Istituto Statale d'Istruzione Superiore
"Arturo Malignani" - Udine

Esami di Stato 2017/2018
Classe 5 TEL B


# La Steganografia

RICCARDO MARIN

13 giugno 2018

Documento prodotto con LaTeX e basato sul template ArsClassica di Lorenzo Pantieri.

# INTRODUZIONE

La rapida evoluzione tecnologica di Internet e dei suoi servizi hanno reso anche l'informazione un bene primario, con suo valore intrinseco, e di conseguenza è cresciuta allo stesso tempo l'importanza di nasconderla o proteggerla; da ciò nasce l'interesse per l'insieme delle tecniche di sicurezza informatica atte a mantenere la riservatezza dei dati, come la crittografia e la steganografia, l'argomento approfondito nella prima parte dell'elaborato. Se queste tecniche venissero violate, ne conseguirebbero gravi conseguenze per la nostra privacy.
Tali strategie vengono anche applicate in contesti dove si cerca di sfuggire ad controllo oppressivo e autoritario; questa questione viene trattata nella seconda parte della tesina, con particolare riferimento a come politica, libertà e censura stanno influenzando la Cina adesso e lo faranno nei prossimi anni.

# INDICE


1 STEGANOGRAFIA 2
- 1.1 Introduzione 2
- 1.2 Servizi 2
- 1.3 Cenni storici 2
- 1.4 Crittografia vs steganografia 4
- 1.5 Modelli steganografici 4
- 1.6 Tipi di steganografia 6
  - 1.6.1 Testo 6
  - 1.6.2 Audio 6
  - 1.6.3 Immagini 7
- 1.7 Applicazioni 8
  - 1.7.1 Watermarking e fingerprinting 8
  - 1.7.2 Terrorismo 8
  - 1.7.3 Libertà di parola 9
- 1.8 Steganalisi 9
- 1.9 StegoMalignani 11
  - 1.9.1 Descrizione 11
  - 1.9.2 Criteri di segretezza 11
  - 1.9.3 Codice sorgente 12

2 THE GREAT FIREWALL OF CHINA 19
- 2.1 Brief History 19
- 2.2 The Great Firewall 20
  - 2.2.1 How does a firewall work? 21
- 2.3 Censorship techniques 21
  - 2.3.1 Forbidden websites and services 22
- 2.4 Circumvention 24
  - 2.4.1 VPN 24
  - 2.4.2 Proxy 24
- 2.5 Reasons 25
  - 2.5.1 Social impact 25
  - 2.5.2 Economical impact 25
  - 2.5.3 Foreign reactions 25
- 2.6 Conclusions and future prospects 25

Bibliografia 27

# 1 STEGANOGRAFIA

## 1.1 INTRODUZIONE

La parola steganografia deriva dal greco, dall'unione del vocabolo *steganos*, ovvero segreto e il termine *graphein*, cioè scrivere. Si tratta dunque di scrittura nascosta, nata nell'antichità tuttavia è in crescita attraverso i mezzi digitali. Come la crittografia, è un modo di proteggere la confidenzialità di dati, ma è molto meno conosciuta.

## 1.2 SERVIZI

Prima dettagliare il suo funzionamento, è bene ricordare che il principale settore di utilizzo della steganografia è la sicurezza delle informazioni. Quest'ultima deve garantire 5 servizi:

- Confidenzialità, garantisce l'accesso o la leggibilità dell'informazione solo da chi è autorizzato a farlo;
- Integrità, garantisce che i dati siano modificati solamente da chi li abbia inviati;
- Autenticazione, non deve poter essere modificata l'identità di chi invia il messaggio;
- Non ripudiazione, garantisce che nè il mittente nè il destinatario possano rifiutare/negare di aver mandato/ricevuto;
- Identificazione, garantisce l'accesso ad un sistema soltanto agli utenti autorizzati a farlo

Inoltre Cisco ha aggiunto di recente un sesto servizio, la disponibilità di connessione, cioè l'utente deve poter connettersi e in modo sicuro. Tale esigenza è emersa per soddisfare il crescente utilizzo del proprio dispositivo (logica del *bring your own device - BYOD*).

## 1.3 CENNI STORICI

I primi usi della steganografia risalgono all'antica Grecia: lo storico Erodoto riporta che un generale fece rasare i capelli ad un messaggero, gli fece tatuare il messaggio sul cranio e attese la ricrescita prima di inviarlo. Un altro metodo (piuttosto superato) era quello di scrivere sotto la cera di una tavoletta scrittoria.
Nel XVI secolo si fece uso anche delle **griglie di Cardano**, dei fogli di carta

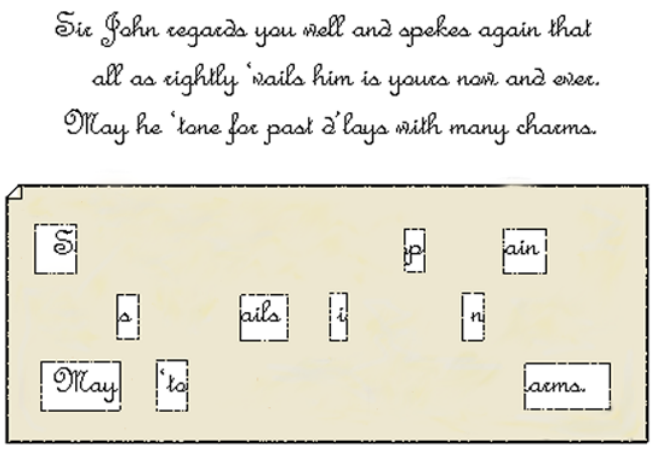


**Figura 1:** Esempio di griglia di Cardano

con dei rettangoli ritagliati in precise posizioni. Applicando la griglia sopra un foglio bianco, il messaggio veniva scritto in corrispondenza dei buchi. Successivamente, si toglieva la griglia e sul foglio si cercava di completare la scrittura del testo attorno alle lettere del messaggio segreto in modo da ottenere un messaggio di senso compiuto. Il destinatario poteva ricostruire il messaggio applicando la stessa griglia al foglio, filtrando il messaggio nascosto.

Fino al diciottesimo secolo anche gli **inchiostri invisibili** ebbero un ruolo non trascurabile nelle scritture segrete. Solitamente la scrittura avveniva con sostanze invisibili dopo l'applicazione rese di nuovo visibili dal destinatario, per esempio se sottoposte a fonti calore.

Più moderno è invece l'uso dell'**acrostico** o "null cipher", consiste nel nascondere un testo in un altro, in modo che possa essere estratto selezionando solo alcuni caratteri. Il seguente messaggio è stato realmente inviato da una spia tedesca durante la seconda guerra mondiale:

*Apparently neutral's protest is thoroughly*
*discounted and ignored. Isman hard hit.*
*Blockade issue affects pretext for embargo on*
*by products, ejecting suets and vegetable oils.*

prendendo solo la seconda lettera di ogni parola diventa:

*Pershing sails from NY (r) June 1*

Un particolare e controverso esempio di steganografia è quello dei **micropunti delle stampanti** (fenomeno conosciuto anche come *Machine Identification Code* o *yellow dots*). Molte delle comuni stampanti a laser dei puntini gialli di dimensioni minuscole, quasi impossibili da individuare a occhio nudo, disposti in modo da codificare certe informazioni come la data e l'ora di stampa e il numero di serie del dispositivo. L'insieme di questi dati permette alle autorità di rintracciare l'autore del documento.

E' stato fortemente criticato dai difensori della privacy e dall'EFF (Electronic Frontier Foundation).

## 1.4 CRITTOGRAFIA VS STEGANOGRAFIA

Entrambe perseguono come obiettivo la sicurezza di una comunicazione; tuttavia, mentre la crittografia rende il messaggio incomprensibile a chi non è autorizzato, la steganografia cerca di rendere invisibile la comunicazione stessa.
Per esempio l'intercettazione del messaggio crittografato di Figura 2 lascia pochi dubbi sul fatto che qualcuno stia cercando di comunicare in modo sicuro.
Ne consegue che il vantaggio della steganografia è l'occultamento dell'atto

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2.0.22 (MingW32)
Comment: Using GnuPG with Thunderbird - http://www.enigmail.net/

hQEMA7+WBg1Qk3uhAQf/TtcTV0FAyAynSi+n7SYSJf7so7G2LQdZHKjIr4cuzvNY
LhlwEeFPyEnkXLGZpQi1whvVM1NGLhksOTKm0zNkeBFUzcYHcqtkpxOKT/8aj/J6
rVlboKlNKVGIjlcGMrqxYaH6vi/WzjZtZiSJ0RrN+VzXTco0sTf7XcPpgfAlNcMW
GjxsDAC0zfq2p05QKJ+neC/fT10cPD5FcmIe+SShld3ESr+PgNVpGOsYRWT4WO9v
r0oRy1UV6BZaGI1FkLuByNGIApZQvganzWnBU3AP02srVIbmLIHj4vVVqhUCXusE
vzjPw9PV7REugk8ZBTrqojyYfXca8IPRtsWe9U4GGtLpAWHN3FlWZRjm2GszQQBy
lLv/qrVBy9rouT6LhBW6xkhBLa5bzgpLBF0q0VS1WvGKFnYtxuomFuIJsYggWn4+
++2QRN/O5LfF76qefUrW1iqxUp/HonVB8OmZjJ+DOgWxc1TUocnIccrJ4oSeDOHg
9qOrwME2oYbtD/FeT9W3gewwmV/tlDCejcbtW6mvZox+XHMnwJO2s3XIjQyukjlx
KXNI7Uiqx2Z7A3XbuxDSHJQt6iEjqjQll+Wyt370Rwf6uhzPCI1yYi37vnyecEXI
ZZJodwTtuRjJpC9BbIDOjQwrAfXej/UeLcycrswnru/TDilMoab527P55PiuHUIt
N4MQlOWDh278rTr8LEe9l4SXie/hgcudzTmqvVFced5HecvEKvb+8UOGvJlnC4YT
I0UA+JephYvqw7HHCdOP4ZY3/mISjBhM2OICLHuBP6i3EslW2/1JGP0W5m5z21Hu
VrQNhnw/QwshmmGm5K+wBu1DVRIXYvGSrVKIRbOi0VtZPvEABy54kILBcceNYlAN
rQCnHXI4fKYwpnmxJ3tThr6KbeCykbDhHQsdlEDM9g2Wq02qcHVj87MDUEShWYUM
uO+o+SInjgg0Fywhq4HR9dDHVt48XNh4eMYd/OjfGOnioLVrQ6zilb48bb57eoFO
cV/Xf16u1fSbY/5lcC27JAcufxzVbracnHpSR3T0yvOy7DoaJolsh5+52NaQjJVa
P/JXihj0OdVVmaf+1pV1yA==
=LPo0
-----END PGP MESSAGE-----
```

**Figura 2:** Esempio di messaggio cifrato con Pretty Good Privacy (PGP)

comunicativo in sè, ma contrariamente a quanto accade con la crittografia, l'intercettazione mette a il rischio il messaggio. L'unione di queste due tecniche, per esempio cifrando il testo prima di nasconderlo in un'immagine, può aumentare significativamente il livello di sicurezza.

## 1.5 MODELLI STEGANOGRAFICI

Lo schema di base della steganografia prevede due ruoli: un messaggio segreto e un messaggio contenitore.
Fondamentalmente esistono due approcci alla steganografia:

- Iniettiva, La steganografia iniettiva è la più utilizzata e modifica un contenitore in modo tale da imprimere un messaggio al suo interno, senza cambiare le sue dimensioni, rendendolo praticamente indistinguibile dall'originale.
  Un approccio possibile è quello di sfruttare il rumore - già presente nel contenitore - per sostituirlo con un segnale.

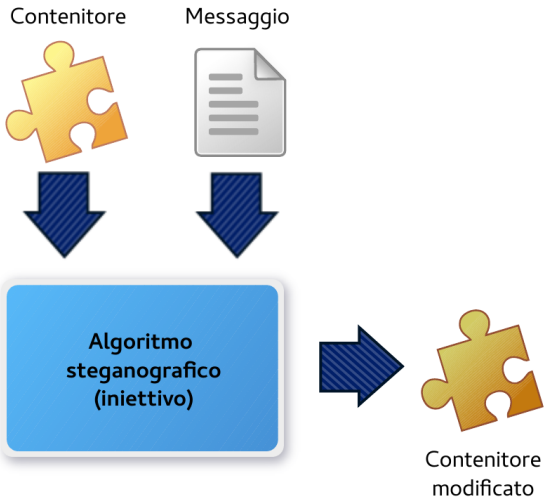


**Figura 3:** Steganografia iniettiva

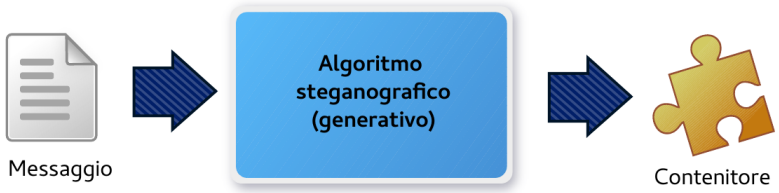


**Figura 4:** Steganografia generativa

Il campo di applicazioni è molto vasto e coincide con quello dei media digitali: immagini, testi, audio o anche video possono diventare mezzo di comunicazione segreto.
La steganografia iniettiva è ritenuta più sicura rispetto a quella generativa, in quanto le dimensioni del contenitore rimangono inalterate dopo l'inserimento dell'informazione segreta.

- Generativa, La steganografia generativa consente di generare a partire dal messaggio segreto un messaggio contenitore che nasconde al suo interno il messaggio segreto.

Per esempio SpamMimic è un servizio online che permette di creare un finto messaggio di spam, in cui in realtà è nascosto il messaggio segreto. Il messaggio può essere inviato per e-mail: un eventuale attaccante passivo lo scambierà per un comune messaggio di spam, senza sospettare che nasconde un messaggio segreto Si tratta di un approccio generativo perché il contenitore (il messaggio di spam) viene generato ad hoc per contenere il messaggio desiderato. Presenta come vantaggio l'impossibilità di confrontare il contenitore con un eventuale contenitore originale, tuttavia questo viene costruito su misura, rendendolo più facilmente individuabile.

## 1.6 TIPI DI STEGANOGRAFIA

La steganografia dei secoli scorsi era principalmente di tipo fisico, e quindi riguardava la scrittura di messaggi con inchiostri invisibili o altri metodi come quello citato in precedenza. Adesso si è spostata da mezzi fisici al digitale, estendendosi a file multimediali ma anche a file system o header di pacchetti TCP/IP.

### 1.6.1 Testo

E' possibile codificare informazioni anche negli spazi di un testo (per esempio con il programma Snow) oppure, anche se è meno sicuro, nelle iniziali, come l'acrostico, citato in precedenza. Anche Spam Mimic, rientra tra i programmi di steganografia di tipo testuale.
Tuttavia, nella steganografia rimane maggiore interesse nei nuovi media digitali, come l'audio e le immagini, che restano i mezzi preferiti per veicolare informazioni nascoste.

### 1.6.2 Audio

Esistono diversi approcci alla steganografia audio. I principali sono:

- Codifica LSB, raggruppa il flusso audio in blocchi da 16 bit e altera il bit meno significativo;

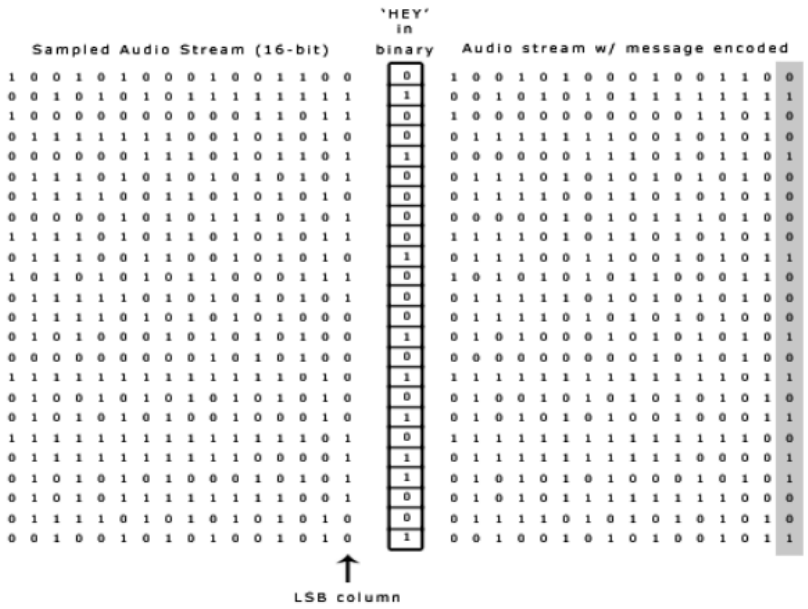


**Figura 5**: Codifica audio LSB

- Codifica di parità, come il precedente, ma agisce sui bit di parità;
- Codifica di fase, divide l'audio in segmenti, di cui cambia la fase iniziale. Una possibile codifica si ottiene associato allo 0 uno sfasamento pari a $\pi/2$ e $-\pi/2$ in corrispondenza di un 1.

### 1.6.3 Immagini

L'inserimento di dati nelle immagini può avvenire in due modalità: nel dominio spaziale e nel dominio delle trasformate.
Nel primo caso modificano direttamente i valori dei pixel, nel secondo trasformano l'immagine sotto forma di serie di coefficienti con cui rappresentare l'immagine.

#### *Dominio spaziale*

Un'immagine digitale può essere definita come una serie di valori digitali, detti pixel. I pixel l'unità elementare dell'immagine e rappresentano il valore di un certo colore in un punto. Possiamo pensare l'immagine come una matrice di pixel. Per descrivere l'intensità dei colori di un pixel solitamente si usa il modello RGB, dove tre numeri (da 0 a 255) indicanti il rosso, il verde e il blu sono combinati assieme per ottenere un'ampia gamma di colori. E' possibile modificare questi valori per salvare delle informazioni, in seguito l'esempio pratico del programma StegoMalignani, che si basa proprio su questo.
Tale modo di operare prende il nome di LSB (Least Significant Bit) con

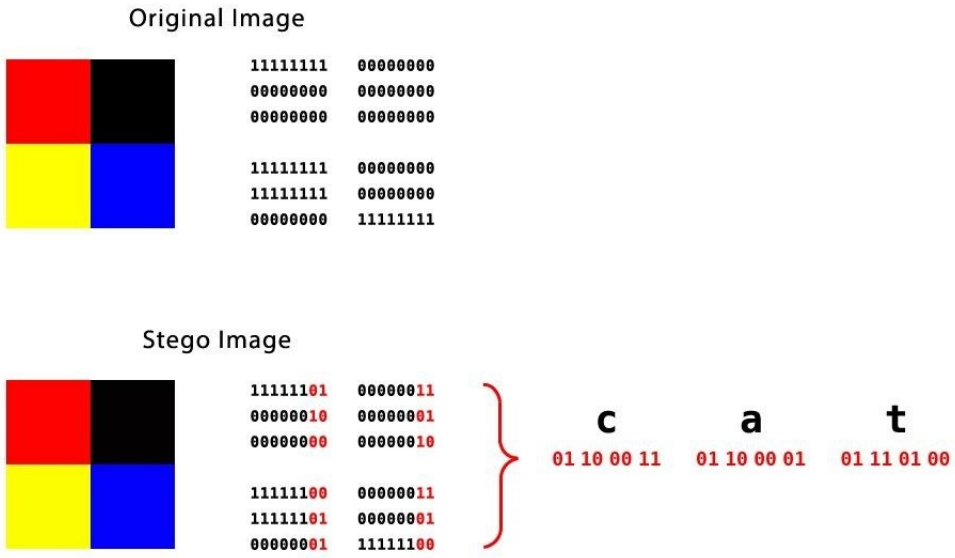


**Figura 6:** Alterazione dei bit meno significativi di un'immagine

riferimento alla modifica del bit meno significativo.
Come si vedrà in seguito, è piuttosto vulnerabile alla steganalisi.

#### *Dominio delle trasformate*

In alternativa alla facilmente individuabile tecnica LSB, nascondere dati nel dominio delle trasformate permette maggiore robustezza agli attacchi. Con questo approccio l'immagine viene scomposta come serie di coefficienti, che vengono poi alterati.

Per esempio, apparemente nelle immagini in formato JPEG non si potrebbero nascondere informazioni nei bit ridondanti, proprio perchè si tratta di un formato *lossy*, che elimina i dati superflui per comprimere le dimensioni dell'immagine. L'algoritmo di compressione JPEG usa la Discrete Cosine Transform (DCT), raggruppando blocchi di 8 x 8 pixel e trasformandoli in 64 coefficienti DCT. Seguendo i principi della codifica LSB, il messaggio può essere incorporato nei bit meno significativi dei coefficienti.

Esiste anche la steganografia video, analoga a quella delle immagini, ma con una banda passante molto più ampia.

## 1.7 APPLICAZIONI

L'obiettivo che si pone la steganografia è quello di nascondere la presenza di una comunicazione segreta tra due o più parti. Nei secoli scorsi ne facevano utilizzo principalmente i governi per inviare messaggi segreti, spesso di tipo militare. Anche oggi viene utilizzata da servizi di intelligence per effettuare comunicazioni riservate. Nel campo del commercio elettronico, la steganografia potrebbe essere utilizzata nell'implementazione di sistemi sicuri di pagamento on-line o in ambito medico per contenere le informazioni sensibili dei pazienti.

### 1.7.1 Watermarking e fingerprinting

Altre applicazioni moderne della steganografia riguardano il diritto d'autore: con il watermarking si applica un logo o una scritta ad un'immagine, in questo modo rimane un marchio del creatore.
In questo modo se un'immagine viene rubata, l'autore può rivendicarne la proprietà.
Il fingerprinting serve invece a identificare il singolo utilizzatore; serve a risalire all'autore iniziale di una fuga di informazioni.

### 1.7.2 Terrorismo

Possiamo dire che le potenzialità di questa tecnica sono molto ampie, fornendo anche libertà di espressione agli attivisti, ma non è da escludere l'abuso da parte di chi usa questo strumento per altri scopi.
Già dall'inizio del secolo la steganografia è stata usata da organizzazioni terroristiche; Osama Bin Laden ne ha fatto uso per comunicazioni all'interno della cellula terroristica, attraverso le immagini presenti nelle inserzioni di eBay.
Considerando che ogni sito web presenta degli elementi grafici e l'immensa quantità di immagini in circolazione, l'individuazione di tali messaggi risulta praticamente impossibile.

### 1.7.3 Libertà di parola

Oggigiorno - oltre a quest'ultimo - si possono identificare due grandi gruppi di utilizzatori: chi cerca di occultare delle informazioni legate a qualche tipo di reato, per la maggior parte organizzazioni terroristiche e chi è costretto trovare modi alternativi di aggirare leggi draconiane che vanno a minacciare la libertà di parola o di espressione.

## 1.8 STEGANALISI

Si definisce steganalisi l'insieme delle tecniche e dei metodi per attaccare un sistema steganografico, perciò volte a rivelare la presenza di un dato nascosto.
Esistono diverse tipologie di attacco in base alle informazioni possedute:

- **Attacco stego-only**
  Si possiedono solo i dati da analizzare, cioè si può analizzare solo lo *stego object*. In questo caso si può assumere un modello di rumore plausibile per il tipo di file in esame tentando di ottenere una percentuale di falsi positivi medi. Si tratta dell'avvenimento più comune, oltre che di quello con minore informazione possibile.

- **Attacco known-cover**
  Si posseggono il dato originale ed il dato alterato. Si tratta dell'eventualità più fortunata, perché analizzando le differenze tra i due, non è difficile intuire l'algoritmo di steganografia che è stato utilizzato.

- **Attacco known-message**
  Si possiedono sia il messaggio segreto sia il contenitore. Analizzando il rapporto tra i bit del messaggio e quelli del contenitore possono essere scovate le relazioni che li legano. Le stesse relazioni possono essere ricercate in altri *stego object* per verificarne la natura.

- **Attacco known-decoding (chosen stego)**
  Si conosce l'algoritmo usato per estrarre il messaggio segreto. Teoricamente, è possibile utilizzare un attacco brute force per estrarre il messaggio dal mezzo steganografico.

- **Attacco known-encoding (chosen message)**
  Si è in possesso dell'algoritmo usato per inserire il messaggio segreto. È possibile studiare gli effetti che producono vari messaggi su potenziali cover, semplicemente realizzando un sufficiente numero di nuovi contenitori di prova. Se questi stessi effetti vengono riscontrati sul mezzo in esame, quest'ultimo risulterà sospetto.

In rete è possibile trovare diversi software che mirano proprio alla steganalisi.

Per quanto riguarda gli attacchi alle immagini i principali metodi sono l'attacco visuale e l'attacco statistico.

Il primo si basa sulle capacità visive umane per individuare artefatti nell'immagine. Il file viene prima filtrato con un algoritmo di filtering e il risultato viene osservato per determinare se è stato nascosto un messaggio o meno.

L'attacco statistico, invece effettuata dei test di tipo statistico: l'analisi si

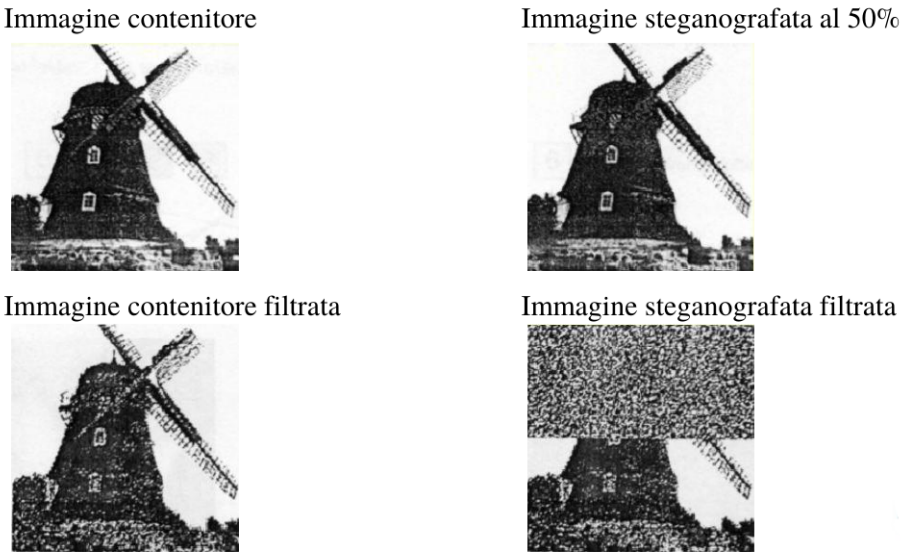


**Figura 7:** Attacco visuale: algoritmo di filtering applicato ad una immagine

fonda sulla distribuzione del colore nell'immagine. Più precisamente, viene effettuato un confronto tra la distribuzione di frequenza dei colori del file sospetto con quella teoricamente attesa.

In caso di messaggi nascosti nell'immagine, l'istogramma evidenzierà un certo grado di equalizzazione nella distribuzione dei colori, la quale presenta maggiore varianza nella sua versione originale. Nel caso di una codifica

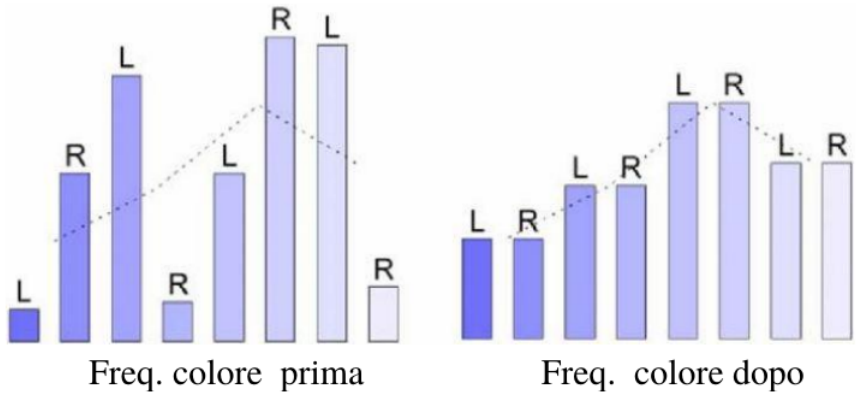


**Figura 8:** Attacco statistico: si basa sul fatto che il numero di pixel con colori adiacenti (differiscono di solo un bit) è molto più alto in un immagine con un contenuto incorporato. L e R stanno ad indicare rispettivamente l'elemento a sinistra e a destra della coppia di colori adiacenti considerata.

LSB le frequenza risulterebbero simili a quelle attese soltanto se i bit da

sovrascrivere fossero equalmente distribuiti.

## 1.9 STEGOMALIGNANI

### 1.9.1 Descrizione

StegoMalignani è un programma per la steganografia iniettiva sostitutiva di immagini BMP, che sfrutta il bit meno significativo (LSB) del blu di ogni pixel.
Per garantire maggiore sicurezza unisce steganografia e crittografia: al momento della codifica, il testo in chiaro viene trasformato in testo cifrato derivante da un chiave scelta dall'utente e successivamente incorporato nell'immagine.
Un possibile futuro miglioramento è la scrittura su pixel in posizioni pseudocasuali nell'immagine, al posto dell'approccio sequenziale, che risulta individuabile dalla steganalisi. In questo modo, i bit alterati sarebbero distribuiti in modo più uniforme e facilmente confondibili con del rumore.

#### *Spazio steganografico in stegomalignani*

Il programma Stegomalignani permette di salvare una discreta quantità di dati testuali. Siccome utilizza solo il bit meno significativo del blu, ogni pixel potrà contenere un bit. Ad 8 pixel (8 bit) equivale un carattere ASCII. Una stima dello spazio di memoria fornito da un'immagine BMP codificata con StegoMalignani è la pari al numero di pixel dell'immagine stessa; perciò numero pixel della base x numero pixel dell'altezza. Ad esempio una foto di 1024 x 768 pixel = 294.912 byte = 288 KByte, circa poco più di metà Divina Commedia.
Se si usassero i LSB dei tre colori RGB, la capacità verrebbe triplicata; tuttavia le modifiche all'immagine, essendo più marcate, sarebbero anche più facilmente individuabili.

### 1.9.2 Criteri di segretezza

La sicurezza delle informazioni in StegoMalignani, ma anche più in generale quando si applica la steganografia sulle immagini, può essere migliorata seguendo dei parametri. I principali sono:

- *Genericità*, il messaggio contenitore deve essere più generico possibile, in modo da non destare sospetti.
- *Diffusione*, nessuno deve possedere il messaggio contenitore, per evitare confronti.
- *Unicità*, il messaggio contenitore va usato una sola volta; è buona pratica distruggerlo dopo l'invio e non riutilizzarlo.

E' anche importante che l'immagine usata sia originale, se esistesse una copia a disposizione pubblicamente o su internet, da un confronto emergerebbe subito la presenza di anomalie.

### 1.9.3 Codice sorgente

```
/*#####################################################
  ###### NOME: StegoMalignani
  ###### AUTORE: Riccardo Marin
  ###### ULTIMO AGGIORNAMENTO (v 1.2): 30/04/18 10:00
  ###### DESCRIZIONE: Programma per permette di scrivere e
  ###### leggere messaggi sulle immagini in formato bmp
  ###### sfruttando il bit meno significativo del blu
  ###### di ogni pixel.
  #####################################################*/

#include <cstdlib> // libreria standard
#include <iostream> // libreria standard
#include <stdio.h> // libreria standard
#include <fstream> // per input/output file
#include <string>  // per manipolare stringhe
#include "EasyBMP.h" // per modificare immagini
#define numVers "1.2" // numero versione attuale

using namespace std;

void mostraUtilizzo() // spiegazione l'utilizzo del programma
{
    cout << "Utilizzo: ./StegoMalignani -comando percorsoImmagineBMP\n\
        n";
    cout << "\t -r/--rivela  Comando per rivelare dati nascosti nell'immagine
        \n";
    cout << "\t -n/--nascondi Comando per nascondere dati nell'immagine\
        n";
    cout << "\t -f/--file    Per specificare un file da/verso cui direzionare i
         dati (opzionale)\n";
}

void cripta (string &testo, string &chiave, string &cifrato)
    { // realizza cifratura vigenere con xor
      string chiaveEstesa = testo;

      for(int i = 0; i < testo.length(); i++) { // ciclo per
         tutta la lunghezza del testo
         chiaveEstesa.at(i) = chiave.at(i\%(chiave.length()));
             // chiave ripetuta se lunghezza testo > lunghezza
             chiave
         // quando avviene xor tra valori uguali, ottengo 0,
            che verrebbe interpretato come fine stringa
         // tramite un artificio si puo ovviare questo
            problema:
         cifrato.at(i) = ((testo.at(i)-'o') ^ (chiaveEstesa.at(
            i)-'o')) + 'o';
       }
}

void nascondi(BMP &img, char percorso[], bool &usaFile, char
    percorsoFile[])
{
```

```
img.ReadFromFile(percorso);

// INPUT MESSAGGIO DA CRIPTARE
string testo;      // testo completo da nascondere
string buffer;     // contenitore parziale/temporaneo
string chiave;     // chiave cifratura del testo
string cifrato;    // testo cifrato

if(usaFile == true) // input da file di testo
{
  ifstream fileTesto; // apro il file
  fileTesto.open(percorsoFile);
  while (getline (fileTesto,buffer)) // salvo una riga
     alla volta fino alla fine del file
  {
     testo = testo + buffer + "\n";
     buffer = "";
  }
  fileTesto.close(); // chiudo il file
  cout << "Input ricevuto dal file\n";
}

else
{
    cout << "Inserisci il testo da nascondere, scrivi \"<end>\" per
        terminare:\n";
    cin >> buffer;
    do
    {
        testo = testo + buffer + "\n";
        buffer = "";
        cin >> buffer;
    } while(buffer != "<end>");
 }

cout << "Inserisci la chiave di cifratura:\n";
cin >> chiave;

cifrato = testo; // stringa cifrata della stessa
   lunghezza del testo in chiaro

cripta(testo, chiave, cifrato);

// SCRIVO SULL'IMMAGINE IL MESSAGGIO

int x=0;       // coordinata x del pixel
int y=0;       // coordinata y del pixel

int altezza=img.TellHeight(); // salvo altezza immagine
int larghezza=img.TellWidth(); // salvo larghezza
   immagine

for(int i=0;i<cifrato.length();i++) // scorro tutto il
   testo
{
```

```
        char carattere = cifrato.at(i); // prendo un carattere
            alla volta

        for(int j=0;j<8;j++)
        {
            int blu = (int) img(x,y)->Blue; // estraggo il
               valore del blu

            blu &= ~1;            // conservo 7 bit e azzero
               il meno significativo
            blu |= (carattere >> j) & 1; // metto in OR il
               risultato con il bit da scrivere

            img(x,y)->Blue = blu; // sovrascrivo il valore blu

            x++;                  // aggiorno coordinate (mi
               sposto a destra)

            if(x==larghezza)      // vado a capo (sono
               arrivato all'ultimo pixel in larghezza)
            {
               x=0;               // riparto dalla prima
                  colonna
               y++;               // linea successiva
            }
        }
    }

    // INSERISCO IL FINE STRINGA

    for(int j=0;j<8;j++)          // un carattere = 8 bit
    {
       int blu = (int) img(x,y)->Blue; // estraggo il valore
          del blu

       blu &= ~1;                 // valore /0 = 00000000

       img(x,y)->Blue = blu;      // sovrascrivo il valore blu

       x++;                       // aggiorno coordinate (mi
          sposto a destra)

       if(x==larghezza)           // vado a capo (sono
          arrivato all'ultimo pixel in larghezza)
       {
            x=0;                  // riparto dalla prima
               colonna
            y++;                  // linea successiva
       }
    }

    img.WriteToFile(percorso);    // output immagine
}
```

```
void rivela(BMP &img, char percorso[], bool &usaFile, char
    percorsoFile[])
{
     img.ReadFromFile(percorso);

     char carattere = '\0'; // singolo carattere decifrato,
        valore \0 = 00000000
     string testo = "";  // azzero inizialmente il testo
     string chiave;
     string decifrato;

     int x=0;       // coordinata x del pixel
     int y=0;       // coordinata y del pixel

     int altezza=img.TellHeight(); // salvo altezza immagine
     int larghezza=img.TellWidth(); // salvo larghezza
        immagine

     do  // scorro tutto il testo fino a che non trovo il
        fine stringa (vedi condizione while)
     {
        for(int j=0;j<8;j++)        // 8 bit = 1 byte = 1
           carattere
        {
            int blu = (int) img(x,y)->Blue; // estraggo il
               valore del blu

            carattere = carattere << 1; // shifto a sinistra
            carattere |= blu & 1;   // isolo l'ultimo bit

            x++;                    // aggiorno coordinate (mi
                sposto a destra nei pixel)

            if(x==larghezza)        // vado a capo (sono
               arrivato all'ultimo pixel in larghezza)
            {
               x=0;                 // riparto dalla prima
                  colonna
               y++;                 // linea successiva
            }
         }

        // inverto ordine bit
        carattere = (carattere & 0xF0) >> 4 | (carattere & 0
           x0F) << 4;
        carattere = (carattere & 0xCC) >> 2 | (carattere & 0
           x33) << 2;
        carattere = (carattere & 0xAA) >> 1 | (carattere & 0
           x55) << 1;

        testo += carattere;         // aggiungo il carattere
           in coda al testo da decifrare

     } while(carattere != '\0');     // in scrittura ogni
        messaggio ha un fine stringa: \0, ovvero 00000000
```

```
    cout << "Inserire chiave di cifratura:\n";
    cin >> chiave;

    decifrato = testo; // stringa in chiaro della stessa
       lunghezza del testo cifrato

    cripta(testo, chiave, decifrato);

    for(int i = 0; i < decifrato.length(); i++) // rimozione
       degli a capo
    {
       if (decifrato.at(i) == '\n') decifrato.at(i) = '';
    }

    decifrato.at(decifrato.length()-1) = ''; // rimozione del
        fine stringa

    // COMUNICO ALL'UTENTE IL RISULTATO

    if(usaFile == true) // nel file di testo
    {
      // apro il file
      ofstream fileTesto;
      fileTesto.open(percorsoFile);
      fileTesto << decifrato; // testo che ho decriptato, lo
          salvo nel file
      fileTesto.close(); // chiudo il file
      cout << "Output inserito nel file.\n";
    }

    else cout << "Messaggio letto: " << decifrato; // oppure
       direttamente nel terminale
}

int main(int argc, char *argv[])
{
      BMP img;
      string percorsoImg = "";
      bool usaFile = false;
      string strPercorsoFile;
      char percorsoFile[20];
      char percorso[20];

      // PARSING PARAMETRI AVVIO

      if (argc < 2 || argc == 4 || argc > 5) // numero di
         parametri insufficiente o eccessivo
      {
        cout << "Numero argomenti errato.\n";
        mostraUtilizzo();
        return 1;
      }
```

```
if ((argv[1] == string("-h")) || (argv[1] == string("--
    help"))) // help, che mostra versione e autore, oltre
     all'utilizzo
{
  mostraUtilizzo();
  cout << "\nVersione " << numVers << " | Riccardo Marin |
      riccardomarin23@gmail.com | 0
      x264834Ebf2Ac311429cFb66C587734Ea742586D5\n";
  return 0;
}

else if( (argv[1] != string("-r")) && (argv[1] != string
    ("--rivela")) && (argv[1] != string("-n")) && (argv[1]
     != string("--nascondi")) ) // l'utente digita un
    comando inesistente
{
    cout << "La tua richiesta non e' chiara.\n";
    mostraUtilizzo();
    return 1;
}

if (argc == 2)    // non viene inserito il percorso
    immagine
{
  cout << "Devi specificare il percorso dell'immagine.\n";
  cout << "[Esempio: ./StegoMalignani -n areaTest/desktop.bmp]\n";

  return 1;
}

else if (argc == 5) // vengono utilizzati i file di
    testo (-f file.txt)
{
   if ((argv[3] == string("-f")) || (argv[3] == string("
       --file")))
   {
      strPercorsoFile = argv[4];
      // fstream vuole il percorso come vettore di
          caratteri
      strcpy (percorsoFile,strPercorsoFile.c_str()); //
           converto stringa in array di char

      // apro il file
      fstream fileTesto;
      fileTesto.open(percorsoFile, ios::in);

      // controllo esistenza file
      if (fileTesto.good() == true) usaFile = true;

      else                                    // se non
          trovo il file
      {
          cout << "Il file di testo non esiste!\n\n";      //
              comunico errore
          return 1;
```

```
            }
            fileTesto.close(); // chiudo file
        }

        else // l'utente digita un comando inesistente
        {
            cout << "La tua richiesta non e' chiara.\n";
            mostraUtilizzo();
            return 1;
        }
    }

    percorsoImg = argv[2]; // salvo il parametro successivo
        (percorso immagine)
    strcpy (percorso,percorsoImg.c_str()); // converto
        stringa in array di char

    // apro il file
    ifstream file;
    file.open(percorso, ios::in);

    if (file.good() == true)  // controllo esistenza file
    {
        // chiama una delle due funzioni disponibili (per
            inserire o estrarre dati dall'immagine) in base
             ai parametri passati dall'utente
        if( (argv[1] == string("-r")) || (argv[1] == string
            ("--rivela")) ) rivela(img, percorso, usaFile,
            percorsoFile);
        else if( (argv[1] == string("-n")) || (argv[1] ==
            string("--nascondi")) ) nascondi(img, percorso,
            usaFile, percorsoFile);

        cout << "\nOperazioni completate con successo!\n";
        return 0;
    }

    else                                    // se non trovo
        il file
    {
        cout << "Il file dell'immagine non esiste!\n"; // comunico
            errore
        file.close();                       // chiudo file
        return 1;
    }
}
```

# 2 THE GREAT FIREWALL OF CHINA

*"I disapprove of what you say, but I will defend to the death your right to say it."*

*— Voltaire*

## 2.1 BRIEF HISTORY

Since Mao Tse-Tung announced the establishment of the People's Republic of China, in 1949, the country has been controlled by an only party, the CCP or Chinese Communist Party. It instituted meaningful reforms that increased literacy, access to health care and distribution of land. However, any opposition to the regime was ruthlessly crushed.
After Mao's death in 1976, Deng Xiaoping became the leader; he pursued economic reforms, turned the country from a planned economy to a mixed economy with open market. Nevertheless, he thought that while opening China to the world would bring in economic development and progress, it would inevitability also bring in unwanted Western beliefs and ideologies that the Chinese government viewed as 'corrupt'.
In 1989 the Tiananmen Square protest sparked, demanding democratic rights and freedom of speech; the mass demonstration was suppressed by Chinese Army.

**Figura 9:** After 1989 events, "tank man" was seen worldwide.

At the end of the century, economic power grew really quickly, becoming one of the largest economies in the world. At the same time, technological progress started to spread in the country. Internet was introduced in 1989 as small-scale projects, then it began to be available to entire nation in 1994.

The Chinese government knew that opening up the Internet was necessary for China to grow economically but feared for the outcomes of opening up the nation to Western ideologies.
In 1998, the Golden Shield Project was initiated. Several projects were planned under the Golden Shield, including security management information system, criminal system as well as the famous Great Firewall of China. The Golden Shield Project was initially implemented to enhance network security, but it soon expanded to include censorship and surveillance. The project enabled maximum control of the Internet by the Chinese Communist Party. Nowadays, restrictions in China are not loosening up as the world becomes more interconnected.

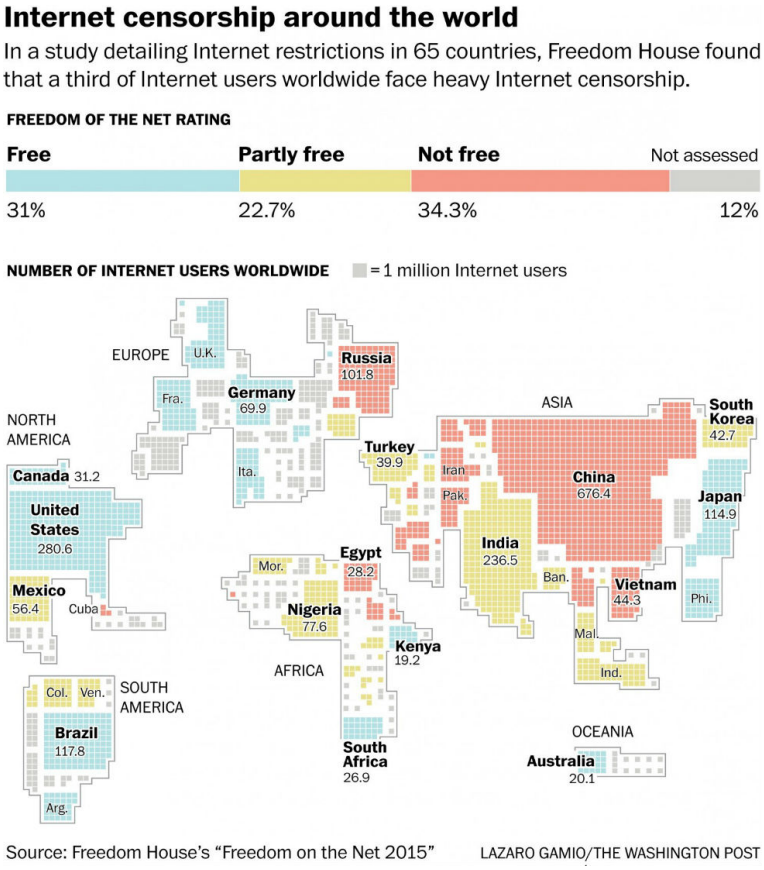


**Figura 10:** Internet censorship around the world in 2015

## 2.2 THE GREAT FIREWALL

The Great Firewall is an infrastructure part of the Golden Shield Project. It was started in 1998, in order to provide network security, surveillance and censorship. It took eight years and $700 million to build, and allowed the government to monitor electronic communications.
To do that, the CCP controls the exit node of China, Internet backbone and of the all eight ISPs in the country.

### 2.2.1 How does a firewall work?

A firewall is a network layer protection; it can be hardware or software. In a nutshell, a firewall is filter between a network and Internet.
The main function of a firewall is packet filtering, done by inspecting TCP

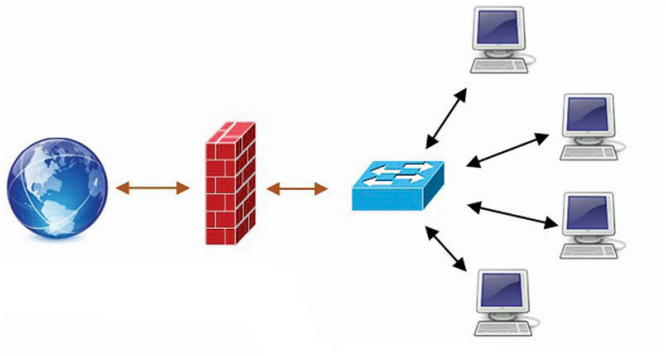

**Figura 11:** Firewall scheme

and UDP packets and comparing to a ”blacklist” or a table stored in the device. There are two main parts: inbound rules and outbound rules. Usually, inbound connections are all blocked except rules that allow a specific request, while the policy for outbound connection is to permit everything except some websites defined by outbound rules. The packet analysis could lead to three possible actions:
-allow: packet is allowed;
-deny: packet blocked and sent back to the sender;
-drop: packet blocked without notification to the sender.
A firewall may also provide additional features such as NAT (translates private to public addresses) and logging.

## 2.3 CENSORSHIP TECHNIQUES

The strict rules of the Great Firewall is done by several levels of filtering. The first level of the Golden Shield block specific domain names and server IP addresses (ip blocking).
The second stage implements keyword censorship: it can detect the content of the websites that citizens visit and reset TCP connection is ”sensitive content” is found.
Thirdly, the developers of the GRW finally managed to identify weaknesses in VPNs. They found that there are some obvious features of the commonly used VPN protocols, such as IPSec, L2TP/TPSec and PPTP, which often use specific ports. Thus, when processing the encrypted connection or “irregular” connection, a distinctive trace is left.
Thus, the Great Firewall was upgraded to detect even such traces.
Moreover, the GFW provide even advanced techniques like DNS Hijacking and Great Cannon.
DNS Hijacking redirects the user when he does a DNS query.

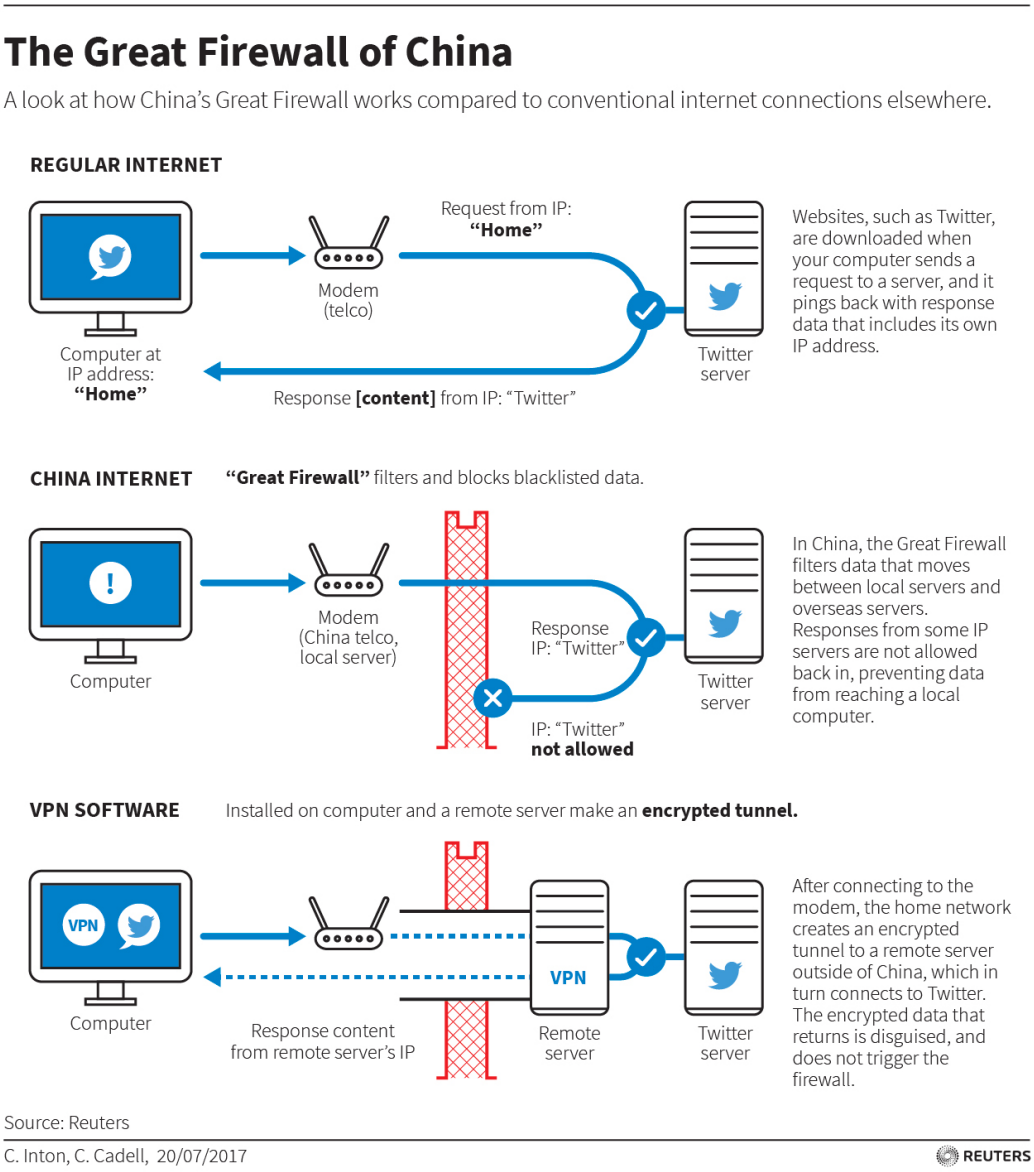


**Figura 12:** Comparison between different kind of connections

The Great Cannon is an offensive tool for launching a denial of service attack against a website or a server, by redirecting massive amounts of traffic. It was used to stop the availability of Github (a web based hosting service for version control, mostly used by programmers) in 2015.

### 2.3.1 Forbidden websites and services

There are certain groups of keywords that are blocked in China. Keywords related to politics and dissent are banned. Sexual content, gambling and violence are blocked as well. Anything linked to Tibet, Xinjiang and their independence is restricted. Also religion is not accessible; for example the Bible is forbidden.

***Some blocked keywords***

According to censorship-monitoring websites China Digital Times and Free Weibo, censored phrases include:

- ”I don’t agree”
- ”migration”
- ”emigration”
- ”re-election”
- ”election term”
- ”constitution amendment”
- ”constitution rules”
- ”proclaiming oneself an emperor”
- ”Winnie the Pooh”

[A nickname that social media users have coined for President Xi]

More censored contents are: 魏京生 - Wei Jingsheng, a human rights activist, 柴玲 - Chai Ling, one of the student leaders in the Tiananmen Square protests of 1989, 六四 - June 4, 美国之音 - Voice of America.

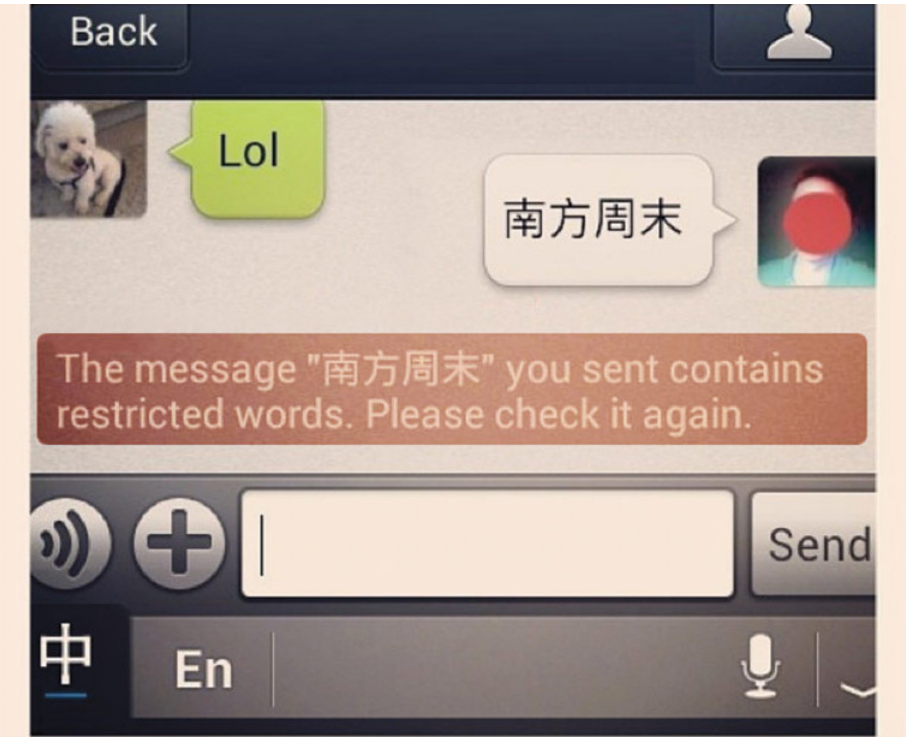


**Figura 13:** Nanfang Zhoumo | Southern Weekend, a liberal newspaper censored on WeChat in January 2013.

## 2.4 CIRCUMVENTION

Luckily, there are some tools and methods that allow to bypass the Great Firewall.

### 2.4.1 VPN

After connecting to a VPN (Virtual Private Network), an encrypted tunnel is created. This tunnel is directly linked to a remote server outside China which in turn connects to a requested website.

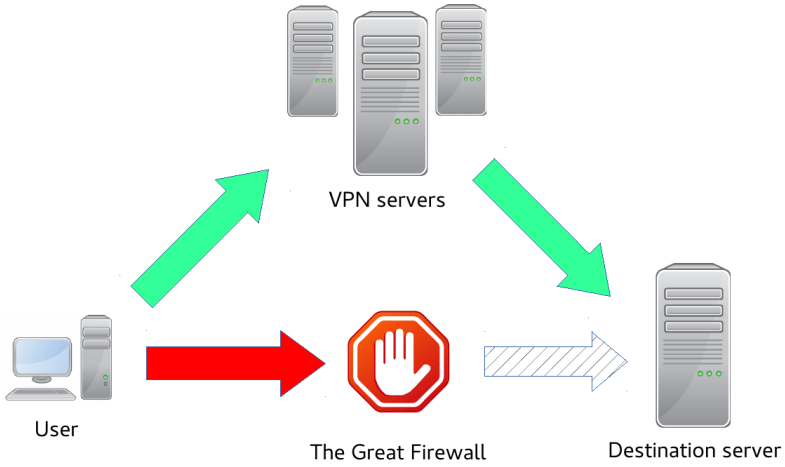


**Figura 14:** VPN scheme

### 2.4.2 Proxy

Shadowsocks is an open-source encrypted proxy project, created in 2012. It is based on SOCKS5 protocol and it is cross-platform.
Shadowsocks is based on a technique called proxying: before connecting to the wider internet, you first connect to a computer other than your own. This other computer is called a "proxy server." When you use a proxy, all your traffic is routed first through the proxy server, which could be located anywhere. By using proxing, packets follow a different route but traffic could be still identified. Thus, Shadowsocks creates and encrypted connection.
From a user perspective, VPN and Shadowsocks seem to work in the same way; this is true, as they both reroute and encrypt traffic.
Nevertheless, - as written before - Chinese censors have been able to find VPN fingerprints and recognize them. These techniques can't work with Shadowsocks, since it is a less centralized system. Each Shadowsocks user creates his own proxy connection, and so each looks a little different from the outside. The program allows you to configure connection, customizing settings; for instance you can install and run Shadowsocks on a server located outside China. Shadowsocks may give hope for freedom on China's internet.

## 2.5 REASONS

### 2.5.1 Social impact

Internet in China is more like an Intranet. 96% of traffic is internal: it goes to Chinese servers. Most of these data centers are in the capital city Beijing and information can be accessed by establishment whenever they want. Some research evidence has indicated that suspicion of the Great Firewall in China and the sense that one is being surveilled online leads to chilled speech and self-censorship, which has been more effective at blocking internet content than the Great Firewall has been.

### 2.5.2 Economical impact

Moreover, The Great Firewall is a form of trade protectionism that has allowed China to grow its own internet giants: Tencent, Alibaba, and Baidu. China has its own version of many foreign web properties, for example: Youku Tudou (YouTube), weibo.com (Twitter), Renren (Facebook) and WeChat (Whatsapp), Zhihu (Quora).
For instance, in March 2018, Alibaba had a revenue of about 39.89 billion dollars, based on 454 milion customers, mainly Chinese. Without the GFW, this large market probably would have been acquired by Amazon or other foreign markets.

### 2.5.3 Foreign reactions

Foreign companies such as Nortel Networks and Cisco Systems were hired to oversee the development of these censorship projects. These companies provided the necessary hardware and software to build the largest security network system in the world, a move that sparked controversy among the global community for its contributions to human rights violations in China.

## 2.6 CONCLUSIONS AND FUTURE PROSPECTS

As far as we know, Internet is a tool that can bring democracy through countries; blogs may often lead to political change but in China the government has succeeded in blocking activists from using the Internet as an effective political tool, by controlling a total online population of 772 millions of Internet users (Jan 2018).
This effort is accelerating as President Xi Jinping consolidates his power. The Chinese leadership has officially abolished term limits the 11th March 2018, giving Mr. Xi outsize authority over the country's direction.
When we talk about technology and the internet, we often look forward to a future that will promote liberalization but authoritarianism also rises with the development of technology, which makes wider and deeper control possible.
In recent years, the Chinese government even spread rumors about the crea-

tion of a 'social credit system' (社会信用体系 - shehui xinyong tixi), a rating to every citizen based on government data regarding their economic and social status. The system works as a mass surveillance tool and uses big data analysis technology. By 2020, as per plans, all of 1.4 billion Chinese citizens in the PRC will be given a personal score on how they behave. Some with low scores are already being punished if they want to travel. The goal is to force people toward behaviours that include obedience to the Party.
The increasing restrictions on the Internet and this new possible scenario seem to mean that China is moving to create the perfect totalitarian dystopia.

# BIBLIOGRAFIA

[1] Nicola Amato, *La Steganografia da Erodoto a Bin Laden. Viaggio attraverso le tecniche elusive della comunicazione*, Nicola Amato, 2016

# SITOGRAFIA


[2] *Inverse Data Hiding in a Classical Image by Using Scalable Image Encryption*, International Journal of Innovative Research in Computer and Communication Engineering
http://www.ijircce.com/upload/2014/march/32_Inverse.pdf

[3] *Information hiding using audio steganography – a survey*, The International Journal of Multimedia & Its Applications
http://aircconline.com/ijma/V3N3/3311ijma08.pdf

[4] *History of China*, Wikipedia
https://en.wikipedia.org/wiki/History_of_China

[5] *Meet Shadowsocks, the underground tool that China's coders use to blast through the Great Firewall*, Quartz
https://qz.com/1072701/meet-shadowsocks-the-underground-tool-that-chinas-coders-use-to-blast-through-the-great-firewall/

[6] *Within the Wall : Perceptions of Censorship by the Average Chinese Netizen*, Stanford University
https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/FreedomOfInformationChina/within-the-wall-perceptions-of-censorship-by-the-average-chinese-netizen-2/index.html

[7] *The evolution of China's Great Firewall: 21 years of censorship*, Hong Kong Free Press
https://www.hongkongfp.com/2017/09/03/evolution-chinas-great-firewall-21-years-censorship/

[8] *Great Firewall*, Wikipedia
https://en.wikipedia.org/wiki/Great_Firewall

[9] *The Global Age of Algorithm: Social Credit and the Financialisation of Governance in China*, chinoiresie.info
http://www.chinoiresie.info/the-global-age-of-algorithm-social-credit-and-the-financialisation-of-governance-in-china/

[10] *China's dystopian social credit system*, theconversation.com
https://theconversation.com/chinas-dystopian-social-credit-system-is-a-harbinger-of-the-global-age-of-the-algorithm-88348