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
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, ...
Symmetric Encryption

Using the same shared key for encryption and decryption:

\[ C = E(K, P) \quad P = E(K, C) \]

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

Symmetric Encryption

Using the same shared key for encryption and decryption:

\[ C = E(K, P) \quad P = E(K, C) \]

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

Asymmetric Encryption

Using public key for encryption and private key for decryption:

\[ C = E(K_{pub}, P) \quad P = E(K_{priv}, C) \]

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!
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 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)

Alice

K_{Alice},
K_{Bob}

KDC

Bob

K_{Bob}

K_{Alice}

Alice

? 

K_{Session}, ID_{Alice}, ID_{Bob}, N_{1}

E(K_{Alice}, [K_{Session}, ID_{Alice}, ID_{Bob}, N_{1}])

E(K_{Bob}, [K_{Session}, ID_{Alice}])

E(K_{Session}, N_{2})

E(K_{Session}, f(N_{2}))
Symmetric Key Distribution
Using Symmetric Encryption

May 10, 2013 9 / 34
Key Distribution Center (KDC)

Symmetric Key Distribution
Using Symmetric Encryption
May 10, 2013 9 / 34
Key Distribution Center (KDC)

Key Distribution Center (KDC)

- $K_{Alice}$
- $K_{Bob}$

- Alice
- $E(K_{Bob}, [K_{Session}, ID_{Alice}])$
- $E(K_{Session}, N_2)$

- Bob
- $K_{Bob}$
Symmetric Key Distribution

Using Symmetric Encryption

May 10, 2013
Key Distribution Center (KDC)

Using Symmetric Encryption
Hierarchical Key Control

Symmetric Key Distribution

Using Symmetric Encryption

May 10, 2013 10 / 34
Hierarchical Key Control

Symmetric Key Distribution Using Symmetric Encryption
Hierarchical Key Control

Symmetric Key Distribution

Using Symmetric Encryption

May 10, 2013 10 / 34
Hierarchical Key Control

\[ K_{KDC,KDC_0}, K_{KDC,KDC_1} \]

\[ K_{KDC_0,A}, K_{KDC,KDC_0} \]

\[ K_{KDC_0,A}, \] Alice

\[ K_{KDC_1,B}, \] Bob

\[ K_{KDC_1,B}. \]
Hierarchical Key Control

Symmetric Key Distribution Using Symmetric Encryption

May 10, 2013 10 / 34
Hierarchical Key Control

$K_{KDC,KDC_0}$, $K_{KDC,KDC_1}$

$E(K_{KDC,KDC_0}, K_{Session})$

$E(K_{KDC_1,B}, K_{Session})$

$K_{KDC_0,A}$, $K_{KDC,KDC_0}$

$K_{KDC_0,A}$

Alice

$K_{KDC_0,A}$, $K_{KDC_1,B}$

Bob

$K_{KDC_1,B}$

Symmetric Key Distribution Using Symmetric Encryption May 10, 2013 10 / 34
Hierarchical Key Control

K_{KDC, KDC_0}, K_{KDC, KDC_1}

E(K_{KDC, KDC_0}, K_{Session})
E(K_{KDC_1, B}, K_{Session})

K_{KDC_0, A}, K_{KDC, KDC_0}

E(K_{KDC_0, A}, K_{Session})
E(K_{KDC_1, B}, K_{Session})

K_{KDC_0, A}

Alice

E(K_{KDC_1, KDC}, K_{Session})
E(K_{KDC_1, B}, K_{Session})

K_{KDC_1, B}, K_{KDC, KDC_1}

K_{KDC_1, B}

Bob
Hierarchical Key Control

\[ K_{KDC_0,A}, K_{KDC,KDC_0}, \]

\[ K_{KDC_0,A}, K_{KDC,KDC_0} \]

\[ K_{KDC,KDC_1} \]

\[ K_{KDC_0,A}, K_{KDC,KDC_0} \]

\[ K_{KDC_0,A}, K_{KDC,KDC_0} \]

\[ K_{KDC_0,A} \]

\[ E(K_{Session}, N) \]

\[ E(K_{Session}, N) \]

\[ E(K_{KDC_1,B}, K_{Session}) \]

\[ E(K_{KDC_1,B}, K_{Session}) \]

\[ E(K_{KDC_1,B}) \]
Hierarchical Key Control

\[ K_{KDC,KDC_0}, K_{KDC,KDC_1} \]

\[ K_{KDC_0,A.}, K_{KDC,KDC_0} \]

\[ K_{KDC_0,A.}, K_{KDC,KDC_0} \]

\[ K_{KDC_0,A.}, K_{KDC,KDC_0} \]

Symmetric Key Distribution

Using Symmetric Encryption

May 10, 2013
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

Host A

Application

Security Service

Host B

Application

Security Service

KDC

Network
Transparent Key Control Scheme

Host A

Application

Security Service

Host B

Application

Security Service

KDC

Network
Transparent Key Control Scheme

Network

KDC

Host A

Application

Security Service

Host B

Application

Security Service

Symmetric Key Distribution Using Symmetric Encryption
Transparent Key Control Scheme

- KDC
- Network
- Application
- Security Service
- Host A
- Host B

Symmetric Key Distribution Using Symmetric Encryption
Decentralized Key Control

Alice ✢ ✫ Bob

E(K_{master}, [K_{Session}, ID_Alice, ID_Bob, N_1])

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

\[ K_{\text{master}} \quad \rightarrow \quad ? \quad \rightarrow \quad K_{\text{master}} \]

\[ E(\text{ID}_\text{Alice}, N_1, \text{ID}_\text{Bob}, N_1) \]

Requires \( \frac{n(n-1)}{2} \) master keys for \( n \) nodes.
Decentralized Key Control

Symmetric Key Distribution
Using Symmetric Encryption
May 10, 2013
Decentralized Key Control

\[ E(K_{\text{master}}, [K_{\text{Session}}, ID_{\text{Alice}}, ID_{\text{Bob}}, N_1]) \]

Requires \( n \left( n - 1 \right) / 2 \) master keys for \( n \) nodes.
Decentralized Key Control

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

Alice

Bob

Using Asymmetric Encryption
Simple Secret Key Distribution

Alice \rightarrow K_{pub \ Alice, ID_{Alice}} \rightarrow Bob
Alice

\[ E(K_{pub\ Alice}, K_{Session}) \]

Bob
Simple Secret Key Distribution

Alice

Bob

Symmetric Key Distribution Using Asymmetric Encryption

May 10, 2013
Simple Secret Key Distribution

Symmetric Key Distribution Using Asymmetric Encryption

[Diagram showing a network with Mallory, Alice, and Bob connected by arrows]
Simple Secret Key Distribution

Symmetric Key Distribution

Using Asymmetric Encryption

May 10, 2013 15 / 34
Simple Secret Key Distribution

Mallory

\[ K_{pub\ Mallory,\ ID_{Alice}} \]

Alice

Bob

Using Asymmetric Encryption
Simple Secret Key Distribution

Alice

Mallory

Bob

\[ E(K_{\text{pub Mallory}}, K_{\text{Session}}) \]
Simple Secret Key Distribution

Symmetric Key Distribution Using Asymmetric Encryption
Simple Secret Key Distribution

Symmetric Key Distribution

Using Asymmetric Encryption

May 10, 2013
Secret Key Distribution with Confidentiality and Authentication

Alice

? → Bob

Using Asymmetric Encryption
Symmetric Key Distribution with Confidentiality and Authentication

Alice

Bob

\(K_{pub\ B.}\) → Alice → ? → Bob → \(K_{pub\ A.}\)

Use signatures!
Secret Key Distribution with Confidentiality and Authentication

\[ K_{pub\ B.} \xrightarrow{} Alice \xrightarrow{E(K_{pub\ B.}, [ID_{A.}, N_1])} Bob \xrightarrow{} K_{pub\ A.} \]
Secret Key Distribution with Confidentiality and Authentication

$$K_{pub \ B.} \rightarrow Alice \leftarrow E(K_{pub \ A.}, [N_1, N_2]) \rightarrow Bob \rightarrow K_{pub \ A.}$$
Secret Key Distribution with Confidentiality and Authentication

\[ E(K_{pub\ B}, [N_2, K_{Session}]) \]

Using Asymmetric Encryption

Symmetric Key Distribution

May 10, 2013
Secret Key Distribution with Confidentiality and Authentication

Alice

Bob

Symmetric Key Distribution

Using Asymmetric Encryption

May 10, 2013 16 / 34
Secret Key Distribution with Confidentiality and Authentication

Use signatures!
Secret Key Distribution with Confidentiality and Authentication

How do we distribute the public keys?

Symmetric Key Distribution

Using Asymmetric Encryption
Secret Key Distribution with Confidentiality and Authentication

How do we distribute the public keys?

Symmetric Key Distribution
Using Asymmetric Encryption

May 10, 2013
Secret Key Distribution with Confidentiality and Authentication

How do we distribute the public keys?

Symmetric Key Distribution
Using Asymmetric Encryption
Secret Key Distribution with Confidentiality and Authentication

How do we distribute the public keys?

Symmetric Key Distribution
Using Asymmetric Encryption
May 10, 2013 17 / 34
Public Key Distribution

General Schemes:

- public announcement
- publicly available directory
- public-key authority
- public-key certificates
Public Key Distribution

Alice

\[ ID_{A}, K_{pub A}. \]

Bob

\[ ID_{A}, K_{pub A}. \]

\[ ID_{A}, K_{pub A}. \]

\[ ID_{A}, K_{pub A}. \]

\[ ID_{A}, K_{pub A}. \]

Webpage

Newsgroups

Mail Signature

Name Card

...
Public Key Distribution

Alice

- ID_A, K_{pubA}
- ID_A, K_{pubA}
- ID_A, K_{pubA}
- ID_A, K_{pubA}
- ID_A, K_{pubA}

Bob

- ID_A, K_{pubA}

Webpage
Newsgroups
Mail Signature
Name Card
...

Easy to forge!
Public Key Distribution

Mallory

- ID_A., K_{pub M.}
- ID_A., K_{pub M.}
- ID_A., K_{pub M.}
- ID_A., K_{pub M.}
- ID_A., K_{pub M.}

Webpage
Newsgroups
Mail Signature
Name Card
...

Bob

Easy to forge!
Public Announcements

Mallory

- $ID_A, K_{pub M}$
- $ID_A, K_{pub M}$
- $ID_A, K_{pub M}$
- $ID_A, K_{pub M}$
- $ID_A, K_{pub M}$

Webpage

Newsgroups

Mail Signature

Name Card

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

\[ ID_A, K_{pub A}, ID_B, K_{pub B} \]

Long latencies due to communication with PKA!
- Alice and Bob may cache public keys.
- Use certificates...
Alice

K_{pub \, PKA}

Public-Key Authority

Bob

K_{pub \, PKA}

Long latencies due to communication with PKA!
▶ Alice and Bob may cache public keys.
▶ Use certificates...

Public Key Distribution
May 10, 2013 21 / 34
Public-Key Authority

\[ E(K_{\text{priv PKA}}, [K_{\text{pub B}}, T_1]) \]

\[ E(K_{\text{pub B}}, [\text{ID}_A, N_1]) \]

\[ E(K_{\text{priv PKA}}, [K_{\text{pub A}}, T_2]) \]

\[ E(K_{\text{pub A}}, [N_1, N_2]) \]

\[ E(K_{\text{pub B}}, [N_2, \text{Session}]) \]

Long latencies due to communication with PKA!

▶ Alice and Bob may cache public keys.

▶ Use certificates...
Public-Key Authority

\[ E(K_{\text{priv PKA}}, [K_{\text{pub B.}}, T_1]) \]

Long latencies due to communication with PKA!

- Alice and Bob may cache public keys.
- Use certificates...
Public-Key Authority

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

- Alice
  - $E(K_{pub\ B.}, [ID_{A.}, N_1])$
- Bob
  - $E(K_{pub\ A.}, [N_1, N_2])$
  - $E(K_{pub\ B.}, [N_2, Session])$

Long latencies due to communication with PKA!

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

May 10, 2013 21 / 34
Public-Key Authority

Alice

Bob

Public-Key Authority

$K_{pub \text{ PKA}}$

$K_{priv \text{ PKA}}$

$ID_{Alice}, T_2$?
Public-Key Authority

Public-Key Authority

$E(K_{priv \text{PKA}}, [K_{pub A}, T_2])$

$K_{pub \text{PKA}}$ ···· Alice

Bob ···· $K_{pub \text{PKA}}$

Long latencies due to communication with PKA!

- Alice and Bob may cache public keys.
- Use certificates...
Public-Key Authority

\[ E(K_{pub A.}, [N_1, N_2]) \]

Long latencies due to communication with PKA!

Alice and Bob may cache public keys.

Use certificates.
Public-Key Authority

\[ E(K_{\text{pub B.}}, [N_2, K_{\text{Session}}]) \]

Long latencies due to communication with PKA!

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

---

Public Key Distribution May 10, 2013 21 / 34
Public-Key Authority

Public-Key Authority

\[ K_{pub\ PKA} \quad \rightarrow \quad Alice \quad \leftarrow \quad \] 

\[ Bob \quad \rightarrow \quad K_{pub\ PKA} \]

Long latencies due to communication with PKA!

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

Public Key Distribution

May 10, 2013 21 / 34
Long latencies due to communication with PKA!

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

$K_{pub\ CA}$  Alice

Bob  $K_{pub\ CA}$

$C_{A} = E(K_{priv\ CA}, [T_{A}, ID_{A}, K_{pub\ A}])$

$C_{B} = E(K_{priv\ CA}, [T_{B}, ID_{B}, K_{pub\ B}])$

$E(K_{pub\ A}, N) = E(K_{pub\ B}, [N, K_{Session}])$
Public-Key Certificates

Certificate Authority

$ID_A, K_{pubA}$

$K_{pubCA}$ — Alice

$K_{pubCA}$ — Bob

$C_{pubCA}$

$C_A = E(K_{privCA}, [T_A, ID_A, K_{pubA}])$

$C_B = E(K_{privCA}, [T_B, ID_B, K_{pubB}])$

$E(K_{pubA}, N) = E(K_{pubB}, [N, K_{Session}])$
Public-Key Certificates

Certificate Authority

\[ C_A. = E(K_{priv\,CA}, [T_A., ID_A., K_{pub\,A.}]) \]

Alice

Bob

Public Key Distribution
Certificate Authority

\[ ID_{B, A} = E(\text{priv CA}, [T_A, ID_A, K_{pub A}]) \]

\[ ID_{B, B} = E(\text{priv CA}, [T_B, ID_B, K_{pub B}]) \]

\[ C_A = E(\text{priv CA}, [T_A, ID_A, K_{pub A}]) \]

\[ C_B = E(\text{pub CA}, N) \]

\[ E(\text{pub CA}, N) = E(\text{pub B}, K_{Session}) \]
Public-Key Certificates

$C_B. = E(K_{priv\,CA}, [T_B., ID_B., K_{pub\,B.}])$

Certificate Authority

$K_{pub\,CA}$ \hspace{2cm} Alice

Bob \hspace{2cm} $K_{pub\,CA}$
Public-Key Certificates

Certificate Authority

$K_{\text{pub} \, CA}$ → Alice → Bob → $K_{\text{pub} \, CA}$

$C_{\text{CA}} = E(K_{\text{priv} \, CA}, [T_{A}, ID_{A}, K_{\text{pub} \, A}])$

$C_{\text{B}} = E(K_{\text{priv} \, CA}, [T_{B}, ID_{B}, K_{\text{pub} \, B}])$

$E(K_{\text{pub} \, A}, N) = E(K_{\text{pub} \, B}, [N, K_{\text{Session}}])$
Public-Key Certificates

Certificate Authority

$K_{pub\ CA}$  $\rightarrow$  $Alice$  $\rightarrow$  $Bob$  $\rightarrow$  $K_{pub\ CA}$

$C_A = E(K_{priv\ CA}, [T_A, ID_A, K_{pub\ A}])$
Public-Key Certificates

Certificate Authority

\[ C_A = E(K_{priv \, CA}, [T_A, ID_A, K_{pub \, A}]) \]

\[ C_B = E(K_{priv \, CA}, [T_B, ID_B, K_{pub \, B}]) \]

\[ C_B, E(K_{pub \, A}, N) \]

\[ K_{pub \, CA} \rightarrow \text{Certificate Authority} \]

\[ \text{Alice} \rightarrow C_B, E(K_{pub \, A}, N) \rightarrow \text{Bob} \rightarrow K_{pub \, CA} \]
Public-Key Certificates

Certificate Authority

\[ K_{pub\,CA} \quad \Rightarrow \quad Alice \quad \xrightarrow{E(K_{pub\,B,} [N, K_{Session}])} \quad Bob \quad \Rightarrow \quad K_{pub\,CA} \]

E(\text{K}_{pub\,B,}, [N, K_{Session}])
Public-Key Certificates

Certificate Authority

$K_{pub CA}$  $\rightarrow$ Alice  $\leftarrow$  $K_{pub CA}$

$K_{priv CA}$  $\rightarrow$  $E(K_{priv CA}, [T_A, ID_A, K_{pub A}])$

$C_{A} = E(K_{priv CA}, [T_A, ID_A, K_{pub A}])$

$C_{B} = E(K_{priv CA}, [T_B, ID_B, K_{pub B}])$

$E(K_{pub A}, N) = E(K_{pub B}, [N, K_{Session}])$

$ID_A, K_{pub A}$

$ID_B, K_{pub B}$
Server Authentication

Certificate Authority

\[ K_{pub\, CA} \rightarrow \text{Client} \rightarrow \text{Server} \]
Server Authentication

Certificate Authority

$K_{pub \, CA}$

Client

$C_{Server}$

Server
Server Authentication

Certificate Authority

\[ K_{pub \, CA} \rightarrow \text{Client} \rightarrow E(K_{pub \, Server}, K_{Session}) \rightarrow \text{Server} \]
Server Authentication

Certificate Authority

$K_{pub CA}$ Client

Server

Public Key Distribution

May 10, 2013 23 / 34
X.509 Certificates

<table>
<thead>
<tr>
<th>Bob’s ID Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob’s Public Key</td>
</tr>
<tr>
<td>CA Information</td>
</tr>
</tbody>
</table>
Create signed digital certificate.
X.509 Certificates

CA Information

Bob’s ID

Bob’s Public Key

CA Information

Signature

Create signed digital certificate.

Hash

E

K_{priv CA}

Verify Bob’s public key.

H

D

K_{pub CA}

Hash

Compare values.
X.509 Hierarchy

```
V ≪ W ≪ X ≪ W ≪ Y ≪ Z ≪ Y ≪ V ≪ U ≪ V ≪ W
```

```
X ≪ C ≪ X ≪ A ≪ Z ≪ B ≪ Z ≪ D
```

Public Key Distribution
X.509 Hierarchy

Public Key Distribution

X.509 Certificates

May 10, 2013 25 / 34
X ≪ W ≫, W ≪ V ≫, V ≪ Y ≫,
Y ≪ Z ≫, Z ≪ B ≫
X.509 Hierarchy

Public Key Distribution

X.509 Certificates

May 10, 2013
X.509 Hierarchy

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

- $V \ll U \gg$
- $U \ll V \gg$
- $V \ll U \gg$
- $V \ll W \gg$
- $W \ll V \gg$
- $W \ll X \gg$
- $X \ll W \gg$
- $X \ll C \gg$
- $X \ll A \gg$
- $Z \ll B \gg$
- $Z \ll D \gg$

Public Key Distribution

May 10, 2013
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):
        Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
X.509 Example

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, 0=Brent Baccala, ...
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
      Modulus (1024 bit):
      Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Basic Constraints: critical
    CA:TRUE
  Signature Algorithm: md5WithRSAEncryption
Public Key Infrastructure

Registration Authority (RA)

Certification Authority (CA)

Validation Authority (VA)

Alice

Bob
Public Key Infrastructure

- Registration Authority (RA)
- Certification Authority (CA)
- Validation Authority (VA)

Alice and Bob are part of the Public Key Infrastructure.
Public Key Infrastructure

Registration Authority (RA)

Certification Authority (CA)

Validation Authority (VA)

Alice → Bob
Public Key Infrastructure

Registration Authority (RA)

Certification Authority (CA)

Validation Authority (VA)

Alice

Bob
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.”
An attacker hacked into the systems of DigiNotar and issued a certificate for Google.

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
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
Further Reading:

Clip-Art:
http://openclipart.org/
http://commons.wikimedia.org/