UNIVERSITÀ DEGLI STUDI DI SALERNO

FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E NATURALI

DOTTORATO DI RICERCA IN

SCIENZE MATEMATICHE, FISICHE E INFORMATICHE

CICLO XI (NUOVA SERIE)

TESI DI DOTTORATO

METODI INNOVATIVI DI INFORMATION FUSION PER LA CRITTOGRAFIA

RELATORE
Prof. Michele Nappi

CANDIDATO
Dott. Elmo Benedetto

CORRELATORE
Prof. Gerardo Iovane

COORDINATRICE
Prof.ssa Patrizia Longobardi

ANNO ACCADEMICO 2012-2013

Indice

INTRODUZIONE

La sicurezza e l'autenticità delle informazioni e della persona sono tra i temi maggiormente trattati negli ultimi anni in ambito informatico. Inoltre lo sviluppo di nuove tecnologie di comunicazione e la rapida diffusione dell'informatica e dei personal computer, rende necessario avere la certezza che i propri dati risultino essere al sicuro da eventuali soggetti, i quali tentano, senza autorizzazione alcuna, di appropriarsene. Il punto di partenza che ha portato allo sviluppo di questa tesi è appunto la realizzazione di idee utilizzabili nell'ambito della sicurezza informatica, che permettano non solo di garantire un alto livello di segretezza, ma che forniscano, con un elevato grado di certezza, l'identità della persona. In quest'ottica è stata sviluppata una piattaforma per la generazione di chiavi d'accesso o crittografiche mettendo insieme la biometria, la crittografia a chiave pubblica e i codici numerici random. Per fare ciò, data la natura eterogenea dei campi proposti, si sono usate tecniche innovative per la fusione dei dati e l'idea di base è quella di fondere le peculiarità di tali tecniche per creare un'infrastruttura capace di rendere sicuro l'accesso e creare codici crittografici per cifrare documentazioni riservate. L'infrastruttura analizzata è chiamata Secure Infrastructure Information Fusion for Authentication and Encryption (SIIFAE). La biometria fornisce all'infrastruttura la capacità di autenticazione dell'individuo, la crittografia a chiave pubblica provvede a rendere sicura una specifica chiave di accesso e permette di criptare delle documentazioni classificate, i codici pseudo casuali forniscono un ulteriore mezzo per garantire sicurezza informatica e casualità, fondamento principale nello sviluppo di applicazioni dedicate alla InfoSecurity. L'infrastruttura permette di creare principalmente un codice di autenticazione sicuro utilizzando la componente biometrica e la componente crittografica derivata dal RSA. Queste componenti vengono sottoposte ad una procedura di fusione dei dati innovativa, detta Information Fusion Ibrida. L'aggettivo ibrido sta ad indicare il fatto che la fusione avviene tra dati eterogenei e soprattutto vengono utilizzate due tecniche fino ad ora mai combinate. In secondo luogo, l'infrastruttura genera un codice crittografico sicuro combinando la componente del RSA e un codice pseudo casuale ideato appositamente per la generazione di numeri basato sulla formulazione dei frattali. Tale codice può essere utilizzato per applicazioni di algoritmi simmetrici One-Time-Pad grazie alla possibilità di generare codici di lunghezza arbitraria, quindi lunghi quanto il testo da cifrare. Vedremo che l'algoritmo di fusione ideato attraverso passaggi opportuni riesce a costruire tali codici di estrema sicurezza. L'algoritmo che genera i due codici rimane invariato a prescindere dal suo utilizzo in modalità Authentication o Encription. Entrambi i codici generati sono stati sottoposti a svariati test che dimostrino l'effettiva proprietà di sicurezza necessaria all'utilizzo nella sicurezza informatica. Tali test sono stati effettuati utilizzando le batterie statistiche del NIST. I test servono a garantire che i codici casuali generati siano adeguati ai sistemi crittografici, chiamati (CSPRNG, Cryptographically Secure Pseudo-Random Numbers Generator). Successivamente viene

mostrato un sistema per realizzare un codice crittografico, che sia utilizzabile per garantire un alto livello di segretezza. La tecnica di fusione modificata è denominato F&NIF (Fractal & Numerical Information Fusion). E' ben noto che nelle infrastrutture della Difesa o in archivi per documentazioni classificate e confidenziali di Enti Pubblici o Privati è frequentemente necessario proteggere o criptare documenti di varia entità al fine di garantire la sicurezza e l'offuscamento delle informazioni tramite chiavi che si basino sull'utilizzo di codici fortemente casuali. Nessuno deve essere in grado di comprendere e decodificare una informazioni segreta! In quest'ottica si è sviluppato una nuova chiave che utilizza la sicurezza e la robustezza dell' algoritmo RSA e la casualità intrinseca di valori generati utilizzando i frattali. L' RSA rappresenta la migliore tecnica crittografica ed assicura il più alto grado di robustezza agli attacchi crittografici. Solo recentemente, marzo 2010, tre studiosi dell'Università del Michigan sono riusciti a decifrare una chiave privata del RSA a 1024 bit, effettuando un attacco di tipo Fault-Based, in circa 100 ore. Questo tipo di attacco, però, parte dal presupposto di essere nelle vicinanze dei componenti interni del sistema vittima ed, utilizzando una falla del protocollo OpenSSL (Secure Sockets Layers), ottenere le informazioni segrete sul software. Per tale motivo le chiavi RSA che utilizzeremo in questo lavoro saranno di 2048 bit. Il frattale rappresenta un modo innovativo per applicazioni di sistemi dinamici caotici. Gli oggetti della natura (alberi, montagne, nuvole, foglie, felci etc.) sono tutti caratterizzati da un aspetto irregolare e non possono essere studiati usando le proprietà della geometria euclidea (rette, poligoni, cerchi). Tutto ciò che si incontra in natura è molto più complesso, frammentato, frastagliato, caotico. Questo ha giustificato l'introduzione di un nuovo tipo di geometria da parte del matematico Benoit B. Mandelbrot nel 1982: la geometria frattale. Un frattale è un insieme F che gode delle seguenti proprietà:

1) Autosimilarità: F è unione di un numero di parti che, ingrandite di un certo fattore, riproducono tutto F; in altri termini F è unione di copie di se stesso a scale differenti.

2) Struttura Fine: F rivela dettagli ad ogni ingrandimento.

3) Irregolarità: F non si può descrivere come luogo di punti che soddisfano semplici condizioni geometriche o analitiche.

4) Dimensione: si definisce un nuovo concetto di dimensione differente dalla concezione classica che sia maggiore della dimensione topologica.

Proprio per queste caratteristiche caotiche dei frattali, in letteratura, si hanno delle applicazioni recenti nel campo della sicurezza e della crittografia. Nel 2004 viene presentato un brevetto per una applicazione che cripta e decripta dati in due dimensioni usando una matrice frattale quadrata derivata da uno specifico valore iniziale che viene inviato tramite un canale sicuro (crittografia simmetrica). Nel 2007, Alia ha introdotto un algoritmo di crittografia asimmetrico in cui la chiave pubblica veniva generata attraverso la formula di Mandelbrot mentre la chiave privata era creata utilizzando la formula di Julia. Nel 2008, Lian et al. hanno proposto un metodo per codificare delle immagini in maniera sicura utilizzando i parametri frattali. Questa codifica permette all'immagine di non cambiare il formato originale del file. Rozouvan, invece, utilizzava l'immagine di un frattale, quindi una matrice, come chiave per cifrare un'altra immagine. Questo tipo di cifratura è utilizzabile nei video real-time, la sua forza è la grande variabilità della chiave a partire dal parametro iniziale. Nel 2009, però, Yoon et al. dimostrarono che il metodo di Rozouvan non era

così sicuro, in quanto, sottoponendo l'algoritmo a tre differenti tipi di attacchi la matrice segreta poteva essere rivelata. Anand et al., nel 2009, propongono un metodo crittografico simmetrico real-time che utilizza il set di quaternioni del frattale di Julia. Non è presente alcuno scambio di chiavi. Ogni host calcola tramite l'uso dei quaternioni real-time la chiave. Stabilendo una connessione SSL, il mittente crea un nuovo timestamp sommando il timestamp del mittente con il timestamp corrente. Il ricevente calcola il timestamp del mittente tramite il nuovo timestamp. La condizione di real-time permette che il timestamp corrente sia uguale per entrambi ed autentica il mittente. Il nuovo timestamp viene, poi, utilizzato per inizializzare la creazione della chiave tramite i quarteroni del frattale di Julia. Nel 2009, viene applicato un nuovo approccio crittografico usando i frattali. Prima viene utilizzato un alfabeto cifrato e tramite la costruzione di un polinomio viene calcolata la trasformazione affine IFS (Iperbolic IFS) viene, quindi, generato l'attrattore frattale. Questo metodo trova applicazioni nell'ambito della steganografia. Nel 2010 viene presentato un brevetto di un metodo per la protezione dei dati attraverso l'utilizzo di sistemi che usano la geometria frattale (Mandelbrot). Risulta un sistema di autenticazione a tempo; in ogni istante si calcola un valore numerico ricavato dal precedente. In questo modo è impossibile ricavare il valore corrente. Come si nota dalla letteratura, vengono spesso utilizzate le proprietà caotiche dei frattali nell'ambito della crittografia. In questo lavoro noi proponiamo un approccio combinato utilizzando RSA e i frattali, questo si basa sull'utilizzo di tecniche di Information Fusion, tramite le quali ottenere una chiave ad alta sicurezza. L'Information Fusion (IF) è un campo nuovo: prevede che i dati provenienti da differenti sorgenti vengano fusi insieme al fine di ottenere una super-informazione. Il campo dell'Information Fusion è comunemente visto come un'area di ricerca multidisciplinare che coinvolge un insieme di aree di ricerca caratterizzate a loro volta da una multidisciplinarietà e da specifiche comunità di ricerca. Il lavoro proposto sfrutta la tecnica di Information Fusion presentata in un precedente lavoro. Questa tecnica, però, è stata sottoposta ad opportune modifiche. In quanto, la tecnica proposta precedentemente generava dei codici identificativi ad alta sicurezza, atti all'autenticazione, che fondevano un codice biometrico (Finger Code) ed un codice numerico basato sulla primalità (Modulo RSA). Nel lavoro proposto, invece, si vuole creare una chiave crittografia fondendo una componente numerica basata sulla primalità (Modulo RSA) ed una componente numerica random generata tramite formule sui frattali. Il metodo di costruzione del nuovo codice (chiave) generato dall'algoritmo di fusione, sarà dipendente dalla chiave privata dell'algoritmo RSA. Nel presente lavoro si andranno ad analizzare tutte le tematiche che entrano in gioco in questa infrastruttura. Questa analisi si articola nei seguenti capitoli:

1. Biometria
2. Crittografia
3. Frattali
4. Information Fusion
5. Secure Infrastructure Information Fusion for Authentication and Encryption (SIIFAE)
6. Fractal and Numerical IF

CAPITOLO 1

LA BIOMETRIA

1.1 – Introduzione

Il termine biometria – letteralmente *misura della vita* – sta ad indicare qualunque tecnica permetta la misurazione delle caratteristiche vitali o fisiche un di essere vivente. Nello specifico si occupa di capire come alcune di queste caratteristiche del corpo umano, uniche per ciascun individuo, possano essere utilizzate come strumento di riconoscimento individuale. L'applicazione della biometria nello studio dei caratteri fisici di un numero imprecisato di individui permette di individuare così una serie di elementi che, per la loro originalità, possono consentire l'identificazione di un soggetto, fornendo una risposta in termini probabilistici. Come si intuisce, una delle principali applicazioni della biometria in ambito informatico, o più comunemente in ambito sicurezza, vede la sostituzione delle comuni *password alfanumeriche* o *chiavi di crittografia*, utilizzate per proteggere documenti o aree ad accesso limitato, con una delle suddette *caratteristiche biometriche.* Difatti si prevengono i casi di sostituzione di persona o di smarrimento/furto della password. In altre parole si viene a creare una nuova categoria che può essere definita come richiesta di *"ciò che l'utente è".*

Possiamo quindi definire un *Sistema Biometrico* come

"un sistema automatico per la verifica e/o il riconoscimento

dell'identità di un individuo, in base ad alcune caratteristiche

fisiologiche e/o comportamentali."

Potendo trovare varie caratteristiche in ogni individuo, è possibile realizzare vari sistemi biometrici, in base appunto al tratto biometrico che viene esaminato per identificarlo.

In particolare, i sistemi biometrici si dividono in due principali categorie rappresentati in Figura 1.1:

- sistemi basati sulle caratteristiche *fisiologiche* dell'individuo
- sistemi basati sulle caratteristiche *comportamentali* dell'individuo.

Alla prima categoria appartengono tratti biometrici quali impronta digitale, biometria del volto, dell'iride, della forma dell'orecchio, la vascolarizzazione della retina, la geometria delle dita e della mano, l'impronta palmare e, non ultimo, il DNA.

Alla seconda categoria appartengono invece tutte le caratteristiche comportamentali, quali la dinamica della digitazione sulla tastiera, il rilevamento dinamico della firma, lo spettro vocale.

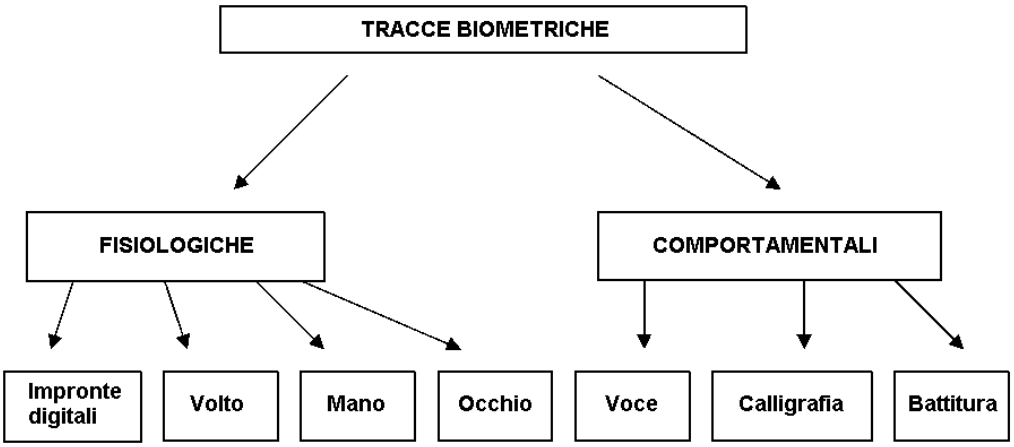


Figura 1.1 - Classificazione delle caratteristiche biometriche

Si possono inoltre distinguere due ambiti applicativi dei sistemi biometrici: *validazione* e *identificazione*

- Nel primo, cioè in caso di *validazione**,*** un qualsiasi individuo afferma una identità ed è quindi chiamato a verificarla. In questo caso ciò che viene fatto è un confronto uno ad uno; l'utente deve fornire quindi, oltre alla propria traccia biometrica, anche un identificativo, ad esempio il Nome/Cognome o un numero di matricola.
- Il secondo ambito invece riguarda l'*identificazione* dell'individuo. In questo caso si ha soltanto una traccia biometrica, ma è ignota l'identità della persona. Si effettua quindi un confronto uno a molti, utilizzando dei *template* memorizzati all'interno un database di tracce biometriche simili, alla ricerca di un possibile riscontro.

In Figura 1.2 sono schematizzate le differenze tra questi due ambiti applicativi.

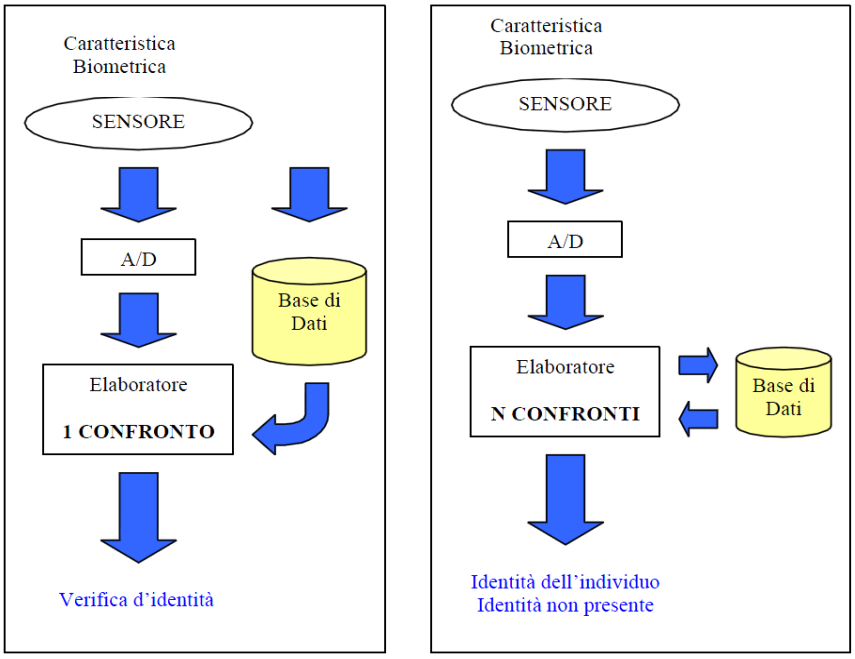


Figura 1.2 – Schema di *validazione* e *identificazione*

Una caratteristica biometrica, per essere utilizzata in un sistema automatizzato per la verifica/identificazione di un individuo, deve possedere le seguenti proprietà:

- *Invariabilità* : devono essere costanti per un lungo periodo di tempo;
- *Misurabilità*: le proprietà devono essere tali da poter essere rilevate facilmente;
- *Singolarità*: le caratteristiche devono avere proprietà sufficientemente uniche da permettere di distinguere una persona da qualsiasi altra;
- *Accettabilità*: l'acquisizione di tali caratteristiche deve essere possibile su un'ampia percentuale della popolazione;
- *Riducibilità*: i dati acquisiti devono poter essere ridotti ad un pattern di facile gestione
- *Affidabilità*: la procedura deve garantire un grado elevato di affidabilità e di riproducibilità;
- *Privacy*: la procedura informatica non deve violare la privacy della persona.

Nella Tabella 1.1 seguente possiamo classificare le principali caratteristiche biometriche in base alla tecnologia di acquisizione e alle loro principali proprietà:

Caratteristiche biometriche	**Tecnologia di acquisizione**	**Invariabilità**	**Singolarità**	**Accettabilità**
Geometria della mano	Ottica	Buona	1:1.000	Molto buona
Geometria delle 2 dita	Ottica	Buona	1:1.000	Molto buona
Retina	Ottica	Molto buona	1:10 milioni	Non buona (invasiva)
Iride	Ottica	Molto buona	1:6 milioni	Buona
Vene superficiali della mano	Ottica	Buona	Non nota	Molto buona
Firma	Dinamica (pressione)	Non Buona	1:10.000	Molto buona
Voce	Elettroacustica	Non Buona	1:10.000	Buona
Volto	Ottica/IR	Buona	Dipendente dal sistema	Buona
Impronte Digitali	Ottica,Capacitiva...	Molto Buona	1:10 milioni	Buona

Tabella 1.1 – Classificazione delle principali caratteristiche biometriche

Come si può intuire dalla tabella, alcuni tratti biometrici sono migliori rispetto ad altri, questo è evidente se si esamina l'attributo *singolarità*. Difatti notiamo come alcune caratteristiche ci permettano di individuare una persona tra un campione di 1000, altre invece arrivano ad un rapporto di 1:10 milioni, indice della bontà del tratto biometrico (vedi impronte digitali e retina). Comunque, la scelta della caratteristica biometrica dipende anche dall'ambito di utilizzo della stessa. La *bontà* di un sistema biometrico viene valutato, oltre che in base alla tecnica di acquisizione della caratteristica, anche in base al comportamento dello stesso in caso di errore. Difatti esistono per ogni sistema biometrico due principali tipi di errore: il *FAR* (*False Acceptance Rate*) e il *FRR* (*False Reject Rate*). Il primo, il *FAR*, è la percentuale di persone riconosciute anche se queste non risultano essere presenti nel database di supporto al sistema. In questo caso abbiamo quindi un caso di *falso positivo*.

Il secondo, il *FRR*, indica la percentuale di persone non riconosciute, quindi non accettate, ma che sono invece presenti nel database. In questo caso abbiamo invece un *falso negativo*. Questi due parametri, se il sistema si trovasse a lavorare in un ambiente ideale, dovrebbero essere nulli, in quanto in queste condizioni il sistema non dovrebbe commettere errori. In realtà questi due errori non hanno mai valore nullo, in particolare sono inversamente proporzionali, ed hanno un andamento monotono decrescente il primo, monotono crescente il secondo. Tramite queste due grandezze, in particolare giocando sul loro rapporto, è possibile regolare il grado di tolleranza del sistema. In Figura 1.3 possiamo vederne l'andamento. Se si considerano i grafici dei valori di *FAR* e *FRR* di un sistema biometrico, troviamo che vi è un punto di incontro tra le due curve in cui i due valori coincidono, per un certo grado di tolleranza. Questo valore è detto *EER* (*Equal Error Rate*), e corrisponde all'errore intrinseco del sistema per quel dato valore di tolleranza.

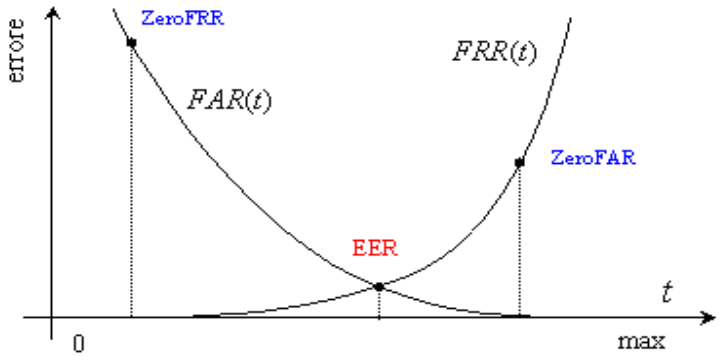


Figura 1.3 – Andamento grafico di *FAR* e *FRR*

Nelle applicazioni reali si predilige avere un maggior numero di falsi rifiuti piuttosto che false accettazioni, al fine di preservare la risorsa che si protegge con il sistema biometrico. Per quanto riguarda gli ambiti applicativi della biometria, come detto in precedenza, l'obiettivo è quello di poter identificare senza possibilità di dubbio l'identità di una persona. Inizialmente i primi ad utilizzare la biometria per l'identificazione sono stati gli organismi militari e quelle organizzazioni in cui era necessario un elevato livello di sicurezza. Con il passare del tempo e con lo sviluppo di hardware/software dedicato, l'utilizzo della biometria si è diffuso rapidamente, tanto da essere diventata una forma di validazione delle informazioni di uso comune.

Tipici esempi di uso della biometria sono:

- Accesso ad aree controllate;
- Sistemi di rilevazione di presenza sul posto di lavoro;
- Controlli alle frontiere e in aeroporti tramite e-card;
- Frode fiscale;
- Pagamenti online;
- Password per il controllo dell'accesso a personal computer, server, documenti;

- Firma elettronica in ambito finanziario (transazioni online);
- Investigazione;
- Schedatura dei criminali.

1.2 – Le impronte Digitali

Come detto, vi sono molte tracce biometriche utilizzabili per lo sviluppo di applicazioni di identificazione/validazione di un individuo, ma per gli obbiettivi di questa tesi si è scelto di effettuare lo studio delle impronte digitali. Più avanti nel paragrafo verranno esposti dei cenni storici riguardanti le impronte stesse, si farà una descrizione dettagliata della conformazione dell'impronta e verranno descritte le tecniche di acquisizione delle informazioni necessarie per poterla utilizzare in ambito informatico.

1.2.1 – Storia

L'impronta digitale è una delle tecniche più antiche utilizzate come strumento di autenticazione e/o firma. Lo studio di tale caratteristica risale al 1686, da parte di un italiano, l'istologo Marcello Malpighi, che per primo descrisse le diverse evoluzioni dei disegni papillari. Il metodo di identificazione mediante impronte digitali era stato già introdotto da William James Herschel (il primo europeo ad essersi reso conto del valore delle impronte digitali a fini dell'identificazione e ad utilizzarle come strumento di firma dei contratti nel suo operato come funzionario della Compagnia delle Indie Orientali) negli anni 1860, ed il suo uso in ambito criminale e giudiziario già proposto da Henry Faulds nel 1880. Con il passare degli anni molti altri si interessarono allo studio delle impronte tra questi Francis Galton, il quale dedicò svariati libri ed articoli all'esposizione dei suoi studi sulle impronte digitali. Misurò la probabilità che due individui diversi possiedano le medesime impronte, ne indagò l'ereditarietà e le caratteristiche in diverse gruppi razziali, ed ideò un sistema per la loro classificazione. Furono le ricerche di Galton, congiunte quelle svolte da Sir Edward Henry (poliziotto britannico) ad impostare su base scientifica lo sviluppo e le applicazioni di questo metodo, favorendone quindi l'effettiva adozione nelle aule giudiziarie. Uno dei principali problemi che Galton ed Henry dovettero affrontare fu quello riguardante la tecnica di raccolta ed estrazione delle peculiarità dell'impronta. A queste difficoltà trovò soluzione l'italiano Giovanni Gasti, criminologo e funzionario di polizia, il quale riprese il metodo proposto da Herschel per il riconoscimento delle impronte e lo perfezionò in modo da renderlo uno strumento affidabile per le investigazioni e inattaccabile in ambito giudiziario. Il metodo Gasti è tutt'ora utilizzato dalle principali polizie del mondo. Un'autentica rivoluzione nell'utilizzo delle impronte digitali, soprattutto a fini criminologici, si è avuta una dozzina di anni fa, quando la crescente potenza dei sistemi di elaborazione permise alla polizia americana di mettere a punto il primo sistema AFIS - Automatic Fingerprint Identification System. Grazie all'utilizzo di questo dispositivo informatico, collegato ad un database, l'impronta digitale ha raggiunto l'attuale importanza.

1.2.2 – Anatomia delle impronte digitali

Il riconoscimento basato sulle impronte digitali, come già detto è di per se una tecnica molto efficiente ed affidabile per l'identificazione degli individui. Inizialmente lo studio delle impronte digitali si basava sullo studio delle forme di particolari punti rintracciabili all'interno di un'impronta, così come descritto dal metodo di *Henry*. Un'impronta digitale è costituita da un insieme di linee dette *Ridge line* o *Creste*, che scorrono parallelamente, formando un disegno denominato *Ridge Pattern*. Due ridge line sono separate da una ipotetica linea, denominata *Flow Line* o *Valle* (vedi Figura 1.4).

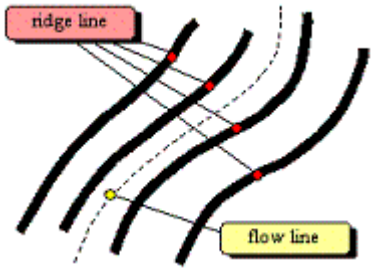


Figura 1.4 – Esempio di *ridge line* e *flow line*

Analizzando l'impronta è, in particolare le creste e il loro andamento nel *dermatoglifo* (cioè quel particolare disegno che le creste cutanee formano su ogni polpastrello delle dita), è possibile notare che queste assumono delle particolari forme, dette *Regioni Singolari*. Queste regioni sono riconducibili a tre tipologie distinte:

- *Core*: caratterizzate da un'insieme di creste che hanno un andamento a forma di “U”
- *Whorl*: caratterizzate da una struttura circolare (ad “O”)
- *Delta*: caratterizzate da una struttura a “Δ”.

In Figura 1.5 possiamo vederne alcuni esempi.

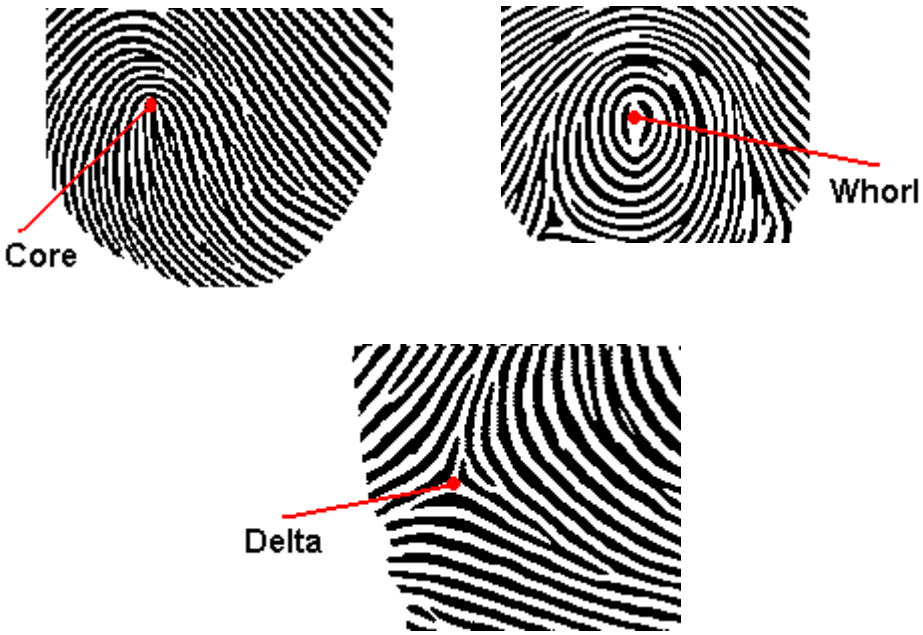


Figura 1,5 – Esempi di andamenti delle *creste* in una impronta

L'andamento delle *flow line* può essere facilmente ricostruito e ben descritto dalla cosi detta *Immagine Direzionale.* Questa immagine, ovviamente estratta dall'impronta digitale, è una matrice discreta di vettori che danno informazioni riguardo l'orientamento e l'andamento locale della ridge line all'interno dell'impronta. Questa immagine direzionale (vedi Figura 1.6) può essere calcolata, ad esempio, tramite l'applicazione dell'operatore di *Sobel* (un operatore derivativo che utilizza maschere di dimensione 3x3) lungo la direzione orizzontale e verticale :

$$Sx=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \quad Sy=\begin{bmatrix}-1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

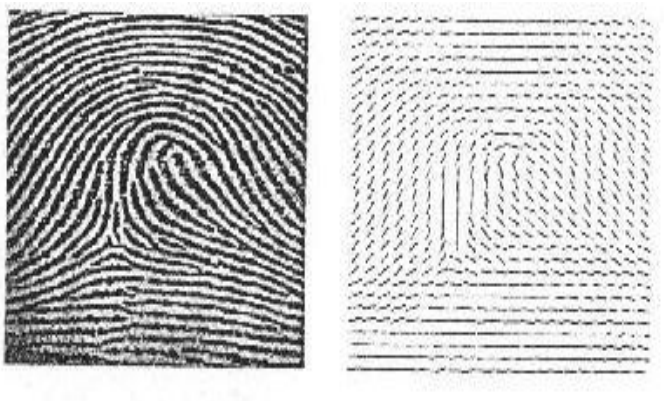


Figura 1.6 – Impronta e rispettiva immagine direzionale

A questo primo studio dell'impronta si aggiungono i risultati ottenuti dagli studi effettuati da Galton, il quale, da una analisi più accurata delle ridge line dell'impronta, ha individuato altre caratteristiche fondamentali, delle micro singolarità chiamate *Minuzie* o, dal nome dello scopritore, *Caratteristiche di Galton.* Queste sono principalmente determinate in termini di *Biforcazioni* e *Terminazioni* delle ridge line, e sono di fondamentale importanza per la discriminazione delle impronte. È comunque possibile individuare all'interno dell'impronta delle minuzie complesse, date dall'unione di due minuzie elementari. Difatti tutti gli algoritmi e i sistemi automatici di estrazione delle minuzie e di identificazione/validazione, considerano soltanto minuzie in termini di terminazioni e biforcazioni, in quanto da queste è possibile ricavare le minuzie "composte". Nella Figura 1.7 . vediamo i tipi di *minuzie* rintracciabili in un'impronta. La parte centrale dell'impronta dove normalmente sono dislocate le singolarità è detta *Pattern Area*, ed è delimitata da due linee principali, denominate *Type Line*, cioè quelle due linee che separano la pattern area dal resto dell'impronta. Le minuzie insieme alla forma e la direzione delle ridge line, costituiscono le *macro caratteristiche* dell'impronta, informazioni utilizzate da gran parte degli algoritmi per la classificazione delle impronte digitali. Le impronte digitali possono essere suddivise in 8 classi principali, in base al numero e alla posizione delle singolarità:

1. *Impronta ad arco semplice:* in questa impronta le creste entrando da uno dei due lati dell'impronta, in maniera crescente arrivano al centro dell'impronta, per poi uscire in maniera decrescente dal lato opposto.

2. *Impronta ad arco a tenda*: impronta con lo stesso andamento del caso 1. ma con un angolo o una piega delle ridge line a forma di Delta al centro dell'impronta.

3. *Impronta ad occhiello ulnare*: impronte in cui una o più creste entrano dal lato destro, si ripiegano, superano la linea immaginaria determinata dal Core ed escono dallo stesso lato d'ingresso.

4. *Impronta ad occhiello radiale*: come la precedente, ma piegate dal lato opposto.

5. *Impronta a spirale semplice*: impronte con almeno due delta e una figura chiusa.

6. *Impronta a doppio occhiello*: impronte con due delta e due loop distinti e accavallati.

7. *Impronta ad occhiello centrale a sacca*: impronta che mostra al centro un occhiello simile ad una sacca.

8. *Impronta casuale*: il cui pattern non ha forma specifica.

Possiamo vederne alcuni esempi in Figura 1.7

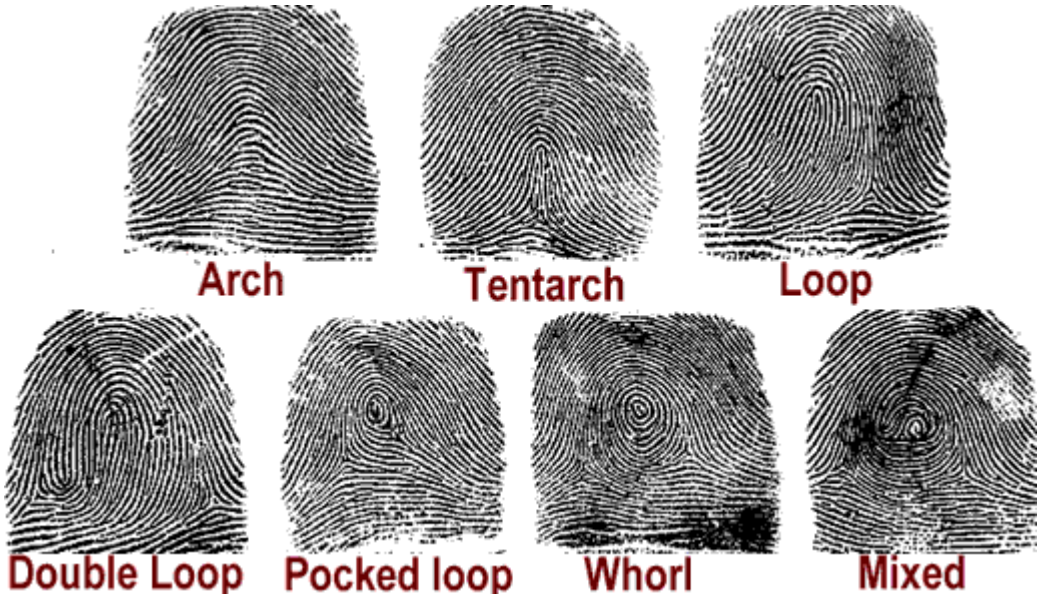


Figura 1.7 - Esempi di impronte in base alle *singolarità*.

1.2.3 – Estrazione delle caratteristiche

Per ogni impronta acquisita vengono estratte, in base alle tecniche previste dall'algoritmo, le minuzie (cioè terminazioni e biforcazioni), in modo tale da ottenere un *finger code* utilizzabile per gli scopi previsti dal sistema biometrico. Prima di effettuare qualunque analisi dell'impronta al fine di individuare le minuzie, vengono eseguite delle operazioni sull'immagine dell'impronta prima di poter applicare l'algoritmo di estrazione. Solitamente l'impronta viene prima *segmentata*, al fine di selezionare l'impronta ed eliminare porzioni di immagine inutili. Successivamente viene applicato un *filtro di Gabor* che permette di migliorare le caratteristiche dell'impronta stessa, ricostruendo porzioni di impronta mancanti o poco chiare, e di separare in maniera netta le creste dalle valli. Infine vengono applicate la *binarizzazione* (*thresholding*), in modo da ottenere una immagine binaria, con valori 0-255, e una *scheletrizzazione*, per ridurre le ridge line ad una struttura minimale. In Figura 1.8 possiamo vedere il risultato dell'applicazione di queste operazioni.

Figura 1.8 – In sequenza: impronta originale, sogliata e scheletrizzata

Dopo queste operazioni è possibile applicare l'algoritmo previsto per l'estrazione delle minuzie. Per ogni minuzia individuata all'interno dell'impronta, vengono memorizzate la posizione, il tipo di minuzia e l'angolo che la tangente alla stessa forma con la direzione orizzontale. Tramite la raccolta di tutte queste informazioni è possibile realizzare un finger code che sia univoco per ciascun individuo catalogato.

1.2.4 – Tecniche di acquisizione

Vi sono svariate metodologie per l'acquisizione delle impronte digitali. Ovviamente lo sviluppo tecnologico ha portato in pochi anni a sviluppare soluzioni sempre più prestanti in termini di tempo e di qualità delle informazioni acquisite. La tecnica più semplice prevede che l'impronta venga acquisita tramite l'impressione del polpastrello dell'individuo, “sporcato” con dell'inchiostro, su di un foglio di carta. Come detto, con l'evoluzione tecnologica, sono stati sviluppati nuovi dispositivi per l'acquisizione dell'impronta, come sensori opto – elettronici, sensori capacitivi e sensori termici. Nei prossimi sotto paragrafi si fornisce una breve descrizione di questi dispositivi.

1.2.4.1 – Sensori Opto-Elettronici

Questi sensori sono stati i primi ad apparire sul mercato. Inizialmente costituiti da una sorgente luminosa, un prisma e un sensore CCD/CMOS per l'acquisizione dell'immagine. Il funzionamento di questi dispositivi si basa sul principio della riflessone/assorbimento della luce. Difatti, grazie alle microparticelle di acqua presenti sulle ridge line, le quali assorbono la luce che le colpiscono, permettono di ottenere una immagine acquisita che sarà scura in corrispondenza appunto delle ridge line, mentre risulterà essere chiara in corrispondenza delle valli, che, non assorbendo la luce emessa dalla sorgente, permettono a questa di giungere al sensore stesso illuminando la scena. Il problema di questi dispositivi è la possibilità di realizzare delle false impronte,a causa della semplicità costruttiva. Ad esempio il lettore di impronte potrebbe acquisire correttamente un impronta anche se realizzata tramite l'utilizzo di lucidi da proiezione. Per ovviare a questo problema sono state introdotte all'interno di questi dispositivi un sistema di lenti focali, poste davanti al sensore CCD.

1.2.4.2 – Sensori Capacitivi

Questi dispositivi fruttano il principio di funzionamento dei condensatori, nei quali la capacità è inversamente proporzionale alla distanza tra le armature. Ogni punto del sensore corrisponde ad una armatura di un condensatore, mentre la seconda armatura verrà sostituita dal polpastrello. In corrispondenza delle creste, la distanza tra le due "armature" sarà ridotta, e questo fornirà un valore di capacità maggiore in output. Invece, in presenza di valli, la capacità risulterà minore. Questi due valori, tradotti in maniera opportuna, permettono di avere una buona ricostruzione dell'impronta. Anche se questi dispositivi sono più performanti dei lettori opto-elettronici, risultano avere un tempo di vita minore, in quanto sono soggetti alle cariche elettrostatiche che si accumulano sulle dita, che a lungo andare rischiano di danneggiare il dispositivo stesso. Inoltre questi dispositivi sono affetti da rumore causato da dita troppo umide o troppo secche, che possono portare immagini poco definite o frastagliate.

1.2.4.3 – Sensori Termici

Questi dispositivi rilevano appunto la differenza di temperatura che vi è tra le creste e le valli del dermatoglifo. Il sensore è quindi formato da una serie di micro sensori di temperatura. Ovviamente questo dispositivo funziona in maniera ottimale quando vi è una consistente differenza di temperatura tra creste e valli. È altresì soggetto alle condizioni ambientali, quindi a bruschi cali di temperatura, che possono far tendere la temperatura del dito a quella del sensore, rischiano di rendere l'impronta acquisita di bassa qualità.

Caratteristiche importanti per un lettore di impronte digitali sono:

- Risoluzione, ovvero il numero di punti per pollice (dpi) con cui l'immagine viene acquisita. È importante che questo parametro non scenda al di sotto della soglia dei 250dpi, per poter ottenere un buon risultato.

- Dimensione del sensore, cioè la parte che si occupa di "leggere" l'impronta digitale. Questo parametro è importante al fine di non ottenere una impronta incompleta.

- Nitidezza e contrasto.

Ciò che distingue una buona impronta da una cattiva è il modo in cui viene fatta l'acquisizione della stessa. Per tutti i dispositivi è importante che il dito non venga posto sul sensore applicando pressioni eccessive o con angoli di attacco troppo accentuati, che non vi siano pieghe cutanee che possano interrompere l'andamento delle creste e ancora che il tempo di acquisizione non sia troppo ridotto (parametro importante nel caso di sensori capacitivi, che solitamente prevedono l'acquisizione tramite scorrimento del dito sul sensore). Questi piccoli accorgimenti permettono di ottenere impronte nitide dalle quali è possibile estrarre un consistente numero di caratteristiche. Estratte le caratteristiche, l'ultima operazione da compiere è quella che prevede la memorizzazione delle informazioni estratte in un database, al fine di poter effettuare dei confronti nel caso in cui sia necessario identificare un individuo. Per garantire la privacy della persona proprietaria dell'impronta digitale, dopo la fase di estrazione delle caratteristiche, queste vengono tradotte dal sistema in un finger code univoco per ciascuna impronta. È proprio questo codice che viene memorizzato all'interno del database, e non l'immagine dell'impronta acquisita. In questo modo si assicura una univocità delle informazioni estratte, la privacy dell'utente, in quanto non c'è nessun

dato (come nome o matricola) legate all'impronta, e si riduce notevolmente la dimensione del database stesso. Difatti, a parità di dimensioni del database, questa tecnica permette di memorizzare un numero molto elevato di impronte rispetto alla memorizzazione delle immagini.

CAPITOLO 2

CRITTOGRAFIA

2.1 Introduzione alla crittografia

Per qualunque organizzazione la sicurezza delle informazioni è un concetto di fondamentale importanza, in quanto è l'unico modo che l'azienda ha di preservare il proprio lavoro e, più in generale, i suoi interessi. Con l'avvento dei sistemi informatici si è resa necessaria l'adozione di strumenti automatizzati che si occupassero della protezione dei dati sensibili. Oltre all'utilizzo di password per la protezione dei sistemi informatici, con l'avvento delle reti e del WEB, si è reso necessario lo studio di sistemi che potessero proteggere i dati e le informazioni scambiate dagli utenti su queste reti. Da qui nasce l'esigenza della crittografia delle informazioni.

Volendo dare una definizione esatta del termine *crittografia* , possiamo dire che è quella

"tecnica che permette di trasformare, tramite specifici algoritmi, un testo in un linguaggio tradizionale, detto plaintext, o testo in chiaro, in un testo che utilizza un linguaggio non comprensibile all'uomo, detto ciphertext, o testo cifrato".

Dal testo cifrato è possibile estrarre il testo originale soltanto se si è in possesso della cosi detta chiave di crittografia, e se si è a conoscenza dell'algoritmo utilizzato per cifrare il messaggio. Questa operazione è nota come *de-crittografia.* La crittografia si divide principalmente in due rami, in base al "tipo" di algoritmo che si sta utilizzando per proteggere il testo e al tipo di chiave utilizzata. Il primo ramo comprende tutti gli algoritmi di crittografia *simmetrica*, anche detti algoritmi a chiave privata, mentre il secondo comprende gli algoritmi di crittografia *asimmetrica* o a chiave pubblica.

2.1.1 Crittografia simmetrica

Un sistema di crittografia simmetrico prevede che vi siano i seguenti elementi:

- *Testo in chiaro*: il messaggio originale da crittografare
- *Algoritmo di crittografia:* l'algoritmo che esegue le trasformazioni sul testo in chiaro
- *Chiave segreta:* un valore indipendente dal testo in chiaro, scelto e condiviso dai due utenti che intendono comunicare, prima che avvenga la cifratura del messaggio. L'algoritmo utilizza la chiave per produrre il testo cifrato, e a chiavi differenti, per lo stesso testo, corrisponderanno testi cifrati differenti.
- *Testo cifrato*: il messaggio fornito in output dall'algoritmo di crittografia, dipendente dalla chiave, che verrà trasmesso sulla rete pubblica.

- *Algoritmo di de-crittografia:* corrisponde all'applicazione dell'algoritmo di crittografia al contrario (da qui il nome di crittografia simmetrica). Accetta come input il testo cifrato e la chiave di crittografia, e fornisce in output il testo in chiaro.

Nella Figura 2.1 e sottostante troviamo uno schema di applicazione della crittografia simmetrica.

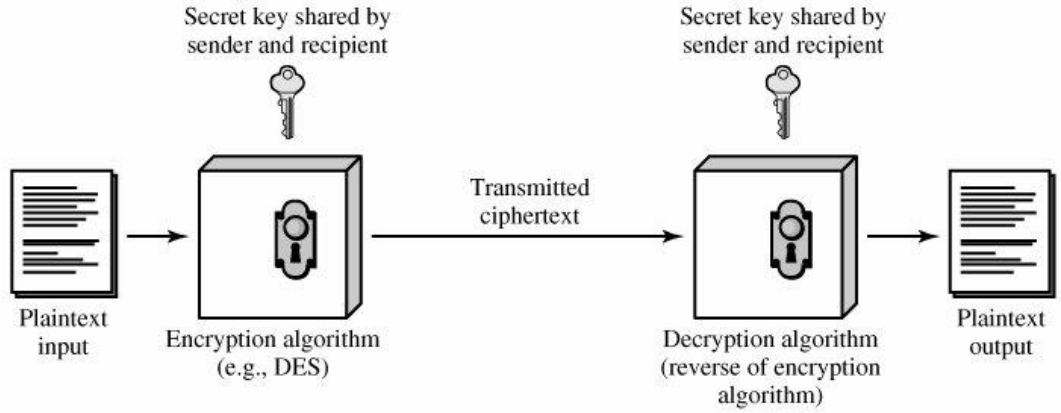


Figura 2.1 – Schema applicativo della crittografia simmetrica

Per un algoritmo di crittografia simmetrico è importante che vi siano due requisiti fondamentali:

- l'algoritmo deve essere talmente robusto da non permettere ad un estraneo, che sia in possesso un certo numero di testi cifrati, e che conosca le operazioni compiute dall'algoritmo, di ricavare il testo in chiaro.
- Il mittente e il destinatario del messaggio devono aver ricevuto la chiave di crittografia in modo sicuro, quindi o tramite l'utilizzo di un canale di comunicazione sicuro, oppure richiedendola ad un ente di distribuzione pubblico di chiavi di crittografia.

Il problema degli algoritmi simmetrici è proprio questo, cioè ottenere e mantenere segreta la chiave di cifratura.

Gli algoritmi simmetrici si dividono, in base alle operazioni che questi applicano al testo da cifrare, in algoritmi di *sostituzione* e in algoritmi di *trasposizione*:

- Gli algoritmi di *sostituzione* sono quegli algoritmi che effettuano un mapping tra ogni carattere del testo in chiaro ed un altro carattere, definito dall'algoritmo e in base alla chiave di crittografia;
- Gli algoritmi di *trasposizione* sono quegli algoritmi che scambiano la posizione dei caratteri del testo in chiaro.

Tra gli algoritmi di sostituzione troviamo:

- Cesare
- Monoalfabetica e Polialfabetica
- Playfair
- Hill
- One Time Pad

Mentre per gli algoritmi di trasposizione abbiamo:

- Rail Fence
- Macchine a rotazione

Più in generale, tra gli algoritmi di crittografia simmetrica più conosciuti ed utilizzati troviamo il *DES* (*Data Encryption Standard*) e le sue varianti *Double-DES* e *Triple-DES* e l'algoritmo *AES* (*Advanced Encryption Standard*).

Gli algoritmi a chiave simmetrica soffrono di due principali categorie di attacchi, quelli a *Forza Bruta* e quelli ad *Analisi Crittografica*. Negli attacchi a *Forza Bruta*, si tentano tutte le possibili chiavi utilizzabili per quell'algoritmo su di una porzione di testo cifrato, finchè non si ottiene una traduzione corretta del testo. Ovviamente la riuscita di questa tecnica dipende dalla lunghezza della chiave e dalla tecnica di cifratura. Invece, negli attacchi ad *Analisi Crittografica*, ci si basa sulla natura dell'algoritmo utilizzato e sfrutta qualche conoscenza sulle caratteristiche generali del testo in chiaro (ad esempio la lingua in cui è scritto, dalla quale si può dedurre la frequenza di occorrenza delle lettere). Nello specifico l'analisi crittografica si divide in due categorie, l'analisi *differenziale* e quella *lineare*. La prima prevede l'analisi delle trasformazioni che il testo in chiaro subisce durante le varie fasi della crittografia, mentre l'altra si basa sull'individuazione delle approssimazioni lineari per descrivere le trasformazioni eseguite dall'algoritmo (applicata principalmente su testi cifrati con DES).

2.1.2 Crittografia asimmetrica

Abbiamo visto fin ora delle tecniche di crittografia simmetriche, cioè che. utilizzano la stessa chiave per cifrare e decifrare il testo. Purtroppo queste tecniche soffrono di numerosi attacchi, che ne riducono l'affidabilità e la sicurezza. Inoltre, come detto prima, è necessario che mittente e destinatario decidano e si scambino la chiave prima di procedere con l'applicazione dello stesso al testo, utilizzando quando possibile canali sicuri. Questa è un'altra possibile falla alla sicurezza del sistema, in quanto un attaccante potrebbe intercettare la comunicazione e venire a conoscenza della chiave. Per ovviare ai problemi cui si può incorrere con la crittografia a chiave privata, ci si è diretti verso nuove frontiere al fine di introdurre dei sistemi di crittografia che potessero fornire livelli di sicurezza maggiore di quelli garantiti dagli algoritmi quali DES e AES. Ci si è così indirizzati verso lo sviluppo della così detta *crittografia a chiave pubblica*. Essa si distacca in modo radicale da tutte le tecniche sviluppate in precedenza. Innanzitutto gli algoritmi a chiave pubblica si basano su funzioni matematiche anziché su operazioni di permutazione e sostituzione. Inoltre utilizza una

chiave asimmetrica, cioè la chiave utilizzata per cifrare il testo è differente, ma correlata, a quella utilizzata per la de-crittografia, ed è possibile scegliere in maniera arbitraria quale delle due chiavi generate utilizzare per la crittografia. Ovviamente, una volta fatta questa scelta, bisogna rispettarla, utilizzando quindi l'altra per la de-crittografia. Il concetto da cui partirono i due inventori della crittografia a chiave pubblica, *Whitfield Diffie* e *Martin Hellmann*, per sviluppare questo tipo di crittografia fu quello di superare due principali problemi della crittografia a chiave privata. Il primo era quello dello scambio delle chiavi tra mittente e destinatario, in quanto anche se tramite l'utilizzo di un centro di distribuzione delle chiavi si poteva incappare in un eventuale furto della chiave segreta. Il secondo invece riguardava l'autenticazione dei soggetti, cioè trovare un metodo per essere certi che un determinato messaggio fosse stato effettivamente prodotto da un determinata persona.

Un algoritmo di crittografia a chiave pubblica deve soddisfare i seguenti requisiti:

- deve essere computazionalmente semplice generare una coppia di chiavi pubblica e privata;
- deve essere computazionalmente semplice cifrare un messaggio conoscendo la chiave pubblica;
- deve essere computazionalmente semplice decifrare un messaggio conoscendo la chiave privata;
- deve essere computazionalmente impossibile ricavare il messaggio in chiaro conoscendo algoritmo e chiave pubblica;
- deve essere computazionalmente impossibile ricavare la chiave privata conoscendo algoritmo e chiave pubblica.

Un possibile riassunto delle operazioni necessarie per applicare la crittografia asimmetrica ad un messaggio, secondo le teorie di Diffie ed Hellmann, vede i seguenti passi:

- Gli utenti che intendono comunicare generano le coppie di chiavi pubblica/privata secondo le tecniche previste dall'algoritmo scelto;
- Gli utente pubblicano in un apposito registro o in un centro di distribuzione delle chiavi le proprie chiavi pubbliche, mentre mantengono rigorosamente segrete le chiavi private;
- Se l'utente A vuole inviare un messaggio all'utente B, dovrà applicare la crittografia utilizzando la chiave PUBBLICA dell'utente B (Pu_b);
- L'utente B, al momento della ricezione del messaggio, per poterlo decrittografare, utilizzerà la propria chiave PRIVATA (PR_A).

In questo modo si garantisce quindi che nessuno possa decrittografare il messaggio, diretto all'utente B, in quanto soltanto lui conosce la sua chiave privata (PR_A), l'unica in grado di decrittografare il messaggio. Abbiamo in questo modo garantito la segretezza del messaggio,

superando così uno dei principali problemi degli algoritmi di crittografia simmetrica. Come detto in precedenza, il secondo dei punti sui quali ci si è basati in fase di definizione delle specifiche della crittografia a chiave asimmetrica, è quello dell'autenticazione del mittente del messaggio. Questo problema può essere superato applicando al messaggio crittografato con la chiave pubblica del destinatario, una ulteriore crittografia, sempre con lo stesso algoritmo, ma utilizzando la chiave PRIVATA del mittente. In questo modo, il primo livello di crittografia può essere facilmente abbattuto applicando la chiave pubblica del mittente (per la de-crittografia). Inoltre, essendo stato crittografato il messaggio utilizzando la chiave privata del mittente, ed essendo lui l'unico a conoscerla, ha assicurato l'autenticazione del messaggio stesso. In Figura 2.2 è possibile vedere uno schema applicativo completo della crittografia asimmetrica, comprendente segretezza e autenticazione del messaggio.

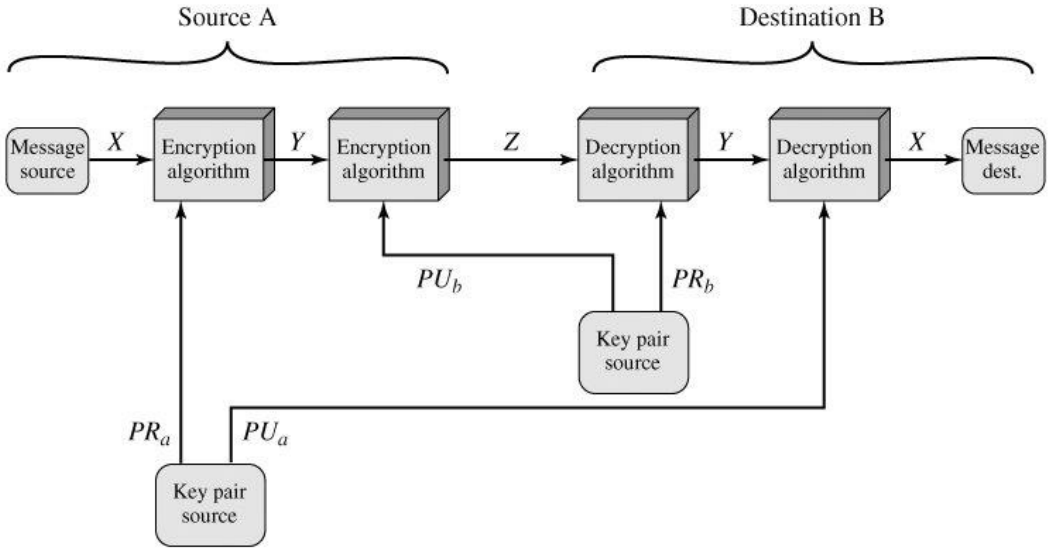


Figura 2.2 – Schema applicativo della crittografia simmetrica

2.2 Algoritmo RSA

RSA risulta essere una delle prime soluzione ad utilizzare in maniera ottimale la crittografia a chiave pubblica. Sviluppato da *Ronald Rivest, Adi Shamir e Len Adleman* (le iniziali dei tre danno appunto il nome all'algoritmo), ricercatori del *MIT*, nel 1977, è da allora l'algoritmo più accettato ed implementato nell'ambito della crittografia a chiave pubblica. Lo schema *RSA* soddisfa tutti i requisiti della cifratura asimmetrica e può essere utilizzato per varie applicazioni:

- *cifratura e decifratura di messaggi*, con garanzia dei requisiti di autenticazione, segretezza, integrità e non ripudiabilità;
- *creazione di firme digitali*, solitamente in combinazione con funzioni hash;
- realizzazione di *protocolli per scambio* sicuro di *chiavi segrete di sessione*.

2.2.1 Descrizione dell'Algoritmo

L'RSA è una cifratura a blocchi in cui il testo e il testo cifrato sono interi compresi tra *0* ed *n-1,* per un dato valore di *n.* Normalmente *n* è pari a *1024 bit.* L'idea di RSA è molto semplice e si basa sulla *difficoltà nel fattorizzare grandi numeri***:** mentre è molto facile moltiplicare tra di loro due grandi *numeri primi*, risulta difficile fattorizzare il loro prodotto. In questo modo, il prodotto può essere reso pubblico insieme alla chiave di codifica. Supponiamo che due individui, indicati con A e B, abbiano necessità di comunicare. In generale il testo in chiaro e il testo cifrato sono generati mediante le seguenti formule:

$C = M^e \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n.$

con:

- *n* : noto ad A e B, calcolato come il prodotto tra due numeri primi *p* e *q*
- *e* : noto solo ad A,
- *d* : noto solo a B.

Inoltre risulta che *e* ed *d* sono inversi moltiplicativi modulo $\phi(n)$, dove $\phi(n)$ è la funzione *Toziente di Eulero* (ovvero il numero di interi minori positivi minori di *n*). Essendo $n=pq$, si ha che $\phi(n) = \phi(pq) = (p-1)(q-1)$.

Da questo è possibile infine ricavare la relazione tra *e* e *d* (calcolato tramite l'algoritmo di Euclide esteso), in quanto risultano essere:

$ed \bmod \phi(n) = M$

da cui le relazioni di congruenza

$ed \equiv 1 \bmod \phi(n)$

$d \equiv e^{-1} \bmod \phi(n)$

Questo vale soltanto se *e* ed *d* sono primi relativi (ovvero se sono primi tra loro, cioè non hanno fattori primi in comune, quindi il loro unico divisore comune è 1) di $\phi(n)$, di conseguenza il $\text{mcd}(\phi(n), d) = 1$. Risulta evidente che la chiave pubblica e la chiave privata per questo algoritmo sono:

$PU = \{ e , n \}$

$PR = \{ d , n \}.$

E' importante, per un corretto funzionamento dell'algoritmo, che siano soddisfatti i seguenti requisiti:

- sia possibile individuare i valori di *e, d, n,* in modo tale che $M^{ed} \bmod n = M;$
- sia relativamente facile calcolare M^e ed $C^d \bmod n$;

- sia impossibile calcolare *d* sulla base di *e* ed *n* (oppure *e* sulla base di *d* ed *n*).

Riassumiamo dunque lo schema *RSA*, che ci permette di applicare le formule di crittografia definite in precedenza (supponendo di porre la coppia di chiavi pubblica e privata come definito sopra):

- p e q sono due numeri primi, privati e scelti
- n = pq è pubblico e calcolato
- e , con mcd ($\phi(n), e) = 1$ ed $1 < e < \phi(n)$, pubblico e scelto.
- $d \equiv e^{-1} \bmod \phi(n)$ privato e calcolato.

Nel paragrafo successivo vedremo come può essere riassunto lo schema completo di applicazione dell'algoritmo di crittografia *RSA* ad un testo *M*:

2.2.2 Applicazione dell'algoritmo

La sequenza di passi per l'applicazione dell'algoritmo *RSA* può essere riassunta come segue (vedi Figura 2.3):

Generazione della chiave:

1. Scegliere *p* , *q* *p* e *q* entrambi primi, con $p \neq q$;
2. Calcolare $n = p * q$
3. Calcolare $\phi(n) = \phi(pq) = (p-1)(q-1)$.
4. Selezionare l'intero *e* t.c. $mcd(\phi(n), e) = 1$ ed $1 < e < \phi(n)$;
5. Calcolare *d* $d \equiv e^{-1} \bmod \phi(n)$;
6. Chiave pubblica $PU = \{ e , n \}$;
7. Chiave privata $PR = \{ d , n \}$.

Crittografia:

1. Testo in chiaro $M < n$
2. Testo cifrato $C = M^e \bmod n$

Decrittografia:

1. Testo cifrato C
2. Testo in chiaro $M = C^d \bmod n$.

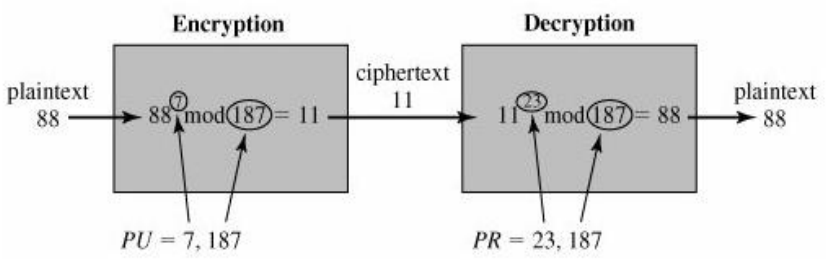


Figura 2.3 - Applicazione dell'algoritmo *RSA*

Prima di poter applicare l'algoritmo, i due partecipanti alla comunicazione devono generare la propria coppia di chiavi pubblica e privata. Questo comporta la scelta dei due numeri primi *p* e *q*, la scelta di *e* ed il conseguente calcolo di *d* (o viceversa). Poiché il prodotto $n = p*q$ sarà noto a chiunque, dovendo essere reso pubblico, è importante scegliere i due numeri primi in maniera tale da non poterli ricavare tramite semplici e ovvi calcoli. Per questo motivo devono essere scelti in un insieme sufficientemente esteso. La tecnica più utilizzata per l'individuazione di questi numeri consiste nello scegliere a caso un numero primo dispari dell'ordine di grandezza desiderato, ed applicare uno qualunque dei test di primalità esistenti (ad esempio l'algoritmo *Miller-Rabin*) per stabilire se questo numero scelto è *probabilmente* primo. Terminata questa verifica, determinati quindi p e q, sarà possibile procedere con il processo di generazione della chiave come specificato in precedenza.

2.2.3 Sicurezza dell'algoritmo

Gli algoritmi di cifratura a chiave pubblica si basano sulla difficoltà di determinare il testo in chiaro, conoscendo quello cifrato, e di determinare la chiave privata, conoscendo quella pubblica. Per violare un sistema di cifratura RSA è necessario dedurre il valore dell'esponente privato d, a partire dalla chiave pubblica $PU = \{ e , n \}$, la cui conoscenza permetterebbe di accedere a tutti i messaggi cifrati e di falsificare la firma digitale dell'utente proprietario delle chiavi. Determinare l'esponente *d* comporta la necessità di conoscere il valore $\phi(n)$, ovvero dei fattori primi che compongono *n*, *p* e *q*.

La resistenza a molti anni di analisi crittografica dimostra che *RSA* è un algoritmo estremamente robusto.

Varie tipologie di attacco per questo algoritmo sono:

- *Forza bruta*: comporta l'applicazione di tutte le possibili chiavi di crittografia. È sufficiente utilizzare chiavi di grandezza opportuna per rendere questo attacco impraticabile, quindi scegliere *p* e *q* in uno spazio sufficientemente grande.
- *Attacchi matematici*: vari approcci, tutti riconducibili alla fattorizzazione di *n*.

- *Attacchi a tempo*: basati sul tempo di esecuzione dell'algoritmo di crittografia, consentono di determinare una chiave privata analizzando il tempo impiegato dal calcolatore per cifrare o decifrare i messaggi.
- *Attacchi a testo cifrato scelto*: basati sulle caratteristiche dell'algoritmo RSA, nei quali l'attaccante ha la possibilità di scegliere un particolare testo cifrato ed ottenere il corrispondente testo in chiaro. In pratica vengono determinati dei valori numerici particolari, i quali, firmati dal mittente, forniscono utili informazioni per l'analisi crittografica.

CAPITOLO 3

FRATTALI

3.1 Introduzione

Ai frattali si è giunti partendo da differenti approcci e seguendo vie di indagine diverse, che all'inizio non avevano tra loro alcun apparente elemento in comune; solo in un secondo tempo ci si è accorti della stretta parentela che intercorre tra i risultati ottenuti nei diversi settori di ricerca. Gli approcci sono i seguenti :

a) Analitico

b) Geometrico;

c) Fisico-dinamico

Un frattale è un insieme F che gode delle seguenti proprietà:

1. **Autosimilarità** : F è unione di un numero di parti che, ingrandite di un certo fattore, riproducono tutto F; in altri termini F è unione di copie di se stesso a scale differenti.
2. **Struttura fine** : F rivela dettagli ad ogni ingrandimento.
3. **Irregolarità** : F non si può descrivere come luogo di punti che soddisfano semplici condizioni geometriche o analitiche.
4. **Dimensione**: si definisce un nuovo concetto di dimensione differente dalla concezione classica che sia maggiore della dimensione topologica.

3.2 Caos deterministico

Il termine deterministico è associato all'idea di fenomeni prevedibili, mentre il caos è riferito a situazioni caratterizzate da assenza di regole e da imprevedibilità. La scoperta del caos deterministico, invece, mostra come modelli matematici deterministici sono in grado di generare andamenti estremamente complessi e imprevedibili, tanto da risultare quasi indistinguibili da sequenze di eventi generati attraverso processi aleatori. Consideriamo ad esempio il semplicissimo moto di una particella puntiforme che salta da un punto ad un altro lungo una linea. La posizione in un qualunque istante sarà determinata, una volta nota la sua posizione iniziale, specificando un algoritmo per il calcolo dei salti. Prendiamo una linea compresa tra 0 e 1 ed usiamo il raddoppiamento di orologio come algoritmo per generare l'itinerario del punto. In pratica il raddoppiamento di un numero minore di ½ procede come al solito mentre un numero maggiore di ½ , quando viene raddoppiato, supera 1 e si considera solo la parte decimale. Ad esempio 0,6 raddoppia a 1,2 che diventa 0,2. Nonostante la sua semplicità, questo algoritmo genera un comportamento talmente complesso ed irregolare da risultare completamente imprevedibile. In altre parole, due

numeri iniziali molto vicini, produrranno sequenze di salti che alla fine potranno essere completamente diverse. La storia del punto quindi, benché completamente deterministica, è talmente sensibile alle condizione iniziale che qualsiasi indeterminazione relativa a questa informazione, per quanto piccola, è sufficiente a distruggere la capacità di previsione dopo un numero finito di salti. Un altro esempio semplice di caos lo si ottiene con l'equazione logistica utilizzata per studiare la crescita di popolazioni di animali che hanno una elevata natalità ed una vita breve come molti insetti. Essa è la seguente

$$P(t+1) = R \cdot P(t)(1-P(t)),$$

dove P è la numerosità della popolazione all'istante t ed R è il fattore di crescita. Il fattore (1-P) rappresenta ciò che può frenare la crescita come mancanza di cibo o condizioni climatiche sfavorevoli. Prendiamo come valore iniziale P(0) = 0,4 e osserviamo che cosa accade iterando la formula per valori diversi di R.

TEMPO	$R = 2$	$R = 1+\sqrt{5}$	$R = 4$
0	0,4	0,4	0,4
1	0,48	0,776656296	0,96
2	0,4992	0,561332525	0,1536
3	0,49999872	0,7968439275	0,52002816
4	0,5	0,523866589	0,9983954912
5	0,5	0,8089732544	0,0064077373
6	0,5	0,5000874618	0,0254667128
7	0,5	0,8090169502	0,0992726372
8	0,5	0,5000000268	0,3576703229
9	0,5	0,809016975	0,918969052
10	0,5	0,5000000268	0,2978597337
11	0,5	0,809016975	0,836557251
12	0,5	0,5000000268	0,5469168673
13	0,5	0,809016975	0,9911952303

14	0,5	0,5000000268	0,0349089831
15	0,5	0,809016975	0,134761384

Tabella 3.1: Stima della crescita della popolazione di animali

Analizzando la tabella notiamo che mentre nel primo caso le soluzioni si stabilizzano già alla quarta iterazione su un valore fisso e nel secondo alla settima, oscillando fra due valori limite, nel terzo vi è un comportamento caotico. Se cambiamo di poco il valore iniziale, mentre nel primo e secondo caso ciò non ha praticamente alcun effetto, nel terzo caso basta una differenza di 10^{-5} per originare, dopo poche iterazioni, nuovi valori totalmente diversi dai precedenti. E' interessante eseguire un grafico di tale equazione facendo variare il valore di R.

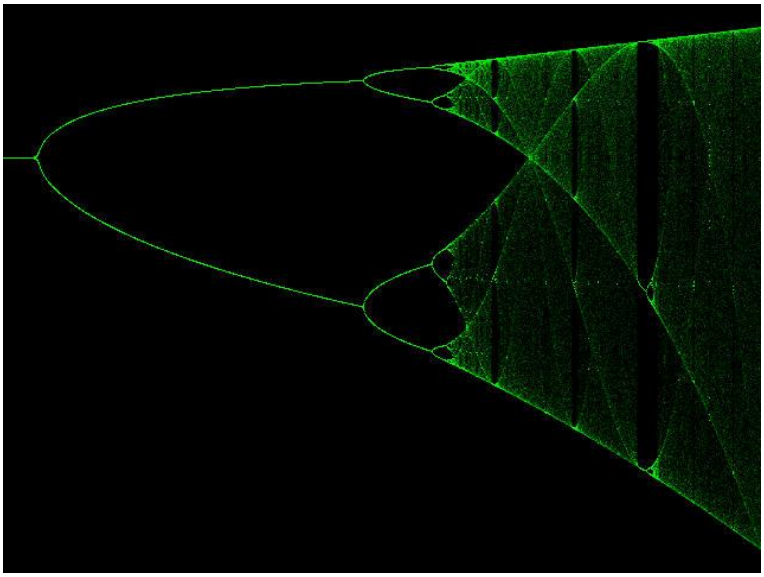


Figura 3.1: Grafico ottenuto facendo variare il fattore *R*

Per valori piccoli di R vi è un solo valore di P che soddisfa l'equazione, ma nel punto critico R = 3 la curva si spezza in due. Più avanti, si presentano successive biforcazioni che stanno ad indicare che P può oscillare fra un numero di valori sempre maggiore e per il valore critico

$$R = 3{,}5699...$$

si hanno una infinità di ramificazioni e ciò rappresenta l'inizio del caos. Mano a mano che ci si avvicina a questo punto, le ramificazioni si infittiscono sempre di più e, se si

confrontano gli spazi fra ramificazioni successive, si trova che ciascuno di essi è leggermente più piccolo di ¼ di quello precedente. Più precisamente, nell'avvicinarsi al punto critico il rapporto tende al valore $\frac{1}{4.669201\ldots\ldots}$. Vi è anche una semplice relazione numerica che governa la velocità di rimpicciolimento fra i denti della biforcazione. Feigenbaum scoprì che nell'avvicinarsi alla regione caotica, ogni spazio è circa 2/5 di quello precedente; più precisamente, il rapporto è $\frac{1}{2,5029\ldots.}$.

Il significato di questi numeri non risiede nel loro valore, quanto nel fatto che si incontrano in contesti completamente differenti ed evidentemente rappresentano una proprietà fondamentale dei sistemi caotici.

3.3 Sistemi dinamici

Il termine dinamico si riferisce a processi che producono cambiamenti, ossia evolvono nel corso del tempo. Cerchiamo quindi di capire come si può simulare, matematicamente, l'evoluzione temporale di un sistema reale. Lo stato di un sistema dinamico è rappresentato da un punto in uno spazio vettoriale e la variabile temporale, utilizzata per misurare il tempo che scorre, può essere pensata come un numero reale o come un numero naturale; nel primo caso si parla di sistemi dinamici continui, nel secondo di sistemi dinamici discreti . Quale delle due rappresentazioni sia più adatta a descrivere un sistema reale, dipende dalla situazione che si sta analizzando. Ad esempio, nella descrizione del moto di un pendolo o dello scorrere dei fluidi, si usano sistemi a tempo continuo mentre, nella descrizione dell'andamento temporale di una popolazione di insetti, caratterizzati da stagioni riproduttive, si usano sistemi a tempo discreto, con unità di tempo pari all'intervallo che intercorre fra due generazioni successive. Il sistema dinamico, quindi, è la legge che esprime la variazione nel tempo, mentre la sua soluzione è l'insieme delle orbite, in funzione delle condizioni iniziali. In un sistema dinamico a tempo continuo, un'orbita può essere pensata come una curva mentre, in un sistema dinamico a tempo discreto, è costituita da una successione di punti.

Un sistema dinamico continuo è descritto da una equazione differenziale vettoriale del primo ordine autonoma

$$\frac{d\vec{z}}{dt} = f(\vec{z}(t))$$

o non autonoma

$$\frac{d\vec{z}}{dt} = f(\vec{z}(t), t)$$

e da una condizione iniziale

$$\vec{z}(0) = \vec{z}_0 .$$

L' esempio più semplice possibile di sistema dinamico continuo si ottiene in dimensione n = 1 con un campo vettoriale che sia una funzione lineare omogenea

$$\frac{dx}{dt} = ax \qquad a \epsilon R$$

L'orbita $x(t)$ che passa per la condizione iniziale $x(0) = x_0$ si può esprimere mediante una funzione esponenziale:

$$x(t) = e^{at} x_0 ,$$

mentre, nel caso bidimensionale

$$\begin{cases} \frac{dx}{dt} = ax \\ \frac{dy}{dt} = by \end{cases}$$

le orbite si ottengono risolvendo separatamente le due equazioni per le variabili x ed y che sono disaccoppiate

$$\begin{cases} x(t) = e^{at} x_o \\ y(t) = e^{bt} y_0 \end{cases}$$

Tra i sistemi dinamici ci sono i sistemi hamiltoniani per i quali

$$\vec{z} = (\vec{q}, \vec{p}) = (q_1, ..., q_n, p_1, ... p_n)$$

ed inoltre

$$f(\vec{z}) = (-\frac{\partial H}{\partial q}, \frac{\partial H}{\partial p})$$

dove $H(\vec{p}, \vec{q})$ è la funzione hamiltoniana di modo che l'equazione differenziale è data dalle equazioni di Hamilton

$$\begin{cases} \frac{\partial \vec{q}}{\partial t} = -\frac{\partial H}{\partial \vec{p}} \\ \frac{\partial \vec{p}}{\partial t} = \frac{\partial H}{\partial \vec{q}} \end{cases}$$

Un esempio ben noto è quello di un oscillatore armonico la cui equazione del moto è un'equazione differenziale del secondo ordine:

$$m\frac{d^2x}{dt^2} + kx = 0 \quad ,$$

dove la variabile x rappresenta lo spostamento rispetto alla posizione di equilibrio della molla e kx è la forza elastica di richiamo.

La lagrangiana in questo caso è

$$L = \frac{1}{2}m\dot{x}^2 - \frac{1}{2}kx^2 \quad ,$$

poiché

$$p = \frac{\partial L}{\partial \dot{x}} = m\dot{x} \Rightarrow \dot{x} = \frac{p}{m}$$

la hamiltoniana è

$$H = p\dot{x} - (\frac{1}{2}m\dot{x}^2 - \frac{1}{2}kx^2) = \frac{1}{2}\frac{p^2}{m} + \frac{1}{2}kx^2 .$$

Anche in questo caso è possibile risalire all'espressione analitica della traiettoria: se assumiamo che il corpo parta da fermo, con uno spostamento iniziale A rispetto alla posizione di equilibrio, ossia prendendo come condizione iniziale

$$\begin{cases} x(0) = A \\ v(0) = 0 \end{cases}$$

allora la soluzione è:

$$x(t) = A\cos\omega_0 t ,$$

con

$$\omega_0 = \sqrt{\frac{k}{m}} .$$

Se consideriamo anche una resistenza passiva, l'equazione si complica un po' e diventa

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = 0 .$$

Nel caso di piccoli coefficienti di smorzamento b, la soluzione è la seguente

$$x = Ae^{-\frac{b}{2m}t}\cos(\omega t + \varphi)$$

con

$$\omega = \sqrt{\frac{k}{m} - (\frac{b}{2m})^2}\ .$$

Ovviamente, l'ampiezza A e l'angolo di fase φ sono determinati dalle condizioni iniziali. Questa soluzione è valida fino a che $b^2 < 4mk$ ed il punto oscilla intorno alla posizione di riposo, ma le oscillazioni massime si vanno attenuando con legge esponenziale. Se $b^2 > 4mk$, la soluzione è

$$x = e^{-\frac{b}{2m}t}(A_1 e^{\gamma t} + A_2 e^{-\gamma t}),$$

essendo

$$\gamma = \sqrt{(\frac{b}{2m})^2 - \frac{k}{m}}\ .$$

Nel caso limite b = 4mk la soluzione si può porre nella forma

$$x = e^{-\frac{b}{2m}t}(A_1 + A_2 t)\ .$$

Le ultime due soluzioni indicano un moto non periodico in cui il punto tende a tornare nella posizione di equilibrio più o meno rapidamente. La traiettoria del sistema è detta regolare se è stabile per variazioni infinitesime della condizione iniziale. Invece la traiettoria del sistema è detta caotica se è instabile per variazioni infinitesime della condizione iniziale.

In questi esempi, la facilità ad ottenere le soluzioni in forma analitica è legata al fatto che abbiamo ottenuto equazioni differenziali particolarmente semplici. Se si considerano invece equazioni differenziali non lineari, come spesso accade nella descrizione dei sistemi reali, trovare una soluzione in forma analitica è in genere molto difficile o addirittura impossibile. La quantità utile per caratterizzare il comportamento caotico è l'esponente di Lyapunov, che identifica l'instabilità esponenziale delle traiettorie a tempi lunghi, ed è dato da

$$\lambda = \lim_{t\to\infty} \frac{1}{t} \ln \left| \frac{\Delta\vec{z}(t)}{\Delta\vec{z}(0)} \right|,$$

dove $\Delta\vec{z}(t)$ è la differenza al tempo *t* tra due traiettorie inizialmente distanziate di una quantità $\Delta\vec{z}(0)$. Analoghe considerazioni valgono anche per modelli dinamici a tempo discreto. In questo caso, preso come unità di misura l'intervallo temporale scelto per scandire il tempo, la legge del moto viene rappresentata sotto forma di equazioni alle differenze

$$x(t+1) = f(x(t))$$

Partendo dalla condizione iniziale, l'intera traiettoria si può ottenere induttivamente: da x(0) si ottiene x(1), il quale può essere preso come nuovo argomento per ottenere x(2) e così via come nella equazione logistica vista nel paragrafo precedente. Possiamo, in definitiva, dare la seguente definizione :

La teoria dei sistemi dinamici è l'insieme dei metodi matematici attraverso i quali si cerca di ottenere, in maniera più o meno esplicita, informazioni sull'operatore di evoluzione temporale partendo da una sua rappresentazione locale (o legge del moto) assegnata in forma di equazioni differenziali o alle differenze.

3.4 Sistemi non lineari

Un sistema dinamico si dice non lineare se è della forma

$$\frac{d\vec{z}}{dt} = f(\vec{z}),$$

con $f(\vec{z})$ non lineare.

Ad esempio sia

$$\frac{dx}{dt} = 1 + x^2$$

allora

$$\frac{dx}{1+x^2} = dt \Rightarrow arctgx = t + c \Rightarrow x = tg(t + c)$$

Il sistema dinamico non ha perciò soluzioni definite per ogni $t \epsilon R$, ma ha soluzione per ogni condizione iniziale $x(t_0) = x_0$.

Le soluzioni ottenute nell'esempio qui sopra, malgrado si tratti di un esempio tra i più semplici possibili di sistema dinamico non lineare, non sono esplicite nello stesso senso delle soluzioni di un sistema dinamico lineare per i seguenti motivi:

1) Prima di tutto, occorre calcolare una primitiva: come è noto esistono funzioni con espressioni analitiche molto semplici, le cui primitive non hanno un'espressione analitica finita come composizione di funzioni trascendenti elementari.
2) In secondo luogo, occorre invertire la primitiva, e l'inversa di una funzione con espressione analitica semplice non è detto che sia esprimibile come composizione di funzioni trascendenti elementari.

Nella maggior parte dei casi di sistemi dinamici non lineari, si è costretti a limitarsi ad uno studio qualitativo delle soluzioni.

Studiando i sistemi non lineari si osservano fenomeni che non si verificano nei sistemi lineari. Come primo esempio consideriamo l'equazione di Duffing

$$m\frac{d^2x}{dt^2} + kx - \frac{1}{6}\mu kx^3 = \frac{d^2x}{dt^2} + \omega^2{}_0 x - \frac{1}{6}\mu\omega^2{}_0 x^3 = 0 \quad ;$$

supponiamo di avere una soluzione approssimata

$$x(t) = Asen\omega t \quad ,$$

sostituendola nella equazione con l'identità

$$x^3 = A^3 sen^3 \omega t = A^3 (\frac{3}{4} sen\omega t - \frac{1}{4} sen3\omega t)$$,

otteniamo

$$\frac{d^2 x}{dt^2} + \omega^2{}_0 (x - \frac{\mu x^3}{6}) = (\omega^2{}_0 - \omega^2 - \mu\omega^2{}_0 \frac{A^2}{8}) A sen\omega t + \mu\omega^2{}_0 \frac{A^3}{24} sen3\omega t$$,

che in generale non è uguale a zero. Nel caso in cui

$$\omega^2 = \omega^2{}_0 (1 - \mu \frac{A^2}{8})$$

e se A^3 è molto piccolo i due termini si annullano. Osserviamo quindi che la pulsazione dipende dall'ampiezza e questa è una caratteristica dei sistemi non lineari. Nei sistemi lineari, invece, sappiamo che la pulsazione dipende dalle proprietà del sistema.

Un'altra proprietà dei sistemi non lineari si può trovare analizzando l'equazione di Van der Pol

$$\frac{d^2 x}{dt^2} + 0{,}1(1 - x^2) \frac{dx}{dt} + x = 0$$,

infatti si vede che x diverge o converge a zero a seconda delle condizioni iniziali. Anche questa proprietà, cioè che la stabilità di un sistema dipende dalle condizioni iniziali, è una caratteristica dei sistemi non lineari. Modificando leggermente l'equazione di Van der Pol

$$\frac{d^2 x}{dt^2} - 0{,}1(1 - x^2) \frac{dx}{dt} + x = 0$$,

si scopre un ulteriore fenomeno caratteristico della non linearità e cioè che l'ampiezza di oscillazione è indipendente dalle condizioni iniziali.

3.5 Spazio delle Fasi

Lo spazio delle fasi è un mezzo per visualizzare con degli assi cartesiani il moto di un sistema nel tempo: le ascisse rappresentano la posizione e le ordinate la velocità. Ad esempio nel caso del pendolo semplice per piccoli angoli abbiamo che

$$\frac{d^2s}{dt^2} + \frac{g}{l}s = 0 \,,$$

dove s è l'ascissa curvilinea, g è l'accelerazione di gravità e l è la lunghezza del pendolo. Quindi è l'equazione di un moto armonico di periodo

$$T = \frac{2\pi}{\omega} = 2\pi\sqrt{\frac{l}{g}} \,.$$

L'unica differenza tra questo sistema dinamico e quello della molla elastica visto nel paragrafo precedente, è che la traiettoria in questo caso non è rettilinea. Nello spazio delle fasi il peso situato all'estremità descrive una curva chiusa. Se si introduce l'attrito l'equazione diventa

$$\frac{d^2s}{dt^2} + b\frac{dx}{dt} + \frac{g}{l}s = 0 \,,$$

ed il pendolo perderà più o meno rapidamente energia in funzione del valore di smorzamento b ed, alla fine, si fermerà nella posizione di equilibrio. In questo caso, nello spazio delle fasi otteniamo una spirale che converge verso un punto fisso noto come attrattore. Se aggiungiamo, poi, oltre all'attrito una qualunque forza esterna con frequenza diversa dalla frequenza naturale del pendolo otteniamo

$$\frac{d^2s}{dt^2} + b\frac{dx}{dt} + \frac{g}{l}s = F\cos\omega t \,.$$

La soluzione è la somma di due termini. Il primo è uguale alla soluzione di smorzamento e quindi diventa trascurabile dopo un certo tempo, il secondo invece rappresenta lo stato di regime del punto. In pratica, dopo un certo tempo il pendolo si lascia guidare dalla forza esterna. Nello spazio delle fasi il punto, dopo aver seguito alcune complesse contorsioni transitorie si avvolge avvicinandosi progressivamente alla curva chiusa, corrispondente alle oscillazioni forzate, e qui rimane continuando a percorrere questa traiettoria, fintanto, che la forza esterna permane. Questa curva è chiamata ciclo limite.

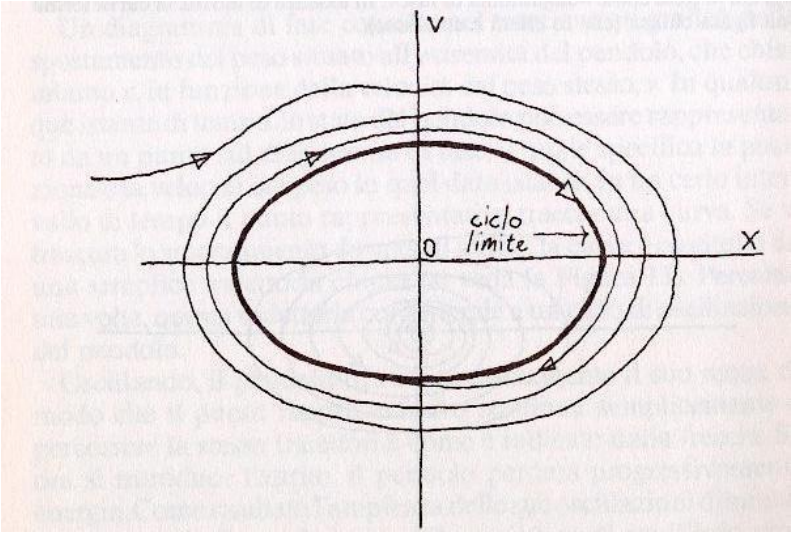


Figura 3.2: Ciclo limite, che caratterizza il moto del pendolo nel caso di forza lineare

Se rendiamo non lineare la forza di richiamo che agisce sul pendolo, ad esempio proporzionale a x^3, il comportamento del pendolo, con attrito modesto, è qualitativamente simile al caso precedente. Il punto parte da una determinata posizione nello spazio delle fasi esegue un qualche tipo di moto e poi si avvicina al ciclo limite. La differenza principale è che inclusi nella curva chiusa che rappresenta il ciclo limite vi sono ora alcuni cappi come si vede nella figura sottostante

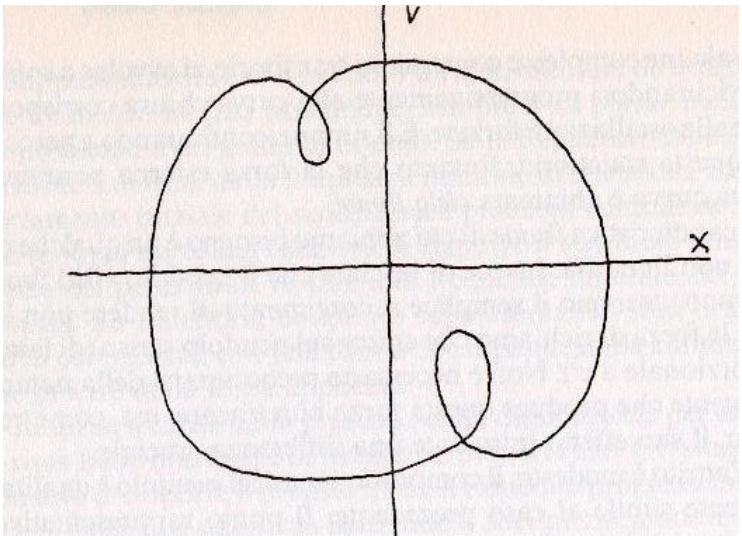


Figura 3.3: Ciclo limite, che caratterizza il moto del pendolo nel caso di forza non lineare

Fisicamente questo è dovuto al fatto che la forza forzante sovrasta temporaneamente la forza di richiamo, facendo in modo che il pendolo dia un piccolo balzo all'indietro ogni volta che si avvicina alla verticale. Se, poi, riduciamo progressivamente l'attrito, in corrispondenza di un valore critico del parametro di smorzamento, la curva del ciclo limite diventa quella della figura sottostante

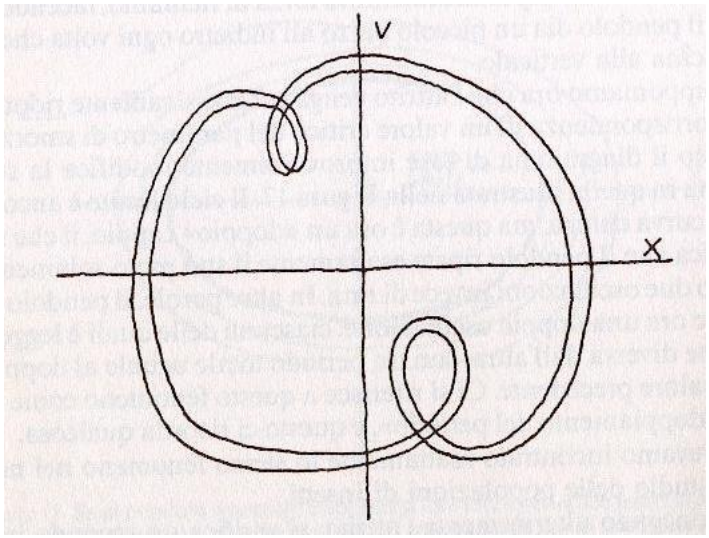


Figura3. 4: Ciclo limite, nel caso di riduzione progressiva dell'attrito

Il ciclo limite è ancora una curva chiusa, ma, ora, il pendolo ripete il suo moto dopo due oscillazioni invece di una con un periodo totale pari al doppio del valore precedente. Riducendo ulteriormente l'attrito, si verifica un secondo improvviso raddoppiamento di periodo, così che il pendolo riproduce esattamente il suo moto dopo quattro oscillazioni.

Continuando a ridurre l'attrito, si ottengono ulteriori raddoppiamenti di periodo come riportato nella figura seguente

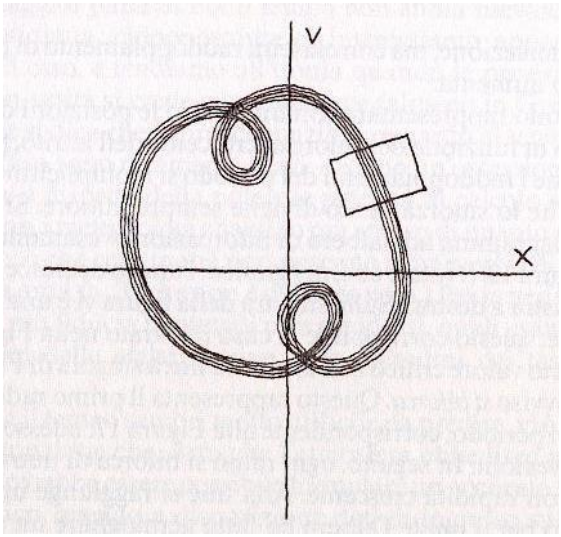


Figura 3.5: Ciclo limite nel caso di ulteriore riduzione dell'attrito

Il ciclo limite si suddivide in una curva multipla che sta ad indicare che il moto del pendolo non è più visivamente periodico. In pratica il moto sta progressivamente diventando caotico.

3.6 Mappa di Poincarè

Verso la fine del XIX secolo, Henry Poincaré, lavorando al problema della stabilità delle orbite di tre corpi celesti soggetti alla mutua attrazione gravitazionale, fu il primo a scoprire che i sistemi dinamici possono avere soluzioni estremamente complicate le quali, pur rimanendo confinate all'interno di un insieme limitato dello spazio delle fasi, non sono periodiche, né quasi periodiche. Uno degli strumenti matematici che egli dovette inventare per dimostrare l'esistenza di tali soluzioni è noto col nome di mappa di Poincarè. Questa tecnica consiste nel ridurre un sistema dinamico ad n dimensioni in una mappa ad n-1 dimensioni. Le mappe possono essere considerate come dei sistemi dinamici discreti. Si tratta di scegliere una oppurtuna ipersuperficie $\sum$ che intersechi le orbite ottenute risolvendo un certo sistema dinamico. La sequenza $\{x_1, x_2, \ldots . x_n\}$ dei punti di intersezione fra le orbite e la superficie definisce una mappa

$$P: \sum \rightarrow \sum$$

sulla superficie $\sum$. In particolare, se $\sum$ interseca un'orbita periodica del sistema dinamico, la mappa P avrà un punto fisso $x_F = P(x_F)$. Pertanto, il problema di studiare la stabilità di un'orbita periodica di un sistema dinamico continuo, si riconduce al problema di studiare la

stabilità di un punto fisso della sua mappa di Poincaré. Tornando all'esempio del pendolo caotico trattato nel paragrafo precedente, possiamo immaginare di guardare attraverso una piccola finestra nello spazio delle fasi e vedere passare il punto rappresentativo che lascia una traccia ad ogni passaggio. In pratica sezioniamo lo spazio delle fasi con una linea e vediamo i punti nei quali, ad ogni successivo passaggio, la traiettoria di fase la interseca.

Rappresentando con un grafico le posizioni delle intersezioni in funzione del valore decrescente dell'attrito, si ottiene un diagramma ad albero di biforcazione come nell'equazione logistica con la sola differenza che nel grafico l'attrito decresce andando da sinistra a destra. Anche se, ora stiamo trattando un sistema completamente differente, appaiono nuovamente i numeri di Feigenbaum 4,669201… e 2,5029…

Sembra che il caos abbia caratteristiche universali e, cioè, che vi sia un qualche ordine di base nel modo in cui il caos si manifesta.

3.7 Attrattori Frattali

Nel 1963, il meteorologo americano Edward Lorenz pubblicò un articolo dal titolo *Deterministic Nonperiodic Flow,* in cui analizzò un modello per la descrizione dei moti convettivi nell'atmosfera. Lo stato dell'atmosfera in questo modello può essere completamente descritto da tre variabili che evolvono in funzione del tempo

1) x(t) : flusso convettivo,
2) y(t) : distribuzione orizzontale della temperatura,
3) z(t) : distribuzione verticale della temperatura;

e da tre parametri

1) σ : rapporto tra viscosità e conduttività termica;
2) ρ : differenza di temperatura (in modulo) tra quella più alta e quella più bassa;
3) β : rapporto tra larghezza ed altezza della parte di cielo.

Questi dati servono per descrivere con apposite equazioni le appropriate leggi di fluido dinamica.

$$\begin{cases} \frac{dx}{dt} = -\sigma x + \sigma y \\ \frac{dy}{dt} = -\rho x - y - xz \\ \frac{dz}{dt} = -\beta z + xy \end{cases}$$

Se non fosse per i termini xy e yz, le equazioni del moto studiate da Lorenz sarebbero lineari e quindi si potrebbe ottenere la soluzione esatta in forma analitica. Lorenz, non si aspettava che la presenza di quelle piccole non linearità, avrebbe creato grandi problemi. Ma quando passò a calcolare gli andamenti delle x(t), y(t) e z(t), questi risultarono alquanto bizzarri e caratterizzati da oscillazioni molto irregolari. La sorpresa fu ancor più grande, quando si accorse che, partendo da condizioni iniziali che differivano in maniera quasi impercettibile, le corrispondenti traiettorie si allontanavano fra loro con rapidità esponenziale, per poi avvicinarsi di nuovo e poi riallontanarsi e così via. In altre parole, dopo un breve periodo iniziale in cui i comportamenti erano quasi uguali, quelli di lungo periodo risultavano completamente diversi. Rappresentando le traiettorie nello spazio delle fasi, Lorenz si rese anche conto che queste andavano a disporsi su una particolare figura che non mutava cambiando le condizioni iniziali. Si trattava di un *attrattore caotico* che venne chiamato a*ttrattore strano di Lorenz* . La sua forma ci dà informazioni di regolarità perché ci dice che, per quanto bizzarre, le traiettorie rimarranno intrappolate all'interno di quella figura. Dagli studi di Lorenz si cominciò a capire e poi si dimostrò rigorosamente che, quando un sistema dinamico è caratterizzato da un attrattore con dimensione frattale, il sistema è caotico. Questo è il legame fondamentale tra frattali e caos.

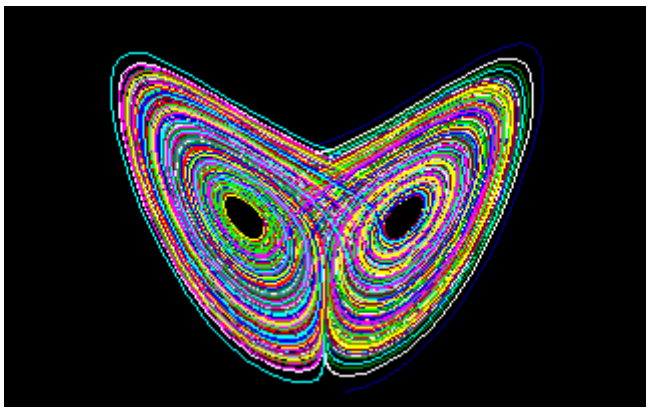


Figura 3.6: Attrattore di Lorenz

3.8 Frattali IFS

Si definisce frattale IFS (Iterated Function System) un frattale ottenuto iterando un insieme di trasformazioni affini.

Una trasformazione affine in un piano è un'applicazione biunivoca **T** che fa corrispondere al punto **P** di coordinate (x, y) il punto **P'** di coordinate (X, Y) secondo la relazione:

$$\begin{cases} X = ax + by + e \\ Y = cx + dy + f \end{cases}$$

dove i coefficienti *a*, *b*, *c*, *d*, *e*, *f* sono numeri reali.

Usando le notazioni dell'algebra lineare si può scrivere l'applicazione **T** sotto forma di prodotto fra matrici:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

si parlerà di Trasformazioni affini, quando:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} \neq 0$$

I frattali che si considereranno sono: la polvere di Cantor, il triangolo di Sierpinski, la curva di Peano ed il frattale di Julia.

3.9 Polvere di Cantor

La polvere di Cantor è uno dei più famosi sottoinsiemi dell'insieme dei numeri reali.

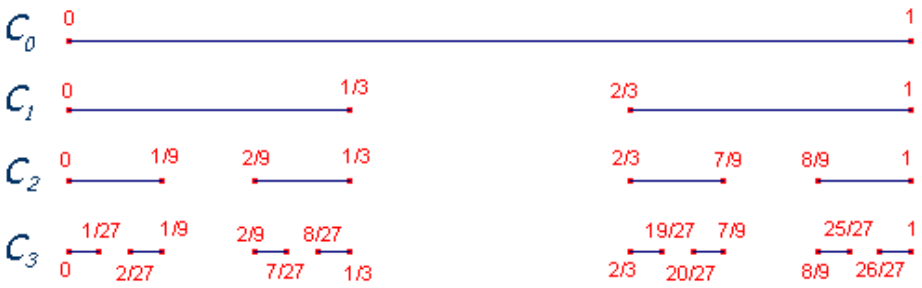
Si consideri l'intervallo [0,1] dei reali, intervallo che indicheremo con C_0.

Tale intervallo viene diviso in tre parti uguali mediante i punti 1/3 e 2/3. Eliminando la parte centrale si ottiene l'insieme C_1= [0,1/3]∪[2/3,1]. Ripetendo il procedimento su ciascuna delle due parti di cui è costituito C_1, si ottiene l'insieme C_2=[0,1/9]∪[2/9,1/3]∪[2/3,7/9]∪[8/9,1]. In modo analogo si hanno gli insiemi C_3, C_4, e così via.

L'insieme di Cantor, o Polvere di Cantor, è l'insieme:

$$C = \bigcap_{n=0}^{\infty} C_n$$

Una rappresentazione grafica, limitata ai primi passi, di questo procedimento può essere la seguente:



Si noti che, per ogni *n*, le due metà di C_{n+1}, sono una copia esatta, ridotta di 1/3, di C_n. Questo significa che, guardare C_{n+1} con una lente che ingrandisca tre volte è lo stesso che guardare C_n ad occhio nudo.

Le trasformazioni affini che generano tale frattale sono:

$$W_1 = \begin{cases} X = \frac{1}{3}x \\ Y = y = 0 \end{cases}$$

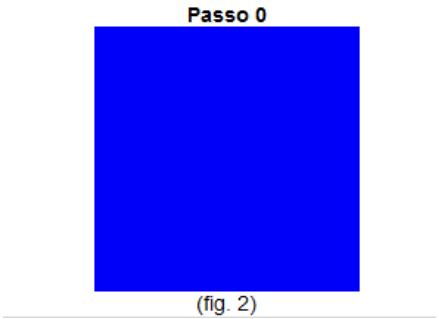
$$W_2 = \begin{cases} X = \frac{1}{3}x + \frac{2}{3} \\ Y = y = 0 \end{cases}$$

$$W_1 \cup W_2 = C_1$$

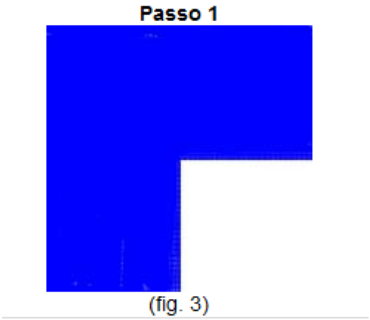
3.10 Triangolo di Sierpinski

Il cosiddetto triangolo di Sierpinski (o Gerla di Sierpinski) tratta di un frattale molto semplice da ottenere. Da un punto di vista strettamente geometrico viene generato con una serie di rimozioni. Si inizia con un quadrato pieno da cui si rimuove un quadrato di lato pari alla metà del quadrato iniziale, formata da tre quadrati. Da ciascuno di questi quadrati si elimina nuovamente il quadrato in basso a destra e si ottiene una figura formata da nove quadrati. Iterativamente si continua fino ad arrivare al risultato finale.

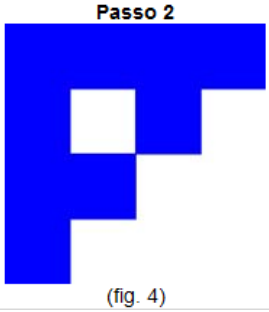
Nelle figure seguenti possiamo osservare i primi 6 passi necessari per ottenere il frattale.



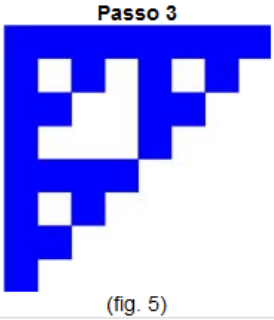
(fig. 2)



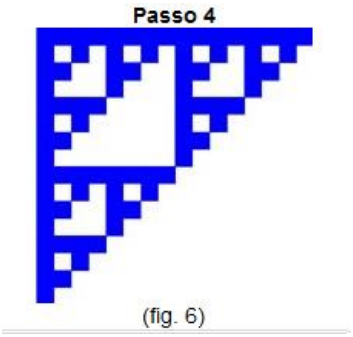
(fig. 3)



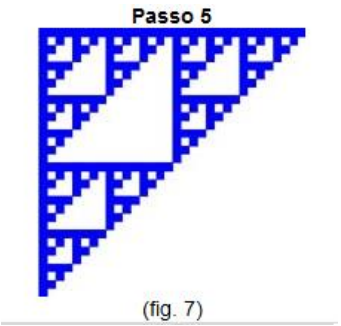
(fig. 4)



(fig. 5)



(fig. 6)



(fig. 7)

Per ottenere tale frattale, basta usare le seguenti tre trasformazioni (T, V, W)

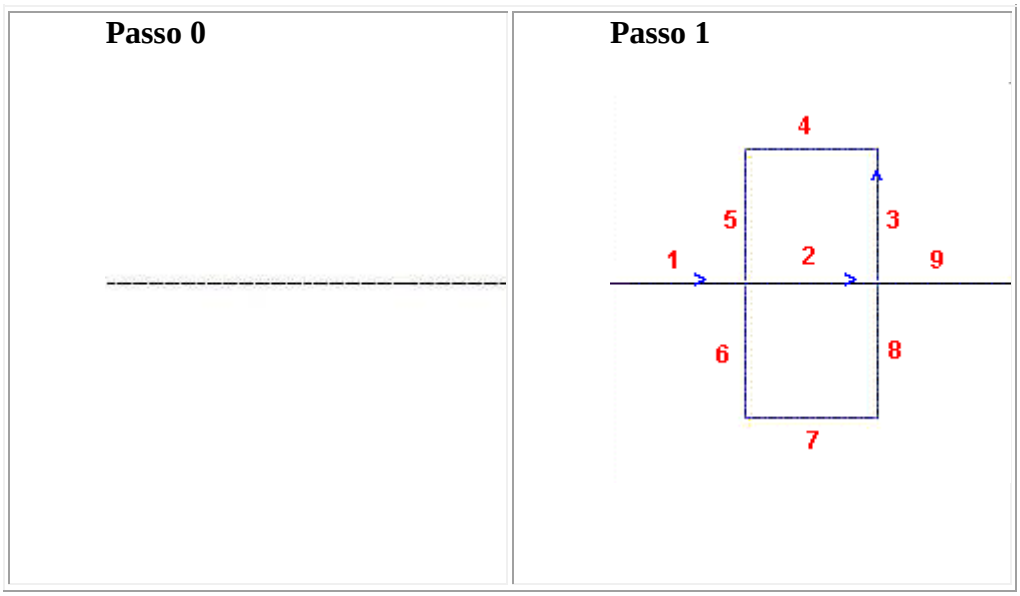
T: $\begin{cases} X = \frac{1}{2}x \\ Y = \frac{1}{2}y \end{cases}$

V: $\begin{cases} X = \frac{1}{2}x \\ Y = \frac{1}{2}y + \frac{1}{2} \end{cases}$

W: $\begin{cases} X = \frac{1}{2}x + \frac{1}{2} \\ Y = \frac{1}{2}y + \frac{1}{2} \end{cases}$

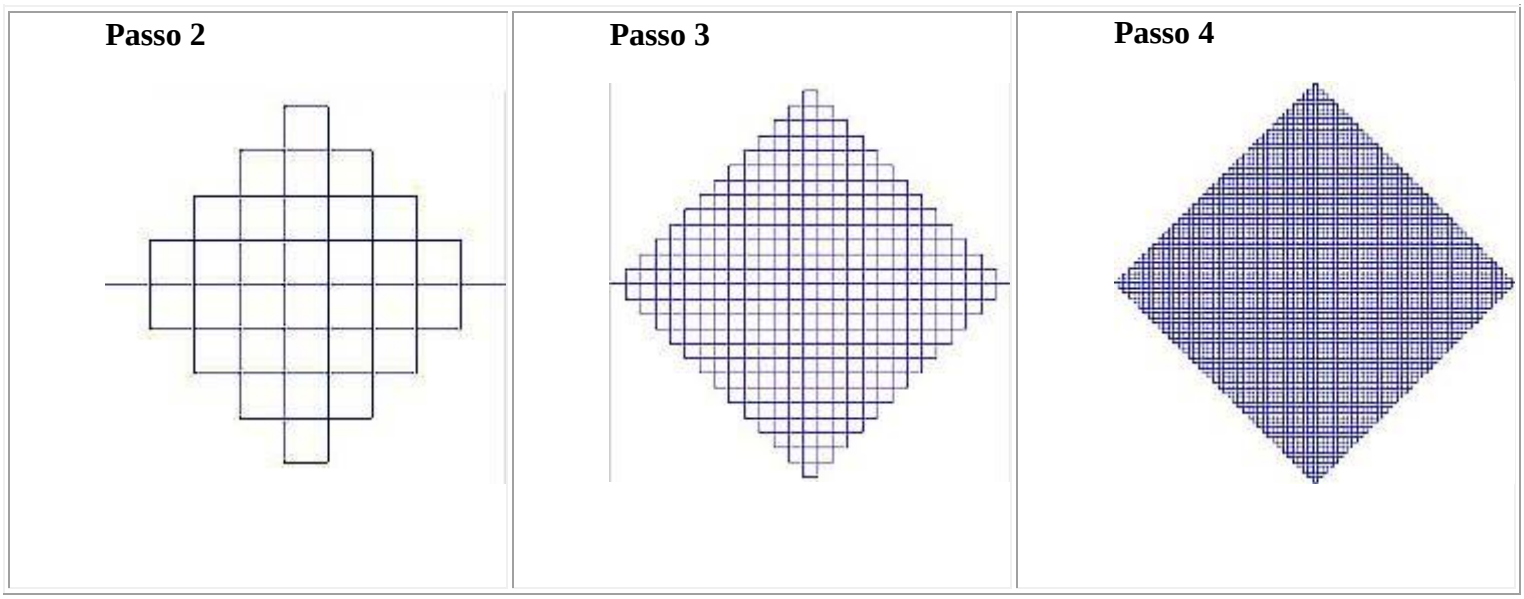
3.11 Curva di Peano

La curva di Peano è una curva che ha la proprietà di riempire tutto un quadrato, cioè la curva di Peano passa per tutti i punti di un quadrato. La figura di partenza è un segmento di lunghezza unitaria (Passo 0). Ad esso vengono applicate nove trasformazioni affini che permettono di ottenere la figura seguente (Passo 1).



Al Passo1 sono evidenziate le nove trasformazioni ed il verso di percorrenza della curva.

Continuando il processo di iterazione la curva riempie tutto il quadrato. La costruzione della curva qui proposta ha come elemento di partenza la diagonale del quadrato.



Il risultato finale (Passo4) è un quadrato frattale.

Questa curva può essere costruita tramite le trasformazioni affini della forma

$$\begin{cases} X = ax + by + e \\ Y = cx + dy + f \end{cases}$$

in cui i coefficienti delle trasformazioni sono i seguenti

Trasformazione	A	b	C	d	E	F
1	1/3	0	0	1/3	0	0
2	1/3	0	0	1/3	1/3	0
3	0	1/3	1/3	0	2/3	0
4	1/3	0	0	1/3	1/3	1/3
5	0	1/3	1/3	0	1/3	0
6	0	1/3	1/3	0	1/3	-1/3
7	1/3	0	0	1/3	1/3	-1/3
8	0	1/3	1/3	0	2/3	-1/3
9	1/3	0	0	1/3	2/3	0

3.12 Insiemi di Julia

Per definire il frattale di Julia si considera un punto $Z_0 = x_0 + iy_0$ nel piano complesso e si applica successivamente la seguente iterazione

$$Z_n = Z^2{}_{n-1} + x_0 + iy_0$$

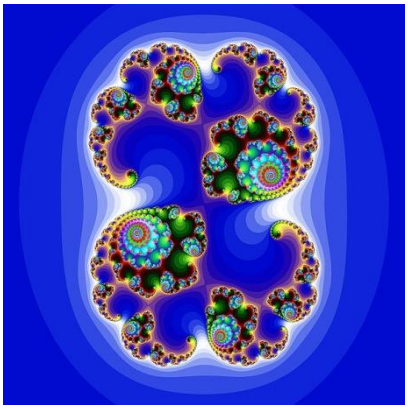
cioè

$$Z_1 = (x_0 + iy_0)^2 + x_0 + iy_0$$
$$Z_2 = Z^2{}_1 + x_0 + iy_0$$
.
.
.

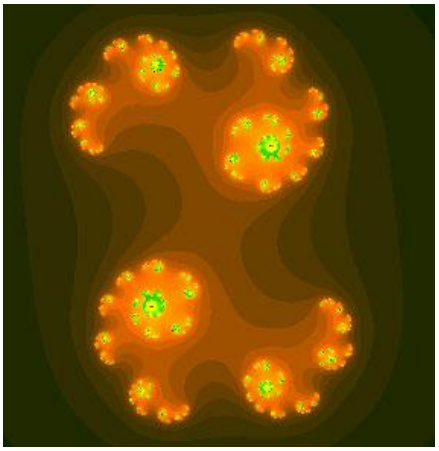
Per la maggior parte delle scelte di x_0 e y_0 la procedura di iterazione fa tendere il punto all'infinito. Esistono, tuttavia, dei valori per cui questo non accade e sono questi punti che costituiscono l'insieme di Mandelbrot.

È importante sapere che vi sono infiniti insiemi di Julia poiché la scelta di x_0 e y_0 non deve sottostare a nessuna restrizione.

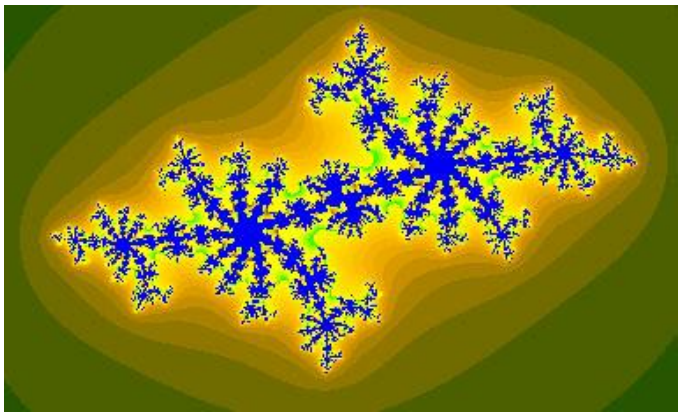
Vengono presentati differenti Frattali di Julia al variare di x_0+iy_0.



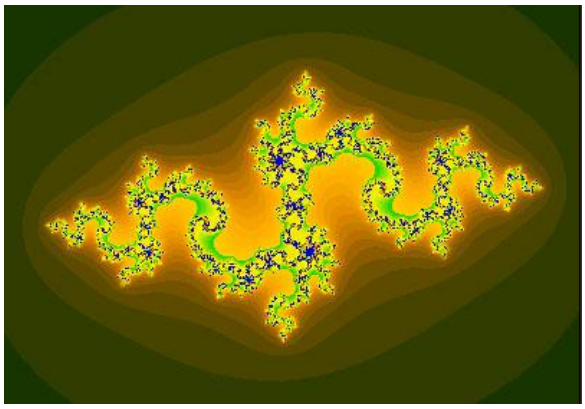
Per x_0+iy_0 = 0,285+ i 0,013



Per x_0+iy_0 = 0,45 - i 0,1428



Per x_0+iy_0 = - 0,70176 - i 0,3842



Per x_0+iy_0 = 0,835 - i 0,2321

3.13 Geometria frattale

E' noto che l'idea che permise di unificare i vari tipi di geometria fu proposta da Klein nel 1872 in un discorso tenuto all'Università di Erlangen. In tale discorso, passato alla storia con il nome di "**Programma di Erlangen"**, il matematico tedesco mostrò come il concetto algebrico di gruppo permette di avere una visione unitaria di tutte le geometrie; infatti egli descrisse la geometria come lo studio delle proprietà che sono invarianti rispetto ad un particolare gruppo di trasformazioni. Ad esempio la geometria euclidea è lo studio delle proprietà invarianti rispetto al gruppo delle similitudini, la geometria affine è lo studio delle proprietà invarianti rispetto al gruppo delle affinità, la geometria proiettiva è lo studio delle proprietà invarianti rispetto al gruppo delle proiettività e così via. Il gruppo di trasformazioni che caratterizza la geometria frattale è costituito dalle mappe bilipschitziane. Ricordiamo a questo punto che una applicazione fra due spazi metrici

$$f : (S, d) \to (S', d')$$

si dice Lipschitziana se esiste una costante $M \geq 0$, detta coefficiente di Lipschitz, tale che $\forall x, y \epsilon S$ risulta

$$d'(f(x), f(y)) \leq Md(x, y)$$

Tali applicazioni sono a distorsione limitata nel senso che la distanza tra le immagini può essere maggiorata con il prodotto di una costante per la distanza tra i punti. Se la funzione è invertibile, si parla di mappa bilipschitziana e tali mappe, che hanno la struttura algebrica di gruppo, caratterizzano la geometria frattale nel senso specificato dal Programma di Erlangen. Le affinità sono ad esempio mappe bilipschitziane. Se $0 \leq M < 1$, l'applicazione lipschitziana si chiama contrazione e per le contrazioni di uno spazio metrico completo in sé è fondamentale il seguente risultato:

TEOREMA DI BANACH : " *In uno spazio metrico completo una contrazione ha un unico punto fisso. Esiste cioè un unico punto x tale che f(x) = x.*

Le mappe Lipschitziane ammettono anche un sovragruppo: le mappe Holderiane.

$$f \quad Holderiana \Leftrightarrow \exists\, c > 0, \exists\, \alpha > 0 : d(f(x), f(y)) \leq cd(x, y)^{\alpha}$$

Se $\alpha = 1 \rightarrow$ f lipschitziana;

se $\alpha = 1$, $c < 1 \rightarrow$ f contrazione.

Se abbiamo una qualsiasi funzione *f*; l'orbita di un punto *x* è data da

$$x, f(x), f^2(x), ..., f^n(x)$$

Supponiamo ora che l'orbita di *x* converga verso un punto y, cioè

$$x, f(x), f^2(x), ..., f^n(x) \rightarrow y$$

Applicando a tale successione l'applicazione *f*, otteniamo

$$f(x), f^2(x),...,f^n(x), f^{n+1}(x)$$

Se la funzione è continua, la successione delle immagini, tramite *f*, convergerà ad *f(y)*

$$f(x), f^2(x) ,..., f^n(x), f^{n+1}(x) \rightarrow f(y)$$

Osserviamo che le due successioni, a meno del primo termine, sono uguali e dunque, se convergono, devono necessariamente convergere ad uno stesso punto[1].

Quindi, quando l'orbita di un punto è convergente, allora il punto verso cui converge è un punto fisso e tale punto è unico. Sia (S,d) uno spazio metrico e consideriamo l'insieme

$$K(S) = \{K \subseteq S : K \text{ compatto}, K \neq 0\}$$

Tale insieme, i cui elementi sono i sottoinsiemi compatti e non vuoti dello spazio metrico, in topologia è noto come iperspazio. Ricordiamo che in uno spazio metrico si definiscono la distanza tra un punto $p \epsilon S$ ed un sotto insieme non vuoto A e la distanza tra due insiemi non vuoti A e B nel seguente modo

$$d(p,A) = \inf\{d(p,a) : a \in A\}$$

$$d(A,B) = \inf\{d(a,b) : a \in A, b \in B\}$$

Il problema che si pose Hausdorff fu di capire se si può metrizzare l'iperspazio. La distanza tra insiemi, definita in (S,d), non verifica la disuguaglianza triangolare e quindi non è una metrica; infatti se

$$A \cap C \neq \phi \quad C \cap B \neq \phi$$

ne segue che

[1] Ricordiamo infatti che essendo in uno spazio metrico, sussiste l'unicità del limite.

$$d(A,C) = d(C,B) = 0$$

e di conseguenza non è vero che

$$d(A,B) \leq d(A,C) + d(C,B)$$

Se invece poniamo

$$d_H(A,B) = \max\left\{\sup_{a\in A} d(a,B), \sup_{b\in B} d(b,A)\right\}$$

$(K(S), d_H)$ diviene uno spazio metrico e d_H è chiamata distanza di Hausdorff.

La distanza di Hausdorff è di particolare importanza perchè soddisfa delle proprietà dello spazio iniziale. Una di queste proprietà è la completezza; si dimostra infatti che la metrica di Hausdorff è completa se e solo se lo spazio di base è completo.

Se abbiamo una funzione continua $f: S \to S$ con S immerso nell'iperspazio K(S), su cui è definita la metrica di Hausdorff, la funzione f si può sollevare[2] in una funzione f_k continua rispetto alla metrica di Hausdorff. Riportiamo il seguente schema:

$$\begin{array}{ccccc} f_k: & (K(S), d_H) & \to & (K(S), d_H) \\ & \uparrow & & \uparrow \\ f: & (S, d_H) & \to & (S, d_H) \end{array}$$

Se in particolare f è una contrazione, che è una applicazione continua, non solo possiamo parlare di sollevamento ma possiamo vedere che la funzione f_k che solleva la contrazione è anch'essa una contrazione con lo stesso coefficiente di f. Dunque si verifica che

[2] La funzione f_k solleva la *f* nel senso che ad un punto *x*, f_k associa *f(x)*: f_k è un'estensione di f.

$$d_H(f_k(A), f_k(B) \leq c d_H(A,B) \qquad \forall\ A, B \in K(S)$$

Procediamo ora ad una generalizzazione dei concetti esposti analizzando meglio anche il concetto di dimensione di similarità. Supponiamo di avere un sistema finito di funzioni continue, ad esempio contrazioni $\{w_1, w_2,...,w_n\}$ di uno spazio metrico (X,d) in sé.

L'insieme formato dallo spazio metrico e dal sistema di contrazioni introdotto è detto **sistema di funzioni iterate** o più brevemente, come già detto, I.F.S. (Itereted Function Sistem):

$$(X, d, w_1, w_2, ..., w_n) = \text{I.F.S.}$$

Come visto, ogni contrazione si può sollevare, sull'iperspazio dei compatti, in una contrazione che mantenga inalterato il coefficiente. Siccome i compatti sono in numero finito e la loro unione è un compatto, solleviamo le *n* contrazioni e ne facciamo la somma. Consideriamo l'applicazione

$$w : A \in K(X) \rightarrow w_1(A) \cup ....\cup w_n(A) \in K(X).$$

Questa applicazione ad un compatto associa un compatto, è una contrazione rispetto alla metrica di Hausdorff ed ammette inoltre come coefficiente di contrazione il massimo tra tutti i coefficienti contrattivi

$$c = \max \{c_1,...,c_n\}$$

Allora consideriamo uno spazio metrico completo (X,d), un sistema di funzioni iterate e la mappa unione dei sollevamenti. Se l'I.F.S. è costituito da contrazioni, anche la mappa unione è una contrazione; inoltre poichè lo spazio di base è completo come spazio metrico, anche l'iperspazio dei compatti è completo ed il teorema di Banach ci garantisce che la mappa unione ha un unico punto fisso che si ottiene muovendosi lungo una qualsiasi orbita. Indicando con A tale attrattore, otteniamo che

$$A = w(A)$$

Ma, per come è definito, sappiamo che

$$A = w_1(A) \ldots \cup w_n(A)$$

pertanto

$$A = w(A) = w_1(A) \cup \ldots \cup w_n(A);$$

L'attrattore si esprime come unione di sue copie contratte ed iterando tale processo otteniamo:

$A = w(A) = w(w_1(A) \cup \ldots \cup w_n(A)) =$

$w_1 \circ w_1(A) \cup w_1 \circ w_2(A) \cup \ldots \cup w_n \circ w_n(A) = \bigcup_{i,j=1,\ldots n} w_i \circ w_j(A)$

Quindi A si esprime come unione finita di sue copie contratte di un certo coefficiente. Infatti, considerando $\bigcup_{i,j=1,\ldots n} w_i \circ w_j(A)$ si osserva che w_j contrae di un certo c_j, la composta w_i o w_j contrae di c_i x c_j, ma essendo c_i e c_j entrambi minori di 1, accade che c_i x c_j è minore di ognuno dei due fattori; quindi la copia è anch'essa contratta.

Iterando tale procedimento, al passo h si ha:

$$\bigcup_{i_1,\ldots i_h=1,\ldots,n} w_{i_1} \circ \cdots \circ w_{i_h}(A) = A$$

Quindi A, al passo h, si esprime come unione finita di sue copie contratte, di coefficiente $C_{i_1} \times \ldots \times C_{i_h}$.

Ora attribuiamo all'attrattore una dimensione, che viene detta **dimensione di similarità**. Prima di introdurre una tale dimensione, facciamo la seguente osservazione:

definiamo un segmento e consideriamo il suo trasformato tramite una similitudine generica, tale trasformato avrà misura pari a ρ (ρ è il coefficiente che individua la similitudini) moltiplicato per la misura del segmento di partenza; se invece consideriamo un quadrato con

una certa area, il suo trasformato avrà come area, l'area del quadrato precedente moltiplicato per ρ^2, e così via.

Sappiamo che A si scrive come somma di sue copie contratte, ovvero

$A = w_1(A) \cup ... \cup w_n(A);$

supponiamo che le copie non si intersechino. e che μ(A) sia la misura di A. Allora

$$\mu(A) = \mu(w_1(A) \cup ... \cup w_n(A)).$$

Ogni w_i(A) è contratta rispetto ad A di c_i, dunque, per l'osservazione precedente,

$$\mu(A) = c_1^s \mu(A) + ... + c_n^s \mu(A),$$

dove *s* è l'eventuale dimensione che non conosciamo ancora.

Essendo μ(A) ≠ 0 possiamo dividere per tale quantità primo e secondo membro; otteniamo

$$c_1^s + ... + c_n^s = 1.$$

Dunque, fissati $c_1,...,c_n$ tali che $0 < c_1 < 1$, $\forall\, i$, dobbiamo vedere se esiste qualche *s* positivo che verifichi la condizione

$$c_1^s + ... + c_n^s = 1.$$

In realtà si tratta di studiare gli zeri della funzione continua $\sum_{i=1}^{n} c_i^s = 1$

Questa funzione ammette una ed una sola soluzione che, in un caso così generale, è difficile da calcolare. Per questo motivo riduciamoci ai sistemi di funzioni iterati costituiti da similitudini contrattive aventi tutte le stesso coefficiente di contrazione c.

Questo significa che se considero l'attrattore di questo sistema di funzioni iterate abbiamo:

$$A = w(A) = w_1(A) \cup \ldots \cup w_n(A),$$

cioè A si esprime come unione finita di sue copie conformi contratte di c. Dunque l'insieme A è un insieme **autosimilare**. In più A è un insieme autosimilare fine, in quanto è possibile ripetere indefinitivamente un tale ragionamento.

Dunque, al passo 2, A avrà questa forma

$$A = \cup_{i,j=1,\ldots n} w_i \circ w_j(A)$$

ovvero A si presenta come una unione finita di sue copie conformi ciascuna contratta di c^2.

Al passo h,

$$A = \bigcup_{i_1,\ldots i_h=1,\ldots,n} w_{i_1} \circ \cdots \circ w_{i_h}(A),$$

cioè si presenterà come unione finita di sue copie conformi, ciascuna ridotta di c^h.

Quindi questo insieme è autosimilare perchè si può scrivere come unione finita di sue copie conformi, ed è autosimilare fine perchè è unione finita di sue copie conformi di grandezza piccola quanto si vuole.

Cerchiamo di calcolare, per tale sistema, l'espressione di s.

Poichè abbiamo definito un I.F.S., in cui tutte le similitudini hanno lo stesso coefficiente di contrazione, l'equazione $\sum_{i=1}^{n} c_i^s - 1$ diventa

$$nc^s = 1$$

$$\Rightarrow c^s = \frac{1}{n} \Rightarrow s = \log_c \frac{1}{n} \Rightarrow s = \frac{\log n}{\log \frac{1}{c}}$$

dove *n* è la lunghezza del sistema di funzioni iterate, cioè il numero delle similitudini contrattive, e 1/c è l'inverso del coefficiente di contrazione.

In definitiva diremo che:

s rappresenta la dimensione di similarità, e se s è non intera si dice che l'attrattore è di natura frattale. Anche se la dimensione di similarità e quella del box-counting sono di facile determinazione, il concetto di dimensione che sembra più adatto a descrivere la geometria frattale è quello proposto per la prima volta da Hausdorff nel 1919. Cominciamo considerando un segmento $\overline{AB}$ di lunghezza L e ricopriamolo con dei regoli di lunghezza s ottenendo ovviamente

$$L = \sum s_i = Ns$$

Noi possiamo variare a piacimento la lunghezza dei regoli arrivando anche al seguente caso limite

$$L = \lim_{s \to 0} \sum s_i$$

Se invece di un segmento vogliamo misurare una linea curva, allora la prima somma rappresenterebbe solo la lunghezza approssimata e per avere la lunghezza esatta dobbiamo necessariamente considerare la formula limite. Se immaginiamo invece di voler calcolare l'area di una linea curva qualsiasi, consideriamo quadrati di lato s ed arriviamo a scrivere

$$A = \lim_{s \to 0} \sum s^2{}_i = 0$$

Infine per il volume, usando cubi di lato s, analogamente scriviamo

$$V = \lim_{s \to 0} \sum s^3{}_i = 0$$

Possiamo compattare le tre relazioni precedenti indicando con M_1, M_2, M_3 le misure monodimensionali, bidimensionali e tridimensionali rispettivamente

$$M_D = \lim_{s\to 0} \sum s^D{}_i \quad , \quad D = 1,2,3$$

Si può facilmente verificare che il limite per $0 < D < 1$ tende all'infinito mentre per $D > 1$ tende a zero e possiamo dunque interpretare $D = 1$ come un ***valore critico*** al disotto del quale il limite diverge, mentre al disopra il limite diventa zero. Non è difficile provare che se consideriamo una superficie piana otteniamo la stessa relazione ma con valore critico $D = 2$, mentre per una figura spaziale il valore critico è D = 3. Se applichiamo queste considerazioni alla polvere di Cantor ed al merletto di Koch otteniamo rispettivamente $M_1 = 0$ ed $M_1 = +\infty$; infatti l'insieme di Cantor ha lunghezza nulla mentre il merletto di Koch infinita. A questo punto generalizziamo nel seguente modo:

dato un insieme A di uno spazio metrico separabile, come ad esempio R^n, si considera un ricoprimento numerabile di A con aperti A_i

$$\{A_i\}_{i\in N}, \quad \bigcup_{i\in N} A_i \supseteq A.$$

di diametro non superiore ad un determinato valore ε, si fissa un numero positivo D e si calcola la somma dei diametri di tutti questi insiemi, elevati al valore scelto D.

$$\sum \delta(A_i)^D$$

Fissato ε ci sono infinite somme possibili, a seconda del diametro effettivo che si prende e a seconda di come si dispongono gli insiemi A_i. L'estremo inferiore di tutte queste possibili somme si chiama misura D-dimensionale ε-approssimata di Hausdorff dell'insieme A

$$\inf\left\{\sum \delta(A_i)^D : \{A_i\}_{i\in N}, \delta(A_i) \le \varepsilon, A \subseteq \bigcup A_i\right\} = \mu^D{}_\varepsilon(A)$$

Il limite di questa misura approssimata, al tendere a zero di ε

$$\lim_{\varepsilon \to 0} \mu^{D}{}_{\varepsilon}(A)$$

è detto misura D-dimensionale di Hausdorff dell'insieme *A*.

Consideriamo il sistema di misure p-dimensionali, due numeri reali p e q , con p<q, un insieme A e un ricoprimento di aperti A_i tali che $\delta(A_i) \leq \varepsilon$. Allora possiamo scrivere:

$$\delta(A_i)^q = \delta(A_i)^{q-p}\, \delta(A_i)^p .$$

Poichè $\delta(A_i) \leq \varepsilon$, vale la seguente disuguaglianza

$$\delta(A_i)^q \leq \varepsilon^{q-p} \delta(A_i)^p$$

passando alle somme otteniamo

$$\sum_{i \in N} \delta(A_i)^q \leq \varepsilon^{q-p} \sum_{i \in N} \delta(A_i)^p .$$

Questa disuguaglianza vale qualunque sia il ricoprimento di aperti, purché $\delta(A_i) \leq \varepsilon$, $\forall\, i$ e questo ci suggerisce che la disuguaglianza può essere trasferita agli estremi inferiori. Ma per definizione si ottiene

$$\mu_{\varepsilon}^{q}(A) \leq \varepsilon^{q-p} \mu_{\varepsilon}^{p}(A) .$$

Al tendere di ε a zero

$\mu_{\varepsilon}^{q}(A) \rightarrow \mu^{q}(A)$ (misura q-dimensionale)

ed analogamente

$\mu_{\varepsilon}^{p}(A) \rightarrow \mu^{p}(A)$ (misura p-dimensionale).

Se $\mu^{p}(A) = l \in R$, dove *l* è un numero reale finito, dalla disuguaglianza si ottiene:

$$\mu_\varepsilon^q(A) \le \varepsilon^{q-p}\mu_\varepsilon^p(A)$$
$$\downarrow \quad \downarrow$$
$$0 \quad l \in R$$

Di conseguenza, per ε→0, anche $\mu^q(A) \to 0$, essendo una quantità positiva; ossia: $\mu^q(A) = 0$ perché minore o al massimo uguale ad una quantità che tende a zero.

Quindi se p<q

$$\mu^p(A) \in R \Rightarrow \mu^q(A) = 0$$

Cioè se una misura è preceduta da una misura che è reale e finita, questa misura deve essere necessariamente nulla. Supponiamo ora che sia $\mu^q(A) \in \mathrm{R}\backslash\{0\}$ ed inoltre supponiamo che

$$\lim_{\varepsilon\to 0} \mu_\varepsilon^p(A) = \mu^p(A) = l \neq 0$$

poiché la disuguaglianza $\mu_\varepsilon^q \le \varepsilon^{q-p}\mu_\varepsilon^p(A)$, deve permanere passando al limite per ε→0, allora

$$\varepsilon^{q-p}\mu_\varepsilon^p(A) \to 0,$$

ossia $\mu^q(A) = 0$.

Ma questo è assurdo avendo supposto che $\mu_\varepsilon^q(A) \neq 0$.

In definitiva nel caso in cui $\mu^q(A) \in \mathrm{R}\backslash\{0\}$, $\mu^p(A)$ non può essere una costante, ma dovrà essere necessariamente +∞.

Quindi per ogni insieme *A* esiste un unico valore, che indichiamo con $dim_H(A)$, tale che per $p < dim_H(A)$ tale misura è infinita, mentre per $p > dim_H(A)$ tale misura è 0. Per $p = dim_H(A)$ tale misura può essere finita oppure no e può coincidere oppure no con la misura (lunghezza, area, volume) tradizionale dell'insieme. Il numero $dim_H(A)$ è noto come ***dimensione di Hausdorff*** dell'insieme *A* e si può vedere, anche se il calcolo esplicito non è banale, che le dimensioni di Hausdorff dei frattali più noti coincidono con le dimensioni per autosimilarità. Ad esempio:

$$\dim_H(Cantor) = \frac{\log 2}{\log 3}$$

e

$$\dim_H(merletto) = \frac{\log 4}{\log 3}$$

Concludiamo osservando che questa definizione di dimensione non ha sempre lo stesso valore della dimensione per autosimilarità. Indicando con $\dim_T$ la dimensione topologica e con $\dim_S$ la dimensione per autosimilarità, vale la seguente fondamentale relazione

$$\dim_T \leq \dim_{HB} \leq \dim_s$$

E' da segnalare però il seguente importante risultato:

Teorema: Condizione necessaria e sufficiente affinchè la $\dim_H$ coincida con la dimensione di similarità di A è che la misura relativa alla $\dim_H$ A delle intersezioni delle copie di primo livello dell'attrattore sia uguale a zero, cioè

$$\dim_H(A) = s(A) \Leftrightarrow \mu^{\dim_H(A)}\left(w_i(A) \cap w_j(A)\right) = 0 \quad \forall i \neq j .$$

La condizione

$$\mu^{\dim_H(A)}\left(w_i(A) \cap w_j(A)\right) = 0 \quad \forall i \neq j$$

ci dice che le immagini di primo livello devono intersecarsi "poco": questa è quella che si chiama *just touching*.

Si dice che un sistema di funzioni iterate è *just touching* quando verifica l'open set condition, cioè quando si può trovare un aperto che contiene le sue copie di primo livello a due a due disgiunte.

Quindi condizione necessaria e sufficiente affinché la dimensione di Hausdorff e quella di similarità coincidano è che sia verificata la open set condition.

Sia A un insieme ed $\{A_i\}$ un suo ricoprimento tale che

$$\delta(A_i) \leq \varepsilon \quad \forall i \in N$$

Sia f una mappa Holderiana ed $f(A \cap A_i)$ un ricoprimento di $f(A)$. Osserviamo che

$$A \cap A_i \subseteq A_i$$

e pertanto

$$\delta(A \cap A_i) \leq \delta(A_i) \leq \varepsilon$$

Presi $x, y \epsilon A \cap A_i$ si ha ovviamente che

$$d(x, y) \leq \delta(A \cap A_i) \leq \varepsilon$$

Applicando la funzione, che è Holderiana, otteniamo

$$d(f(x), f(y)) \leq c d(x, y)^a \leq c\varepsilon^a$$

Siccome questo vale per ogni coppia di punti di A, allora vale anche per l'estremo superiore e quindi vale per il diametro di $f(A \cap A_i)$. Cioè

$$\delta(f(A \cap A_i)) \leq c\delta(A \cap A_i)^\alpha \leq c\varepsilon^\alpha$$

Sia μ^p una misura di Hausdorff ed eleviamo a $\frac{p}{a}$ ottenendo

$$\delta(f(A \cap A_i))^{\frac{p}{\alpha}} \leq c^{\frac{p}{\alpha}} \delta(A \cap A_i)^p$$

Facendo le somme su *i* otteniamo

$$\mu^{\frac{p}{\alpha}}_{c\varepsilon^\alpha}(f(A)) \leq c^{\frac{p}{\alpha}} \mu^p_\varepsilon(A)$$

Se facciamo tendere ε a zero, anche c→0 e la disuguaglianza permane. Quindi possiamo scrivere, per mappe Holderiane,

$$\mu^{\frac{p}{\alpha}}(f(A)) \leq c^{\frac{p}{\alpha}}\mu^{p}(A)$$

Se a = 1 e cioè la mappa è Lipschitziana, otteniamo

$$\mu^{p}(f(A)) \leq c^{p}\mu^{p}(A)$$

Se

$$p = \dim_{H}(A)$$

vuol dire che per q > p

$$\mu^{q}(A) = 0$$

e quindi anche

$$\mu^{q}(f(A)) = 0$$

Le misure di Hausdorff di $f(A)$ per valori maggiori di p sono tutte uguali a zero e questo vuol dire che

$$\dim_{H}(f(A)) \leq \dim_{H}(A)$$

Dunque, le mappe Lipschitziane, riducono la dimensione di Hausdorff.

Definendo invece un mappa bilipschitziana, il risultato precedente applicato prima ad A ed $f(A)$ tramite f, e poi ad $f(A)$ e A tramite l'inversa, consente di ottenere

$$\begin{cases} \dim_H(f(A)) \le \dim_H(A) \\ \dim_H(A) \le \dim_H(f(A)) \end{cases} \Rightarrow \dim_H(A) = \dim_H(f(A))$$

Abbiamo dimostrato in tal modo che la dimensione di Hausdorff è un invariante per mappe bilipschitziane.

CAPITOLO 4

INFORMATION FUSION

4.1 Introduzione all'Information Fusion

L'Information Fusion è un campo di ricerca relativamente nuovo, prevede che i dati provenienti da più sensori di acquisizione vengano fusi insieme al fine di ottenere una "super" informazione, nella quale il livello di dettaglio dei dati e la loro precisione è molto maggiore rispetto a ciò che si può ottenere valutando i dati singolarmente.

Uno degli settori che per primo ha sfruttato le tecniche di fusione dei dati è quello militare, ma con il passare degli anni e grazie allo sviluppo tecnologico, queste tecniche hanno visto una forte espansione ed una grande applicazione anche in ambiti civili.

Molto spesso capita di confondere i due termini *Information Fusion* e *Data Fusion*. Vediamo in Figura 4.1, uno schema esplicativo dei due concetti:

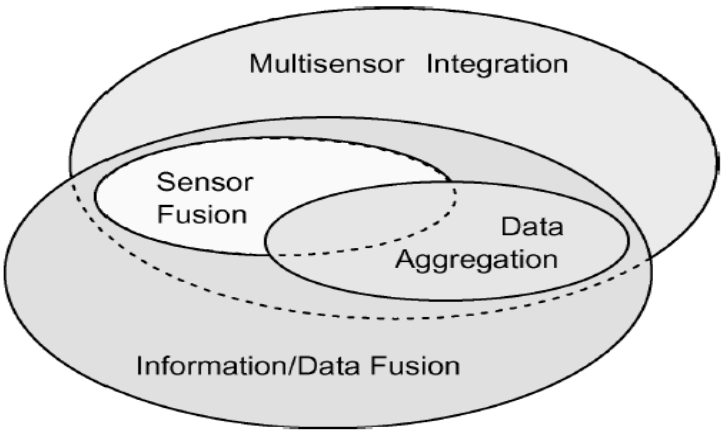


Figura 4.1 – Suddivisione di *information fusion*

Da un'analisi della stessa è chiaro che entrambi i termini *Data Fusion* ed *Information Fusion* possono essere usati come sinonimi.

Inoltre risulta evidente che:

- il *Multisensor/Sensor Fusion* è un sottoinsieme dell'Information Fusion che opera con sorgenti derivate da sensori;
- il *Data Aggregation* definisce un altro sottoinsieme dell'Information Fusion che mira a ridurre il volume dei dati e può manipolare qualsiasi tipo di dato/informazione inclusi quelli derivanti dai sensori;
- il *Multisensor Integration* applica l'Information Fusion per prendere decisioni basate sull'utilizzo di sensori ed informazioni associate (es. estratte da database) al fine di interagire con l'ambiente.

Di conseguenza il *Multisensor/Sensor Fusion* è interamente contenuto nell'intersezione tra *Multisensor Integration* ed *Information/Data Fusion*.

Sono state date, negli anni, alcune definizioni che ci permettono di delineare, in maniera più consona, l'argomento trattato:

"*Un processo multilivello e multiforma tratta*

l'individuazione automatica, l'associazione, la

correlazione, la stima e la combinazione di dati e

informazioni provenienti da sorgenti singole e

multiple".

Viene fornita una definizione di data fusion in cui vengono evidenziate le relazioni con l'informazione e la conoscenza:

"La Data Fusion è un processo di creazione di

conoscenza adattativa in cui diversi elementi di

simili o dissimili osservazioni (DATI), sono

allineati, correlati e combinati in set organizzati e

indicizzati (INFORMAZIONI), che verranno

successivamente valutati per modellare, comprendere e spiegare (CONOSCENZA) la composizione e il comportamento del dominio sotto osservazione".

Un'altra chiara e completa definizione che spesso viene referenziata è quella di Dasarathy:

"l'information fusion comprende la teoria, le tecniche e i tools concepiti e impiegati per sfruttare la sinergia nelle informazioni acquisite da sorgenti multiple (sensori, database, informazioni raccolte da umani ecc), tali che l'azione o la decisione risultante sia in un certo qual modo migliore (quantitativamente e qualitativamente in termini di accuratezza, robustezza, ecc) di quanto sarebbe possibile se le sorgenti fossero usate individualmente senza un tale sfruttamento di sinergie".

Per ultima, non in ordine di importanza, ma meramente cronologico (2007), si cita quella di *Jie Yu* che ne sintetizza e puntualizza i caratteri principali oltre a fornire una rappresentazione semplificata e basilare del modo in cui l'*Information Fusion* lavora indipendentemente dal problema da affrontare :

"L'information fusion è un processo informativo che tratta con :

- *[associazione correlazione e combinazione di dati ed informazioni] da*
- *[singoli e multipli sensori o sorgenti] per archiviare*
- *[stime raffinate di parametri, caratteristiche, eventi e comportamenti] per entità osservate in un determinato campo di osservazione".*

Riassumendo, la fusione di informazioni (anche ridondanti) provenienti da differenti sorgenti può :

- *Ridurre l'ambiguità* della scelta finale.
- *Ridurre la vulnerabilità* di un sistema.
- *Accrescere la precisione* di un sistema.

- *Migliorare la robustezza* di un sistema.
- *Migliorare l'affidabilità* di un sistema.

4.2 Classificazione di un sistema di Information Fusion

Un sistema di Information Fusion può essere classificato sulla base di vari aspetti. Tale classificazione può essere basata sulla relazione tra le varie sorgenti dei dati (cooperativi, ridondanti, complementari), oppure sul livello di astrazione tra i dati manipolati nel corso del processo di fusione (dimensione, segnale, caratteristica, decisione), ed ancora sul livello di astrazione dell'input e dell'output dei processi di fusione. Sulla base delle relazioni che sussistono tra le varie sorgenti, un IF può essere classificato come :

- *Complementare*: le informazioni fornite dalle sorgenti rappresentano porzioni di informazioni complementari di una stessa scena, quindi l'IF può essere applicata per ottenere pezzi di informazioni più complete.
- *Ridondante*: se due o più sorgenti indipendenti forniscono lo stesso pezzo di informazione, questi possono essere fusi per ottenere informazioni più accurate.
- *Cooperativo*: due sorgenti indipendenti sono cooperative quando le informazioni da loro fornite sono fuse in una nuova informazione, tipicamente più complessa di quelle originali, che meglio rappresenta la realtà.

Gli IF possono essere classificati, sulla base del livello di astrazione in quattro categorie:

- *Low-Level Fusion*. Tale livello è anche conosciuto come signal (measurement) level fusion. I dati (non processati) vengono forniti come input e combinati in nuovi pezzi di informazioni più accurati dei primi (in tale maniera si riducono le eventuali interferenze).
- *Medium-Level Fusion*. Tale livello è anche conosciuto come feature (attribute) level fusion. Attributi o caratteristiche di un'entità (per esempio forma, texture, posizione) vengono fuse per ottenere una mappa delle caratteristiche che possono essere utilizzate per altri scopi (ad esempio segmentazione o rilevameto di un oggetto).
- *High-Level Fusion*. Tale livello è anche conosciuto come symbol (decision) level fusion. Questo prende come input decisioni o rappresentazioni simboliche e le combina per ottenere decisioni più affidabili.
- *Multilevel Fusion*. Tale livello occorre quando il processo di fusione abbraccia dati provenienti da differenti livelli di astrazione

L’infrastruttura proposta ha, nel primo caso, come feature (caratteristiche) le minuzie estratte dall'impronta digitale e la chiave dell'algoritmo di crittografia. Mentre, nel secondo approccio, le feature sono rappresentate dalla sequenza random dei numeri generati attraverso i frattali e dalla chiave crittografica. Su queste caratteristiche si vuole agire per realizzare una fusione dei dati in modo da ottenere una chiave d’accesso che abbia un livello di sicurezza maggiore, nella prima scelta, grazie all’univocità dell’impronta, ed un codice crittografico più sicuro e casuale, nella seconda scelta, grazie all’inserimento di metodi generanti caos come i frattali. Per concludere non si può non citare la classificazione di Dasarathy in cui i processi di IF sono categorizzati sulla base del livello di astrazione di input e di output delle informazioni.

Di seguito si specificano le cinque categorie individuate:

- *Data In–Data Out* (DAI-DAO). In questa classe l’IF tratta i dati non processati e risultati anch’essi non processati, anche se in alcuni casi più accurati ed attendibili.
- *Data In–Feature Out* (DAI-FEO). L’IF utilizza i dati non elaborati (grezzi) provenienti dalle sorgenti al fine di estrarre caratteristiche o attributi che descrivono un’entità (oggetto, situazione, ecc.).
- *Feature In–Feature Out* (FEI-FEO). Tale tipologia di fusione opera sull’insieme di caratteristiche al fine di migliorare una “feature” o estrarne una nuova.
- *Feature In–Decision Out* (FEI-DEO). Tale tipologia di fusione opera su un insieme di caratteristiche di un’entità al fine di generarne una rappresentazione simbolica o di derivarne una decisione.
- *Decision In–Decision Out* (DEI-DEO). Le decisioni possono essere fuse in maniera tale da ottenere nuove decisioni o dare efficacia a quelle precedenti.

Tale classificazione può sembrare un’estensione di quella presentata nella precedente sezione caratterizzata da una granularità più fine. Di seguito si evidenziano tali corrispondenze: DAI-DAO corrisponde al Low Level Fusion, FEI-FEO al Medium Level Fusion, DEI-DEO al High Level Fusion, e DAI-FEO e FEI-DEO sono inclusi in Multilevel Fusion. La principale differenza è che Dasarathy specifica i livelli di astrazione basandosi non solo sugli input ma anche sugli output di un processo di fusione riducendo possibili ambiguità.

4.3 Classificatione JDL

Per progettare la tipologia di sistemi di fusione che viene proposta è necessario porre un particolare accento sul JDL model, poiché base di molti tra i modelli esistenti oltre che di diverse soluzioni applicative.

La nozione di fusione di dati è stata formalizzata dalla community che si occupa di tale categoria attraverso il JDL Data Fusion Model.

Tale modello ha lo scopo di:

- Fornire un punto di riferimento per discussioni relazionate all’IF;

- Facilitare la comprensione dei tipi di problemi per i quali è possibile utilizzare tecniche di IF;
- Standardizzare le caratteristiche dei problemi relazionati alla fusione;
- Fornire un framework per automatizzare le soluzioni;
- Classificare i differenti tipi di processi di fusione.

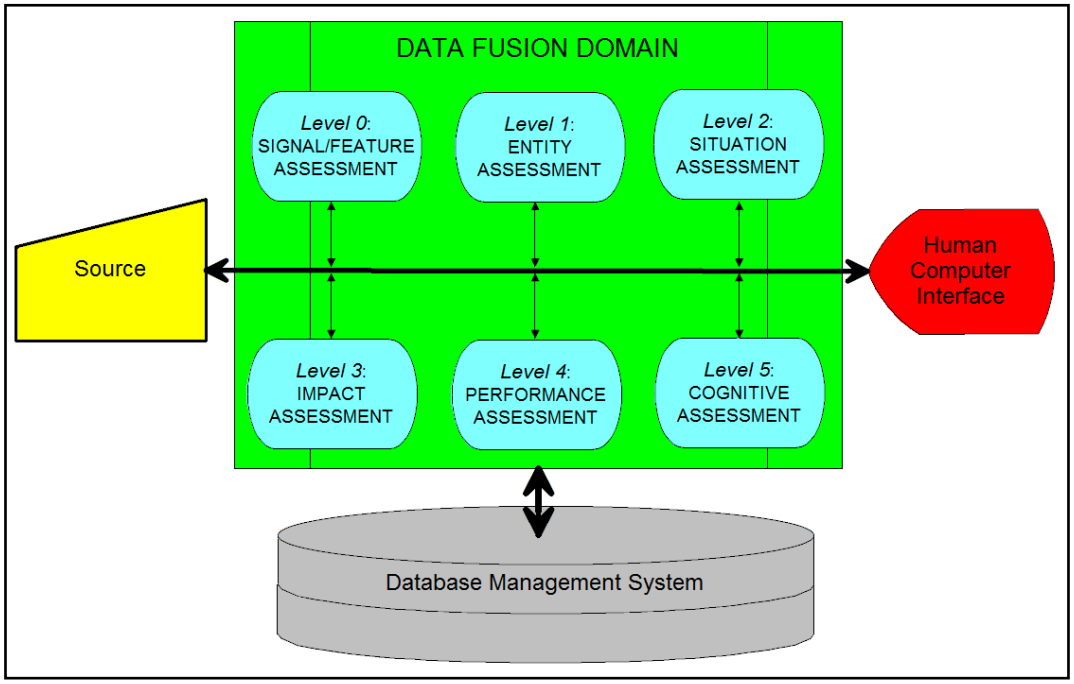


Figura4. 1: Modello JDL

Il JDL model [Figure 4.2] è caratterizzato da *Sorgenti*, un *Database Management System*, una *Human Computer Interface* e cinque livelli di processo.

Le *Sorgenti* sono quelle che forniscono le informazioni di input e possono essere sensori, conoscenze pregresse (per esempio, riferimenti ed informazioni geografiche), database, o input forniti dall'utente.

Il *Database Management System* supporta la memorizzazione dei dati utilizzati e forniti dal sistema di IF.

La *Human Computer Interface* è l'insieme di meccanismi che consentono all'utente di interagire con il sistema o di ricevere notifica dei risultati di fusione attraverso allarmi, display, grafici e/o suoni.

In particolare, il JDL model fornisce una vista funzionale dei processi di fusione, definiti attraverso i livelli di seguito specificati:

- Livello 0: stima e predizione dello stato di un oggetto osservabile sulla base dell'associazione e caratterizzazione di dati a livello di pixel/segnale.

- Livello 1: stima e predizione dello stato delle entità sulla base di inferenze ricavate dalle osservazioni catturate (vengono combinati i dati dei sensori per ottenere una stima accurata della posizione, della velocità, degli attributi e dell'identità dell'oggetto/individuo).
- Livello 2: stima e predizione dello stato delle entità sulla base delle relazioni tra entità ed eventi nel contesto del loro ambiente.
- Livello 3: stima e predizione degli effetti delle azioni pianificate in varie situazioni.
- Livello 4: acquisizione adattiva dei dati e processamento per supportare gli obiettivi e migliorare le prestazioni.

I vari livelli sono, quindi, differenziati principalmente sulla base del tipo di processo di stima. Successivamente, è stato introdotto anche il Livello 5: Cognitive/User Refinement.

Tale modello, prende in considerazione una fusione tra dati dello stesso tipo mentre, nell'Information Fusion Ibrida, si utilizzano dati eterogenei.

Tale modello, quindi, risulta difficile da associare ad un approccio ibrido di tale fusione; nonostante ciò, è possibile classificare la nostra applicazione a livello 1 del modello JDL, qualora lo si estenda opportunamente per gestire la tipicità legata a dati ibridi.

4.4 Problematiche legate all'Information Fusion e letteratura

Sistemi biometrici che integrano (fondono) informazioni ad un fase precedente a quella decisionale, cioè intesa come quella fase durante la quale si autentica oppure no un individuo (si fornisce quindi una risposta VERO/FALSO), risultano essere più performanti in quanto contengono una maggiore quantità di informazioni circa i dati biometrici in input rispetto ai punteggi di matching degli algoritmi decisionali, fornendo quindi un risultato della verifica più affidabile.

La fusione a questo livello si riferisce, infatti, al combinare vettori di differenti caratteristiche ottenuti utilizzando sensori multipli o impiegando algoritmi differenti di *feature-extraction* sullo stesso sensore di acquisizione.

Quando i vettori delle caratteristiche contengono dati omogenei, ad esempio impressioni multiple dello stesso dito dell'utente, un singolo vettore finale delle caratteristiche può essere ottenuto da una media pesata dei singoli vettori delle caratteristiche estratti ad ogni impressione, creando, così, la cosiddetta *Super-Mappa*.

Quando i vettori estratti contengono informazioni relative a differenti modalità di rilevazione delle caratteristiche, è possibile applicare una tecnica di concatenazione dei due vettori in modo da ottenere un singolo vettore.

Il problema si pone quando vengono trattati set di caratteristiche incompatibili, ad esempio vettore contenente le minuzie di una impronta digitale ed un altro contenente i coefficienti estratti da un algoritmo di biometria del volto.

In questo caso ovviamente la media pesata dei valori non è applicabile, in quanto si parla di dati relativi a caratteristiche differenti. Inoltre, non è applicabile neanche la tecnica di concatenazione, sempre per il motivo appena citato.

L'integrazione al livello *feature* risulta essere più difficoltoso da realizzare in quanto, oltre ai problemi appena espressi, le relazioni tra spazi di caratteristiche provenienti da differenti sistemi biometrici possono non essere noti. Nel caso in cui le relazioni tra gli spazi sono note in anticipo, potrebbe essere necessario eliminare informazioni che risultano avere un alto livello di correlazione.

Questo richiede l'applicazione di algoritmi di selezione delle caratteristiche prima della classificazione.

La concatenazione di due vettori delle caratteristiche può generare un vettore di grandi dimensioni e non sempre unidimensionale. Questo può causare dei problemi in fase di storage e di elaborazione del vettore stesso. Tale problema è molto sentito in quelle applicazioni in cui i tempi di risposta del sistema devono essere molto ridotti, come nel caso di un sistema di identificazione dell'individuo.

Ancora, molti sistemi commerciali non forniscono l'accesso ai vettori delle caratteristiche estratti, il che comporta l'impossibilità di effettuare la fusione delle caratteristiche.

Riassumendo il problema di questo approccio, nell’ambito di studio proposto, risulta molto performante quando si trattano informazioni simili, intese come due tracce biometriche estratte da più fasi di acquisizione della stessa caratteristica biometrica. Risulta invece inaccettabile per dati di differente provenienza.

In letteratura si tende a fare una differenza tra fusione a basso ed alto livello.

La maggior parte delle ricerche condotte finora ha interessato i livelli più bassi quali elaborazione di segnali e fusione multisensore dei dati, mentre ad alto livello ancora il campo è abbastanza inesplorato.

Nell’ottica della Fusione Multisensore, in cui le informazioni vengono catturate da differenti sensori per essere combinate, sono presenti i maggiori lavori.

Nel 1997, Hong e Jain, integrarono le impronte digitali con la biometria del volto per l’identificazione dell’individuo. Nel 1999 è stata effettuata una fusione tra sistemi diversi di verifica delle impronte digitali, combinandoli tra loro in modo da rendere il sistema risultante provvisto delle caratteristiche migliori di ognuno dei sistemi di verifica. Negli anni a seguire le fusioni tra componenti biometriche sono state numerose: nel 2001 un sistema multimodale biometrico fondeva informazioni della biometria del volto, delle impronte digitali e della geometria della mano; nel 2003 un sistema combinava immagini del volto e dell’orecchio per facilitare il riconoscimento dell’individuo; nel 2004 un metodo per la fusione delle componenti biometriche del volto e dell’impronta della mano; nel 2006 un sistema per la fusione tra i dati del volto e dell’iride per ridurre gli svantaggi dei singoli approcci biometrici con la costruzione di un Database multimodale biometrico per la convalida dell’utente. Nel 2007, è stato sviluppato un sistema unimodale biometrico che estrae da un unico sensore le informazioni riguardanti l’impronta digitale, l’impronta della mano e la sua geometria che combinate tra loro verificano l’identità della persona.

A differenza degli altri sistemi multimodali, l'utente non dove utilizzare diversi sensori, le tre componenti biometriche possono essere estratte tutte dalla stessa immagine.

Come detto precedentemente, spesso si parla di tre possibili livelli di fusione:

- Fusione a livello di estrazione di *feature*
- Fusione a livello di confronto di *score*
- Fusione al livello *decision*.

Nel 2008, Tao e Veldhuis hanno sviluppato un metodo di fusione ibrida che combina le fusioni a livello *score* ed a livello *decision*. Ovviamente, questo comporta una maggiore robustezza del sistema proposto. Tale metodo di fusione si basa sul lemma fondamentale di Neyman-Pearson, che, come è noto, è un criterio stocastico che rende più robusto l'approccio al livello *decision*.

Iovane et al., nel 2009, propongono un sistema altamente performante per stabilire le proprietà intellettuali di un file utilizzando tecniche avanzate di Information Fusion. Tali tecniche variano al variare del file (testuale, immagine o multimediale) e permettono la creazione di Blue Code biometrici attraverso meccanismi di differenti watermarking basati, rispettivamente, su wavelet e wavelet packet. Questo sistema di Info-Security è applicabile nella pubblica amministrazione.

Per quanto riguarda la fusione ibrida, cioè fusione tra dati eterogenei, nel 2001 da Poh e Korczak, è stato proposto un sistema di fusione tra il viso e la voce. Questo sistema è stato elaborato per l'autenticazione della persona. Un lavoro simile è stato svolto nel 2004 dal gruppo di ricerca di Toh, combinando tra loro impronte digitali e voce. Nel 2004, inizia lo sviluppo dei sistemi ibridi per riuscire ad abbattere i problemi che le tradizionali misure di sicurezza, quali password, hanno nell'ambito dell'autenticazione personale. Anche i sistemi biometrici soffrono di alcuni limiti. Per questi motivi, l'idea più efficace per raggiungere un livello di sicurezza maggiore è quella di combinare i due fattori.

Connie et al., hanno sviluppato un sistema basato sul prodotto interno di numeri pseudo-casuali e le caratteristiche dell'impronta della mano, generando un codice utente compatto. Tale codice, denominato PalmHash, secondo i test proposti, produce un tasso di errore pari a zero per l'autenticazione dell'individuo. Un approccio simile è stato proposto da Jin et al fondendo impronte digitali e numeri pseudo-casuali nel sistema denominato BioHashing. In questo sistema, all'immagine dell'impronta digitale viene applicata prima la Wavelet integrata e in seguito la trasformata di Fourier-Mellin. Viene effettuato il prodotto interno con il numero pseudo-casuale, applicato un threshold e generato un vettore di bit. Anche in questo caso il tasso di errore è pari a zero. Le critiche sul presunto *zero errori* dei precedenti lavori furono molte; in particolare, Kong et al., esplicitarono tali problemi, identificando il punto debole dell'approccio del BioHashing nel token del numero pseudo-casuale; quest'ultimo poteva essere perso, ma le condizioni dei test erano state fatte partendo dall'assunto che il codice numerico non fosse mai perso, dimenticato o condiviso. Intanto, nel 2005, veniva proposto il passaporto elettronico che impiegava due nuove tecnologie: RFID (Radio-Frequency Identification) e la biometria. Nel 2006, Lumini e Nanni, hanno proposto un metodo multimodale che combina le caratteristiche dell'impronta digitale con

quelle del volto, solo che queste ultime sono fuse con numeri pseudo-casuali. Gli stessi autori, nel 2008, hanno proposto una variante del sistema sviluppato nel 2006 in cui il volto viene sostituito dall'impronta della mano, che viene combinata con una chiave personale costituita da numeri pseudo-casuali. In questo lavoro viene esposto anche un esempio in cui un impostore ruba la chiave: nonostante ciò il sistema, con un errore vicino allo zero, riconosce l'impostore come tale.

Nel 2008 viene proposto un protocollo di sicurezza delle reti, basato sulla fusione tra una password e la biometria del volto per l'autenticazione dell'utente. Il match biometrico viene effettuato da un modulo in esecuzione sulla workstation dell'utente, il quale viene autenticato da un *mobile agent* proveniente da un server affidabile.

Nel presente lavoro, sfruttando i principi su cui si basano le tecniche di Information Fusion classiche, è stato sviluppato un algoritmo innovativo che utilizza una tecnica di fusione dei dati.

Questa è stata principalmente studiata per permettere la fusione di dati eterogenei, quali la chiave di un algoritmo di crittografia basata sulla primalità e una componente biometrica (ad esempio le impronte digitali), andando a creare una chiave di crittografia ibrida.

La stessa tipologia di fusione è applicata per creare il codice segreto ottenuto dalla combinazione della sequenza numerica pseudo-random e dalla chiave dell'algoritmo di crittografia.

In entrambi i casi, ci si è basati sull'idea di rendere tanto più casuale, per un utente esterno, il metodo di costruzione delle nuove chiavi generate dall'algoritmo di fusione, rendendo, però, il tutto dipendente dalla chiave basata sulla primalità.

CAPITOLO 5

Secure Infrastructure Information Fusion for Authentication and Encryption (SIIFAE)

5.1 Introduzione

Gli studi precedentemente illustrati da un punto di vista metodologico e teorico in questo capitolo vengono presentati in un dimostratore applicativo.

Viene di seguito riportato lo schema a blocchi [Figure 5.1] che elenca le fasi principali di tale applicazione, che prende il nome di *Secure Infrastructure Information Fusion for Authentication and Encryption* (**SIIFAE**).

Lo schema è suddiviso in due parti, che rappresentano, rispettivamente, le due modalità di utilizzo dell'applicazione:

- *Authentication*: che genera il codice necessario per l'autenticazione di un individuo, **Hybrid Finger Code**;
- *Encryption*: che genera la nuova chiave crittografica, **Hybrid New Key**.

La modalità *Authentication* utilizza:

- *Fingerprint Algorithm*: che si occupa dell'estrazione delle minuzie e delle singolarità da una impronta digitale e della generazione del *Finger Code*;

mentre l'*Encryption* utilizza:

- *Fractal Number Algorithm*: che genera un numero pseudo-casuale mediante le formule derivate dai frattali, cioè il *Fractal Random Number*.

Infine, entrambe le modalità, utilizzano:

- *RSA Algorithm*: che ha lo scopo di creare il *Module* e la *Private Key*;
- *Hybrid Information Fusion Algorithm*: che rappresenta il fulcro dell'applicazione e si occupa di fondere i dati attraverso una tecnica innovativa che permette la creazione sia del **Hybrid Finger Code** che del **Hybrid New Key**.

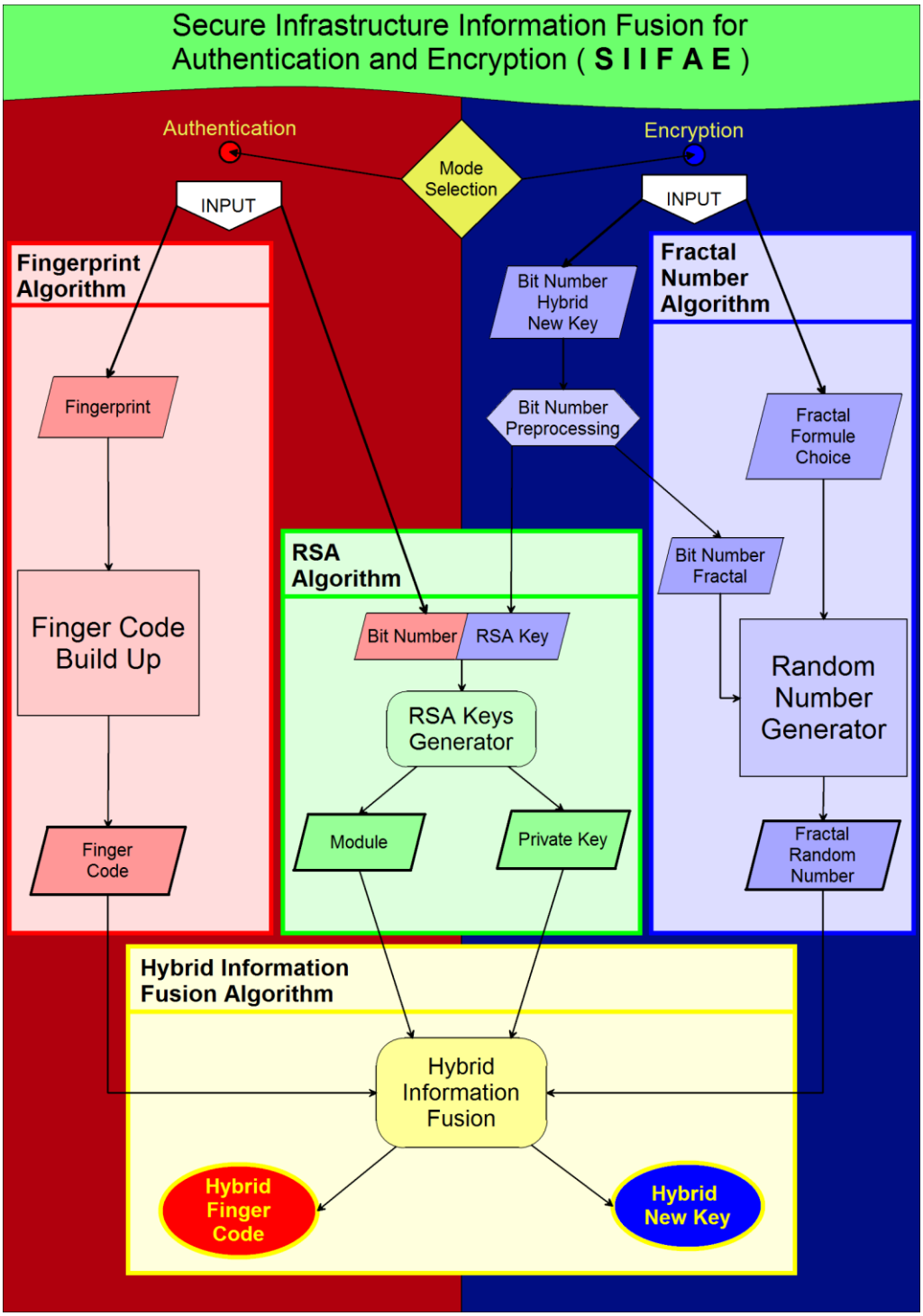


Figura 5.1: Diagramma Infrastruttura SIIFAE

Quindi, nella modalità di *Authentication* la fusione dei dati avviene tra il *Finger Code* dell'impronta ed il prodotto di due numeri primi (*Module*) del RSA.

Per la modalità di *Encryption*, invece, la fusione è tra il numero pseudo-casuale generato tramite le formule frattali (Julia, Cantor, Sierpinski e Peano), cioè il *Fractal Random Number*, ed il prodotto di due numeri primi (*Module*) del RSA.

Qui di seguito, si analizzeranno più in dettaglio le singole modalità di utilizzo e gli algoritmi che le compongono.

5.2 Autentication

Come detto precedentemente, la modalità di *Authentication*, ha lo scopo di verificare l'identità di un individuo con un elevato grado di certezza utilizzando l'idea innovativa di correlare due campi differenti della sicurezza informatica: Biometria e Crittografia a chiave pubblica.

Come componente biometrica si è deciso di utilizzare l'impronta digitale, tecnica più antica e più utilizzata per l'identificazione dell'individuo. Come componente crittografica si è scelto il Modulo del RSA, ad oggi, il migliore algoritmo di Crittografia a chiave pubblica che basa la sua forza esattamente sulla difficoltà di fattorizzare il prodotto di due numeri primi grandi. Tale prodotto è esattamente il Modulo.

Le parti relative al *RSA Algorithm* e al *Hybrid Information Fusion Algorithm* verranno analizzate in dettaglio in seguito in quanto, come già detto, sono algoritmi comuni ad entrambe le modalità di *Authentication* ed *Encryption*.

L'impronta utilizzata è una traccia biometrica dalla quale è possibile estrarre differenti tipi di informazione, cioè *singolarità* e *minuzie*, tramite le quali calcolare ulteriori parametri. Effettuando una scelta mirata di queste caratteristiche è possibile ottenere un vettore di dati biometrici modulare più o meno complesso aumentando così anche la robustezza del *Hybrid Finger Code* che si andrà a creare.

Come Input, per tale modalità si ha, quindi, l'immagine di un'impronta digitale e la dimensione in bit della chiave dell'algoritmo RSA.

La prima fase dell'*Authentication* è rappresentata dal *Fingerprint Algorithm*.

Lo schema a blocchi, Figura 5.2, riporta le fasi principali di tale algoritmo.

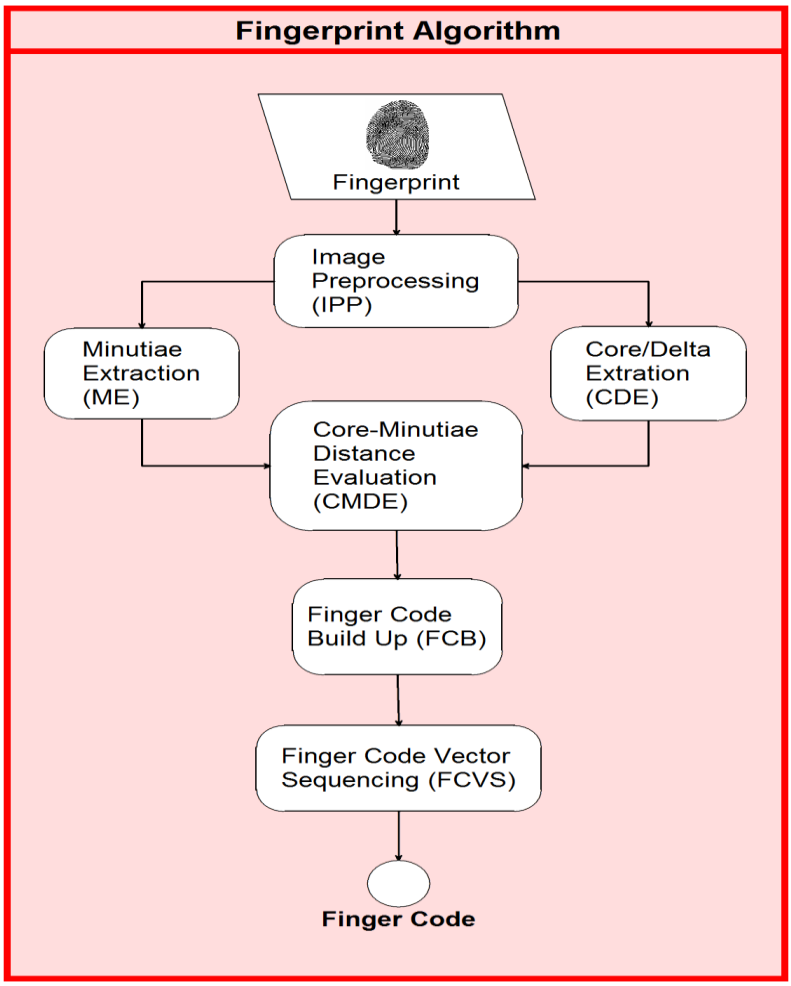


Figura 5.2: Algoritmo Finger print

Si analizzeranno più in dettaglio le singole operazioni di questo algoritmo.

5.3 Image Preprocessing

Prima di poter applicare le operazioni di estrazione delle caratteristiche e di creazione del Finger Code all'impronta appena caricata è necessario apportare le dovute manipolazioni all'immagine dell'impronta, al fine di ottenere una sorgente dei dati che sia il più possibile raffinata. Spesso, in immagini di bassa qualità, la struttura delle ridge line risulta essere poco definita, causando la generazione di false minuzie, la perdita di minuzie fondamentali ed errori di localizzazione. Per questi motivi è necessario applicare delle funzioni di *enhancement* all'immagine in modo da assicurare un funzionamento ottimale dell'algoritmo che si occupa dell'estrazione.

Con tale obiettivo, si applica il *Filtro di Gabor*.

Nella Figura 5.3 è possibile vedere un esempio di Filtro di Gabor 2D.

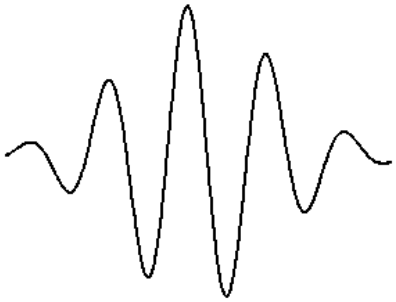


Figura 5.3 – Filtro di Gabor bidimensionale.

Si sceglie di utilizzare questo filtro in quanto ha delle proprietà selettive in base alla frequenza e all'orientamento. Queste proprietà permettono di impostare il filtro in modo tale da avere la risposta massima in corrispondenza delle ridge line. (Figura 5.4).

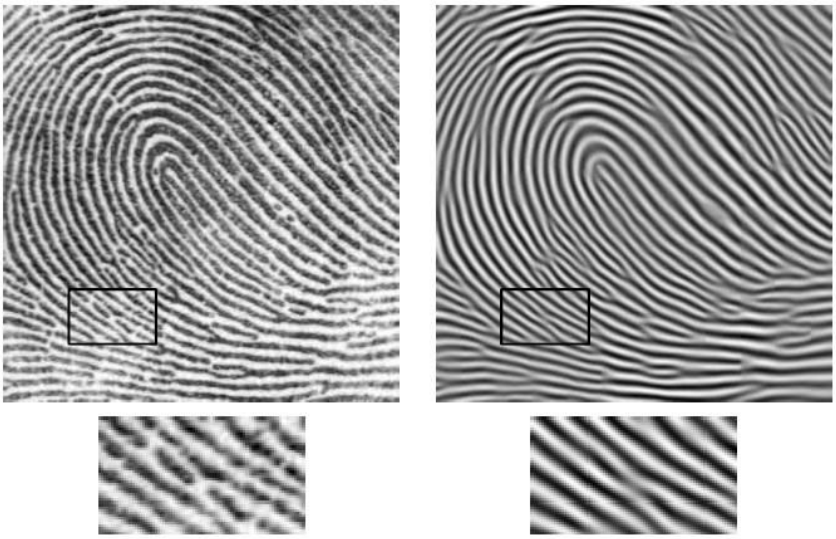


Figura 5.4 – Applicazione del filtro di Gabor all'immagine di una impronta

Un’altra operazione necessaria da applicare all’immagine dell’impronta è il *threshold* che permette di trasformare l'immagine originale in scala di grigi, con valori compresi tra 0 e 255, in una binaria con valori 0 o 255 (Figura 5.5).

Figura 5.5 – Impronta sogliata

Ottenuta, quindi, una immagine binaria, è necessario applicare la funzione di *thinning* (assottigliamento) all'impronta per poterne estrarre le minuzie e le singolarità. Il *thinning* è una operazione morfologica usata per rimuovere i pixel di *foreground* da immagini binarie. Vengono considerati pixel di *foreground* quelli il cui valore è 0, mentre quelli di *background* hanno valore 255. Essa può essere usata per diverse applicazioni, ma è particolarmente utile per la scheletrizzazione. In questo algoritmo è utilizzata per ridurre le linee di una immagine ad un singolo pixel di spessore.

Per ottenere una immagine totalmente scheletrizzata è necessario applicare il thinning ripetutamente, fin quando non viene effettuata più alcuna modifica all'immagine, cioè quando nessun pixel viene più eliminato.

Un esempio di immagine scheletrizzata è mostrato in Figura 5.6.

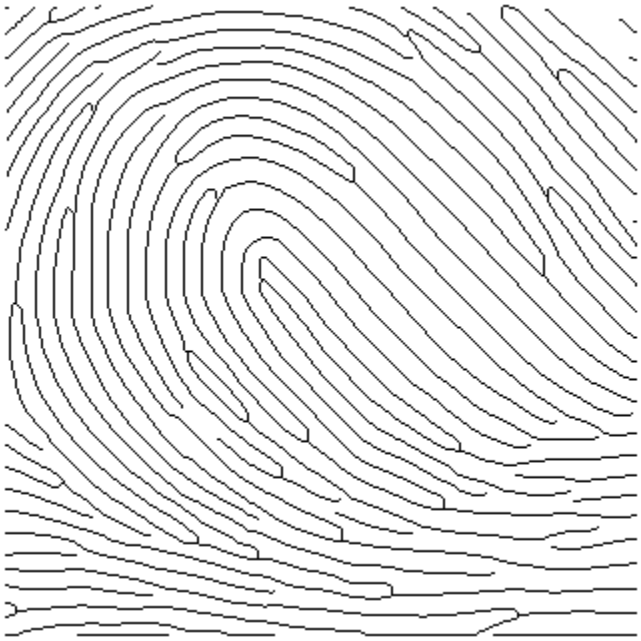


Figura 5.6 – Impronta scheletrizzata

5.4 Core/Delta Extraction

Prima di poter estrarre le minuzie dall'impronta è necessario individuare i possibili Core e Delta esistenti nell'impronta.

L'algoritmo utilizzato per trovare le singolarità, proposto da Iovane, consiste nel creare due immagini **PX** e **PY**, la cui dimensione sarà pari alla metà di quelle dell'immagine di partenza, costruite nel seguente modo:

for (i=2; i<width; i+=2){

for (j=2; j<height; j+=2){

dx = img(i,j+1) - img(i,j-1);

dy = img(i+1,j) - img(i-1,j);

PX(i/2,j/2) = dy * dx;

PY(i/2,j/2) = abs(dy) - abs(dx);

}

}

Sulle due matrici appena create si applica la convoluzione utilizzando un filtro *Gaussiano*.

La convoluzione porterà ad ottenere matrici con valori non più appartenenti all'intervallo [0, 255]. Questo comporta un'operazione di sogliatura che inizializzi a zero i valori negativi ed a 255 quelli positivi. In Figura 5.7 è possibile vedere il risultato delle operazioni appena descritte.

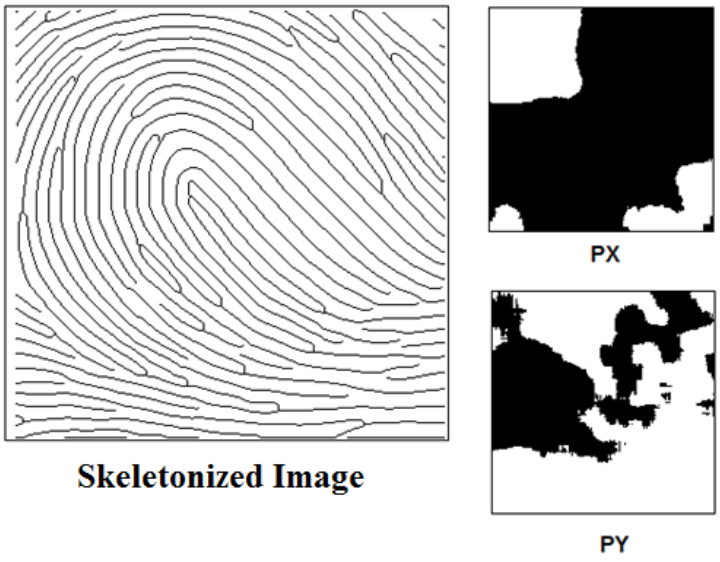


Figura 5.7 – Immagine scheletrizzata, PX e PY.

La somma di queste matrici, **PX** e **PY**, da luogo a delle zone omogenee in cui è possibile trovare valori pari a 510, ovvero zone in cui sia **PX** che **PY** hanno valori pari a 255. È proprio sul bordo di queste zone che è possibile trovare core e delta.

Per individuare queste zone è necessario applicare l'*operatore di Sobel* lungo la direzione orizzontale e verticale alle due matrici **PX** e **PY**. In questo modo è possibile estrarre i contorni delle aree in cui i pixel hanno valore 255, come si può vedere nella Figura 5.8.

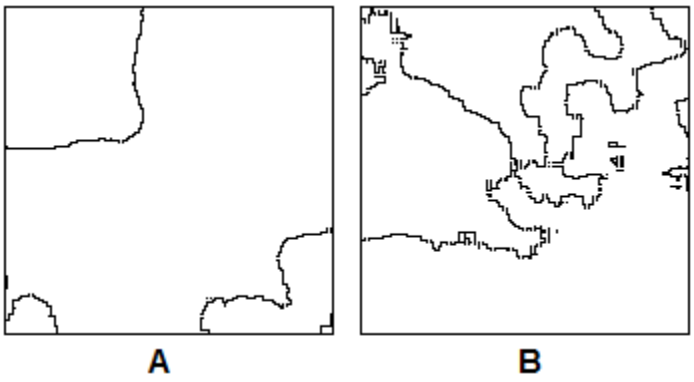


Figura 5.8 – Operatore di Sobel applicato alle matrici PX (A) e PY (B)

Estratti i contorni delle due aree, non resta che effettuare l'*AND* tra le due immagini così da ottenere i punti in cui l'unione tra le due matrici diano come risultato 510. Questi punti saranno le singolarità presenti nell'impronta (vedi Figura 5.9).

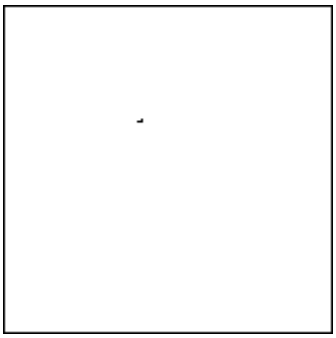


Figura 5.9 – Singolarità individuata

Notiamo come il punto individuato corrisponde esattamente al core presente nell'immagine dell'impronta scheletrizzata.

Vengono considerati falsi core/delta, quindi eliminati, quelle singolarità che si trovano ai bordi dell'immagine, in quanto, in genere, core e delta si trovano nella parte centrale dell'impronta.

Eliminati i falsi positivi, non resta che esaminare l'immagine per individuare il tipo di singolarità individuata, tramite il seguente calcolo (**PX** e **PY** sono le stesse calcolate in precedenza):

$$\boldsymbol{A} = [\mathrm{PX}(i, j-1) - \mathrm{PX}(i, j+1)] * [\mathrm{PY}(i-1, j) - \mathrm{PY}(i+1, j)]$$

$$\boldsymbol{B} = [\mathrm{PY}(i, j-1) - \mathrm{PY}(i, j+1)] * [\mathrm{PX}(i-1, j) - \mathrm{PX}(i+1, j)]$$

I parametri ***A*** e ***B*** permettono di distinguere core e delta. Nella Tabella 5.1 sono presentati i diversi casi:

Tabella5.1 – Distinzione tra Core e Delta in base ai parametri A e B

Singolarità	*Parametro*	*Caso 1*	*Caso 2*
Core	A	≥ 0	≥ 0
	B	≥ 0	≤ 0
Delta	A	< 0	< 0
	B	> 0	< 0

5.5 Minutiae Extraction

Trovate le singolarità è possibile eseguire le operazioni per l'estrazione delle minuzie. La prima fase prevede l'individuazione di biforcazioni e terminazioni. Vengono create delle maschere di dimensione 3x3 che corrispondono appunto alle possibili biforcazioni e terminazioni individuabili all'interno di una impronta.

Si esamina ogni pixel di foreground dell'impronta scheletrizzata, estraendo l'otto-connesso, e "contando" il numero di transizioni, da pixel di background a pixel di foreground. Questo numero permette di individuare se si è in presenza di una minuzia e in tal caso quale tipo di minuzia si sta esaminando, cioè terminazione o biforcazione. L'otto-connesso viene confrontato con una delle possibili maschere che identificano la minuzia, ponendo il risultato all'interno di una matrice 3x3. Se il valore del pixel di posizione *i j* nell'otto-connesso combacia con il corrispondente elemento nella maschera, nella matrice temporanea avremo valore 0, altrimenti 1. Il match totale si ha in caso di assenza di pixel pari ad 1. Se il valore ottenuto è zero (quindi match totale con la maschera), abbiamo trovato una minuzia. Di questa vengono considerate:

- tipo
- posizione rispetto all'asse *x*;
- posizione rispetto all'asse *y*;
- angolo.

Questa fase, però, genera un grande numero di false minuzie, molte delle quali si trovano sul bordo dell'impronta (false terminazioni). Un esempio è mostrato in Figura 5.10. È necessario quindi

rimuoverle, andando ad eliminare tutte quelle minuzie che, in base ad una certa tolleranza, si trovano sul bordo lungo la direzione orizzontale e verticale.

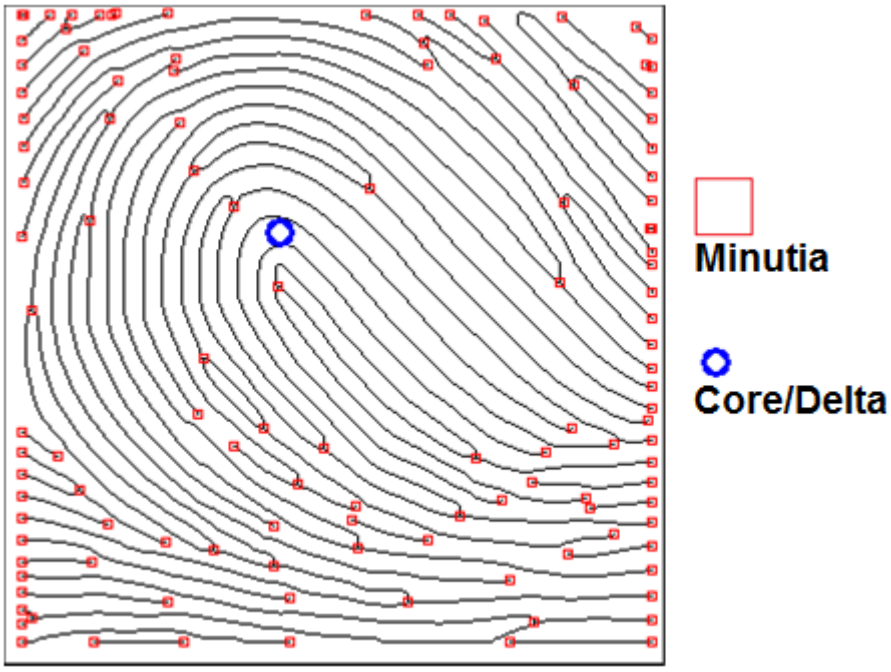


Figura 5.10 – False minuzie

Individuate le minuzie e cancellate quelle di bordo è possibile visualizzare le minuzie e le singolarità individuate. In Figura 5.11 possiamo vedere il risultato di tale operazione.

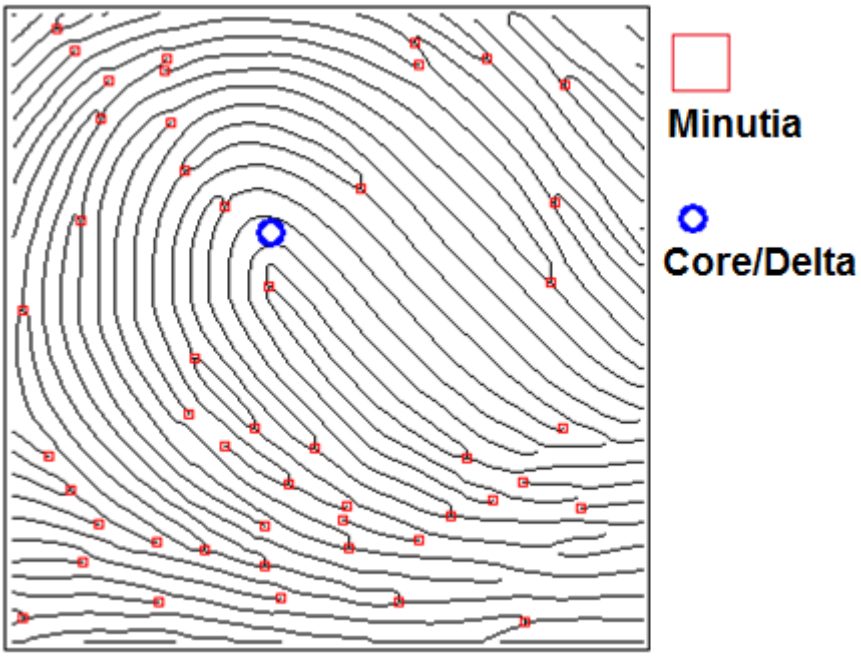


Figura5.11 Esempio di visualizzazione di minuzie e singolarità

5.6 Core-Minutiae distance Evaluation

Le singolarità vengono estratte per ottenere un sistema di riferimento che sia indipendente da roto-traslazioni, al quale riferire ogni minuzia individuata durante la fase di estrazione delle stesse. Nello specifico, viene calcolata la *Distanza Euclidea* tra ogni minuzia individuata e il Core dell'impronta estratto in precedenza. Come detto, ogni impronta può avere anche due Core, in tal caso è necessario calcolare la distanza media tra le due singolarità e in seguito calcolarne la distanza dalle minuzie. Questo valore di distanza andrà ad espandere le informazioni riguardanti la singola minuzia.

5.6.1 Finger Code build up

Le informazioni precedenti, quindi:

1. tipo di minuzia;
2. posizione rispetto all’asse *x*;
3. posizione rispetto all’asse *y*;
4. angolo;
5. distanza euclidea della minuzia dal core;

andranno a comporre le informazioni e le caratteristiche che specificano ogni singola minuzia.

Queste informazioni sono calcolate per ogni minuzia trovata.

5.6.2 Finger Code vector sequencing

Estratte le minuzie, è quindi possibile creare il nostro Finger Code, da utilizzare durante la fase di fusione dei dati. Questa operazione viene eseguita leggendo le informazioni di ogni minuzia e scrivendone il contenuto, carattere per carattere, all'interno di un vettore (la cui dimensione sarà dipendente dal numero di minuzie estratte dall'impronta) (Figura 5.12).

Figura 5.12 – Finger Code

5.7 Encryption

La modalità Encryption ha l'obiettivo di creare un codice crittografico più sicuro e casuale utilizzando la randomicità intrinseca dei frattali e la sicurezza del RSA. Queste vengono combinate attraverso l'algoritmo di fusione. La bontà dell'algoritmo per la generazione di numeri pseuco-casuali attraverso i frattali rende più robusto e casuale anche la *Hybrid New Key* che si creerà. Quindi, come Input, per tale modalità si ha la dimensione in bit della *Hybrid New Key* che si vuole ottenere e la scelta tra le quattro formule dei frattali che si desidera utilizzare per generare il numero pseudo random. Una volta fatto ciò vi è una fase di preprocessing per calcolare la dimensione della chiave del RSA e la dimensione del numero pseudo-casuale in funzione della dimensione della *Hybrid New Key* da generare. Questo avviene attraverso le seguenti formule:

$$dimkey = \frac{dim}{2}$$

$$dimfract = dim - dimkeys$$

con

$$dimkeys > 512\ bit$$

e

$$dimkeys = c * 512\ bit$$

in cui:

dim: dimensione Nuova Chiave Crittografica (maggiore di 520)

dimkeys: dimensione chiave RSA

dimfract: dimensione numero generato con i frattali

In Tabella 1 si possono vedere degli esempi esplicativi:

Tabella 5.2: Dimensione in bit dei vettori RSA e Frattale in funzione della Nuova Chiave Crittografica

Dim	***dimkeys***	***dimfract***
1024	512	512
2048	1024	1024
4096	2048	2048
8192	4096	4096
16384	8192	8192

Come detto in precedenza, per il generatore di numeri pseudo-casuali ideato, si è pensato di utilizzare le formule matematiche che derivano da strutture intrinsecamente caotiche come i frattali. È stato, però, necessario apportare delle modifiche a queste formule, tra cui l'utilizzo di due semi per ogni generatore. La prima fase dell'*Encryption* è il *Fractal Number Algorithm*.

Lo schema a blocchi, Figura 5.13, riporta le fasi principali di tale algoritmo.

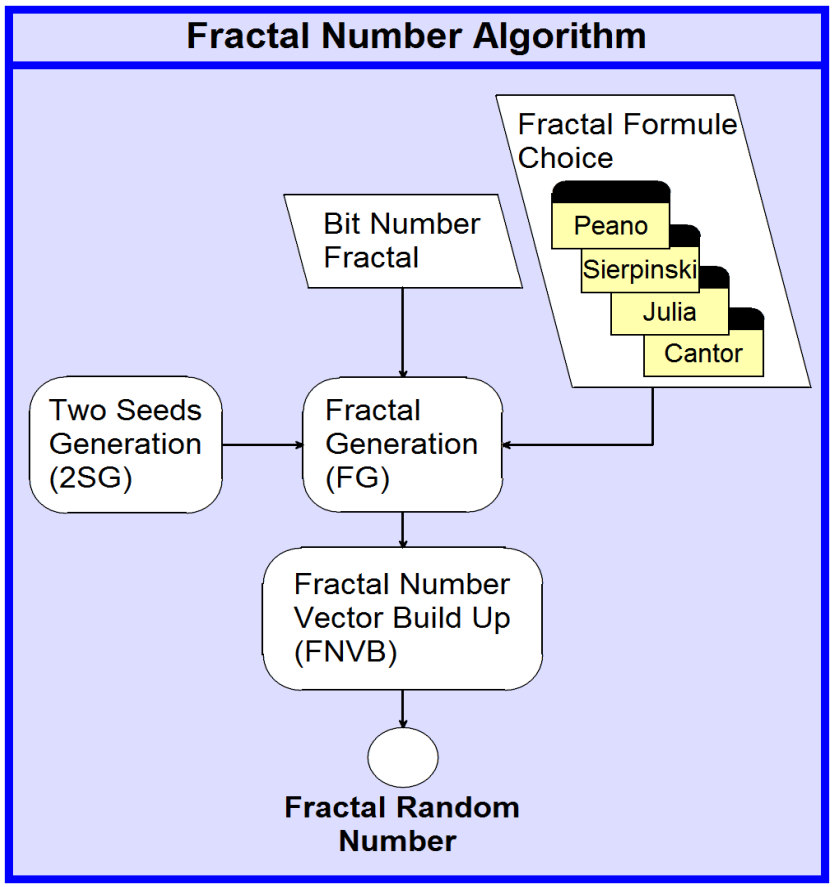


Figura 5.13: Algoritmo Fractal number

5.7.1 Two seeds generation

Come detto in precedenza, si è scelto di utilizzare 2 semi da applicare alle formule di generazione. Questa scelta permette di superare i problemi legati alla periodicità e alla ripetibilità delle sequenze pseudo-casuali.

Tali semi rolling, random ed indipendenti sono entrambi inizializzati a dei valori temporali. Ognuno dei due semi avrà valori differenti grazie all'utilizzo di differenti funzioni di temporizzazione.

Il primo è la combinazione di più dati tra cui millisecondi, giorno e secondi espressi in UTC (Universal Time Coordinated). Questi sono completati dal valore restituito da un'altra funzione di tempo che da informazioni relative al numero di clock trascorsi dall'inizio dell'esecuzione del programma (***seme1***).

Il secondo, invece, è rappresentato dai secondi trascorsi da un istante fissato (***seme2***).

Inoltre, il tempo, una volta trascorso non è più riproponibile, quindi, i semi saranno di volta in volta differenti (*reclocking*).

5.7.2 Fractal generation

Verranno utilizzate le trasformazioni affini dei frattali visti in precedenza.

Questa strategia permette di ottenere numeri pseudo-casuali attraverso le formule dei frattali IFS (***Polvere di Cantor***, ***Triangolo di Sierpinski***, ***Curva di Peano*** e ***Insieme di Julia***).

A tali formule sono state apportate delle modifiche.

I coefficienti sono frazionari, quindi, adoperando delle opportune modifiche computazionali dei frattali, si otterranno i coefficienti interi necessari ai vari generatori per creare sequenze numeriche pseudo casuali intere.

Queste varianti sono state apportate, utilizzando la stessa strategia, sulle formule di ***Cantor***, ***Sierpinski*** e ***Peano***. Tali modifiche non sono state necessarie per il frattale di ***Julia***.

Inoltre, in tutte le formule sono state fatte delle scelte di inizializzazione opportune utilizzando due semi distinti.

Applicando una di queste formule frattali si otterrà un numero intero pseudo random di dimensione in bit desiderata.

5.7.3 Fractal number vector build up

Generato il numero intero attraverso una delle formule proposte è possibile creare il *Fractal Random Number* da utilizzare durante la fase di fusione dei dati. Questa operazione viene eseguita leggendo carattere per carattere il numero e scrivendolo all'interno di un vettore. (Figura 5.14)

```
Fractal Random Number

64644193930675089773556955809994702001800207643996559126017751809898819293921740
96930155454346032466155219919839202699908358675761556913920039233270578640025755
82204489397055744461044965515631650228018131146869625824485012233089298217668527
44748214863077214488734386700092815219095622515924489967517197909610591674035639
62085732791522900854375066064487723041900938472830810337106235986290536780913890
81441116833817732050417093704062596941079111783544152499818241022006474507105742
76016614625289057207827685182577848202636173066416309810251977466580618721790946
31697454729511957089964732301901296041264783263160212080178943266815576500784522
96934472696587162511302289652627269442471383328680582897772740449720784329450004
91255292395711225334652207102
```

Figura 5.14

5.8 RSA

Come già esposto, le due modalità di *Authentication* e di *Encryption* hanno delle fasi comuni rappresentate dal *RSA Algorithm* e dal *Hybrid Information Fusion Algorithm*. Analizzando il primo, l'input di tale algoritmo è rappresentato dalla dimensione in bit delle chiavi da generare.

Viene riportato lo schema a blocchi [Figura 5.15] delle operazioni per l'estrazione della *Private Key* e del *Module* del *RSA Algorithm*.

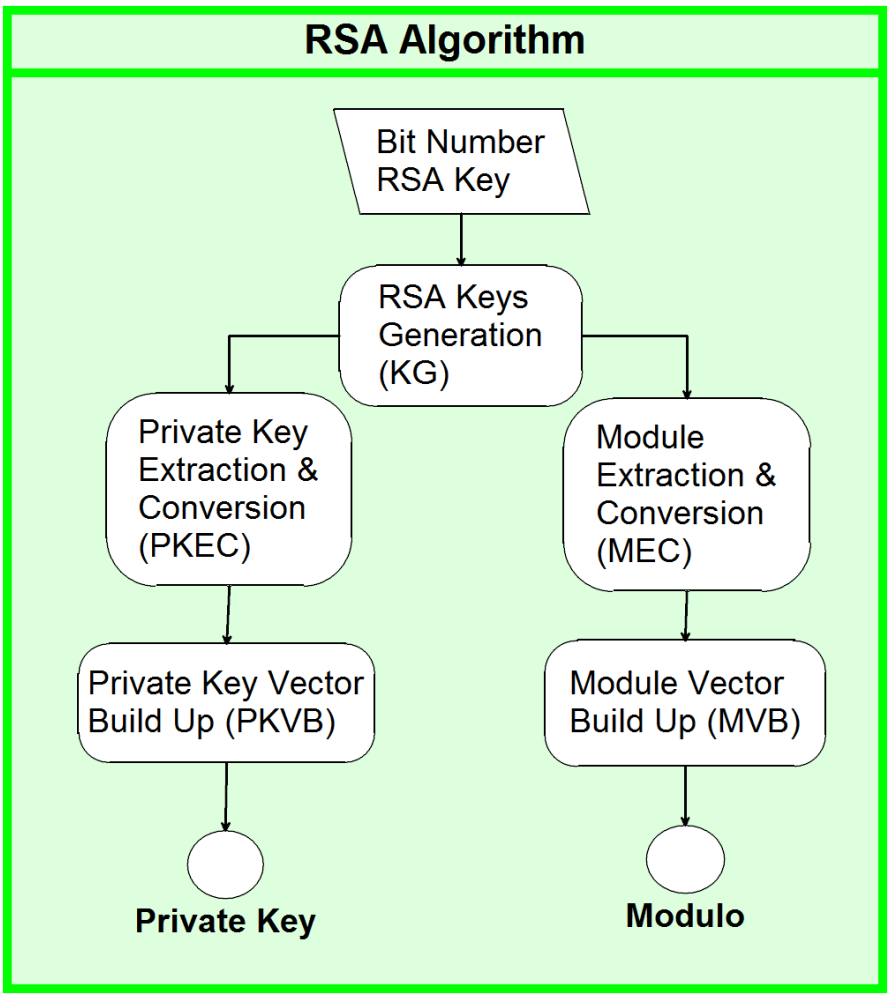


Figura 5.15: RSA Algorithm

Per poter proseguire nella fusione, è necessario disporre della *Private Key* e del prodotto tra i due numeri primi (*Module*), generati tramite l'uso delle classi interne del *Framework .NET 3.5*.

Il *Framework* mette a disposizione una classe che genera un oggetto *RSACryptoServiceProvider*, il quale contiene tutte le informazioni necessarie per eseguire la crittografia di un dato attraverso le specifiche del *RSA Algorithm*, per cui sono disponibili *Module* e *Private Key* necessari per i nostri scopi. L'unico inconveniente è rappresentato dal fatto che le informazioni memorizzate all'interno di tale oggetto (una stringa) hanno una struttura XML e risultano essere convertite in *Base 64*, in quanto questa conversione riduce notevolmente la lunghezza del dato.

Per questo motivo è necessario effettuare, una volta estratto il modulo e l'esponente privato dall'oggetto *RSACryptoServiceProvider*, la conversione dalla suddetta base a quella decimale. Questa operazione di conversione si appoggia ad una mappa hash contenente i corrispondenti valori decimali, per ogni cifra in *Base 64*.

Quindi, vengono estratti dalla stringa XML i dati necessari e convertiti in decimale. In Figura 5.16 e Figura 5.17 vengono mostrati i risultati di tali operazioni.

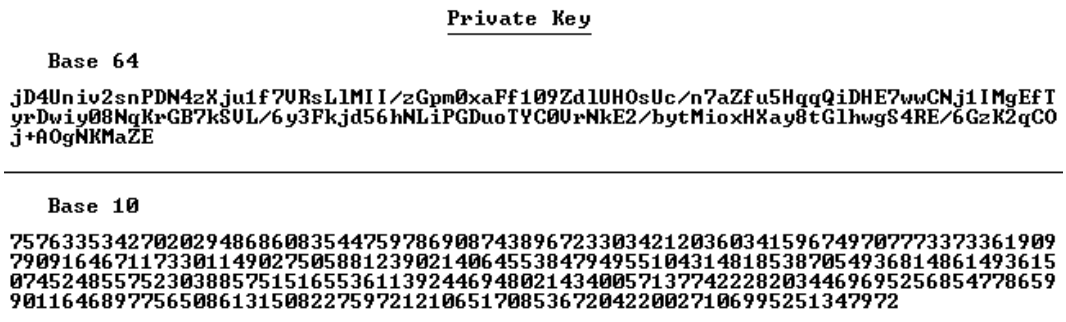


Figura 5.16 – Chiave Privata in base 64 e conversione in decimale

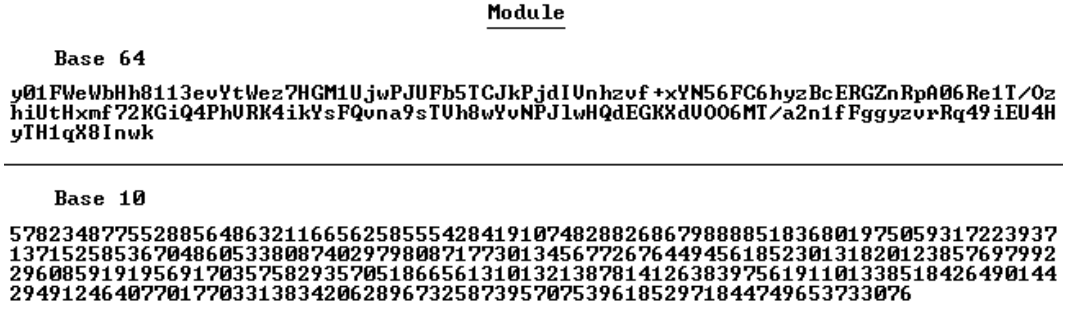


Figura 5.17 –Modulo in base 64 e conversione in decimale

Generato il numero decimale sia della Private Key che del Module tali valori vengono inseriti carattere per carattere in due vettori.

5.9 Hybrid Information Fusion

Il nodo cruciale dell'infrastruttura **SIIFAE** è, appunto, nell'algoritmo di Information Fusion, algoritmo innovativo per le caratteristiche di eterogeneità che lo caratterizzano, in quanto è stato ideato per applicazioni di fusione dei dati che possono essere di varia natura, unico vincolo da rispettare è la creazione di vettori numerici dei dati.

La grande elasticità del metodo viene evidenziata dall'utilizzo che viene presentato, cioè la possibilità di creare *Hybrid Finger Code* per l'*Authentication* e *Hybrid New Key* per l'*Encryption*.

Infatti, l'algoritmo che viene presentato è utilizzato da entrambe le modalità dell'infrastruttura.

Gli Input di questo algoritmo sono rappresentati dagli output degli algoritmi precedentemente descritti e variano in funzione della modalità scelta.

Quindi, nel caso dell'*Authentication* gli Input dell'algoritmo di fusione saranno:

- *Finger Code*;
- *Module*;
- *Private Key*;

mentre, per l'*Encryption* saranno:

- *Fractal Random Number*
- *Module*
- *Private Key*.

Viene di seguito riportato lo schema a blocchi [Figura 5.18] che visualizza le fasi fondamentali del *Hybrid Information Fusion Algorithm*.

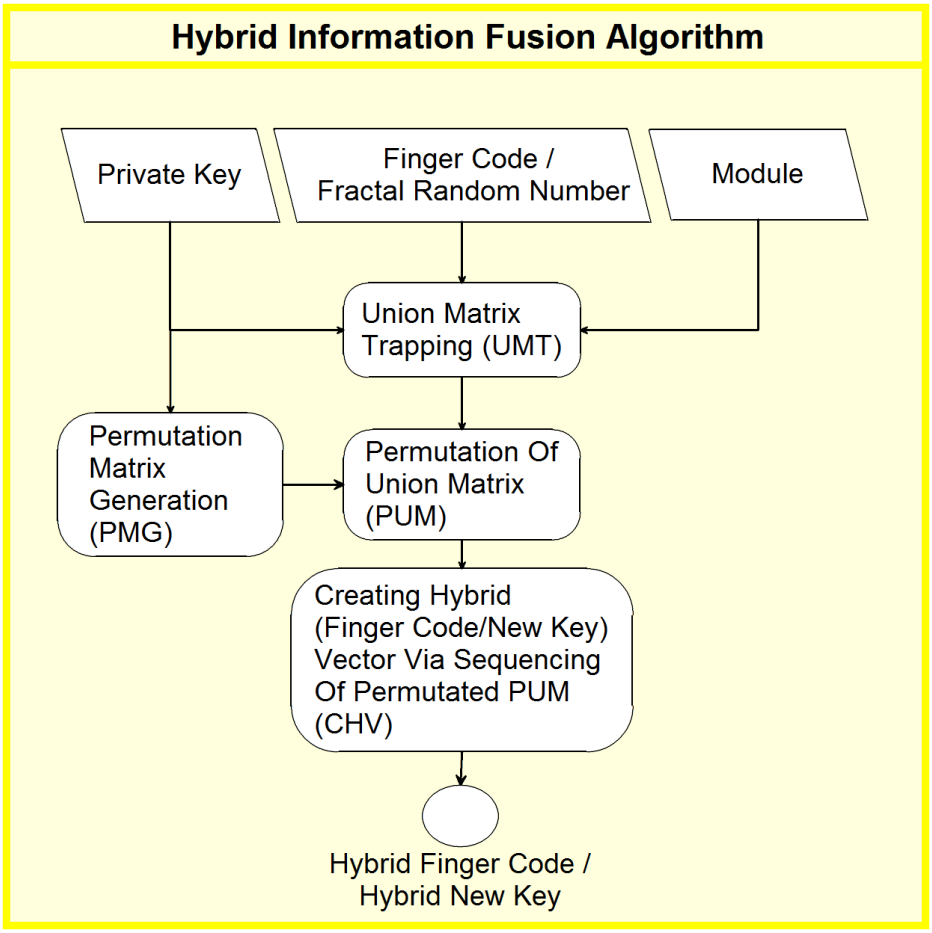


Figura 5.18: Hybrid Information Fusion Algorithm

Quindi, come già detto, tale algoritmo ha lo scopo di ottenere, in base alla modalità di utilizzo, un Hybrid Finger Code oppure un Hybrid New Key partendo da informazioni di tipo biometrico e crittografico, nel primo caso, e di tipo numerico e crittografico, nel secondo. In ogni caso, le componenti da fondere saranno sempre due vettori numerici.

"Union Matrix Trapping"

Fase cruciale di tale algoritmo è la trasformazione dei due vettori in una matrice. È importante che tale matrice sia quadrata e il suo ordine sia dipendente dal numero di componenti dei due vettori. Si analizzeranno in dettaglio le operazioni da eseguire.

Sia $a \in Z^m$ e $b \in Z^n$ due vettori, tali che: *a* contiene il *Finger Code* o il *Fractal Random Number* e *b* il prodotto tra i due numeri primi (*Module*). Sia $s \in Z$:

$s = m + n;$

sia, inoltre, $q \epsilon Z$:

$$q = \lceil \sqrt{s} \rceil$$

intero contenente la radice di *s*.

Affinché la matrice sia quadrata, è necessario che:

$nz1 = q - mod\ (m, q)$,

$nz2 = q - mod\ (n, q)$;

questi rappresentano il numero di elementi di padding di ciascun vettore; per cui definiamo i vettori $a1 \epsilon Z^{m1}$ e $b1 \epsilon Z^{n1}$ con i primi elementi uguali ai vettori *a* e *b* e gli ultimi elementi rispettivamente con *nz1* componenti aggiuntive per il primo vettore ed *nz2* per il secondo. Quindi, le dimensioni saranno:

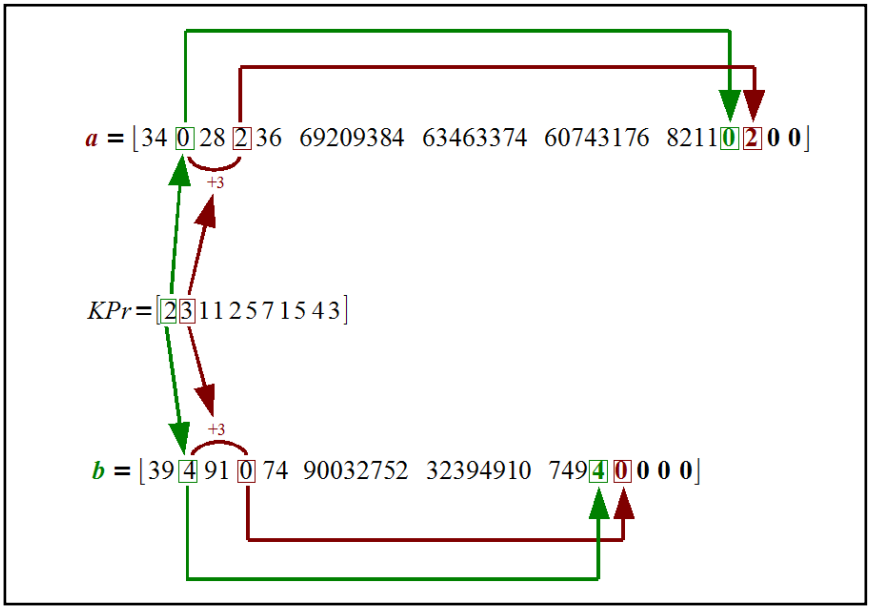
$m1 = m + nz1$,

$n1 = n + nz2$.

Non si effettua uno zero padding ma un riempimento opportuno identificato dalla *Private Key* del RSA.

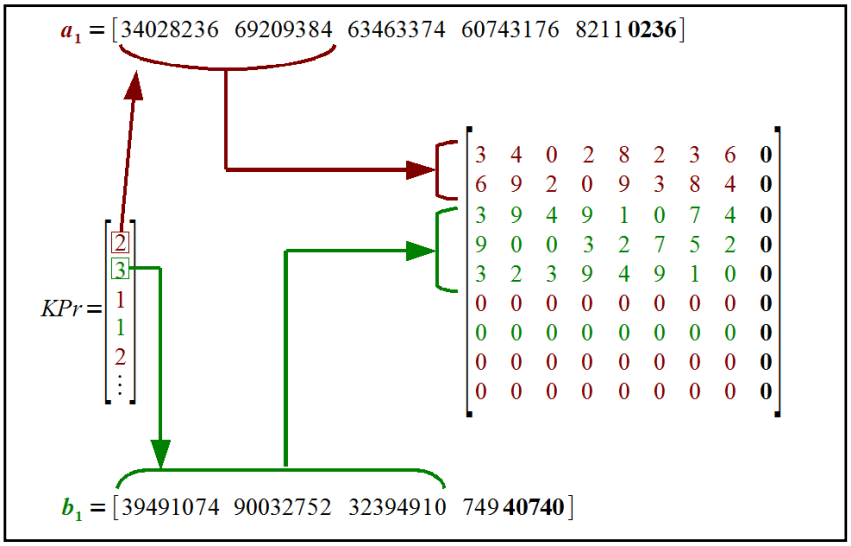
La prima componente della *Private Key* rappresenta l'indice del primo valore da aggiungere come padding; la seconda componente sommata all'indice precedente fornisce l'indice del secondo valore di padding e così via, in maniera ciclica.

Viene riportato un esempio di tale operazione.



Ciascun vettore verrà suddiviso in blocchi che costituiranno le righe della matrice unione *U*, matrice contenente i due vettori inseriti opportunamente secondo i valori della *Private Key* del RSA. Nella costruzione della matrice *U*, l'ordine di inserimento dei blocchi è dettato dalla *Private Key*, generata dal *RSA Algorithm* (considerata anch'essa come un vettore). Si indicizza con '0' il vettore *a1* e con '1' il vettore *b1*. La prima componente della *Private Key* (modulo 2) specifica da quale dei due vettori si deve iniziare la composizione della matrice. Inoltre, il valore della prima componente del vettore della chiave privata definisce anche il numero dei blocchi del vettore da inserire nella matrice; mentre, il valore della componente successiva definisce il numero di blocchi dell'altro vettore, e così via in alternanza. Nel caso in cui vengono inseriti tutti i blocchi di uno dei vettori, si inseriscono sequenzialmente tutti i blocchi dell'altro.

In Figura è visualizzata tale operazione.



Bisogna verificare, infine, che la matrice ottenuta sia effettivamente quadrata, in quanto, in alcuni casi, si possono ottenere un numero di blocchi totale maggiore/minore rispetto al numero degli elementi per ogni singolo blocco, ottenendo così una matrice rettangolare. In questo caso, è sufficiente aggiungere il numero di colonne o righe necessarie per raggiungere una dimensione tale da rendere la matrice quadrata. Sia:

$PadTot = q^2 - (m + n)$,

$Diff = Pad - PadTot$

Allora, si distinguono tre casi:

Diff < 0 => aggiunta di una riga

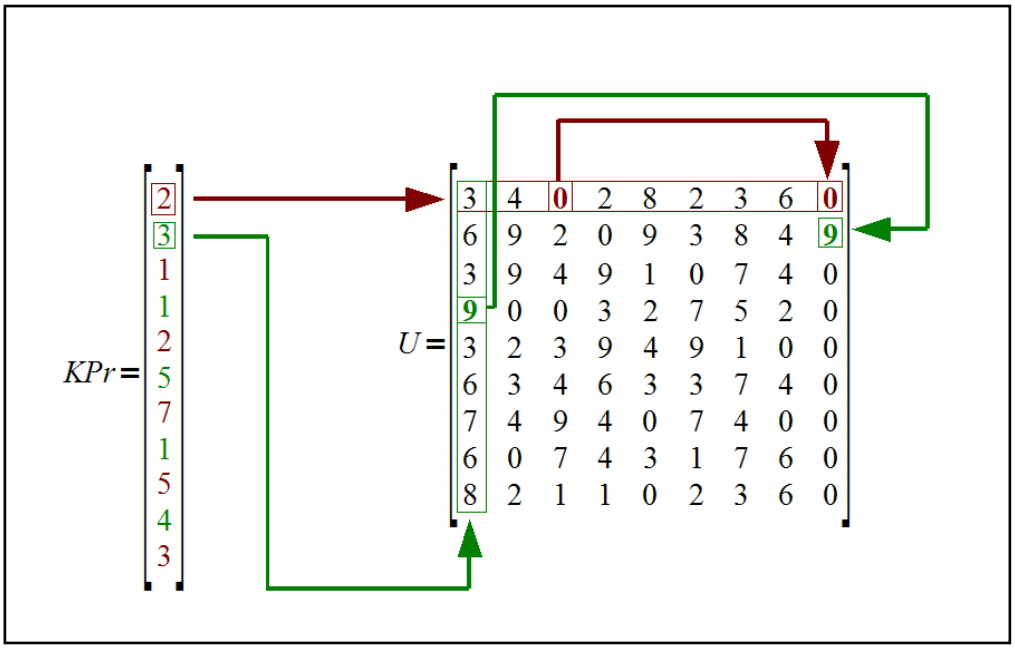
Diff = 0 => no padding aggiuntivo

Diff > 0 => aggiunta di una colonna

La riga o la colonna di padding da aggiungere viene creata sfruttando la *Private Key*. Una volta costruita la matrice, il valore della prima componente della chiave identifica l'elemento della prima riga da cui prende la prima componente del padding; il valore della seconda componente della

chiave identifica l'elemento della prima colonna da cui prende la seconda componente del padding e così via.

In Figura vengono visualizzate tali operazioni.



Costruita la matrice Unione *U* quadrata, viene creata una matrice di permutazione *P*. La matrice di permutazione è creata opportunamente in funzione della *Private Key*.

Vengono considerati un numero di elementi della *Private Key* pari all'ordine della matrice (*n*).

Agli elementi ripetuti nella chiave vengono sostituiti, in maniera crescente, i numeri da *0* a *n-1* mancanti, in modo da essere sicuri che tutti i numeri siano presenti una sola volta.

Per la costruzione della matrice di permutazione viene inserito l'elemento 1 nella colonna identificata dalla chiave privata modificata.

Viene considerato un inserimento degli elementi riga per riga, la colonna invece viene specificata dal valore contenuto nelle singole componenti della chiave.

Quindi, il valore della prima componente della chiave rappresenta il numero della colonna, della prima riga, in cui inserire 1.

L'elemento 1 della seconda riga si inserirà nella colonna identificata dalla componente che si trova alla distanza specificata dalla prima componente della chiave. In maniera circolare, sempre con lo stesso passo, si inseriscono gli altri elementi unitari dalla matrice di permutazione.

Per avere una più chiara visione, si osservi lo schema seguente [Figure 5.19].

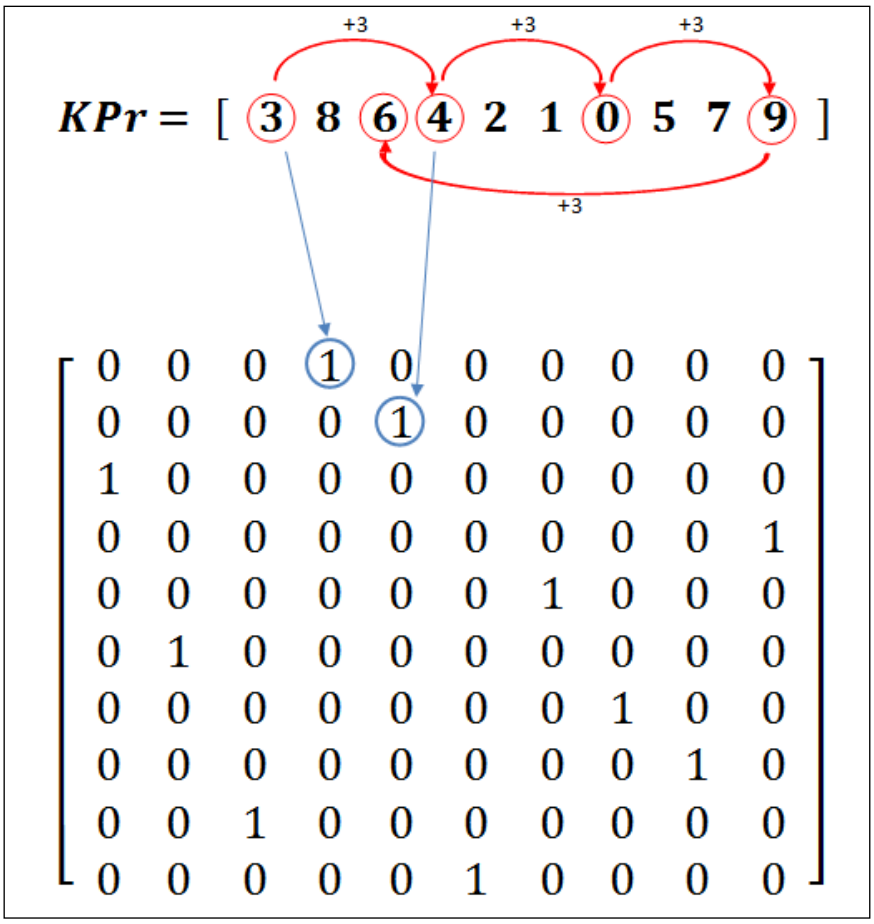


Figura 5.19

Si applica, infine, il prodotto tra la matrice Unione U e la matrice di permutazione P appena creata:

$F = U\,P$

ottenendo, così, la matrice di fusione *F*.

La matrice di fusione *F* ottenuta verrà scomposta e concatenata secondo le righe per poter costruire il vettore di output *V* , cioè, in base alla modalità scelta, *Hybrid Finger Code* o *Hybrid New Key*.

L'output ricavato è strettamente legato alla *Private Key*, all'impronta o al numero pseudo-casuale ed al *Module*.

Le informazioni contenute all'interno appaiono casuali all'utente esterno, ottenendo così l'obiettivo desiderato.

5.10 Applicativo SIIFAE

Viene mostrato, di seguito, come appare all'utente l'interfaccia grafica dell'applicazione SIIFAE.

La schermata iniziale, [Figura 5.20], presenta inizialmente la scelta della modalità di utilizzo:

- *Authentication*;

- *Encryption*.

Una volta effettuate la scelta, si attiverà la tabella a destra relativa alla modalità selezionata, che permette l'inserimento dei dati.

Completato l'inserimento degli input relativi alla modalità è possibile far partire l'applicazione mediante il *button* **Run**.

Dopo l'esecuzione degli algoritmi visti precedentemente viene visualizzato l'output, *Hybrid Finger Code* o *Hybrid New Key*, nella *textbox* in basso.

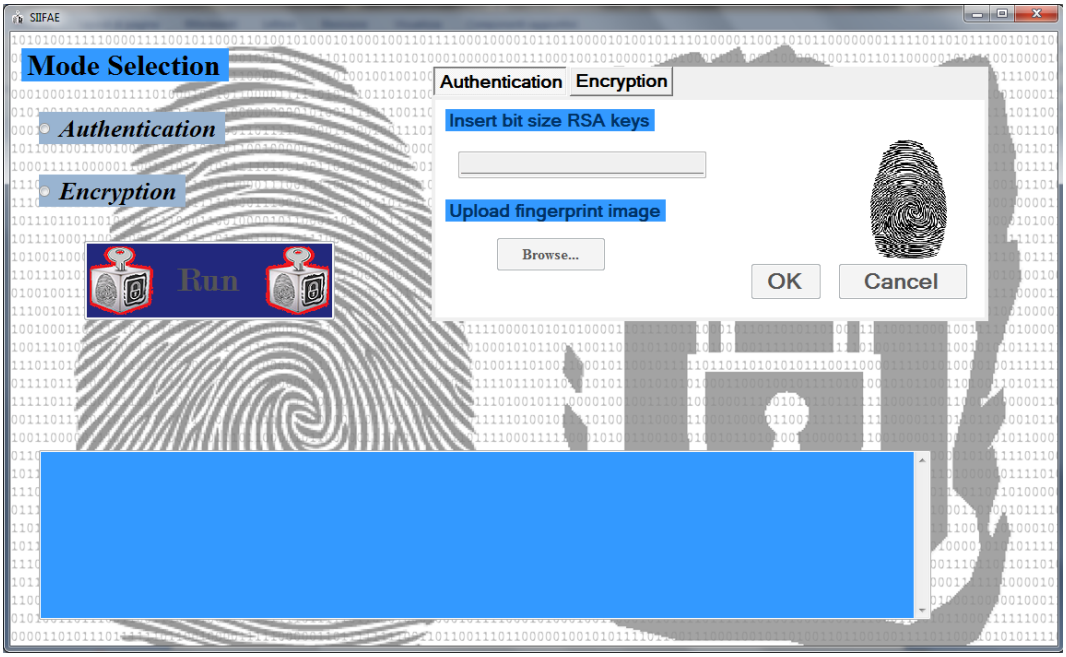


Figura 5.20: Schermata iniziale

Effettuata la scelta della modalità, è possibile inserire gli input relativi.

In Figura 5.21 , viene visualizzata la modalità *Authentication*.

Viene, quindi, richiesta la dimensione in bit della chiave del RSA da utilizzare ed il caricamento dell'immagine dell'impronta.

Inseriti i dati vengono verificati attraverso il *button* **OK**.

Viene visualizzato un messaggio che ricorda i dati inseriti e l'impronta caricata per permettere all'utente la verifica.

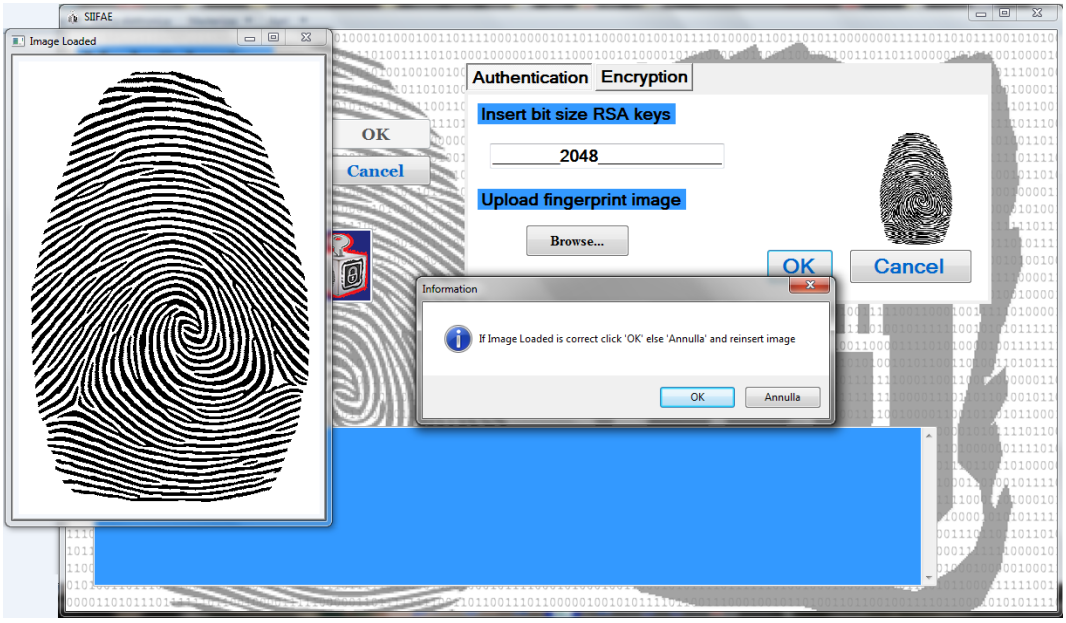


Figura 5.21: Inserimento dati

Verificati tutti i dati inseriti, cliccando sul *button* **Run** viene avviata l'applicazione. In [Figura 5.22], viene focalizzato uno step di esecuzione, più precisamente, la conclusione del *Fingerprint Algorithm* con la visualizzazione delle singolarità e delle minuzie sull'impronta.

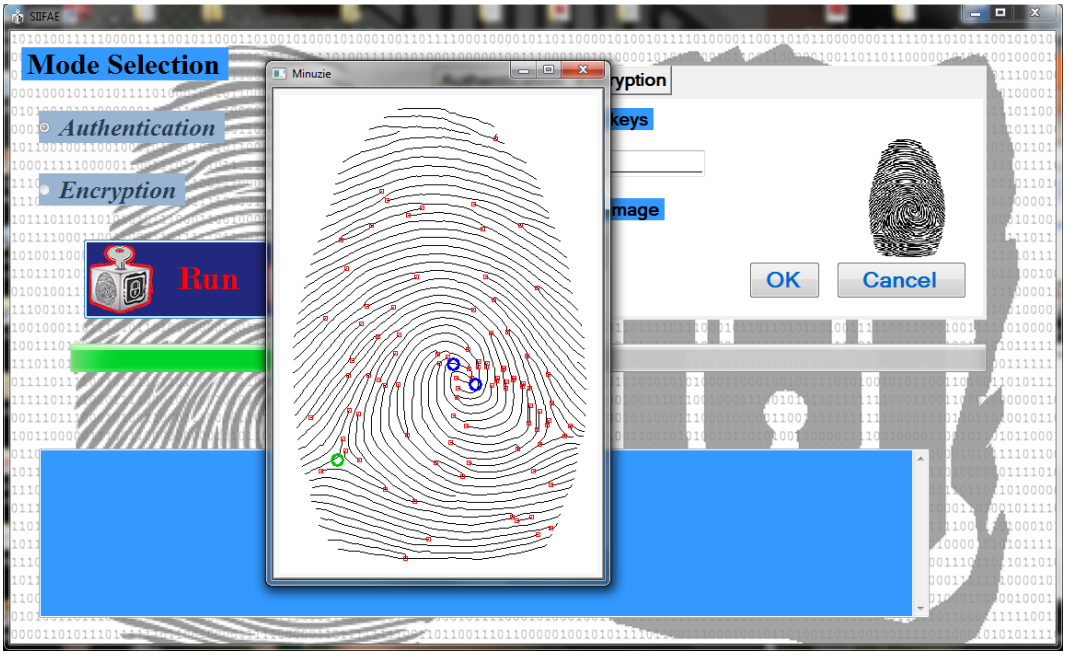


Figura5.22: step di esecuzione

Terminata l'esecuzione dell'applicazione viene riportato il *Hybrid Finger Code* [Figura 5.23]

È anche possibile, cliccando sul *button* **Save to file**, salvare il codice generato.

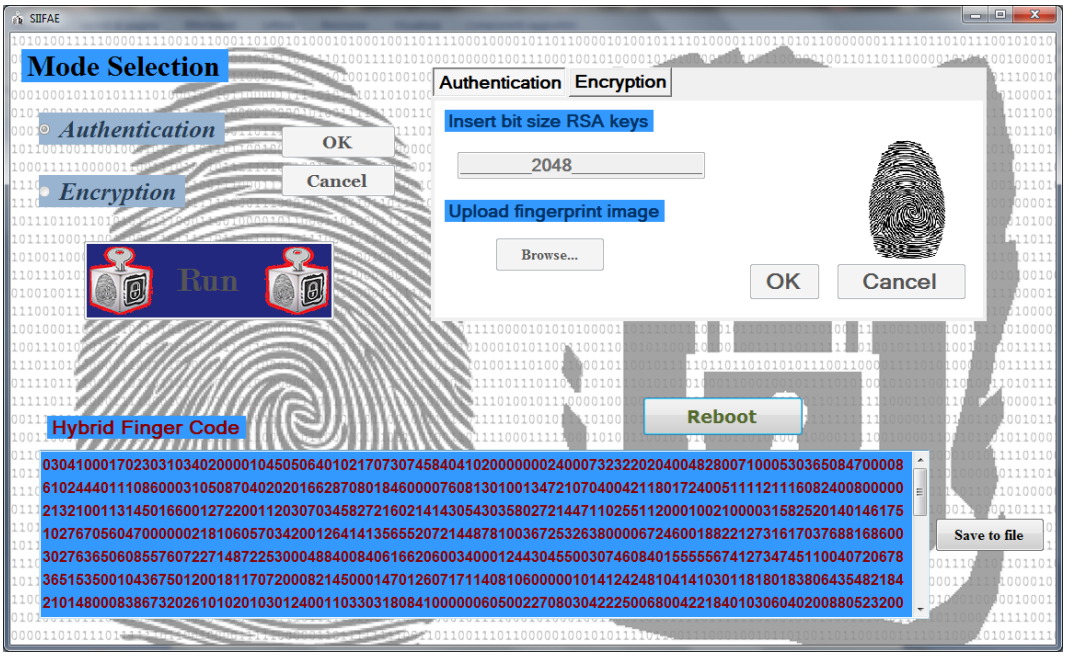


Figura5.23: Fine applicazione

Utilizzando il *button* **Reboot** vengono cancellati tutti i dati precedenti contenuti in memoria in modo da rendere l'applicativo pronto per un'altra esecuzione.

Scegliendo la modalità di *Encryption* viene aperta la schermata a destra per l'inserimento degli input. Viene richiesto l'inserimento della dimensione in bit della *Hybrid New Key* da creare e la scelta della formula frattale da utilizzare. Come per la precedente modalità viene fatta una verifica dei dati inseriti [Figura 5.24].

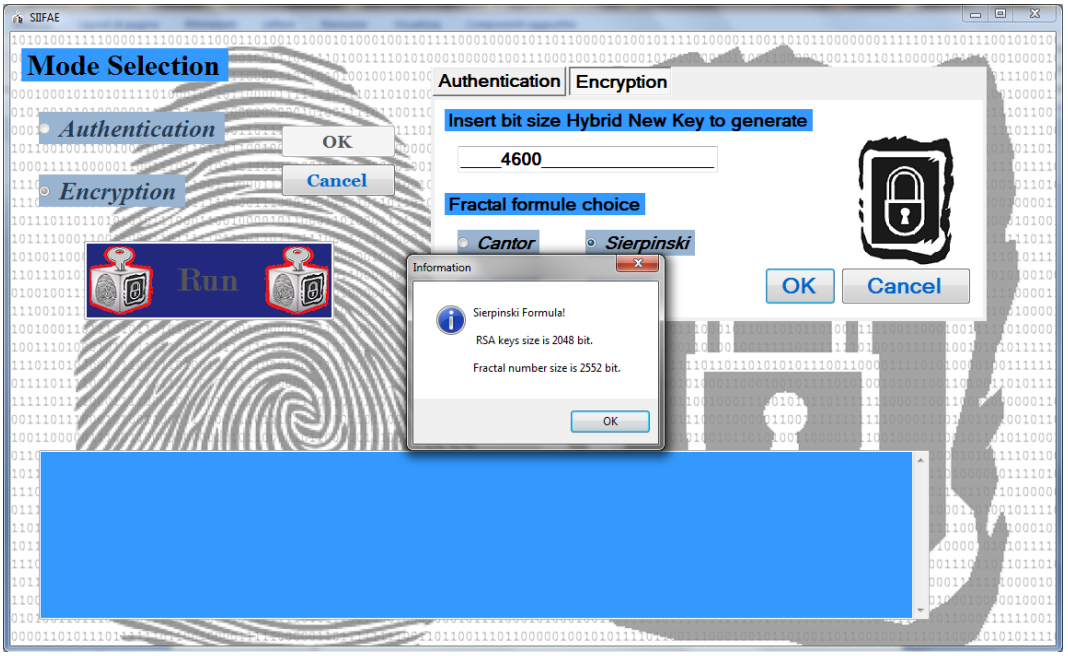


Figura5.24:Inserimento input seconda modalità

Verificato il corretto inserimento dei dati è possibile avviare l'esecuzione con il *button* **Run**. Nella figura 5.25 viene riportata la fase di esecuzione dell'applicazione.

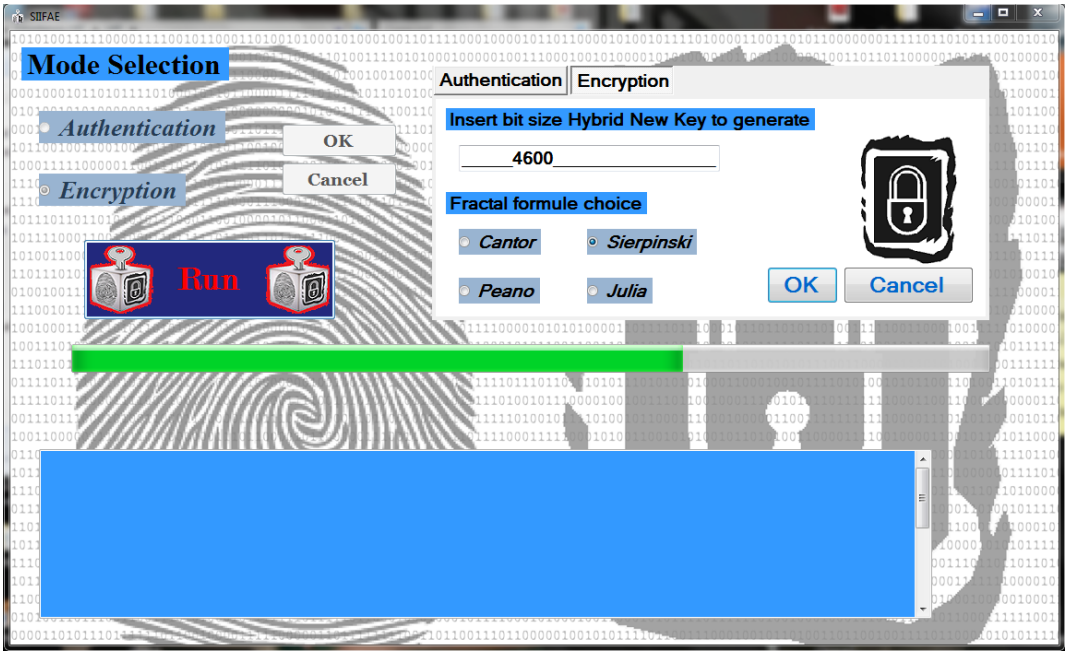


Figura5.25: in esecuzione

Terminata l'esecuzione dell'applicazione viene visualizzata la *Hybrid New Key*. Come nel caso precedente, è possibile salvare la chiave e reinizializzare l'infrastruttura **SIIFAE** [Figura 5.26].

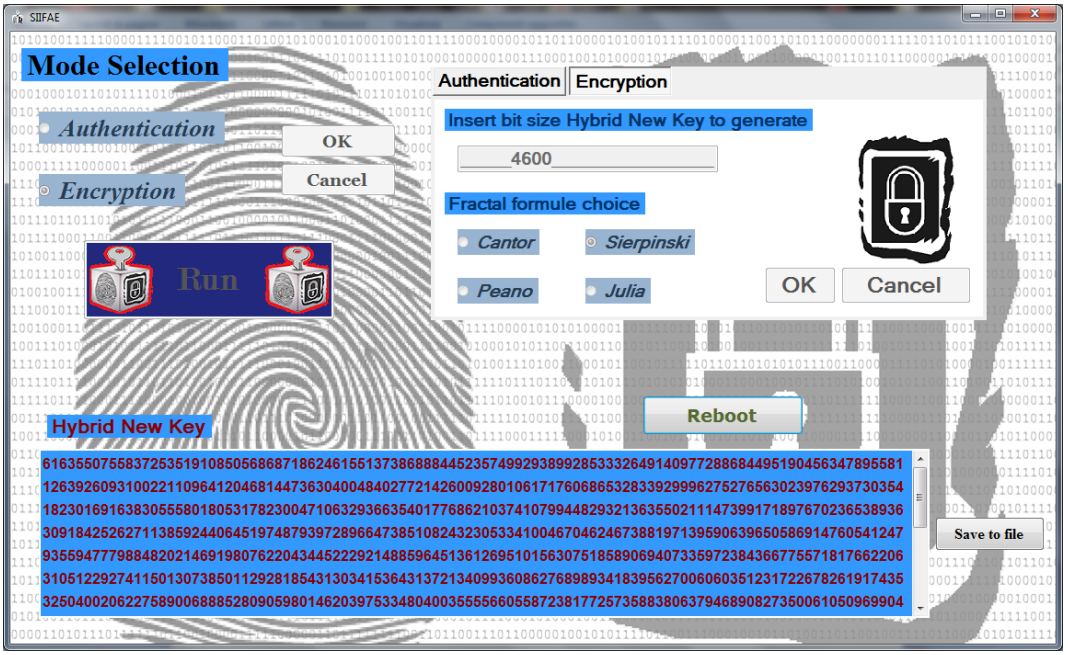


Figura5.26:fine esecuzione

CAPITOLO 6

Fractal & Numerical Information Fusion

6.1 Introduzione

In questo capitolo viene mostrato un sistema per realizzare un codice crittografico, che sia utilizzabile per garantire un alto livello di segretezza usando tecniche di Information Fusion (IF). Nel dettaglio, si è deciso di fondere due codici; uno generato tramite un algoritmo di crittografia a chiave pubblica e uno creato mediante l'utilizzo dei frattali in precedenza analizzati. La tecnica di Information Fusion utilizzata è stata presentata nel precedente capitolo con un differente utilizzo ed opportune modifiche. Infatti, nel precedente lavoro, veniva creato un codice di accesso identificativo, mentre, in questo, viene generata una chiave crittografica fortemente casuale da utilizzare in ambito di cifratura. La tecnica di fusione modificata è denominata F&NIF (Fractal & Numerical Information Fusion). In pratica, la tecnica proposta precedentemente generava dei codici identificativi ad alta sicurezza, atti all'autenticazione, che fondevano un codice biometrico (Finger Code) ed un codice numerico basato sulla primalità (Modulo RSA). In questo capitolo, invece, si vuole creare una chiave crittografia fondendo una componente numerica basata sulla primalità (Modulo RSA) ed una componente numerica random generata tramite i frattali. Anche in questo caso, il metodo di costruzione del nuovo codice (chiave) generato dall'algoritmo di fusione, sarà dipendente dalla chiave privata dell'algoritmo RSA.

6.2 Infrastruttura

Viene di seguito riportato lo schema descrittivo [Figure 6.1] che elenca le fasi principali del sistema che permette di ottenere la fusione dei dati richiesta.

Lo schema è suddiviso in tre parti:

- *Fractal Number Algorithm* : per la generazione del codice numerico basato su uno degli algoritmi Frattali (Julia, Cantor, Sierpinski e Peano);
- *RSA Algorithm*: per la creazione del codice numerico e della chiave privata;
- *F&NIF Algorithm*: per la fusione dei dati.

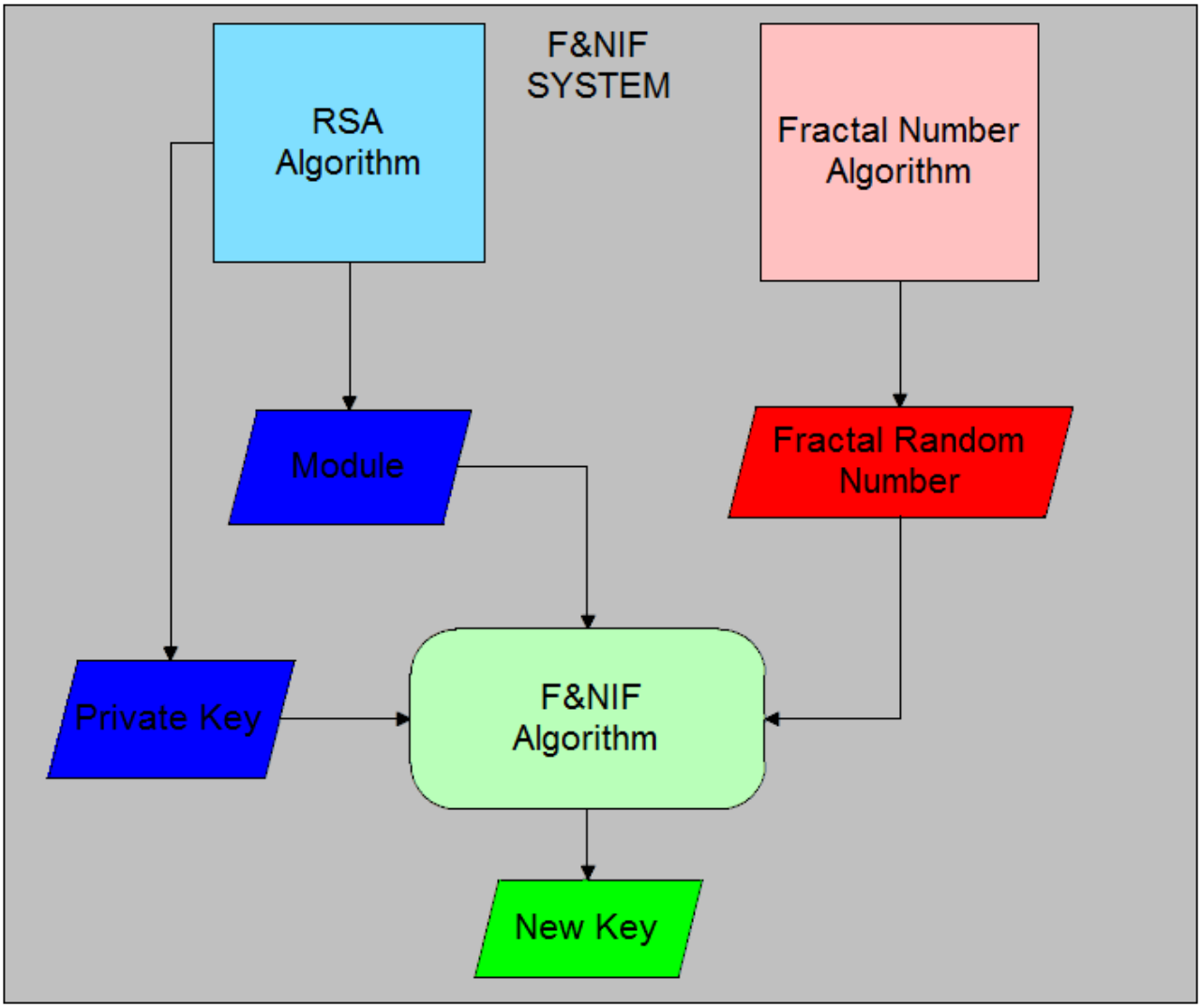


Figure 6.1: Schema a blocchi Sistema F&NIF

Il generatore di codici Frattali (*Fractal Number Algorithm*) comprende le già citate formule di Frattali IFS (Iterated Function System): Cantor, Sierpinski, Peano e Julia.

L'algoritmo di *Information Fusion* (*F&NIF Algorithm*) differisce dall'algoritmo precedente, in quanto, non verrà più generato un codice di autenticazione e vengono apportate forti modifiche sulla costruzione della matrice di permutazione. Per realizzare questo algoritmo ci si è basati sull'idea di rendere, per un utente esterno, il più possibile casuale il metodo di costruzione della nuova chiave crittografica. In particolare, tale metodo, gestisce il modo in cui le componenti dei due array vengono poste all'interno della matrice che andrà a costituire il codice. Per questo motivo, si è scelto di utilizzare la primalità e la segretezza data dalla chiave privata dell'algoritmo RSA. Difatti, si sfrutta il valore della chiave stessa per decidere quanti elementi di ogni singolo array (componente frattale e componente numerica) porre all'interno della matrice unione.

6.3 Analisi dei dati

Attraverso l'algoritmo di generazione di numeri pseudo-casuali derivanti dai frattali si ricava il numero random (Fractal Random Number). Nella Figure 6.2 si può vedere un esempio:

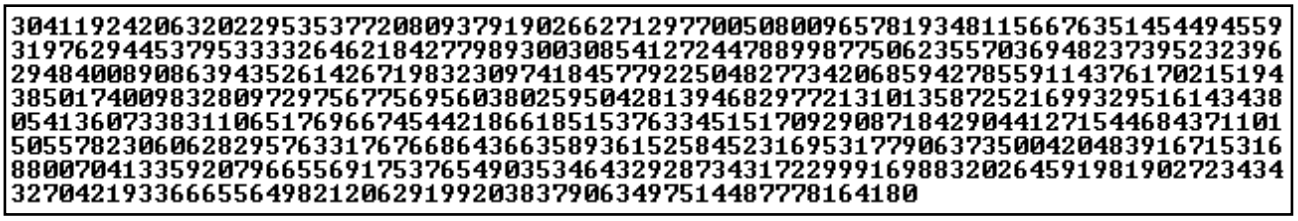


Figure 6.2: Fractal Random Number (2048 bit)

Il prodotto dei due numeri primi (Module) [Figure 6.3] e la chiave privata (Private Key) [Figure 6.4] vengono ricavate attraverso l'algoritmo RSA.

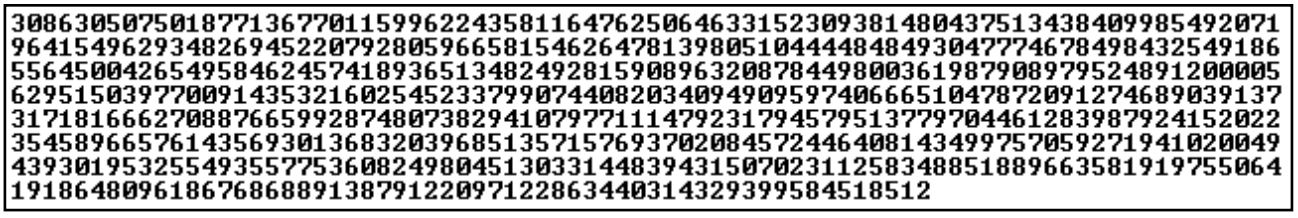


Figura 6.3: Module (2048 bit)

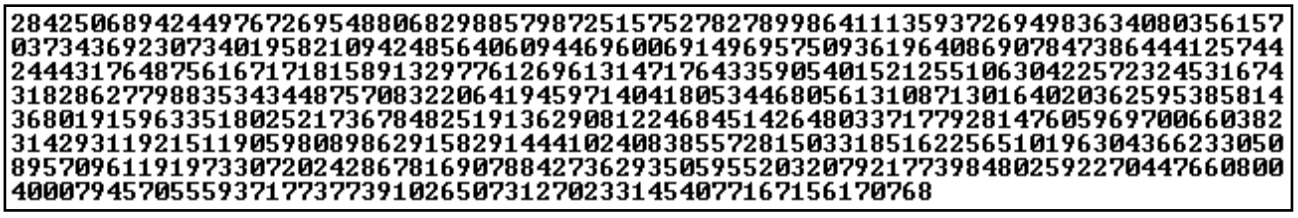


Figura 6.4: Private Key (2048 bit)

Tramite la fusione di questi dati, come spiegato in precedenza, si ottiene una Nuova Chiave di Sicurezza (New Key) [Figure 6.5] che contiene le informazioni precedenti in un ordine che appare, ad un intruso, quanto mai casuale.

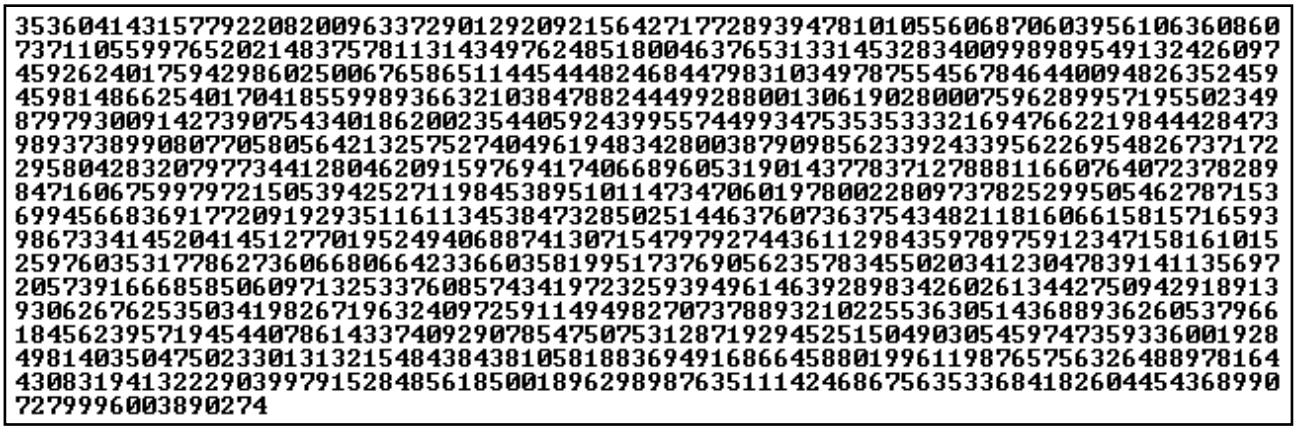


Figura 6.5: New Key (4096 bit)

Per dimostrare la casualità delle chiavi generate, sono state fatte delle analisi statistiche su campioni di 500 codici utilizzando i test statistici del *NIST*.

I test statistici del NIST verificano la randomicità di un sequenza binaria di dimensione fissa.

Questi test sono utilizzati per verificare la qualità di generatori di numeri pseudo-casuali per sistemi crittografici.

Per poter applicare tali test alla Nuova Chiave creata sono state apportate delle modifiche al codice, in quanto tali test possono essere applicate solo a sequenze numeriche in binario.

I test del NIST che sono stati utilizzati sono:

- **Frequency:** Calcola la proporzione di 0 e 1 che sono presenti nella sequenza.
- **Block Frequency**: Calcola la proporzione di 0 e 1 che sono presenti in blocchi da *M* - bit nella sequenza.
- **Cumulative Sums:** Calcola la massima distanza da 0 di un percorso random definito dalla somma cumulativa in cui agli 0 e 1 della sequenza sostituisce i valori -1 e +1.
- **Runs:** Calcola il numero totale di runs nella sequenza, in cui una run è una sequenza ininterrotta di bit identici.
- **Longest Run:** Calcola la lunghezza massima di run di 1 dentro blocchi di *M* – bit nella sequenza

Il risultato che determina la randomicità di una sequenza è il *P-value*, valore compreso tra 0 e 1. In base a tale valore una sequenza viene considerata random quando il *P-value* è maggiore di 0,01.

Nei test effettuati si sono utilizzate come dimensioni della chiave private del RSA e del numero pseudo-random 2048 bit. La chiave risultante è di dimensione pari a 4096 bit.

Per ogni formula di frattale è stato fatto lo stesso test.

I risultati del P-value relativi alla sequenza generata dal Sistema F&NIF sui 500 codici generati utilizzando la formula di Cantor sono riportati in Tabella 61:

Test NIST	***P-value***
Frequency	0.775337
Block Frequency	0.155499
CumulativeSums1	0.042808
CumulativeSums2	0.074791
Runs	0.880145
LongestRun	0.177628

Tabella 6.1: P-value Test NIST (Cantor)

I risultati per la formula di Sierpinsky sono in Tabella 6.2:

Test NIST	***P-value***
Frequency	0.530120
Block Frequency	0.125200
CumulativeSums1	0.599693
CumulativeSums2	0.426272
Runs	0.498313
LongestRun	0.805569

Tabella 6.2: P-value Test NIST (Sierpinski)

Quelli relativi alla formula di Peano sono riportati in Tabella 6.3:

Test NIST	***P-value***
Frequency	0.674543
Block Frequency	0.603841
CumulativeSums1	0.759756

CumulativeSums2	0.262249
Runs	0.157251
LongestRun	0.135720

Tabella 6.3: P-value Test NIST (Peano)

Infine, i risultati ottenuti dalle 500 prove effettuate utilizzando le formule di Julia sono riportate in Tabella 6.4:

Test NIST	***P-value***
Frequency	0.370262
Block Frequency	0.107512
CumulativeSums1	0.498313
CumulativeSums2	0.587274
Runs	0.246750
LongestRun	0.534146

Tabella 6.4: P-value Test NIST (Julia)

Tali risultati di P-value sono tutti superiori a 0,01, questo dimostra che tutte le sequenze di 500 codici generati attraverso le formule dei frattali e RSA di dimensione 4096 bit possono essere considerati una sequenza random.

Per evidenziare l'andamento randomico della sequenza viene riportato il grafico [Figura 6.6] che rimarca maggiormente il divario dei risultati con il limite inferiore di randomicità 0,01.

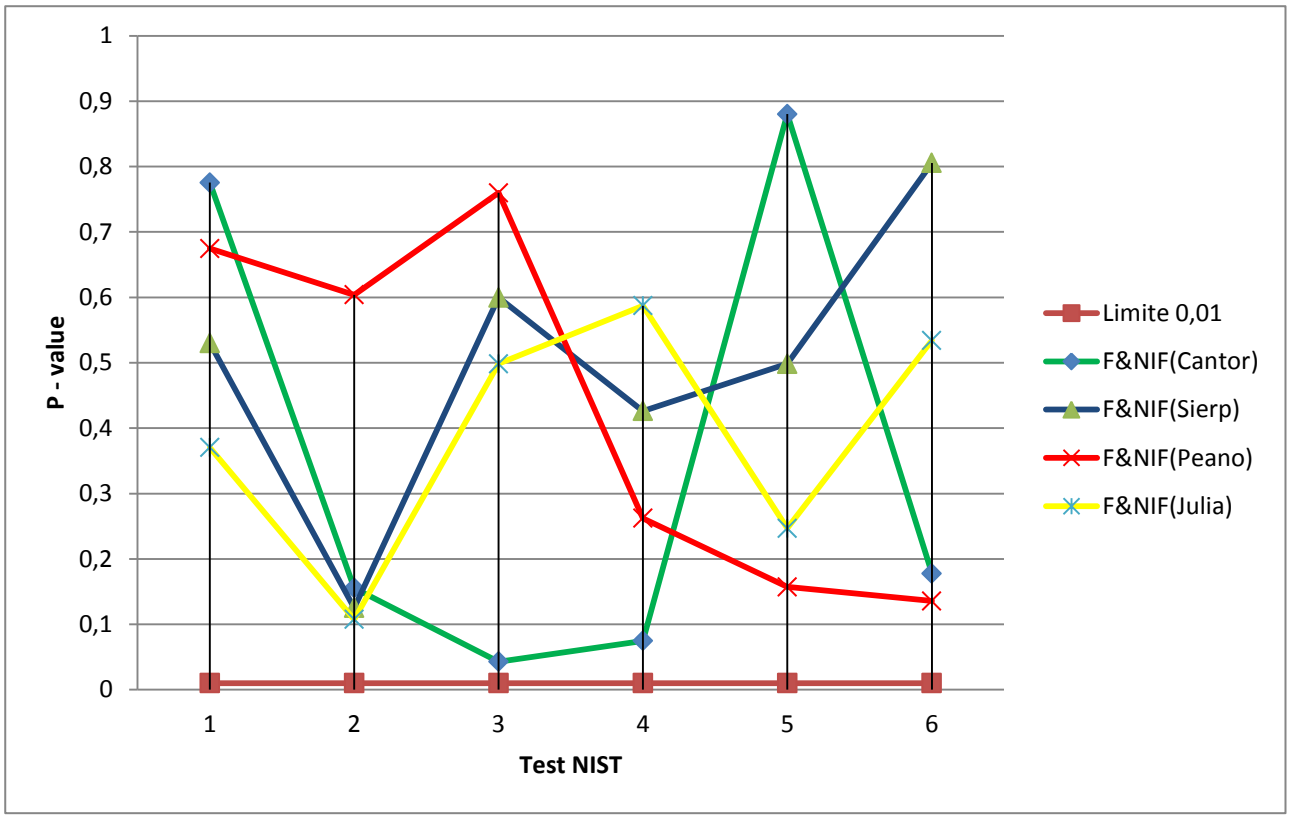


Figure 6.6: P-value

Tale proprietà di randomicità sottolinea la possibilità di utilizzo in ambiti di sicurezza. L'idea potrebbe essere, come detto in precedenza, quella di utilizzare tale chiave casuale per algoritmi di crittografia quali One-Time-Pad.

Conclusioni

L'infrastruttura **SIIFAE** rappresenta un metodo innovativo di considerare la sicurezza informatica nelle sue più ampie sfaccettature. In questo momento storico, il mondo si sta rendendo conto di come siano inappropriate le attuali misure di sicurezza informativa che i Governi e le aziende utilizzano per la protezione delle loro informazioni. Esempi, più noti, di tale situazione sono: la rottura dell'RSA a 1024 bit nel Michigan in USA, il super virus *Stuxnet*, virus in grado di attaccare i sistemi informatici *SCADA* (*Supervisory Control And Data Acquisition*) utilizzati per gestire anche le centrali nucleari e, forse, quello più eclatante, *Wikileaks* che ha reso noto a tutti la fragilità nella protezione dell'informazione ed ha fatto scoprire che la guerra dell'informazione è cominciata. Tutto ciò porta ad essere al passo con i tempi per non essere succubi di attacchi hacking e cracking. La creazione di un metodo che permetta di combinare in maniera sicura e casuale le caratteristiche di autenticazione dell'impronta digitale, la sicurezza del modulo del RSA e la randomicità dei numeri generati attraverso formule innovative dei frattali, evidenzia la svolta che si è deciso di intraprendere nel mondo della security. In letteratura non sono presenti lavori in questa direzione, le formule e le combinazioni dell'information fusion ibrida sono state create opportunamente per

rendere possibile tale combinazione di dati eterogenei e per offrire sia al codice d'autenticazione che alla chiave crittografica la caratteristica randomica necessaria ad applicazioni di Info Security come quella proposta. Inoltre, l'algoritmo innovativo e ibrido di fusione che sta alla base dell'infrastruttura, presenta una filosofia del tutto indipendente sia dalla componente biometrica utilizzata, per quanto riguarda l'autenticazione, sia dalla componente numerica randomica, per quanto concerne la crittografia. I risultati riportati dai test del NIST dimostrano in entrambi i casi l'effettiva randomicità del Hybrid Finger Code e del Hybrid New Key, quindi sottolineando la reale possibilità di utilizzo nella sicurezza informatica. L'idea di unire in un'unica applicazione caratteristiche di autenticazione e crittografia permette un più svariato utilizzo. Una possibile applicazione di tale tecnica di fusione dei dati è legata all'autenticazione della persona, ad esempio, per garantire l'accesso ad aree riservate, a documentazioni classificate e confidenziali, privilegi per azionamenti di infrastrutture militari o di difesa ecc. Inoltre, la possibilità di costruire chiavi di grandezza differente e, soprattutto, sequenze numeriche random, offre alla modalità crittografica una applicazione pratica in algoritmi crittografici quali One-Time-Pad, ad oggi, algoritmo crittografico *perfetto*. Le applicazioni possono essere svariate: dal cifrare documentazioni classificate e confidenziali a proteggere dati di tipo riservato o militare. L'idea è quella di portare avanti questo campo di ricerca, magari utilizzando altri codici biometrici derivanti dalla biometria del volto, problema ancora aperto, ideare nuove tecniche di Information Fusion ibrida applicabili non solo alla crittografia o all'autenticazione ma anche alla steganografia.

Appendice 1 (Watermarking)

Dopo l’avvento di Internet, gli utenti hanno a disposizione un numero sempre maggiore d'informazioni in formato digitale. Il proprietario si potrebbe trovare di fronte a copie illegali che metterebbero in discussione la paternità dei propri documenti. Quindi è sorta l’esigenza di una infrastruttura che riesca a garantire la protezione globale dei dati in modo tale che i diritti di produttori e consumatori di informazioni siano tutelati. In informatica, il termine ***watermarking*** si riferisce all'inclusione di informazioni all'interno di un file, che può essere successivamente rilevato o estratto per trarre informazioni sulla sua provenienza. Tali informazioni, lasciano il documento accessibile ma contrassegnato in modo permanente. Esse possono essere in sovraimpressione, quindi evidenti, o nascoste all'interno del file. La tecnica del watermarking può essere utilizzata con diversi scopi: rendere manifesto a tutti gli utenti chi sia il proprietario del documento (nel caso in cui il marchio sia visibile); dimostrare l'originalità di un documento non contraffatto; evitare la distribuzione di copie non autorizzate. Anche se il digital watermarking è una tecnica recente, il watermark è conosciuto sin dal XIII secolo. A Bologna nel 1282 venne usata una tecnica dai produttori di carta per identificare il proprio prodotto, ma anche per francobolli e altri documenti amministrativi, per scoraggiarne la contraffazione. I watermark invisibili non sono percettibili dall'occhio umano sotto le normali condizioni visive. Essi sono maggiormente d'aiuto nell' individuare e perseguire, piuttosto che nello scoraggiare, un eventuale ladro. Sono costituiti da un’immagine sovrimpressa che non può essere vista ma che può essere individuato algoritmicamente. Un watermark è detto **fragile** se viene distrutto, quando l’immagine è manipolata digitalmente, in un qualunque modo utile a provare l’autenticità dell’immagine. Se il watermark si presenta ancora intatto, significa che questa non è stata modificata. Queste caratteristiche possono essere importanti per ammettere le immagini digitali come prova in ambito giudiziario. I watermark **semifragili** sono progettati in modo da andare distrutti in seguito a qualsiasi cambiamento che superi una certa soglia specificata dall'utente: una soglia zero individua perciò un watermark fragile. Un watermark **robusto** resiste invece alle comuni operazioni e trasformazioni dei dati ad es. nel caso di immagini digitali al filtraggio, alla compressione, al resizing ecc.,utili nei contesti in cui la proprietà deve essere provata o garantita. La creazione di sportelli elettronici on line e a distanza comporta uno snellimento ed una riorganizzazione che migliora la qualità dei servizi offerti. L’interesse della Pubblica Amministrazione (PA) per l’autenticazione e il controllo degli accessi ad applicazioni informatiche critiche e a dati sensibili da parte dei fruitori di servizi erogati on-line, si basano su tecnologie Biometriche. Più in generale nell’ambito dell’autenticazione e del riconoscimento nei sistemi informatici si utilizzano caratteristiche fisiche uniche e non riproducibili. Le tecniche che agiscono sul dominio spaziale operano variazioni sui pixel che costituiscono l’immagine.

g(x,y) = T[f(x,y)] dove la f(x,y)] è l’immagine da modificare, g(x,y) è l’immagine modificata tramite l’operatore di trasformazione T il quale agisce su f ed è definito in un intorno del pixel (x,y) in esame.

Le tecniche che agiscono sul dominio delle frequenze si basano sulle modifiche di una trasformata nel dominio delle frequenze. Le più usate sono la trasformata di Fourier, del Coseno discreta o Wavelet dell’immagine che si considera

1. Calcolare la trasformata portando l'immagine nel dominio delle frequenze;

2. Convolvere il risultato per un filtro particolare;

3. Effettuare la trasformata inversa per tornare al dominio spaziale.

Presentiamo un ipotesi di sistema per attribuire la proprietà intellettuale di un file grazie all'utilizzo di tecniche avanzate di Information Fusion (IF) in spazi trasformati biometric based. L'IF è diversa a seconda che si tratti di un

a) **file testuale;**

b) **immagini ;**

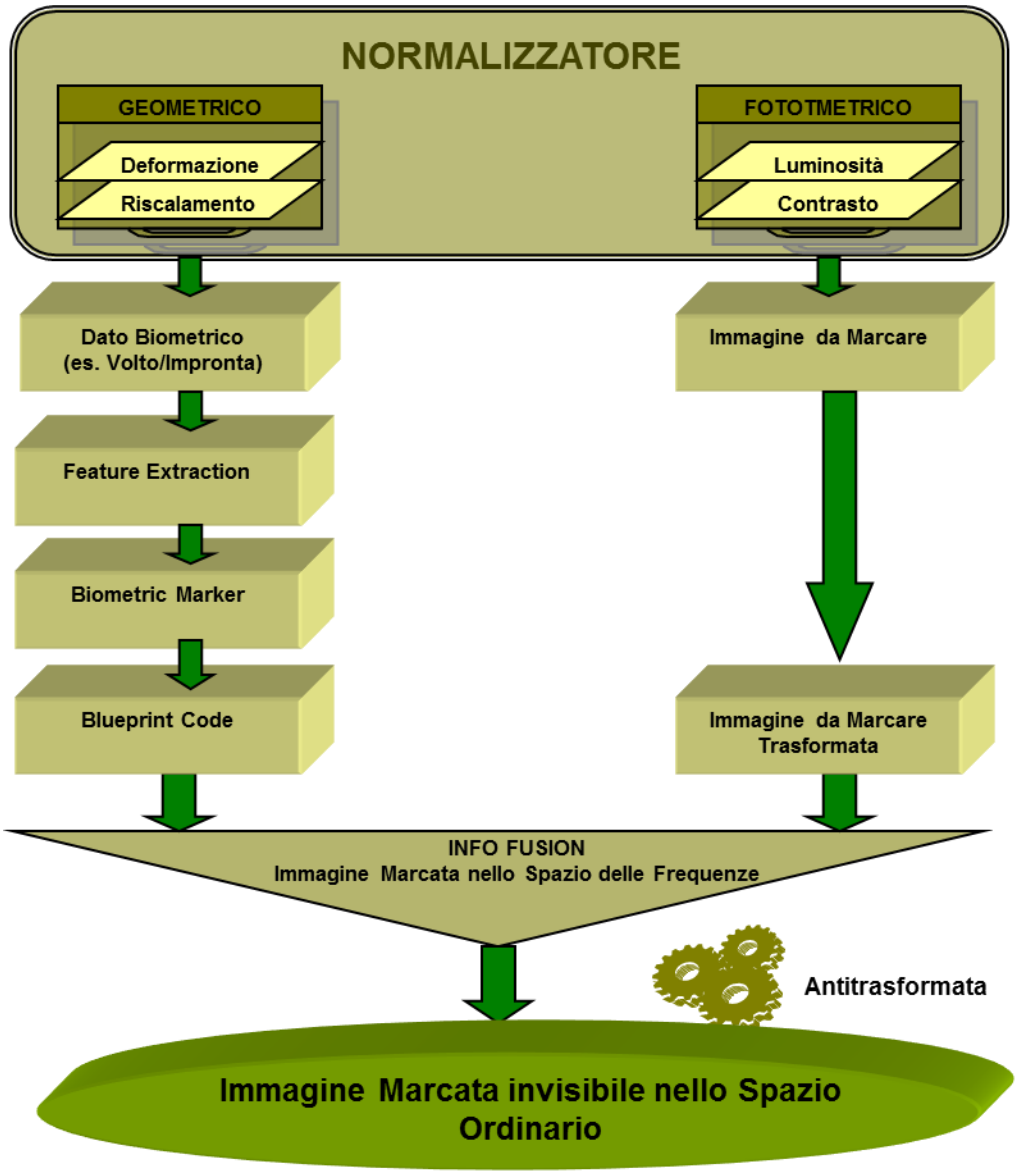
c) **multimedia;**

Si procede alla creazione di "blueprint code" (Marker Biometrico) con diversi meccanismi di watermarking basati rispettivamente

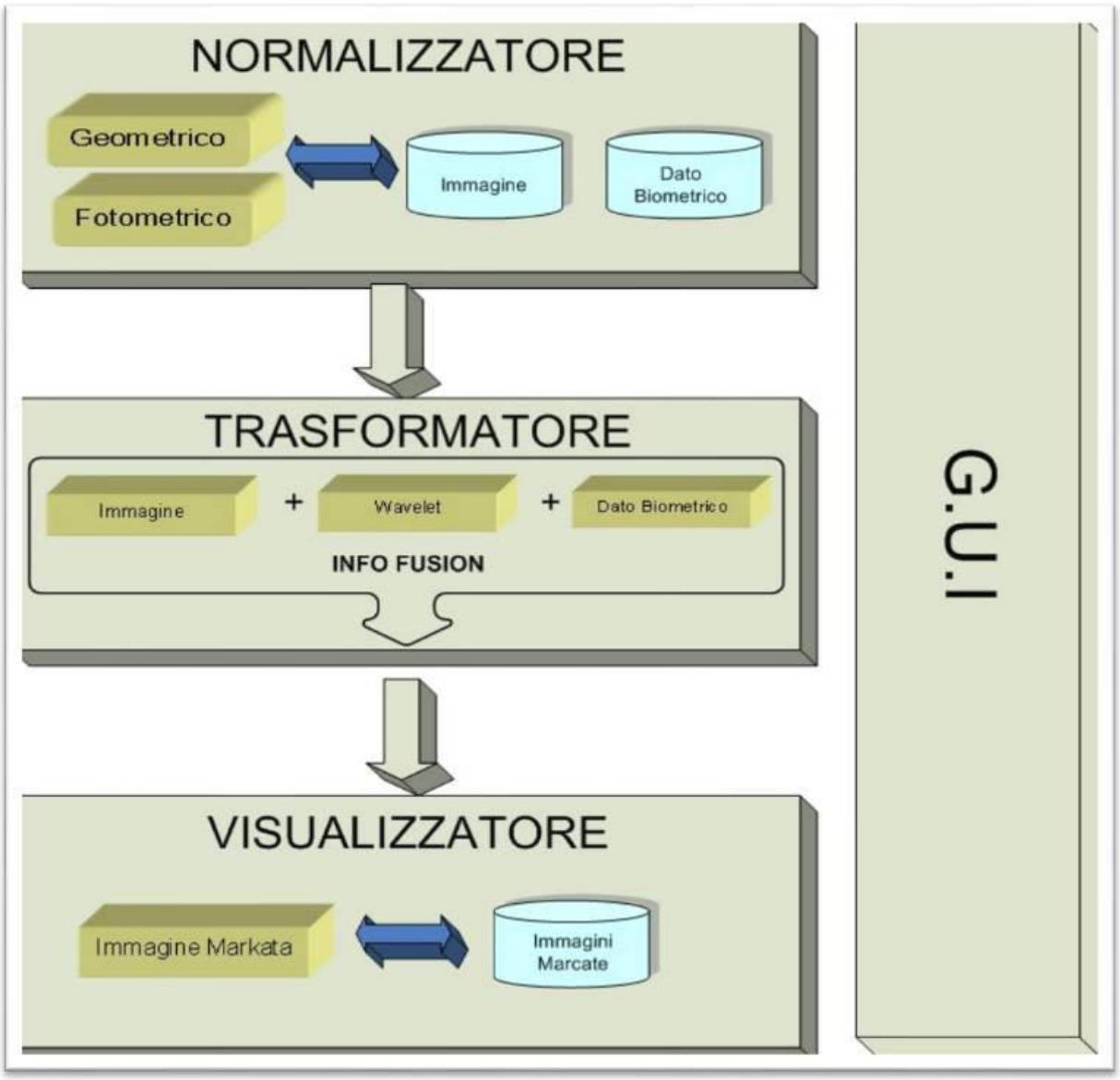
a) **wavelet;**

b) **multiwavelet;**

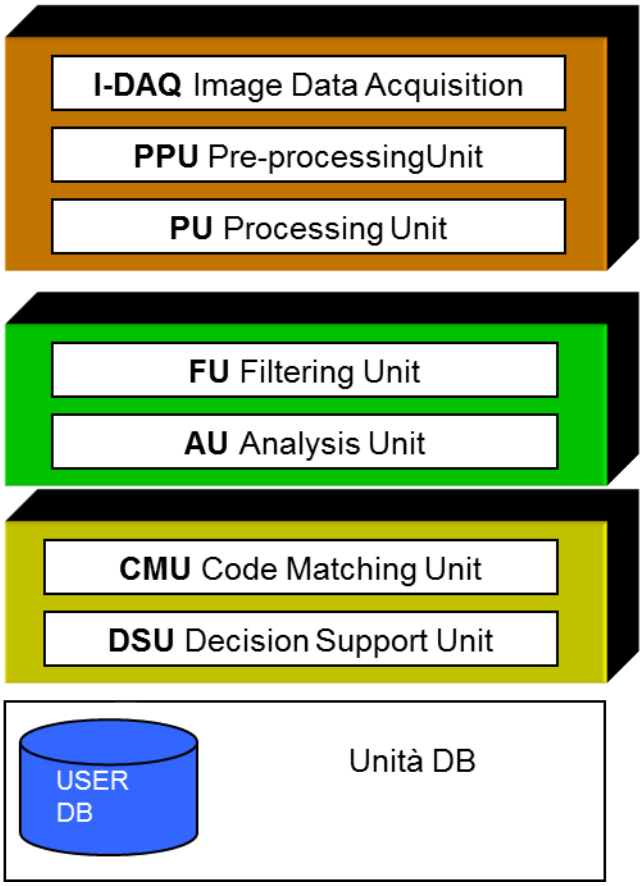
c) **packet wavelet.**

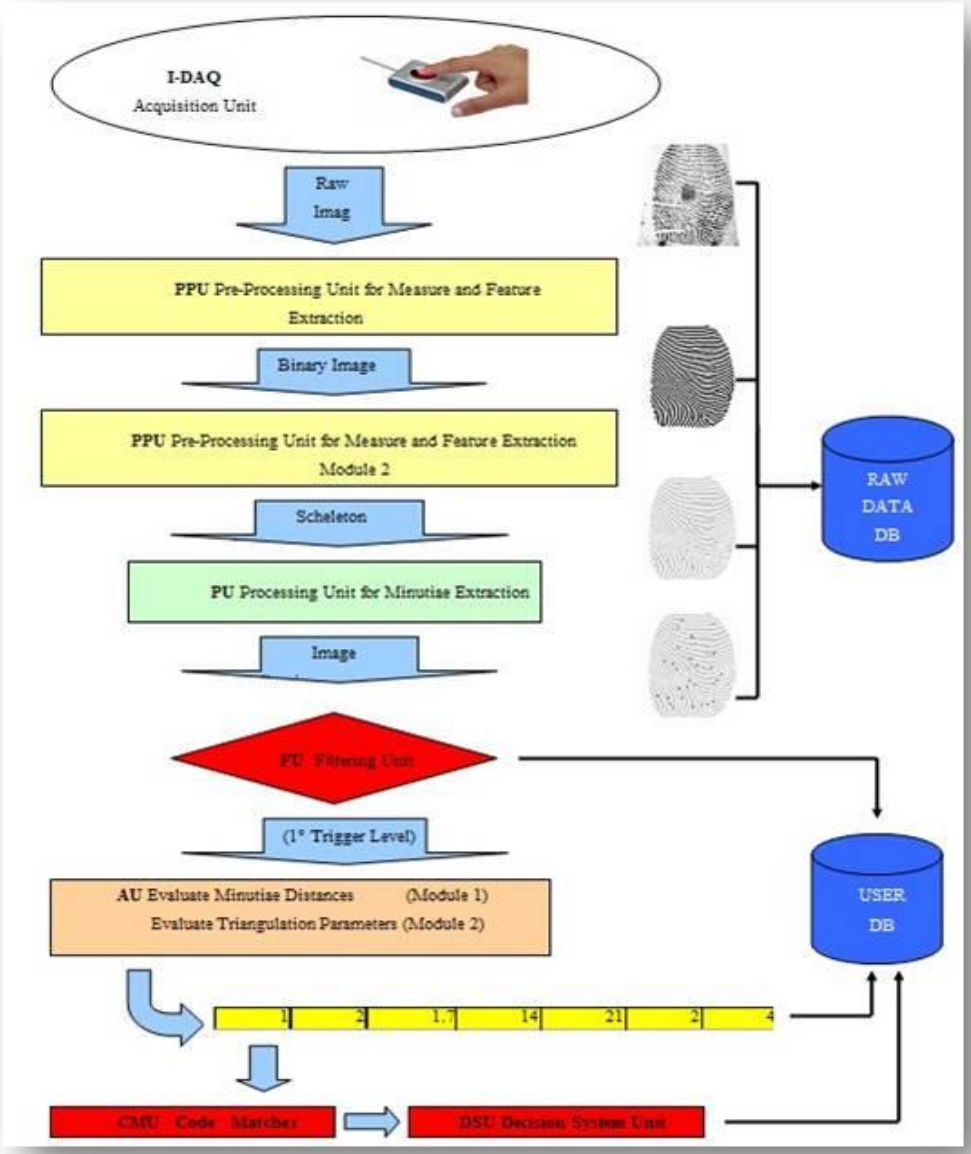


L’architettura logica è la seguente



Le unità che costituiscono la componente architetturale


I-DAQ Image Data Acquisition
PPU Pre-processingUnit
PU Processing Unit
FU Filtering Unit
AU Analysis Unit
CMU Code Matching Unit
DSU Decision Support Unit
USER DB
Unità DB



a) Il sistema risponderà al requisito di attribuzione della Proprietà Intellettuale grazie al'Info Fusion in spazi trasformati con l'utilizzo di biometric data.
b) Grazie al reverse Engineering il sistema potrà essere utile alla tracciabilità nella PA o Aziende in cui è rilevante tracciare l'accesso alle informazioni da parte di operatori/utenti diversi.

APPENDICE 2 (PDE Surface)

Nel contesto della face recognition e reconstruction sono stati studiati i più avanzati modelli basati su tecniche FEM e VEM. Diversi sono stati i riscontri relativi all'ambito della sicurezza, dell'animazione (computer graphics) o della realtà virtuale. Obiettivo primario è l'adattamento al contesto sicurezza. Eyad Elyan and Hassan Ugail, nel 2007 hanno proposto un modello basato su PDE e finalizzato alla ricostruzione della geometria di una immagine. Essa viene intesa come un insieme di surface patches, ognuna risultante dalla integrazione di una PDE e dall'utilizzo di opportune boundary curves identificate tramite scansioni tridimensionali del volto. Una PDE surface è intesa come una superficie parametrica X(u; v) funzione dei due parametri u e v e definita su un dominio limitato , avendo specificato i "boundary data" sul contorno della frontiera.

Assumendo $\{\Omega: 0 \leq u \leq 1{,}0 \leq v \leq 2\pi\}$ si considerano le curve

$$X(0,v) = P_0(v); X(s,v) = P_s(v); X(t,v) = P_t(v); X(1,v) = P_1(v)$$

dove $P_0(v)$ e $P_1(v)$ definiscono i contorni della superficie a u= 0 e u = 1. $P_s(v)$e $P_t(v)$ rappresentano le curve intermedie. Ricorrendo a tale formulazione, la soluzione analitica è

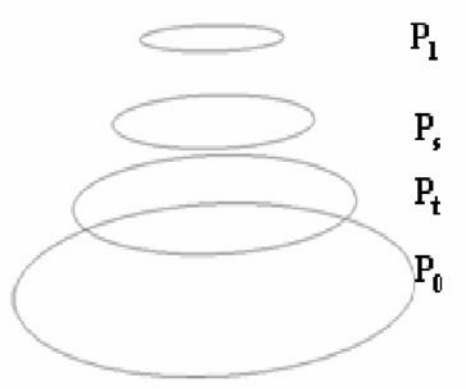
$$X(u,v) = \hat{X}(u)\cos(nv) + \hat{X}(u)\sin(nv) + R(u,v)$$

Con n intero ed R(u,v) che viene computata con metodi di approssimazione spettrale.

La superficie è vista come soluzione della seguente equazione alle derivate parziali

$$\frac{\partial^4 X}{\partial u^4} + \frac{\partial^4 X}{\partial u^2 \partial v^2} + \frac{\partial^4 X}{\partial v^4}$$

La figura in alto è generata dall’equazione con le sottostanti condizioni al contorno



Ovviamente è impensabile sperare di ottenere una rappresentazione analitica di un volto tramite condizioni al contorno da inserire nell’equazione, però potrebbe essere interessante provare a dividere il volto in tante sotto parti di cui cercare di ottenere una rappresentazione analitica. Maggiori saranno le divisioni è più il risultato è preciso. Una volta ottenute le relazioni analitiche, potrebbero studiarsi le proprietà metriche tramite la geometria differenziale. Incollando i risultati, potremmo avere una mappatura continua di un volto tramite la curvatura gaussiana o tramite ad esempio i simboli di Christoffell.

BIBLIOGRAFIA

[1] First international competition for fingerprint verification algorithms,FVC2000, http://bias.csr.unibo.it/fvc2000/default.asp.

[2] Federal information processing standards publication 180-2, SECURE HASH STANDARD, 2002.

[3] J. Fierrez Aguilar, J. Ortega Garcia, D. Garcia Romero, and J. GonzalezRodriguez. A comparative evaluation of fusion strategies for multimodal biometric verification. In *AVBPA03*, pages 830–837, 2003.

[4] M.A. Alia and A.B. Samsudin. A new digital signature scheme based on mandelbrot and julia fractal sets. *American Journal of Applied Sciences*, 4(11):848 – 856, 2007.

[5] M. Barnsley. *Fractals everywhere*. Academic Press, Boston, 1988.

[6] E. Blasch and S. Plano. Jdl level 5 fusion model: user refinement issues and applications in group tracking. *SPIE Aerosense*, 4729:270–279, 2002.

[7] G. Cantor. *De la puissance des ensembles pareaits de points*. Acta Mathematica 2, 1884.

[8] R. Cappelli, D. Maio, D. Maltoni, and A. Erol. Synthetic fingerprintimage generation. *15th International Conference on Pattern Recognition (ICPR'00)*, 3:3475, 2000.

[9] K.I. Chang, K.W. Bowyer, S. Sarkar, and B. Victor. Comparison and combination of ear and face images in appearance-based biometrics. *PAMI*, 25(9):1160–1165, 2003.

[10] C.H. Chen and C.T. Chu. Fusion of face and iris features for multimodal biometrics. In *ICB06*, pages 571–580, 2006.

[11] G. Chetty, D. Sharma, and B.M. Balachandran. An agent based multifactor biometric security system. In *KES '08: Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part III*, pages 245–251, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] T. Connie, A. Teoh, M. Goh, and D. Ngo. Palmhashing: a novel approach for dual-factor authentication. *Pattern Anal. Appl.*, 7(3):255–268, 2004.

[13] B.V. Dasarathy. Information fusion - what, where, why, when, and how? *Information Fusion*, 2(2):75–76, 2001.

[14] R.L. Devaney. *Caos e Frattali - Matematica dei sistemi dinamici ed applicazioni al calcolatore*. Addison-Wesley Italia, Milano, 1990.

[15] W. Diffie and M.E. Hellman. Multiuser cryptographic techniques. In *AFIPS '76: Proceedings of the June 7-10, 1976, national computer conference and exposition*, pages 109–112, New York, NY, USA, 1976. ACM.

[16] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[17] K.J. Falconer. *Fractal Geometry: Mathematical foundations and Applications*. John Wiley and Sons, New York, 1990.

[18] G. Feng, D. Hu K. Dong, and D. Zhang. When faces are combined with palmprints: A novel biometric fusion strategy. In *BioAW04*, pages 332–341, 2004.

[19] L. Hong and A. Jain. Integrating faces and fingerprints for personal identification. *IEEE transactions on pattern analysis and machine intelligence*, 20:1295–1307, 1997.

[20] G.B. Huntress. *Encryption using Fractal Key*. United States Patent 6782101, 2004.

[21] A.K. Hyder, E. Shakbazian, and E. Waltz, editors. *Multisensor Fusion*. NATO ASI on Multisensor Data Fusion, 2002.

[22] G. Iovane and F.S. Tortoriello. *Frattali e geometria dell'universo*. Aracne Editrice S.R.L., 2005.

[23] A. Jain, S. Prabhakar, and S. Chen. Combining multiple matchers for a high security fingerprint verification system. *Pattern Recognition Letters*, 20:11–13, 1999.

[24] A.T.B. Jin, D.N.C. Ling, and A. Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, 37(11):2245–2255, 2004.

[25] A. Juels, D. Molnar, and D. Wagner. Security and privacy issues in epassports, 2005.

[26] A. Karlsson. Dependable and generic high-level information fusion – methods and algorithms for uncertainty management. Technical Report HS- IKI -TR-07-003, 2007.

[27] A. Kong, K.H. Cheung, D. Zhang, M. Kamel, and J. You. An analysis of biohashing and its variants. *Pattern Recognition*, 39(7):1359–1368, 2006.

[28] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White. Revisiting the jdl data fusion model ii. In *In P. Svensson and J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004)*, pages 1218–1230, 2004.

[29] A. Lumini and L. Nanni. An advanced multi-modal method for human authentication featuring biometrics data and tokenised random numbers. *Neurocomputing*, 69(13-15):1706 – 1710, 2006. Blind Source Separation and Independent Component Analysis - Selected papers from the ICA 2004 meeting, Granada, Spain, Blind Source Separation and Independent Component Analysis.

[30] R.C. Luo, C.C. Yih, and K.L. Su. Multisensor fusion and integration: approaches, applications, and future research directions. *Sensors Journal, IEEE*, 2(2):107–119, 2002.

[31] J.F. Riendeau M. Ciet, A.J. Farrugia. *Method and apparatus for data protection system using geometry of fractals or other chaotic systems*. United States Patent Application Publication 20100031039, 2010.

[32] B.B. Mandelbrot. *The fractal geometry of nature*. W.H. Freeman and Company, New York, 1982.

[33] B.B. Mandelbrot. *Gli oggetti frattali. Forma, caso e dimensione*. Giulio Einaudi Editore, Torino, 1987.

[34] B.B. Mandelbrot. *Nel mondo dei frattali*. Di Renzo Editore, Collana I Dialoghi, 2001.

[35] E.F. Nakamura, A.A.F. Loureiro, and A.C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.*, 39(3):9, 2007.

[36] E.F. Nakamura, F.G. Nakamura, C.M.S. Figueiredo, and A.A.F. Loureiro. Using information fusion to assist data dissemination in wireless sensor networks. *Telecommunication Systems*, 30(1):237–254, 2005.

[37] L. Nanni and A. Lumini. A multi-modal method based on the competitors of fvc2004 and on palm data combined with tokenised random numbers. *Pattern Recognition Letters*, 29(9):1344–1350, 2008.

[38] M. Nappi and D. Riccio Moderne tecniche di elaborazione di immagini e biometria

[39] Department of Defence. Data fusion subpanel of the joint directors of laboratories, technical panel for c3. data fusion lexicon, 1991.

[40] Department of Defence. Dsto (defence science and technology organization) data fusion special interest group. data fusion lexicon, 1994.

[41] B. Olsson, P. Nilsson, B. Gawronska, A. Persson, T. Ziemke, and S.F. Andler. An information fusion approach to controlling complexity in bioinformatics research. In *CSBW '05: Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference - Workshops*, pages 299–304, Washington, DC, USA, 2005. IEEE Computer Society.

[42] G. Peano. Sur une courbe qui remplit toute une aire plaine. *Math.Annalen*, 36:157 – 160, 1890.

[43] N. Poh and J. Korczak. Hybrid biometric person authentication using face and voice features. In *In Proc. AVBPA*, pages 348–353, 2001.

[44] C. Pohl and J. L. Van Genderen. Multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, 19(5):823–854, 1998.

[45] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[46] A. Ross and A. Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24(13):2115 – 2125, 2003. Audio- and Video-based Biometric Person Authentication (AVBPA 2001).

[47] A. Ross, A. Jain, and J.Z. Qian. Information fusion in biometrics. In *AVBPA '01: Proceedings of the Third International Conference on Audioand Video-Based Biometric Person Authentication*, pages 354–359, London, UK, 2001. Springer-Verlag.

[48] V. Rozouvan. Modulo image encryption with fractal keys. *Optics and Lasers in Engineering*, 47(1):1 – 6, 2009.

[49] N. Ruggeri. Principles of pseudo-random number generation in cryptography, 2006.

[50] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST(National Istitute of Standards and Technology), 2008.

[51] P. Smart. Knowledge-intensive fusion for improved situational awareness. 2005.

[52] A. Steinberg and C. Bowman. Rethinking the jdl data fusion levels. In *NSSDF Conference Proceedings*. JHAPL, 2004.

[53] A. Steinberg, C. Bowman., and F. White. Revisions to the jdl data fusion model. In *Joint NATO/IRIS*, 1998.

[54] A. Steinberg, C. Bowman., and F. White. Revisions to the jdl data fusion model. In *Sensor Fusion: Architectures, Algorithms, and Applications III*, 1999.

[55] Q. Tao and R. Veldhuis. Hybrid fusion for biometrics: Combining score level and decision-level fusion. *Computer Vision and Pattern Recognition Workshop*, pages 1–6, 2008.

[56] K.A. Toh, W.Y. Yau, E. Lim, L. Chen, and C.H. Ng. Fusion of auxiliary information for multi-modal biometrics authentication. In *ICBA*, pages 678–685, 2004.

[57] O. Ushmaev and S. Novikov. Biometric fusion: Robust approach. In *Authentication (MMUA). Workshop on Multimodal User*, 2006.

[58] L. Wald. Some terms of reference in data fusion. *IEEE Transactions on Geosciences and Remote Sensing*, 37:1190–1193, 1999.

[59] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House, Inc., 1990.

[60] F. Yang, B. Ma, Q. Wang, and D. Yao. Information fusion of biometrics based-on fingerprint, hand-geometry and palm-print. In *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on*, pages 247–252, 2007.

[61] D. Zhang and A. Jain. *Advances in Biometrics: International Conference, ICB 2006, Hong Kong, China, January 5-7, 2006, Proceedings (Lecture Notes in Computer Science)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[62] A. Pellegrini, V. Bertacco, and T.M. Austin. Fault-based attack of rsa authentication. In *Design, Automation and Test in Europe(DATE)*, pages 855–860, 2010.

[63] P.M.R. Anand, G. Bajpai, and V. Bhaskar. Real-time symmetric cryptography using quaternion julia set. *IJCSNS International Journal of Computer Science and Network Security*, 9(3):20 – 26, 2009.

[64] E.J. Yoon and K.Y. Yoo. Cryptanalysis of a modulo image encryption scheme with fractal keys. *Optics and Lasers in Engineering*, 48(7-8):821 – 826, 2010.

[65] S. Lian, X. Chen, and D. Ye. Secure fractal image coding based on fractal parameter encryption. *Fractal Complex Geometry, Pattern, Scaling in Nature and Society*, 17(2):149 – 160, 2009.

[66] N.M.G. Al-Saidi and M.R.Md. Said. A new approach in cryptographic systems using fractal image coding. *Journal of Mathematics and Statistics*, 5(3):183 – 189, 2009.

[67] www.frattali.it

[68] P. Davies .Il cosmo intelligente Mondadori 1989