Cryptology
& Security
Initiative

# Comment cacher un message dans une image fixe ?


Jang Schiltz

Assistant Professeur à l'Université du Luxembourg

Philippe Niederkorn

Chercheur au CRP – Gabriel Lippmann


11-06-2004

# Structure de l'exposé (1)

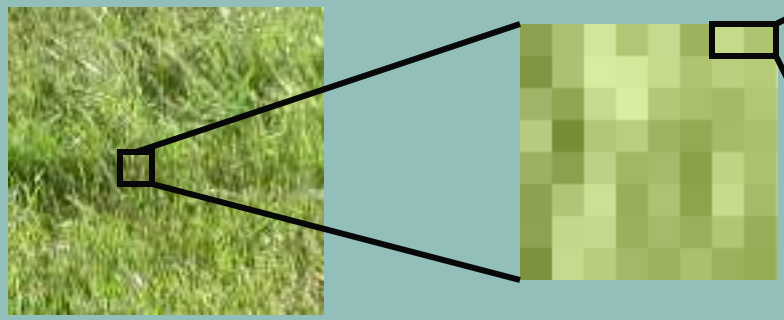
Cryptology
Security
tiative

1. Substitution du bit le moins significatif
2. Images codées avec système de palette
3. Le format JPEG
4. Cacher le message dans les coefficients de la DCT
5. Quelques méthodes récentes

# Substitution du bit le moins significatif

Cryptology & Security Initiative

image agrandissement pixels digitalisés



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | R |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | G |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | B |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | R |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | G |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | B |

message secret

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0 |

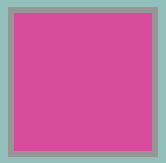


Comment cacher un message dans une image fixe?

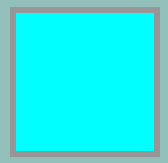
Université du Luxembourg
CRP - Gabriel Lippmann

# Bit le moins significatif

Cryptology & Security Initiative



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| R | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 R |
| G | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 255 G |
| B | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 255 B |



couverture

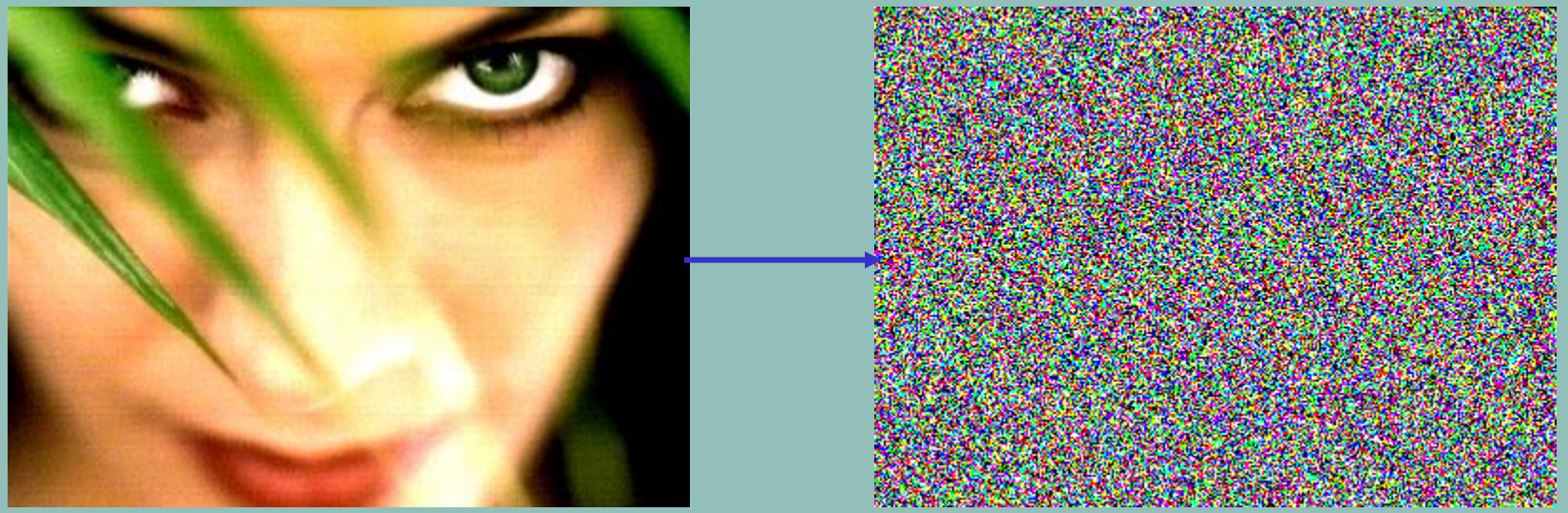
Comment cacher un message dans une image fixe?



Université du Luxembourg
CRP - Gabriel Lippmann

# Attaque visuelle

Cryptology
& Security
Initiative



stéganogramme

Comment cacher un message dans une image fixe?



Université du Luxembourg
CRP - Gabriel Lippmann

# Attaques

Cryptology & Security Initiative

***détection*** — Attaque visuelle ou statistique.

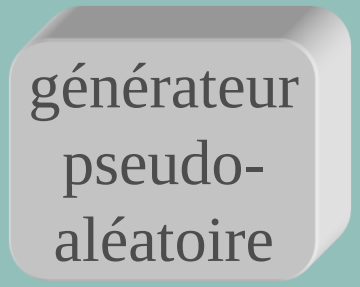
***extraction*** — Immédiate dès que l'on a conscience de la présence d'un message dissimulé.

***destruction*** — Très facile, puisqu'il suffit de remplacer les bits les moins significatifs par une suite aléatoire.

***falsification*** — Elémentaire, si l'on parvient à déterminer la nature du pré-traitement du message.

# Utilisation d'un générateur pseudo-aléatoire






2131234234132311432312112434212…

semence du générateur
=
clé de couverture

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

message secret

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0 |

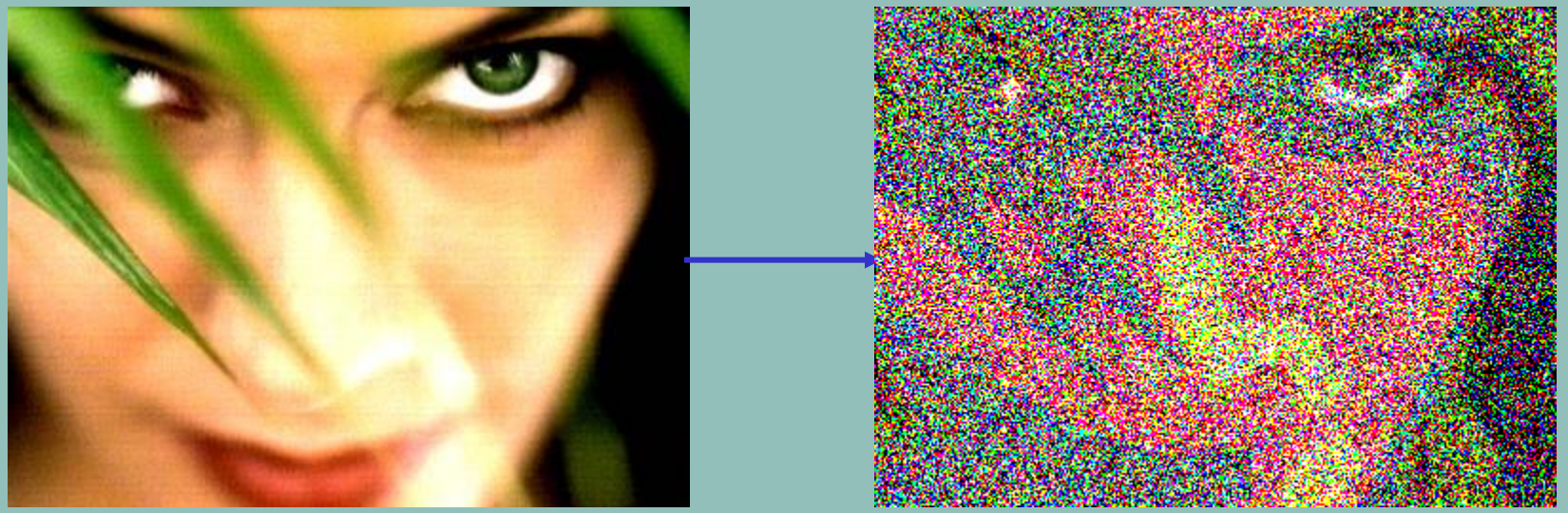


Comment cacher un message dans une image fixe?

Université du Luxembourg
CRP - Gabriel Lippmann

# Attaque visuelle

Cryptology
& Security
Initiative



stéganogramme

Comment cacher un message dans une image fixe?



Université du Luxembourg
CRP - Gabriel Lippmann

# Attaques



***détection***

L'analyse statistique reste possible : plus il y a d'espace entre les bits altérés, plus les chances de détection sont faibles...

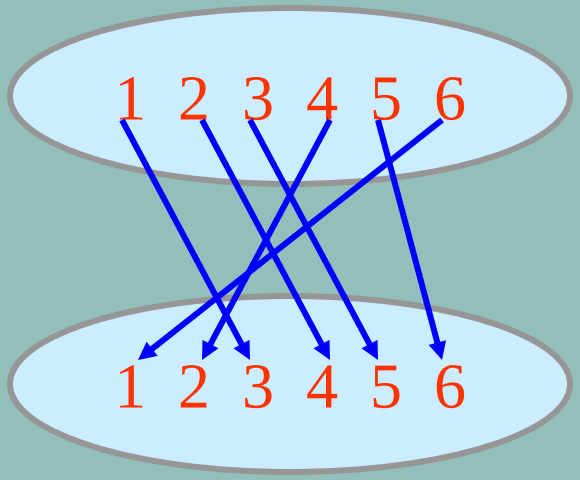
***destruction***

Même si aucun message n'est détecté, il est toujours possible de remplacer à titre préventif les bits les moins significatifs par une suite aléatoire.

# Ajout d'une permutation



permutation




message secret

| 0 | 1 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|

message secret permuté

| 0 | 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|

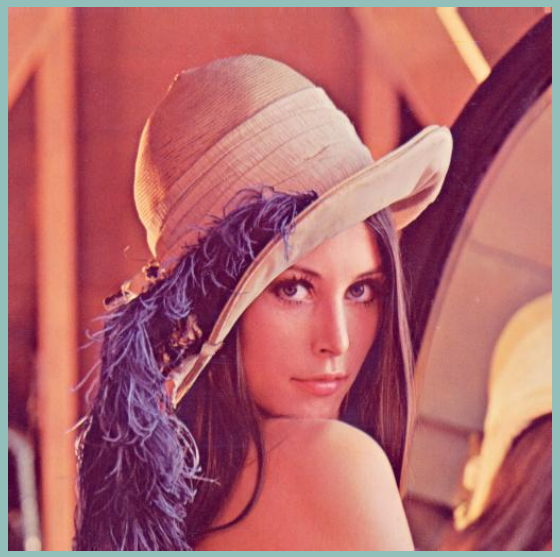
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |



Comment cacher un message dans une image fixe?

# Adaptation du chiffre de Francis Bacon





couverture

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 255 | 255 | 255 | 153 | 53 | 2 | 45 | 102 | 79 |

principe :

nombre pair $\Rightarrow$ 0

nombre impair $\Rightarrow$ 1

couverture prétraitée

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 254 | 254 | 254 | 153 | 53 | 2 | 45 | 102 | 79 |

texte secret

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

stéganogramme

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 254 | 255 | 254 | 154 | 53 | 3 | 45 | 102 | 79 |



Comment cacher un message dans une image fixe?

Université du Luxembourg
CRP - Gabriel Lippmann

# Exemple

Cryptology & Security Initiative

couverture

stéganogramme

message secret

Le lion, terreur des forêts,
Chargé d'ans et pleurant son antique prouesse,
Fut enfin attaqué par ses propres sujets,
Devenus forts par sa faiblesse.
Le cheval s'approchant lui donne un coup de pied;
Le loup, un coup de dent; le boeuf, un coup de corne.
Le malheureux lion, languissant, triste, et morne,
Peut à peine rugir, par l'âge estropié.
Il attend son destin, sans faire aucunes plaintes,
Quand voyant l'âne même à son antre accourir:
«Ah! c'est trop, lui dit-il; je voulais bien mourir;
Mais c'est mourir deux fois que souffrir tes atteintes.»

Comment cacher un message dans une image fixe?



Université du Luxembourg
CRP - Gabriel Lippmann

# Taille maximale du message caché

Cryptology
& Security
Initiative

format RGB : 3 octets par pixel

| | | |
|---|---|---|
| remplacement du bit de poids faible | image secret d'un 1/8 de la taille de la couverture | on peut cacher 3 caractères dans 8 pixels |
| remplacement des 2 bits les moins significatifs | image secret d'un 1/4 de la taille de la couverture | on peut cacher 3 caractères dans 4 pixels |

# Utilisation de plusieurs couvertures

Cryptology & Security Initiative


couverture personnelle
=
facilement retraçable !!

⊕

message secret

bits de poids faible d'une image de la NASA

⊕

bits de poids faible d'une image de Disneyworld

⊕

bits de poids faible d'une image d'un film célèbre


Comment cacher un message dans une image fixe?



Université du Luxembourg
CRP - Gabriel Lippmann

# Pourquoi cela fonctionne ?



L'addition bit par bit

| x | y | x⊕y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**$x \oplus x = 0$**
**chaque élément est son propre opposé**

additionner = soustraire

$$C = M \oplus N \oplus D \oplus L$$
$$\Rightarrow 0 = M \oplus C \oplus N \oplus D \oplus L$$
$$\Rightarrow N = M \oplus C \oplus D \oplus L$$

# Cacher un bit dans toute une région



$$p(I) = \sum_{j \in I} LSB(c_j) \pmod 2$$

$p(I_1) = 0$ ⟸ $p(I_1) = 1$

$p(I_2) = 1$

$p(I_3) = 1$ ⟸ $p(I_3) = 0$

$p(I_4) = 0$ ⟸ $p(I_4) = 1$

$p(I_5) = 1$

$p(I_6) = 0$

message secret

| 0 | 1 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|

couverture

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $I_1$ | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| $I_2$ | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| $I_3$ | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| $I_4$ | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| $I_5$ | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| $I_6$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |



Comment cacher un message dans une image fixe?

Université du Luxembourg
CRP - Gabriel Lippmann

# Exemple : les logiciels Stegodos et wbStego4

Cryptology
& Security
Initiative

Stegodos

wbStego4

fonctionne sous DOS

fonctionne sous Windows 95/98, Windows NT 4.0 et Windows 2000

permet de cacher un fichier de moins de 8kb dans n'importe quel type de couverture

permet de cacher un fichier dans une image BMP, un texte au format ASCII ou ANSI, une page HTML ou un fichier PDF

# Structure de l'exposé (2)

Cryptology
Security
tiative

1. Substitution du bit le moins significatif
2. Images codées avec système de palette
3. Le format JPEG
4. Cacher le message dans les coefficients de la DCT
5. Quelques méthodes récentes

# Images codées avec système de palette



format RGB
rouge = 256 valeurs
vert = 256 valeurs
bleu = 256 valeurs

16.777.216
couleurs
différentes !!


palette
réduite

GIF : palette de 256 couleurs
BMP : palette de $2^n$ couleurs

gain important
de place !!



Comment cacher un message dans une image fixe?

# Cacher le message dans la palette


2 entrées voisines dans palette
=
pas forcément deux couleurs proches !!

substitution LSB doit être modifiée

utiliser une palette initiale de 128 au lieu de 256 couleurs et leur associer les numéros pairs

ajouter après chaque couleur une couleur très ressemblante

utiliser les méthodes LSB classiques

ou bien

réordonner les couleurs de la palette de façon à ce que les couleurs qui se suivent soient proches

utiliser les méthodes LSB classiques

# Attaques



Utiliser une palette initiale de 128 au lieu de 256 couleurs et leur associer les numéros pairs; ajouter après chaque couleur une couleur très ressemblante.

***détection***

Il suffit d'examiner la palette de couleurs pour repérer les paires de couleurs voisines similaires.

***extraction***

***destruction***

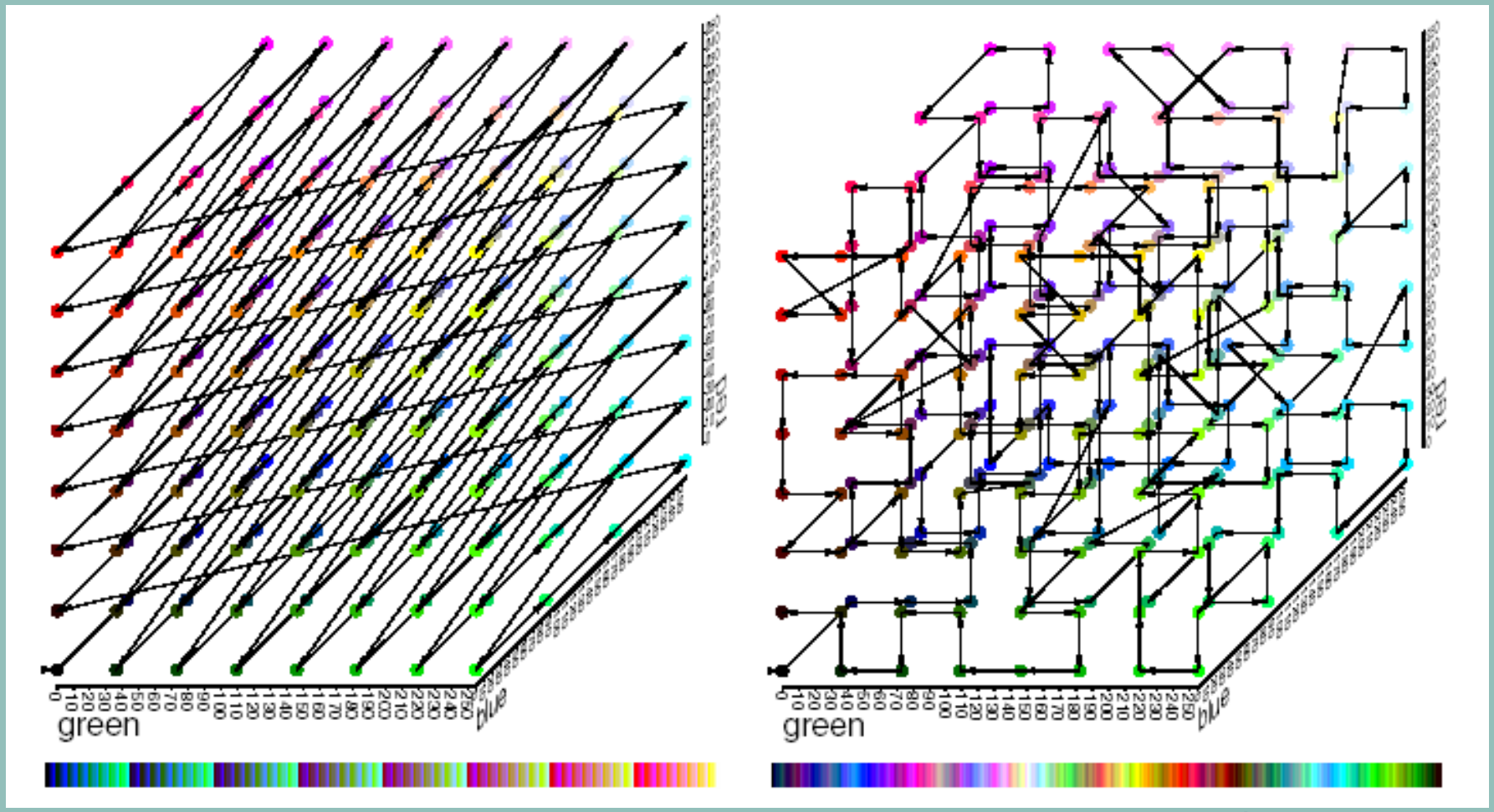
***falsification***

Les techniques expliquées précédemment pour le LSB pur restent d'application ici, sous les mêmes conditions et avec la même efficacité.

# Réordonner la palette

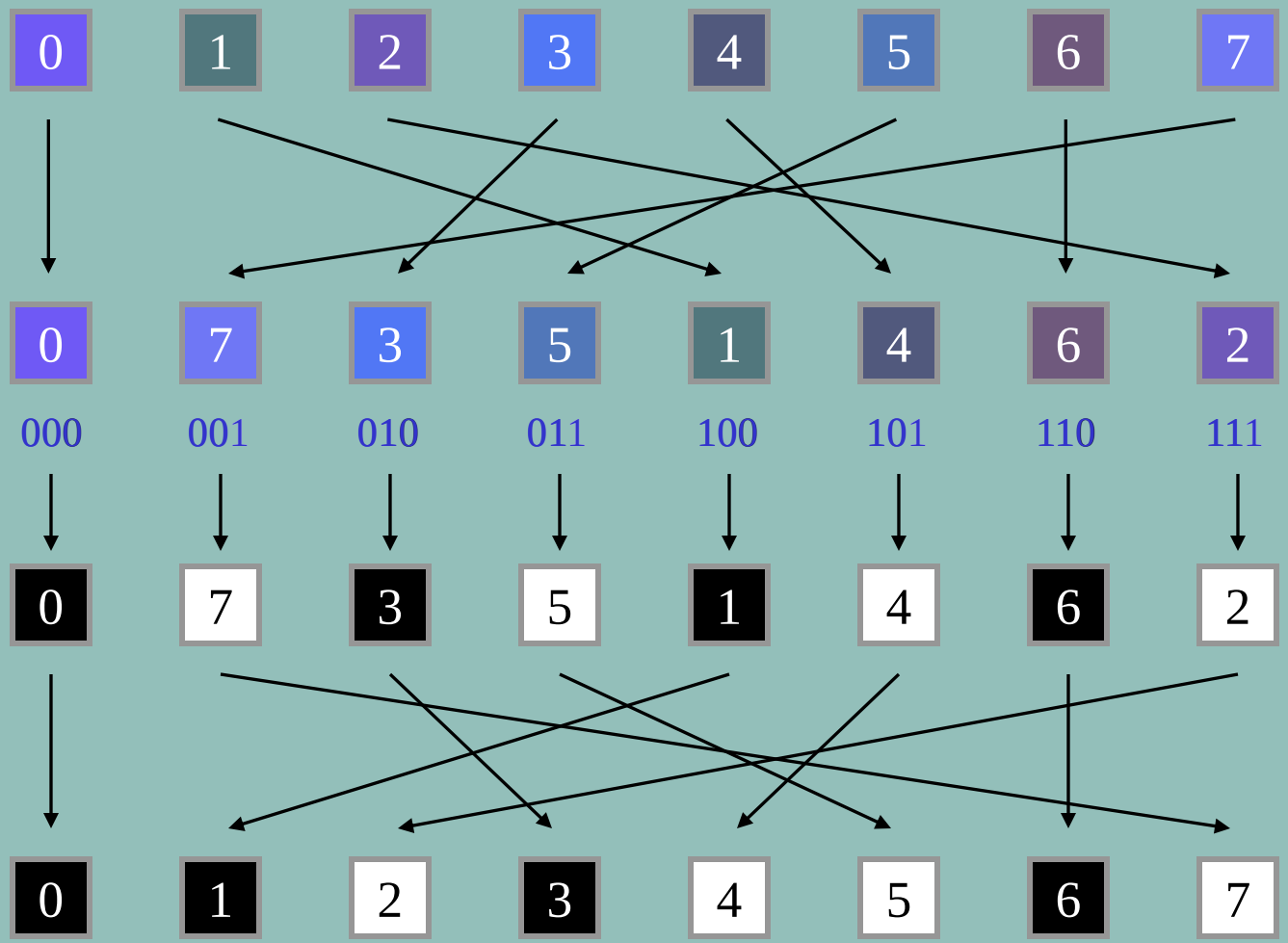


Réordonner les couleurs de la palette de façon à ce que les couleurs qui se suivent soient proches; utiliser les méthodes LSB classiques.



# Attaque visuelle

Cryptology & Security Initiative






Comment cacher un message dans une image fixe?

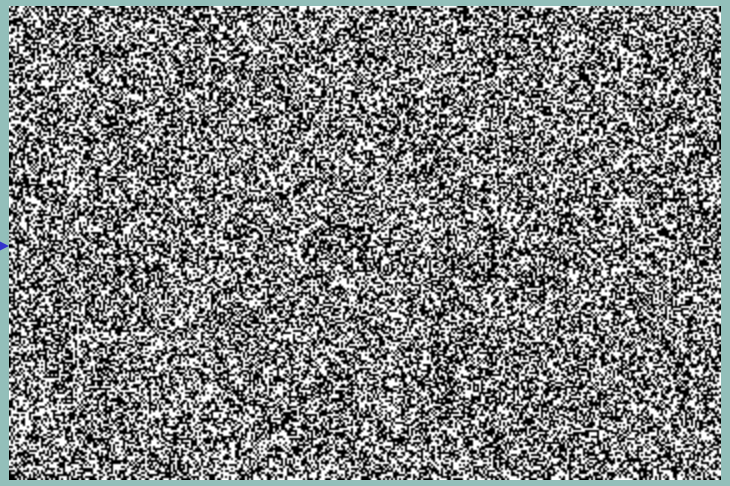
Université du Luxembourg
CRP - Gabriel Lippmann

# Illustration

Cryptology & Security Initiative

couverture

stéganogramme



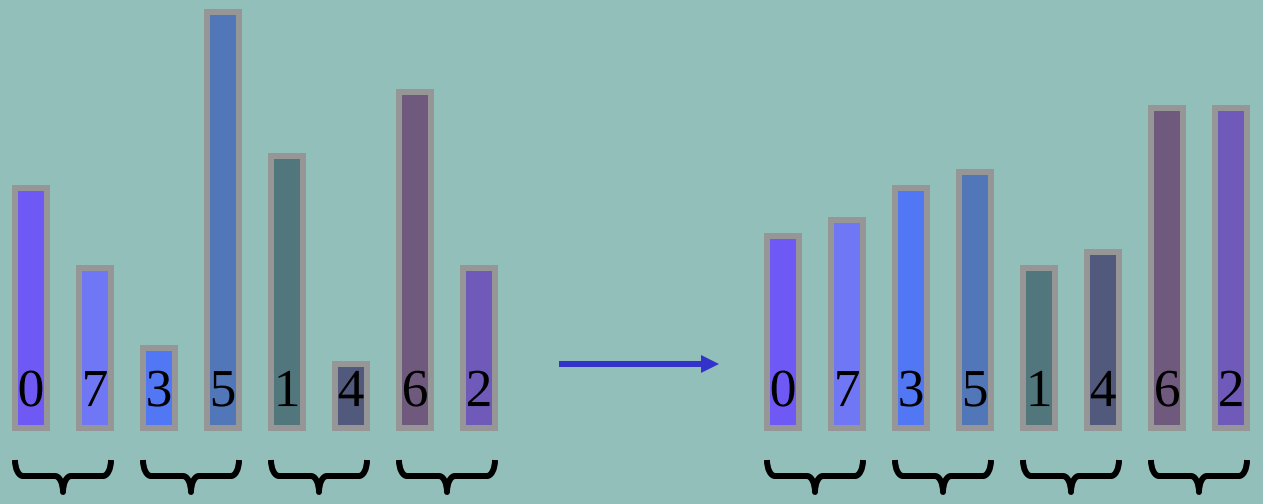
Comment cacher un message dans une image fixe?



Université du Luxembourg
CRP - Gabriel Lippmann

# Attaque statistique






couverture

stéganogramme

répartition caractéristique

# Attaques

Cryptology & Security Initiative

***détection***

Atttaque visuelle ou statistique.

***extraction***

***destruction***

***falsification***

Les techniques expliquées précédemment pour le LSB pur restent d'application ici, sous les mêmes conditions et avec la même efficacité.

# Exemple : les logiciels S-Tools, Ez – Stego et Hide and Seek

Cryptology
& Security
Initiative

S-Tools

Ez-Stego
Hide and Seek

fonctionne sous WinDos, Windows 3.11, Windows95 et Windows NT

fonctionnent sous Windows, respectivement DOS

permet de cacher un fichier dans une couverture GIF ou BMP, ainsi que des fichiers WAV et sur des disquettes DOS

permettent de cacher un fichier dans une couverture au format GIF ou BMP

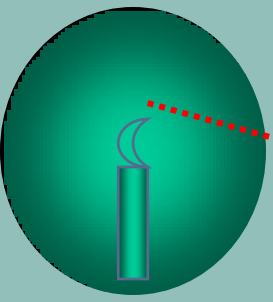
# Structure de l'exposé (3)

Cryptology & Security Initiative

1. Substitution du bit le moins significatif
2. Images codées avec système de palette
3. **Le format JPEG**
4. Cacher le message dans les coefficients de la DCT
5. Quelques méthodes récentes

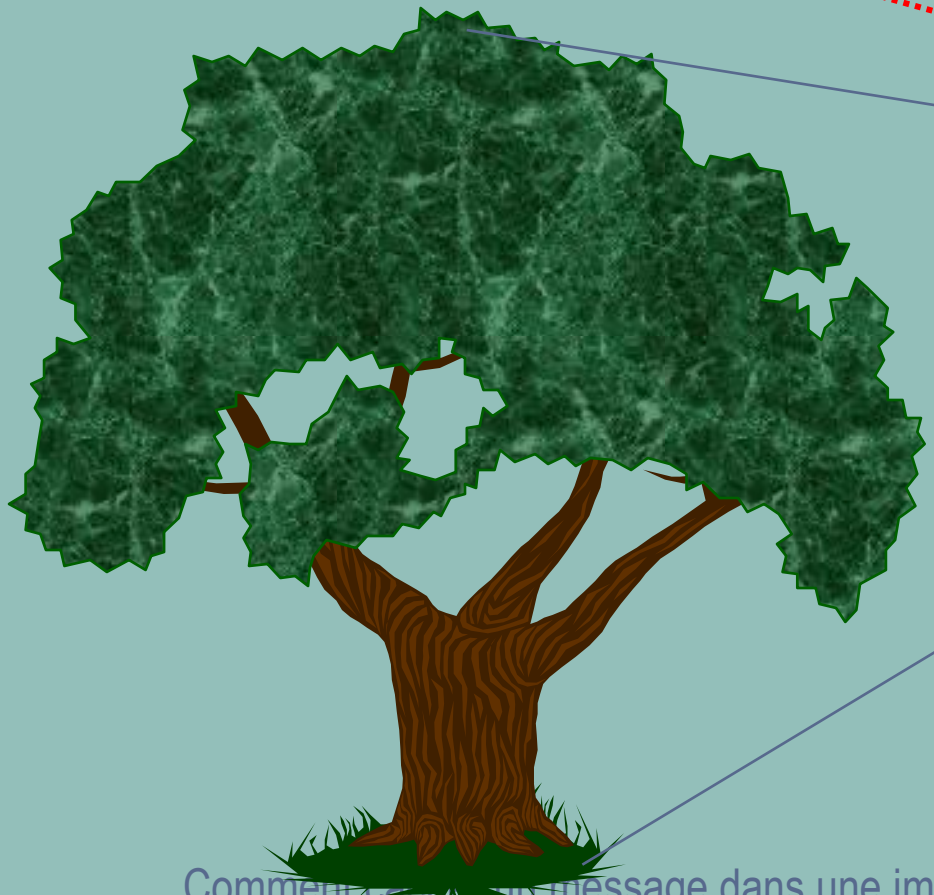
# Le format JPEG

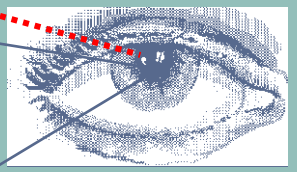
Cryptology & Security Initiative



16 km

luminosité > couleur





forme > détails

Comment cacher un message dans une image fixe?



Université du Luxembourg
CRP - Gabriel Lippmann

# Exprimer la couleur à l'aide de la luminosité

Cryptology
& Security
Initiative

luminance

$$Y = 30\% \text{ rouge} + 59\% \text{ vert} + 11\ \% \text{ bleu}$$

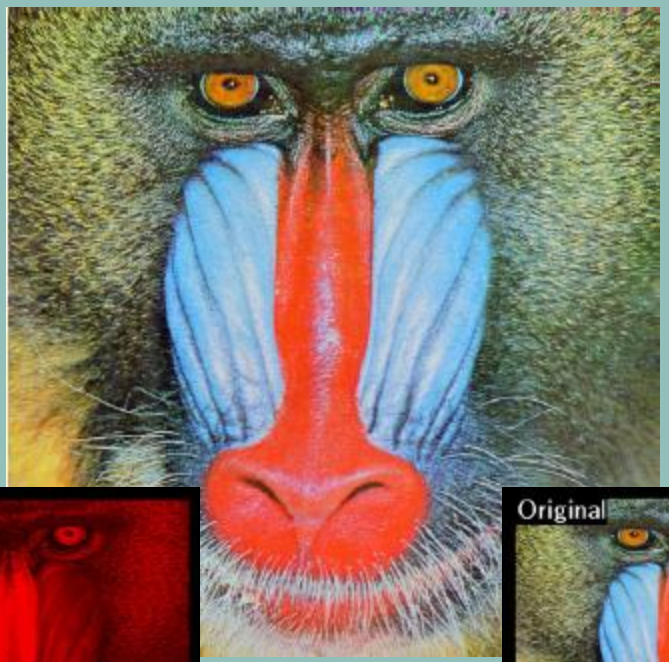
chrominance bleue

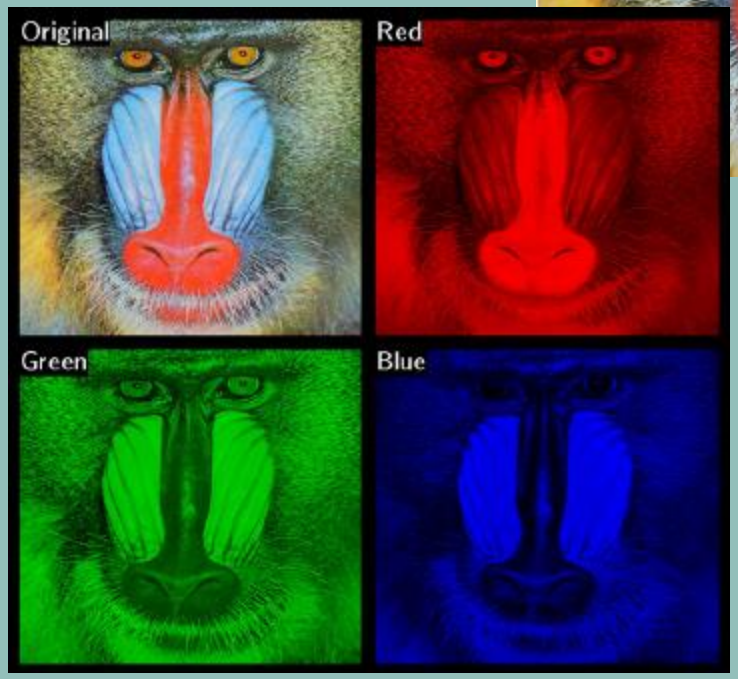
$$U = \text{bleu} - \text{luminance}$$

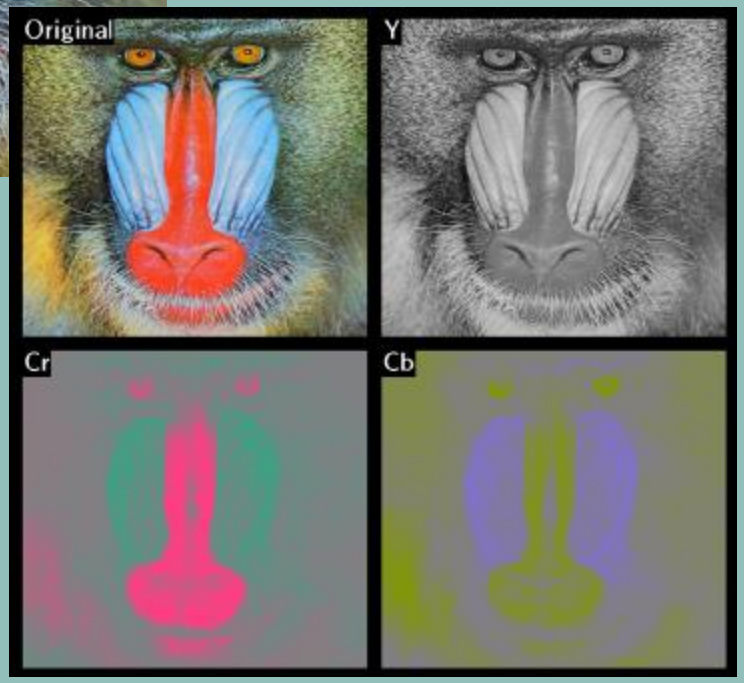
chrominance rouge

$$V = \text{rouge} - \text{luminance}$$

# RGB et YUV






Comment cacher un message dans une image fixe?



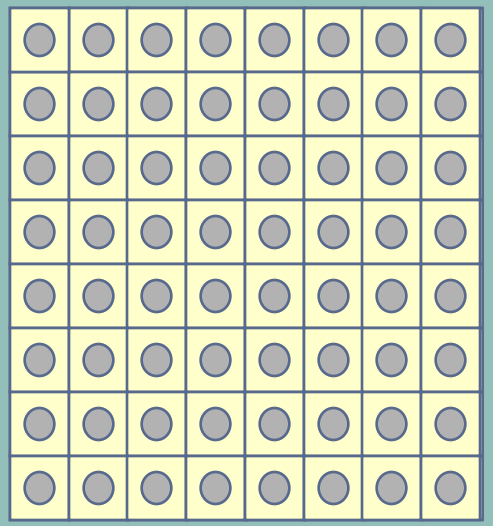
Cryptology
& Security
Initiative

Université du Luxembourg
CRP - Gabriel Lippmann

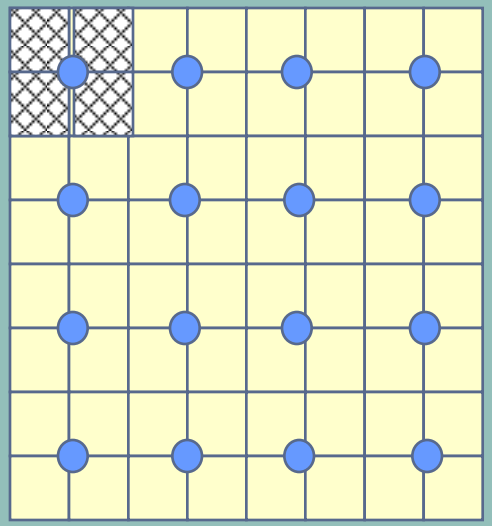
# JPEG : priviligier la luminosité



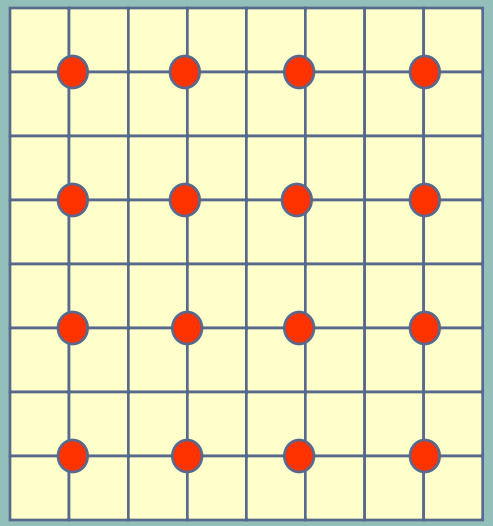
luminance




chrominance bleue



chrominance rouge



Y

64 valeurs

U

16 valeurs

V

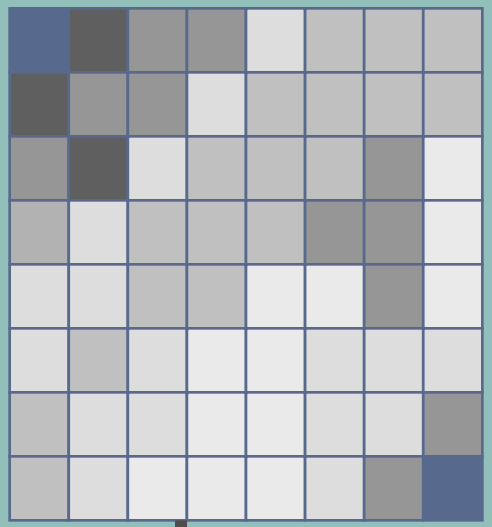
16 valeurs

96 valeurs (au lieu de 3 * 64 = 192)

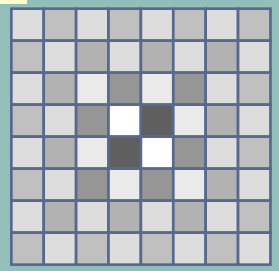
# Transformation DCT

Cryptology & Security Initiative

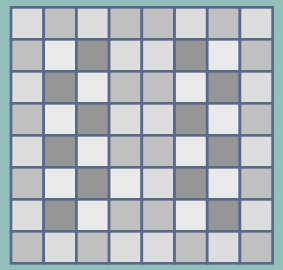
$$I = 0{,}135 * I_0 + 0{,}082 * I_1 + 0{,}105 * I_2 + \ldots + \ldots + 0{,}001 * I_{62} + 0{,}001 * I_{63}$$



calculer la ressemblance avec 64 images de référence



**0,001** $I_{63}$

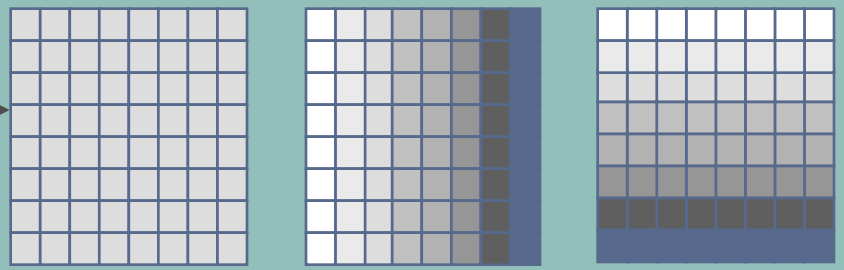


**0,001** $I_{62}$

.....

calculer ressemblance

$I_0$ $I_1$ $I_2$



**0,135** **0,082** **0,105**



Université du Luxembourg
CRP - Gabriel Lippmann

# Quantification



$$0{,}135 * I_0 + 0{,}082 * I_1 + 0{,}105 * I_2 + \ldots$$

$$\ldots + 0{,}001 * I_{62} + 0{,}001 * I_{63}$$

choix du pas de quantification

quantification

table de quantification

| | $c_0$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | . . . | $c_{61}$ | $c_{62}$ | $c_{63}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| coefficients quantifiés | 25 | 15 | 21 | 8 | 0 | 0 | 0 | 0 | 3 | . . . | 0 | 0 | 0 |

séquences de coefficients nuls

compression conservatrice



Comment cacher un message dans une image fixe?

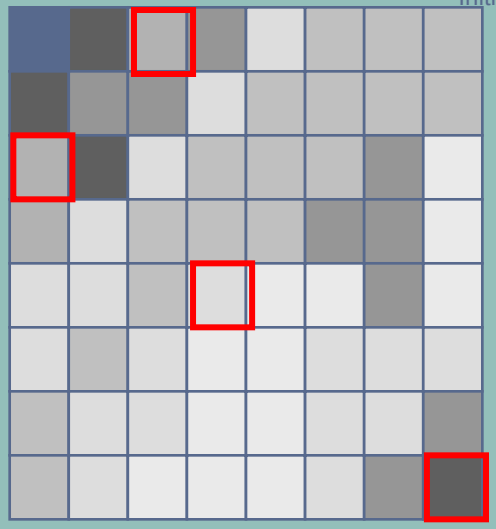
# La décompression

Cryptology & Security Initiative

perte de précision

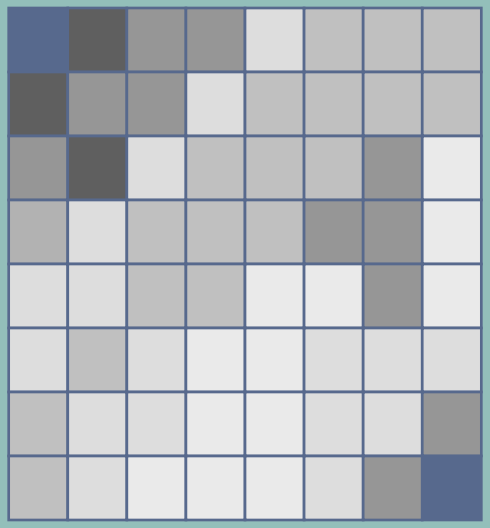
$0{,}13\mathbf{0} * I_0 + 0{,}08\mathbf{0} * I_1 + 0{,}10\mathbf{0} * I_2 + \ldots$

IDCT

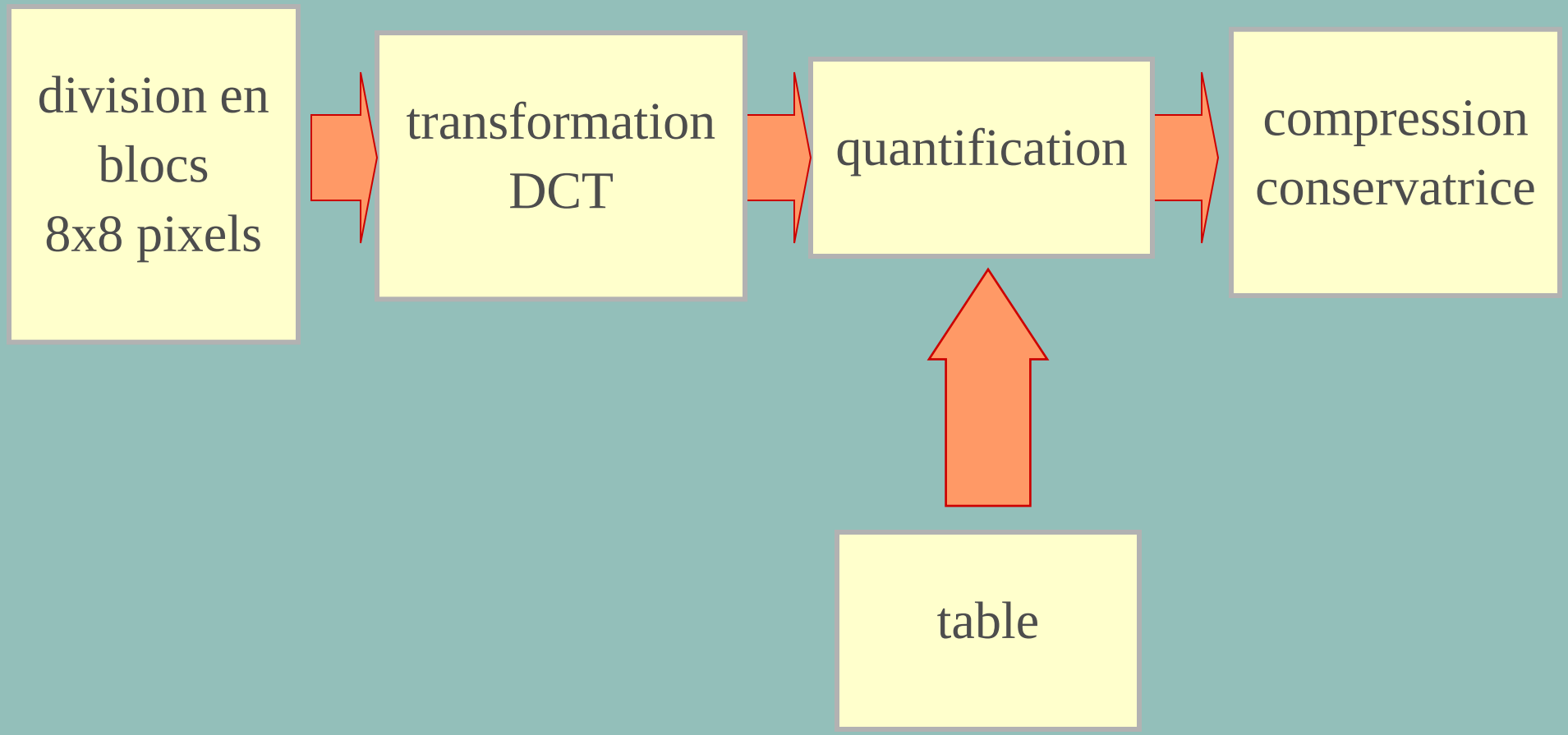


déquantification

image initiale



# JPEG : récapitulatif



# Structure de l'exposé (4)

Cryptology & Security Initiative

1. Substitution du bit le moins significatif
2. Images codées avec système de palette
3. Le format JPEG
4. Cacher le message dans les coefficients de la DCT
5. Quelques méthodes récentes

# Substitution du bit le moins significatif

Cryptology
& Security
Initiative

la compression JPEG change en général les bits de poids faible

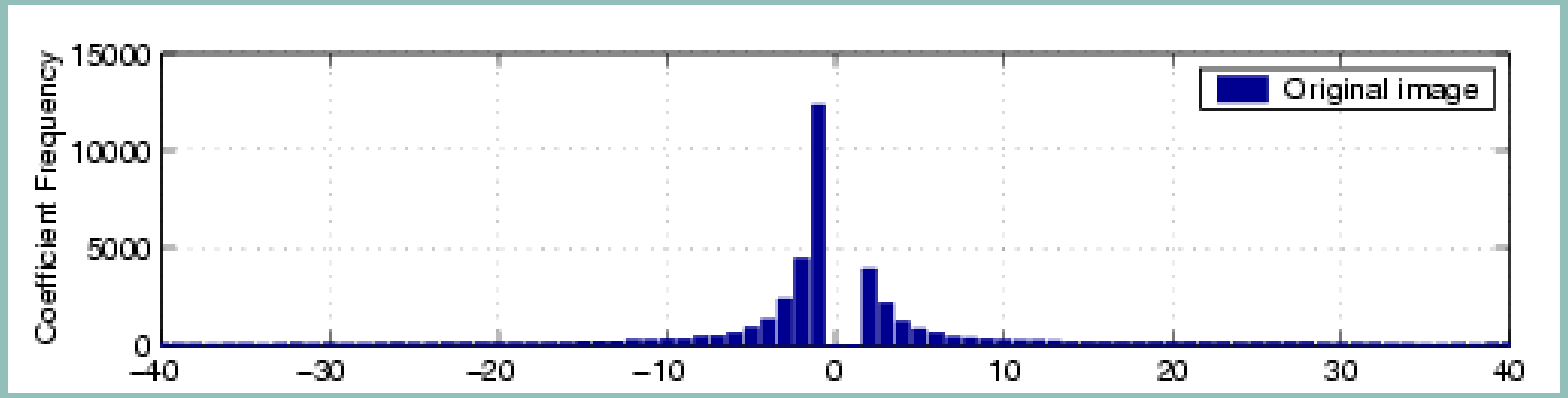
substitution LSB doit être modifiée

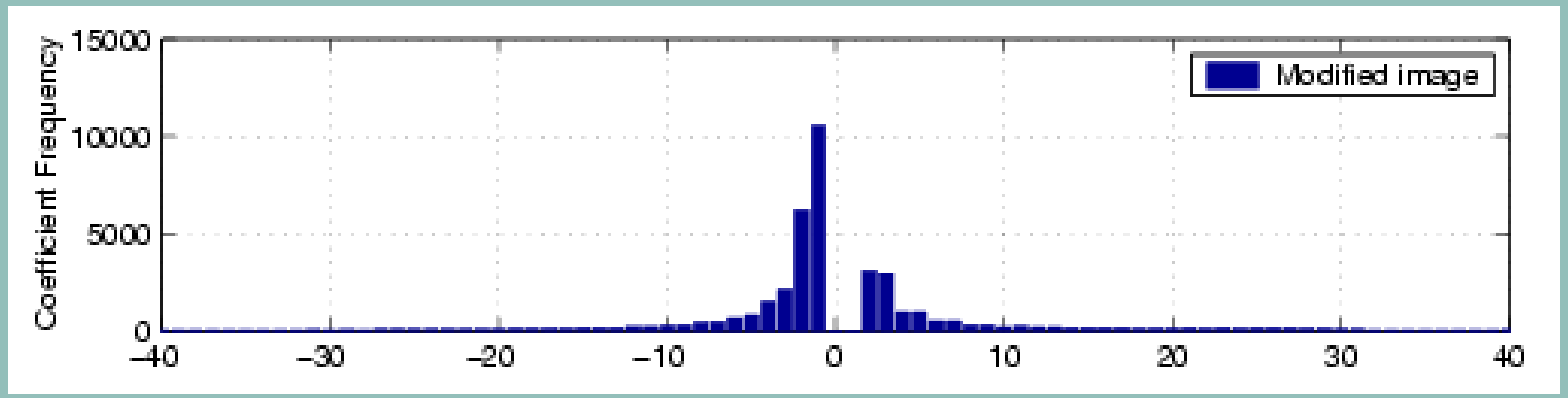
on substitue le message secret aux bits les moins significatifs des coefficients quantifiés de la DCT

Attention :
ne pas changer les coefficents nuls !

# Attaque statistique

Cryptology
& Security
Initiative



# Exemple : les logiciels JPHIDE et Jpeg-Jsteg

Cryptology
& Security
Initiative

JPHIDE

Jpeg-Jsteg

fonctionne sous WinDOS, Windows, Unix et Linux

fonctionne sous WinDOS et Windows

permet de cacher un fichier dans une couverture au format JPEG

permet de cacher un fichier dans une couverture au format JPEG

# Comment éviter la stéganalyse statistique ?

Cryptology
& Security
Initiative

méthode en 3 étapes

1. Analyse de la couverture et détermination des bits qu'on peut changer sans en changer l'aspect

dépend du format de la couverture

2. Détermination, à l'aide d'un générateur pseudo-aléatoire, d'un sous-ensemble de ces bits dans lequel le message est caché

ne dépend pas du format de la couverture

cela change les propriétés statistiques de la couverture

3. Appliquer une transformation correctrice pour rétablir les propriétés statistiques initiales

dépend du format de la couverture

on utilise les bits restants trouvés en 1

# Le processus de substitution



semence du générateur pseudo-aléatoire
=
clé de couverture
& clé de chiffrement

2 méthodes

existence d'un estimateur a priori pour la taille du message secret qu'une couverture peut cacher

substitution probabiliste qui minimise les modifications de la couverture nécessaires

utilisation de codes correcteurs qui diminuent la vraisemblance de détection

# Les transformations correctrices

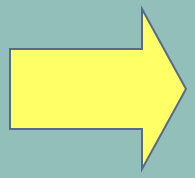


1re tentative

Si un bit est changé de 0 à 1, changer un bit assez proche de 1 à 0

Efficace contre le test de corrélation de l'entropie et contre le test de Maurer

Mais, cela ne conserve pas la distribution des coefficients DCT !!



inefficace contre les tests du chi-deux

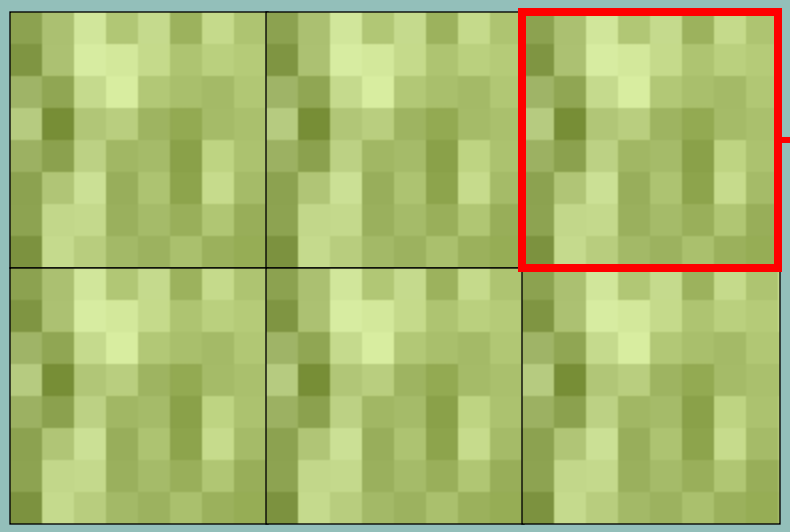
2eme tentative

Si un coefficient DCT est changé de 2i vers 2i+1, changer un coefficient DCT adjacent de 2i+1 vers 2i

cela fonctionne !!

# Discontinuité aux frontières





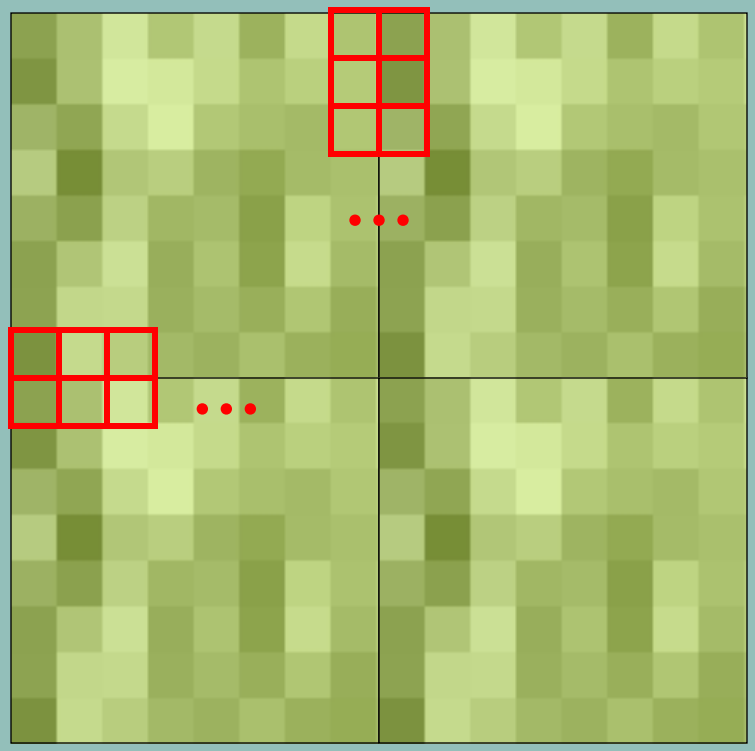
Chaque coefficient de la DCT contribue à l'aspect de tous les pixels du bloc.

Si on modifie un coefficient de la DCT dans un bloc, on fait apparaître des différences entre les pixels de la frontière du bloc altéré et les pixels de la frontière des blocs voisins !

# Détection par discontinuité



*détection*



# Exemple : le logiciel Outguess

Cryptology
& Security
Initiative

Outguess

fonctionne sous Unix et Linux

permet de cacher un fichier dans une couverture au format JPEG ou PNM

# Structure de l'exposé (5)

Cryptology & Security Initiative

1. Substitution du bit le moins significatif
2. Images codées avec système de palette
3. Le format JPEG
4. Cacher le message dans les coefficients de la DCT
5. Quelques méthodes récentes

# Cacher le message dans la DCT

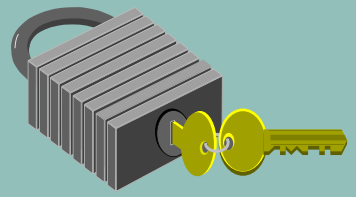


chaque bit du message est caché dans un bloc de 8x8 pixels

générateur pseudo-aléatoire

choix d'un bloc $b_i$

2e clé de couverture = choix de 2 fréquences $u_{i1}$ et $u_{i2}$ dans $b_i$



semence du générateur = clé de couverture

coefficients DCT des 2 fréquences doivent avoir même valeur dans table de quantification et être de fréquence moyenne

Comment coder ?

I-ème bit du message = 0, si $DCT(u_{i1}) < DCT(u_{i2})$

I-ème bit du message = 1, si $DCT(u_{i1}) > DCT(u_{i2})$

en cas de besoin échange des coefficients

# Choix aléatoire des pixels à changer

Cryptology & Security Initiative

générateur pseudo-aléatoire

(8, 20) (1,4) (5, 16) (56, 59) (587, 122) …

message secret

| 0 | 1 | … |
|---|---|---|

semence du générateur
=
clé de couverture

| | | | | | |
|---|---|---|---|---|---|
| 153 | 53 | 2 | 159 | 216 | 62 |
| 27 | 78 | 82 | 234 | 48 | 16 |
| 63 | 61 | 212 | 216 | 48 | 39 |
| 254 | | | | | 56 |
| 3 | | | | | 59 |
| 38 | | | | | 48 |
| 52 | | | | | 26 |
| 76 | 99 | 196 | 33 | 38 | 2 |
| 45 | 102 | 79 | 21 | 156 | 64 |
| 19 | 182 | 7 | 48 | 178 | 6 |

Avantage :
méthode plus robuste !!
message secret caché
à travers toute la
couverture

1. On génère un couple de nombres
2. On compare les pixels correspondants et on les oublie s'ils ne sont pas « proches »
3. On code un 0 par le fait que le premier des pixels est plus petit que le deuxième (on les échange en cas de besoin)
4. On code un 1 par le fait que le premier des pixels est plus grand que le deuxième (on les échange en cas de besoin)

# Détection

Cryptology & Security Initiative

La méthode basée sur la discontinuité aux frontières des blocs reste applicable dans le cas des fichiers jpeg.

***détection***

Une autre direction de recherche envisagée est l'utilisation de techniques d'intelligence artificielle et de reconnaissance optique pour "apprendre" à un logiciel à reconnaître les stéganogrammes des messages sans contenu dissimulé.