浙江大学计算机学院
数字媒体与网络技术

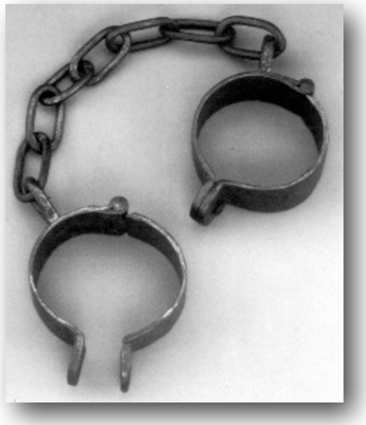
Digital Asset Management
数字媒体资源管理

4. Digital Rights Management

任课老师：张宏鑫
2020-10-22

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

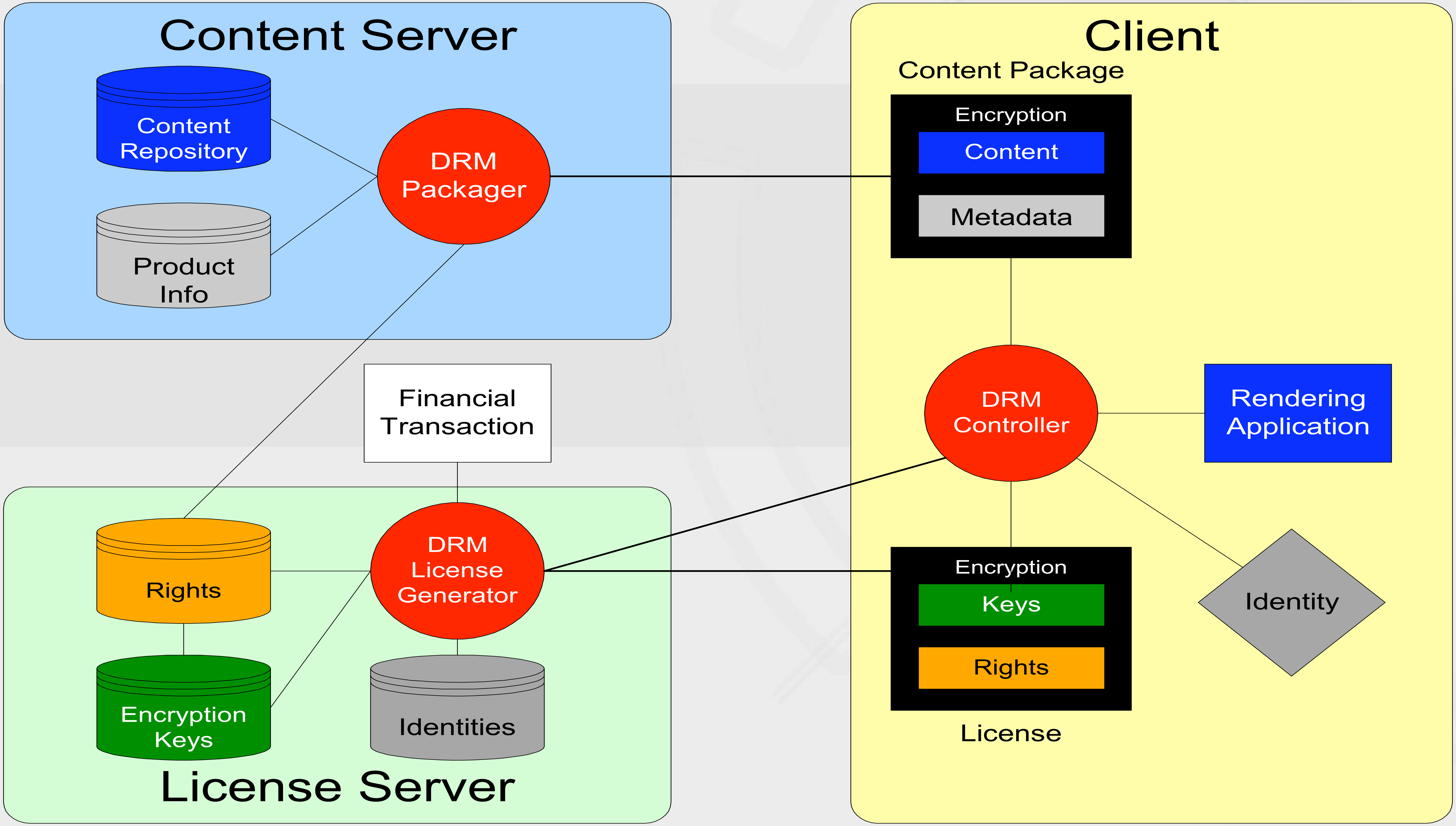
create/use

User

Content

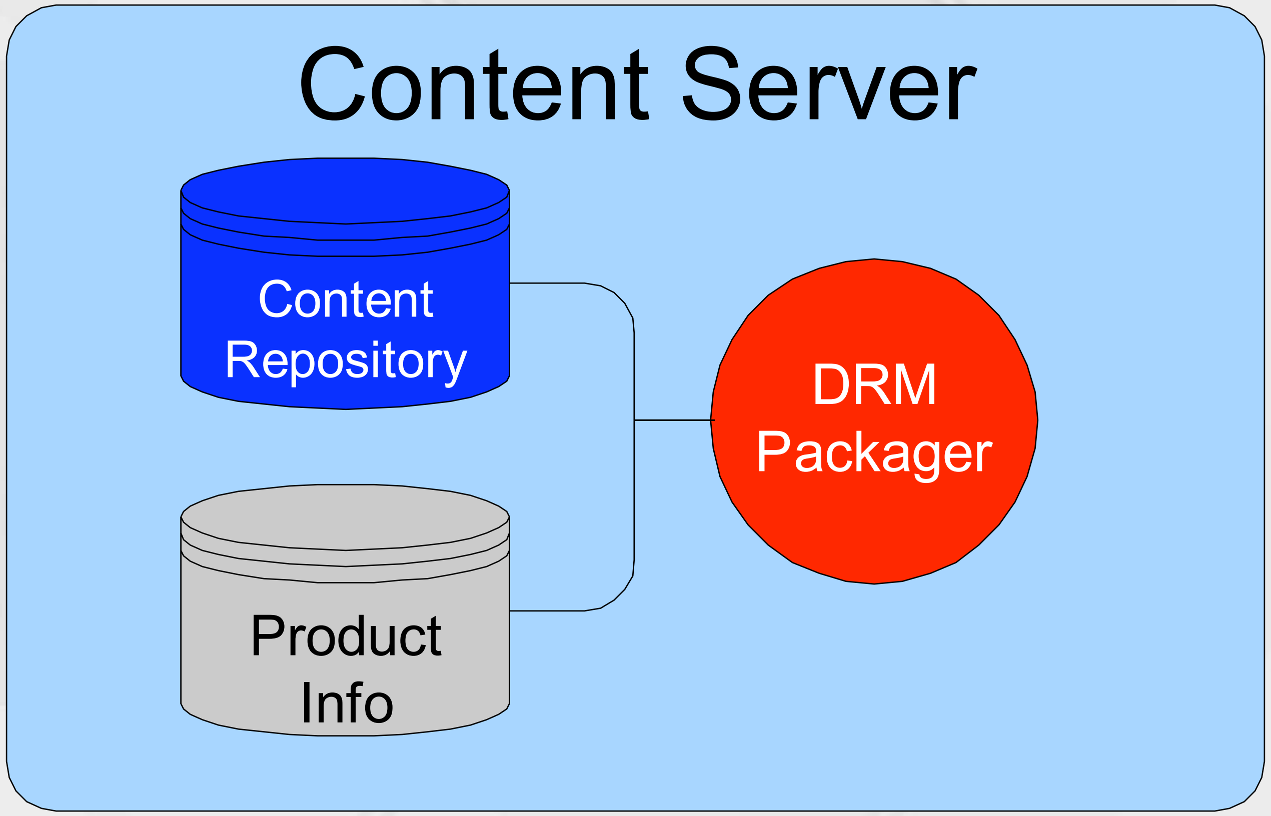
DRM basic Model

DRM Reference Architecture



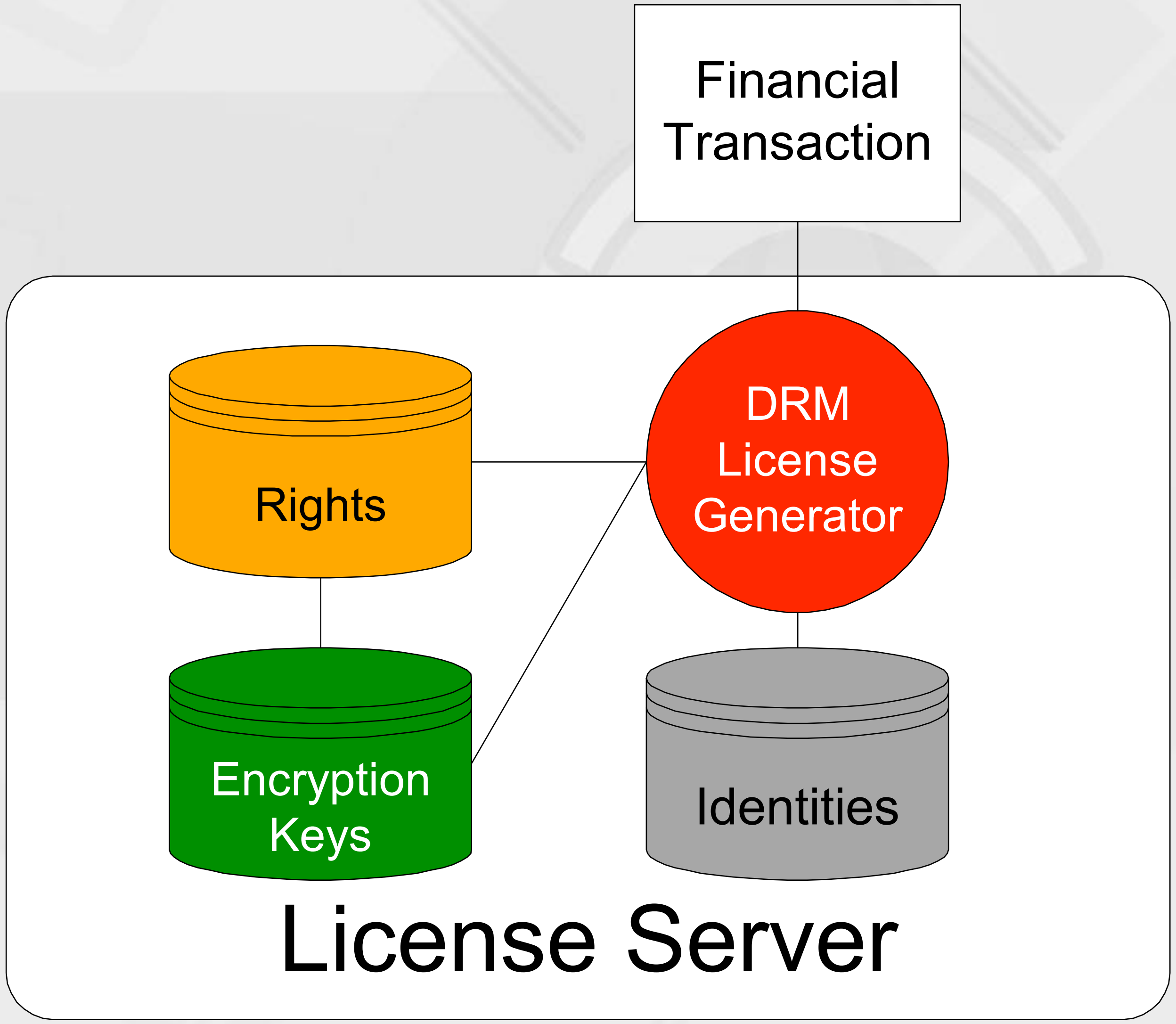
Content Server

- Content Repository
  - –Content Management system
  - –Digital Asset Management system
  - –File server
- Product Info
  - **–Rights**
  - –Product metadata
- DRM Packager
  - –Packages content with metadata
  - **–Encrypts**



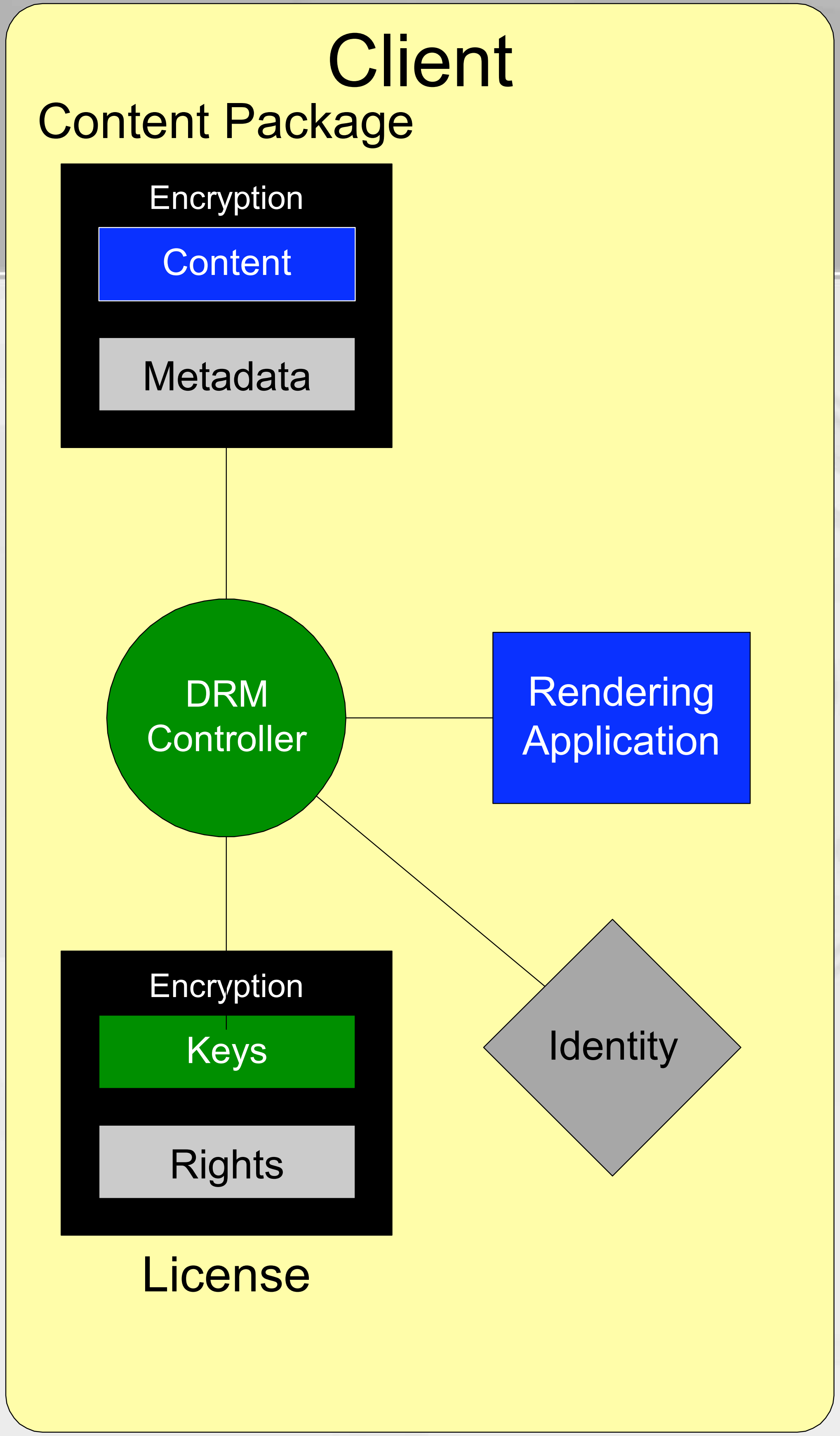
License Server

- Encryption key repository
- User identity database
  - –Usernames
  - –Machine IDs
- DRM License Generator



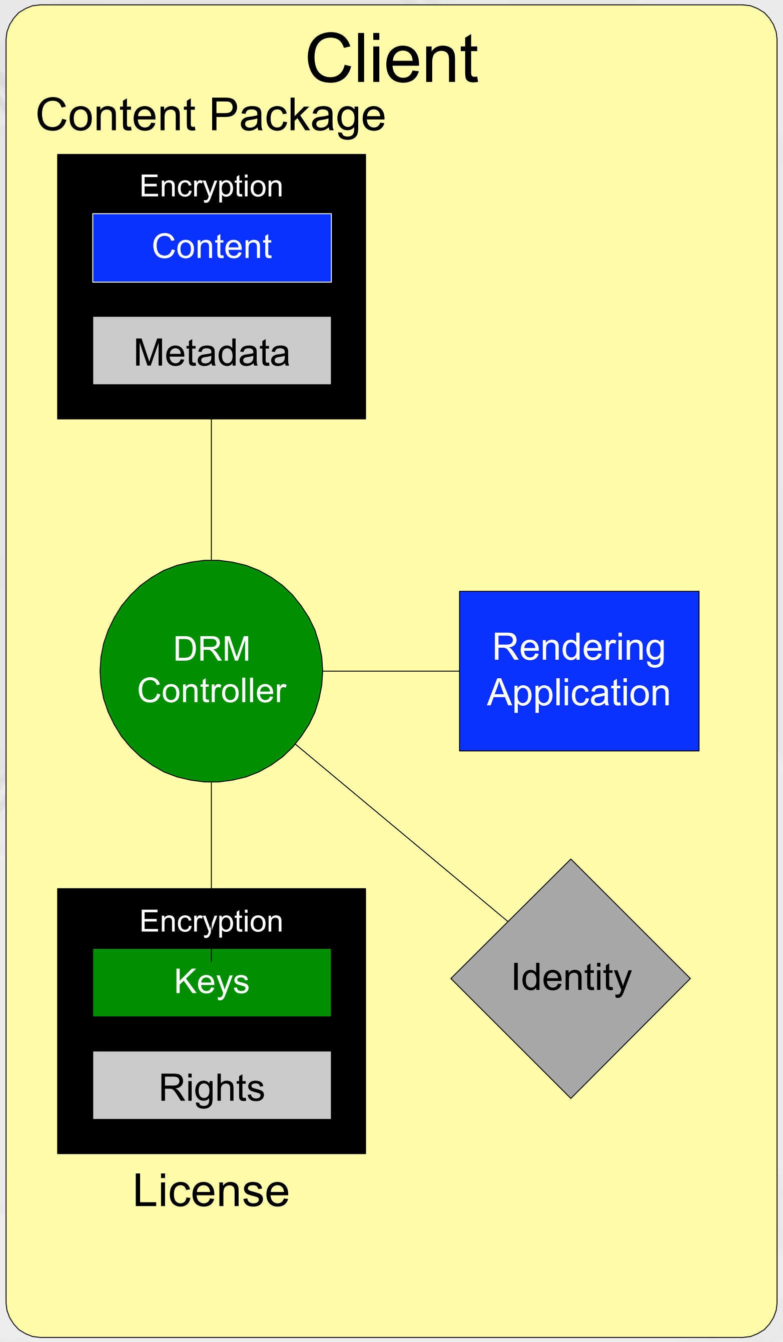
Client

- DRM Controller
  - –Nerve center of process
- Rendering application
- Content packages
- Licenses
- Identity



Processes - User Initiation

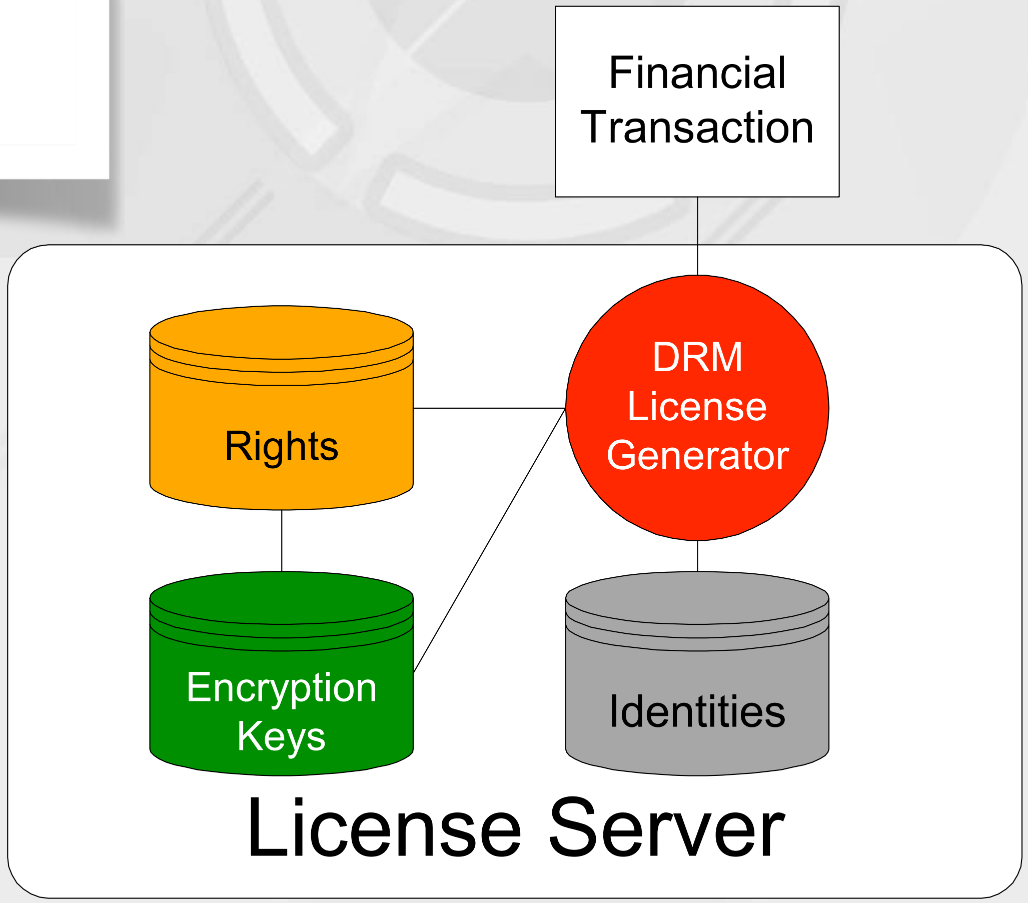
- User obtains content package
- User requests operation
  - view, play
- DRM controller collects info
  - Content
  - Identity
  - Requested rights
- DRM controller:
  - license generator



Processes - License Generation

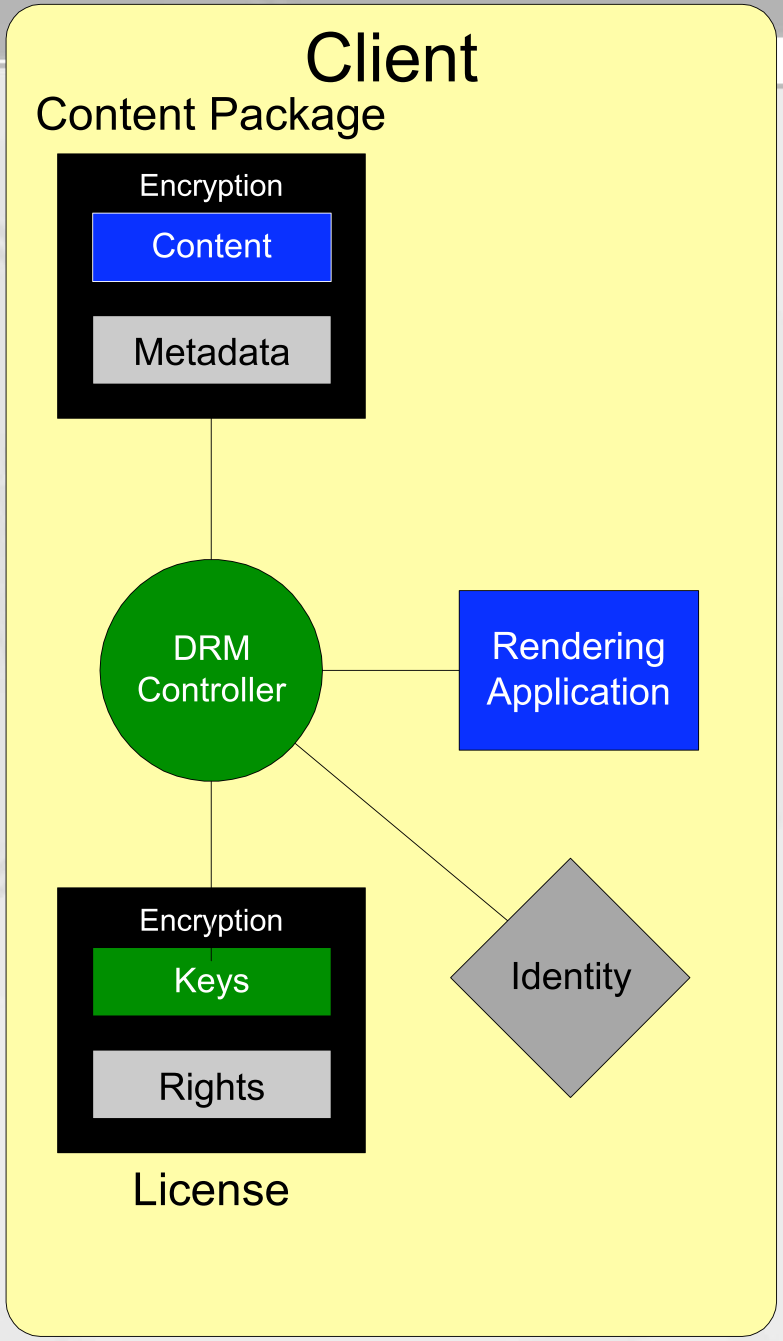
DRM License Generator…

- –Checks content & identity
- –Obtains keys from key repository
- –Creates & sends license to client
- –Generates financial transaction, where necessary



Processes - User Completion

- DRM Controller…
  - Receives license
  - Extracts keys from license
  - Decrypts content
  - Generates financial transaction, where necessary
  - Hands content to rendering application
- Rendering application plays content



- 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

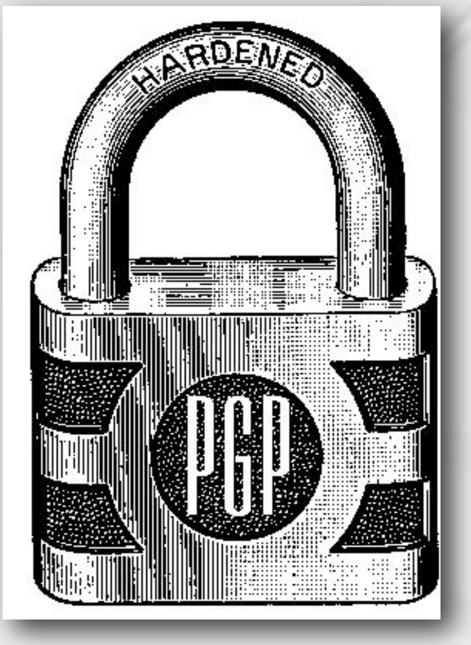
Thus, we see ...

- 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

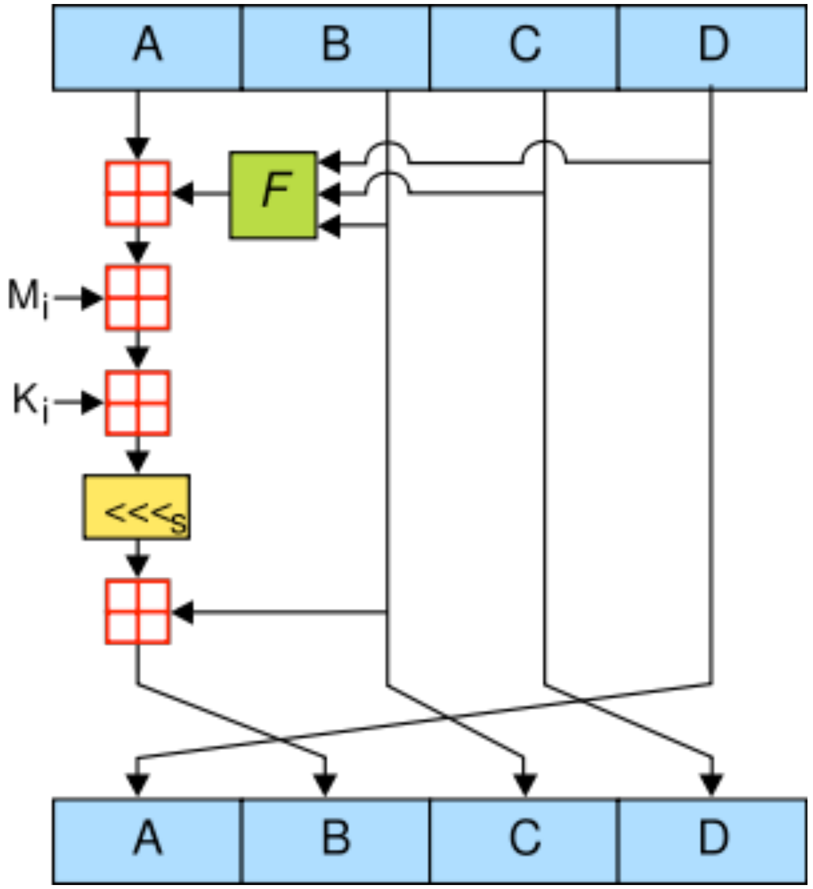
Encryption

- MD5：数据指纹
- DES：对称加密
- RSA/ECC：非对称加密

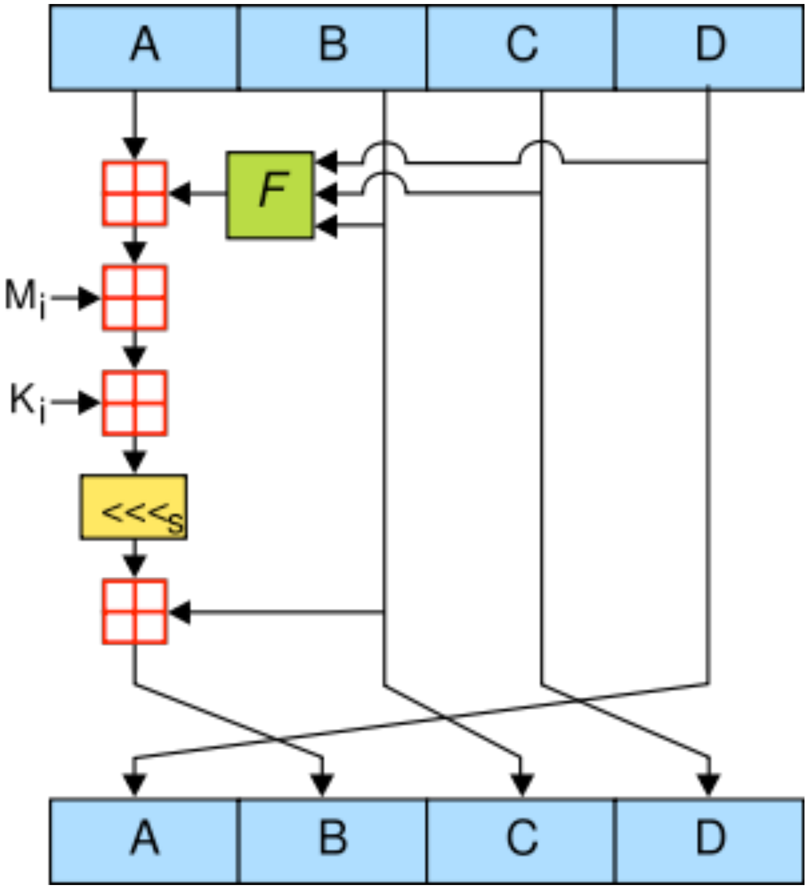
MD5

(Message Digest Algorithm version 5)

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

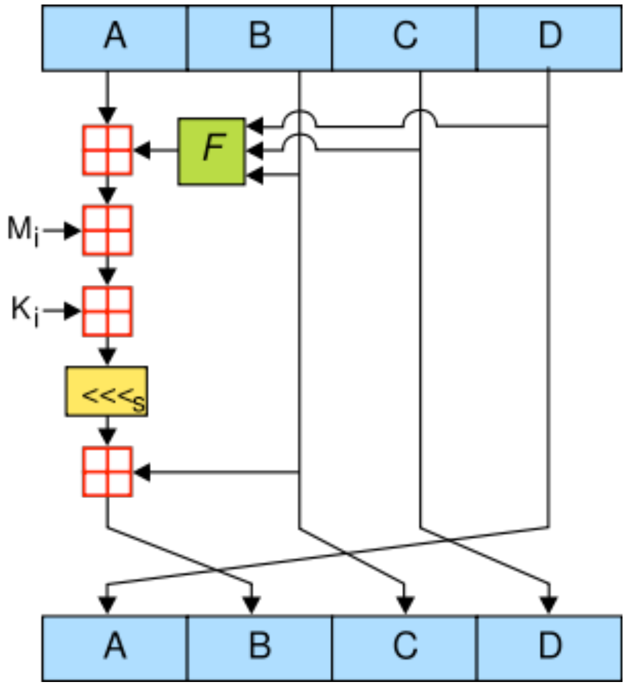


MD5 算法



- 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）

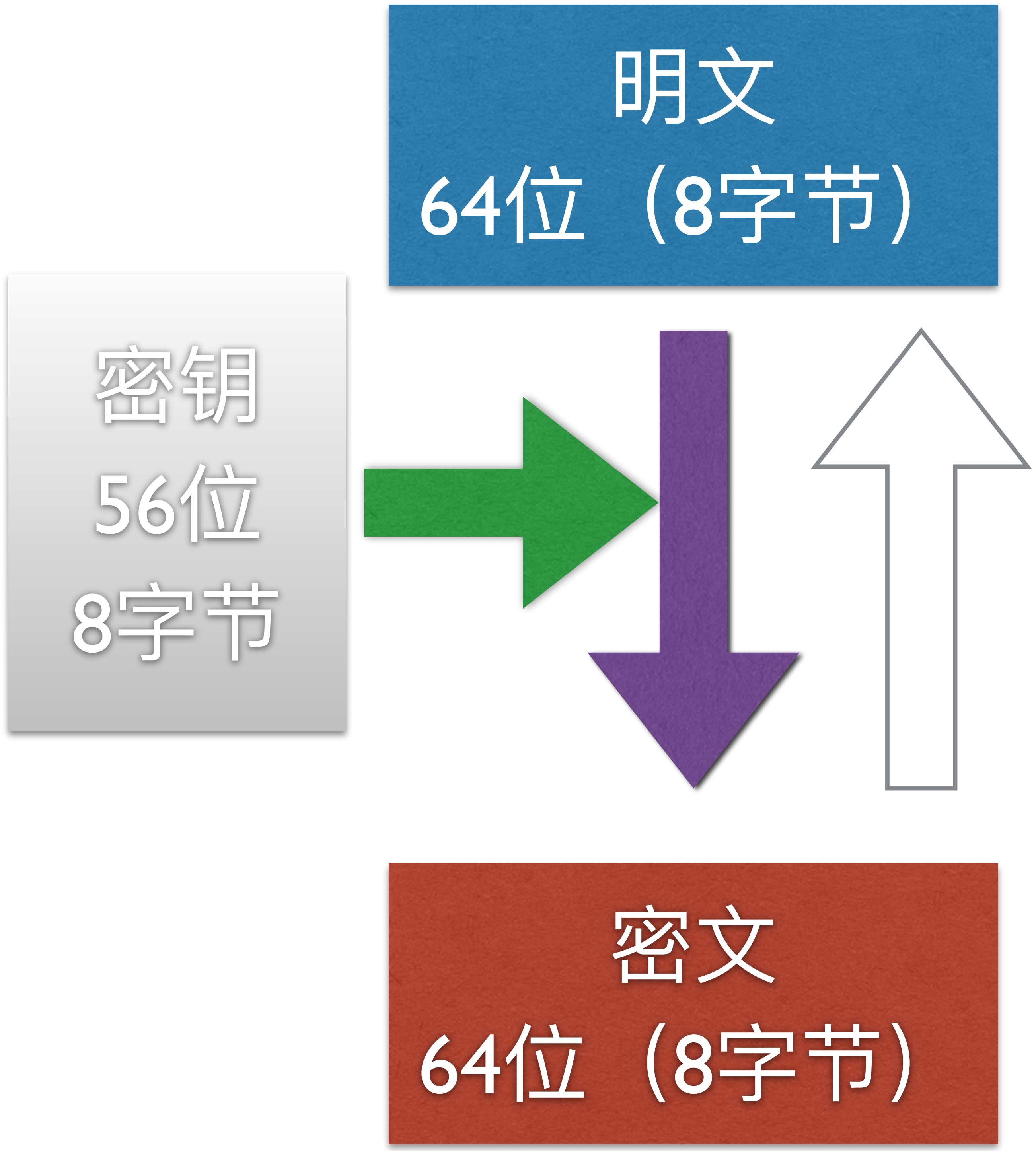
MD5算法

- 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

加密体系开始崩溃

- 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

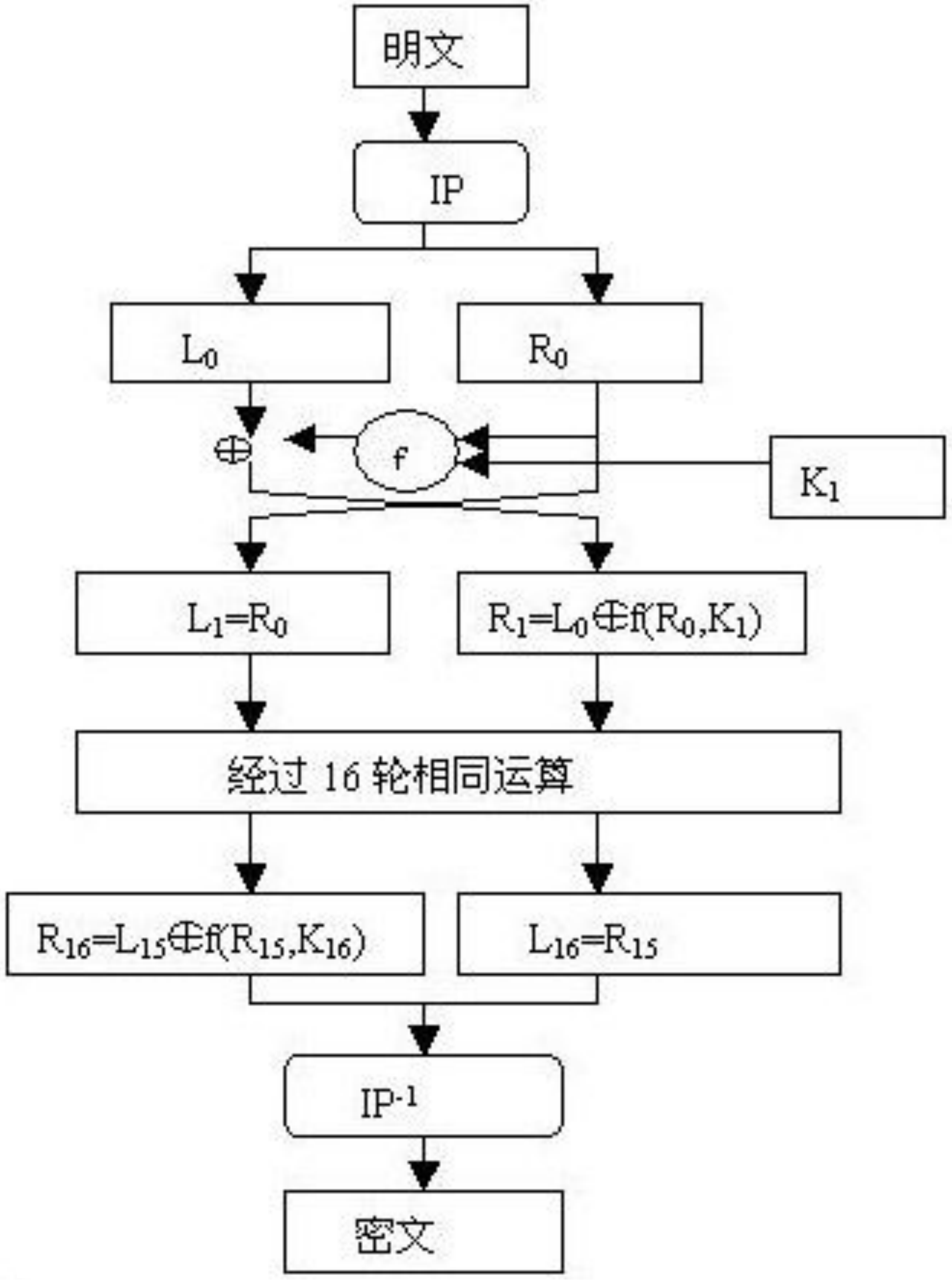
对称加密： DES /



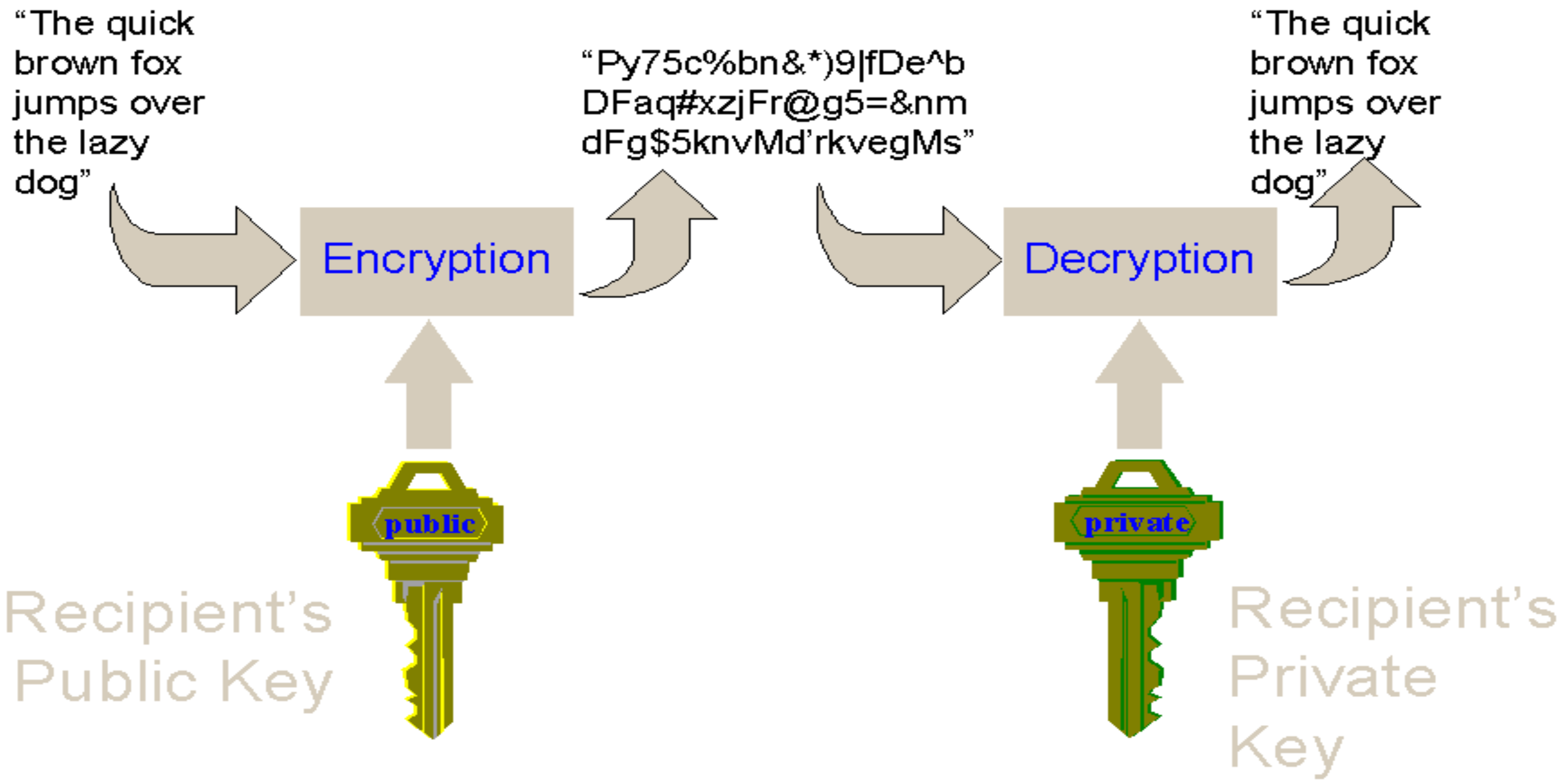
明文
64位（8字节）

密钥
56位
8字节

密文
64位（8字节）



Basic Idea of Cryptography （非对称加密）



“The quick brown fox jumps over the lazy dog”

Encryption

“Py75c%bn&*)9|fDe^b DFaq#xzjFr@g5=&nm dFg$5knvMd’rkvegMs”

Decryption

“The quick brown fox jumps over the lazy dog”

Recipient’s Public Key

Recipient’s Private Key

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

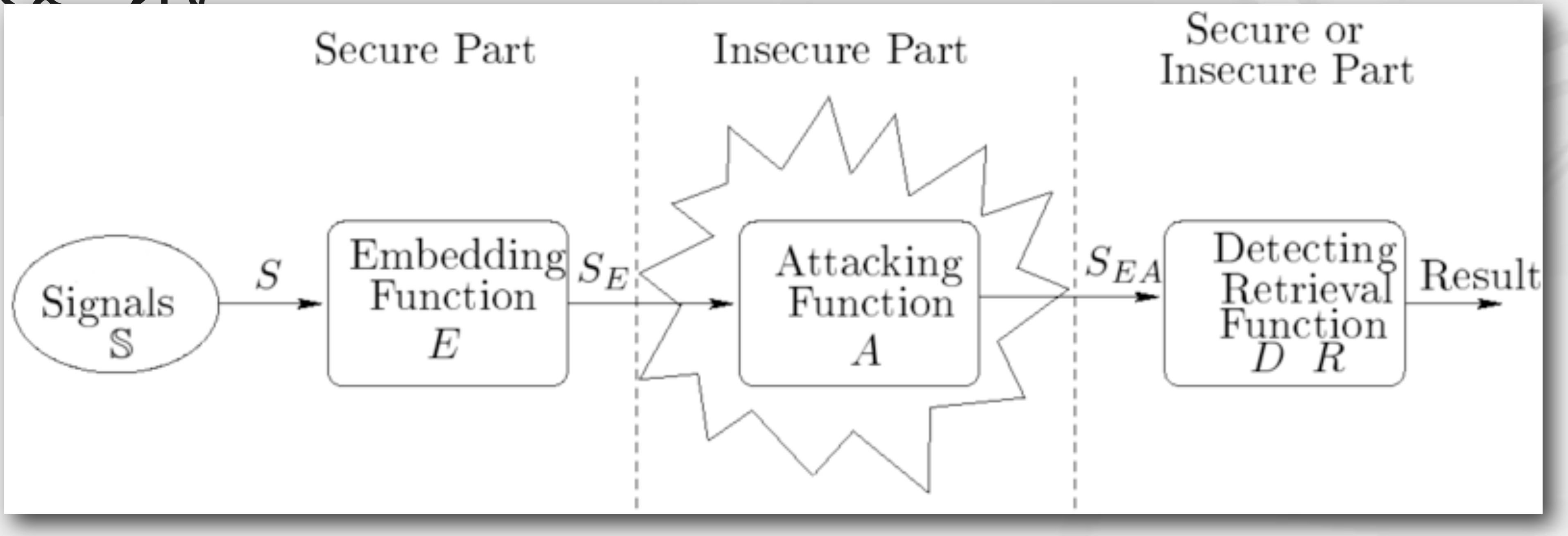
How Was CSS cracked?

- 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

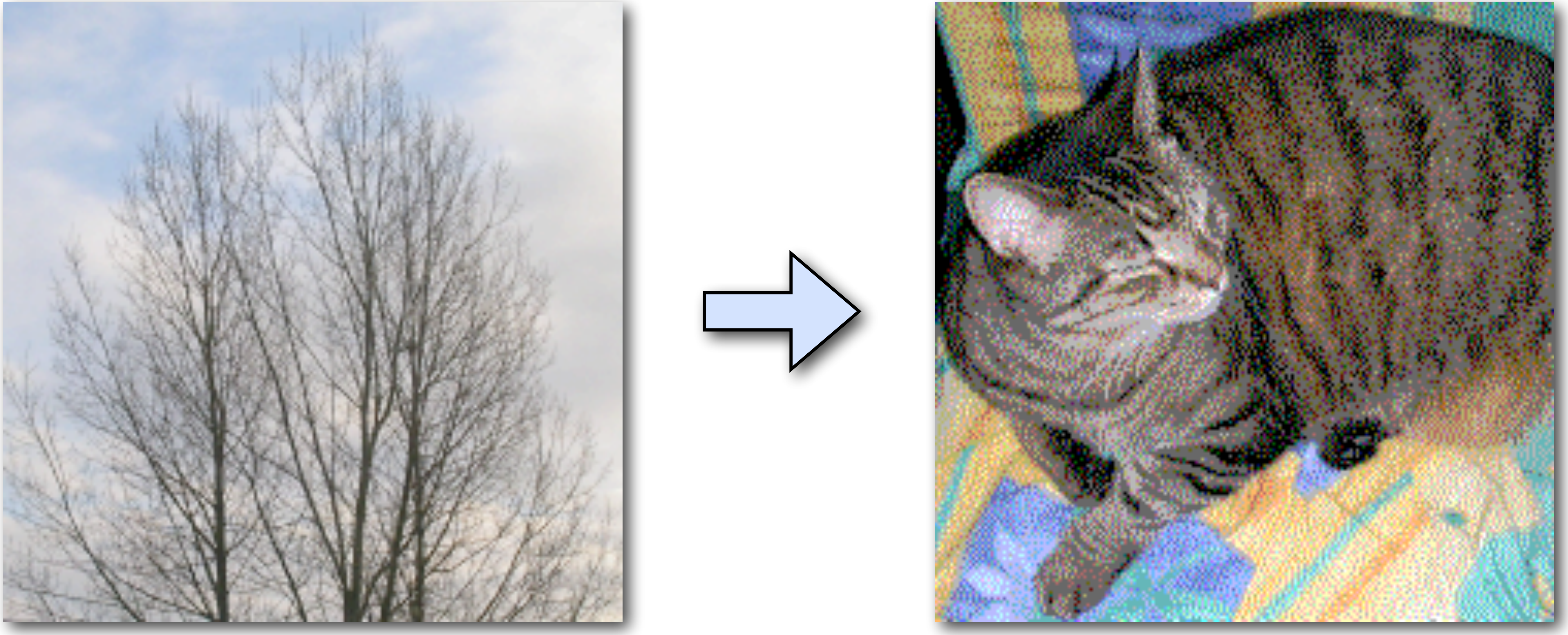
- Original goal of CSS: even if one P is compromised, others are still sound
- Flaw: weakness in the algorithm allowed *all* P's to be compromised once a single P was found
  - Why wasn't this flaw discovered *before* the algorithm went into production players?

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

About homework-03

Digital Watermark

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

Digital Watermark

- Image

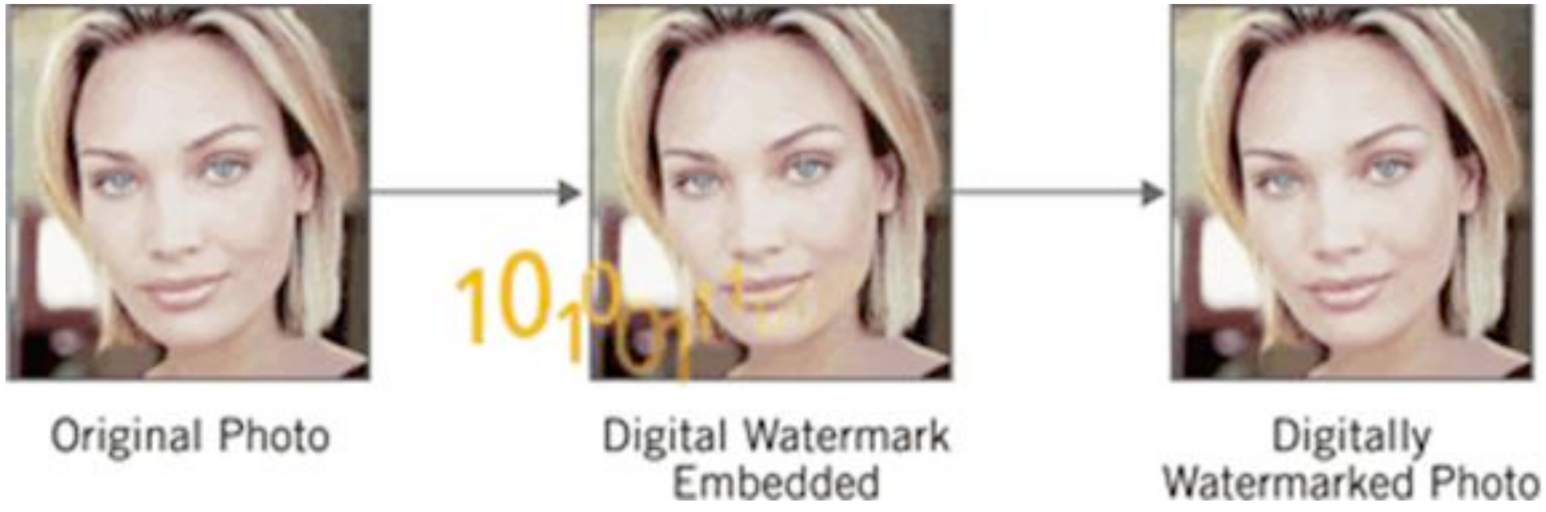
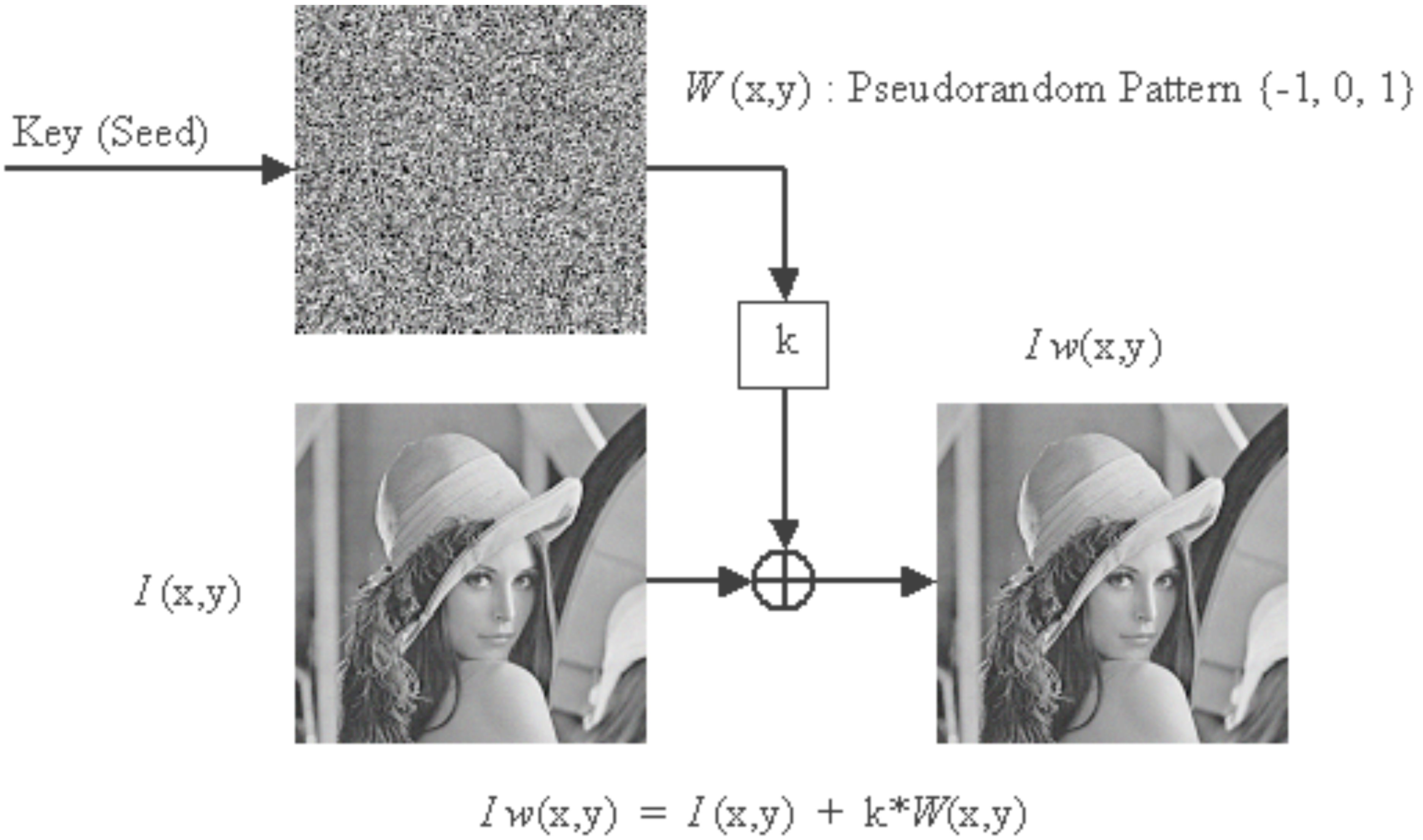
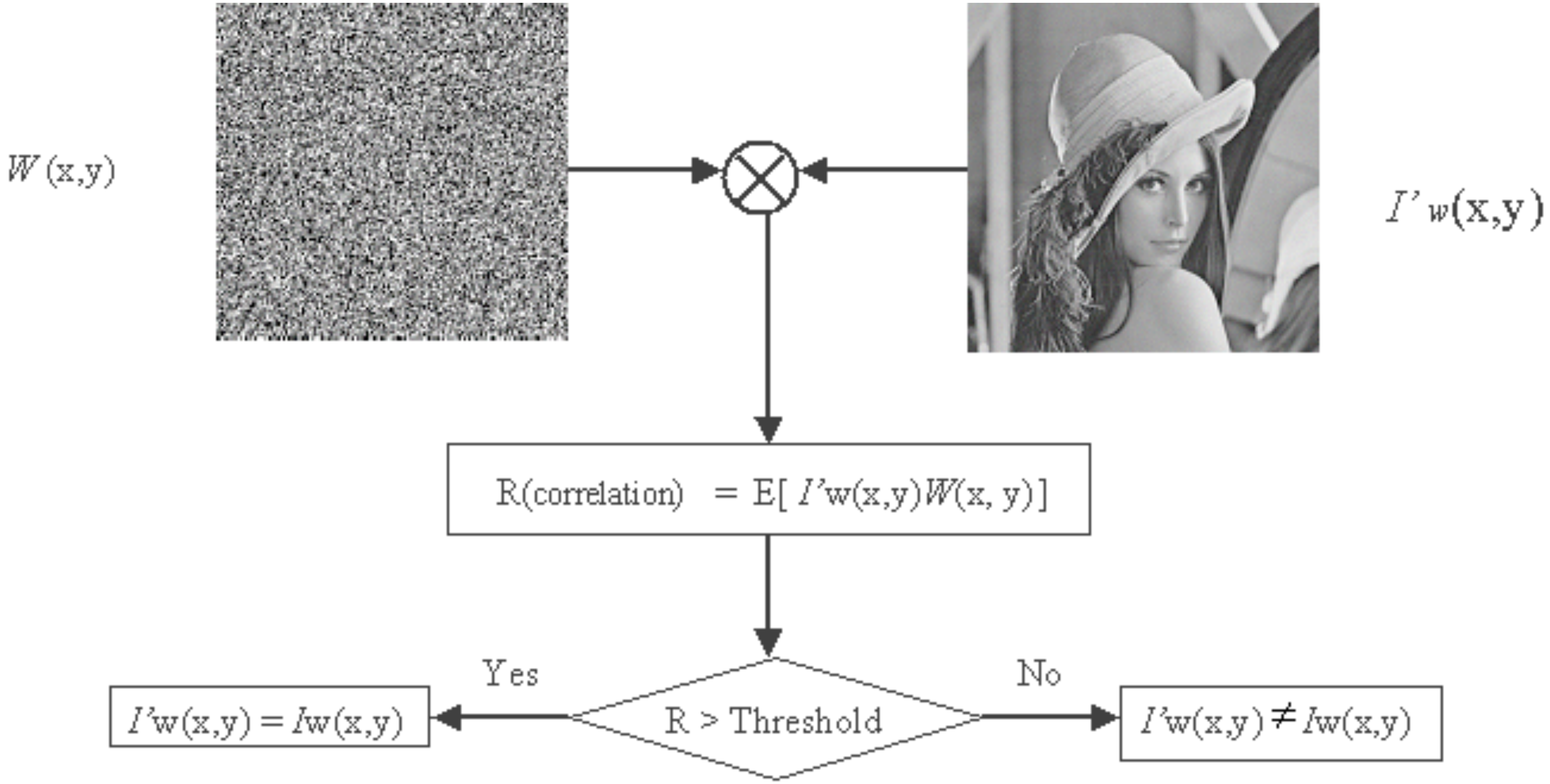


Image watermarking


W (x,y) : Pseudorandom Pattern {-1, 0, 1}
Key (Seed)
k
I w(x,y)
I (x,y)
I w(x,y) = I (x,y) + k*W(x,y)

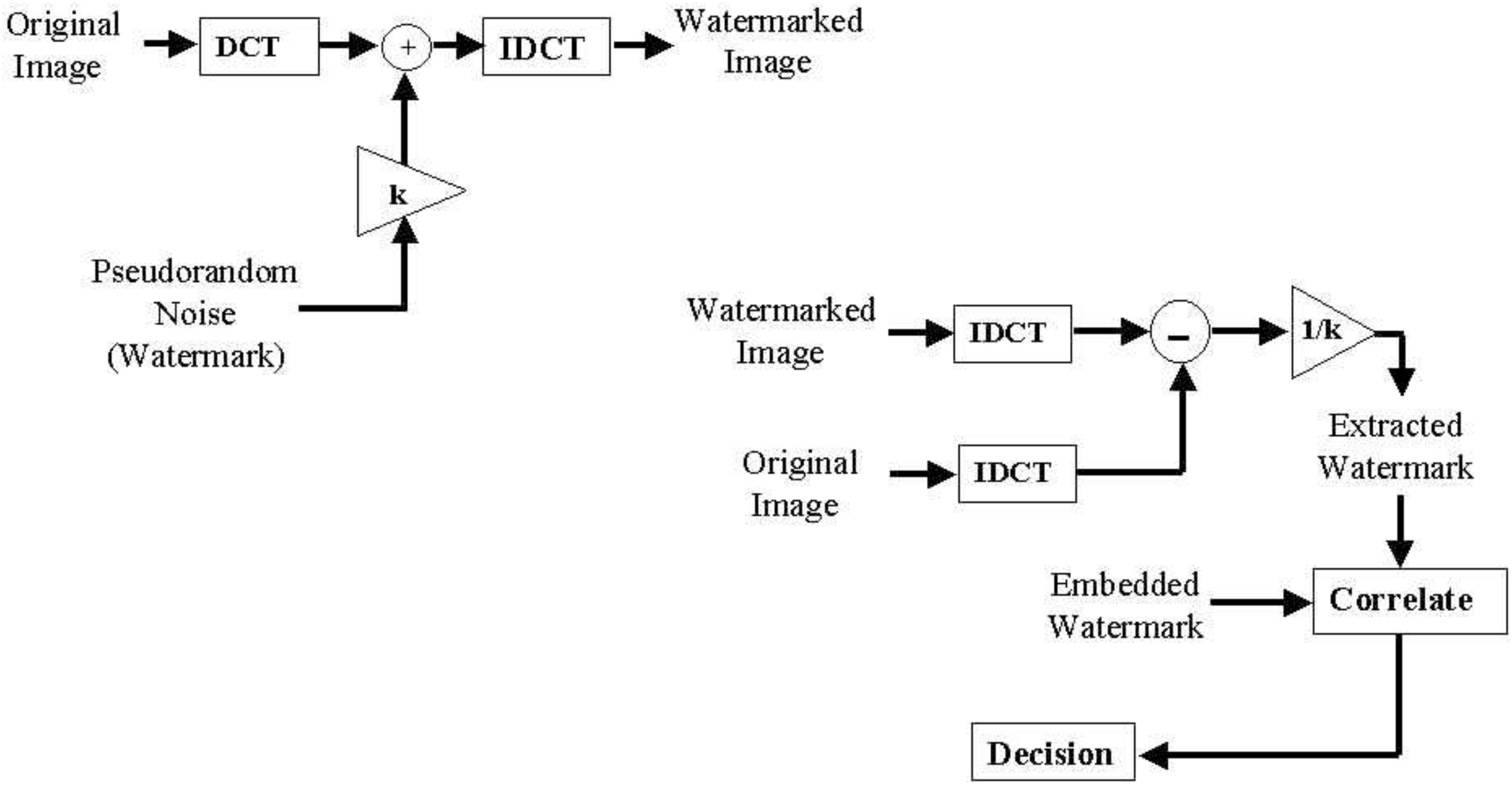
Embedding

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/steganography/mp3stego/index.html

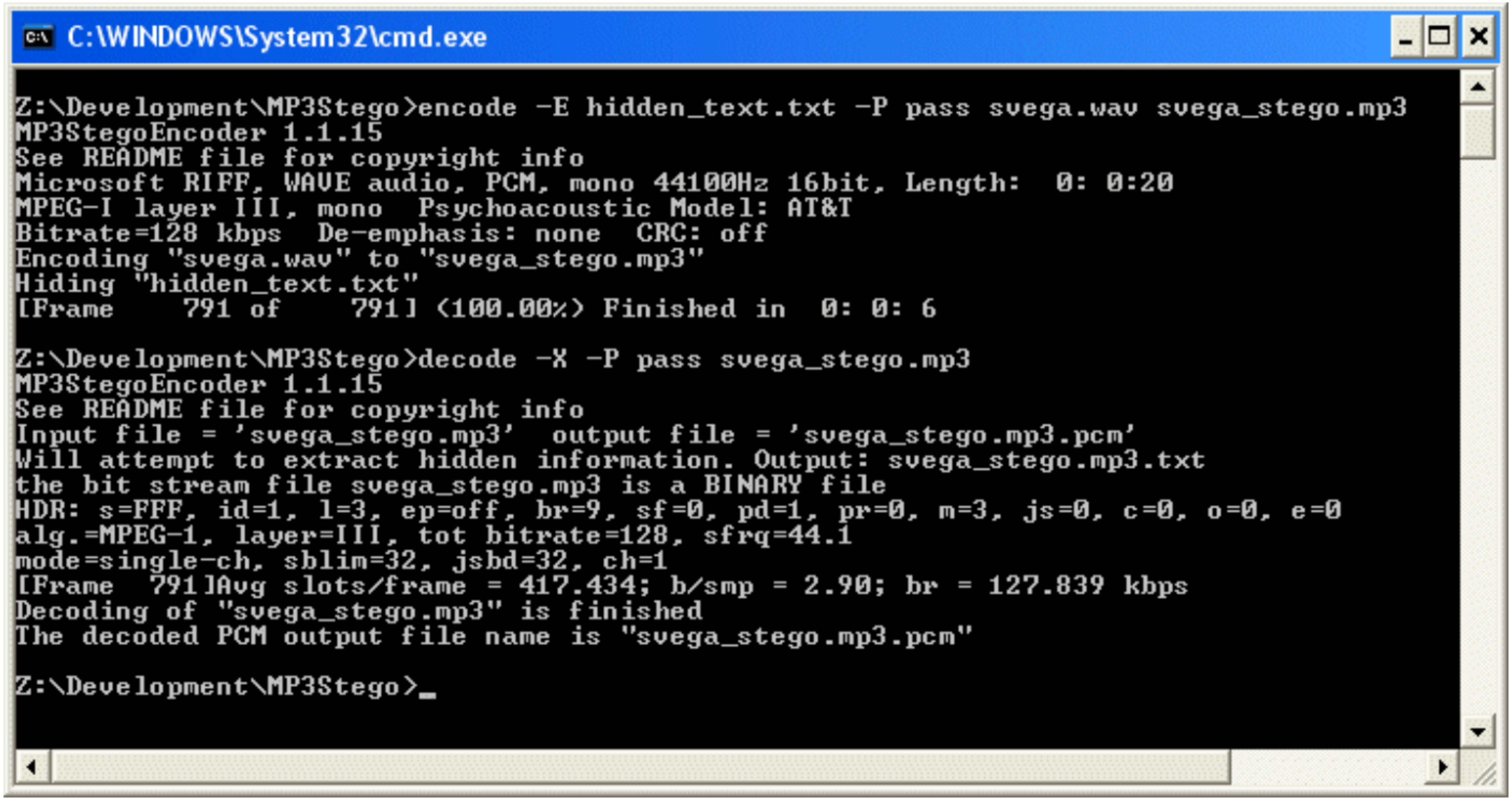
Usage example

`encode -E hidden_text.txt -P pass svega.wav svega_stego.mp3`

compresses svega.wav (mono, 44.1 kHz, 16bit encoded) and hides hidden_text.txt. The hidden text is encrypted using *pass* as a password. This produces the output called svega_stego.mp3. If no information was hidden, you would obtain this.

`decode -X -P pass svega_stego.mp3`

uncompresses svega_stego.mp3 into svega_stego.mp3.pcm and attempts to extract hidden information. The hidden message is decrypted, uncompressed and saved into svega_stego.mp3.txt.



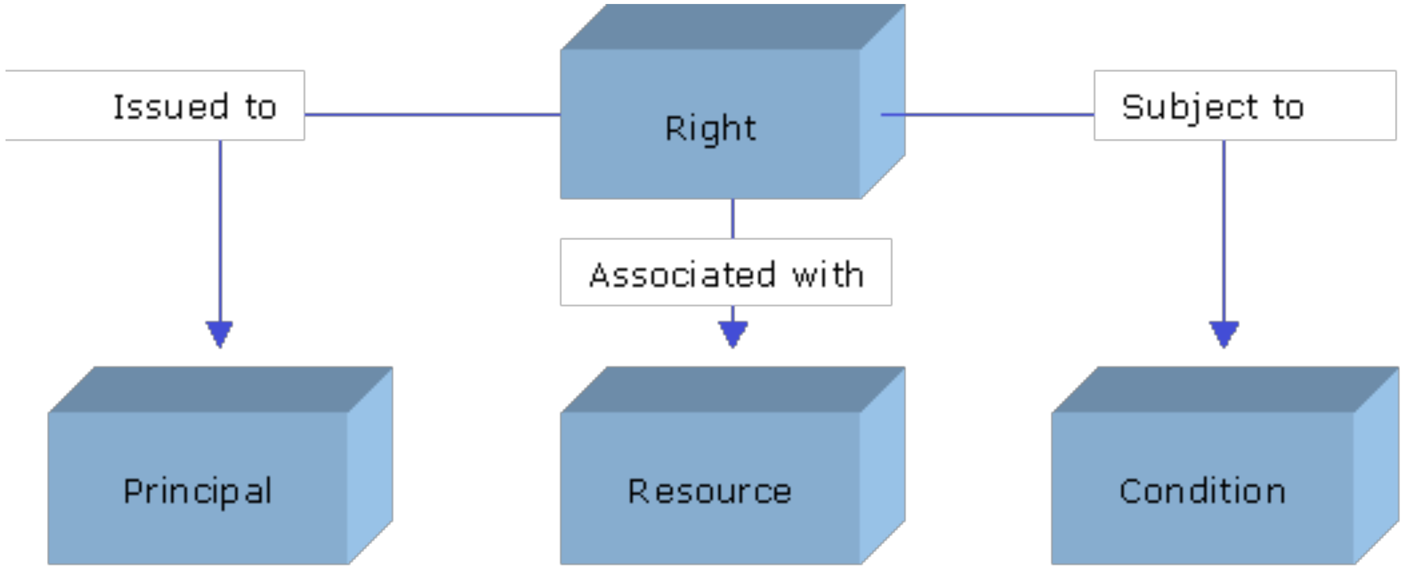
```
Z:\Development\MP3Stego>encode -E hidden_text.txt -P pass svega.wav svega_stego.mp3
MP3StegoEncoder 1.1.15
See README file for copyright info
Microsoft RIFF, WAVE audio, PCM, mono 44100Hz 16bit, Length:  0: 0:20
MPEG-I layer III, mono  Psychoacoustic Model: AT&T
Bitrate=128 kbps  De-emphasis: none  CRC: off
Encoding "svega.wav" to "svega_stego.mp3"
Hiding "hidden_text.txt"
[Frame    791 of    791] (100.00%) Finished in  0: 0: 6

Z:\Development\MP3Stego>decode -X -P pass svega_stego.mp3
MP3StegoEncoder 1.1.15
See README file for copyright info
Input file = 'svega_stego.mp3'  output file = 'svega_stego.mp3.pcm'
Will attempt to extract hidden information. Output: svega_stego.mp3.txt
the bit stream file svega_stego.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=3, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=single-ch, sblim=32, jsbd=32, ch=1
[Frame  791]Avg slots/frame = 417.434; b/smp = 2.90; br = 127.839 kbps
Decoding of "svega_stego.mp3" is finished
The decoded PCM output file name is "svega_stego.mp3.pcm"

Z:\Development\MP3Stego>_
```

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

What is digital rights management? (again)

- Digital rights management is the process of:
  - recording,
  - transmitting,
  - interpreting and
  - enforcing digital rights.

- E.g. DRM involves recording information on the digital rights of a LO or asset, delivering both the LO and its licences and conditions of use to the user and protecting the LO against unauthorised use.

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

WHY JOIN?
emusic
25
FREE
DOWNLOADS
FREE TRIAL
NO RISK
OFFER
WORKS WITH THE IPOD® OR ANY MP3 PLAYER / CHOOSE FROM OVER 2 MILLION SONGS / SONGS JUST 33¢ OR LESS AFTER FREE TRIAL
START YOUR FREE TRIAL
Pixies

- 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.

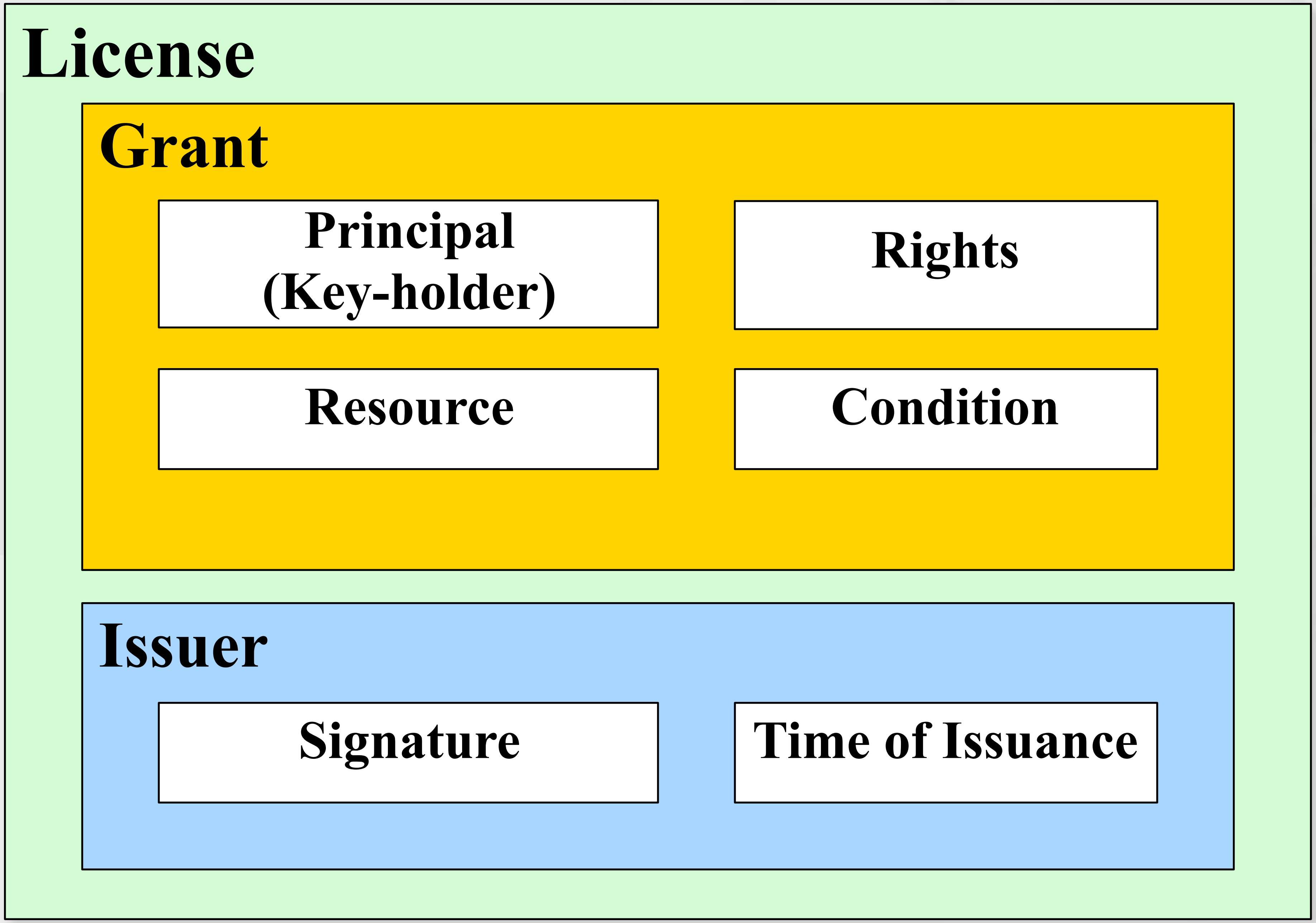
An example REL: XrML 2.X

- 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

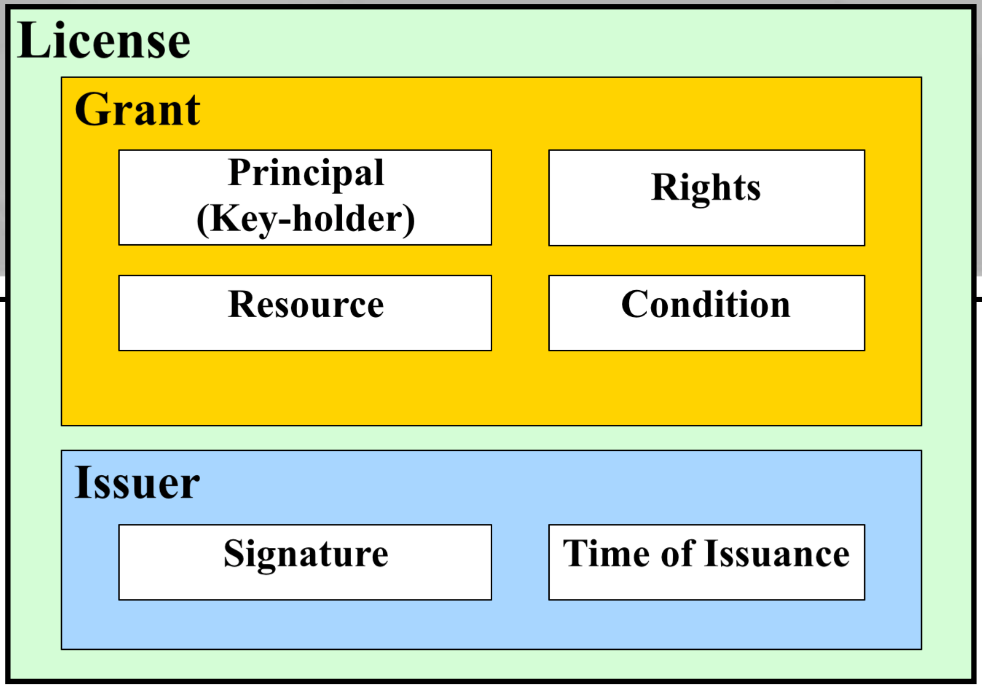
- In the RM context, XrML 2.X allows content owners a systematic way to express their intent for distribution and consumption.
- Like other policy languages, XrML 2.X **licenses** (statements) declare authorizations, but cannot enforce compliance.
  - Systems that consume XrML 2.X licenses must be trusted 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>
```

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

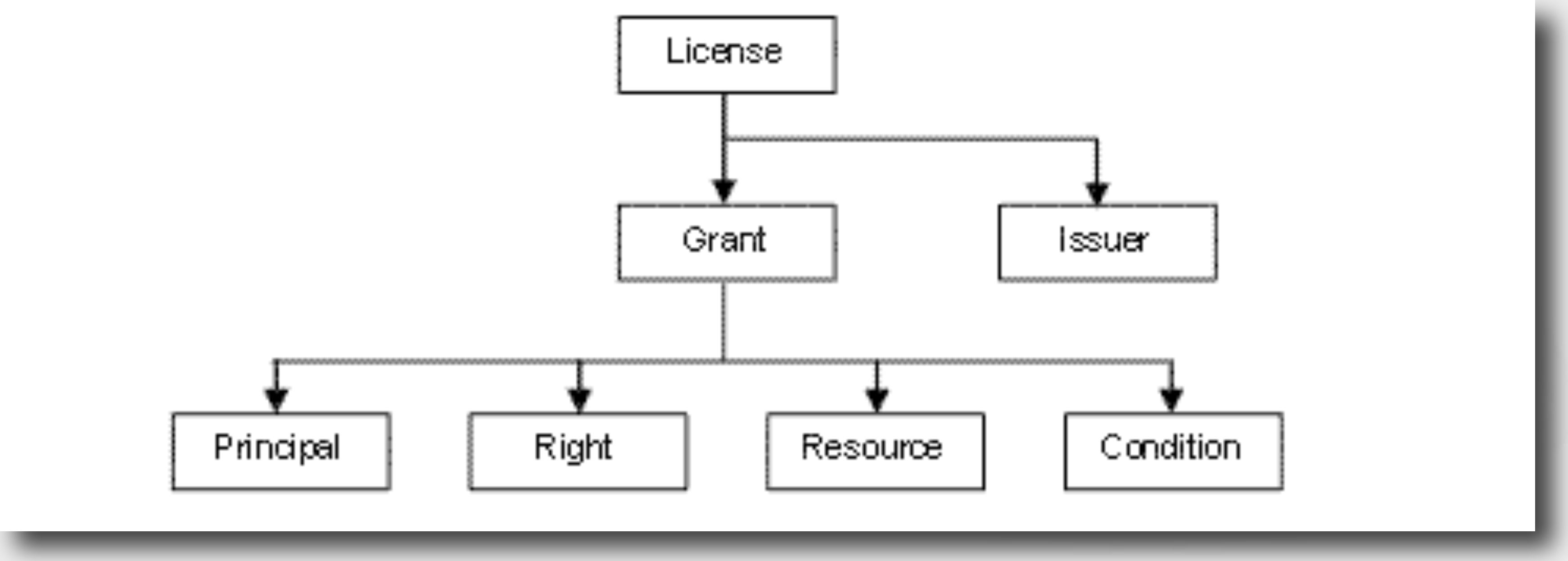
XrML Key Language Features

- 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



XrML 2.X and Multiple Authorities

- XrML 2.X offers a new level of expressiveness
  - –Enables representation of a wider range of scenarios.
- Example scenario: evaluating authorizations from multiple authorities for a resource.
  - –Today, RM systems operate using a “closed-world assumption.”
    - Any action not explicitly authorized by the content owner is prohibited.
  - –Copyright doesn’t work like this.
    - Copyright is a liability-based system.
    - Some actions are permitted by law even if they are not explicitly authorized by the copyright holder.
  - –How might we use XrML 2.X to represent authorizations as well as limitations built into the law?

XrML 2.X and Multiple Authorities (cont’d)

- Content creators are given exclusive rights by law; these rights are then licensed to consumers.
- Limitations on the exclusive rights contained in a copyright can be thought of as independent grants of licenses by Congress to the consumer.
    - “Congress says every library has the right to make an archival copy of a work” (17 U.S.C. 108).
    - Variables allow us to write licenses that apply to (potentially undefined) sets of content and users.
    - Congressional grants can be conditioned on possession of a licensed copy of the work.
- RM systems would need to recognize both the content owner as well as Congress as authorities for a given work.

Evaluating Policy Expressions

- RM systems attach policy expressions to content and then project that policy along with the content into a remote system.
  - Policy creators need to have confidence that the receiving system will faithfully implement the defined policies.
- For years in security research, we've built protocols that depend on trusted computing bases (TCBs) at their core.
  - The TCB must behave as expected, because it's the part of the system which you have to implicitly trust.

Attestable TCBs

- For RM systems, having a TCB locally is not sufficient to ensure very high levels of trust
  - We need to be able to prove the existence & reliance on a TCB to a remote party.
  - “Attestation”
- A content author is only going to allow content & policy to flow to TCBs (and, recursively, applications) he believes are going to behave properly.
  - “Behave” == implement policy as defined
- Content consumers are only going to let code they understand run their systems.

Trust is Central to Attestable TCBs

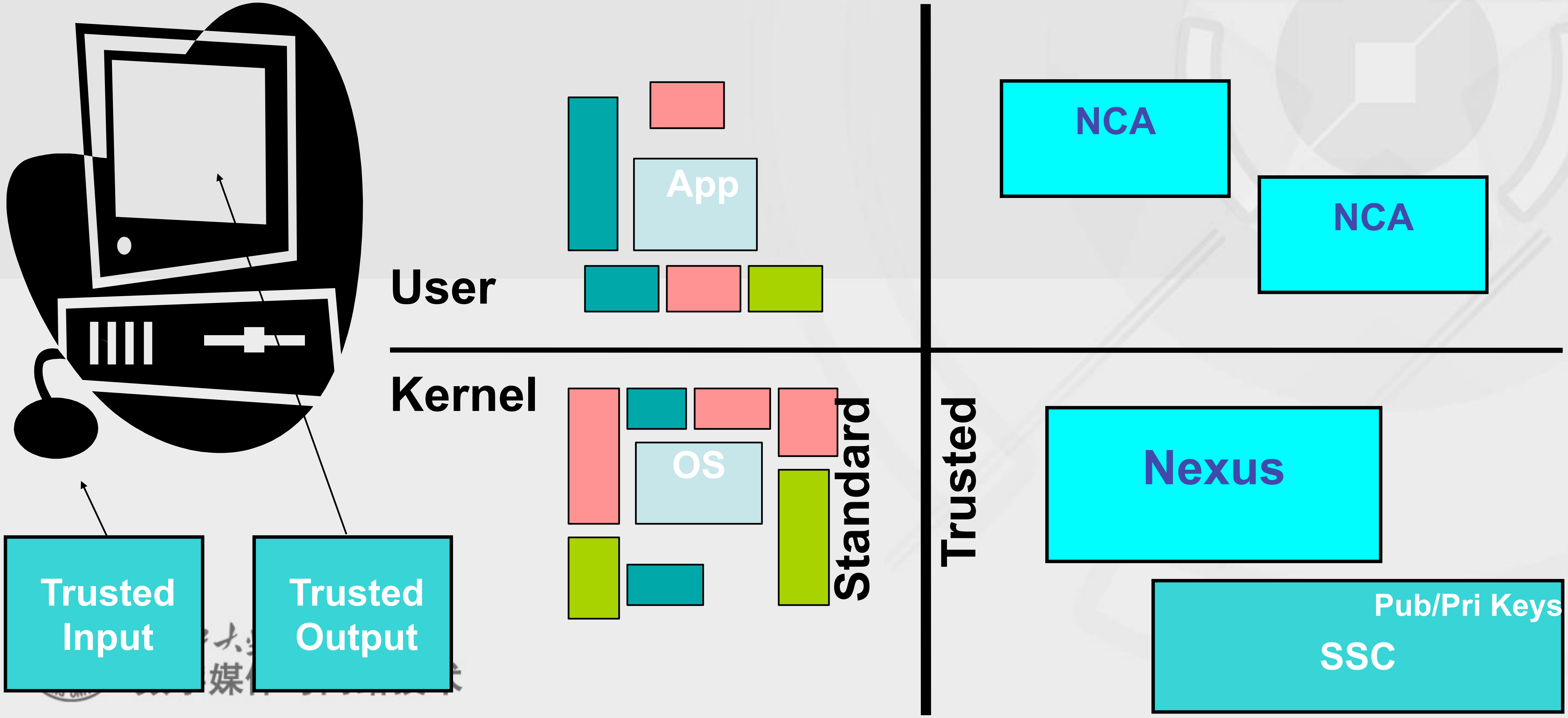
- Four elements that must be present in order to trust a TCB
  - I know who / what the it is, and that it is not an imposter
  - I know its state – it has been properly initialized
  - I know that it cannot be tampered with
  - I know that my communication with it is private and tamper-proof

Building Attestable TCBs

- There are two separate industry initiatives today trying to build attestable TCBs on evolutions of PC hardware and software.
  - TCPA – Trusted Computing Platform Alliance
  - Microsoft’s Next-generation Secure Computing Base (NGSCB)
- TCPA is specifying changes to the PC hardware that can make attestations.
  - Goal is to be able to sign statements about the entire software stack running on top of the PC, from the moment power is turned on forward
- NGSCB has a somewhat different focus.
  - Goal is to create a separate, parallel execution environment inside PCs that is rigidly controlled by the user, and make attestations about only that code.
  - Additionally, provide sealed storage, curtained memory and secure I/O with the user.

NGSCB – How It Works

- Subdivide the execution environment by adding a new mode flag to the CPU.



Attestation in NGSCB

- Attestation is a recursive process
  - The SSC (security chip) always knows the unspoofable identity of the running nexus.
  - Assuming it does, the SSC can then attest to (make signed statements about) the identity of the nexus.
    - SSC has a digital signature key pair, plus some certificates for that key pair.
  - The nexus in turn can attest to the identity of nexus computing applications (NCAs)
- If you accept the certificates & digital signature key pair as belonging to an uncorrupted SSC, then you can trust the statements the SSC makes about the running nexus.

Attestation and RM Systems

- Why would RM system builders be interested in the attestation feature?
  - –Attestation allows a host machine to query what software is running on a remote machine before sending it content.
- Examples:
  - –In an enterprise RM environment, servers could be configured to only release classified documents to non-portable machines.
  - –Before sending personal information to a server, a client could demand proof that the server is running a software stack certified to comply with privacy-protecting principles.
  - –In a consumer RM environment, content could be licensed such that it could freely migrate among all devices within a single “household”.
- Operation of the PC is never blocked; the hardware simply will not lie about the software running on top of it.
  - –Servers can choose not to talk to clients they don’t like.

Summary

- Two security technologies:
  - –Rights expression languages (RELs)
  - –Attestable TCBs
- These technologies provide a number of new security features for computing platforms, including advances in secret storage and policy expression, evaluation and projection.
- RM systems built on today’s platforms are useful for a wide variety of solutions; the features provided by RELs and attestable TCBs will further expand that set.

Agenda

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

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)

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?

课后思考题

- 数字版权管理与区块链技术融合的可能性?
  - BAT： https://www.basicattentiontoken.org/
  - Steemit：https://steemit.com
  - Singulardtv：https://singulardtv.com
  - More …
  - 大家回去各自收集资料