Nyckelhantering och autentisering

Daniel Bosk

Avdelningen för informations- och kommunikationssytem (IKS),
Mittuniversitetet, Sundsvall.

keyauth.tex 1352 2013-10-01 09:25:08Z danbos

Översikt

1. Symmetric Key Distribution
   - Symmetric Crypto
   - Key Distribution Centre (KDC)
   - Authentication
   - Kerberos IV
   - Kerberos V
2. Asymmetric Key Distribution
   - Asymmetric Crypto and Hash Functions
   - Diffie–Hellman Key Exchange
   - Public-key Certificates
3. Federated Identity Management
   - Identity Management
   - Identity Federation

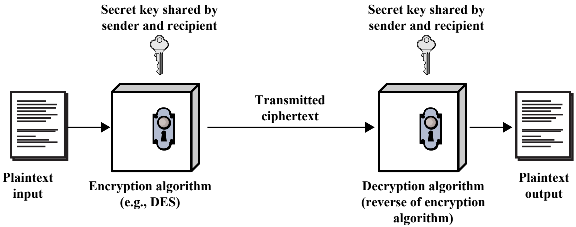
Litteratur

The lecture covers chapter 4 “Key Distribution and User Authentication” in [Sta13] and chapter 3 “Protocols” in [And08]. When you are done studying the material you should solve problems 4.1, 4.2, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, and 4.11 in [Sta13].

Översikt

1. Symmetric Key Distribution
   - Symmetric Crypto
   - Key Distribution Centre (KDC)
   - Authentication
   - Kerberos IV
   - Kerberos V
2. Asymmetric Key Distribution
   - Asymmetric Crypto and Hash Functions
   - Diffie–Hellman Key Exchange
   - Public-key Certificates
3. Federated Identity Management
   - Identity Management
   - Identity Federation

Symmetric Crypto


Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figur : An overview of symmetric crypto. Image: [Sta13].

Key Distribution Centre (KDC)

- Deliver a key k from A to B. By themselves or third party.
- If A and B share a key k, generate a key k' and transmit it using k: $A \rightarrow B: E_k(k')$.
- Secure connection to third party C, C delivers key to A and B.

Mittuniversitetet
MID SWEDEN UNIVERSITY

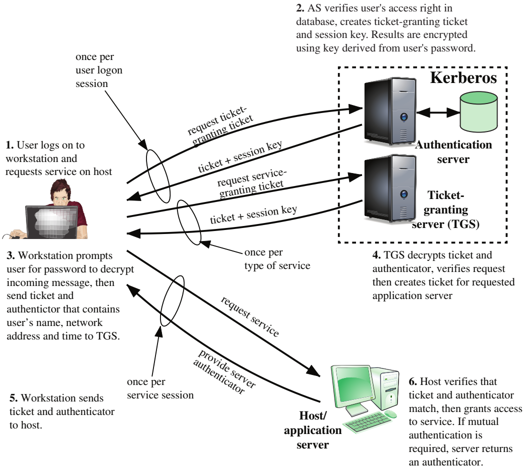
Key Distribution Centre (KDC)

Session Key Temporary key used between A and B.

Permanent Key Key used to distribute session keys.

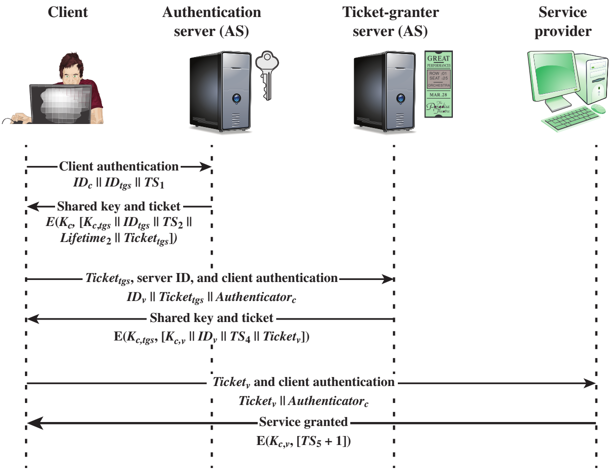
Key Distribution Centre The central entity with which permanent keys are shared and by whom session keys are generated.

Authentication


2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.
once per user logon session
Kerberos
Authentication server
Ticket-granting server (TGS)
request ticket-granting ticket
ticket + session key
request service-granting ticket
ticket + session key
1. User logs on to workstation and requests service on host
once per type of service
3. Workstation prompts user for password to decrypt incoming message, then send ticket and authentictor that contains user’s name, network address and time to TGS.
4. TGS decrypts ticket and authenticator, verifies request then creates ticket for requested application server
request service
provide server authenticator
once per service session
5. Workstation sends ticket and authenticator to host.
Host/ application server
6. Host verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

Figur : An overview of Kerberos. Image: [Sta13].

Kerberos IV



Client

Authentication server (AS)

Ticket-granter server (AS)

Service provider

Client authentication

$ID_c \parallel ID_{tgs} \parallel TS_1$

Shared key and ticket

$E(K_c, [K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}])$

$Ticket_{tgs}$, server ID, and client authentication

$ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$

Shared key and ticket

$\mathrm{E}(K_{c,tgs}, [K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v])$

$Ticket_v$ and client authentication

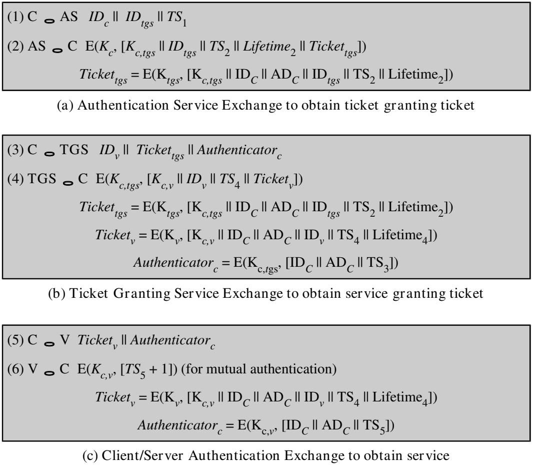
$Ticket_v \parallel Authenticator_c$

Service granted

$\mathrm{E}(K_{c,v}, [TS_5 + 1])$

Figur : An overview of Kerberos IV authentication dialogue. Image: [Sta13].

Kerberos IV



(1) C $\rightarrow$ AS $ID_c \| ID_{tgs} \| TS_1$

(2) AS $\rightarrow$ C $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$$Ticket_{tgs} = \mathrm{E}(\mathrm{K}_{tgs}, [\mathrm{K}_{c,tgs} \| \mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{ID}_{tgs} \| \mathrm{TS}_2 \| \mathrm{Lifetime}_2])$$

(a) Authentication Service Exchange to obtain ticket granting ticket

(3) C $\rightarrow$ TGS $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) TGS $\rightarrow$ C $E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$$Ticket_{tgs} = \mathrm{E}(\mathrm{K}_{tgs}, [\mathrm{K}_{c,tgs} \| \mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{ID}_{tgs} \| \mathrm{TS}_2 \| \mathrm{Lifetime}_2])$$

$$Ticket_v = \mathrm{E}(\mathrm{K}_v, [\mathrm{K}_{c,v} \| \mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{ID}_v \| \mathrm{TS}_4 \| \mathrm{Lifetime}_4])$$

$$Authenticator_c = \mathrm{E}(\mathrm{K}_{c,tgs}, [\mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{TS}_3])$$

(b) Ticket Granting Service Exchange to obtain service granting ticket

(5) C $\rightarrow$ V $Ticket_v \| Authenticator_c$

(6) V $\rightarrow$ C $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

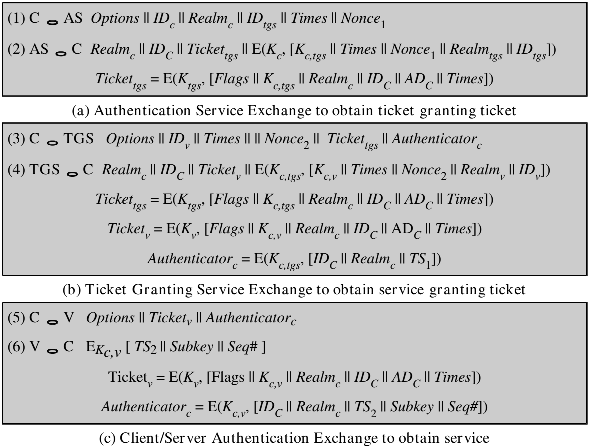
$$Ticket_v = \mathrm{E}(\mathrm{K}_v, [\mathrm{K}_{c,v} \| \mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{ID}_v \| \mathrm{TS}_4 \| \mathrm{Lifetime}_4])$$

$$Authenticator_c = \mathrm{E}(\mathrm{K}_{c,v}, [\mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{TS}_5])$$

(c) Client/Server Authentication Exchange to obtain service

Figur : Kerberos IV authentication protocol. Image: [Sta13].

Kerberos V



(1) C → AS $\mathit{Options} \parallel \mathit{ID}_c \parallel \mathit{Realm}_c \parallel \mathit{ID}_{tgs} \parallel \mathit{Times} \parallel \mathit{Nonce}_1$

(2) AS → C $\mathit{Realm}_c \parallel \mathit{ID}_C \parallel \mathit{Ticket}_{tgs} \parallel \mathrm{E}(K_c, [K_{c,tgs} \parallel \mathit{Times} \parallel \mathit{Nonce}_1 \parallel \mathit{Realm}_{tgs} \parallel \mathit{ID}_{tgs}])$

$$\mathit{Ticket}_{tgs} = \mathrm{E}(K_{tgs}, [\mathit{Flags} \parallel K_{c,tgs} \parallel \mathit{Realm}_c \parallel \mathit{ID}_C \parallel \mathit{AD}_C \parallel \mathit{Times}])$$

(a) Authentication Service Exchange to obtain ticket granting ticket

(3) C → TGS $\mathit{Options} \parallel \mathit{ID}_v \parallel \mathit{Times} \parallel \parallel \mathit{Nonce}_2 \parallel \mathit{Ticket}_{tgs} \parallel \mathit{Authenticator}_c$

(4) TGS → C $\mathit{Realm}_c \parallel \mathit{ID}_C \parallel \mathit{Ticket}_v \parallel \mathrm{E}(K_{c,tgs}, [K_{c,v} \parallel \mathit{Times} \parallel \mathit{Nonce}_2 \parallel \mathit{Realm}_v \parallel \mathit{ID}_v])$

$$\mathit{Ticket}_{tgs} = \mathrm{E}(K_{tgs}, [\mathit{Flags} \parallel K_{c,tgs} \parallel \mathit{Realm}_c \parallel \mathit{ID}_C \parallel \mathit{AD}_C \parallel \mathit{Times}])$$

$$\mathit{Ticket}_v = \mathrm{E}(K_v, [\mathit{Flags} \parallel K_{c,v} \parallel \mathit{Realm}_c \parallel \mathit{ID}_C \parallel \mathrm{AD}_C \parallel \mathit{Times}])$$

$$\mathit{Authenticator}_c = \mathrm{E}(K_{c,tgs}, [\mathit{ID}_C \parallel \mathit{Realm}_c \parallel \mathit{TS}_1])$$

(b) Ticket Granting Service Exchange to obtain service granting ticket

(5) C → V $\mathit{Options} \parallel \mathit{Ticket}_v \parallel \mathit{Authenticator}_c$

(6) V → C $\mathrm{E}_{K_{c,v}}\,[\,\mathit{TS}_2 \parallel \mathit{Subkey} \parallel \mathit{Seq\#}\,]$

$$\mathrm{Ticket}_v = \mathrm{E}(K_v, [\mathrm{Flags} \parallel K_{c,v} \parallel \mathit{Realm}_c \parallel \mathit{ID}_C \parallel \mathit{AD}_C \parallel \mathit{Times}])$$

$$\mathit{Authenticator}_c = \mathrm{E}(K_{c,v}, [\mathit{ID}_C \parallel \mathit{Realm}_c \parallel \mathit{TS}_2 \parallel \mathit{Subkey} \parallel \mathit{Seq\#}])$$

(c) Client/Server Authentication Exchange to obtain service

Figur : Kerberos V authentication protocol. Image: [Sta13].

Kerberos V

Environmental Differences

- Encryption system dependence.
- Internet protocol dependence.
- Byte ordering.
- Ticket lifetime.
- Authentication forwarding.
- Interrealm authentication.

Kerberos V

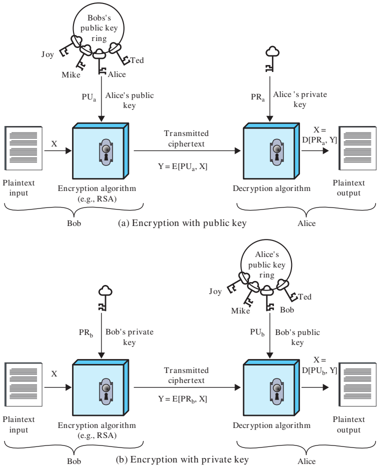
Technical differences

- Double encryption.
- Propagating Cipher Block Chaining instead of CBC.
- Session and subsession keys.
- Password attacks.

Översikt

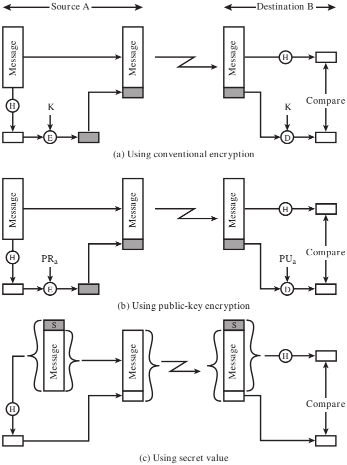
1. Symmetric Key Distribution
   - Symmetric Crypto
   - Key Distribution Centre (KDC)
   - Authentication
   - Kerberos IV
   - Kerberos V
2. Asymmetric Key Distribution
   - Asymmetric Crypto and Hash Functions
   - Diffie–Hellman Key Exchange
   - Public-key Certificates
3. Federated Identity Management
   - Identity Management
   - Identity Federation

Asymmetric Crypto and Hash Functions



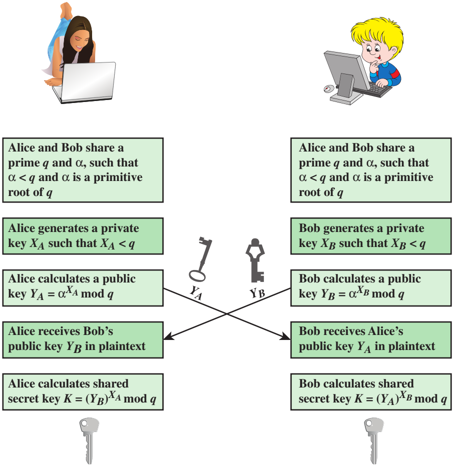
Figur : An overview of asymmetric crypto. Image: [Sta13].

Asymmetric Crypto and Hash Functions



Figur : An overview of using hash functions for message integrity and authentication. Image: [Sta13].

Diffie–Hellman Key Exchange



Alice and Bob share a prime q and α, such that $\alpha < q$ and α is a primitive root of q

Alice generates a private key X_A such that $X_A < q$

Alice calculates a public key $Y_A = \alpha^{X_A} \bmod q$

Alice receives Bob's public key Y_B in plaintext

Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$

Alice and Bob share a prime q and α, such that $\alpha < q$ and α is a primitive root of q

Bob generates a private key X_B such that $X_B < q$

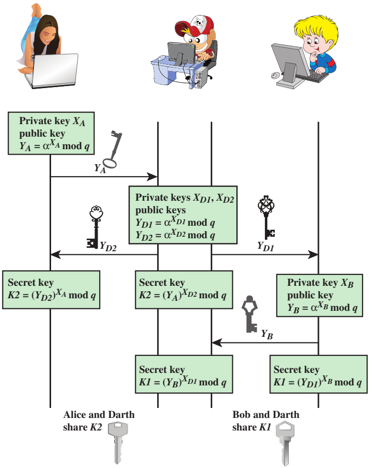
Bob calculates a public key $Y_B = \alpha^{X_B} \bmod q$

Bob receives Alice's public key Y_A in plaintext

Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$

Figur : A schematic overview of the Diffie–Hellan Key Exchange algorithm. Image: [Sta13].

Diffie–Hellman Key Exchange



Private key X_A
public key
$Y_A = \alpha^{X_A} \bmod q$

Y_A

Private keys X_{D1}, X_{D2}
public keys
$Y_{D1} = \alpha^{X_{D1}} \bmod q$
$Y_{D2} = \alpha^{X_{D2}} \bmod q$

Y_{D2}

Y_{D1}

Secret key
$K2 = (Y_{D2})^{X_A} \bmod q$

Secret key
$K2 = (Y_A)^{X_{D2}} \bmod q$

Private key X_B
public key
$Y_B = \alpha^{X_B} \bmod q$

Y_B

Secret key
$K1 = (Y_B)^{X_{D1}} \bmod q$

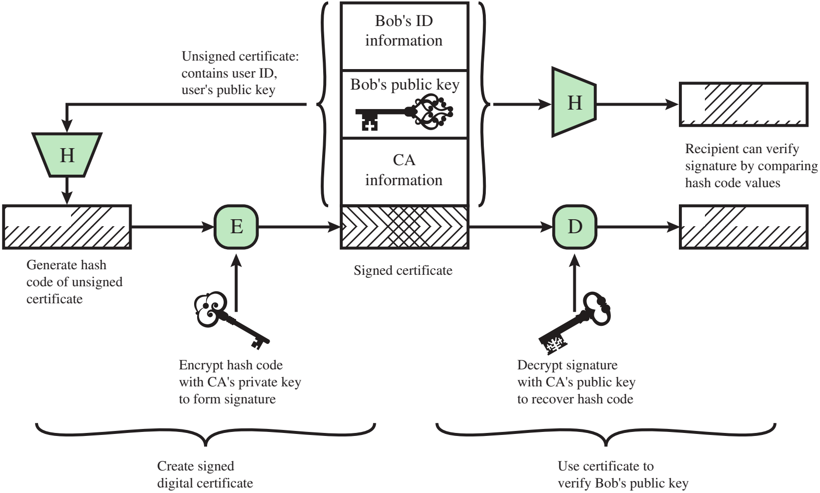
Secret key
$K1 = (Y_{D1})^{X_B} \bmod q$

Alice and Darth
share $K2$

Bob and Darth
share $K1$

Figur : Schematic overview of a Man-in-the-Middle Attack. Image: [Sta13].

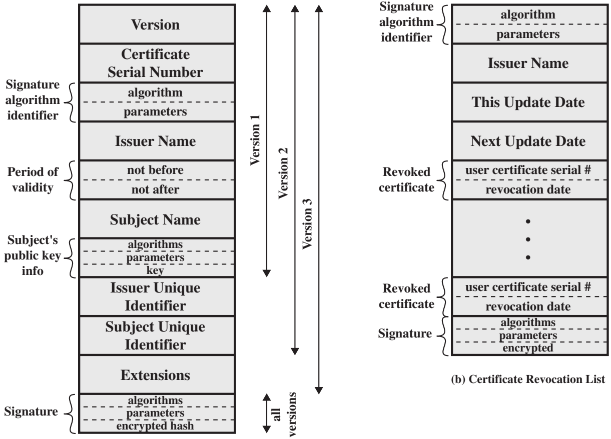
Public-key Certificates


Bob's ID information
Bob's public key
CA information
Unsigned certificate: contains user ID, user's public key
H
H
E
D
Signed certificate
Generate hash code of unsigned certificate
Encrypt hash code with CA's private key to form signature
Decrypt signature with CA's public key to recover hash code
Recipient can verify signature by comparing hash code values
Create signed digital certificate
Use certificate to verify Bob's public key

Figur : An overview of use of public-key certificates. Image: [Sta13].

Public-key Certificates

X.509

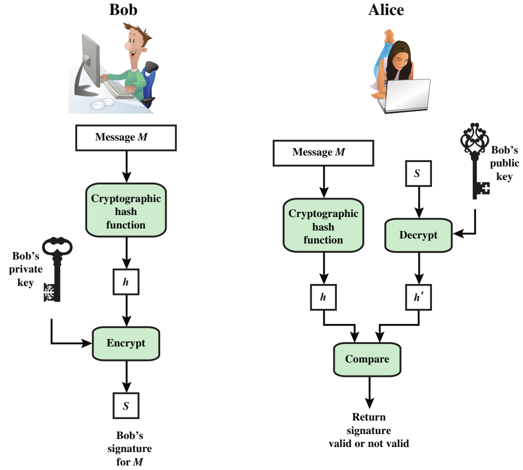


(a) X.509 Certificate

(b) Certificate Revocation List

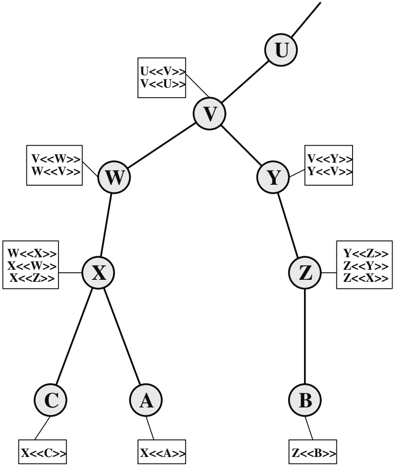
Figur : An overview of X.509 certificate format. Image: [Sta13].

Public-key Certificates



Figur : An overview of the digital signature process. Image: [Sta13].

Public-key Certificates

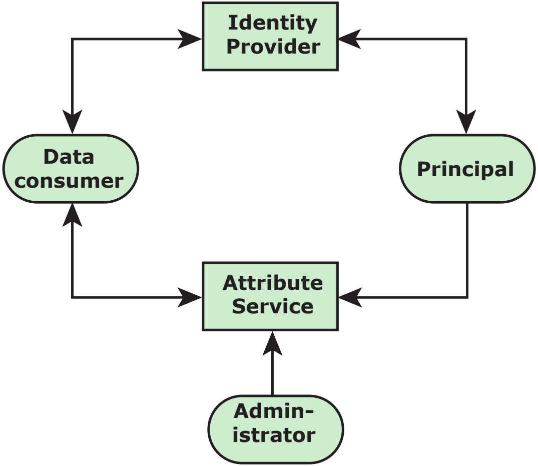


Figur : The X.509 certificate hierarchy. Image: [Sta13].

Översikt

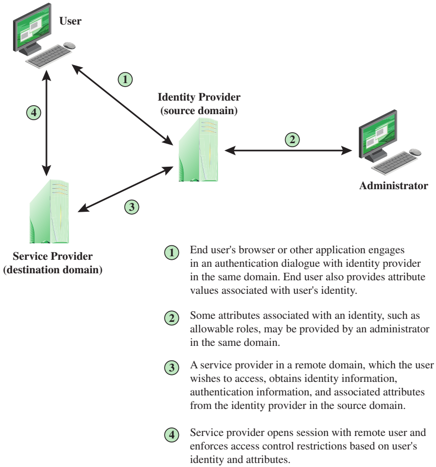
1. Symmetric Key Distribution
   - Symmetric Crypto
   - Key Distribution Centre (KDC)
   - Authentication
   - Kerberos IV
   - Kerberos V
2. Asymmetric Key Distribution
   - Asymmetric Crypto and Hash Functions
   - Diffie–Hellman Key Exchange
   - Public-key Certificates
3. Federated Identity Management
   - Identity Management
   - Identity Federation

Identity Management



Figur : An overview of a generic identity management system. Image: [Sta13].

Identity Federation



1. End user's browser or other application engages in an authentication dialogue with identity provider in the same domain. End user also provides attribute values associated with user's identity.

2. Some attributes associated with an identity, such as allowable roles, may be provided by an administrator in the same domain.

3. A service provider in a remote domain, which the user wishes to access, obtains identity information, authentication information, and associated attributes from the identity provider in the source domain.

4. Service provider opens session with remote user and enforces access control restrictions based on user's identity and attributes.

Figur : An overview of federated identity systems. Image: [Sta13].

Referenser I

Ross J. Anderson. *Security engineering : a guide to building dependable distributed systems.* 2. utg. Indianapolis, IN: Wiley, 2008. ISBN: 978-0-470-06852-6 (hbk.) URL: http://www.cl.cam.ac.uk/~rja14/book.html.

William Stallings. *Network security essentials : applications and standards.* 5. utg. International Edition. Pearson Education, 2013. ISBN: 978-0-273-79336-6.