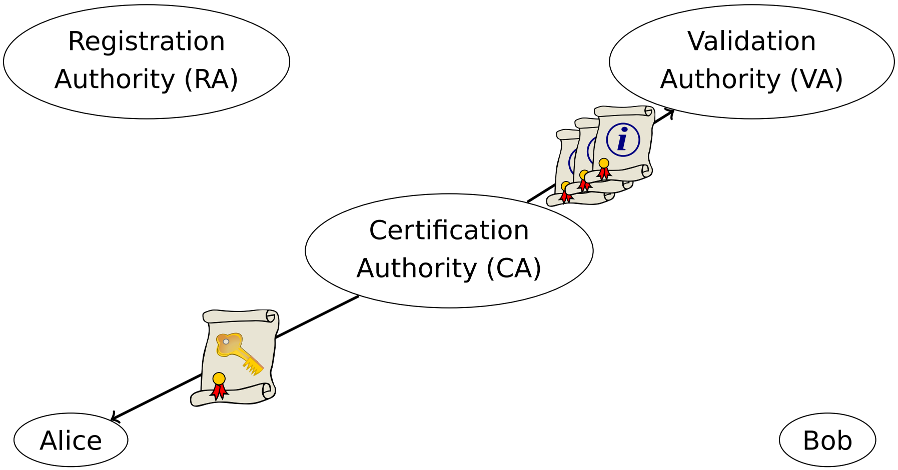
Public Key Infrastructure



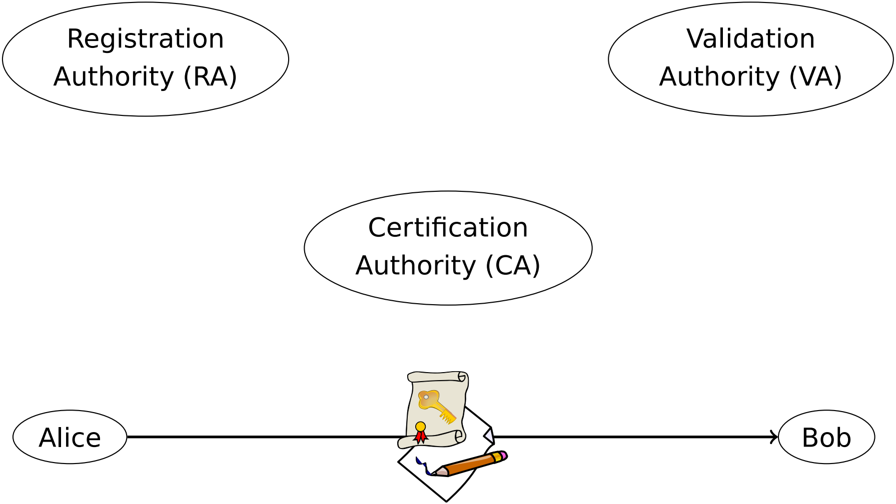
Public Key Infrastructure



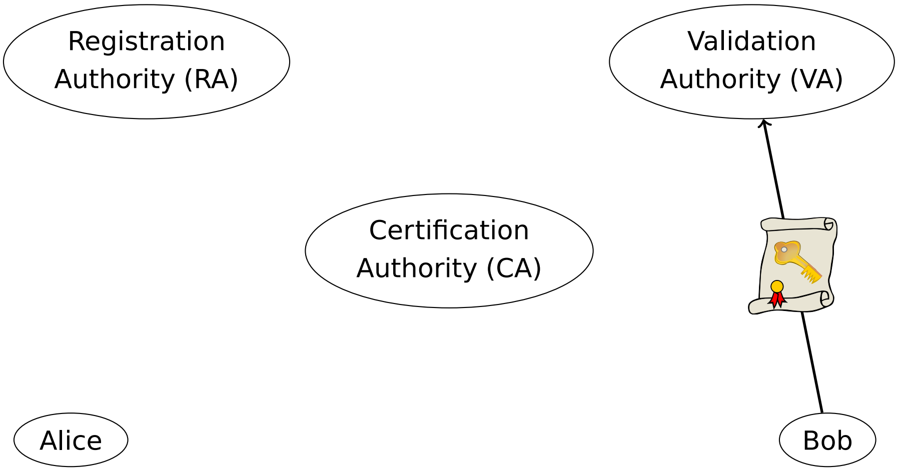
Public Key Infrastructure



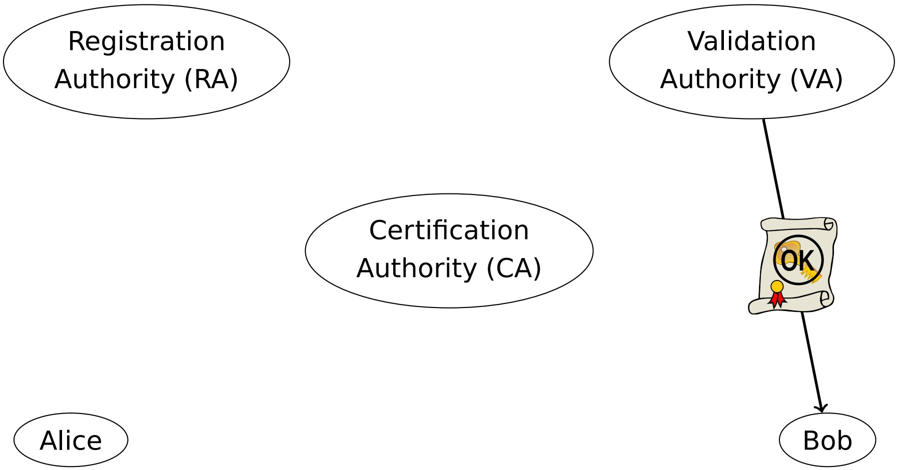
Public Key Infrastructure



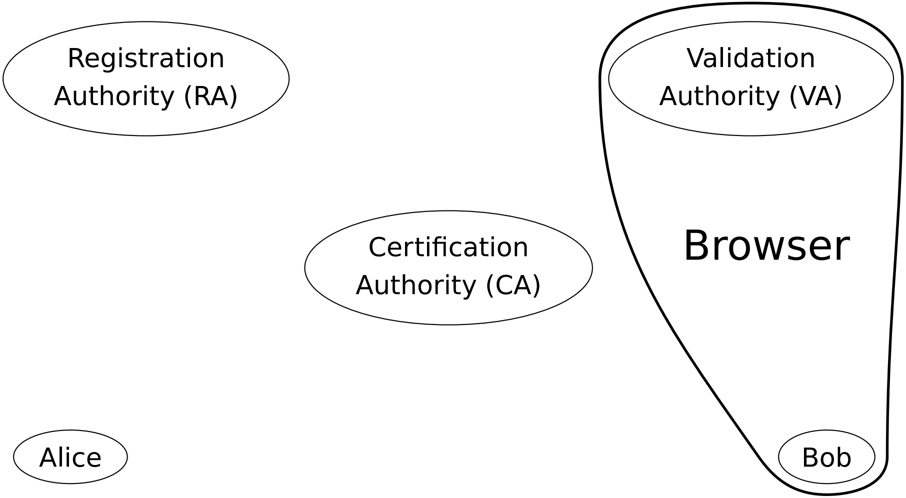
Public Key Infrastructure



Public Key Infrastructure



Public Key Infrastructure



Comodo Security Breach, March 15, 2011

- A user account with an affiliate registration authority had been compromised.
- The attacker issued nine certificate signing requests.
- Certificates for issued for:
  - mail.google.com
  - login.live.com
  - www.google.com
  - login.yahoo.com
  - login.skype.com
  - addons.mozilla.org
- The attack was traced to IP address 212.95.136.18 in Tehran, Iran.
- The origin of the attack may be the “result of an attacker attempting to lay a false trail.”

DigiNotar Fraudulent Certificates, July 10, 2011

- An attacker hacked into the systems of DigiNotar and issued a certificate for Goolge.
- This certificate was subsequently used by unknown persons in Iran to conduct a man-in-the-middle attack against Google services.
- After this certificate was found, DigiNotar belatedly admitted dozens of fraudulent certificates had been created, including certificates for the domains of Yahoo!, Mozilla, WordPress and The Tor Project.
- Google blacklisted 247 certificates in Chromium, but the final known total of misissued certificates is at least 531.
- DigiNotar also controlled an intermediate certificate which was used for issuing certificates as part of the Dutch government's public key infrastructure "PKIoverheid" program.

TURKTRUST and the Chain of Trust, 2012

- TURKTRUST sent two intermediate certificates to organisations that had requested regular certificates.
- One of the certificates was revoked at the request of the customer who received it.
- The other organisation now had the ability to sign SSL certificates for any domain name it chose.
- There is no known malicious use of this intermediate certificate — but TURKTRUST should never have issued it in the first place.

Risks

- Comodo:
- DigiNotar:
- TURKTRUST:

Risks

- Comodo:
  - Authenticity and legitimacy of signing request.
- DigiNotar:
- TURKTRUST:

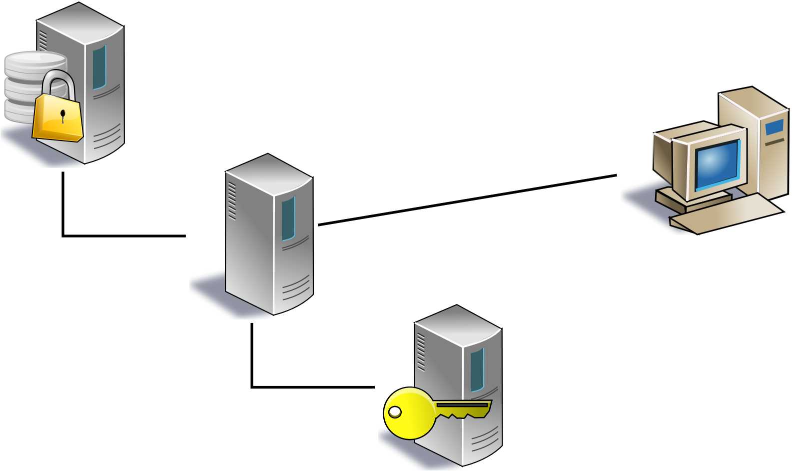
Risks

- Comodo:
  - Authenticity and legitimacy of signing request.
- DigiNotar:
  - Security and integrity of certificate authority.
- TURKTRUST:

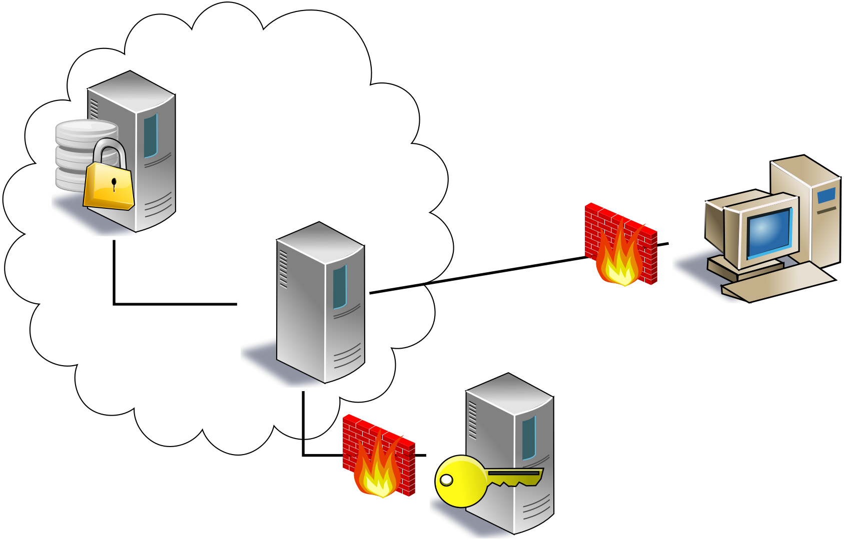
Risks

- Comodo:
  - Authenticity and legitimacy of signing request.
- DigiNotar:
  - Security and integrity of certificate authority.
- TURKTRUST:
  - Just do it right.

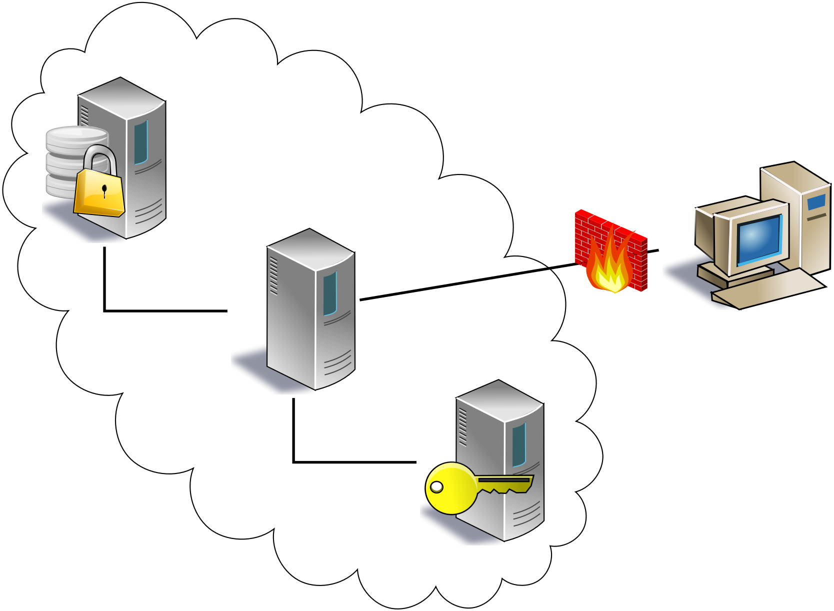
Key Management Strategies in the Cloud



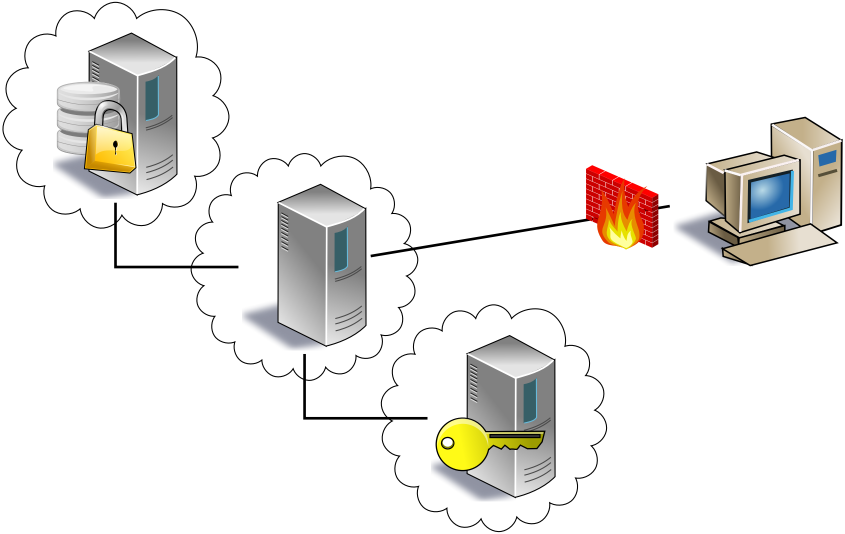
Key Management Strategies in the Cloud



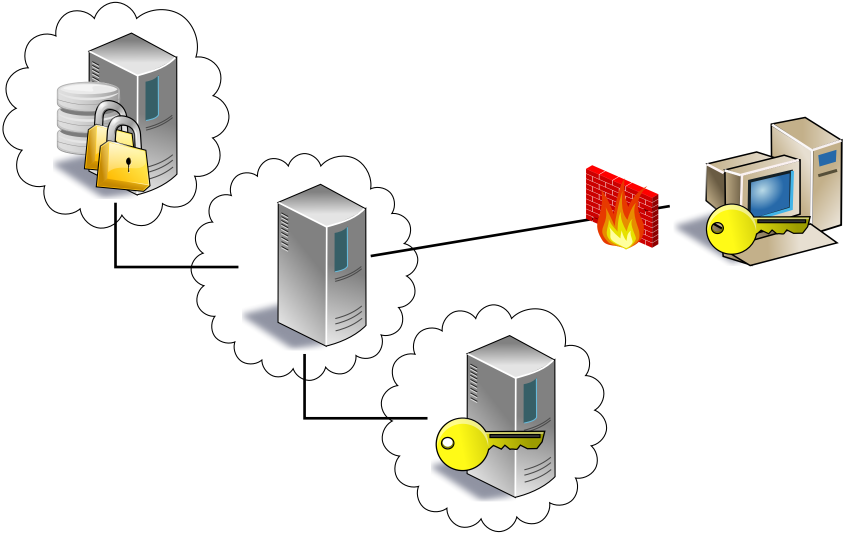
Key Management Strategies in the Cloud



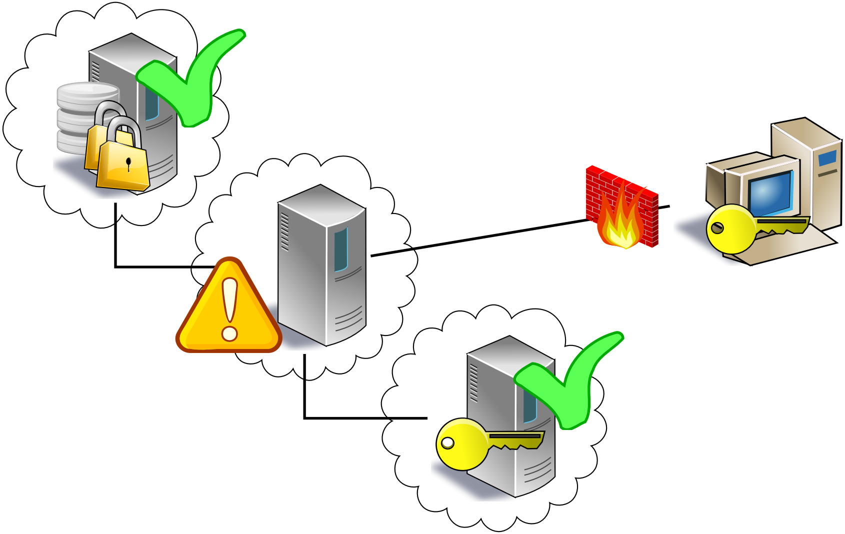
Key Management Strategies in the Cloud



Key Management Strategies in the Cloud



Key Management Strategies in the Cloud



Key Management Strategies in the Cloud

Fully Homomorphic Encryption

- May be used to protect data during computation in the Cloud.
- Far away from being practical; might never be feasible.

Fully Homomorphic Encryption

- May be used to protect data during computation in the Cloud.
- Far away from being practical; might never be feasible.

"Visions of a fully homomorphic cryptosystem have been dancing in cryptographers' heads for thirty years. I never expected to see one. It will be years before a sufficient number of cryptographers examine the algorithm that we can have any confidence that the scheme is secure."

Bruce Schneier

References

Further Reading:

William Stallings, *Cryptography and Network Security: Principles and Practice*, 5th ed. Prentice Hall, Upper Saddle River, NJ, USA. January, 2010.

Clip-Art:

http://openclipart.org/

http://commons.wikimedia.org/