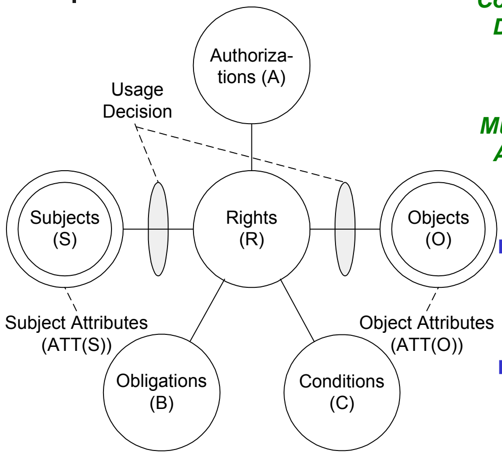
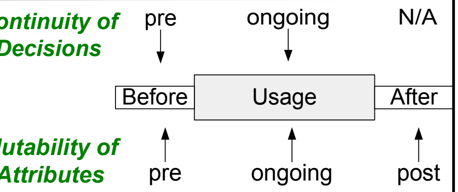
UCON Attribute Mutability, UCON Architectures

ISA 767, Secure Electronic Commerce

Xinwen Zhang, xzhang6@gmu.edu
George Mason University

UCON$_{ABC}$ Model Components: 3 Decision Factors & 2 Properties



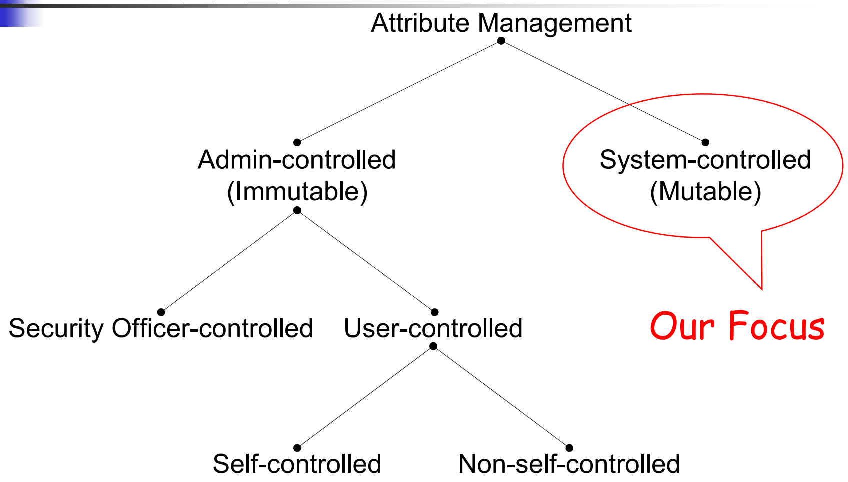


- Continuity Property: Decision can be made during usage for continuous enforcement
- Mutability Property: Attributes can be updated as side-effects of subjects' actions

Attributes in Usage Control

- Attributes are information or properties associated with subjects or objects
  - E.g., ID, Role, Clearance/classification, membership, credit, etc.
- Subject Attributes and Object Attributes are used for authorization decision
- Attributes may have to be updated
  - **Immutable Attributes:** Attribute updates can be made by administrative actions
  - **Mutable Attributes:** attributes can be modified as side effects of usage

Attribute Management Taxonomy



Attribute Management

Admin-controlled (Immutable)

System-controlled (Mutable)

Security Officer-controlled

User-controlled

Our Focus

Self-controlled

Non-self-controlled

Attribute Management:
Admin-controlled vs. System-controlled

- Admin-controlled (Immutable)
  - Updates involve administrative decisions and actions
  - Admin can be security officer, user (self, non-self)
- System-controlled (Mutable)
  - Updates are made as side effects of users' usage on objects.
  - Our focus is here

Mutable Attributes

- Temporary Attributes (stateless)
  - Alive only for a single usage
  - Exist only in mutable attributes
  - E.g., Usage start time, last active time, etc.
- Persistent Attributes (stateful)
  - Live for multiple usage decisions
  - Exist in both mutable and immutable attributes
  - E.g., Total usage hours, user credit balance, etc.
- Utilization of temporary attributes is a design decision and can be eliminated in some cases.
  - Temporary subject attributes can be stored as a form of elements of persistent object attributes

Mutability Variations

- Mutability for
  - Exclusive/Inclusive Attributes
    - History based policies
    - E.g., Dynamic SOD, Chinese Wall policy
  - Consumable/creditable Attributes
    - E.g., Limited # of Usage, payment, mileage, etc
  - Immediate Revocation
    - To support continuous control throughout usages
  - Obligation
    - Attribute update as a result of obligation fulfillment
  - Dynamic Confinement
    - E.g., High Watermark in MAC

Mutability for Exclusive/Inclusive Attributes

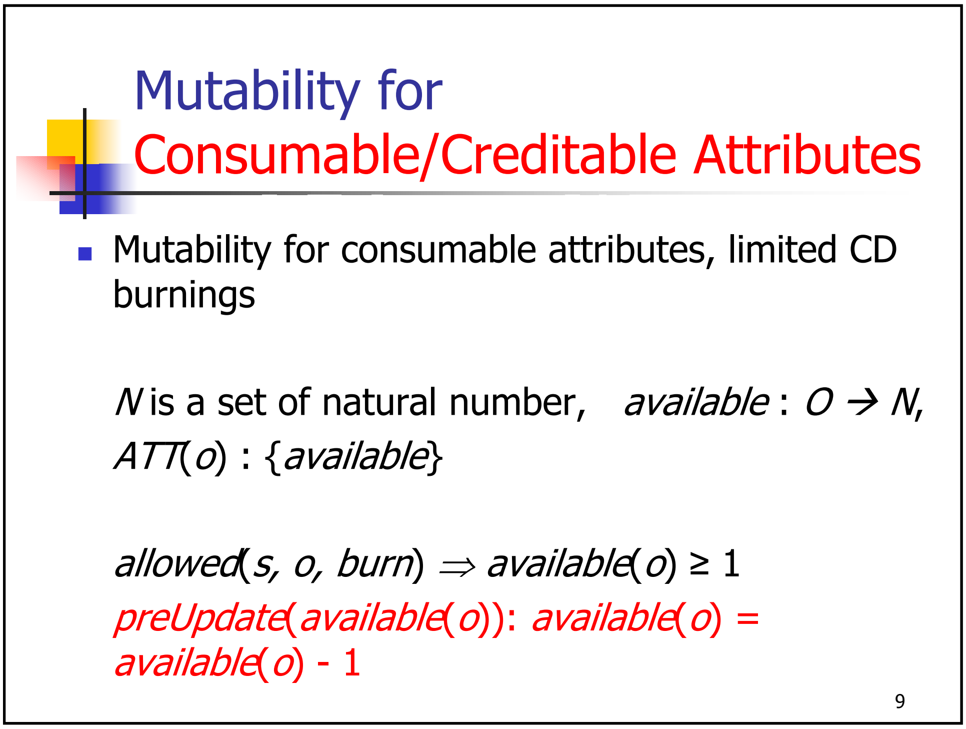
- *Object-based DSOD*

  ID is a set of identification number. *T* is a set of object type name.
  ROLE is a partially ordered set of role names.

  $uid : S \rightarrow ID, \quad sRole : S \rightarrow 2^{ROLE}, \quad type : O \rightarrow T$
  $prepareId : O \rightarrow ID, \quad issueId : O \rightarrow ID, \quad R : issue; prepare$
  $ATT(s) = \{uid, sRole\}, \quad ATT(o) = \{type, prepareId, issueId\}$

  $allowed(s, o, prepare) \Rightarrow type(o) = \text{`}check\text{'}, sRole(s) \geq \text{`}purchaseClerk\text{'}$
  $preUpdate(prepareId(o)) : prepareId(o) = uid(s)$

  $allowed(s, o, issue) \Rightarrow type(o) = \text{`}check\text{'}, sRole(s) \geq \text{`}accountClerk\text{'},$
  $\boldsymbol{uid(s) \neq prepareId(o)}$
  $preUpdate(issueId(o)) : issueId(o) = uid(s)$

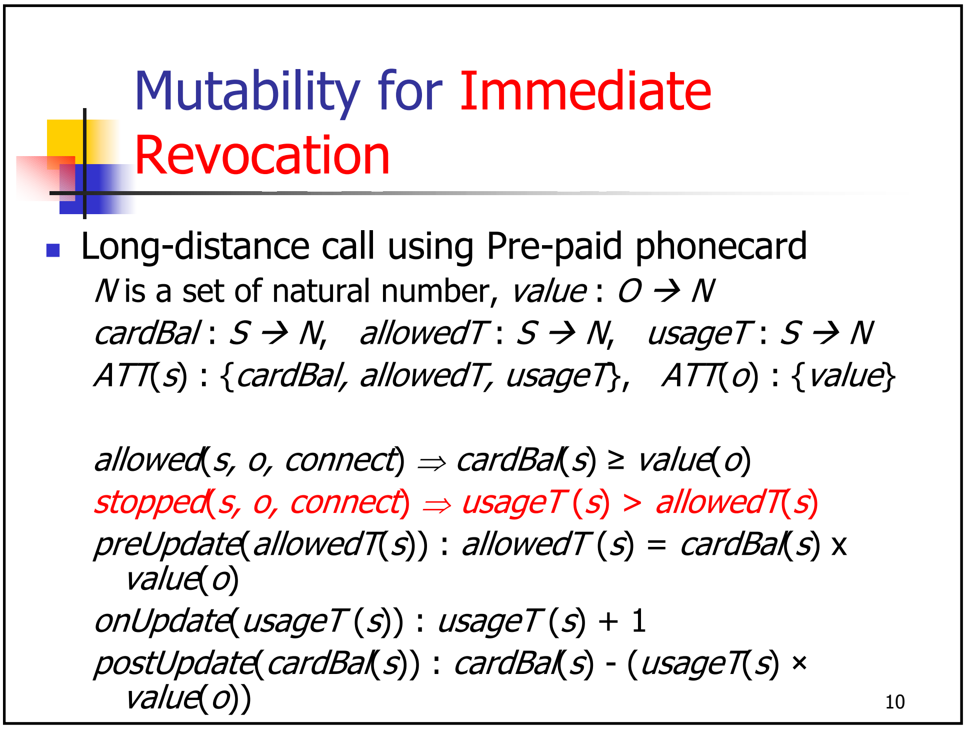


Mutability for Consumable/Creditable Attributes

- Mutability for consumable attributes, limited CD burnings

  N is a set of natural number, $available : O \rightarrow N$, $ATT(o) : \{available\}$

  $allowed(s, o, burn) \Rightarrow available(o) \geq 1$
  $preUpdate(available(o)): available(o) = available(o) - 1$



Mutability for Immediate Revocation

- Long-distance call using Pre-paid phonecard
  N is a set of natural number, $value : O \rightarrow N$
  $cardBal : S \rightarrow N,\quad allowedT : S \rightarrow N,\quad usageT : S \rightarrow N$
  $ATT(s) : \{cardBal, allowedT, usageT\},\quad ATT(o) : \{value\}$

  $allowed(s, o, connect) \Rightarrow cardBal(s) \geq value(o)$
  $stopped(s, o, connect) \Rightarrow usageT(s) > allowedT(s)$
  $preUpdate(allowedT(s)) : allowedT(s) = cardBal(s) \text{ x } value(o)$
  $onUpdate(usageT(s)) : usageT(s) + 1$
  $postUpdate(cardBal(s)) : cardBal(s) - (usageT(s) \times value(o))$

Mutability for Obligation

- License agreements for first time users only

$OBS = S, \quad OBO = \{license\_agreement\}, \quad OB = \{agree\}$

$registered : S \rightarrow \{yes, no\}, \quad ATT(s) = \{registered\}$

$$getPreOBL(s, o, r) = \begin{cases} (s, license\_agreement, agree), & \text{if } registered(s) = \text{'no'}; \\ \varnothing, & \text{if } registered(s) = \text{'yes'}. \end{cases}$$

$allowed(s, o, r) \Rightarrow preFulfilled(getPreOBL(s, o, r))$

$preUpdate(registered(s)) : registered(s) = \text{`} yes\text{'}$

Mutability for Dynamic Confinement

- MAC policies with high watermark property

L is a lattice of security labels with dominance relation $\geq$

$clearance : S \rightarrow L, \quad maxClearance : S \rightarrow L,$

$classification : O \rightarrow L$

$ATT(S) = \{clearance, maxClearance\}, ATT(O) = \{classification\}$

$allowed(s, o, read) \Rightarrow maxClearance(s) \geq classification(o)$

$preUpdate(clearance(s)) : clearance(s) = LUB(clearance(s), classification(o))$

Discussion

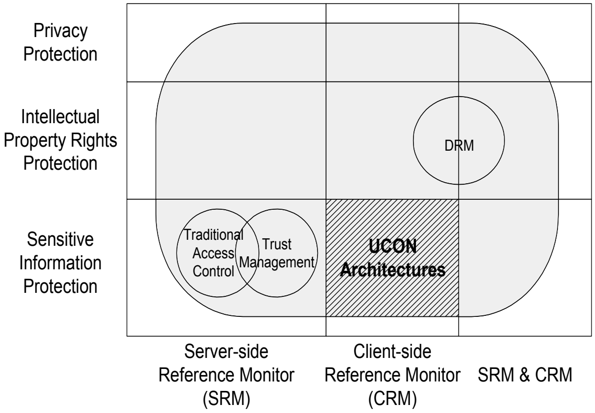
- Mutability variations are not mutually exclusive
  - Multiple mutability variations can be used in a single example.
- Updates can be made on either subject attributes or object attributes
  - In some cases, a policy can be realized by utilizing either subject attributes or object attributes

Conclusions and Future Works

- Consolidated analysis of Attributes and Attribute mutability in a single framework of usage control
  - Temporary and persistent attributes
  - Taxonomy of attribute management
  - Mutable attributes and variations of mutability
  - Mutability with continuity property
- Future research
  - Attribute management for admin-controlled attribute updates (immutable attributes)
  - Further study on attribute mutability

Usage Control Architectures

UCON Architectures


Privacy Protection
Intellectual Property Rights Protection
Sensitive Information Protection
DRM
Traditional Access Control
Trust Management
UCON Architectures
Server-side Reference Monitor (SRM)
Client-side Reference Monitor (CRM)
SRM & CRM

- We narrow down our focus so we can discuss in detail how UCON can be realized in architecture level
  - Sensitive information protection X CRM
- First systematic study for generalized security architectures for digital information dissemination
- Architectures can be extended to include payment function

Security Architectures for Controlled Digital Information Dissemination

- To develop systematic security architectures for controlling and tracking digital information dissemination and its use.
- We are focusing on Payment-Free Type (PFT).
  - Control dissemination solutions of PBT have been developed actively in commercial sector
  - However, no systematic study for more generalized security architectures for controlled digital information dissemination has been done
  - Architectures can be extended to include payment function
- Most for confidentiality
  - Controlled information sharing

Three Factors of Security Architectures

- Security Architectures have been developed based on the following three factors
- Three factors
  - Virtual Machine (VM)
  - Control Set (CS)
  - Distribution Style

Three Factors of Security Architectures (continued)

- Virtual Machine (VM)
  - A module that runs on top of vulnerable computing environment and has control functions to provide the means to control and manage access and usage of digital information
  - Foundation of use-control technologies
  - Needs for specialized (trusted) client software/hardware

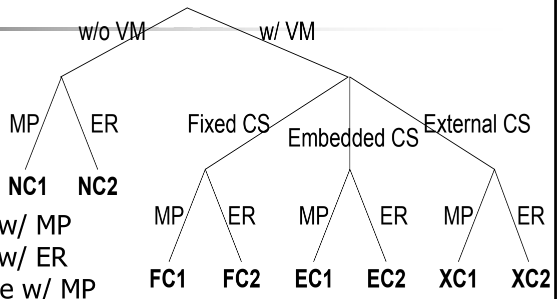
Three Factors of Security Architectures (continued)

- Control Set (CS)
  - A list of access rights and usage rules that is used by the virtual machine to control a recipient's access to and usage of digital information
    - A *fixed control set* is hardwired into the virtual machine
    - An *embedded control set* is bound to each digital object
    - An *external control set* is separate and independent from the digital object

Three Factors of Security Architectures (continued)

- Distribution Style
  - Message Push (MP) style
    - Digital information is sent to each recipient
  - External Repository (ER) style
    - Each recipient obtains the digital information from dissemination server on the network

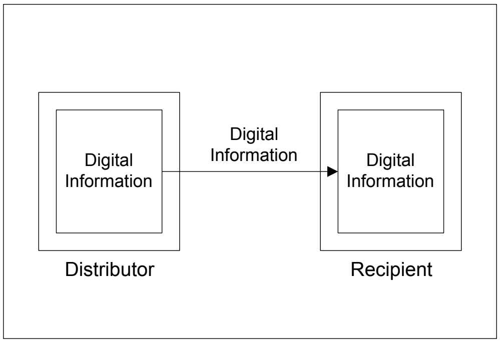
Architecture Taxonomy



VM: Virtual Machine
CS: Control Set
MP: Message Push
ER: External Repository

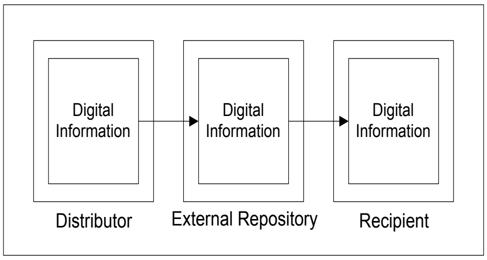
NC1: No control architecture w/ MP
NC2: No control architecture w/ ER
FC1: Fixed control architecture w/ MP
FC2: Fixed control architecture w/ ER
EC1: Embedded control architecture w/ MP
EC2: Embedded control architecture w/ ER
XC1: External control architecture w/ MP
XC2: External control architecture w/ ER

No Control Architecture w/ Message Push (NC1)



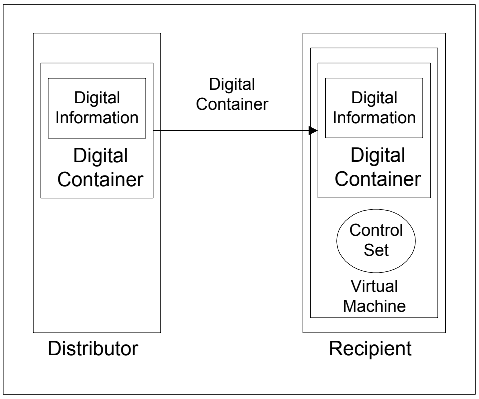
- Distributor directly sends a copy of digital contents to each recipient
- Each recipients stores the copy of digital information at local storage
- After distribution, no direct means to control the distributed digital information
- To access the digital information from multiple system, the recipient needs to transport the information

No Control Architecture w/ External Repository (NC2)



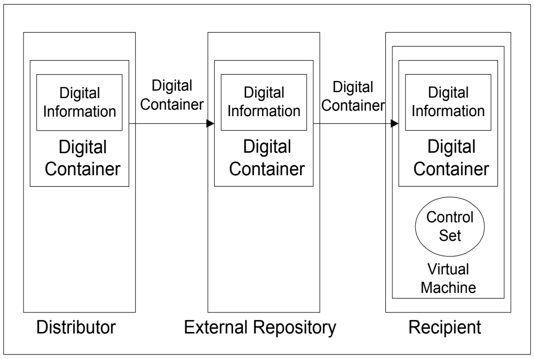
- Digital information is sent to an external repository server for distribution
- A recipient must connect to the external repository to access the digital content
- Once a recipient has received the digital contents, there is no way to control access or usage

Fixed Control Architecture w/ Message Push (FC1)



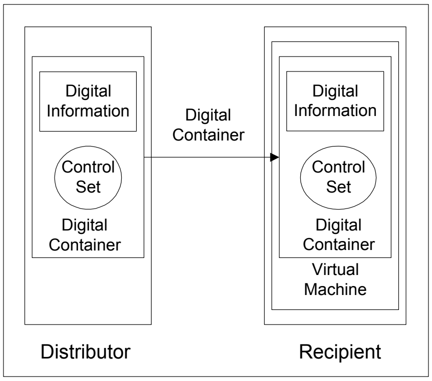
- Digital content is encapsulated in a digital container
- Control set is encoded into virtual machine
- The control set cannot be changed after the distribution of the virtual machine
- Access is controlled based on control set
- Each recipient should keep the received information for further access to it

Fixed Control Architecture w/ External Repository (FC2)



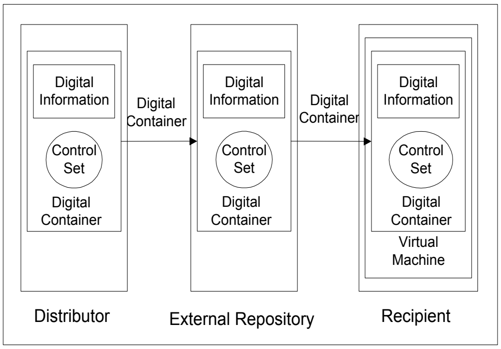
- Similar to FC1, except that digital container is sent to external repository for distribution
- A recipient must connect to the external repository to access or download the digital container
- Accessibility to the content by a single recipient from multiple computers

Embedded Control Architecture w/ Message Push (EC1)



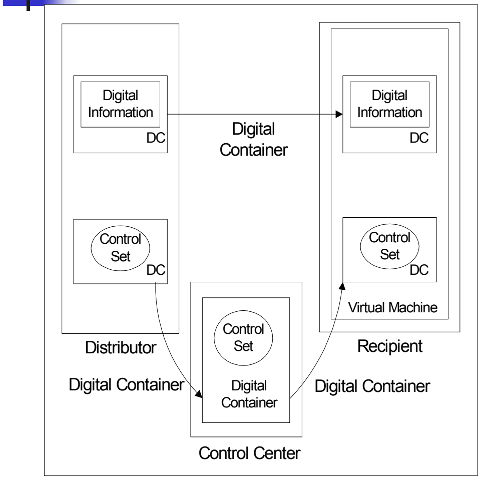
- Control set is embedded in the digital container with digital information
- Distributed content will be controlled based only on the pre-set access rights and usage rules
- After distribution, distributor cannot change the control set of the distributed digital content
- Recipients can access digital content without any network connection
- Only pre-set revocation is available

Embedded Control Architecture w/ External Repository (EC2)



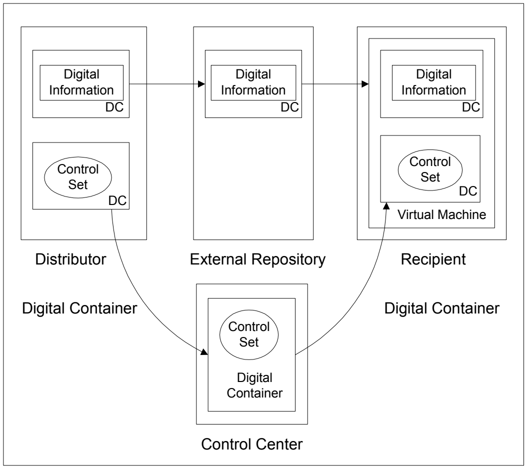
- Digital container is sent to the external repository server for distribution
- If digital container is prohibited from being locally stored, the distributor can revoke a previous granted access by changing control set

External Control Architecture w/ Message Push (XC1)


Digital Information
DC
Control Set
DC
Distributor
Digital Container
Digital Container
Digital Information
DC
Control Set
DC
Virtual Machine
Recipient
Digital Container
Control Set
Digital Container
Control Center

- Control set can be encapsulated independently from digital content
- Two possible options:
  - Network connection is always required
  - Network connection is required from time to time (one time connection is possible)

External Control Architecture w/ External Repository (XC2)


Digital Information
DC
Control Set
DC
Distributor
Digital Container
Digital Information
DC
External Repository
Digital Information
DC
Control Set
DC
Virtual Machine
Recipient
Digital Container
Control Set
Digital Container
Control Center

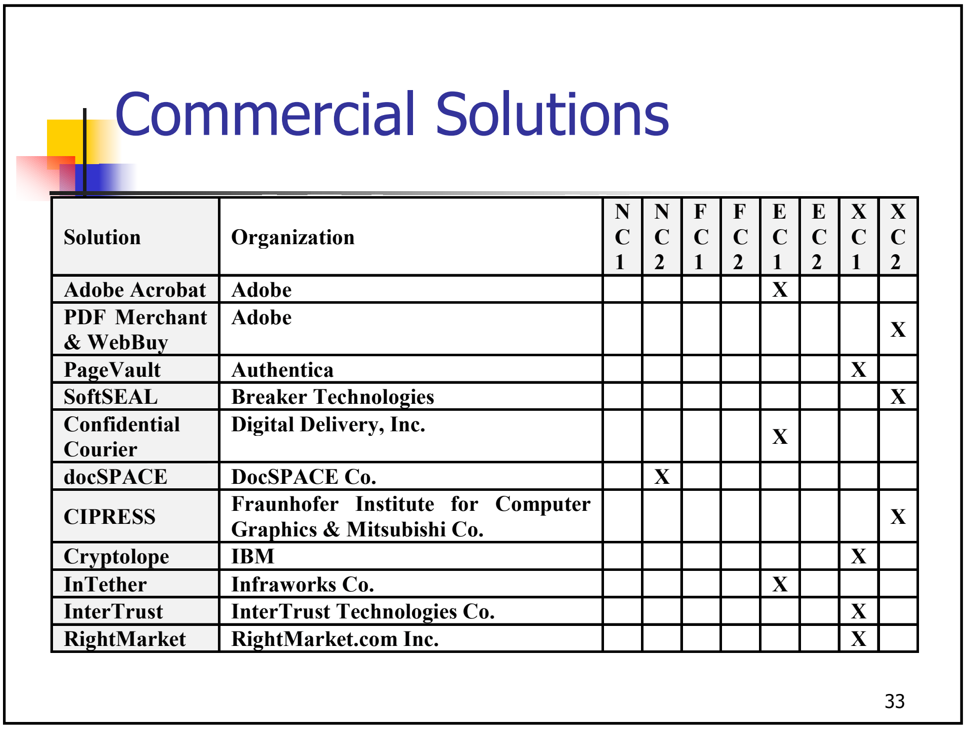
- Separation of content and access rights
- 4 variations
  - Both encapsulated digital content and encapsulated control set can be stored on recipient’s local storage
  - Encapsulated digital content is freely available, but control set cannot be locally stored
  - Only encapsulated control set can be stored
  - Neither can be stored locally

Security Characteristics

	Characteristics	NC1	NC2	FC1	FC2	EC1	EC2	XC1	XC2
C1	Disseminator can control access and usage of disseminated digital information			Y	Y	Y	Y	Y	Y
C2	Disseminator can change recipients’ access rights after dissemination						Y	Y	Y
C3	Re-disseminated digital information can be protected			Y	Y	Y	Y	Y	Y
C4	Special client software (virtual machine) is vulnerable to attacks			Y	Y	Y	Y	Y	Y
C5	Tracking re-disseminated digital information is possible	Y	Y	Y	Y	Y	Y	Y	Y

Functional Characteristics

	Characteristics	NC1	NC2	FC1	FC2	EC1	EC2	XC1	XC2
C6	Disseminated digital container is reusable for other recipients by re-dissemination							Y	Y
C7	Digital information does not have to be on recipient’s storage		Y		Y		Y		Y
C8	Digital information can be accessible from any machine if it is connected to network		Y		Y		Y		Y
C9	Recipient should carry digital information to access it from multiple machines	Y		Y		Y		Y	
C10	Special client software (virtual machine) is required			Y	Y	Y	Y	Y	Y
C11	In case of large digital information, download time can be significantly costly		Y		Y		Y		Y
C12	Every access to digital information requires network connection.								
C13	The architecture can be supported without network connection	Y		Y		Y			
C14	Control center trusted by both distributors and recipients is mandatory							Y	Y



Commercial Solutions

Solution	Organization	NC1	NC2	FC1	FC2	EC1	EC2	XC1	XC2
Adobe Acrobat	Adobe					X			
PDF Merchant & WebBuy	Adobe								X
PageVault	Authentica							X	
SoftSEAL	Breaker Technologies								X
Confidential Courier	Digital Delivery, Inc.					X			
docSPACE	DocSPACE Co.		X						
CIPRESS	Fraunhofer Institute for Computer Graphics & Mitsubishi Co.								X
Cryptolope	IBM							X	
InTether	Infraworks Co.					X			
InterTrust	InterTrust Technologies Co.							X	
RightMarket	RightMarket.com Inc.							X	