ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

FACOLTA' DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea Triennale in Informatica

LA STEGANOGRAFIA E I SUOI MOLTEPLICI USI

Tesi di Laurea in Sicurezza

Relatore :

Prof. Ozalp Babaoglu

Presentato da :

Pasquale Paladino

III Sessione

Anno Accademico 2009-2010

RINGRAZIAMENTI

*Desidero innanzitutto ringraziare il Professore **Ozalp Babaoglu** per i preziosi insegnamenti durante gli anni di università, la pazienza che ha avuto con me in determinati momenti e soprattutto per avermi fatto appassionare alla sua materia.*

Ringrazio l'università di Bologna e in particolare la facoltà di Scienze matematiche fisiche e naturali per avermi dato la disponibilità di utilizzare i loro mezzi e le loro strutture per poter arrivare alla conclusione di questo percorso di studi.

Ringrazio tutte le persone che lavorano nelle sedi prima citate per la loro cordialità e disponibilità.

Ringrazio i miei compagni di corso, in particolare Giuseppe e Mirko, per i numerosi consigli durante la stesura dell'elaborato.

Ringrazio tutti i miei amici per il supporto e la vicinanza che mi hanno dimostrato in questi anni di università.

Ringrazio i nonni e tutta la famiglia che con il loro esempio e affetto rappresentano la mia base e il trampolino di lancio per il mio futuro.

Ringrazio con affetto i miei genitori per aver creduto in me, avermi sostenuto nelle mie scelte e soprattutto per avermi finanziato. Vi voglio bene.

Infine resta solo mia sorella (santa donna) che ogni giorno mi ha sopportato nei miei momenti di sconforto e mi ha aiutato a superarli.

Indice dei contenuti

1 – Introduzione 5

2 – Steganografia in generale 8

2.1 – Definizione di Steganografia 8

2.2 – Steganografia nella storia 8

2.3 – Crittografia e Steganografia a confronto 13

3 – Steganografia in dettaglio 15

3.1 – Il problema dei prigionieri 15

3.2 – Steganografia Sostitutiva 22

3.3 – Steganografia Selettiva 27

3.4 – Steganografia Costruttiva 28

3.5 – Il cover object ideale 28

4 – La Steganalisi 31

4.1 – Descrizione e tipi di attacco 31

5 – Steganografia nei file immagine 35

5.1 – File immagini 35

5.2 – Forme di compressione 36

5.3 – Steganografia applicata alle immagini 37

5.4 – Formato immagini GIF 38

5.5 – Metodi per celare informazioni nelle immagini 42

5.6 – LSB 43

5.7 – Masking and Filtering 45

5.8 – Algoritmi e trasformazioni 46

5.9 – JSTEG 49

5.10 – OutGuess 50

5.11 – Sistema Steganografico sicuro 51

6 – Steganografia nei file audio 57

6.1 – Informazioni sui file audio 57

6.2 – Steganografia applicata ai file audio 59

6.3 – LSB (audio) 60

6.4 – Spread Spectrum 61

6.5 – Phase Coding 62

6.6 – Echo Data Hiding 62

7 – Presente e passato della Steganografia 64

7.1 – La Steganografia oggi e sviluppi futuri 64

8 – Definizioni 66

9 – Riferimenti 67

1- INTRODUZIONE

La comunicazione, oggigiorno, è intesa come scambio di informazioni fra più soggetti e si è evoluta prepotentemente con l'espansione dei computer su larga scala. Internet è diventato un enorme ragnatela che si appoggia a molteplici canali di comunicazione per collegare tutti i terminali che ne fanno parte (computer, palmari, cellulari ecc..). I canali di comunicazione possono essere visti come enormi autostrade nelle quali è possibile osservare il traffico presente e non solo. I satelliti, i dispositivi bluetooth e le reti Wireless, ad esempio, diffondono nell'etere informazioni in quantità astronomiche che possono essere intercettate e lette facilmente da persone non autorizzate. Proprio per questo si usano sistemi di protezione dell'informazione che sfruttano tecniche di crittografia per nascondere il significato della comunicazione. La loro robustezza si basa su un altissimo costo computazionale per decifrare il messaggio e, in ogni caso, in tutto questo, non viene preso in considerazione l'aspetto dell'invisibilità di tale informazione. Possono verificarsi infatti situazioni in cui è necessario rendere il più possibile invisibile il messaggio da trasmettere. Nello spionaggio industriale, ad esempio, il computer di un tecnico può essere facilmente tenuto sotto controllo analizzando il contenuto dei pacchetti IP che attraversano il gateway dell'azienda o tenendo sotto controllo la sua e-mail tramite un'opportuna gestione del server di posta elettronica.

Se affrontiamo invece il problema a livello "più in grande", autorità nazionali come la Cina o sistemi di intercettazioni internazionali cercano di monitorare per quanto possibile il contenuto dell'informazione presente in internet, anche tenuto conto di un'insorgente ondata di terrorismo dove da un lato (quello dei terroristi) la comunicazione non deve in nessun modo essere intercettata, dall'altro (la sicurezza pubblica) si cerca di rilevare la presenza di dati nascosti. L'invisibilità dell'informazione inoltre è importante anche nelle comunicazioni interne dei servizi di

sicurezza nazionali, infatti anche in questo frangente i dati devono essere il più possibile sicuri di non essere intercettati da “entità” esterne.

Con lo sviluppo di nuove forme di comunicazione, nasce anche il desiderio quindi di poter monitorare l'informazione per salvaguardare sicurezze nazionali o interessi privati. Questo senso di paura nell'essere controllati genera la necessità di sviluppare nuove tecniche per una comunicazione sicura, nascosta da occhi e orecchie indiscrete. Va altresì fatta una distinzione fra rendere il messaggio incomprensibile e rendere invisibile la comunicazione.

La tecnica che si utilizza nel primo caso è la Crittografia : il messaggio diventa incomprensibile a chi non è in possesso della “chiave”, ovvero del codice che permette la decodifica in chiaro dei dati. Chi progetta tecniche crittografiche si preoccupa di proteggere da occhi indiscreti il contenuto del messaggio da trasmettere considerando possibile un'intercettazione della comunicazione durante la trasmissione.

Nel secondo caso invece è compito della Steganografia rendere invisibile la comunicazione. La steganografia è un modo poco conosciuto di proteggere la confidenzialità dei dati durante la comunicazione. Ci possono essere delle situazioni, come quelle precedentemente descritte, in cui esiste la necessità di trasmettere informazioni senza destare il sospetto di chi controlla il mezzo trasmissivo. Inviare il messaggio attraverso la crittografia può generare dubbi sul perché tale informazione è stata resa illeggibile. Infatti tutti i messaggi che vengono cifrati hanno il problema della facile riconoscibilità che li rende inutilizzabili in circostanze particolari come ad esempio in un sistema dove è previsto un controllo selettivo sul traffico informativo. È chiaro che per i motivi più svariati, anche solo per la semplice curiosità che contraddistingue l’uomo, vengono incrementati i motivi e le cause per cui un determinato utente voglia scoprire cosa si celi dietro l’utilizzo di tale tecnica. Al contrario, mai nessun utente o organo di controllo concentrerà forze su chi invia o riceve dati 'normali' come file multimediali o comune corrispondenza. La steganografia utilizza questo concetto di invisibilità della comunicazione per trasmettere informazioni

'segrete'. In altri termini, non desta sospetti chi trasmette ad un conoscente un'immagine digitale con un bel panorama, ma se questi prima di inviare la foto ne modifica leggermente il contenuto inserendovi un messaggio nascosto all'interno, la foto risulterà ancora più interessante per il destinatario. La tecnica steganografica applicata all'immagine permette, in questa situazione, di inserire nella foto dati o messaggi che nulla hanno a che vedere con il loro supporto: il panorama. La caratteristica fondamentale che deve soddisfare la steganografia durante l'inserimento del messaggio è la non alterazione visiva dell'immagine. In altri termini, un osservatore umano in grado di vedere l'immagine prima della sua modifica e dopo l'inserimento dell'informazione, non deve essere in grado di distinguerne le differenze.

Gli utilizzatori di steganografie devono però scontrarsi con l'eventualità che, ad esempio, la loro immagine steganografata sia identificata come tale da programmi di identificazione dei messaggi nascosti utilizzati da chi vuole tenere sotto controllo le comunicazioni. Questi programmi prendono il nome di steganalizzatori e sfruttano le alterazioni introdotte durante l'inserimento del messaggio sul rumore di fondo o su statistiche di elevato ordine dell'immagine. Si può quindi dire che uno steganalizzatore si dice ideale quando è in grado di identificare senza errore tutte e sole le immagini steganografate. Viceversa una tecnica steganografica è ottima quando nessun steganalizzatore è in grado di individuarla.

In tutto questo scenario è chiaro che la steganografia fallisce nel momento stesso che per qualsivoglia motivo si scopra che il messaggio sia diverso da quello che si vuole far intendere. Quindi un attenzione particolare va fatta nel cercare di evitare situazioni particolari in cui si possa essere scoperti. In questo elaborato quindi cercheremo di studiare a fondo quelle che sono le varie possibilità e le varie opportunità che la Steganografia ci pone davanti nell’ambito della comunicazione. Cominceremo con alcuni cenni storici sulla Steganografia, esamineremo quelle che sono le varie sfaccettature che contraddistinguono la Crittografia dalla Steganografia, parleremo in dettaglio di quelle che sono le varie tecniche Steganografiche in circolazione , faremo degli esempi in dettaglio di algoritmi steganografici utilizzati su vari tipi di formati

immagine e audio, parleremo di Steganalisi e infine esamineremo quelli che sono gli utilizzi attuali ed eventualmente futuri in questo campo della comunicazione "nascosta".

2- STEGANOGRAFIA IN GENERALE

2.1 – Definizione di Steganografia

Il termine Steganografia è composto da due parole di origine greca *στεγανός* e *γραφία* che significano rispettivamente "nascosto" e "scrittura". Il significato che quindi si deduce da questa parola è "scrittura nascosta", ovvero l'insieme delle tecniche e delle metodologie che permettono a due o più entità (persone o macchine) di comunicare tra loro in modo da occultare a possibili ascoltatori, o terze persone non gradite, non tanto il contenuto del messaggio (Crittografia) ma la stessa esistenza di una comunicazione o un messaggio riservato. In sintesi la steganografia è l'arte di nascondere un messaggio confidenziale all'interno di un messaggio contenitore (reso pubblico) che normalmente è di aspetto diverso, non sospettabile e riconducibile al messaggio segreto.

2.2 – La steganografia nella storia

I primi scritti che documentano l'utilizzo di tecniche steganografiche risalgono proprio al tempo della civiltà greca. Si parla infatti di incisioni su tavolette di legno del messaggio, poi lo scritto veniva coperto con della cera in modo da dare l'impressione che la tavoletta fosse inutilizzata. Essa invece aveva al suo interno un messaggio nascosto e mai nessuno avrebbe sospettato della sua esistenza, a parte il destinatario naturalmente.

Il primo testo stampato di steganografia fu scritto da Johannes Trithemius (1462-1516) con il titolo Steganographia. Trithemius, nome italianizzato dell'umanista e teologo tedesco Johannes von Heidenberg, detto Tritheim, lo scrisse tra il 1499 e il 1500, dopodiché il libro circolò a lungo in forma manoscritta. Stampato nel 1606, venne poco dopo iscritto nell'Index Librorum Prohibitorum in quanto "pericoloso e colmo di superstizioni". Una curiosità: la prefazione del libro annuncia in modo provocatorio - anche se oscuro - la presenza di un messaggio nascosto (il dilemma è stato risolto solo nel 1998 da Jim Reeds della AT&T Labs [7]).

Un esempio di steganografia è riportato in uno scritto, dove Erodoto racconta la storia di un nobile persiano che fece tagliare a zero i capelli di uno schiavo fidato al fine di poter tatuare un messaggio sulla sua testa; una volta che i capelli furono ricresciuti lo mandò a destinazione con la sola istruzione di tagliarseli nuovamente. Sia nel caso delle tavolette di cera che dello schiavo, se una terza persona avesse intercettato il mezzo di comunicazione, difficilmente sarebbe riuscita a scoprire l'esistenza del messaggio nascosto. È proprio questa l'idea che sta alla base della steganografia: senza conoscere la tecnica steganografica adottata non deve essere possibile né risalire al vero scopo della comunicazione, né tantomeno sospettarne l'esistenza.

Gli antichi Romani usavano scrivere fra le righe di un testo utilizzando un inchiostro fatto con sostanze naturali come il succo di limone, l'aceto o il latte. Il messaggio nascosto diventava visibile una volta che il testo veniva avvicinato ad una fonte di calore. Questi inchiostri invisibili sono noti anche come inchiostri simpatici.

Figura : ***Esempio di inchiostro simpatico***

Le griglie di Cardano (1501–1626) erano fogli di materiale rigido nei quali venivano ritagliati fori rettangolari a intervalli irregolari; applicando la griglia sopra un foglio di carta bianca, il messaggio segreto veniva scritto nei buchi (ciascun buco poteva contenere una o più lettere), dopodiché si toglieva la griglia e si cercava di completare la scrittura del resto del foglio in modo da ottenere un messaggio di senso compiuto, il quale poi veniva inviato a destinazione. Applicando sul foglio una copia esatta della griglia originaria, era possibile leggere il messaggio nascosto. Ecco un esempio di come un semplice messaggio "CASA" possa essere nascosto nel primo canto della Divina Commedia di Dante:

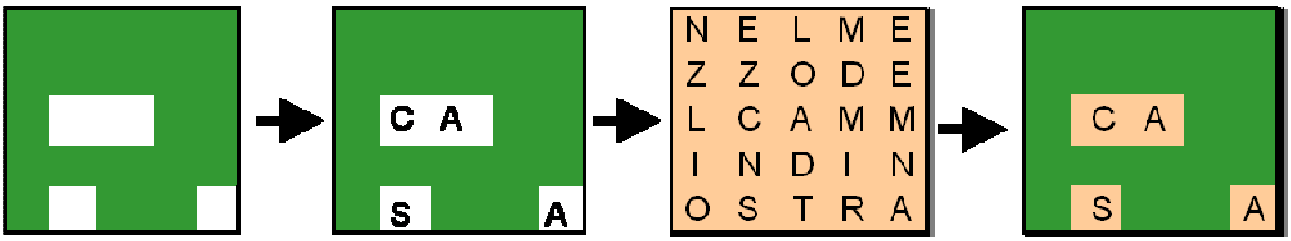
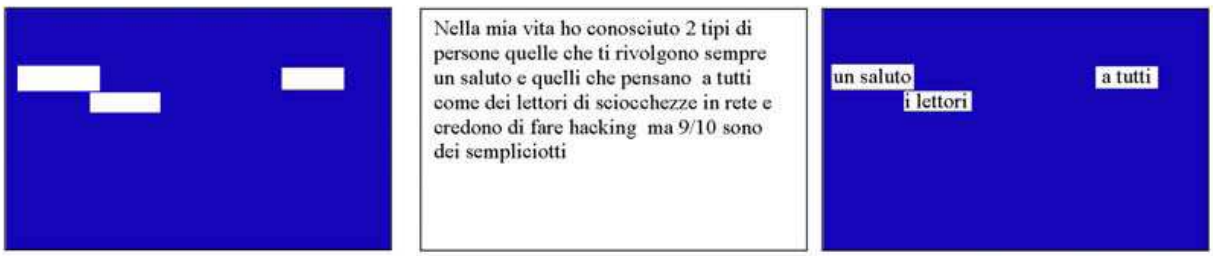


Figura : ***Esempio di griglia di Cardano***

Oppure come un messaggio di semplice saluto possa essere estratto da un messaggio di senso diverso sempre avendo a disposizione l'apposita griglia:



Cartoncino ritagliato | Messaggio segreto | Sovrapposizione Cartoncino

Figura : ***Esempio 2 di griglia di Cardano [5]***

Una tecnica inventata invece dal F.B.I. durante la seconda guerra mondiale fu quella dei micropunti fotografici [6]: si tratta di fotografie della dimensione di un punto dattiloscritto che, una volta sviluppate e ingrandite, possono diventare pagine stampate di buona qualità.

Figura : ***Esempio di micro punti fotografici***

Sempre durante la seconda guerra mondiale furono impiegate come inchiostri simpatici sostanze molto sofisticate, come ad esempio gli inchiostri al cobalto, che possono essere resi visibili solo mediante l'uso di particolari reagenti chimici.

Un altro modo di nascondere un messaggio è quello di celarlo all'interno di un altro testo. Un acrostico è una poesia - o un testo di qualsiasi tipo - composta intenzionalmente in modo tale che, unendo le prime lettere di ogni capoverso, si ottenga un messaggio di senso compiuto. Esistono numerose varianti di questa semplice idea di base, come il testo che segue:

"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."

Questa frase fu spedita via radio da una spia nazista durante la seconda guerra mondiale e apparentemente sembra non dire niente di importante. Invece la costruzione della frase è stata fatta tenendo conto del messaggio da nascondere. Per capire meglio bisogna leggere in modo diverso la frase estraendo la seconda lettera di ogni parola e osservare il messaggio segreto:

“Pershing sails from NY June 1”

Ci sono naturalmente delle interpretazioni da fare, come ad esempio la lettere “i” finale che deve essere intesa come un numero, però per il resto è tutto chiaro.(c’è solo una “r” in più). Tale tecnica può essere applicata in questa forma o in forme più evolute come, ad esempio, inserendo il messaggio attraverso punteggiatura, giocando su possibili errori di battitura degli spazi. E’ chiaro come sia una tecnica piuttosto semplice, ma comunque raggiunge il suo scopo.

Andando avanti con gli esempi di steganografia molto interessanti sono le opere di Shakespeare. Secondo molti studiosi, infatti, alcune opere dell’inglese possono essere attribuite al noto scrittore e statista Francis Bacon. Questo perché all’interno di tali scritti vi sono diversi testi nascosti che contengono il nome di Bacon stesso. A rafforzare questa ipotesi contribuiscono interessanti retroscena che accomunano Shakespeare e Bacon.[3]

Anche oggi la steganografia viene utilizzata come veicolo politico-militare. Ad esempio, nel famoso quotidiano americano “USA Today” [4] del 10 luglio del 2002 si legge: “Ultimamente al-Queda ha inviato centinaia di messaggi crittografati nascosti in fotografie digitali sul sito eBay.com. Molti dei messaggi sono stati inviati da café pakistani e librerie pubbliche di tutto il mondo…“. E ancora: “Ufficiali americani dicono che azzam.com contiene messaggi crittografati nelle sue immagini e nei suoi testi (pratica conosciuta come steganografia). Essi affermano che i messaggi contengono istruzioni per i nuovi attacchi di al-Queda”.

In sintesi, la steganografia è l'arte di nascondere un messaggio in un supporto ospitante. Come supporto può essere inteso qualsiasi mezzo, oggetto fisico o file nel

caso di elaborazione informatica. È Chiaro, quindi, che con il diffondersi di computer ed internet è aumentata anche la possibilità di appoggiarsi a queste nuovi mezzi per trasmettere messaggi nascosti. L'ideazione di nuove tecniche steganografiche non si appoggiano più solo a documenti di testo per trasmettere informazione, ma anche a qualsiasi formato digitale, come file multimediali e file di immagini.

2.3 – Crittografia e Steganografia a confronto

La Crittografia ha l'obiettivo di rendere protetto un dato canale di comunicazione da un eventuale ascoltatore non autorizzato, impedendogli cosi di accedere ai messaggi scambiati su quel canale. La Steganografia invece si prefigge il compito di nascondere il messaggio segreto e renderlo "invisibile" agli occhi del terzo incomodo , utilizzando sempre un canale di pubblico accesso. Il fine ultimo della crittografia è quello di proteggere il contenuto di un messaggio, preservando, attraverso un metodo di cifratura, la confidenzialità di tale messaggio. Anche se un testo cifrato è perfettamente riconoscibile come tale (vedi esempio in seguito), è fondamentale che un estraneo, che venga in possesso del messaggio, non possa in alcun modo accedere alle informazioni contenute in esso o poterle modificare. Lo scopo della steganografia, invece, è quello di impedire che un estraneo abbia il seppur minimo indizio che stia avvenendo un trasferimento di dati confidenziali. Nessuno deve poter sospettare che si voglia trasmettere materiale riservato: la steganografia fallisce nel momento in cui la trasmissione viene scoperta, anche se non si è in grado di decrittarne il contenuto.

Esempio di testo cifrato con Pretty Good Privacy(PGP):

qANQR1DBwU4D/TlT68XXuiUQCADfj2o4b4aFYBcWumA7hR1Wvz9rbv2BR6WbUsyZBIEFtjyqCd96qF38sp9IQiJIKlNaZfx2GLRWikPZwchUXxB+AA5+lqsG/ELBvRac9XefaYpbbAZ6z6LkOQ+eE0XASe7aEEPfdxvZZT37dVyiyxuBBRYNLN8Bphdr2zvz/9Ak4/OLnLiJRk05/2UNE5Z0a+3lcvITMmfGajvRhkXqocavPOKiin3hv7+Vx88uLLem2/fQHZhGcQvkqZVqXx8SmNw5gzuvwjV1WHj9muDGBY0MkjiZIRI7azWnoU93KCnmpR60VO4rDRAS5uGl9fioSvze+q8XqxubaNsgdKkoD

+tB/4u4c4tznLfw1L2YBS+dzFDw5desMFSo7JkecAS4NB9jAu9K+f7PTAsesCBN
ETDd49BTOFFTWWavAfEgLYcPrcn4s3EriUgvL3OzPR4P1chNu6sa3ZJkTBbri
DoA3VpnqG3hxqfNyOlqAkamJJuQ53Ob9ThaFH8YcE/VqUFdw+bQtrAJ6NpjIx
i/x0FfOInhC/bBw7pDLXBFNaXHdlLQRPQdrmnWskKznOSarxq4GjpRTQo4hp
CRJJ5aU7tZO9HPTZXFG6iRIT0wa47AR5nvkEKoIAjW5HaDKiJriuWLdtN4O
XecWvxFsjR32ebz76U8aLpAK87GZEyTzBxdV+lH0hwyT/y1cZQ/E5USePP4oK
WF4uqquPee1OPeFMBo4CvuGyhZXD/18Ft/53YWIebvdiCqsOoabK3jEfdGExce
63zDI0==MpRf

Il messaggio, oltre ad essere, evidentemente, privo di alcun significato semantico ed a non rispettare alcuna regola sintattica, presenta caratteri che appaiono con un ordine casuale (le occorrenze dei vari caratteri sono equiprobabili), a differenza di qualsiasi messaggio di senso compiuto, in italiano o in inglese o in qualsiasi altra lingua, dove esiste un rapporto di frequenza tra le varie lettere (per esempio, nelle parole della lingua italiana la lettera “a” è molto più frequente della lettera “z”). Per tali motivi, se si considera un ambiente di trasmissione in cui una entità di qualsivoglia genere (sia essa di controllo o di disturbo) effettui un esame sul contenuto delle trasmissioni, suddetta entità potrà dedurre con facilità che si tratta di un messaggio criptato o comunque di un messaggio sospetto. In conclusione, il messaggio di cui sopra è un esempio di crittografia, ma non è steganografia. Dal punto di vista della steganografia, la trasmissione, una volta che la presenza di un messaggio nascosto è stata individuata, può considerarsi non riuscita anche se il messaggio nascosto non verrà decrittato; la crittografia, invece, considera fallita una comunicazione quando un estraneo è in grado di ottenere il messaggio originale.

3- STEGANOGRAFIA IN DETTAGLIO

3.1 – Il problema dei prigionieri

Il primo teorico a studiare approfonditamente il problema della Steganografia fu G.J. Simmons attraverso quello che è passato alla storia come il “problema dei prigionieri”. La storia racconta di due persone, Alice e Bob, che sono state rinchiuse in una prigione in celle separate. Hanno il problema di poter comunicare liberamente tra loro per poter organizzare un piano di fuga e l’unica possibilità che hanno è quella di scambiare dei messaggi attraverso il secondino di nome Wendy che li controlla. Naturalmente i messaggi che devono scambiarsi non possono essere espliciti altrimenti il secondino li metterebbe in isolamento bloccando il loro piano in partenza. Quindi i due devono riuscire ad inventarsi un modo per poter far passare i loro messaggi segreti sotto forma di messaggi innocui (immagini o testo), in modo che possano passare tranquillamente tra le mani di Wendy senza il pericolo di essere bloccati o compresi.

Il modello generale per lo studio della Steganografia può essere riassunto in questo schema :

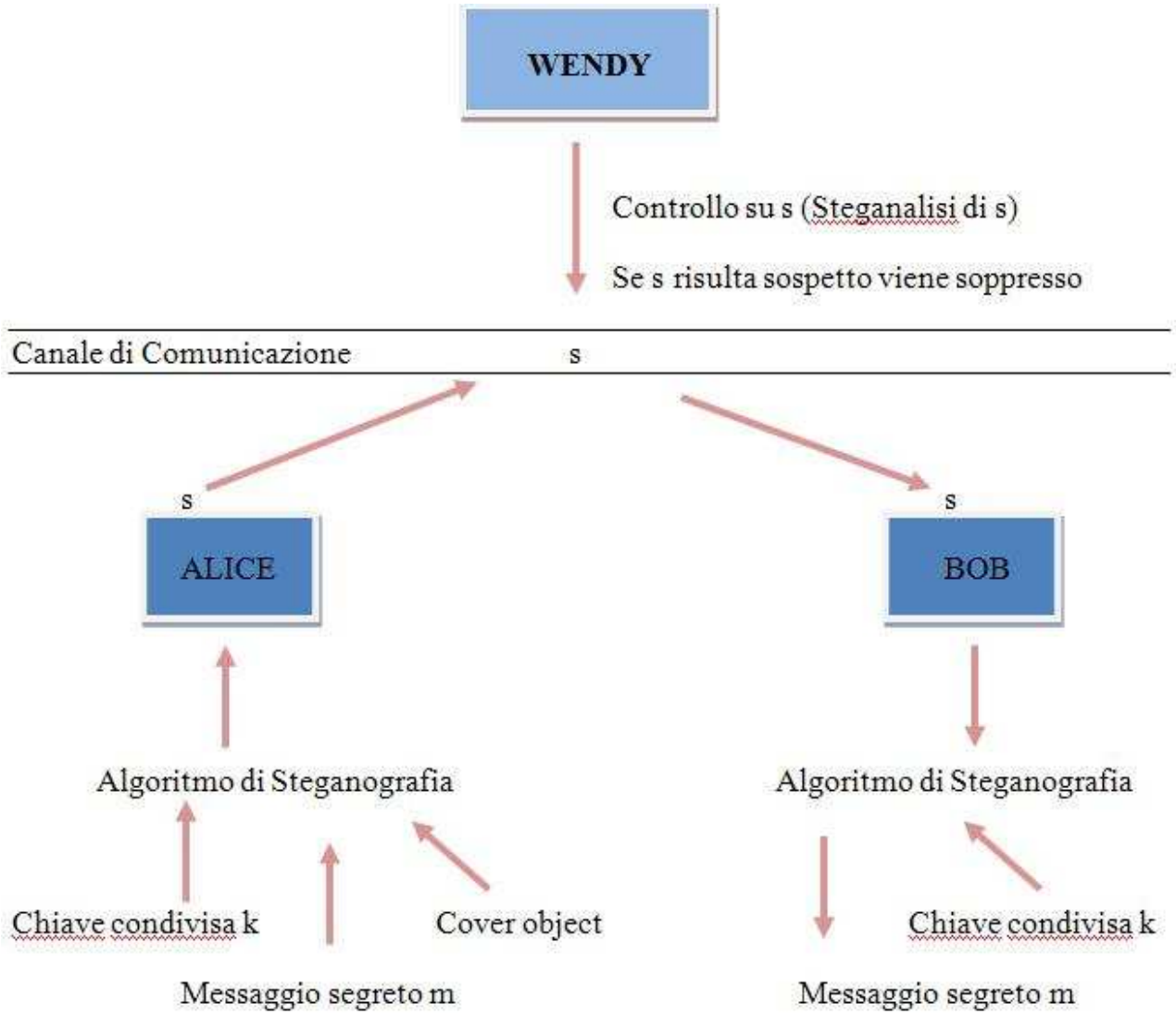


Figura : ***Schema Steganografico di Simmons***

Alice intende mandare un messaggio segreto “m” a Bob, per farlo lo nasconde in un oggetto di copertura (cover object) “c” ed eventualmente anche combinandolo con una chiave segreta “k” conosciuta sia da Alice che da Bob. Alice otterrà un messaggio segreto (Stego Object) “s” che verrà spedito sul canale di comunicazione (nel nostro caso per mano di Wendy) a Bob.

Altro esempio con questa volta un immagine a fare da contenitore :

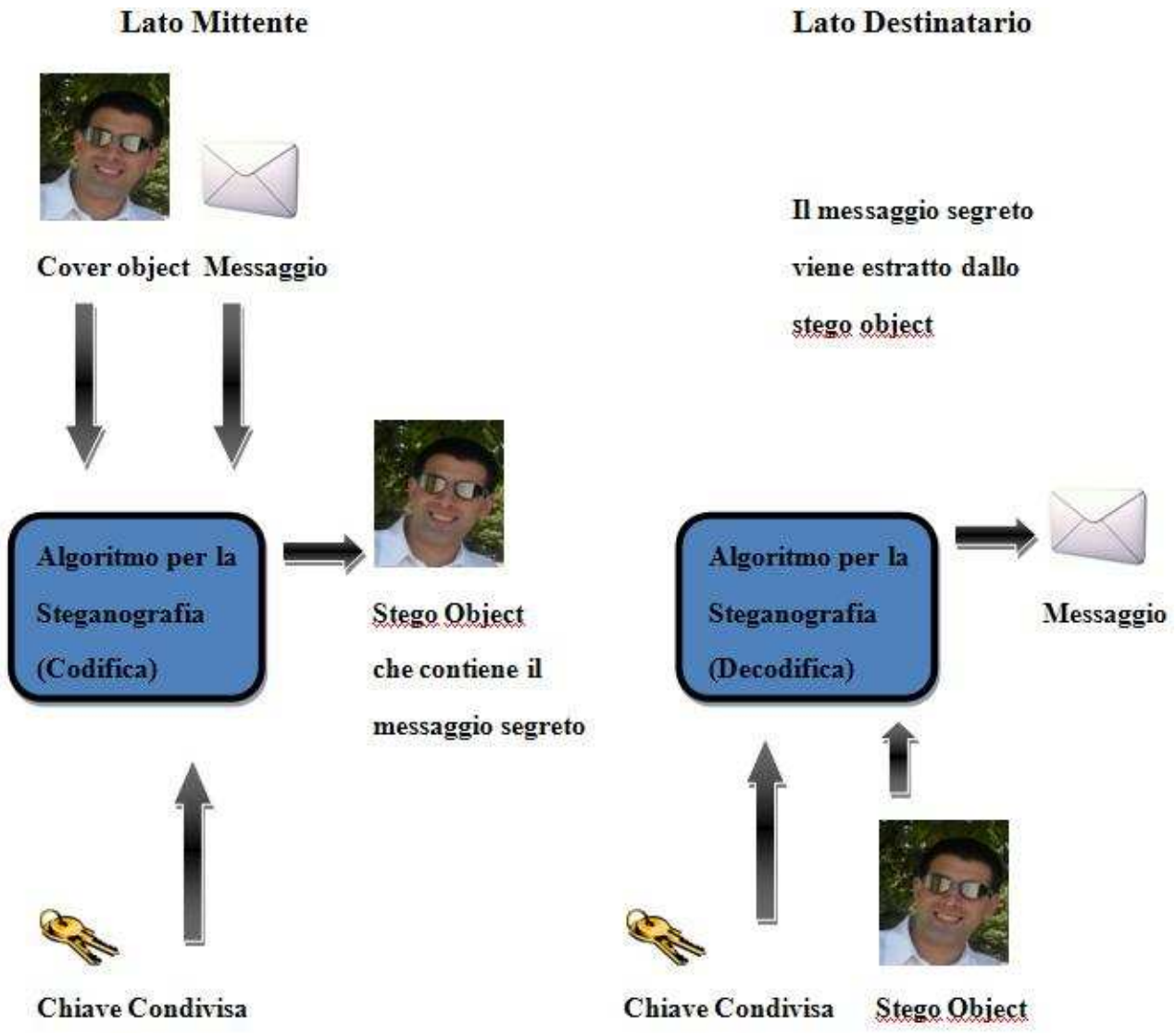


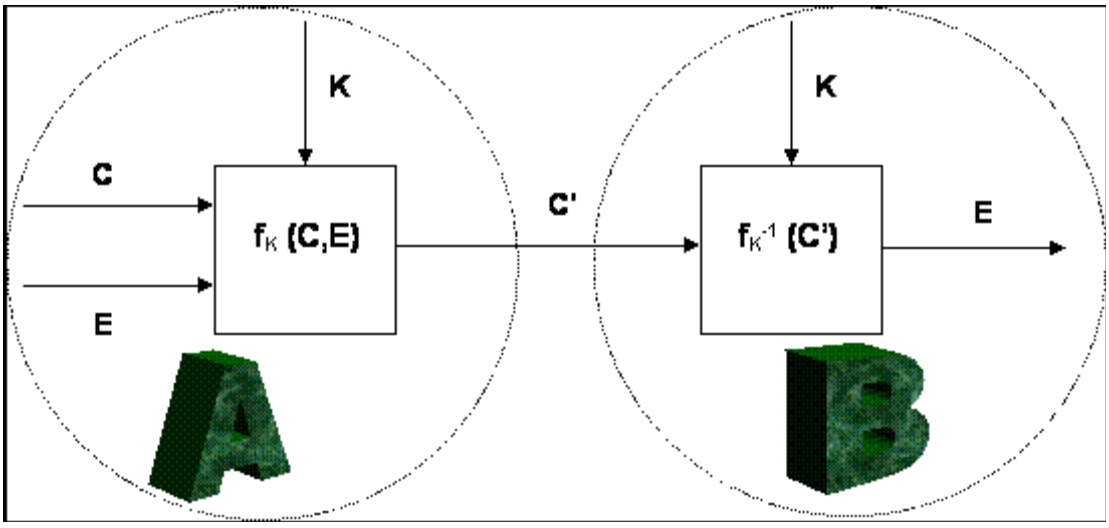
Figura ***: Schema Steganografico***

L'algoritmo steganografico utilizzato per combinare chiave segreta, messaggio e oggetto può essere segreto (conosciuto da Bob e Alice, ma non da Wendy) oppure pubblico. Di norma l'algoritmo è sempre pubblico, mentre la sola parte segreta è la chiave k. Nel nostro caso specifico, Wendy è a conoscenza dell'algoritmo di Steganografia e quindi al momento che riceve i messaggi potrebbe comportarsi come

un disturbatore/controllore passivo o attivo. In caso Wendy abbia un ruolo **passivo** allora si limiterà solamente a controllare i messaggi per valutare se ci siano contenuti segreti oppure no. Se trova qualcosa bloccherà la comunicazione tra i due prigionieri , altrimenti lascerà passare.

In caso invece abbia un ruolo **attivo** allora non si limiterà solamente a controllare il messaggio, ma farà in modo di alterarlo per renderlo illeggibile al destinatario.È chiaro quindi che il compito di un buon algoritmo steganografico sia quello di evitare che il disturbatore/controllore (Wendy) possa in qualche modo pensare che il messaggio contenga qualcosa di nascosto e quindi intervenire. La Steganalisi invece va in aiuto di Wendy offrendogli degli strumenti utili per cercare di distingue un messaggio normale (cover object) da un messaggio segreto (stego object). In ogni caso, Wendy deve operare la sua analisi non conoscendo la chiave segreta condivisa da Alice e Bob e, spesso, senza conoscere neppure l'algoritmo che i due stanno utilizzando per nascondere i loro messaggi. Ben si capisce che la steganalisi è di per sé un problema molto complesso. Comunque, solitamente non si richiede a Wendy di scoprire anche il contenuto del messaggio segreto, laddove ella dovesse riscontrarne la presenza; questa assunzione abbassa notevolmente la complessità del problema, rendendo un po' più agevole il compito del disturbatore/controllore.

Altro schema di rappresentazione del sistema steganografico [8]:



Dove:

• E è il messaggio segreto da nascondere;

• C è il contenitore;

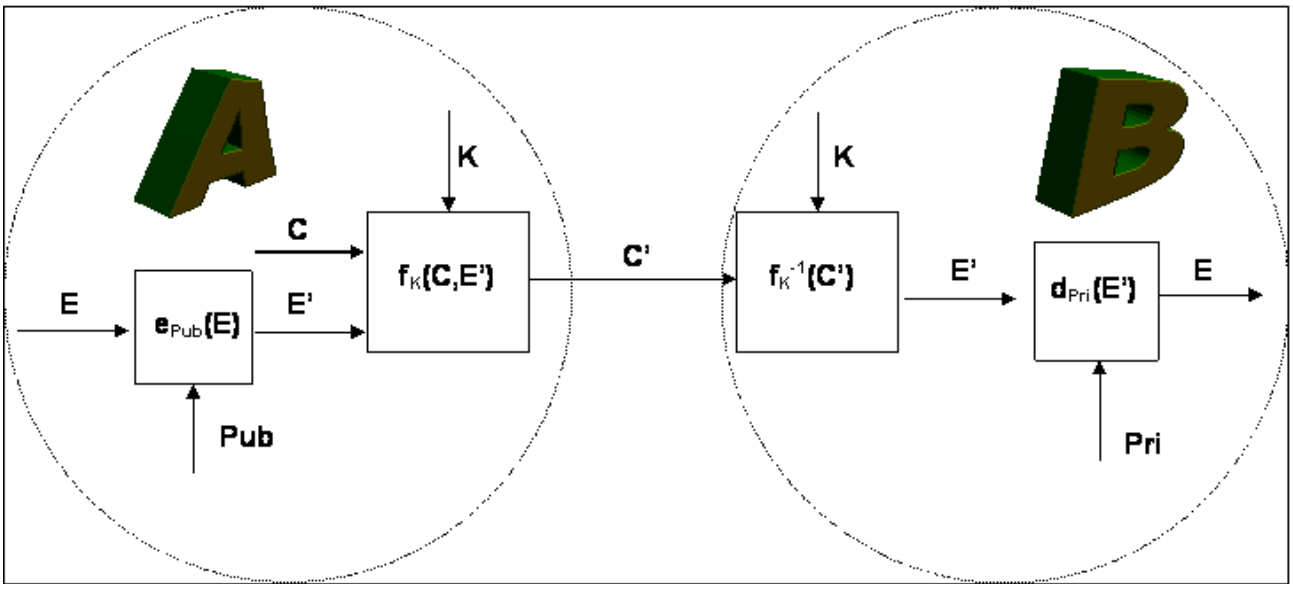
• C' è il frammento stego ottenuto incapsulando E in C;

• K è la chiave segreta che A e B devono conoscere;

• fK(C,E) è la funzione steganografica che nasconde E in C usando la chiave K;

• fK-1(C’) è la funzione inversa di f che, sfruttando la chiave K e partendo dal frammento stego C' ricevuto, riesce a risalire ad E. Se la chiave fornita è la stessa usata dal mittente per nascondere il messaggio segreto e se C’ è lo stesso frammento prodotto dal mittente (potrebbe essere stato modificato da un attaccante), allora la funzione di estrazione produrrà effettivamente il messaggio segreto originale E.

Il sistema esaminato comunque ha come base fondante per funzionare il fatto che ci sia stato in precedenza uno scambio sicuro di una chiave privata k tra i due utenti della comunicazione. Nel caso in cui non ci sia stata questa possibilità, esiste un modo diverso di affrontare il problema che esiste anche in crittografia, ovvero l’utilizzo di una chiave pubblica conosciuta dal destinatario.



Rispetto allo schema precedente vi è l'aggiunta del concetto di chiave pubblica [8]:

• Pub è la chiave pubblica di B che A conosce;

• ePub(E) è la funzione di encoding che prende in input la chiave pubblica di B (Pub) e il messaggio nascosto E da cifrare: il risultato di tale operazione è il messaggio E', che costituirà l'input per l'algoritmo di steganografia vero e proprio;

• E' è il messaggio segreto e cifrato che viene incapsulato all'interno del cover C;

• Pri è la chiave privata di B, grazie alla quale si riesce a risalire ad E partendo da E'.

• dPri(E') è la funzione di decoding che decifra il messaggio segreto e cifrato E', sfruttando la chiave privata di B (Pri) e che dà in output il messaggio segreto E;

Perché questo metodo funzioni, ognuno ha bisogno di sapere come estrarre il messaggio segreto da un potenziale stego object. Questo algoritmo di estrazione può essere applicato anche ai files che non contengono messaggi nascosti: infatti, non è importante che un file contenga un messaggio segreto o meno, il risultato sarà comunque una stringa di dati random che solo B sarà capace di decifrare con successo.

L’inconveniente di questo sistema sta nel fatto che ogni qualvolta si riceve un potenziale stego object si deve estrarre il potenziale testo cifrato e provare a decifrarlo con la propria chiave privata (senza essere sicuri di trovare un messaggio).

Abbiamo visto, quindi, come ciò che caratterizza un algoritmo steganografico sia la presenza di un messaggio contenitore che conserva al suo interno un messaggio segreto di norma disgiunto dal primo per significato e valore. Le prime distinzioni nel campo della steganografia tra le diverse tecniche partono proprio da come differenziare il contenitore in modo da essere il più segreto possibile. Alcune di queste tecniche consentono di "iniettare" il messaggio segreto dentro un messaggio contenitore già esistente, modificandolo in modo tale da contenere sia il messaggio originale che il messaggio “nascosto”, rendendolo praticamente indistinguibile dall'originale. Indichiamo l'insieme di queste tecniche con il termine steganografia **iniettiva**.

Figura : ***Schema Steganografia iniettiva***

Esistono tuttavia altre tecniche steganografiche che hanno capacità proprie di generare potenziali messaggi contenitori e utilizzano il messaggio segreto per "pilotare" il processo di generazione del contenitore. Per queste tecniche adottiamo il termine steganografia **generativa**.

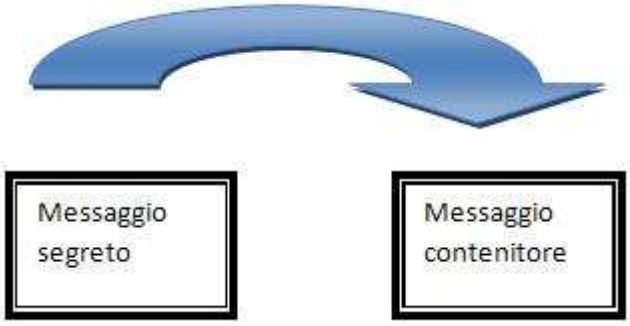


Figura : ***Schema steganografia generativa***

Di seguito vengono descritte le principali classi di tecniche steganografiche:

- la steganografia sostitutiva
- la steganografia selettiva
- la steganografia costruttiva

3.2 – Steganografia Sostitutiva

Le prime tecniche steganografiche importanti di cui esamineremo le varie sfaccettature sono le tecniche Sostitutive. Queste tecniche sono le più usate in questo campo e in genere ci si riferisce alla steganografia proprio per indicare questo tipo di variante. Tali tecniche si basano sul fatto che la maggior parte dei canali di comunicazione (linee telefoniche, trasmissioni radio, ecc.) trasmettono segnali che sono sempre accompagnati da qualche tipo di rumore. Questo rumore può essere sostituito da un segnale (il messaggio segreto) che è stato trasformato in modo tale che, a meno di conoscere una chiave segreta, è indistinguibile dal rumore vero e proprio, e quindi può essere trasmesso senza destare sospetti. Quasi tutti i programmi si basano su questa idea, sfruttando la grande diffusione di file contenenti una codifica digitale di immagini, animazioni e suoni; spesso questi file sono ottenuti da un processo di conversione

analogico/digitale e contengono qualche tipo di rumore. Per esempio un'immagine prodotta da uno scanner è soggetta a essere affetta da errore.

L’idea base impiegata dalla maggior parte dei programmi consiste semplicemente nel sostituire i "bit meno significativi", o per meglio dire i bit ridondanti presenti nei cover object, con i bit che costituiscono il file segreto. Naturalmente i bit che devono essere sostituiti devono essere scelti in modo da non provocare variazioni evidenti al cover object e quindi evitare di compromettere l’integrità del contenitore.

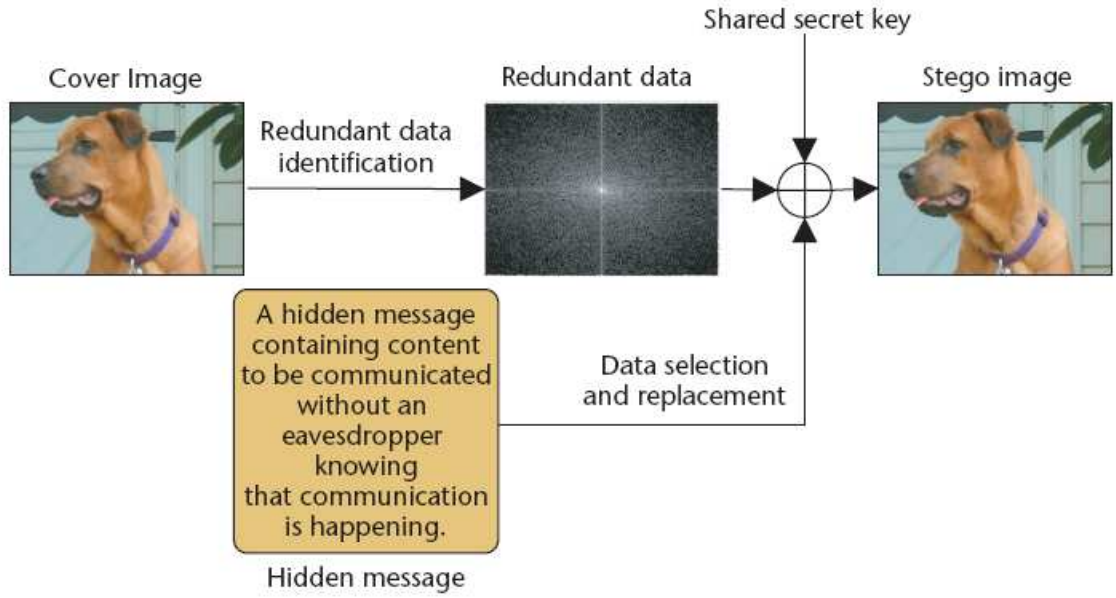


Figura : ***Esempio di Steganografia Sostitutiva [9]***

Spesso l'immagine che ne risulta non è distinguibile a occhio nudo da quella originale ed è comunque difficile dire se eventuali perdite di qualità siano dovute alla presenza di informazioni nascoste oppure all'errore causato dall'impiego di uno scanner poco preciso, o ancora alla effettiva qualità dell'immagine originale prima di essere digitalizzata.

Un esempio in cui viene solitamente rappresentata un'immagine contenitore prodotta da uno scanner è la codifica RGB a 24 bit: l'immagine consiste di una matrice M x N di punti colorati (pixel) e ogni punto è rappresentato da 3 byte, che indicano rispettivamente i livelli dei colori rosso, verde e blu che costituiscono il colore.

Supponiamo che uno specifico pixel di un'immagine prodotta da uno scanner sia rappresentato dalla tripla (12, 241, 19) (si tratta di un colore tendente al verde, dato che la componente verde predomina fortemente sulle altre due); in notazione binaria, le tre componenti sono:

==========

12 = 00001100

241 = 11110001

19 = 00010011

==========

quelli che in precedenza abbiamo chiamato i "bit meno significativi" dell'immagine sono gli ultimi a destra, cioè 0 per il primo valore,1 per il secondo e il terzo, e sono proprio quelli che si utilizzano per nascondere il messaggio segreto. Se volessimo nascondere in quel pixel l'informazione data dalla sequenza binaria 101, allora bisognerebbe effettuare la seguente trasformazione:

==========

00001100 --> 00001101 = 13

11110001 --> 11110000 = 240

00010011 --> 00010011 = 19

==========

La tripla è così diventata (13, 240, 19); si noti che questo tipo di trasformazione consiste nel sommare 1, sottrarre 1 o lasciare invariato ciascun livello di colore, quindi il colore risultante differisce in misura minima da quello originale. Dato che un solo pixel può contenere un'informazione di 3 bit, un'immagine di dimensioni M x N può contenere un messaggio segreto lungo fino a (3*M*N)/8 byte, per esempio

un'immagine 1024x768 può contenere 294912 byte. Un esempio visivo di questo tipo di algoritmo è la figura sottostante, dove si può vedere un particolare dell'immagine della Gioconda normale e subito dopo essere stata steganografata. I piccoli punti in grigio (evidenziati con i cerchi rossi) che appaiono, rappresentano i byte inseriti ma si notano solo con un ingrandimento dell'immagine e possono essere facilmente scambiate per rumore.

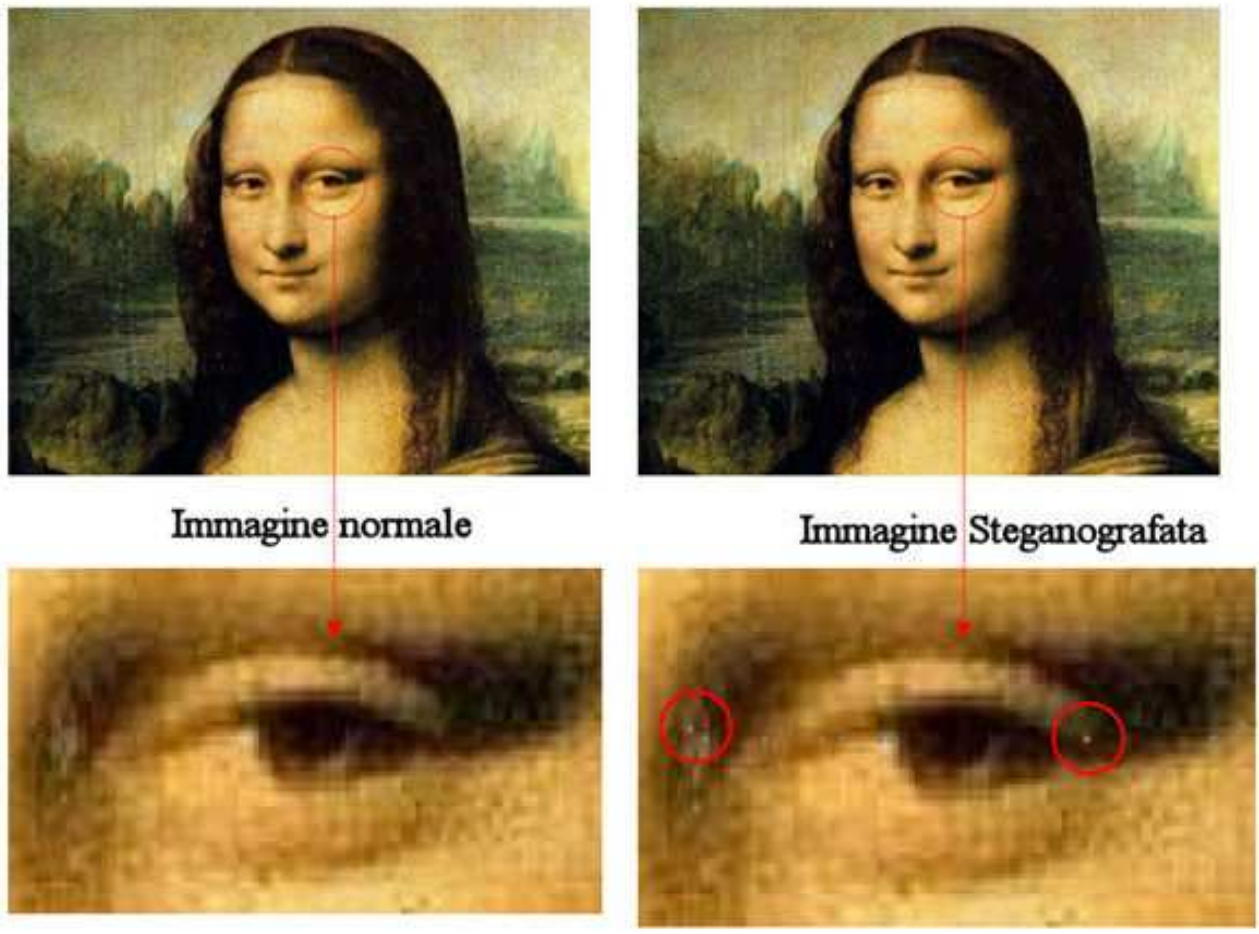


Figura : ***Gioconda Steganografata [5]***

In sintesi, quindi, abbiamo visto quelli che possono essere gli utilizzi della steganografia sostituiva , ma è chiaro come al posto di una semplice immagine di uno scanner ci possa essere qualsiasi altro contenitore adatto allo scopo (musica, video ecc..). A questo proposito ci sono almeno un paio di regole che è giusto osservare quando si cerca di creare un algoritmo efficace :

- Cercare di evitare di utilizzare immagini o file che siano ad accesso comune come da siti pubblici o comunque cose facilmente reperibili (immagini incluse in pacchetti software, video su youtube ecc..)
- Cercare di evitare di utilizzare sempre gli stessi contenitori per non dare modo ad eventuali controlli di trovare facilmente i cover object in questione. È consigliabile quindi variare spesso i propri contenitori attraverso l'utilizzo di scanner e convertitori da analogico a digitale e distruggere gli originali dopo averli usati.

Si capisce come utilizzare, quindi, un contenitore di scarsa importanza (o nulla) per queste tecniche è molto vantaggioso per nascondere al suo interno messaggi di una certa segretezza. Più il contenitore dà poco nell'occhio e crea poco sospetto e più si è sicuri che il messaggio segreto arrivi al destinatario senza problemi. Un problema che però può portare al fallimento questo tipo di tecnica steganografia è proprio intrinseco alla sostituzione di informazioni all'interno del contenitore. Infatti, spesso, al cover object si provoca un certo tipo di cambiamento del rumore di quest'ultimo che può far insospettire chi eventualmente controlla il messaggio. Lo scenario è il seguente: si suppone che il nemico/controllore disponga di un modello del rumore e che utilizzi tale modello per controllare i file che riesce a intercettare. Se il rumore presente in un file non è conforme al modello, allora il file è da considerarsi sospetto. Si può osservare che questo tipo di attacco non è per niente facile da realizzare, data l'impossibilità pratica di costruire un modello che tenga conto di tutte le possibili sorgenti di errori/rumori, tuttavia in proposito esistono degli studi che in casi molto specifici hanno avuto qualche successo. Per cercare di risolvere quest'ultimo problema vengono in aiuto della steganografia sostitutiva le altre due diverse tecniche steganografiche, ovvero, la steganografia selettiva e quella costruttiva.

3.3 – Steganografia Selettiva

La steganografia selettiva oggigiorno rimane ancora ad un livello teorico e non ci sono ancora stati tentativi seri per creare un algoritmo che rispecchi questo tipo di tecnica. Il motivo per cui non si è passati dalla teoria alla pratica è dovuto al fatto che questa tecnica, anche risolvendo il problema della steganografia sostitutiva, non permette di avere un guadagno superiore alle spese richieste. Infatti al costo esorbitante si contrappone una scarsa quantità di informazione che è possibile nascondere. Comunque sia, l'idea base su cui si fonda il tutto, è quella di procedere per tentativi fino a quando non si verifica una certa condizione. Facciamo un esempio per chiarire meglio. Si fissi una funzione hash semplice da applicare ad un'immagine in forma digitale (Guarda definizione di funzione Hash); per semplificare al massimo, assumiamo che la funzione valga 1 se il numero di bit uguali a 1 del file che rappresenta l'immagine è pari, altrimenti valga 0 (si tratta di una assunzione davvero poco realistica ma, come dicevamo, questa discussione ha valore esclusivamente teorico). Così, se vogliamo codificare il bit 0 procediamo a generare un'immagine con uno scanner; se il numero di bit dell'immagine uguali a 1 è dispari ripetiamo di nuovo la generazione, e continuiamo così finché non si verifica la condizione opposta. Il punto cruciale è che l'immagine ottenuta con questo metodo contiene effettivamente l'informazione segreta, ma si tratta di un'immagine "naturale", cioè generata dallo scanner senza subire alcuna manipolazione successiva. L'immagine è semplicemente sopravvissuta a un processo di selezione (da cui il nome della tecnica), quindi non si può dire in alcun modo che le caratteristiche statistiche del rumore presentino una distorsione rispetto ad un modello di riferimento. È ovvio che sul piano pratico questa soluzione è inaccettabile perché è molto dispendiosa in termini di tempo ed oltretutto permette di nascondere una quantità d'informazione molto modesta.

3.4 – Steganografia Costruttiva

La steganografia costruttiva affronta il problema citato poco fa nel modo più diretto, tentando di sostituire il rumore presente nel contenitore utilizzato con l'informazione segreta opportunamente modificata in modo da imitare le caratteristiche statistiche del rumore originale. Secondo questa concezione, un buon sistema steganografico dovrebbe basarsi su un modello del rumore e adattare i parametri dei suoi algoritmi di codifica in modo tale che il falso rumore contenente il messaggio segreto sia il più possibile conforme al modello. Essendo comunque un approccio valido presenta alcuni svantaggi che è meglio esaminare. Innanzitutto, non è facile costruire un modello del rumore: la costruzione di un modello del genere richiede grossi sforzi ed è probabile che qualcuno, in grado di disporre di maggior tempo oppure di risorse migliori, riesca a costruire un modello più accurato, riuscendo ancora a distinguere tra il rumore originale ed un suo sostituto. Inoltre, se il modello del rumore utilizzato dal metodo steganografico dovesse cadere nelle mani del nemico, questi potrebbe analizzarlo per cercarne possibili difetti e quindi utilizzare proprio il modello stesso per controllare che un messaggio sia conforme ad esso. Così, il modello, che è parte integrante del sistema steganografico, fornirebbe involontariamente un metodo di attacco particolarmente efficace proprio contro il sistema stesso.

3.5 – Il cover object ideale

Si è detto nei paragrafi precedenti che diversi tipi di file possono essere dei contenitori ideali per un qualsivoglia messaggio segreto. Abbiamo parlato di immagini, file audio, video e aggiungiamo anche file system e pagine html. Un esempio curioso di questo utilizzo della steganografia in molteplici contenitori è quella che riguarda il ministro inglese Margaret Thatcher che, preoccupata per il continuo accrescere di fughe

di notizie dai suoi ambienti più vicini, fece programmare i loro word processor in modo che il nome dello scrivente fosse codificato nella spaziatura delle parole. La domanda che ora sorge spontanea è questa : come mai ci sono talmente tanti tipi di contenitori e oggigiorno ci si ostina ancora ad utilizzare le immagini come contenitore ideale?

La risposta è semplice e ci arriveremo aiutandoci con l'esempio di Alice e Bob redatto da G.J. Simmons di cui abbiamo parlato nel capitolo 2.4. Ricordiamo che Alice e Bob avevano il problema di comunicare liberamente tra loro perché avevano il secondino Wendy che controllare i loro messaggi. Ora, Alice e Bob non avrebbero potuto scegliere come contenitore del loro messaggio segreto un file facile da trovare in circolazione (ad esempio loghi, foto di attori famosi, hit musicali ecc…) poiché la stessa Wendy avrebbe potuto venirne in possesso facilmente. In quel caso avrebbe fatto subito a capire che i messaggi scambiati dai due erano comunque sospetti, gli bastava solamente confrontare i file e avrebbe soppresso la comunicazione tra i due.

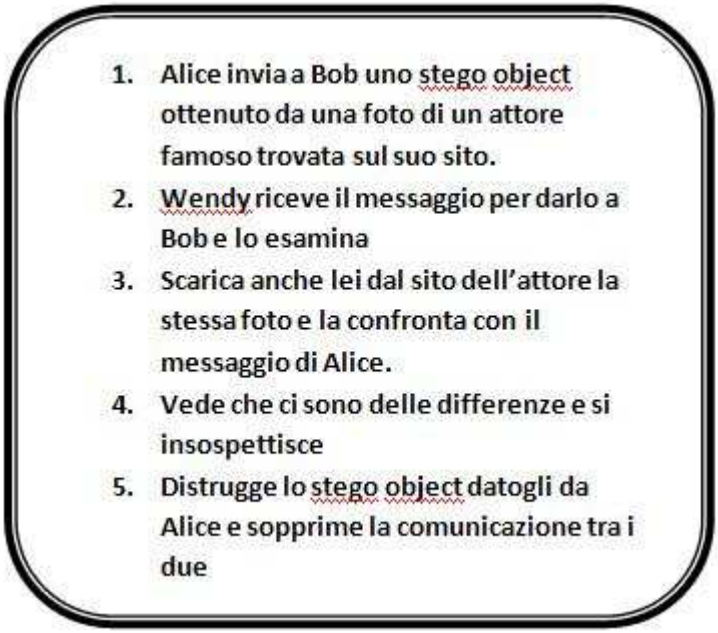


Figura : ***Scenario in cui si sceglie un cover object di facile reperibilità***

Si capisce, quindi, che l'utilizzo di cover object inediti, o per lo meno non di accesso pubblico, è la base di partenza migliore per avere una comunicazione sicura.

Ad esempio basta scattare una foto sul momento e diventerà un ottimo cover object, oppure utilizzare musica poco conosciuta, utilizzare una pagina html cambiando ciò che ci è utile (produrre errori ortografici,frasi senza senso,utilizzare gli spazi in file testuali o alcuni tag dei file html [10]). Le immagini, soprattutto quelle compresse, contengono del rumore difficilmente modellabile, dal momento che la compressione usata per alleggerirli rende più complicato capire se ci sono state distorsioni di altro tipo; in più, il web rappresenta un comodo canale di trasmissione per esse (difatti, molti sono i siti che consentono di pubblicare immagini in maniera anonima) e l'enorme diffusione delle immagini, sia in Internet sia come allegati in e-mail, le rende difficilmente sospettabili e poco realisticamente controllabili singolarmente. Se si hanno particolari esigenze di sicurezza, esiste la possibilità di scegliere cover object con una capacità molto superiore alle proprie esigenze. Distribuire i bit del messaggio segreto su un volume maggiore di dati permette di diluire i mutamenti da effettuare, rendendo praticamente impossibile compiere un attacco al sistema steganografico. Le immagini si prestano naturalmente a questo fine: invece di utilizzare i bit meno significativi di tutti i pixel di un'immagine, si può scegliere di modificarne solo uno ogni venti.

In conclusione, le immagine compresse sono il miglior tipo di cover, oggi, per effettuare trasmissioni steganografiche.

4- LA STEGANALISI

4.1 – Descrizione e tipi di attacco

La steganalisi è l'insieme dei metodi e delle tecniche capaci di attaccare un sistema steganografico. Per la Crittografia c'è la crittoanalisi che ha il compito di svelare l'output cifrato di un sistema crittografico, mentre per la Steganografia esiste la Steganalisi che ha lo scopo di esaminare un messaggio e di etichettarlo con certezza come sospetto (contenente, cioè, al suo interno un messaggio occulto) o come innocuo (privo di informazioni nascoste).

In altre parole, "dove nella crittoanalisi il fine è quello di rivelare il dato, nella steganalisi il fine è quello di scoprirne la presenza" [12]. Fare steganalisi è intrinsecamente un compito complesso poiché, spesso, non si è in possesso di tutte le informazioni necessarie per compiere un esame che possa portare ad esiti certi. Si tratta, molte volte, di indovinare i passi che avrebbe potuto compiere il mittente del presunto messaggio steganografico, dove con indovinare si intende cercare il cammino più probabile scelto dal mittente nell'insieme delle mosse possibili.

In letteratura, si suole suddividere le varie procedure di attacco steganalitico specificando il tipo di informazioni che si posseggono. Ecco quindi un immagine riassuntiva che descrive a pieno tutte le tipologie di attacco e il momento giusto in cui vanno a intervenire nella comunicazione.

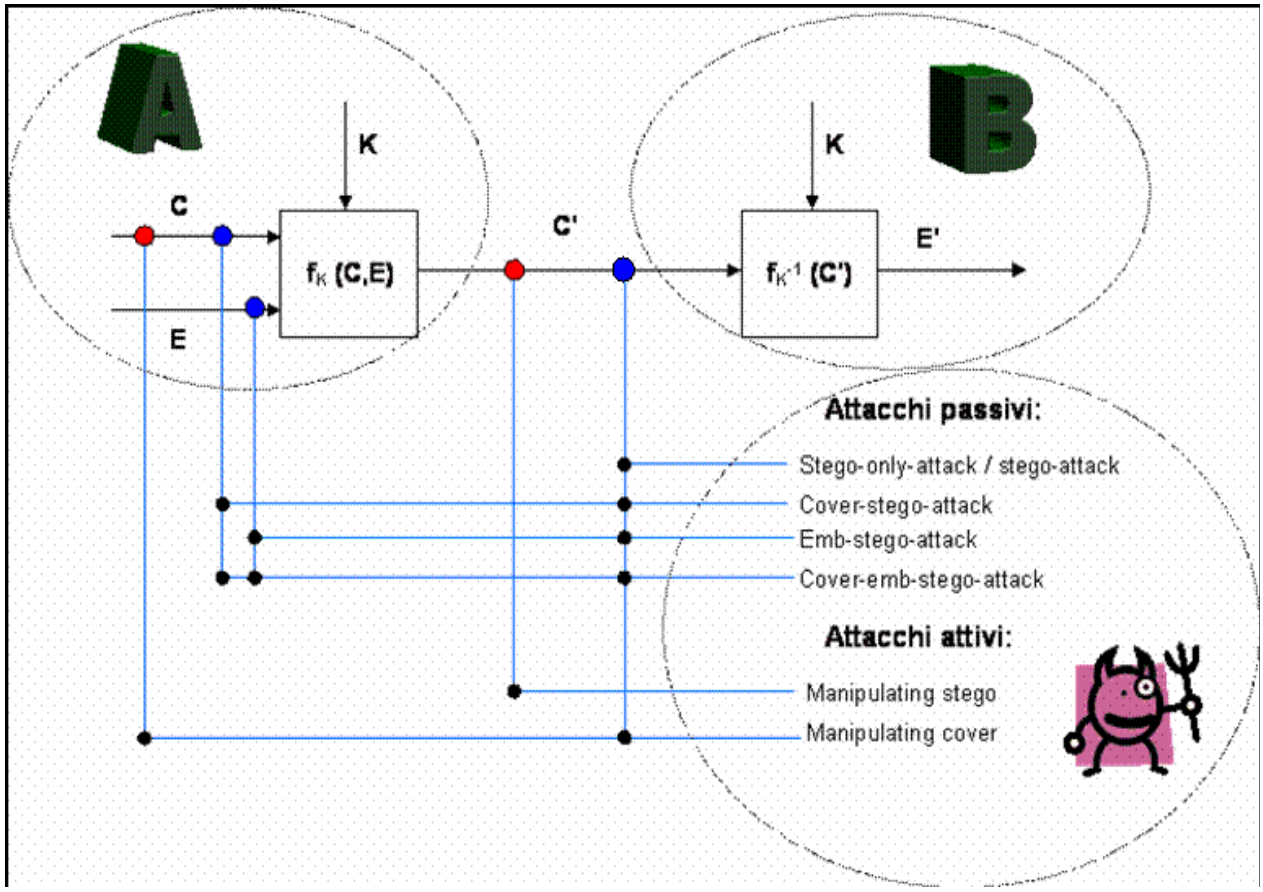


Figura : ***Schema attacchi steganografici***

- **Stego-only-attack** : l'attaccante ha intercettato il frammento stego ed è in grado di analizzarlo. è il più importante tipo di attacco contro il sistema steganografico perché è quello che occorre più di frequente nella pratica.

- **Stego-attack** : il mittente ha usato lo stesso cover ripetutamente per nascondere dati diversi. L'attaccante possiede un frammento stego diverso ma originato dallo stesso cover. In ognuno di questi frammenti stego è nascosto un diverso messaggio segreto e quindi si capisce che il cover è stato alterato.

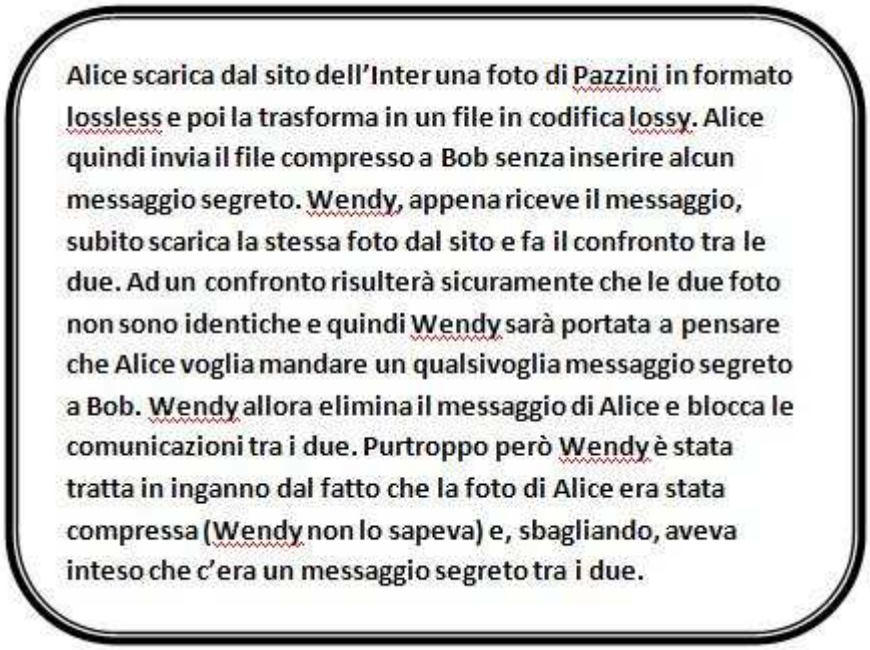
- **Cover-stego-attack** : l’attaccante ha intercettato il frammento stego e sa quale cover è stato usato per crearlo. Ciò fornisce abbastanza informazioni all'attaccante per poter risalire al messaggio segreto. È il caso riassunto nel capitolo precedente, dove Wendy possiede sia lo stego object sia il cover object. 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.

- **Cover-emb-stego-attack** : l’attaccante ha "tutto" poiché ha intercettato il frammento stego, conosce il cover usato e il messaggio segreto nascosto nel frammento stego.

- **Manipulating the stego data** : l’attaccante è in grado di manipolare i frammenti stego. Il che significa che l’attaccante può togliere il messaggio segreto dal frammento stego (inibendo la comunicazione segreta).

- **Manipulating the cover data** : l’attaccante può manipolare il cover e intercettare il frammento stego. Questo può significare che con un processo più o meno complesso l’attaccante può risalire al messaggio nascosto.

Particolare attenzione va riposta nel caso del cover-stego-attack, poiché lo stego object che si intercetta potrebbe essere la versione compressa del cover. Esponiamo uno scenario molto semplificato, ma che potrebbe realmente verificarsi.



Questo esempio, nella sua banalità, mette in luce quanto sia difficile analizzare uno stego object da parte di una qualsiasi entità. Infatti, qualsiasi immagine che sia il risultato di una conversione analogico/digitale (come le immagini scattate con una macchinetta digitale), così come qualsiasi immagine compressa, contiene del rumore molto difficile da predire poiché legato ad una moltitudine di variabili (se la macchina fotografica è di scarsa qualità potrebbe inserire nell'output disturbi non ipotizzabili da nessun modello del rumore generale).

Al momento esistono diverse tecniche steganalitiche applicate con successo a diversi software steganografici disponibili in commercio [11][13][14][15]. Esse verranno presentate nei prossimi capitoli che trattano proprio gli approcci già noti alla steganografia sulle immagini.

5- STEGANOGRAFIA NEI FILE IMMAGINE

Questo capitolo offre un' infarinatura generale sulle tecniche steganografiche a nostra disposizione oggigiorno che intervengono su contenitori di tipo immagine. Il materiale seguente è ispirato a diversi articoli, i quali costituiscono una guida completa dei metodi disponibili oggi e delle tecniche per attaccarli con successo.

5.1 – File Immagini

Le immagini sui computer sono viste come array di numeri che rappresentano l'intensità della luce in vari punti. Questi punti si chiamano pixel e formano il raster data dell'immagine. Un'immagine comune ha dimensioni pari a 640 x 480 pixel e 256 colori (o 8 bit per pixel). Le immagini digitali sono tipicamente salvate in formati da 24 bit o da 8 bit per pixel. È chiaro quindi che, rispetto ad una immagine ad 8 bit, un'immagine a 24 bit fornisca uno spazio molto maggiore per nascondere l'informazione: tale spazio può essere abbastanza ampio per inserire un testo molto lungo (il nostro messaggio segreto). In generale, tutte le variazioni di colore per i pixel sono derivate dai tre colori primari: rosso (red), verde (green) e blu (blue). Ogni colore primario è rappresentato da 1 byte: le immagini a 24 bit usano 3 bytes per pixel per rappresentare un valore del colore. Questi 3 bytes possono essere rappresentati in valori esadecimali, decimali e binari. È chiaro come la rappresentazione dei pixel si ripercuota anche sulla dimensione dell'immagine stessa.

Ad esempio se si ha un immagine da 24 bit, con formato 1024 x 768 pixel, avrà più di 2 milioni di pixel, ognuno dei quali con una definizione tale da produrre un immagine di oltre 2 Mbytes. Molte volte, per evitare di trasmettere immagini cosi grandi , si utilizzano delle forme di compressione del file che danno sicuramente dei vantaggi.

5.2 – Forme di compressione

Le tecniche di compressione dati si dividono in due grandi categorie:

- **compressione dati lossless**: comprime i dati attraverso un processo senza perdita d'informazione che sfrutta le ridondanze nella codifica del dato.
- **compressione dati lossy**: comprime i dati attraverso un processo con perdita d'informazione che sfrutta le ridondanze nell'utilizzo dei dati;

Le tecniche senza perdita (lossless) consentono di preservare l'informazione originale in ogni sua parte. È l'unica via possibile quando si devono comprimere file di testo, programmi, documenti, database, schemi elettrici ecc. Quindi è il miglior modo di comprimere i file quando si tratta di usare tecniche steganografiche, poiché non vanno a inficiare su quelli che possono essere i rumori di fondo. Un esempio di tecnica lossless è il formato ZIP, il quale consente di archiviare o trasmettere uno o più file risparmiando sulle risorse necessarie (spazio su disco o tempo di trasmissione). Al momento in cui vengono recuperati i file dallo ZIP (decompressione) questi risultano indistinguibili dagli originali. Un altro esempio di caso in cui viene usata la compressione senza perdita è quello delle immagini non fotografiche, come gli schemi, i disegni o le icone. Per questo scopo esistono formati come il GIF , BMP o il più recente PNG. L'immagine compressa con uno di questi formati mantiene esattamente l'aspetto originale fino al dettaglio più insignificante.

D'altro canto, le tecniche con perdita di informazione (lossy) permettono anche delle compressioni molto spinte, quindi un grande risparmio di risorse, a discapito però della qualità dell'immagine o dell'audio che si è voluto comprimere. Sì perché generalmente queste tecniche si usano per comprimere i file multimediali. Pur mantenendo minima la perdita di qualità, il risparmio rispetto ad una compressione lossless sulla stessa informazione è sempre decisamente apprezzabile. Le informazioni multimediali come audio o video, in origine sono troppo grandi per essere agevolmente trasmesse o

memorizzate, quindi si preferisce avere una piccola riduzione della qualità ma nel contempo file molto più leggeri.

È chiaro quindi come questo tipo di compressione crea dei problemi alle tecniche steganografiche rispetto a quello lossless. Alcuni esempi sono la compressione di immagini in formato JPEG, largamente usata in fotografia digitale e sul Web, la compressione video in formato XviD oppure la compressione audio in formato MP3.

5.3 – Steganografia applicata alle immagini

Come abbiamo detto già in precedenza, il processo di inserimento dei dati segreti (embedding data) in una immagine richiede due file :

- Il primo file richiesto è l'immagine non sospetta che conterrà l'informazione nasconta, comunemente chiamata cover object.
- Il secondo file richiesto è il messaggio segreto che bisognerà nascondere nel precedente. Il messaggio può essere un plaintext (vedi), un cyphertext (vedi), un'altra immagine o qualsiasi altra cosa che possa essere contenuta in un flusso di bit.

Una volta combinati, la cover image ed il messaggio in essa inserito formano uno stego object. In aggiunta a questo può esserci una stego key, ovvero una sorta di password che può essere usata per nascondere, e poi successivamente decodificare, il messaggio.

5.4 – Formato immagine GIF

Il primo esempio che esamineremo per quanto riguarda la steganografia applicata alle immagini sarà quello basato su palette (tavolozza), che è in sintesi un sottoinsieme prestabilito di colori da poter utilizzare. Nei formati che ne fanno uso (un esempio è il formato GIF), i pixel della bitmap sono vincolati ad assumere come valore uno dei colori presenti nella palette: in questo modo è possibile rappresentare i pixel con dei puntatori alla palette, invece che con la terna esplicita RGB (red, green e blue) che abbiamo esaminato negli scorsi capitoli. Ciò in genere permette di ottenere dimensioni inferiori della bitmap, ma il reale vantaggio è dato dal fatto che le schede grafiche di alcuni anni fa utilizzavano proprio questa tecnica e quindi non potevano visualizzare direttamente immagini con un numero arbitrario di colori. Il caso più tipico è quello delle immagini in formato GIF con palette di 256 colori, ma le palette possono avere anche altre dimensioni. Come è facile immaginare, un'immagine appena prodotta da uno scanner a colori sarà tipicamente costituita da più di 256 colori diversi, tuttavia esistono algoritmi capaci di ridurre il numero dei colori utilizzati mantenendo il degrado della qualità entro limiti accettabili. Si può osservare che, allo stesso modo in cui avviene con il formato JPEG, non è possibile iniettare informazioni sui pixel prima di convertire l'immagine in formato GIF, perché durante il processo di conversione c'è perdita di informazione (osserviamo, anche, che questo non vale per le immagini a livelli di grigi: tali immagini infatti sono particolarmente adatte per usi steganografici).

La soluzione che viene di solito adottata per usare immagini GIF come contenitori è dunque la seguente: si riduce il numero dei colori utilizzati dall'immagine a un valore inferiore a 256 ma ancora sufficiente a mantenere una certa qualità dell'immagine, dopodiché si finisce di riempire la palette con colori molto simili a quelli rimasti. A questo punto, per ogni pixel dell'immagine, la palette contiene più di un colore che lo possa rappresentare (uno è il colore originale, gli altri sono quelli simili ad esso che sono stati aggiunti in seguito), quindi abbiamo una possibilità di scelta. Tutte le volte che esiste una possibilità di scelta fra più alternative, è possibile

nascondere un'informazione: questo è uno dei principi fondamentali della steganografia. Se le alternative sono due possiamo nascondere un bit (se il bit è 0, scegliamo la prima, se è 1 la seconda); se le alternative sono quattro possiamo nascondere due bit (00 → la prima, 01 → la seconda, 10 → la terza, 11 → la quarta) e così via. La soluzione appena discussa dell'utilizzo di GIF come contenitori è molto ingegnosa ma purtroppo presenta un problema: è facile scrivere un programma che, presa una GIF in ingresso, analizzi i colori utilizzati e scopra le relazioni che esistono tra di essi; se il programma scopre che l'insieme dei colori utilizzati può essere ripartito in sottoinsiemi (cluster) di colori simili, è molto probabile che la GIF contenga informazione steganografata. Di fatto, questo semplice metodo di attacco (che potremmo definire, giusto per intenderci, clusterizzazione) è stato portato avanti con pieno successo in [14], tanto da rendere poco sicuro, in generale, l'uso di immagini GIF come cover object.

Per mostrare quanto sia ampia la gamma di tecniche steganografiche, accenniamo a un'altra possibilità di nascondere informazioni dentro immagini GIF. Come abbiamo detto, in questo formato viene prima memorizzata una palette e poi la matrice bitmap (compressa con l'algoritmo LZW [vedi def.]) dei puntatori alla tavolozza. Se scambiamo l'ordine di due colori della palette e contemporaneamente tutti i puntatori ad essi, otteniamo un file diverso che corrisponde però alla stessa immagine: dal punto di vista dell'immagine, il contenuto informativo dei due file è identico. La rappresentazione di immagini con palette è quindi intrinsecamente ridondante, dato che ci permette di scegliere un qualsiasi ordine dei colori della palette (purché si riordinino correttamente i puntatori ad essi). Se i colori sono 256, esistono 256! modi diversi di scrivere la palette, quindi esistono 256! file diversi che rappresentano la stessa immagine. Inoltre, è abbastanza facile trovare un metodo per numerare univocamente tutte le permutazioni di ogni data palette (basta, per esempio, considerare l'ordinamento sulle componenti RGB dei colori). Dato che abbiamo 256! possibilità di scelta, è possibile codificare log(256!) = 1683 bit, cioè 210 byte. Si noti che questo numero è indipendente dalle dimensioni dell'immagine, in altre parole è possibile iniettare 210

bytes anche su piccole immagini del tipo icone 16 × 16 semplicemente permutando in modo opportuno la palette.

Un altro esempio di come possano essere nascoste informazioni segrete nelle immagini di formato GIF ci è dato dal software EzStego, sviluppato da Romana Machado. Questo software inserisce il messaggio segreto in un file GIF, senza alcuna informazione sulla sua lunghezza. EzStego non modifica la tavolozza dei colori, ma crea invece una copia ordinata della palette.

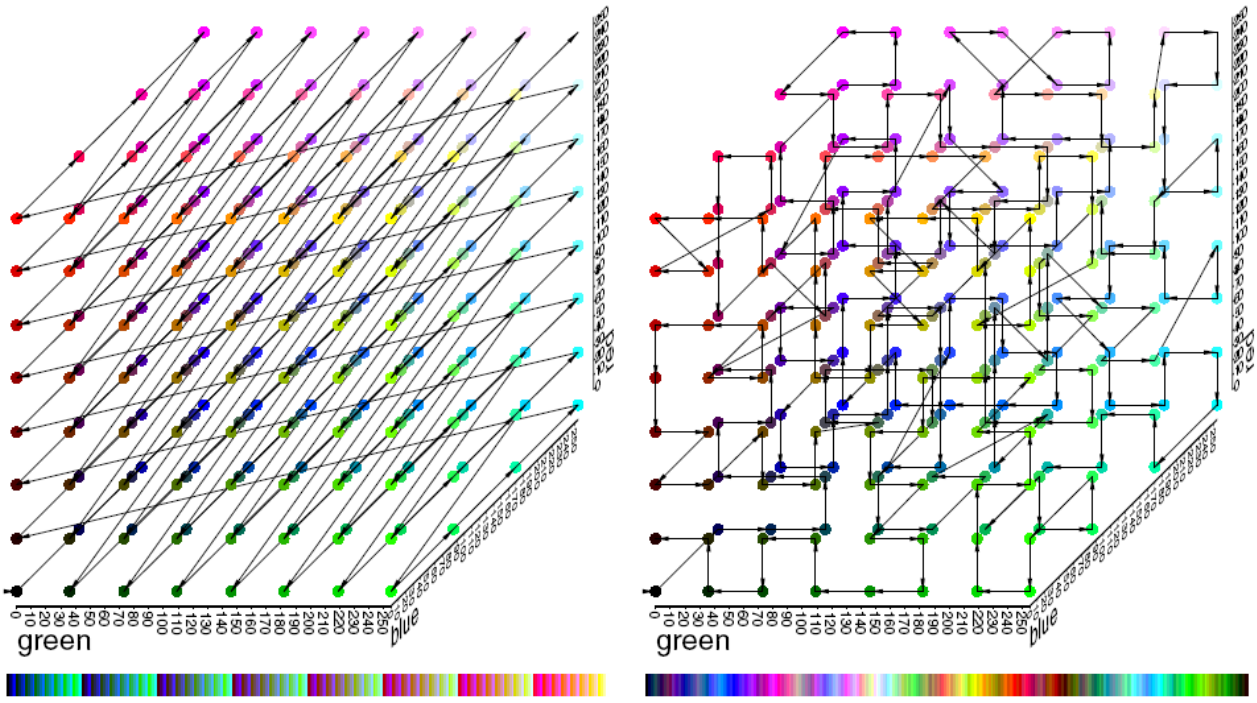


Figura : ***Ordine dei colori nella palette (a sinistra) e ordine dei colori fatto da EZStego (a destra), tratto da [19]***

Come si vede dall'immagine sopracitata , la copia di palette è costruita in modo che sia minima la differenza visiva tra due colori in essa adiacenti. Ordinare in base ai soli valori di luminanza è sconsigliabile, perché due colori con la stessa luminanza potrebbero essere radicalmente diversi alla vista. Si può immaginare ogni colore come un punto in uno spazio tridimensionale, il cubo RGB. Nella parte sinistra della figura, ovvero come viene rappresentato l'ordine numerico nella palette originale, i colori appaiono molto più ordinati rispetto alla parte destra. Sulla destra della figura i colori sono ordinati secondo l'algoritmo di EzStego, al fine di seguire il percorso più breve

attraverso il cubo RGB, percorso che corrisponde alla differenza visiva minima tra ogni valore di colore. La funzione di inserimento di EzStego lavora linea dopo linea, sulla sequenza di pixel dell'immagine a partire dall'angolo in alto a sinistra. Dopo l'embedding ogni pixel contiene un valore steganografico (cioè, un bit del messaggio). Tale valore steganografico di un pixel è rappresentato dal bit meno significativo dell'indice che quel pixel ha nella palette ordinata. La funzione di embedding confronta il valore da inserire (il bit del messaggio) con il LSB (Last Significant Bit) dell'indice del pixel, e sostituisce, se necessario, l'indice del pixel con il suo vicino nella tavolozza ordinata.

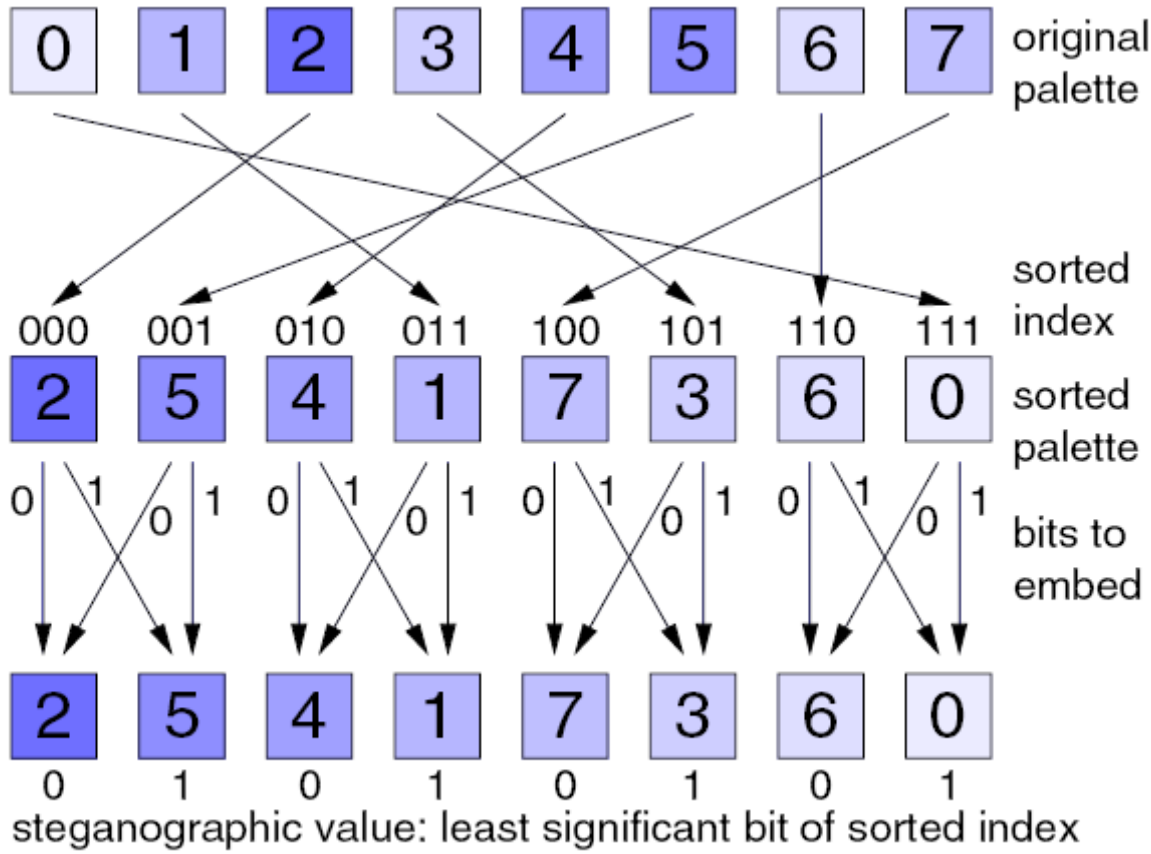


Figura : ***La funzione di embedding di EzStego, tratto da [19]***

La figura mostra la funzione di inserimento di EzStego con una palette ridotta. Per esempio, troviamo l'indice 2 per un dato pixel nel cover object. Se vogliamo inserire un 1, basta sostituire l'indice 2 con 5, mentre se vogliamo inserire uno 0 non

dobbiamo fare nulla (si osservi la parte bassa della figura). Poiché il colore dell'indice 2 nella palette originale si trova all'indice 000 (= 0) della tavolozza ordinata, ed il colore di indice (non ordinato) 5 si trova all'indice (ordinato) 001 (= 1), questi due colori sono molto simili, essendo adiacenti nella tavolozza ordinata e, quindi, sono visivamente quasi impossibili da distinguere. Un cambio dall'indice 2 all'indice 5 (e viceversa) è impercettibile ai nostri occhi, fino a che non si compara direttamente lo stego object con l'immagine cover. Uno dei limiti maggiori di EzStego risiede nella facile operazione (che chiunque può effettuare) di estrazione del messaggio originale. In breve, EzStego fa steganografia ma non crittografia. Inoltre, sebbene EzStego non soffra degli attacchi di clusterizzazione esso è molto sensibile agli attacchi visuali che descriveremo più avanti.

5.5 – Metodi per celare informazioni nelle immagini

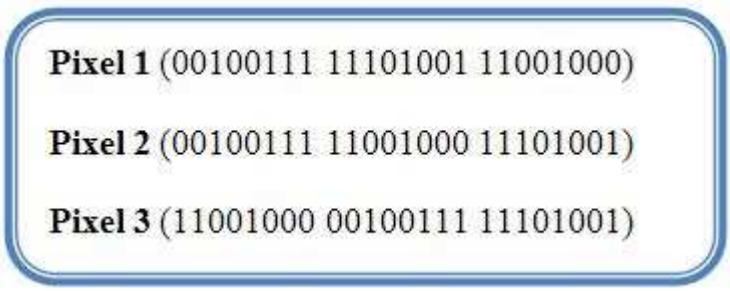
In questo paragrafo esamineremo quelle che sono le varie tecniche per nascondere informazioni nelle immagini. L'inserimento diretto di messaggi segreti può richiedere la codifica di ogni bit dell'informazione nell'immagine, oppure l'inserimento selettivo del messaggio in aree con rumore che attirino meno l'attenzione, come nel caso delle immagini, quelle aree in cui ci sia una grande variazione di colori. Il messaggio può anche essere sparpagliato casualmente all'interno dell' immagine come abbiamo visto negli esempi precedenti. I metodi più comuni per nascondere l'informazione in immagini digitali sono :

- least significant bit (LSB) insertion (inserimento nel'ultimo bit significativo)
- masking and filtering (mascheramento e filtraggio)
- algoritmi e trasformazioni

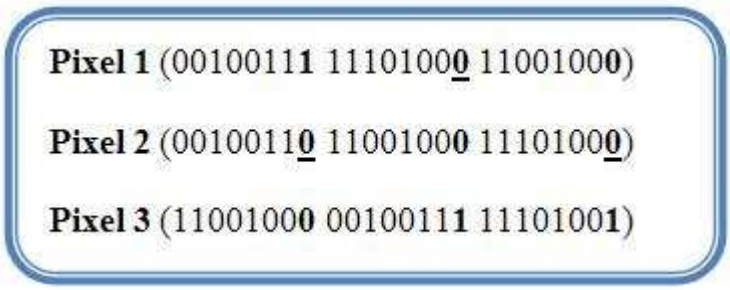
Ognuna di queste tecniche può essere applicata, con vari gradi di successo, a differenti file di immagini.

5.6 – Least significant bit insertion (LSB)

Il primo metodo che serve per l'embedding di informazioni in una cover object è l'inserimento nel LSB. Sfortunatamente, questo metodo è poco robusto anche rispetto ad una leggera manipolazione dell'immagine. La tecnica base impiegata dalla maggior parte dei programmi, consiste semplicemente nel sostituire i bit meno significativi delle immagini digitalizzate con i bit che costituiscono il file segreto. I bit meno significativi sono assimilabili ai valori meno significativi di una misura, cioè quelli che tendono a essere affetti da errori. Spesso l'immagine che ne risulta non è distinguibile a occhio nudo da quella originale ed è comunque difficile dire se eventuali perdite di qualità siano dovute alla presenza di informazioni nascoste oppure all'errore causato dall'impiego di uno scanner poco preciso, ad una macchina fotografica scadente o ancora alla effettiva qualità dell'immagine originale prima di essere digitalizzata. Convertire un'immagine da un formato come GIF o BMP, che ricostruisca esattamente il messaggio originale (compressione lossless), in un JPEG, che non lo ricostruisce esattamente (compressione lossy), e viceversa potrebbe distruggere l'informazione nascosta nei LSB. Per nascondere un'immagine nei LSB di ogni byte di una immagine a 24 bit, si possono usare i 3 bytes di ogni pixel. Una immagine 1024 × 768 può potenzialmente nascondere un totale di 2359296 bit (294912 byte) di informazione. Se si comprime il messaggio da nascondere prima di inserirlo, si può nascondere una maggiore quantità di informazione. Alla vista, lo stego object risultante apparirà identico al cover object utilizzato. Un esempio di utilizzo dell'algoritmo può essere quello di inserire la lettera "A" in 3 pixel, assumendo che non ci sia una compressione che possa cambiare i bit.



Questo rappresenta il raster data originale di 3 pixel (9 byte) di un cover object. La lettera “A” ha come valore binario 100000011 e inserita nel messaggio si otterrebbe :



I bit sottolineati sono gli unici 3 bit realmente modificati negli 9 byte usati. Come si può vedere anche nell’esempio, in media, la LSB insertion, richiede che soltanto metà dei bit in un’immagine vengano modificati. In più, si possono nascondere dati sia nel LSB sia nel secondo LSB e ancora l’occhio umano non sarebbe in grado di distinguere tra stego object e cover object. La tecnica appena applicata nell’esempio rappresenta il cuore della steganografia sostitutiva, anche se di fatto ne esistono numerose variazioni. Innanzitutto è ovvio che tutto quello che abbiamo detto vale non solo per le immagini, ma anche per altri tipi di media, per esempio suoni e animazioni digitalizzati. Inoltre lavorando con le immagini come file contenitori, non sempre si inietta l'informazione al livello dei pixel, ma si è costretti a operare su un livello di rappresentazione intermedio: è il caso, per esempio, delle immagini in formato JPEG, nel quale le immagini vengono memorizzate solo dopo essere state compresse con una tecnica che tende a preservare le loro caratteristiche visive piuttosto che l'esatta informazione contenuta nella sequenza di pixel. Se iniettassimo delle informazioni in una bitmap e poi la salvassimo in formato JPEG, le informazioni andrebbero perdute, poiché non sarebbe possibile ricostruire la bitmap originale. Per poter utilizzare anche

le immagini JPEG come contenitori, è necessario lavorare su una trasformazione da valori di colore a frequenze di colore, cosa che vedremo più avanti in dettaglio.

5.7 – Masking and Filtering

Altre tecniche che andremo ad analizzare sono quelle di masking e filtering. Queste tecniche vengono comunemente riservate per immagini a 24 bit e grayscale [vedi def.], nascondono informazioni marcando una immagine, in una maniera simile al watermarking della carta. Le tecniche di watermarking [vedi def.] possono essere applicate senza pericolo di distruzione dell'immagine dovuta alla compressione lossy poiché sono maggiormente integrate nell'immagine. Watermarks visibili non fanno parte della steganografia per definizione e la differenza tra loro è innanzitutto negli scopi. La steganografia tradizionale nasconde le informazioni, mentre i watermarks estendono l'informazione e diventano un attributo del cover object. I watermarks digitali possono includere informazioni quali copyright, ownership o licenze, come mostrato in figura.

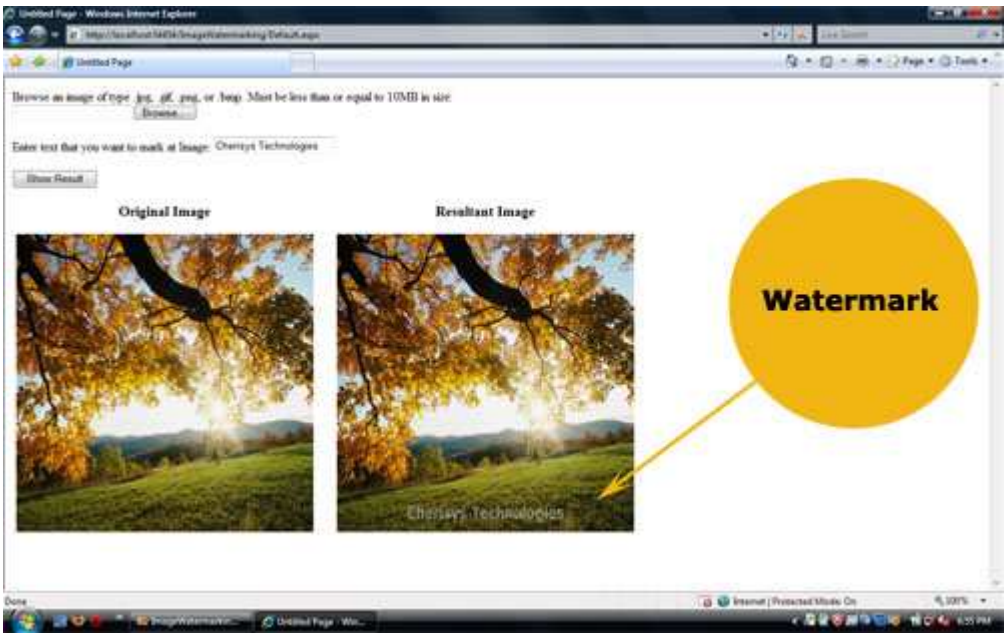


Figura : ***Esempio di watermarking***

Nella steganografia, l’oggetto della comunicazione è il messaggio nascosto, mentre nei watermarks digitali, l’oggetto della comunicazione è il cover object. Per creare la watermarked image in figura, si aumenta la luminosità della masked area del 15%. Se si dovesse incrementare la luminosità di una percentuale inferiore, l’occhio umano non sarebbe in grado di distinguere le cose. È ora possibile usare l’immagine con il watermark per nascondere il plaintext o informazioni codificate. Il masking è più robusto della LSB insertion rispetto alla compressione, al cropping e ad alcune operazioni di image processing. Le tecniche di masking inseriscono informazioni in aree significative in modo che il messaggio nascosto risulti maggiormente integrato con il cover object rispetto al semplice nasconderlo in un livello "rumoroso". Ciò rende, ad esempio, il masking più adatto della LSB insertion per immagini JPEG di tipo lossy.

5.8 – Algoritmi e trasformazioni

Parlando nei capitoli precedenti della manipolazione del LSB insertion abbiamo notato che è un modo semplice e veloce di nascondere informazioni, ma è poco robusto rispetto a piccoli cambiamenti risultanti da image processing o compressione lossy. Questo tipo di compressione è un vantaggio invece delle immagini JPEG rispetto ad altri formati: immagini ad alta qualità possono essere salvate in file relativamente piccoli usando metodi di compressione JPEG; proprio per questo le immagini JPEG si sono diffuse enormemente su Internet. Un tool steganografico che integra gli algoritmi di compressione per nascondere informazioni è Jsteg, a cui, vista la sua importanza, è dedicato un paragrafo apposito successivamente. Le immagini JPEG impiegano la trasformata discreta del coseno (DCT) per ottenere la compressione. DCT non è di per sé una trasformata lossy, però lo diventa se i valori del coseno sono calcolati con precisione numerica limitata, il che introduce errori di approssimazione nel risultato finale. Nello specifico JPEG è un algoritmo lossy specializzato nel togliere

informazione che comunque l'occhio non noterebbe nel cambiamento di una certa immagine. Ecco le varie fasi dell'algoritmo :

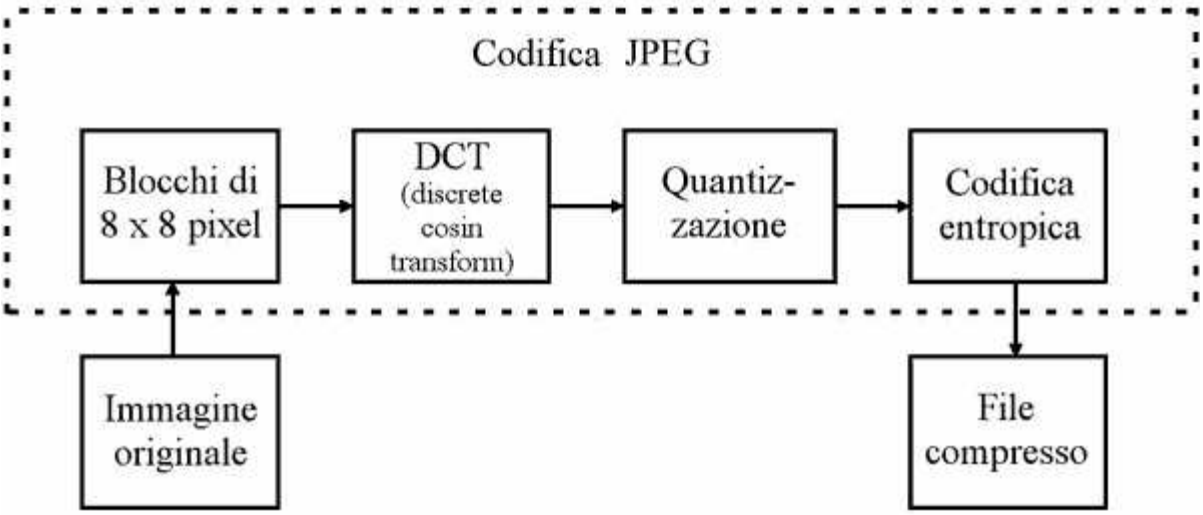


Figura : ***algoritmo JPEG***

Anche altre proprietà dell'immagine, come la luminosità, possono essere altresì manipolate. È il caso del patchwork [vedi def.] e tecniche simili che usano il "redundant pattern encoding" o metodi di diffusione spettrale per spargere l'informazione nascosta nelle cover images. Questi approcci possono favorire la protezione contro l'image processing, come ad esempio il cropping e la rotazione, nascondendo le informazioni meglio del semplice masking. Inoltre, supportano la manipolazione delle immagini più prontamente dei tools basati sulla LSB insertion. Nell'usare il redundant pattern encoding occorre effettuare un trade off fra la dimensione del messaggio e la robustezza.

Ad esempio, un messaggio di piccole dimensioni può essere riportato molte volte su una stessa immagine in modo che, se la stego image viene tagliata (cropping), c'è un'alta probabilità che il watermark resti ancora visibile. Un messaggio di grosse dimensioni, invece, può essere inserito solo una volta perché occuperebbe una porzione molto più grande dell'area dell'immagine.

Figura : ***Esempio di watermark con messaggio segreto di piccole dimensioni***

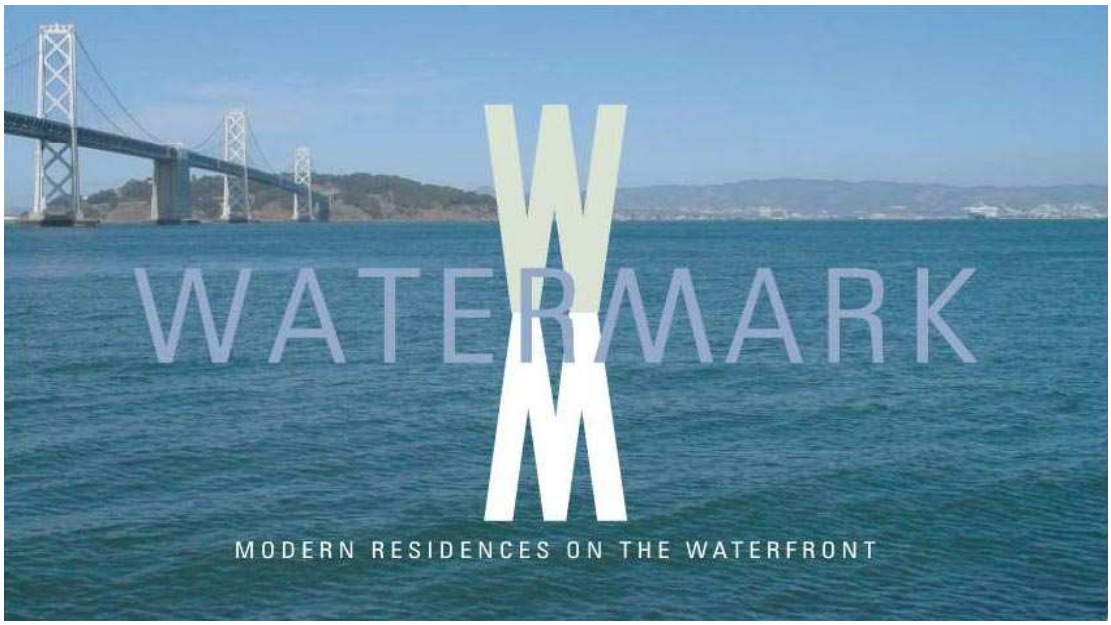


Figura : ***esempio di watermark con messaggio segreto di grosse dimensioni[20]***

Altre tecniche permettono di cifrare e disperdere (to scatter) i dati nascosti in un'immagine. La tecnica di scattering fa apparire il messaggio simile a del rumore. I fautori di questo approccio sostengono che anche se i bit del messaggio venissero estratti, sarebbero inutili senza l'algoritmo e la stego key per decodificarli. Ad esempio, il White Noise Tool è basato sulla tecnica di diffusione spettrale e dei salti in frequenza, che diffonde il messaggio attraverso l'immagine. Invece di avere x canali di comunicazione che sono modificati con una formula fissa ed una chiave, il White Noise Storm diffonde otto canali in un numero random generato da alcuni parametri quali dimensione della finestra di processamento e del canale dati. Ogni canale rappresenta 1 bit, cosicché ogni finestra d'immagine contiene 1 byte di informazione e molti bits inutilizzati. Questi canali ruotano, si scambiano e si sostituiscono tra loro per ottenere una diversa permutazione dei bit. Ad esempio, il bit 1 potrebbe essere sostituito dal bit 7, oppure entrambi i bit potrebbero ruotare di una posizione verso destra. Le regole di sostituzione sono imposte dalla stego key e dai dati random della precedente finestra. Le tecniche di scattering e di cifratura facilitano la protezione contro l'estrazione del messaggio nascosto, ma non contro la distruzione del messaggio tramite image processing. Un messaggio diffuso nei LSBs di una immagine è ancora poco robusto rispetto alla distruzione a causa della compressione lossy e dell'image processing, in quanto si tratta di un testo in chiaro inserito nei LSBs. Il ruolo classico della steganografia nella sicurezza è quello di affiancare la crittografia, non di sostituirla. Se un messaggio nascosto è criptato, una volta scoperto deve anche essere decriptato, il che fornisce un ulteriore livello di protezione.

5.9 – JSteg

Questo algoritmo, concepito da Derek Upham, è stato il primo sistema steganografico pubblicamente disponibile per immagini JPEG. L'algoritmo di embedding sostituisce sequenzialmente il LSB di ogni coefficiente DCT con i bit del messaggio segreto. Si tratta di un algoritmo estremamente semplice, che è riportato in figura.

```
Input: message, cover image
Output: stego image
while data left to embed do
        get next DCT coefficient from cover image
        if DCT ≠ 0 and DCT ≠ 1 then
                get next LSB from message
                replace DCT LSB with message LSB
        end if
        insert DCT into stego image
end while
```

Figura : ***Algoritmo JSteg***

Questo algoritmo non richiede una chiave segreta, per cui chiunque conosca il sistema steganografico utilizzato è in grado di recuperare, senza alcuna difficoltà, il messaggio nascosto da JSteg.

5.10 – OutGuess

OutGuess 0.1, creato da Niels Provos, modifica JSteg migliorando il passo di codifica con l'uso di un generatore di numeri pseudo-random per selezionare i coefficienti DCT da correggere. Il seme del generatore è una chiave condivisa dal mittente e il destinatario. Il LSB di ogni coefficiente prescelto viene modificato come con JSteg. Si veda la figura sottostante per i dettagli dell'algoritmo.

```
Input: message, shared secret, cover image
Output: stego image
initialize PRNG with shared secret
while data left to embed do
        get pseudo-random DCT coefficient from cover image
        if DCT ≠ 0 and DCT ≠ 1 then
            get next LSB from message
            replace DCT LSB with message LSB
        end if
        insert DCT into stego image
end while
```

*Figura : **Algoritmo OutGuess 0.1***

5.11 – Sistema steganografico sicuro e attacchi ad esso

Dopo avere esaminato alcune tecniche steganografiche di tipo sostitutivo, questo paragrafo tratta dei problemi relativi alla loro sicurezza. Prima, però, è utile ricordare che le norme che valgono generalmente per i programmi di crittografia dovrebbero essere osservate anche per l'utilizzo dei programmi steganografici. Per ciò che riguarda le specifiche caratteristiche della steganografia, si tengano presenti i seguenti principi:

- Evitare di utilizzare come contenitori di messaggi segreti immagini e file che siano di utilizzo comune e facilmente reperibili (immagini incluse in pacchetti software o siti importanti ecc..).
- Evitare di utilizzare sempre lo stesso file come cover object. La cosa migliore è sempre quella di creare ad hoc sempre nuovi contenitori difficilmente ritracciabili, convertirli da analogico a digitale attraverso mezzi specifici e distruggere gli originali per sicurezza.

Le tecniche sostitutive precedentemente viste consistono nel rimpiazzare un elemento di scarsa importanza (che risulti, in sostanza, ridondante) nel file cover con il messaggio segreto che vogliamo nascondere. Quello che viene ritenuto il principale difetto di tali tecniche è che in genere la sostituzione operata può alterare le caratteristiche statistiche del rumore presente nel media utilizzato. Un sistema steganografico è sicuro se è in grado di resistere agli attacchi steganalitici. Di questi concetti abbiamo già discusso nei paragrafi iniziali , ma era doveroso riportarli a mente per affrontare due temi interessanti. Il primo riguarda cosa contraddistingue un sistema steganografico sicuro e il secondo come si può battere la steganografia.

Per quanto riguarda il primo tema possiamo dire che la teoria dell'informazione ci permette di essere più specifici su cosa si intenda per sistema perfettamente sicuro. Sono stati proposti modelli effettivamente basati sulla teoria dell'informazione [vedi bib.] che, come si è già avuto modo di sottolineare, assumono che il guardiano sia passivo. Si assume, ancora, che l'avversario/controllore abbia conoscenza completa del sistema di codifica, ma che non conosca la chiave segreta condivisa da mittente e destinatario legittimo. Il controllore deve (vuole) costruire un modello per la distribuzione di probabilità Pc di tutti i possibili cover media e per la distribuzione Ps dei possibili stego media. L'avversario può, in questo modo, usare la detection theory per decidere tra due ipotesi :

- **Ipotesi C** : dove C sta ad indicare che lo stego object non contiene alcuna informazione occulta.
- **Ipotesi S** : dove S indica che lo stego object trasporta un messaggio segreto.

Quindi, date queste premesse, possiamo assumere che un sistema è perfettamente sicuro se non esiste alcuna regola di decisione tra le due ipotesi C ed S che sia più affidabile del semplice tirare ad indovinare. Precisiamo, se ce ne fosse bisogno, che mittente e destinatario sono in accordo sul sistema di codifica/decodifica e sulla chiave condivisa.

Passando invece al secondo tema in analisi , ovvero come si può battere un sistema steganografico, portiamo qui alcuni esempi di algoritmi oggigiorno applicati.

L’attacco visuale è uno dei più noti ed efficaci metodi per l’individuazione di contenuti steganografici all’interno di file in formati lossless. Questa tipologia di attacco, inventata da Westfeld e Pfitzmann [vedi bib.], ha modificato radicalmente la teoria steganografica perché ha dimostrato che l’assunzione relativa alla casualità del bit meno significativo di un pixel (cioè che l’insieme dei LSBs possa essere paragonato a rumore) è solo una leggenda metropolitana. L’idea degli attacchi visuali è quella di rimuovere tutte le parti dell’immagine che coprono il messaggio segreto. L’occhio umano può, a quel punto, distinguere se c’è la presenza di un potenziale messaggio, oppure se l’immagine è innocua. La rimozione avviene attraverso un filtraggio, diverso a seconda del tool steganografico usato per l’embedding, ed ha la struttura in figura :

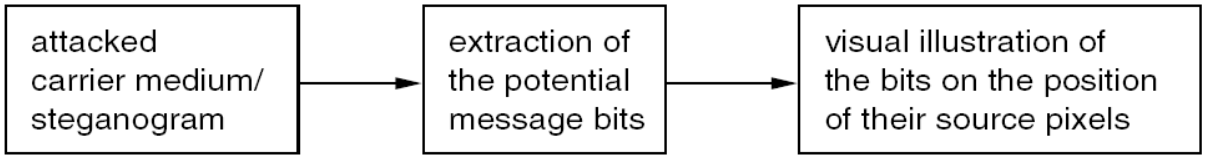


Figura : ***Schema di attacco visuale***

Un esempio concreto di quello che stiamo dicendo può essere mostrare come EzStego possa fallire. Un algoritmo ad attacco visuale può battere la steganografia filtrando i valori dei pixel di un’ immagine attraverso una funzione di estrazione, come suggerito nello schema precedente. EzStego infatti usa i colori dei pixels, definiti nella palette, per determinare l’inserimento dei bit del messaggio. Il filtro che permette di attaccare EzStego sostituisce la palette originale con una bicolore che contiene solo il bianco ed il nero. La figura seguente illustra questa sostituzione di bit.

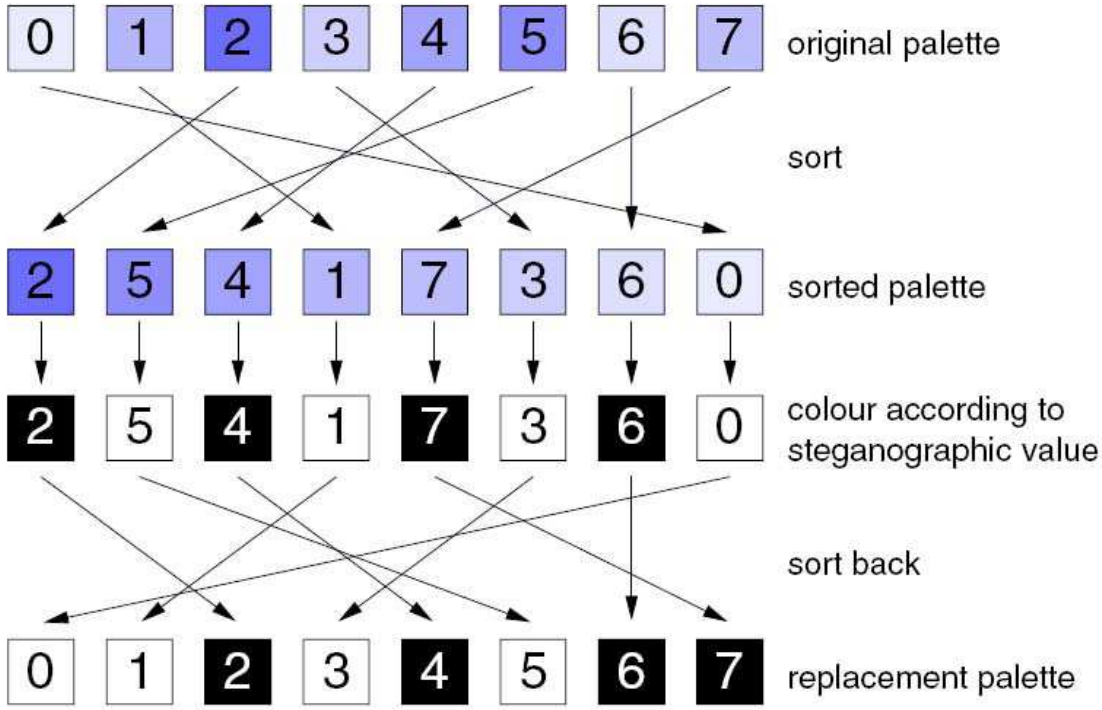
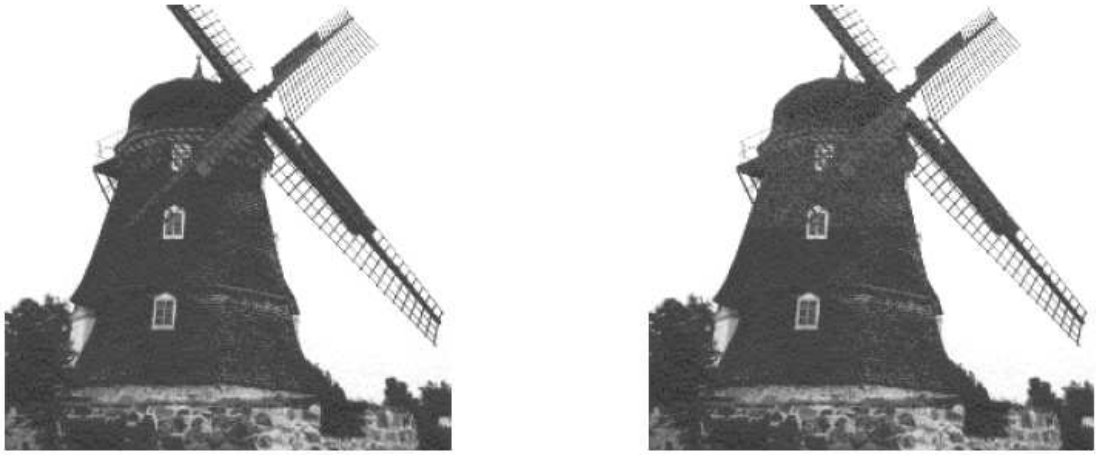


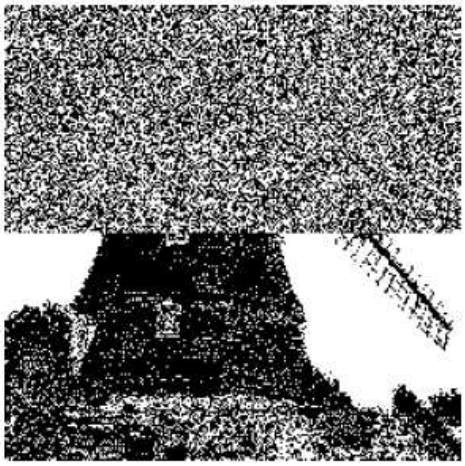
Figura : ***Attacco visuale ad EzStego[19]***

I colori che hanno un indice pari nella palette ordinata diventano neri, il resto diventano bianchi. I messaggi segreti che non utilizzano tutta la capacità possibile, lasciano parte del cover immodificata, poiché EzStego usa un sequential embedding. È facile a quel punto riconoscere questi messaggi a occhio nudo,come si evince dalle immagini.



La prima immagine a sinistra è senza un contenuto steganografico, mentre la seconda a destra ha un messaggio segreto contenuto in essa.

A questo punto osserviamo le due immagini filtrate dall'algoritmo :



La prima a sinistra risulta pulita al controllo, mentre la seconda è facilmente rintracciabile come "strana" ad occhio nudo. Quindi la seconda sarà chiaramente un immagine steganografata che il controllore farà in modo di eliminare. Una volta che abbiamo potuto constatare la validità di tale algoritmo dobbiamo dire che ha anche alcuni svantaggi. Il più grande inconveniente dell'attacco visuale risiede nell'impossibilità di automatizzare il processo di analisi, poiché esso non può essere portato avanti senza l'intervento di un operatore umano che sia in grado di notare anomalie nell'immagine filtrata. Un altro inconveniente consiste nella lentezza del processo di filtraggio dei bit. Per tali motivi, è stato sviluppato l'attacco statistico, che può essere automatizzato senza problemi.

La principale caratteristica di un sistema steganografico, come abbiamo visto, è quella di mantenere segreta la presenza di un messaggio nascosto in un file contenitore, però questi sistemi, a causa della loro natura invasiva, lasciano dietro di loro tracce individuabili nello stesso stego object, come anche nel messaggio segreto. Infatti, modificare il contenitore cambia le sue proprietà statistiche, così che un ascoltatore (avversario) può scoprire le distorsioni presenti nelle proprietà statistiche del risultante stego object. Il processo di ricerca di tali distorsioni prende il nome di steganalisi statistica. L'idea dell'attacco statistico è di confrontare la distribuzione di frequenza dei

colori di un potenziale file stego con la distribuzione di frequenza teoricamente attesa per un file stego.

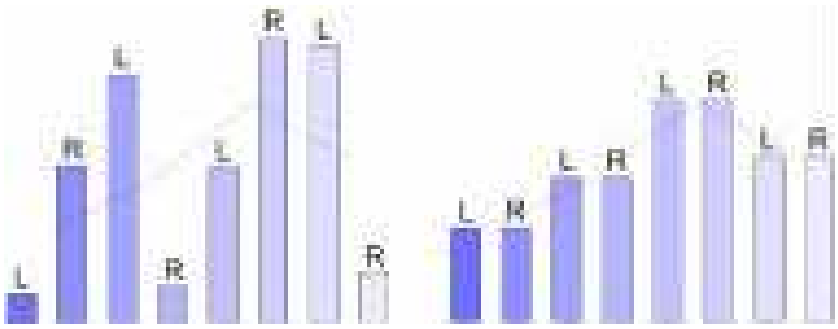
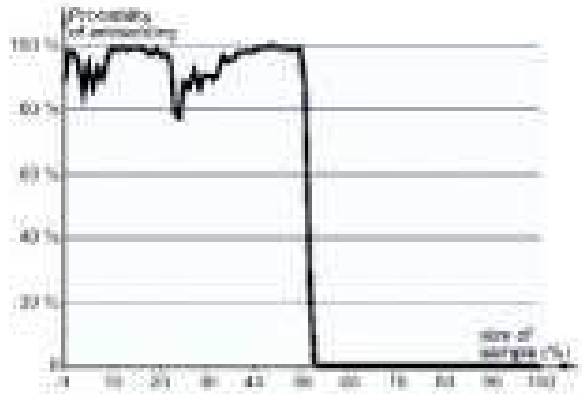


Figura : ***Frequenze immagine steganografata***

Come si può vedere dalla figura , le frequenze di un immagine contenitore dopo l'embendding si eguagliano a due a due e questo è sintomo del fatto che nell'immagine è stato inserito un qualche messaggio segreto.



Da questa figura si può vedere come nella prima parte dell'immagine si nasconde un messaggio segreto, mentre nella seconda parte, dopo la caduta libera, non si nasconde nessun dato.

6- STEGANOGRAFIA NEI FILE AUDIO

Dopo aver parlato in lungo e in largo di tecniche varie di steganografia applicate alle immagini, è ora di concentrasi su un altro tipo di contenitore , ovvero i file audio. Innanzi tutto faremo una panoramica generale di come i file audio vengono rappresentati e memorizzati, poi porteremo alcuni esempi di algoritmi steganografici applicati a tali contenitori e infine come battere questi ultimi.

6.1 – Informazioni sui file audio

Il segnale audio è per sua natura un segnale analogico, ossia un segnale che varia in modo continuo nel tempo e la cui ampiezza assume un'infinità di valori compresi tra un valore massimo ed uno minimo. Perciò, innanzitutto tale segnale deve essere convertito in un segnale digitale, cioè discreto che assume un insieme finito di valori.

La conversione analogico-digitale è ottenuta tramite due sottoprocessi denominati campionamento e quantizzazione. Con il campionamento, ad intervalli regolari (intervallo di campionamento), vengono prelevati dal segnale audio analogico i valori delle ampiezze assunti in un preciso istante. Successivamente, mediante la fase di quantizzazione, i valori delle ampiezze prelevati dal segnale analogico vengono discretizzati, ossia arrotondati all'intero immediatamente più vicino considerato [7][21][22].

Lo standard di codifica maggiormente impiegato nei segnali audio è il PCM, acronimo di Pulse Code Modulation (modulazione a codifica di impulsi), il quale impiega una frequenza di campionamento di 44.1 KHz con una risoluzione di rappresentazione dei livelli a 16 bit. Esistono due formati di file audio molto comuni

che utilizzano la codifica PCM con quantizzazione lineare a 16 bit e sono il WAV (Windows Audio-Visual) e l'AIFF (Audio Interchange File Format).

Le frequenze di campionamento più comuni sono 8 KHz, 9.6 KHz, 10 KHz, 12 KHz, 16 KHz, 22.05 KHz, 44.1 KHz e 48 KHz. Il tasso di campionamento influisce sulle procedure di inserimento di dati segreti nei file audio in due modi: il primo è quello di imporre un limite superiore sulla porzione dello spettro di frequenza utilizzabile, mentre il secondo deriva dal fatto che aumentando la frequenza di campionamento aumenta anche la quantità di dati che è possibile occultare nel file audio.

Il problema legato a un file audio stereo (due canali, destro e sinistro) campionato a 44.1 KHz a 16 bit di risoluzione, come per esempio il WAV, è determinato dalle notevoli dimensioni di spazio in memoria richieste (circa 10MB per ogni minuto di riproduzione). Per questo motivo spesso si ricorre all'uso di formati audio compressi, come l'ormai divulgatissimo MP3.

Il formato MP3 (Moving Picture Expert Group-1/2 Audio Layer 3) è un algoritmo di compressione audio di tipo lossy in grado di ridurre drasticamente la quantità di dati richiesti per memorizzare un suono, rimanendo comunque una riproduzione accettabilmente fedele del file originale non compresso [23]. In sostanza, l'algoritmo di compressione MP3 elimina dal segnale audio le frequenze non importanti per il sistema uditivo umano, ossia quelle molto alte e molto basse, inoltre elimina le ridondanze derivanti dalla trasmissione stereo su due canali. Con il formato MP3 è possibile memorizzare un minuto di audio in poco meno di 1 MB.

6.2 – Steganografia applicata ai file audio

Come descritto già precedentemente, i file audio subiscono una conversione analogico-digitale attraverso i vari algoritmi di compressione e attraverso questi subiscono inevitabilmente delle forme di approssimazione dei dati che agevolano l'applicazione delle tecniche steganografiche. Infatti, dopo le immagini, i file audio sono i contenitori maggiormente impiegati nella steganografia digitale. Inoltre, i file audio contengono delle informazioni ridondanti che possono essere sostituite con dati segreti da queste tecniche , senza provocare significative degradazioni della qualità sonora del file cover audio.Tuttavia, l'occultamento nei file audio risulta essere più impegnativo e problematico di quello nelle immagini, a causa della maggiore sensibilità del sistema uditivo umano rispetto a quello visivo [vedi bib.]. Bisogna perciò stare molto attenti alla quantità di rumore, e alla sua potenza, che eventualmente si inietta nel file audio e alla tipologia ed entità delle modifiche eseguite per fini steganografici.

Ma nonostante la notevole sensibilità del sistema uditivo, anche questo possiede alcune lacune e debolezze di fondo che possono essere sfruttate dalle tecniche di occultamento. Una delle debolezze è conosciuta con il nome di "mascheramento audio", ovvero quando si utilizzano suoni di notevole intensità in sovrapposizione a suoni invece molto deboli, oppure quando si sovrappongono suoni con frequenze simili. Altra debolezza dell'udito è rappresentata dal fatto che esso non è in grado di percepire la fase assoluta di un suono, ma soltanto quella relativa. Infine, esistono svariate tipologie di distorsioni ambientali talmente comuni che, nonostante siano percepite dall'ascoltatore, vengono in molti casi sistematicamente ignorate. Abbiamo fatto un elenco quindi di situazioni in cui le caratteristiche del suono possono essere alterate in modo da risultare impercettibili all'udito. Ed è qui che la steganografia compisce e permette di intervenire per inglobare messaggi segreti.

La scelta della tecnica di occultamento nei file audio è innanzitutto vincolata dalla tipologia del mezzo trasmissivo che verrà impiegato per trasferire il segnale audio

dal mittente al destinatario. Infatti, il trasferimento può avvenire in un ambiente puramente digitale, analogico o via etere. A seconda quindi dell'ambiente trasmissivo utilizzato varia la quantità di dati che si può celare nel file audio. Un valore tipico della capacità di occultamento nei file sonori risulta essere quello di 16 bps (bit per secondo), ma esso può variare da 2 bps a 128 bps.

Detto questo, adesso esaminiamo quelle che sono le tecniche di occultamento nei file audio maggiormente impiegate :

- **LSB (Least Significant Bit) insertion**
- **Spread Spectrum**
- **Phase Coding**
- **Echo Data Hiding**

6.3 – LSB (Least Significant bit) insertion sui file audio

La tecnica LSB, già affrontata per le immagini, prevede la sostituzione dei bit meno significativi dei campioni dell'onda sonora, ottenuti tramite la conversione analogico-digitale, con i bit che costituiscono l'informazione segreta [7]. Se tale tecnica viene praticata nel modo giusto, le alterazioni del suono che ne derivano sono molto piccole e assai poco probabili da rilevare. E' fondamentale a questo scopo, per i file audio, effettuare una selezione pseudocasuale dei campioni da modificare, poiché la semplice sostituzione dei bit di campioni presi in sequenza causa la nascita di distorsioni udibili dall'ascoltatore. La procedura LSB rappresenta il metodo più semplice per occultare dati segreti nei cover audio e consente di celare un'elevata quantità di dati.

I punti deboli di questa tecnica sono rappresentati dal fatto di introdurre solitamente un fastidioso rumore di fondo, avvertibile dall'orecchio umano, e dal

possedere una scarsa resistenza alle manipolazioni, quali la compressione lossy e l'introduzione di segnali di disturbo nel canale di trasmissione.

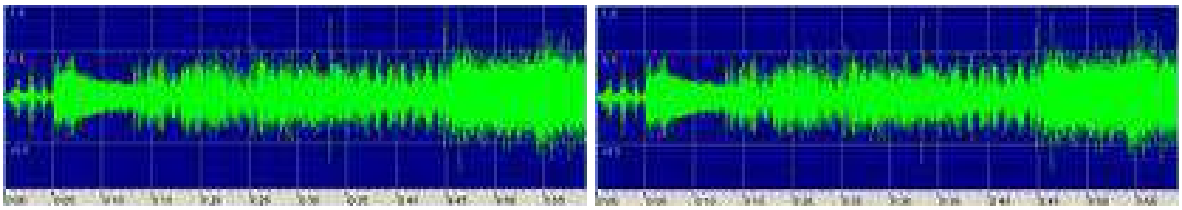


Figura : ***Esempio di utilizzo di tecnica LSB su file WAV. A sinistra file originale, a destra quello steganografato con un file di testo.***

6.4 – Spread Spectrum

La tecnica dello spread spectrum consiste nella diffusione del segnale relativo ai dati segreti lungo quello del segnale cover audio; si ottiene così l'effetto di propagare l'energia spettrale del segnale segreto, facendola diminuire in densità e mantenendola solitamente al di sotto del livello del rumore, con il risultato di renderne assai difficile l'identificazione [7]. Al fine di veicolare l'inserimento dei dati segreti nel cover audio, il segnale del messaggio segreto viene dapprima modulato con una sequenza di rumore pseudocasuale, ottenuta tramite un generatore di rumore pseudocasuale pilotato da una chiave segreta condivisa da mittente e destinatario, e successivamente inserito nel file audio. Come per la tecnica LSB, anche la procedura di Spread Spectrum introduce nel file sonoro un rumore additivo di fondo che può essere percepito dall'udito umano. Affinché tale rumore risulti impercettibile vengono impiegati dei particolari filtri che ne attenuano la potenza. Il punto di forza di questa tecnica steganografica è rappresentato dall'elevata sicurezza, mentre la capacità di occultamento solitamente si attesta sui 4 bps.

6.5 – Phase Coding

La tecnica di Phase coding consiste nel dividere il segnale del contenitore audio in tanti blocchi e tramite fourier (DFT) vengono calcolate le differenze di fase tra i blocchi consecutivi. Al primo blocco viene assegnata una fase iniziale e quella dei successivi blocchi viene impostata in modo da non superare una certa soglia. Tutto poi viene concatenato per formare il segnale audio finale. È uno dei metodi più interessanti perché permette di codificare molte informazioni lasciando quasi inalterato il rapporto segnale rumore. L'orecchio umano non distingue variazioni di fase assolute (quella del primo blocco si può variare a piacere …) ma solo relative. Solo se le differenze sono eccessive può essere percepito qualcosa di strano da chiunque. La lunghezza dei blocchi e gli intervalli in cui sono inseriti i dati segreti devono essere noti al ricevente per poter estrapolare il messaggio segreto.

6.6 – Echo Data Hiding

Come la dicitura inglese suggerisce, la tecnica Echo Data Hiding consiste nell'occultare dati segreti in un segnale audio ospite introducendo in esso degli echi [7]. Si basa sul fatto che l'orecchio umano non riesce a percepire due frequenze uguali, ma con ampiezza diversa, divisi da un piccolo intervallo di tempo. Rimanendo entro questo intervallo, si può produrre un echo di un segnale a 2 distanze predefinite che rappresentano l'informazione nascosta 0 oppure 1. Diminuendo il ritardo (offset) tra il suono originale e l'eco, i due segnali tendono a mescolarsi e a confondersi. Ad un certo punto, nell'ordine dei millisecondi, il ritardo è tale che il sistema uditivo umano non è più in grado di distinguere separatamente i due suoni. L'eco viene percepito come una risonanza aggiunta, un arricchimento del suono, e non come una distorsione.

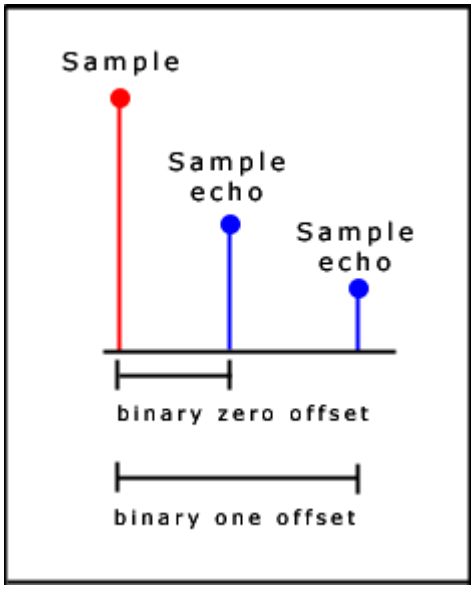


Figura : ***Esempio di contenitore audio con la presenza di echi.***

Il contenitore viene diviso in blocchi e ad ognuno di questi si applica il processo di codifica appena descritto. Poi i blocchi vengono concatenati nuovamente per ricreare un file che all'ascolto si presenta 'come' quello di partenza. Per la decodifica si deve dividere in blocchi allo stesso modo come per la codifica, poi tramite la trasformata di fourier si riesce a quantificare l'echo (la distanza dal segnale più forte) e stabilire se decodificare 0 oppure 1. Con il metodo Echo Data Hiding è possibile codificare e decodificare informazione segreta, nella forma binaria digitale, in un file audio provocando soltanto una minima degradazione del suono originale. Questo significa che le modifiche apportate al segnale audio originale non vengono percepite dal sistema uditivo umano, oppure le risonanze aggiunte vengono percepite come un arricchimento del suono stesso, ma mai come una distorsione.

L'Echo Data Hiding evidenzia una buona robustezza, e una discreta capacità di occultamento (tipicamente sui 16 bps), soprattutto se utilizzata con file audio ad alta qualità. Invece, l'algoritmo lavora meno bene con file audio in cui vi siano già delle degradazioni (rumore), oppure dove vi siano degli intervalli di silenzio, i quali rendono assai arduo e pericoloso l'inserimento degli echi al loro interno.

7- PRESENTE E PASSATO DELLA STEGANOGRAFIA

7.1 – La Steganografia oggi e sviluppi futuri

Gli utilizzi al giorno d'oggi della Steganografia spaziano in molti ambiti del sapere umano. In tal senso, molte organizzazioni militari e di intelligence mirano all'utilizzo di queste tecniche per conservare al meglio le loro comunicazioni segrete a discapito di tecniche crittografiche che non potrebbero essere usate per gli stessi scopi.

Infatti, come abbiamo dimostrato nei capitoli precedenti, è molto facile per una qualsiasi entità di controllo sulle comunicazioni (come ad esempio regimi dittatoriali o repressivi) intercettare materiale cifrato e bloccarlo. Nel campo del commercio elettronico invece la steganografia potrebbe essere utilizzata nell'implementazioni di sistemi sicuri di pagamento on-line, in aggiunta ai normali protocolli di protezione crittografici esistenti quali SSL (Secure Socket Layer) e SET (Secure Electornic Transactions).

Un altro uso interessante potrebbe essere quello di implementare un sistema di codifica e decodifica per le multinazionali televisive ad oggi esistenti sul satellite o digitale terrestre.

Entrando invece nel campo della sicurezza nei trasporti, un utilizzo della steganografia potrebbe essere quello di nascondere nelle foto dei passaporti elettronici delle informazioni segrete che le forze dell'ordine possono sfruttare. Ad esempio possono garantire maggiore sicurezza, avere informazioni dettagliate sulle persone esaminate e segnalare queste eventuali persone come criminali.

Altro campo interessante è quello della medicina dove si potrebbero sfruttare le immagini radiografiche per celare dati sensibili del paziente , risolvendo cosi in modo elegante problemi legati alla privacy.

Un altro esempio interessante è inglobare in qualsiasi client di posta e anche browser web proprio un sistema di steganografia.

Oltre a questi sviluppi più o meno già portati a termine, ci sono anche delle sfaccettature più pericolose nell'utilizzo di tali tecniche. Un esempio sono gli utilizzi che gruppi terroristici fanno di talune tecniche per poter comunicare ordini, informazioni e notizie ai loro subalterni o associati. I primi allarmi pubblici che riguardano questa situazione sono stati dati da alcuni giornali americani nel 2001.[16][17] Questi articoli riportavano come i terroristi stessero usando la steganografia per sottrarre le loro comunicazioni ai nuovi, e più stringenti, controlli previsti dalle leggi USA. Le pratiche steganografiche, sempre secondo questi articoli, sarebbero consistite nell'occultamento di messaggi all'interno di immagini pubblicate in Internet durante aste on-line (il sito prediletto sarebbe stato quello di eBay). Sebbene non venisse riportata alcuna informazione tecnica che potesse supportare questi allarmi, i summenzionati articoli sono stati oggetto di grande attenzione ed hanno prodotto un'eco notevole. Esistono, in ogni modo, preoccupanti indizi a supporto dell'uso terroristico della steganografia. "C'è la pista di Jamal Beghal, il leader del commando che stava preparando un attentato all'ambasciata americana di Parigi: l'uomo, addestrato in Afghanistan dove aveva incontrato un luogotenente di Bin Laden, aveva istruito il suo gruppo affinché tutte le comunicazioni interne fossero fatte attraverso immagini "ritoccate" pubblicate sulla Rete"

È chiaro come delle situazioni del genere siano portate da un utilizzo sbagliato e deprecabile di tali tecniche, ma che purtroppo appartengono a determinate frange di estremismi a livello mondiale. Così come delle forbici possono tagliare un pezzo di carta e possono anche ferire, cosi la steganografia può essere sia usata per scopi apprezzabili (come la tutela della privacy oppure la difesa di materiale sottoposto a copyright) sia per fini deprecabili (organizzazione di piani criminosi, invio di immagini pedo-pornografiche contenute in immagini dall'aspetto innocuo).

8- DEFINIZIONI UTILI

Cover Object : Si definisce cover object (o cover medium) l'oggetto che sarà usato come contenitore per l'inserimento del messaggio. Diversi oggetti sono stati utilizzati in progetti reali come contenitori per messaggi steganografici: immagini, files audio, video, file systems e pagine html solo per citarne alcuni. Il cover object designa il contenitore originale prima che sia effettuata qualsiasi operazione su di esso.

Stego Object : Si definisce stego object (o stego medium) l'oggetto, risultato dell'algoritmo di steganografia (quindi dopo che sono state effettuate operazioni su di esso), che contiene (trasporta) al suo interno il messaggio.

PGP : Pretty Good Privacy (PGP) è un programma che permette di usare autenticazione e privacy crittografica. Nelle sue varie versioni è probabilmente il crittosistema più usato al mondo.

Funzione Hash : una funzione hash è una qualsiasi funzione definita in modo da dare risultati ben distribuiti nell'insieme dei valori possibili; tipicamente questo si ottiene decomponendo e mescolando in qualche modo le componenti dell'argomento

Plaintext e Cyphertext : il primo è l'informazione o il messaggio da cifrare, il secondo è il testo dopo la cifratura

Waterwarking : si riferisce all'inclusione di informazioni all'interno di un file multimediale o di altro genere, che può essere successivamente rilevato o estratto per trarre informazioni sulla sua origine e provenienza.

Patchwork : è un metodo che marca delle aree dell' immagine, dette appunto patches.

Cropping : è un metodo che serve per tagliare e ridurre la grandezza delle immagini

9- RIFERIMENTI

[1] **http://it.wikipedia.org/wiki/Steganografia**

[2] **http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0001/Steganografia/introduzione.html**

[3] **Penn Leary**: "The Second Cryptographic Shakespeare" Westchester House; Revised edition (July 1990). Riferimenti al libro al **http://home.att.net/~mleary/**

[4] **USA Today**: "Hatred, hidden messages", Reperibile al **http://www.usatoday.com/news/world/2002/07/10/web-terror-cover.htm** (link locale)

[5] **http://www.arruzzoli.it/article/Steganografia.pdf**

[6] **Paolo Cavallo**, Tecniche di steganografia per immagini codificate tramite la dwt ed analisi comparativa con steganografia per immagini jpeg, tesi, 2001.

[7] **Serena Longhini**, Procedure steganografiche e protocolli speciali per l'occultamento e la trasmissione dei dati, tesi, 2003.

[8**] http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0203/steganografia.pdf**

[9] **Niels Provos and Peter Honeyman,** Hide and Seek: An Introduction to Steganography, IEEE SECURITY & PRIVACY, 2003.

[10] **SNOW**, Steganographic nature of whitespace, **http://www.darkside.com.au/snow**

[11] **Niels Provos and Peter Honeyman,** Hide and Seek: An Introduction to Steganography, IEEE SECURITY & PRIVACY, 2003.

[12] **Claudio Agosti**, Steganografia, l'arte della scrittura nascosta, slides, 2003.

[13**] J. Fridrich, M. Goljan, R. Du**, Steganalysis Based on JPEG Compatibility.

[14**] N. F. Johnson, Z. Duric and S. Jajodia,** Information Hiding: Steganography and Watermarking – Attacks and Countermeasures, Kluwer Academic Publishers, Boston Dodrecht London (2000).

[15] **N. Provos, P. Honeyman**, Detecting Steganographic Content on the Internet, CITI Technical Report 01-11, 2001.

[16] **Jack Kelley**, Terror groups hide behind Web encryption, USA Today, febbraio 2001.

[17] **Declan McCullagh,** Secret Messages Come in .Wavs, Wired News, febbraio

2001.

[18] **http://it.wikipedia.org/wiki/Compressione_dei_dati**

[19] **A.Westfeld and A. Pfitzmann**, Attacks on Steganographic Systems, Proc. Information Hiding 3rd Int'l Workshop, Springer Verlag, 1999, pp. 61-76.

[20] **http://tanadelratto.blogspot.com/2010/07/editoria-digitale-che-punto-siamo.html**

[21] **W. Bender, D. Gruhl, N. Morimoto, A. Lu**, "Techniques for data hiding", IBM Systems Journal, vol. 35, 1996. **http://www.almaden.ibm.com/cs/people/dgruhl/313.pdf**.

[22] **Gary C. Kessler**, "An Overview of Steganography for the Computer Forensics Examiner", Computer & Digital Forensics Program, Champlain College, Burlington. http://www.wetstonetech.com/f/stego-kessler.pdf.

[23] **http://it.wikipedia.org/wiki/MP3**

[24**] http://www.snotmonkey.com/work/school/405/methods.html**