

STEGANO

Steganography = 'verborgen schrijven'.

Met deze techniek kan 'n bestand (het 'geheim') worden opgenomen in 'n ander bestand (de 'drager'). We doen dit door elke bit (!) van 'geheim' te plaatsen in de **LSB** van een byte in de 'drager'.

Als we voor de drager een plaatje (png, gif, etc.) dan zal de verandering in het plaatje voor het oog vrijwel niet waarneembaar zijn. (evenzo voor audio en video bestanden.)

Bovendien zijn de plaatjes vaak groot genoeg voor een lange tekst.

Stel we hebben een plaatje van 3 kleuren (RGB) en 1600 x 900 pixels dat zijn er $3 \times 1600 \times 900 = 4.320.000$ pixels. Dat is genoeg ruimte voor 540.000 bytes oftewel 540 kB.

(met 80 tekens op een regel is dit 6750 regels, dus ongeveer 100 pagina's tekst)

Als je een kleinere tekst in een dergelijke drager wilt opnemen, dan kun je een (willekeurig) patroon gebruiken om bepaalde bytes over te slaan. Dit maakt detectie, laat staan lezen, vrijwel onmogelijk.

Bedenk ook dat slechts de helft van de (gebruikte) bytes in de drager gewijzigd wordt!

Deze techniek wordt 'enthousiast' gebruikt op het internet!

We gaan hier een versimpelde versie van deze techniek implementeren.

Voor geheim en drager gebruiken we een tekst bestand. (niet ideaal, wel eenvoudiger te implementeren)

Codeer instructies :

Voor deze opdracht is een random generator gemaakt in een aparte klasse. (bestanden generator.h en generator.cpp). Deze code bestanden zul je moeten 'mee compileren'.

1. De functie **maakRegel** moet een string van opgegeven lengte geven. Deze string moet gevuld zijn met willekeurige tekens uit 'alfabet'. Gebruik hiervoor de generator. Deze is al gemaakt als globale variabele in de 1^e regel code van `stegano.cpp`.
De methode `generator()` geeft een random getal tussen 0 en 25 (inclusief de grenzen)
Deze regel fungeert als drager voor de 'geheime' tekst.
2. De functie `schrijf` moet voor elke bit van elk character in geheim, deze bit plaatsen in de LSB van de characters in de regel.
Gebruik hiervoor de bitwise operatoren **&=** en **|=** met de juiste masks.
3. De functie `leest` van elk character in de aangeboden string de LSB en combineert deze tot een character. (elke 8 bits → 1 char) Ook hier zijn de bitwise operatoren nodig. Ook de shift left operator moet hier gebruikt worden.
Bovendien moet je goed tellen wanneer de byte compleet is.

Zie ook: **mask.pdf**

STEGANO

Gegeven is de volgende code :

stegano.cpp:

```
#include <iostream>
#include <sstream>
#include <string>
#include "generator.h"
#include "stegano.h"

generator gen(0,25);
std::string alfabet = "abcdefghijklmnopqrstuvwxyz";

... hier de functies van stegano.h ...

int main()
{
    std::string geheim = "Denkend aan Holland";
    std::string regel = maakRegel(geheim.length() * 8);
    schrijf(regel,geheim);
    std::string ontvangst = lees(regel);
    std::cout << ontvangst << std::endl;
    return 0;
}
```

Merk op:

1. De inhoud van de '**main**' routine is niet van belang voor de test.
Je mag hier test-code opnemen voor de gemaakte functies, geheel naar eigen inzicht.
2. De 'signatuur' van de functies mag niet worden aangepast. Wijzigingen geven fouten in de test.

Upload instructies:

Upload de volgende bestanden:

1. stegano.cpp