浙江大学计算机学院
数字媒体与网络技术

Digital Asset Management
数字媒体资源管理

4. Digital Rights Management

任课老师：张宏鑫
2011-10-09

About homework-02

Digital Rights Management Revisit

- DRM and movie industry: DVD CSS
- DRM and music industry:
  - audio CD: from sony BMG
  - internet music: iTunes store
- E-Books: Adobe Acrobat, M$ Reader, Kindle

- permission
- restriction
- obligation

Rights

own

over

- rights holder
- end customer

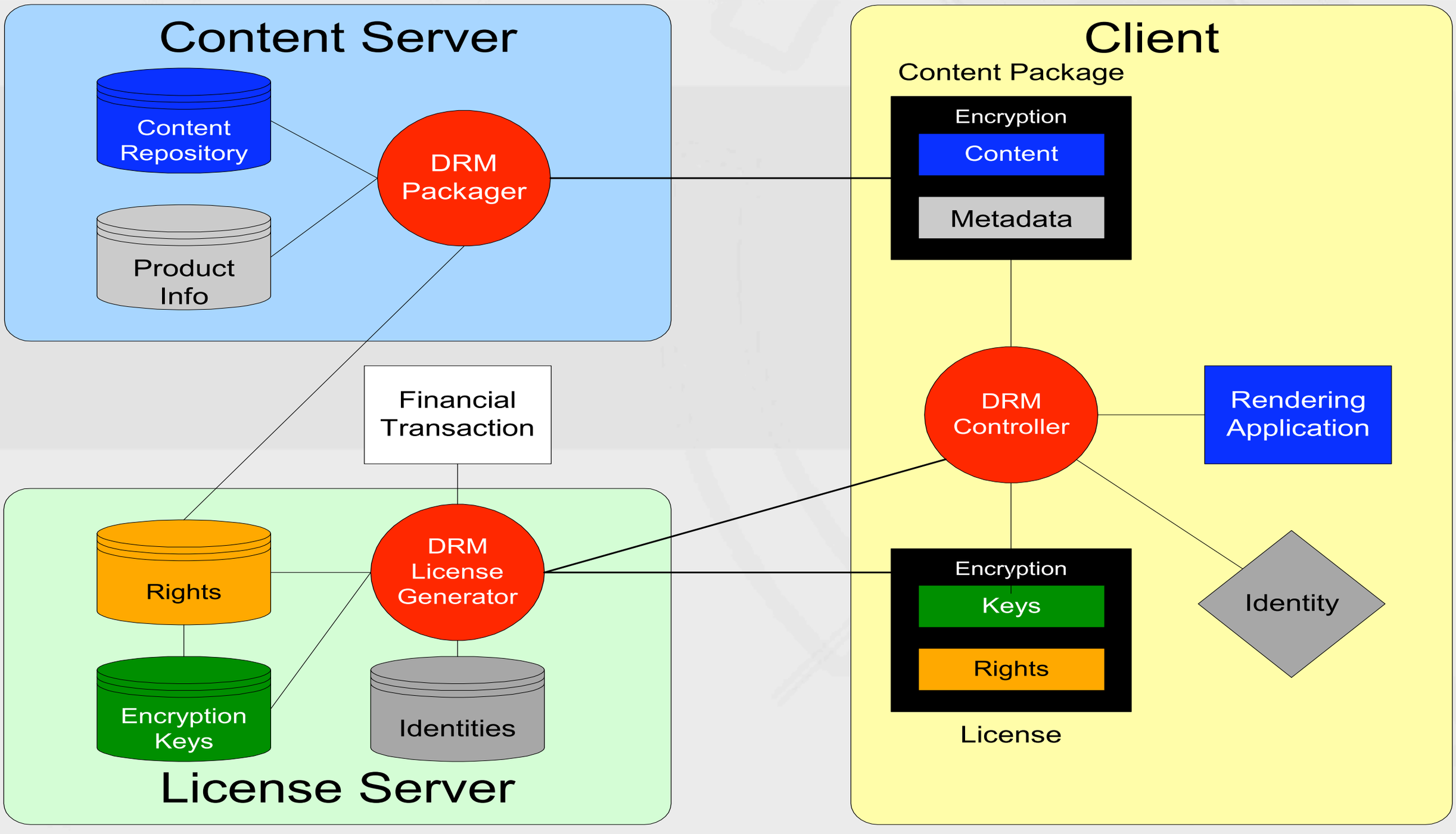
create/use

User

Content

DRM basic Model

DRM Reference Architecture



- 守护数字文档,数字版权管理:一个商业难题 [新华网 2006年7月7日]
  - 在国内某著名兵工厂的一次老总级别会议后,一份电子版的会议纪 要被秘密地发送到了几个有权限的重要人物手中,三个小时之后,这份文件将会自动销毁.
- 一个商业难题
- 新销售体系
- 待填补的市场

Thus, we see ...

- **DRM** can help ensure companies, corporations, and other entities who share similar business that:
  - Rights are tracked at ingestion
  - Access is controlled during production processes
  - Protection for the content extends throughout product life-cycles

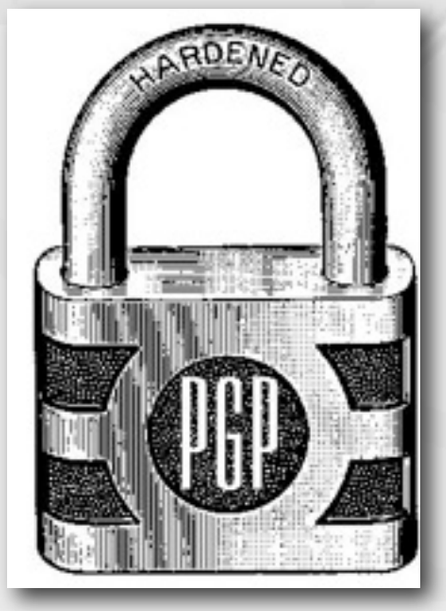
Thus, we see ...

- Additionally, **DRM** can integrate persistent content protection with content management to ensure:
  - Proper business practices
  - Implementation of new business models
  - Compliance with regulatory requirements in industries such as financial services, healthcare, and government

- 首批广播影视数字版权管理标准完成起草
  - http://news.cctv.com/china/20081108/105830.shtml
  - http://space.tv.cctv.com/video/VIDE1226188087000110

Previous Technologies

- PKI – Public Key Infrastructure
- PGP – Pretty Good Privacy
- S/MIME
- Access Control Systems
- Smart Cards
- Biometrics



How are these technologies different to DRM?

- Only protect the data in transit
  - E.g. over the Internet or on CD
- Once the data is opened, it can be:
  - edited
  - copied
  - printed
  - saved as an unprotected file

And then

- Redistributed to anyone else in an unprotected format.

Rely on TRUST once the content is delivered

Protecting Digital Intellectual Property

- Preventing Copying with Encryption
  - 加密
- Preventing Copying with Watermarking
  - 水印

Preventing Copying With Encryption (加密)

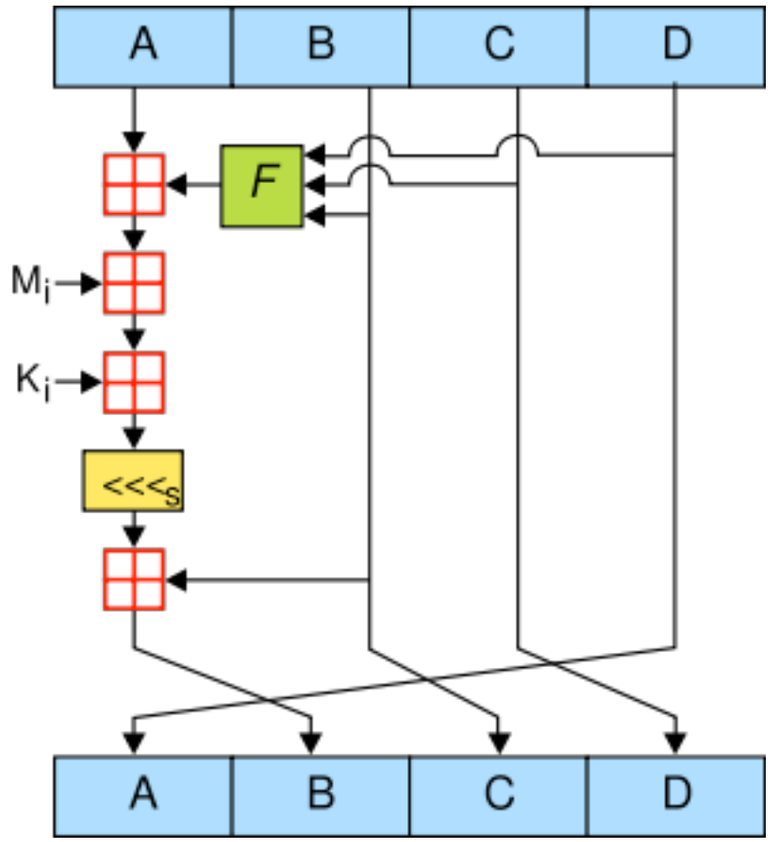
- Encryption is the scrambling of a message
  - Simple one is Caesar encryption
  - To decrypt (decode) message, you need one or more *Keys*
  - Also need an encryption *algorithm*, that specifies how to apply the key to the message to produce the scrambled message
- Symmetric key crypto: same key used for encrypt/decrypt
- Public key (we'll talk about the details later…):
  - Keys come in matched pairs: one encrypts, other decrypts
  - Given one key, you cannot deduce the other

Encryption

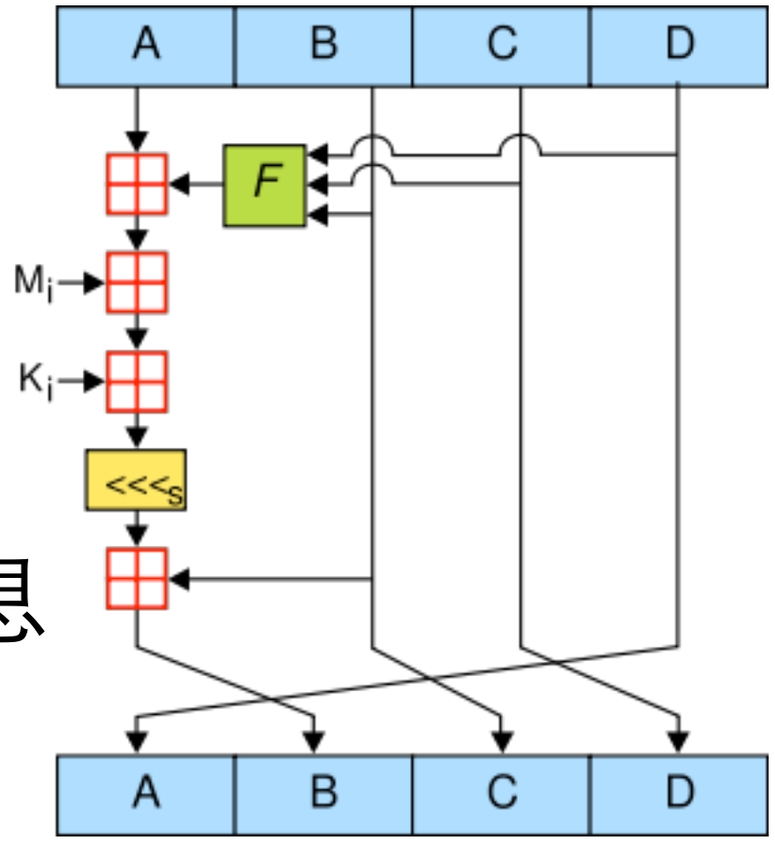
- RSA
- DES
- MD5

MD5
(Message Digest Algorithm version 5)

- MD5 is widely used in the open source world
- Enough for data sharing
- But not so safe



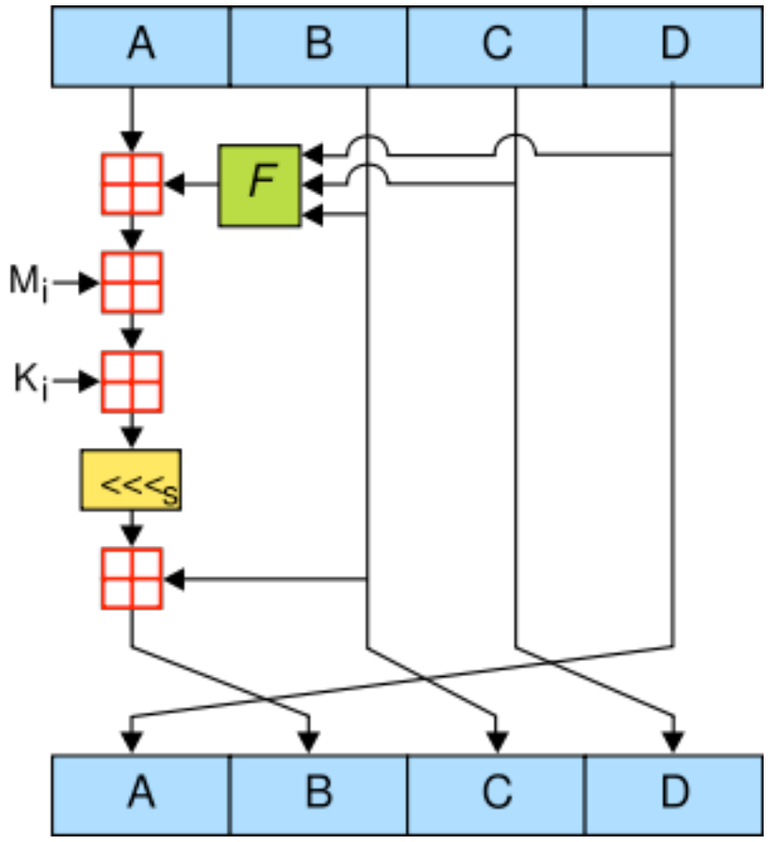
MD5 算法



- 输入：以512位分组来处理的信息
  - 每一分组又被划分为16个32位子分组
  - 对信息进行填充，使其位长对512求余的结果等于448
  - N*64+56个字节

- 输出：四个32位分组，构成128位散列
  - Hashing：散列

MD5算法



- Chaining Variable
  - A=0x01234567， B=0x89abcdef，C=0xfedcba98， D=0x76543210

- 循环运算
  - A到a，B到b，C到c，D到d
  - 主循环有四轮
    - 一轮进行16次操作
    - 每次操作对a、b、c和d中的其中三个作一次非线性函数运算

MD5 算法

- 基本函数
  - F(X,Y,Z) =(X&Y)|((~X)&Z)
  - G(X,Y,Z) =(X&Z)|(Y&(~Z))
  - H(X,Y,Z) =X^Y^Z
  - I(X,Y,Z)=Y^(X|(~Z))
  - & 表示“与”，|表示“或”，
    ~表示“非”，^表示“异或”

- 基本操作
  - FF(a, b, c, d, Mj, s, ti)
    a = b + ((a + F(b, c, d) + Mj + ti) << s)
  - GG(a, b, c, d, Mj, s, ti)
    a = b + ((a + G(b, c, d) + Mj + ti) << s)
  - HH(a, b, c, d, Mj, s, ti)
    a = b + ((a + H(b, c, d) + Mj + ti) << s)
  - II(a, b, c, d, Mj, s, ti)
    a = b + ((a + I(b, c, d) + Mj + ti) << s)
  - Mj表示消息的第j个子分组（从0到15）

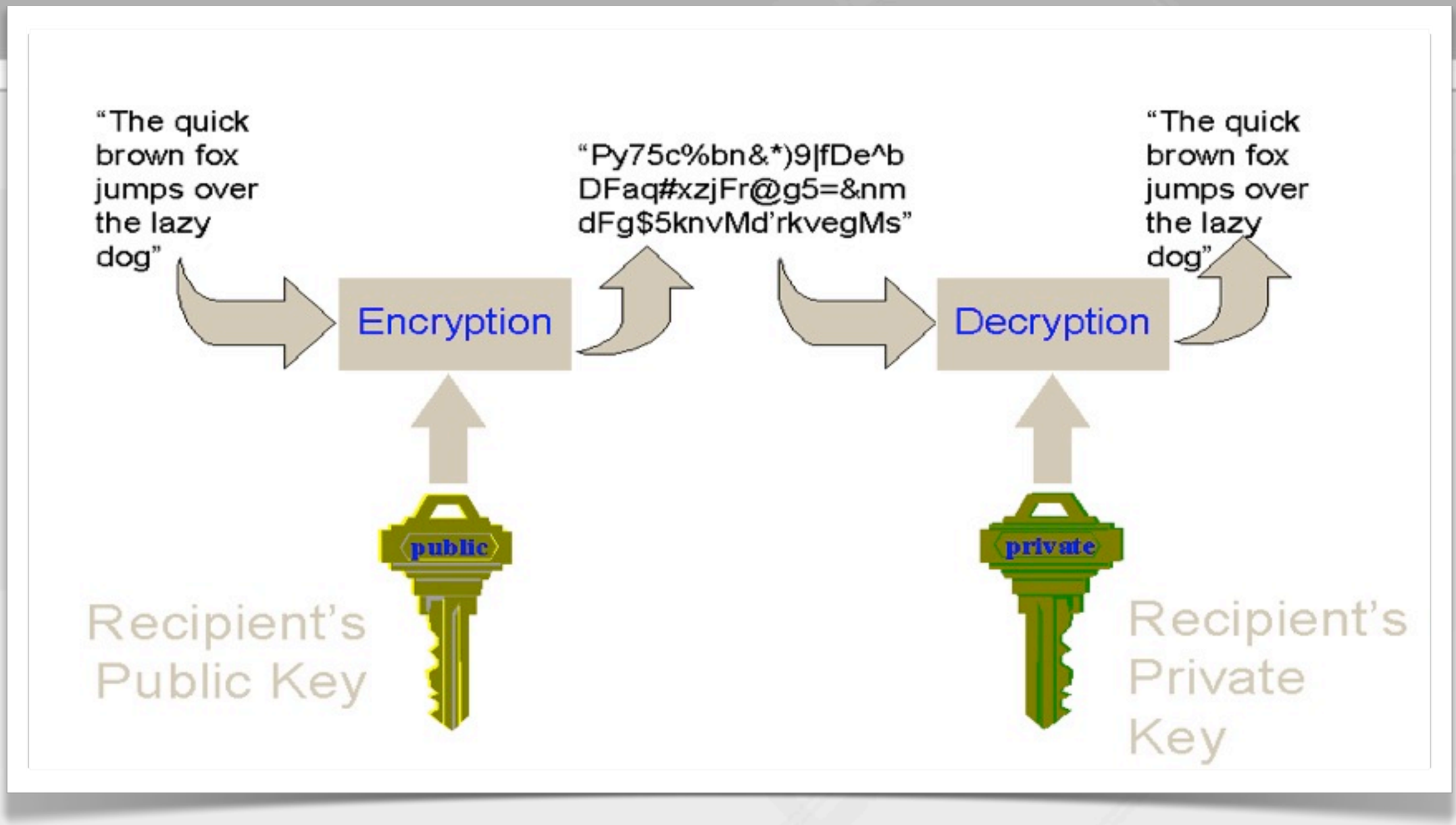
MD5算法

- 在第i步中，ti是4294967296*abs(sin(i))的整数部分，i的单位是弧度。
- 完成上述64步操作之后，将A、B、C、D分别加上a、b、c、d。然后用下一分组数据继续运行算法
- 最后的输出是A、B、C和D的级联。
- 例：（可试用python中的md5实现：hashlib）
  - MD5 ("") = d41d8cd98f00b204e9800998ecf8427e
  - MD5 ("abc") = 900150983cd24fb0d6963f7d28e17f72

开源密码体系的崩溃

- 山东大学的王小云教授
  - [Crypto 2004], 利用hash碰撞原理，攻破MD5、HAVAL-128、 MD4和RIPEMD算法
  - 2005年8月, 给出攻击SHA－1的算法

Basic Idea of Cryptography



Think of encryption key as sealing an envelope, and decryption key as unsealing it.

How do you “break” encryption?

- Usual assumptions of cryptography…
  - Adversary knows details of algorithm (not in WWII!)
  - Adversary may know something about nature of messages (why would this help?)
  - Adversary *doesn’t know decryption key(s)*
- Hard: exploit mathematical weakness in the algorithm
- Hard: guess key by (educated) trial and error
- Usually easier: attack some weaker part of the system
  - Usually, trick system into revealing a key
  - Chain is only as strong as weakest link!

DVD Content Scrambling System (CSS)

- To each **licensed DVD player** corresponds a **decryption key**:
  - P1, P2, …, Pn
- Each disc is encrypted under its own key, call it D
  - n copies of D are stored on the disc; each copy encrypted with one player's P
  - Player finds a D that it can decrypt, then uses D to play disc

- DVD player is a trusted client
  - It's not supposed to ever reveal any D, or its own P
  - What happens if either of these occur?
  - Why can't you convert DVD to another format?
  - Why can't you make direct copies of a DVD onto another disc (copying the D keys along with the content?)

Early DeCSS timeline...

- Sep ’99, DeCSS released as open-source Linux DVD player
- Dec ’99, DVDCCA sues 500 individuals in California for hosting DeCSS, alleging trade-secret violations
- Jan ’00, MPAA sues 2600.com in New York under DMCA’s copyright protection circumvention laws
- Jan ’00, DVD Source Code Distribution Contest
- Jan ’00 Jon Johansen arrested in Norway, later released
- Aug 00 MPAA wins DMCA suit in NYC

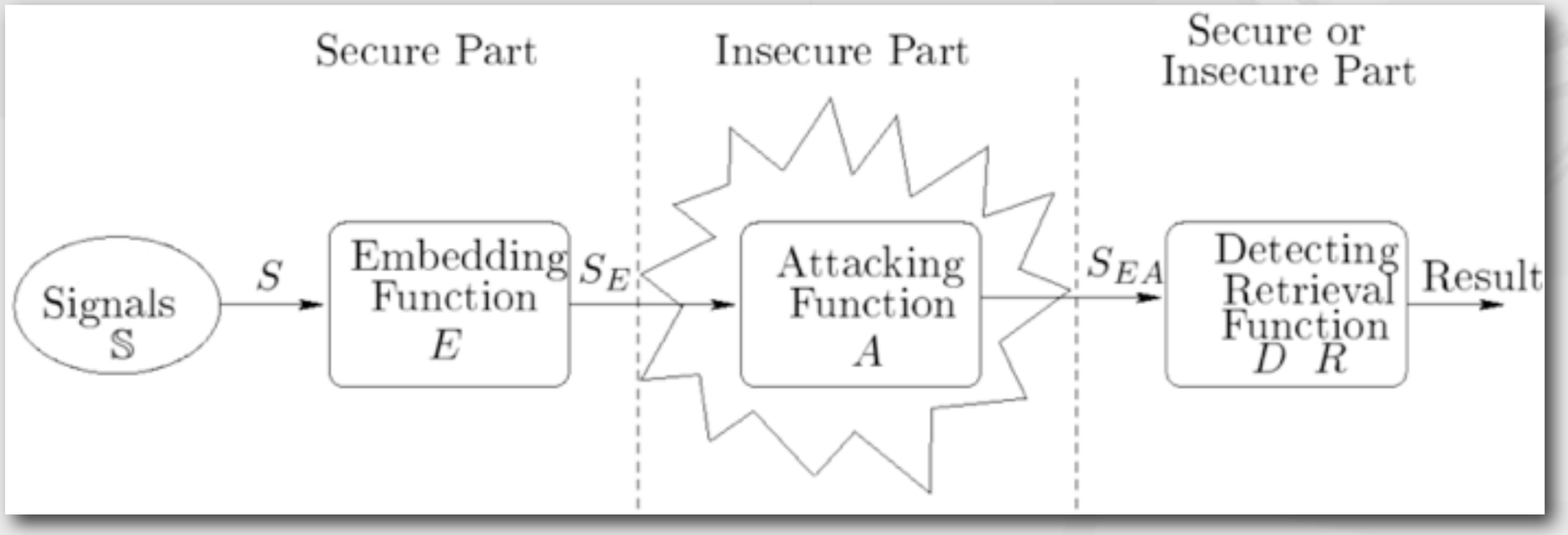
How Was CSS cracked?

- Idea =>
- P must appear somewhere in the decryption code of a trusted player
  - Hardware players difficult to reverse-engineer/probe
  - Software players maybe easier? …turns out yes!
  - Later analysis revealed weaknesses in CSS…it probably could have been broken *without* first recovering a key

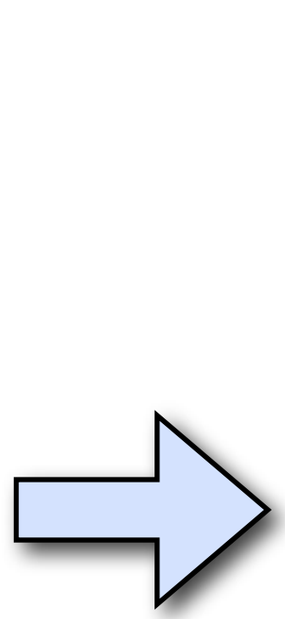
- Original goal of CSS: even if one P is compromised, others are still sound
- Flaw: weakness in the algorithm allowed *all* P's to be compromised once a single P was found
  - Why wasn't this flaw discovered *before* the algorithm went into production players?

Preventing Copying With Watermarking (水印)

- digital art
- 票据防伪
- 数据隐藏
- 隐蔽通讯



Stenography



1.removing all but the last 2 bits of each color component

2.X 85

Digital Watermark

- Invisible ink on multimedia data
  - image
  - video
  - music
  - graphics

Digital Watermark

- Image

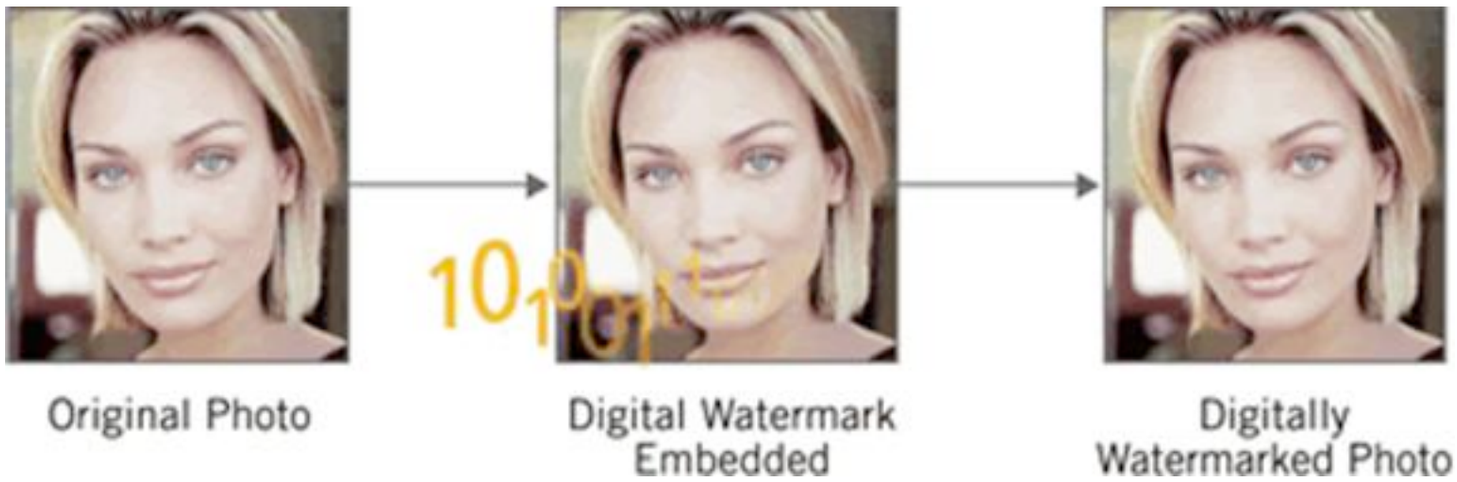
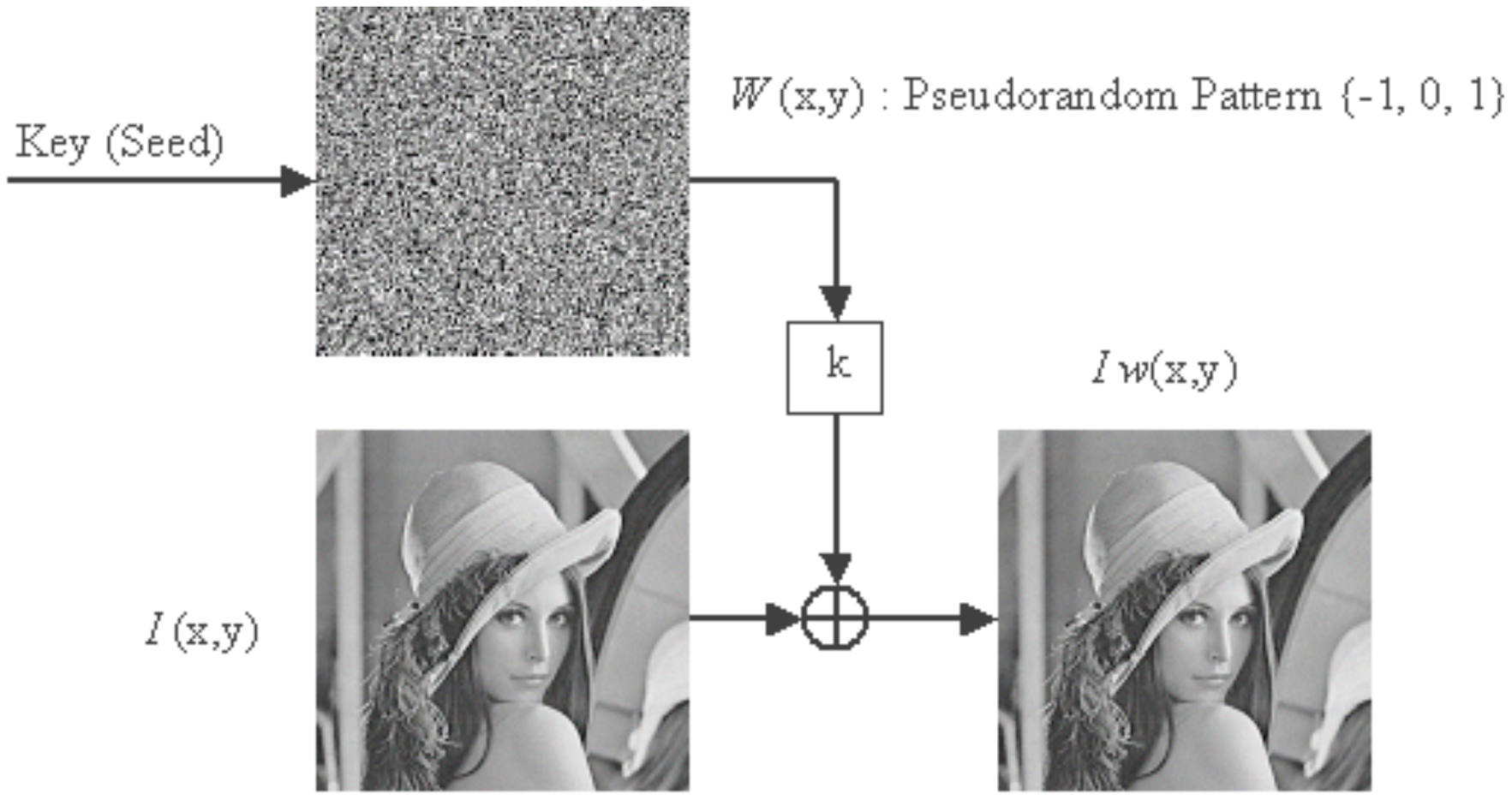


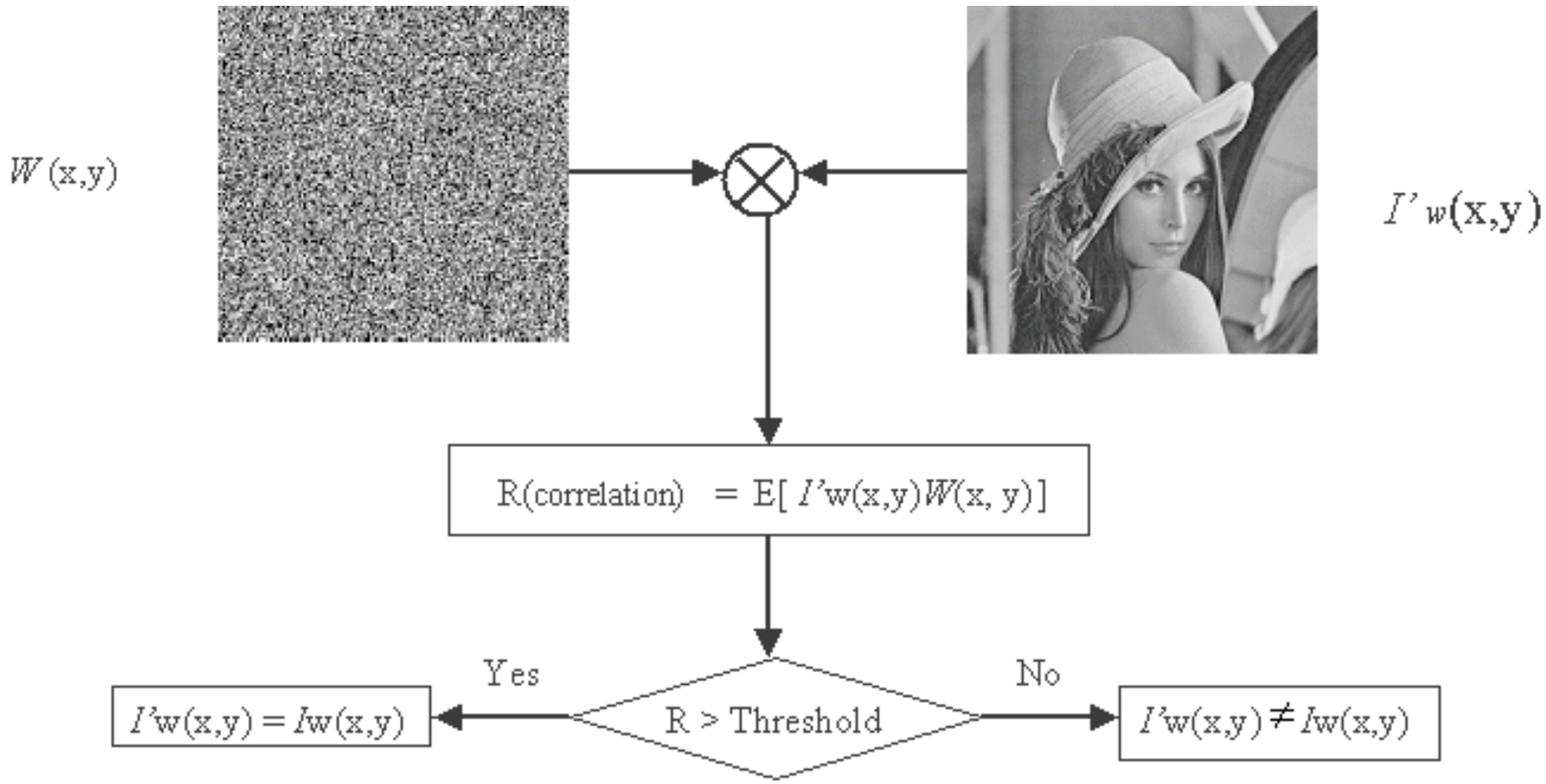
Image watermarking



$$I_w(x,y) = I(x,y) + k * W(x,y)$$

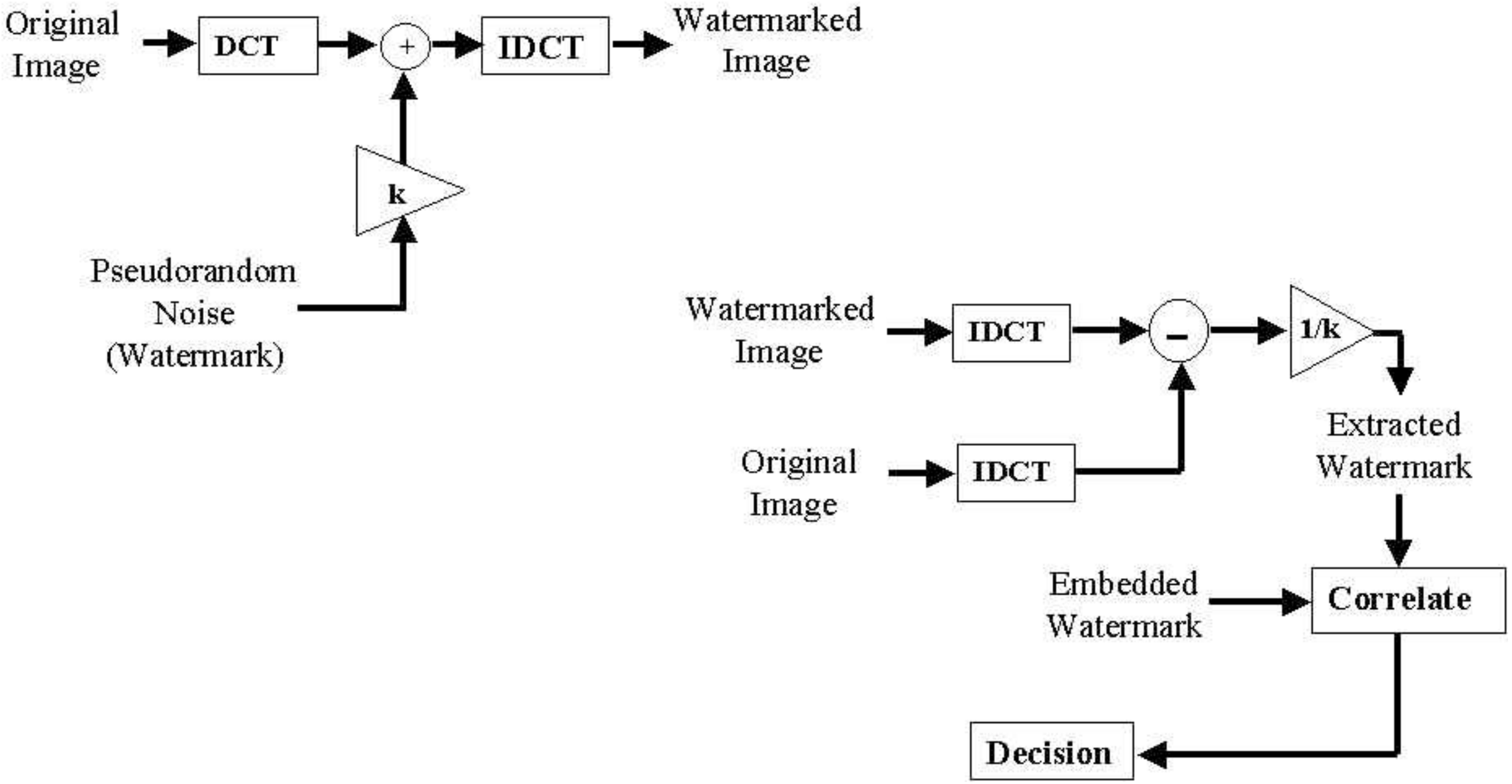
Embedding

Image watermarking



Detecting

DCT based algorithm



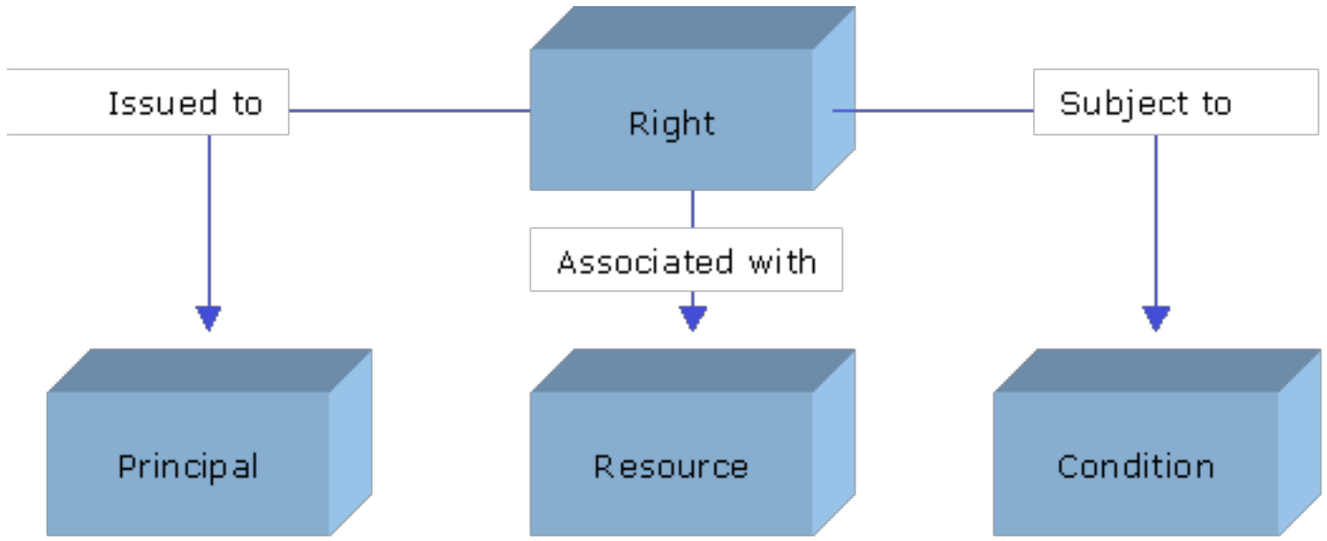
REF: http://scien.stanford.edu/pages/labsite/2001/ee368/projects2001/dropbox/project06/

Digital Watermark

- Music: mp3stego
  http://www.petitcolas.net/fabien/steganography/mp3stego/index.html

Digital Rights Management – Rights Expression Language (REL)

Metadata for DRM



- MPEG-21: REL
  - A digital item
    - is a structured digital object with a standard representation, identification and metadata
  - User
    - is any entity that interacts in the MPEG-21 environment or makes use of digital items

Rights model

- Render rights
  - View, Print, Play or Execute
- Transport rights
  - Copy, Move, Loan
- Derivative work rights
  - Edit, Embed, Extract
- Utility rights
  - Backup, Caching, Data integrity

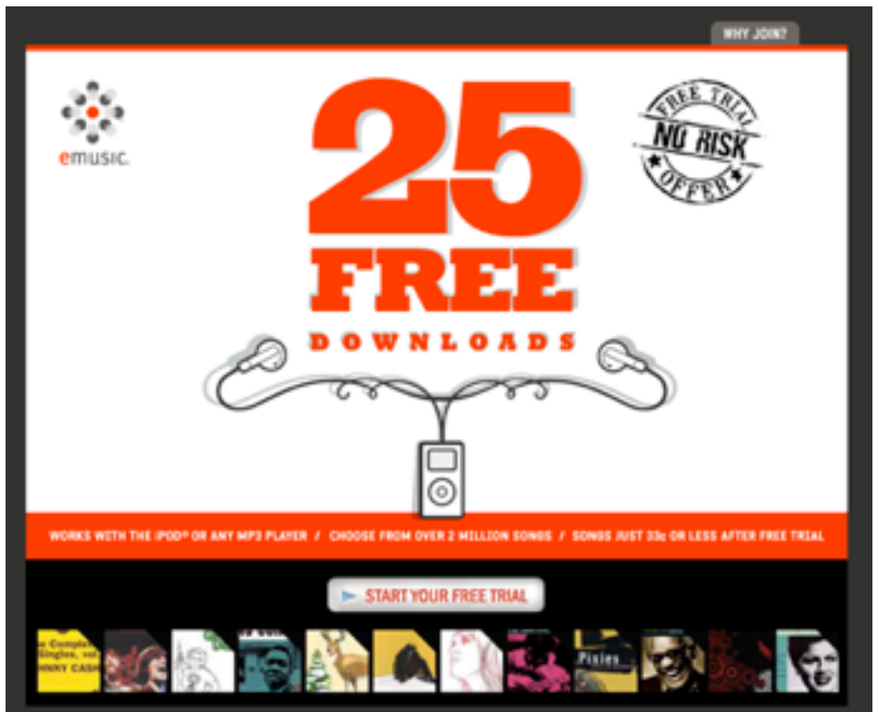
DRM technologies and associated devices

Name	Used in	Date to use	Description
Fairplay	ipod, iphone, itunes	2003+	The purchased music files are encoded as AAC, then encrypted with an additional format that renders the file exclusively compatible with iTunes and the iPod
3-play	Microsoft Zune	2006+	Music files that are received wirelessly from other Zune devices can be played only a maximum of three times on the device.
Janus WMA DRM	All PlaysForSure Devices	2004+	Janus is the codename for portable version of Windows Media DRM for portable devices.
OMA DRM	Implemented in over 550 phone models	2004+	A DRM system invented by the Open Mobile Alliance to control copying of cell phone ring tones

DRM opposition

digital rights management
= digital restrictions management ?

DRM-free



- Apple began selling "DRM-Free" music through their iTunes store in April of 2007
- the DRM-Free iTunes files were still embedded with each user's account information

Digital Rights Expression Languages

- Rights may be managed using digital rights expression languages.
- DRELs specify the permissions given to
  - users, distributors and repositories
  - and the conditions and obligations that have to be satisfied for these permissions to be exercised.

Rights Expression Language (REL)

- A standard way to express and interpret rights specification for interoperability.
- Comprehensive, generic, precise and extensible.

- eXtensible rights Markup Language (**XrML**).
  - XrML 2.0 : MPEG REL
- Open Digital Rights Language (**ODRL**).
  - ODRL 1.1 : OMA (Open Mobile Alliance) REL

General description of RELs

- A rights expression language (REL) is a type of policy authorization language.
  - Focus is on expressing rights granted by one party to another.
  - Issuance and delegation rights for other grants are core concepts.
    - Can be used to model lending, loans, transfers of rights.

- REL design goals:
  - Provide a flexible, extensible mechanism for expressing authorizations.
  - Enable interoperability across various policy evaluation systems.
  - Make it easy for policy authors (e.g. content owners) to express their desired policies.

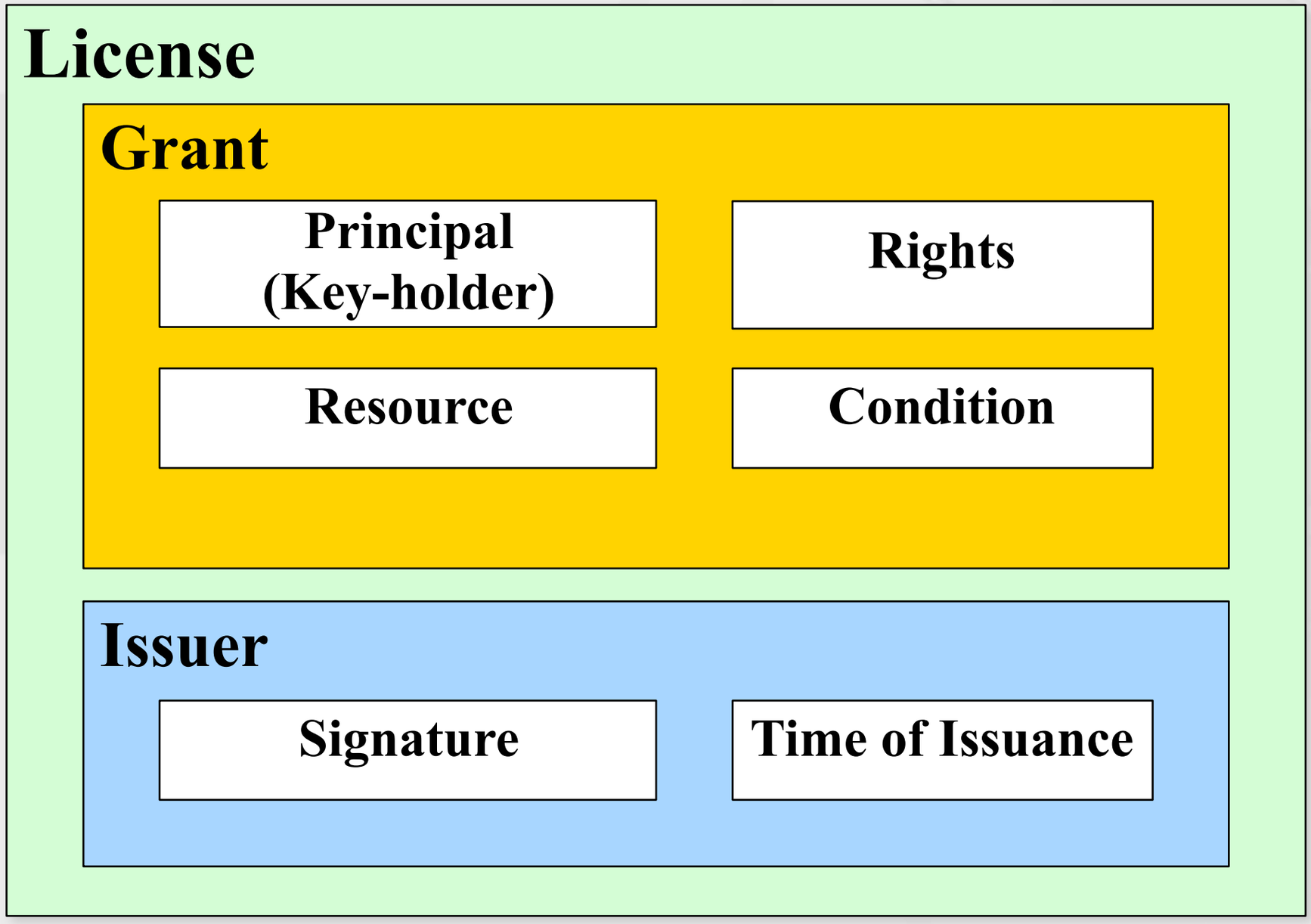
An example REL: XrML 2.X

- XrML, the *XML Rights Management Language*, is a standard currently under development

XrML introduction

- The only REL in working DRM systems.
- Specification language:
  - Programmers specify high-level rights in a license file.
  - An XrML interpreter parses the license file.
  - REL SDK for building an XrML interpreter.
- Data model:
  - License, grant, principal, right, resource and condition

XrML license



XrML 2.X

- In the RM context, XrML 2.X allows content owners a systematic way to express their intent for distribution and consumption.
- Like other policy languages, XrML 2.X **licenses** (statements) declare authorizations, but cannot enforce compliance.
  - Systems that consume XrML 2.X licenses must be trusted by the license issuer to properly enforce the grants specified within the license.
- Licenses are digitally signed by the issuer to protect their integrity.
- Licenses may be embedded within content or move independently.

Semantic of a Grant

- Every XrML 2.X grant has the following form:
  - Issuer authorizes principal to exercise a right with respect to a resource subject to conditions.
  - A license is a collection of one or more grants made by the same issuer.

- Grants may be **chained** together:
  - Bill's RM system trusts Tom and his delegates.
  - Tom delegates the right to license printing to John.
  - John issues a license: "Bill has the right to print the book."
  - Therefore Bill can print the book.

Sample XrML 2.X License

```
<?xml version="1.0" encoding="UTF-8" ?>
 <license>
 <grant>
  <keyHolder>  …  </keyHolder>
  <mx:play />
  <mx:diReference>
  <mx:identifier>
       urn:mpeg:example:2002:twotonshoe:album
   </mx:identifier>
   </mx:diReference>
 </grant>
 <issuer> … </issuer>
</license>
```

浙江大学计算机学院
数字媒体与网络技术

XrML authorization model

- Input
  - Principal
  - Right
  - Resource
  - Time interval
  - Licenses
  - Designated “root grants” (implicitly trusted)

- Output
  - “No”
  - “Yes,” unconditionally
  - “Maybe,” if a set of conditions are also met

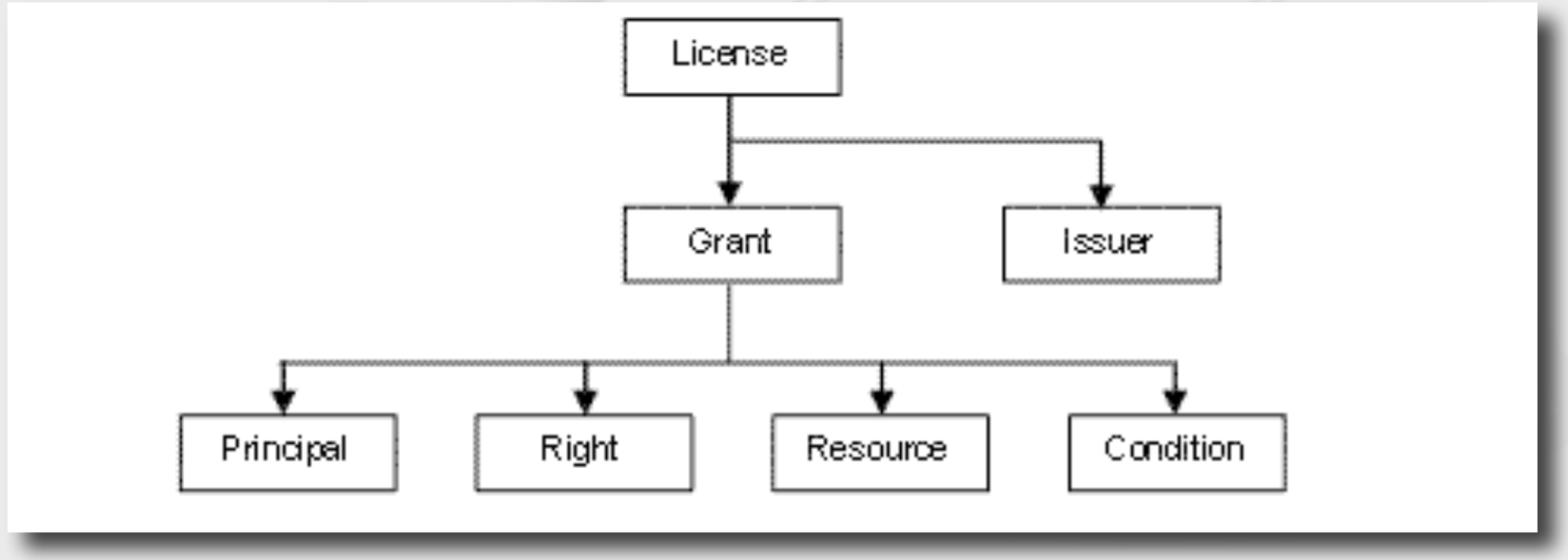
XrML Key Language Features

- Mechanisms for enhanced expressivity
  - Patterns, variables and quantifiers
  - Grouping grants
  - Delegation
- Meta-rights
  - Issue
  - Obtain
  - Revocation
  - PossessProperty
- Linking conditions
  - PrerequisiteRight

MPEG-21 REL

- Derived from XrML
- 3 Components:
  - Kernel set
  - Standard extension
  - multimedia extension

structure of a simple license



Agenda

- Overview
- Introduction of DRM (Sony & DRM)
- Protecting Digital Intellectual Property
- Rights Expression Language (REL)
- Case Study – Existing DRM systems

浙江大学计算机学院
数字媒体与网络技术

Case Studies

InterTrust

- Original DRM vendor (with IBM)
  - May have coined the term
  - Originally called Electronic Publishing Resources
  - First implementations in hardware
  - Major patent portfolio
- New technology: Rights|System
  - Framework for multiple devices
    - Rights|Desktop for PCs
    - Rights|TV for settop boxes
    - Rights|PDA for handheld devices
    - Rights|Phone for Symbian mobile phones
  - Public encryption algorithms

IBM EMMS

- Developed in IBM labs over period of 8 years
- Cross-device, like InterTrust
- Integration with IBM server components
  - WebSphere
  - DB2
  - Service Provider Delivery Environment (SPDE)

浙江大学计算机学院
数字媒体与网络技术

Microsoft

- 1st generation: Windows Media Player
- 2nd generation: Digital Asset Server
  - Server for Microsoft Reader E-Books
  - Uses subset of XrML
- 3rd generation: “Unified DRM” (RMS)
  - One DRM for all devices & platforms
  - Open API for rendering app developers
  - XrML based

MacroVision (1985-)

- Copy protection technique for VHS tapes
- Inserts special signals into the vertical blanking interval of NTSC protocol
  - affects automatic gain control in most VCRs, but is ignored by most televisions
  - difficult to remove from the original signal
- Makes subsequent recordings shake and have periods of bright and dark frames

Apple's FairPlay Technology

- DRM for iTunes
  - playing, recording, and sharing of files
- Moves beyond "protection only"
  - allows media to be shared among devices
  - allows others to listen to (but not copy) music
  - allows music to be burned to an audio CD, which loses the DRM protection

How FairPlay Works

- iTunes uses encrypted MP4 audio files
- Acquire decryption key by trying to play song
  - player generates a unique ID
  - sends this ID to the iTunes server
  - if there are less than N authorizations in your account, the server responds with decryption key
- The decryption key itself is encrypted so cannot be given to another machine

Discussion

- Is FairPlay too lenient, too stringent, or just about right?
- What is your experience with this DRM?
- What happens if Apple decides to stop supporting FairPlay?