

INFS 767 Fall 2001
RBAC Architectures and Mechanisms

Prof. Ravi Sandhu

AUTHORIZATION, TRUST AND RISK

- ❖ **Information security is fundamentally about managing**
 - authorization and
 - trust

so as to manage risk

© Ravi Sandhu 2001 2

THE OM-AM WAY

What? A
S
S
U
R
A
N
C
E

Objectives
Model
Architecture
Mechanism

How?

© Ravi Sandhu 2001 3

LAYERS AND LAYERS

- ❖ Multics rings
- ❖ Layered abstractions
- ❖ Waterfall model
- ❖ Network protocol stacks
- ❖ Napoleon layers
- ❖ RoFi layers
- ❖ OM-AM
- ❖ etcetera

© Ravi Sandhu 2001 4

OM-AM AND MANDATORY ACCESS CONTROL (MAC)

What? A
S
S
U
R
A
N
C
E

No information leakage
Lattices (Bell-LaPadula)
Security kernel
Security labels

How?

© Ravi Sandhu 2001 5

OM-AM AND DISCRETIONARY ACCESS CONTROL (DAC)

What? A
S
S
U
R
A
N
C
E

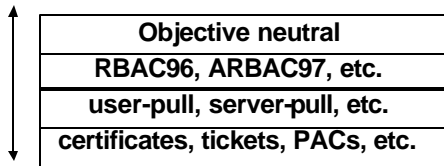
Owner-based discretion
numerous
numerous
ACLs, Capabilities, etc

How?

© Ravi Sandhu 2001 6

OM-AM AND ROLE-BASED ACCESS CONTROL (RBAC)

What?



How?

A
S
S
U
R
A
N
C
E

© Ravi Sandhu 2001

7

DISTRIBUTED RBAC (DRBAC) CASE STUDY

- ❖ Approximately a dozen physical sites
- ❖ Approximately 2-3 simulation models/site
- ❖ Fewer than 100 roles structured in a very shallow hierarchy
 - A subset of roles is used in any single simulation model
- ❖ Fewer than 100 users
- ❖ A user uses only one role at a time
 - Convenient but not critical
- ❖ Moderate rate of change

© Ravi Sandhu 2001

8

DISTRIBUTED RBAC (DRBAC) CASE STUDY

- ❖ **Permission-role assignment**
 - Locally determined at each simulation model
- ❖ **User-role assignment**
 - A user can be assigned to a role if and only if all simulation models using that role agree
 - A user is revoked from a role if and only if any simulation model using that role revokes the user

© Ravi Sandhu 2001

9

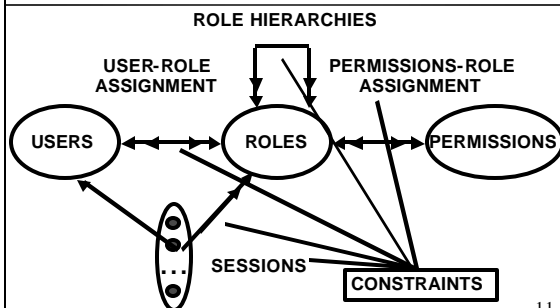
DISTRIBUTED RBAC (DRBAC) CASE STUDY

- ❖ Each simulation model has a security administrator role authorized to carry out these administrative tasks
- ❖ A simulation model can assign permissions to a role X at any time
 - even if X is previously unused in that simulation model
- ❖ Consequently any simulation model can revoke any user from any role!

© Ravi Sandhu 2001

10

RBAC3



© Ravi Sandhu 2001

11

MODEL CUSTOMIZATION

- ❖ Each session has a single role
- ❖ $SM = \{sm1, \dots, smk\}$, simulation models
- ❖ $OP = \{op1, \dots, opl\}$, operations
- ❖ $P = SM \times OP$, permissions
- ❖ $SMA = \{sma1, \dots, smk\}$, administrative roles
- ❖ $R \subset SMA = \bar{A}$
- ❖ Admin: $SM \leftarrow SMA$

© Ravi Sandhu 2001

12

MODEL CUSTOMIZATION

- ❖ Can formalize the administrative rules given earlier
- ❖ For each simulation model designate a unique user to be the chief security administrator who is authorized to assign and revoke users from the security administrator role for that model

© Ravi Sandhu 2001

13

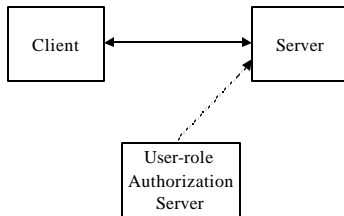
DRBAC ARCHITECTURES

- ❖ **Permission-role**
 - Enforced locally at each simulation model
- ❖ **Permission-role administration**
 - Enforced locally at each simulation model
 - May need to communicate to other simulation models
- ❖ **User-role**
 - See following slides
- ❖ **User-role administration**
 - Centralized or decentralized

© Ravi Sandhu 2001

14

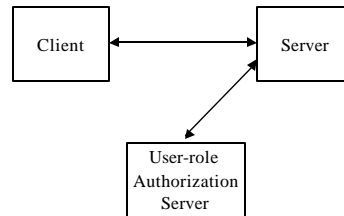
SERVER MIRROR



© Ravi Sandhu 2001

15

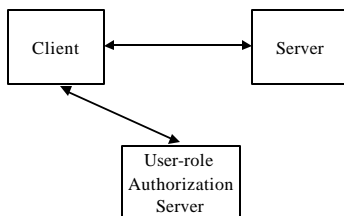
SERVER-PULL



© Ravi Sandhu 2001

16

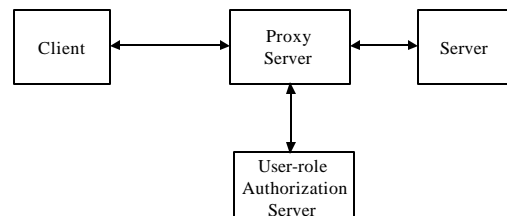
USER-PULL



© Ravi Sandhu 2001

17

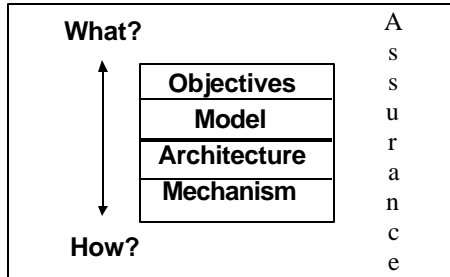
PROXY-BASED



© Ravi Sandhu 2001

18

THE OM-AM WAY



© Ravi Sandhu 2001

19

Secure Attribute Services on the Web

❖ WWW (World Wide Web)

- widely used for electronic commerce and business
- supports synthesis of technologies
- mostly, Web servers use identity-based access control
 - scalability problem

© Ravi Sandhu 2001

20

Background

- ❖ **An attribute**
 - a particular property of an entity
 - e.g., role, identity, SSN, clearance, etc.
- ❖ **If attributes are provided securely,**
 - Web servers can use those attributes
 - e.g., authentication, authorization, access control, electronic commerce, etc.
- ❖ **A successful marriage of the Web and secure attribute services is required**

© Ravi Sandhu 2001

21

User-Pull Architecture



© Ravi Sandhu 2001

22

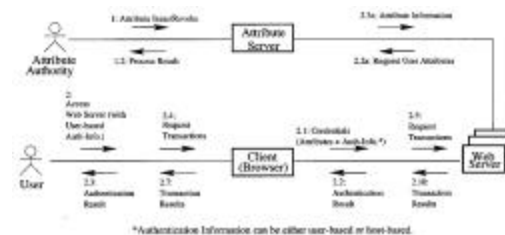
User-Pull Architecture

- ❖ **Each user**
 - pulls appropriate attributes from the Attribute Server
 - presents attributes and authentication information to Web servers
- ❖ **Each Web server**
 - requires both identification and attributes from users
- ❖ **High performance**
 - No new connections for attributes

© Ravi Sandhu 2001

23

Server-Pull Architecture



© Ravi Sandhu 2001

24

Related Technologies

- ❖ **Cookies**
 - in widespread current use for maintaining state of HTTP
 - becoming standard
 - not secure
- ❖ **Public-Key Certificates (X.509)**
 - support security on the Web based on PKI
 - standard
 - simply, bind users to keys
 - have the ability to be extended

© Ravi Sandhu 2001

25

Cookies

	Domain	Flag	Path	Cookie_Name	Cookie_Value	Secure	Date
Cookie 1	acme.com	TRUE	/	Name	Alice	FALSE	12/31/99
...							
Cookie n	acme.com	TRUE	/	Role	manager	FALSE	12/31/99

© Ravi Sandhu 2001

26

Security Threats to Cookies

- ❖ **Cookies are not secure**
 - No authentication
 - No integrity
 - No confidentiality
- ❖ **can be easily attacked by**
 - Network Security Threats
 - End-System Threats
 - Cookie Harvesting Threats

© Ravi Sandhu 2001

27

Secure Cookies on the Web



© Ravi Sandhu 2001

28

A Set of Secure Cookies

```
Test Editor V1.0.1 - cookies.txt, dir:/home/jarvis/netscape
# Netscape HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file. Do not edit.

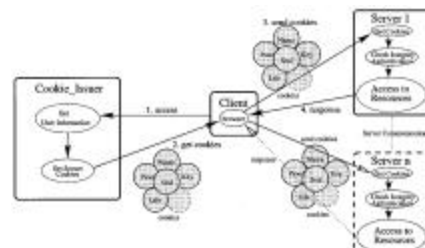
11st_gnu.edu TRUE / FALSE 918322569 Name Alice
11st_gnu.edu TRUE / FALSE 918322569 Role Manager
11st_gnu.edu TRUE / FALSE 918322567 Password
PR46MBE115CFW8MA91K0WJ2K123065D23F7899291D6V4J135R888F7-54J086AF72E1CF4
T3R0R5KCC2NCF5/44P2O5HAKWABU6F76G1BC7P2-q1T9P0K1CT9H4L650W961A90==788Q

11st_gnu.edu TRUE / FALSE 918322579 IP 129.174.144.95
11st_gnu.edu TRUE / FALSE 918322564 Sess
042D98E146AV8P8AM6BUK0WJ2K123065D23F7899291D6V4J135R888F7-54J086AF72E1CF4
03U1P5F088Q711142SC8W91#0B-R88S4770K9AF78X9L12-3X)H550#2AAAAY3712w8786AC1
2981230A9163H888888377V8B-05=0481
```

© Ravi Sandhu 2001

29

How to Use Secure Cookies



© Ravi Sandhu 2001

30

Applications of Secure Cookies

- ❖ User Authentication
- ❖ Electronic Transaction
- ❖ Eliminating Single-Point Failure
- ❖ Pay-per-Access
- ❖ Attribute-based Access Control

Authentication Cookies

	Domain	Flag	Path	Cookie Name	Cookie Value	Secure	Expiration
IP_Cookie	acme.com	TRUE	/	IP_Cookie	209.174.130.88	FALSE	1/3/01 13:05:00
Priv_Cookie	acme.com	TRUE	/	Priv_Cookie	hashed_password	FALSE	1/3/01 13:05:00
KT_Cookie	acme.com	TRUE	/	Kerberos_Ticket	{Alice, Kc=}Kc	FALSE	1/3/01 13:05:00
Sign_Cookie	acme.com	TRUE	/	Sign_Cookie	Signature_of_Alice	FALSE	1/3/01 13:05:00

Server-Pull Architecture

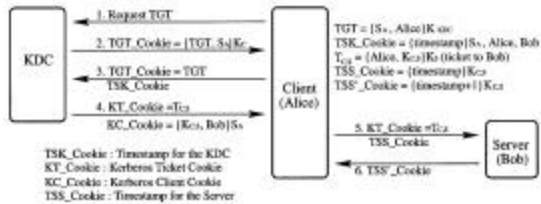
- ❖ Each user
 - > presents only authentication information to Web servers
- ❖ Each Web server
 - > pulls users' attributes from the Attribute Server
- ❖ Authentication information and attribute do not go together
- ❖ More convenient for users
- ❖ Less convenient for Web servers

Secure Cookies for Electronic Transactions

	Domain	Flag	Path	Cookie Name	Cookie Value	Secure	Expiration
Name_Cookie	acme.com	TRUE	/	Name_Cookie	Alice*	FALSE	1/3/01 13:05:00
Cart_Cookie	acme.com	TRUE	/	Cart_Cookie	numbers=1234567890&clothes=00123456	FALSE	1/3/01 13:05:00
Checkout_Cookie	acme.com	TRUE	/	Checkout_Cookie	EX=1234567890101101201301401501601701801901	FALSE	1/3/01 13:05:00
Like_Cookie	acme.com	TRUE	/	Like_Cookie	123456	FALSE	1/3/01 13:05:00
Priv_Cookie	acme.com	TRUE	/	Priv_Cookie	hashed_password	FALSE	1/3/01 13:05:00
Key_Cookie	acme.com	TRUE	/	Key_Cookie	encrypted_key**	FALSE	1/3/01 13:05:00
Sec_Cookie	acme.com	TRUE	/	Sec_Cookie	{numbers=1234567890&clothes=00123456}&EX=1234567890101101201301401501601701801901	FALSE	1/3/01 13:05:00

* Session ID can be encrypted in the cookie.
 ** Sec of Cookies can be either MAC or signed message digest of cookies.
 Note: Priv_Cookie can be replaced with one of the other authentication cookies in Figure 4.1.

Kerberos-Based Authentication by Secure Cookies



Secure Cookies for Pay-Per-Access

	Domain	Flag	Path	Cookie Name	Cookie Value	Secure	Expiration
Name_Cookie	acme.com	TRUE	/	Name_Cookie	Alice*	FALSE	1/3/01 13:05:00
Ticket_Cookie	acme.com	TRUE	/	Ticket_Cookie	EK=098765432101101201301401501601701801901	FALSE	1/3/01 13:05:00
Like_Cookie	acme.com	TRUE	/	Like_Cookie	123456	FALSE	1/3/01 13:05:00
Priv_Cookie	acme.com	TRUE	/	Priv_Cookie	hashed_password	FALSE	1/3/01 13:05:00
Key_Cookie	acme.com	TRUE	/	Key_Cookie	encrypted_key**	FALSE	1/3/01 13:05:00
Sec_Cookie	acme.com	TRUE	/	Sec_Cookie	{numbers=1234567890&clothes=00123456}&EX=1234567890101101201301401501601701801901	FALSE	1/3/01 13:05:00

* Session ID can be encrypted in the cookie.
 ** Sec of Cookies can be either MAC or signed message digest of cookies.
 Note: Priv_Cookie can be replaced with one of the other authentication cookies in Figure 4.1.

Secure Cookies for RBAC

Name	Domain	Path	Cookie Name	Cookie Value	Secure	Date
Name_Cookie	www.com	/	Name	Allow	FALSE	12/1/99
Role_Cookie	www.com	/	Role	Manager	FALSE	12/1/99
Life_Cookie	www.com	/	Life_Cookie	1234567	FALSE	12/1/99
Pass_Cookie	www.com	/	Pass_Cookie	Encrypted_Password	FALSE	12/1/99
IP_Cookie	www.com	/	IP_Cookie	129.174.142.88	FALSE	12/1/99
Send_Cookie	www.com	/	Send_Cookie	Tag:al,Signature	FALSE	12/1/99

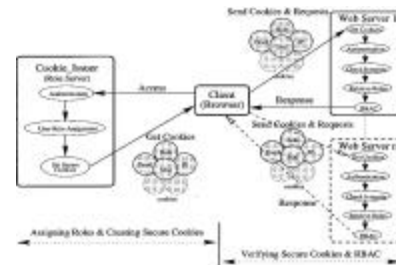
Cookie_Issuer Signs on the Cookies

* Blank of the passwords is an alternative in the interest of the Pass_Cookie.

© Ravi Sandhu 2001

37

RBAC on the Web by Secure Cookies



© Ravi Sandhu 2001

38

X.509 Certificate

- ❖ Digitally signed by a certificate authority
 - to confirm the information in the certificate belongs to the holder of the corresponding private key
- ❖ Contents
 - version, serial number, subject, validity period, issuer, optional fields (v2)
 - subject's public key and algorithm info.
 - extension fields (v3)
 - digital signature of CA
- ❖ Binding users to keys
- ❖ Certificate Revocation List (CRL)

© Ravi Sandhu 2001

39

X.509 Certificate

```

Certificate Content:
-----
Certificate:
    Data:
        Version: 3 (0x02)
        Serial Number: 1 (0x00000001)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: CN=www.com, OU=www.com, O=www.com, C=US
        Validity:
            Not Before: Aug 20 00:00:00 1999
            Not After:  Aug 20 23:59:59 1999
        Subject: CN=www.com, OU=www.com, O=www.com, C=US
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public-Key: (1024 bit)
                Modulus:
                    00:01:02:03:04:05:06:07:08:09:0a:0b:0c:0d:0e:0f:
                    10:11:12:13:14:15:16:17:18:19:1a:1b:1c:1d:1e:1f:
                    20:21:22:23:24:25:26:27:28:29:2a:2b:2c:2d:2e:2f:
                    30:31:32:33:34:35:36:37:38:39:3a:3b:3c:3d:3e:3f:
                    40:41:42:43:44:45:46:47:48:49:4a:4b:4c:4d:4e:4f:
                    50:51:52:53:54:55:56:57:58:59:5a:5b:5c:5d:5e:5f:
                    60:61:62:63:64:65:66:67:68:69:6a:6b:6c:6d:6e:6f:
                    70:71:72:73:74:75:76:77:78:79:7a:7b:7c:7d:7e:7f:
                    80:81:82:83:84:85:86:87:88:89:8a:8b:8c:8d:8e:8f:
                    90:91:92:93:94:95:96:97:98:99:9a:9b:9c:9d:9e:9f:
                    a0:a1:a2:a3:a4:a5:a6:a7:a8:a9:aa:ab:ac:ad:ae:af:
                    b0:b1:b2:b3:b4:b5:b6:b7:b8:b9:ba:bb:bc:bd:be:bf:
                    c0:c1:c2:c3:c4:c5:c6:c7:c8:c9:ca:cb:cc:cd:ce:cf:
                    d0:d1:d2:d3:d4:d5:d6:d7:d8:d9:da:db:dc:dd:de:df:
                    e0:e1:e2:e3:e4:e5:e6:e7:e8:e9:ea:eb:ec:ed:ee:ef:
                    f0:f1:f2:f3:f4:f5:f6:f7:f8:f9:fa:fb:fc:fd:fe:ff:
                Exponent: 65537 (0x010001)
            Private key algorithm: rsaEncryption
            Private-Key: (1024 bit)
                Modulus:
                    00:01:02:03:04:05:06:07:08:09:0a:0b:0c:0d:0e:0f:
                    10:11:12:13:14:15:16:17:18:19:1a:1b:1c:1d:1e:1f:
                    20:21:22:23:24:25:26:27:28:29:2a:2b:2c:2d:2e:2f:
                    30:31:32:33:34:35:36:37:38:39:3a:3b:3c:3d:3e:3f:
                    40:41:42:43:44:45:46:47:48:49:4a:4b:4c:4d:4e:4f:
                    50:51:52:53:54:55:56:57:58:59:5a:5b:5c:5d:5e:5f:
                    60:61:62:63:64:65:66:67:68:69:6a:6b:6c:6d:6e:6f:
                    70:71:72:73:74:75:76:77:78:79:7a:7b:7c:7d:7e:7f:
                    80:81:82:83:84:85:86:87:88:89:8a:8b:8c:8d:8e:8f:
                    90:91:92:93:94:95:96:97:98:99:9a:9b:9c:9d:9e:9f:
                    a0:a1:a2:a3:a4:a5:a6:a7:a8:a9:aa:ab:ac:ad:ae:af:
                    b0:b1:b2:b3:b4:b5:b6:b7:b8:b9:ba:bb:bc:bd:be:bf:
                    c0:c1:c2:c3:c4:c5:c6:c7:c8:c9:ca:cb:cc:cd:ce:cf:
                    d0:d1:d2:d3:d4:d5:d6:d7:d8:d9:da:db:dc:dd:de:df:
                    e0:e1:e2:e3:e4:e5:e6:e7:e8:e9:ea:eb:ec:ed:ee:ef:
                    f0:f1:f2:f3:f4:f5:f6:f7:f8:f9:fa:fb:fc:fd:fe:ff:
                Exponent: 65537 (0x010001)
        Signature:
            Algorithm: sha1WithRSAEncryption
            Full Name (SHA1 Digest, RSA Signature, ASN1 Encapsulated)
            00:01:02:03:04:05:06:07:08:09:0a:0b:0c:0d:0e:0f:
            10:11:12:13:14:15:16:17:18:19:1a:1b:1c:1d:1e:1f:
            20:21:22:23:24:25:26:27:28:29:2a:2b:2c:2d:2e:2f:
            30:31:32:33:34:35:36:37:38:39:3a:3b:3c:3d:3e:3f:
            40:41:42:43:44:45:46:47:48:49:4a:4b:4c:4d:4e:4f:
            50:51:52:53:54:55:56:57:58:59:5a:5b:5c:5d:5e:5f:
            60:61:62:63:64:65:66:67:68:69:6a:6b:6c:6d:6e:6f:
            70:71:72:73:74:75:76:77:78:79:7a:7b:7c:7d:7e:7f:
            80:81:82:83:84:85:86:87:88:89:8a:8b:8c:8d:8e:8f:
            90:91:92:93:94:95:96:97:98:99:9a:9b:9c:9d:9e:9f:
            a0:a1:a2:a3:a4:a5:a6:a7:a8:a9:aa:ab:ac:ad:ae:af:
            b0:b1:b2:b3:b4:b5:b6:b7:b8:b9:ba:bb:bc:bd:be:bf:
            c0:c1:c2:c3:c4:c5:c6:c7:c8:c9:ca:cb:cc:cd:ce:cf:
            d0:d1:d2:d3:d4:d5:d6:d7:d8:d9:da:db:dc:dd:de:df:
            e0:e1:e2:e3:e4:e5:e6:e7:e8:e9:ea:eb:ec:ed:ee:ef:
            f0:f1:f2:f3:f4:f5:f6:f7:f8:f9:fa:fb:fc:fd:fe:ff:
    -----
    
```

© Ravi Sandhu 2001

40

Smart Certificates

- ❖ Short-Lived Lifetime
 - More secure
 - typical validity period for X.509 is months (years)
 - users may leave copies of the corresponding keys behind
 - the longer-lived certificates have a higher probability of being attacked
 - No Certificate Revocation List (CRL)
 - simple and less expensive PKI

© Ravi Sandhu 2001

41

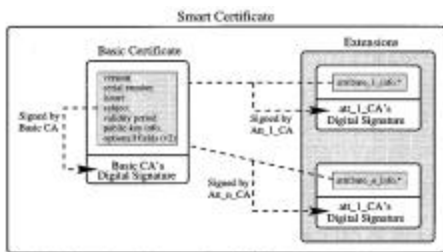
Smart Certificates

- ❖ Containing Attributes Securely
 - Web servers can use secure attributes for their purposes
 - Each authority has independent control on the corresponding information
 - basic certificate (containing identity information)
 - each attribute can be added, changed, revoked, or re-issued by the appropriate authority
 - e.g., role, credit card number, clearance, etc.
 - Short-lived certificate can remove CRLs

© Ravi Sandhu 2001

42

Separate CAs in a Certificate



* attributes info, attributes, attributes issuer, validity period of attributes, etc.

© Ravi Sandhu 2001

43

Smart Certificates

- ❖ **Postdated Certificates**
 - The certificate becomes valid at some time in the future
 - possible to make a smart certificate valid for a set of duration
 - supports convenience
- ❖ **Confidentiality**
 - Sensitive information can be
 - encrypted in smart certificates
 - e.g. passwords, credit card numbers, etc.

© Ravi Sandhu 2001

44

A Smart Certificate

```

Certificate Content
-----
Version: 3 (2001)
Serial Number: 1
Issuer: C=US, E=Sandhu@Ravi_Sandhu.com, OU=Ravi_Sandhu, CN=Ravi_Sandhu
Validity Period: Not Valid Yet
Public Key Info: RSA, Public Key Algorithm: RSA, Public Key Data:
-----BEGIN PUBLIC KEY-----
MIGIQAIBAgEAAQ==
-----END PUBLIC KEY-----
Signature: sha1WithRSAEncryption
-----BEGIN CERTIFICATE-----
MIICCAIBAgEAAQ==
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
MIICCAIBAgEAAQ==
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICCAIBAgEAAQ==
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
MIICCAIBAgEAAQ==
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICCAIBAgEAAQ==
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
MIICCAIBAgEAAQ==
-----END PRIVATE KEY-----

```

© Ravi Sandhu 2001

45

Applications of Smart Certificates

- ❖ **On-Duty Control**
- ❖ **Compatible with X.509**
- ❖ **User Authentication**
- ❖ **Electronic Transaction**
- ❖ **Eliminating Single-Point Failure**
- ❖ **Pay-per-Access**
- ❖ **Attribute-based Access Control**

© Ravi Sandhu 2001

46

Injecting RBAC to Secure a Web-based Workflow System

Gail-Joon Ahn and Ravi Sandhu
George Mason University

Myong Kang and Joon Park
Naval Research Laboratory

WORKFLOW MANAGEMENT SYSTEMS

- ⇒ **Control and coordinate processes that may be processed by different processing entities**
- ⇒ **Received much attention**
- ⇒ **Marriage with Web technology**
- ⇒ **Minimal security services**

© Ravi Sandhu 2001

48

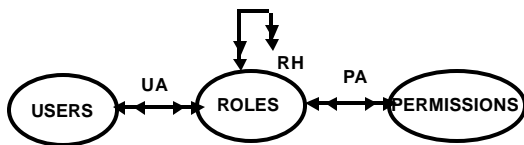
OBJECTIVE

- Inject role-based access control (RBAC) into an existing web-based workflow system

WHY RBAC?

- A mechanism which allows and promotes an organization-specific access control policy based on roles
- Has become widely accepted as the proven technology

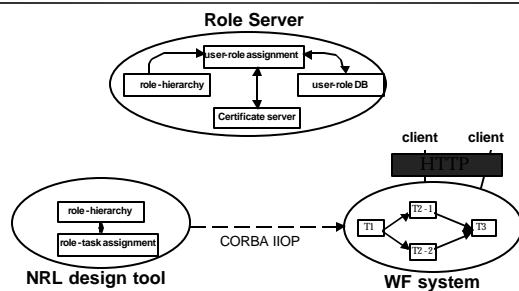
SIMPLIFIED RBAC MODEL



ROLE-BASED SECURE WORKFLOW SYSTEM

- Workflow Design Tool
- Workflow (WF) System
- Role Server

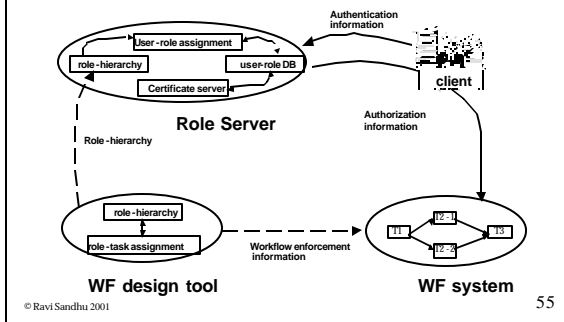
BASIC COMPONENTS



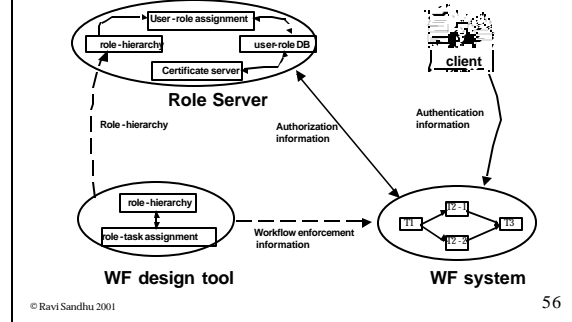
ARCHITECTURES

- USER-PULL STYLE**
- SERVER-PULL STYLE**

USER-PULL STYLE



SERVER-PULL STYLE



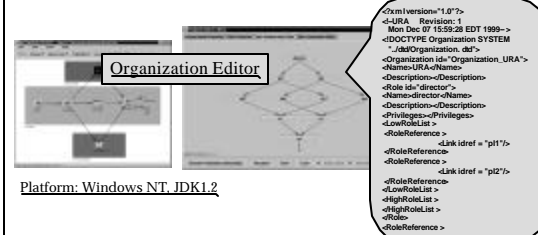
NRL (Naval Research Lab.) DESIGN TOOL

- design workflow model
- create role and role hierarchies
- assign role to task
- exporting role hierarchies to role server

© Ravi Sandhu 2001

57

NRL DESIGN TOOL (Cont'd)



© Ravi Sandhu 2001

58

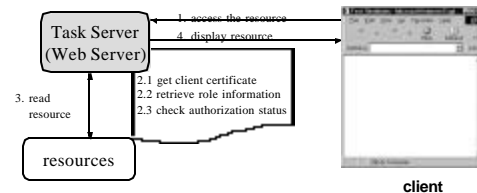
WORKFLOW SYSTEM

- each task server is web server
- user should present client authentication certificate
- user's privilege is authorized by content of certificate (specially client's role information)

© Ravi Sandhu 2001

59

ROLE AUTHORIZATION ON WORKFLOW SYSTEM



© Ravi Sandhu 2001

60

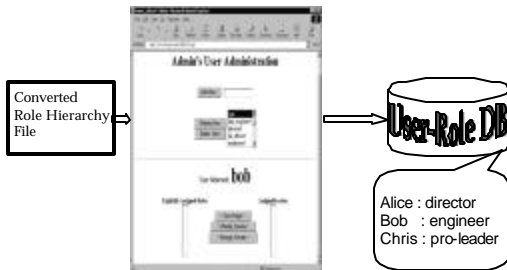
ROLE SERVER

- ⇒ User Role Assignment
- ⇒ Certificate Server

USER ROLE ASSIGNMENT

- ⇒ maintain role hierarchies and user database
- ⇒ assign users to roles
- ⇒ generate user-role database

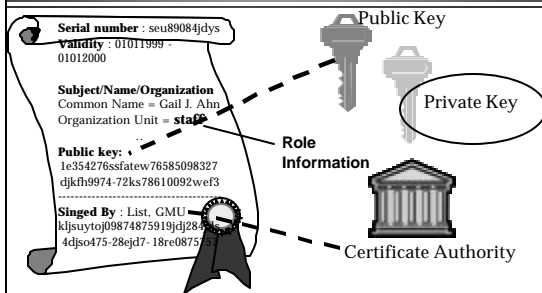
USER ROLE ASSIGNMENT (Cont'd)



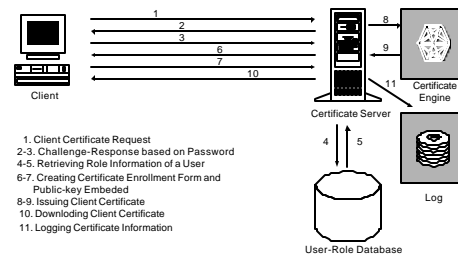
CERTIFICATE SERVER

- ⇒ authenticate client
- ⇒ retrieve client's role information from user-role database
- ⇒ issue certificate with client's role information

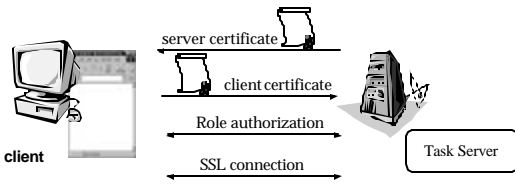
X.509 CERTIFICATE



CERTIFICATE ISSUE



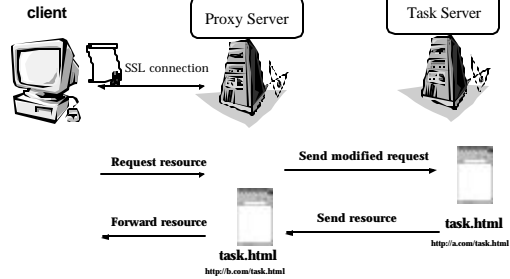
CERTIFICATE AUTHORIZATION OVER SSL



© Ravi Sandhu 2001

67

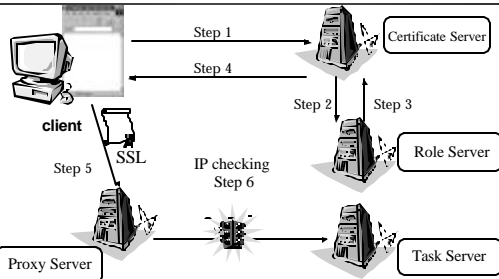
REVERSE PROXYING (MINIMAL CHANGES IN SERVER SIDE)



© Ravi Sandhu 2001

68

FINAL SCENARIO



© Ravi Sandhu 2001

69