Scuola Dottorale di Ingegneria
Sezione di Ingegneria dell'Elettronica Biomedica,
dell'Elettromagnetismo e delle Telecomunicazioni

TESI DI DOTTORATO

GESTIONE DELLA SICUREZZA NELLE COMUNICAZIONI RADIO DI ULTIMA GENERAZIONE

SECURITY MANAGEMENT IN LAST GENERATION RADIO COMMUNICATIONS

Candidato
Daniele Blasi

Docente guida
Prof. Alessandro Neri

Roma, 1 marzo 2009

Indice

1 Introduzione ---------- 4

1.2 Insiemi di chiavi segrete ---------- 6

1.2.1 Cardinalità delle chiavi ---------- 7

1.3 Struttura del lavoro di tesi ---------- 7

2 Sistemi a Phase-Hopping ---------- 9

2.1 Modulazioni con offset di fase ---------- 9

2.2 Sistemi OFDM a Phase-Hopping ---------- 11

2.2.1 Insiemi di chiavi private associate all'algoritmo ---------- 13

2.2.2 Impatto del phase-hopping sule prestazioni del sistema ---------- 14

2.3 Sistemi Hash PH-OFDM ---------- 14

2.3.1 Autenticazione dell'utente basata su algoritmi di hash ---------- 15

2.3.2 Hash e FEC ---------- 16

2.3.3 Costruzione di Hash "robusti"---------- 19

2.4 Un sistema PH-OFDM basato su hash robusti ---------- 22

3 Turbo Codici ---------- 28

3.1 Turbo Codici su modulazioni *L*-arie ---------- 28

3.2 Dai Turbo Codici alla Turbo Equalizzazione ---------- 37

3.3 Equalizzazione e l'approccio a chiavi di sessione ---------- 41

4 Sicurezza e Codifica di Canale ---------- 45

4.1 Modello matematico ---------- 47

4.1.1 Turbo Codici Permutati ---------- 49

4.2 Analisi delle prestazioni ---------- 52

4.2.1 Analisi delle prestazioni dei Turbo Codici Permutati ---------- 56

4.3 Simulazione del sistema ---------- 65

5 Sistemi TH CSMA Persistenti ---------- 73

5.1 Modello matematico di un TH-CSMA-Slotted-*p*-Persistente -------- 73

5.2 Comportamento del sistema in presenza di attacchi ---------- 78

5.3 Conclusioni ---------- 81

Riferimenti ----------- 82

1 Introduzione

Oggi le comunicazioni che avvengono su portante radio sono largamente diffuse. I punti critici di questo tipo di trasmissione sono molteplici. Inanzitutto, dal momento in cui il segnale non è inviato su una portante fisica, c'è la possibilità che qualcuno, (spesso indicato in letteratura come *man in the middle*), possa impadronirsi del suo contenuto informativo. Chiaramente l'intruso, per avere accesso ai dati, deve essere in grado di risalire tutta la pila procollare utilizzata. Il primo scoglio che l'attaccante deve superare è a livello fisico. In genere ci si affida a tecniche standard di cifratura. Nel presente lavoro, sono proposti dei sistemi in cui i dati cifrati sono inviati su segnali che hanno delle proprietà tali da non poter essere demodulati con successo senza una conoscenza a priori di tali caratteristiche. Ad esempio possono essere sfruttati alcuni gradi di libertà della tecnica di modulazione impiegata come l'offset in fase della portante. Un altro aspetto importante è rappresentato dall'accesso alla risorsa condivisa, ossia al canale, che, nei sistemi reali è limitata in banda. Questo punto è critico essenzialmente per due motivi: a) un attaccante, una volta entrato in una rete privata, potrebbe venire in possesso di dati personali o riservati; b) l'uso di una parte della risorsa da parte di un utente non autorizzato, implica che la relativa porzione di banda non sia più disponibile per gli utenti autorizzati all'accesso, e quindi riduce la portata effettiva del sistema. Ad esempio nei sistemi che prevedono un autenticatore, o un punto di accesso, è proprio quest'ultimo che deve controllare se le richieste di accesso possono essere accolte. Per raggiungere questo obiettivo, dovrà verificare se il contenuto informativo e la pila protocollare utilizzata risultano idonei. Ovviamente l'autenticatore dovrà risalire la pila ISO-OSI a partire dal livello fisico fino ad arrivare a quello di applicazione per verificare se i dati (ad esempio una password) sono corretti. Affinché l'utente venga riconosciuto è, in genere, sufficiente che i dati siano impacchettati in modo tale che la posizione e la quantità di extra-informazione introdotta ad ogli livello sia quella corretta, e che il contenuto della parte dati

del livello di rete, di sessione e di applicazione sia quello desiderato. Nella presente tesi verrà mostrato come questo sistema di autenticazione può essere notevolmente migliorato introducendo dei controlli anche a livello MAC (Medium Access Control) e fisico. Ad esempio il centro di autenticazione potrebbe dover verificare se il protocollo di accesso al canale è stato usato correttamente, se la matrice generatrice di un eventuale codice convoluzionale sistematico utilizzato è quella giusta, e infine se il segnale fisico soddisfa tutte le proprietà del caso. Dal punto di vista dell'attaccante, per eludere i meccanismi di sicurezza associati agli alti strati della pila ISO-OSI spesso ci si affida a software, disponibili sul web, in grado di ricercare in tempi relativamente brevi la soluzione. Di converso effettuare un reverse-engeegniring su codice convoluzionale o su un turbo codice non è un qualcosa alla portata di tutti sia da un punto di vista di *know-how* che di disponibilità di risorse di calcolo. Inoltre, apparati come analizzatori di spettro ad alta frequenza (1-30Ghz) e i demodulatori di segnali hardware o software ad alte prestazioni possono essere molto costosi dal momento in cui impiegano tecnologie avanzate come le schede DSP (Discrete Signal Processor) e FPGA (Field Programmable Gate Array).

In alcuni casi, il canale radio può anche essere rumoroso e la tratta può essere soggetta al fenomeno dei cammini multipli, quindi i segnali modulati numericamente in transito potrebbero essere affetti da errori e da interferenza intersimbolica. Di vitale importanza quindi l'uso di strategie di equalizzazione e di codifica di canale adeguate.

Nella maggior parte dei casi il problema dell'autenticazione e della correzione degli errori introdotti dal canale di comunicazione sono trattati in modo separato, il che comporta una notevole riduzione della capacità del sistema e quindi della banda effettiva. In questo lavoro è mostrato come sia possibile usare i codici a correzione dell'errore come dei veri e propri codici ad autenticazione, ossia la ridondanza dovuta all'algoritmo di FEC (Forward Error Correction) non serve solo a correggere, ma può essere considerata come un hash dei dati. Se l'hash ricevuto coincide con quello memorizzato (tramite un canale sicuro, ad esempio cablato, o dedicato) durante una fase detta di *registrazione*, l'utente è riconosciuto e quindi può accedere alle risorse di interesse.

1.2 Insiemi di chiavi segrete

Un generico sistema di comunicazioni wireless è caratterizzato diversi parametri di livello fisico. Molti di questi riguardano, dal punto di vista del trasmettitore, la modulazione, la regole di codifica e la stima del canale. Al livello di collegamento possono essere invece adoperati diversi protocolli MAC.

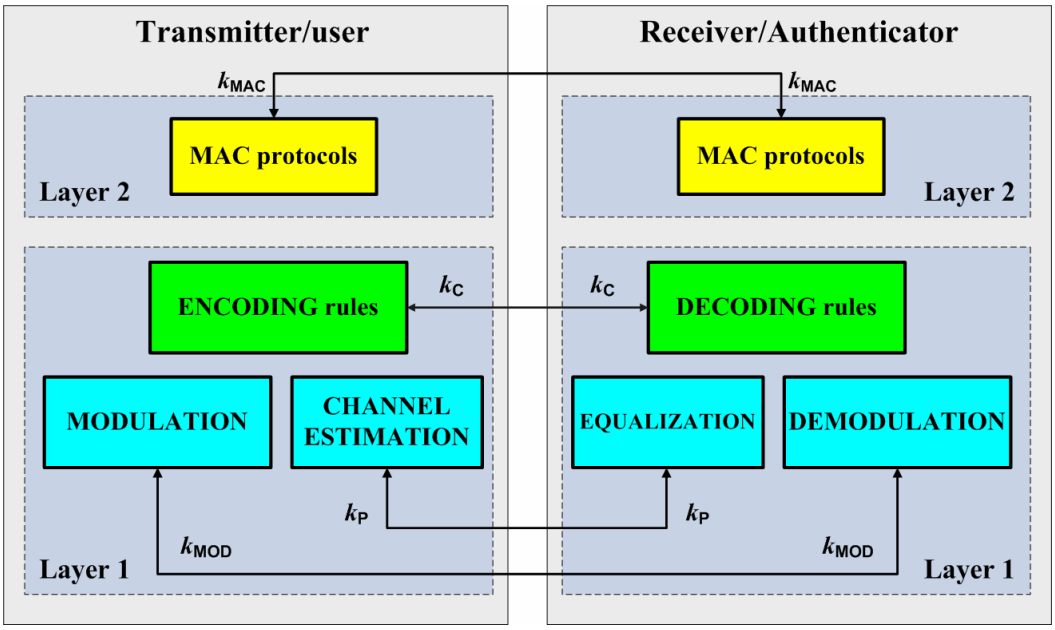

Fig.1 insiemi di chiavi segrete

In Fig.1 è illustrato il ruolo giocato dagli insiemi di chiavi segrete. Si suppone che le chiavi siano consegnate all'utente per mezzo di un canale ritenuto sicuro. Si prendano in considerazione i seguenti insiemi di chiavi:

1) $\boldsymbol{k}_{\mathbf{MOD}}$, utilizzato per fissare alcuni gradi di libertà relative alla modulazione adoperata;
2) $\boldsymbol{k}_{\mathbf{C}}$, il cui scopo è quello di stabilire le caratteristiche del codificatore (e quindi del decodificatore);
3) $\boldsymbol{k}_{\mathbf{P}}$, che determina il modo in cui viene effettuata la stima di canale, più nello specifico decide quali dati vengano impiegati a tale scopo;
4) $\boldsymbol{k}_{\mathbf{MAC}}$, in base al quale viene configurato il protocollo di livello MAC.

È importante far riferimento al termine *insieme di chiavi* piuttosto che *chiavi*, poichè non tutti gli elementi del sistema possono essere indicizzati da un solo registro.

Per esempio $\boldsymbol{k}_{\mathbf{MOD}}$ potrebbe includere $k_{\mathrm{MOD}}^{(1)}$, che sceglie il filtro di trasmissione $h_{\mathrm{T}}(t)$, e un'altra chiave $k_{\mathrm{MOD}}^{(2)}$, la quale decide il valore di un particolare grado di libertà della costellazione di riferimento, ad esempio l'offset di fase.

Perciò un insieme di chiavi private di sessione può essere indicato in generale come:

$$\boldsymbol{k} = \left\{ k^{(1)}, \quad k^{(2)}, \quad \cdots, \quad k^{(n)} \right\} \tag{1}$$

essendo n il numero di elementi di $\boldsymbol{k}$.

1.2.1 Cardinalità delle chiavi

Ogni chiave dell'insieme descritto dalla (1) appartiene in realtà a un diverso insieme, o in questo caso è più appropriato parlare di spazio di chiavi. In altre parole $k^{(j)} \in \boldsymbol{K}^{(j)}$, $j = 1, 2, \cdots, n$. In generale, indicando con $|\boldsymbol{K}^{(j)}|$ la cardinalità dell'insieme $\boldsymbol{K}^{(j)}$, più $|\boldsymbol{K}^{(j)}|$ è grande, e più $k^{(j)}$ è sicura, ossia è più difficile che possa essere trovata mediante un attacco di tipo *brute-force*. Non è detto però che $|\boldsymbol{K}^{(j)}|$ possa essere accresciuta a dismisura. Può accadere infatti che se il parametro indicizzato da k(j) varia troppo, ciò vada a scapito delle prestazioni, ossia dell'integrità informativa. Come si vedrà in seguito, ciò può senza dubbio accadere se il parametro in questione è l'offset della costellazione o l'interleaving impiegato.

1.3 Struttura del lavoro di tesi

Nel capitolo 2 sono descritti i sistemi con autenticazione e tutela della privacy basati su un algoritmo di livello fisico incentrato sul *phase-hopping*. È anche descritta una variante dell'algoritmo che implica il coinvolgimento delle funzioni di hash. Un aspetto interessante di questa alternativa risiede nella possibilità di utilizzo l'hash come algoritmo di correzione dell'errore. L'incremento del guadagno di codifica è pagato però con un maggior costo computazionale e quindi conun ritardo di elaborazione del segnale più significativo. Viene

mostrato comunque come a seconda della capacità di calcolo del sistema si può ridurre o aumentare la capacità correttiva e contestualmente il costo computazionale.

Nel capitolo 3 sono descritti i Turbo Codici, prestando particolare attenzione alla struttura matematica sia del codificatore che del decodificatore. Viene illustrata inoltre la moderna tecnica congiunta di decodifica ed equalizzazione iterativa nota come Turbo Equalizzazione, ponendo l'attenzione agli aspetti in comune con i turbo codici e a come la stessa tecnica può rappresentare un valore aggiunto nell'ambito di un algoritmo di sicurezza di livello fisico.

Nel capitolo 4, che rappresenta il cuore della tesi, viene fornita dapprima un'esaustiva trattazione matematica dei codici A-FEC ossia i codici di autenticazione. A tal proposito sono indagate formalmente le loro prestazioni in termini di probabilità di impersonificazione, sostituzione e inganno. Il modello generale viene particolarizzato ai *permuted-A-Turbo-Codes* per i quali viene elaborata una tecnica matematica per calcolare, sulla base di una procedura di Neyman-Pearson, la soglia impiegata nello step di verifica del codice. Tale soglia è inoltre adattativi, ossia dipendente dalle condizioni di rumorosità del canale. La forza di questo algoritmo risiede nella sua natura soft: l'utente è riconosciuto se il funzionale di verosimiglianza della parola di codice, usata come se fosse un hash, supera la soglia, dove quest'ultima sarà tanto più alta quanto migliori saranno le condizioni del canale. Tutto lo schema di codifica pilotata dall'opportuno insieme di chiavi e decodifica guidata dallo stesso insieme è stato simulato utilizzando come livello fisico dei segnali OFDM (Orthogonal Frequency Division Multiplexing) con costellazioni QAM (Quadrature Amplitude Modulation) sulle sotto-portanti.

L'algoritmo di sicurezza proposto nel capitolo 5 fa infine riferimento ad una tecnica di accesso al canale CSMA (Carrier Sense Multiple Access) in cui il protocollo *p*-persistent è sovrapposto ad una TDMA (Time Division Multiple Access) basata sul TH (Time Hopping).

2 Sistemi a Phase-Hopping

Le modulazioni digitali angolari come la PSK (Phase Shift Keying), la QAM e la TCM (Trellis Code Modulation) sono largamente adottate nelle comunicazioni che avvengono su ponti radio, o nello scambio di contenuti multimediali, sia nella versione a singola portante che OFDM.

La demodulazione di questo tipo di segnali è di tipo coerente. In sostanza richiede la conoscenza, istante per istante, oltre che della frequenza di simbolo f_L, anche della frequenza centrale f_c e dell'offset di fase della costellazione φ_0. In letteratura sono disponibili diversi algoritmi per la stima e l'aggiornamento di f_c e f_L. Gran parte di queste soluzioni sfrutta la stabilità degli oscillatori locali dei trasmettitori su periodi di osservazione sufficientemente lunghi

In realtà, l'impiego di modulazioni DPSK (Differential PSK) non richiede la perfetta conoscenza di f_c e nemmeno di φ_0. In tutte le applicazioni in cui le prestazioni fanno la differenza, le modulazioni digitali di fase assolute sono preferite alle differenziali visto che in media sono caratterizzate da BER migliori di 3dB [1].

2.1 Modulazioni con offset di fase

Un segnale basato su una modulazione con offset di fase, all'uscita del filtro adattato del ricevitore (ossia ad un campione per simbolo), può essere scritto, a *k*-esimo tempo di simbolo, come:

$$\underline{x}(k) = \sum_k c_k e^{j\varphi_k} = \sum_k c_k e^{j(\theta_k + \phi_0)} \qquad (2),$$

dove ϕ_0 è l'offset della costellazione. Se ϕ_0 è sconosciuta, è impossibile demodulare correttamente il segnale. Si noti come un segnale PSK con offset di fase può essere

caratterizzato mediante la (2) con c_k = cost. Anche i segnali ϕ_0-D*M*PSK hanno a che fare con il parametro di fase ϕ_0, ma questa variabile gestisce in realtà la rotazione della costellazione. Esistono diversi metodi per rilevare il parametro di rotazione in una modulazione ϕ_0-D*M*PSK, in particolare quando $\phi_0 = \pi/M$. A titolo d'esempio, si può fare riferimento ad una modulazione π/M-D*M*PSK. In questo caso le fasi dell'inviluppo complesso relativi a successivi periodi di simbolo sono generate in accordo al seguente schema:

$$\begin{aligned} &\varphi_0 = \theta_0 + \frac{\pi}{M} \\ &\varphi_1 = \theta_0 + \theta_1 + \frac{\pi}{M} \\ &\cdots \quad \cdots \quad \cdots \\ &\varphi_k = \theta_{k-1} + \theta_k + \frac{\pi}{M} \\ &\cdots \quad \cdots \quad \cdots \end{aligned} \qquad (3).$$

I termini $\pm\pi/M$ alternano la costellazione tra due sotto-costellazioni ad *M* punti. Dunque qualora la frequenza centrale stimata f_0 coincida esattamente con la portante effettiva f_c, il diagramma I-Q del segnale mostrerà, su un periodo di osservazione opportunamente lungo, 2*M* points (ossia. 4 fasi per una $\pi/2$-DBPSK, 8 per una $\pi/4$-DQPSK e così via). Se il segnale ricevuto è riportato in banda-base utilizzando una $f_0 = f_c \pm \frac{f_L}{2M}$ (f_c e f_L sono rispettivamente la frequenza portante e la frequenza di simbolo) viene forzata una rotazione pari a $\Delta\varphi_k = 2\pi \frac{f_0 - f_c}{f_L} = \pm\frac{\pi}{M}$ ad ogni *k*-esimo tempo di simbolo. In queste condizioni solo *M* punti saranno visualizzati nonostante la modulazione sia in effetti una π/M-D*M*PSK (ossia 2 punti per una $\pi/2$-DBPSK, 4 per una $\pi/4$-DQPSK e così via).

For questo motivo non è conveniente usare la rotazione delle modulazioni differenziali con offset come un grado di libertà su cui basare un algoritmo di sicurezza di livello fisico.

Si preferisce far riferimento a segnali trasmessi con modulazioni PSK con offset, ma assolute.

2.2 Sistemi OFDM a Phase-Hopping

Su un segnale OFDM, in ogni sotto-portante può essere adottata una diversa costellazione con offset, il cui inviluppo complesso è del tipo (2).

Dunque l'inviluppo complesso di un segnale OFDM a Phase-Hopping (PH-OFDM) trasmesso durante l'intervallo di simbolo [0, T) è:

$$\underline{x}(t) = rect_T(t)\sum_{n=0}^{N-1} c_n e^{j(2\pi n\Delta f t+\varphi_n)}\ , \quad c_n = a_n + jb_n \qquad (4),$$

dove N è il numero di portanti OFDM, Δf = 1/T, c_n il punto relativo all'n-esima sotto-costellazione, e φ_n uno shift in fase arbitrario per l'n-esima portante, il quale deve essere noto al ricevitore per demodulare correttamente il segnale. Un segnale OFDM standard è dato dalla (4) con $\varphi_n = 0$, per ogni n.

Al fine di ridurre l'interferenza con i sistemi che utilizzano le bande adiacenti, in genere solo un sottoinsieme $\tilde{N}$ di tutte le sotto-portanti è effettivamente utilizzato.

Il demodulatore OFDM converte in forma digitale le componenti I e Q del segnale ricevuto, tramite una frequenza di campionamento pari a $f_s = N/T$. Introducendo $\mathbf{t} = [t_0\ t_1 \cdots\ t_{N-1}]^{\mathbf{T}}$ come vettore temporale di riferimento, il segnale OFDM standard può essere rappresentato per mezzo di una matrice $\mathbf{S(t)}$ $N \times \tilde{N}$, dove N è il numero di campioni e $\tilde{N}$ il numero di portanti.

$$\mathbf{S(t)} = \left[c_0 \mid c_1 e^{j2\pi\Delta f\,\mathbf{t}} \cdots \mid c_{\tilde{N}-1} e^{j2\pi(\tilde{N}-1)\Delta f\,\mathbf{t}} \right] =$$

$$= \begin{bmatrix} c_0 & c_0 & \cdots & c_0 \\ c_1 e^{j2\pi\Delta f\, t_0} & c_1 e^{j2\pi\Delta f\, t_1} & \cdots & c_1 e^{j2\pi\Delta f\, t_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ c_{\tilde{N}-1} e^{j2\pi(\tilde{N}-1)\Delta f\, t_0} & c_{\tilde{N}-1} e^{j2\pi(\tilde{N}-1)\Delta f\, t_1} & \cdots & c_{\tilde{N}-1} e^{j2\pi(\tilde{N}-1)\Delta f\, t_{N-1}} \end{bmatrix} \qquad (5).$$

In questo modo l'inviluppo a tempo-discreto è perfettamente caratterizzato sia nel dominio del tempo che in quello della frequenza. Sia

$$\mathbf{M}(\varphi) = diag\left(e^{j\varphi_0}\ e^{j\varphi_1} \cdots e^{j\varphi_{\tilde{N}-1}}\right) = \begin{bmatrix} e^{j\varphi_0} & 0 & \cdots & 0 \\ 0 & e^{j\varphi_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & e^{j\varphi_{\tilde{N}-1}} \end{bmatrix} \tag{6}$$

la matrice di correzione delle fasi usata dal modulatore , dove $\varphi = [\varphi_0\ \varphi_1 \cdots \varphi_{\tilde{N}-1}]$ è il vettore di phase-hopping. La versione digitale della (4), quando $\varphi_n \neq 0$ è perciò ottenuta tramite la somma di ogni colonna di $\mathbf{M}(\varphi)\mathbf{S}(\mathbf{t})$, ossia di

$$\begin{bmatrix} c_0 e^{j\varphi_0} & c_0 e^{j\varphi_0} & \cdots & c_0 e^{j\varphi_0} \\ c_1 e^{j(2\pi\Delta f\, t_0 + \varphi_1)} & c_1 e^{j(2\pi\Delta f\, t_1 + \varphi_1)} & \cdots & c_1 e^{j(2\pi\Delta f\, t_{N-1} + \varphi_1)} \\ \vdots & \vdots & \ddots & \vdots \\ c_{\tilde{N}-1} e^{j[2\pi(\tilde{N}-1)\Delta f\, t_0 + \varphi_{\tilde{N}-1}]} & c_{\tilde{N}-1} e^{j[2\pi(\tilde{N}-1)\Delta f\, t_1 + \varphi_{\tilde{N}-1}]} & \cdots & c_{\tilde{N}-1} e^{j[2\pi(\tilde{N}-1)\Delta f\, t_{N-1} + \varphi_{\tilde{N}-1}]} \end{bmatrix} \tag{7}$$

In ricezione deve essere utilizzato l'operatore di correzione inverso $\mathbf{D}(\boldsymbol{\varphi}) = diag(e^{-j\varphi_1}\quad e^{-j\varphi_2}\quad \ldots\quad e^{-j\varphi_N})$ prima di applicare qualsiasi schema di demodulazione digitale quale QAM o PSK. Si può infatti dimostrare come $\mathbf{M}(\varphi)\mathbf{S}(\mathbf{t})\mathbf{D}(\varphi) = \mathbf{S}(\mathbf{t})$.

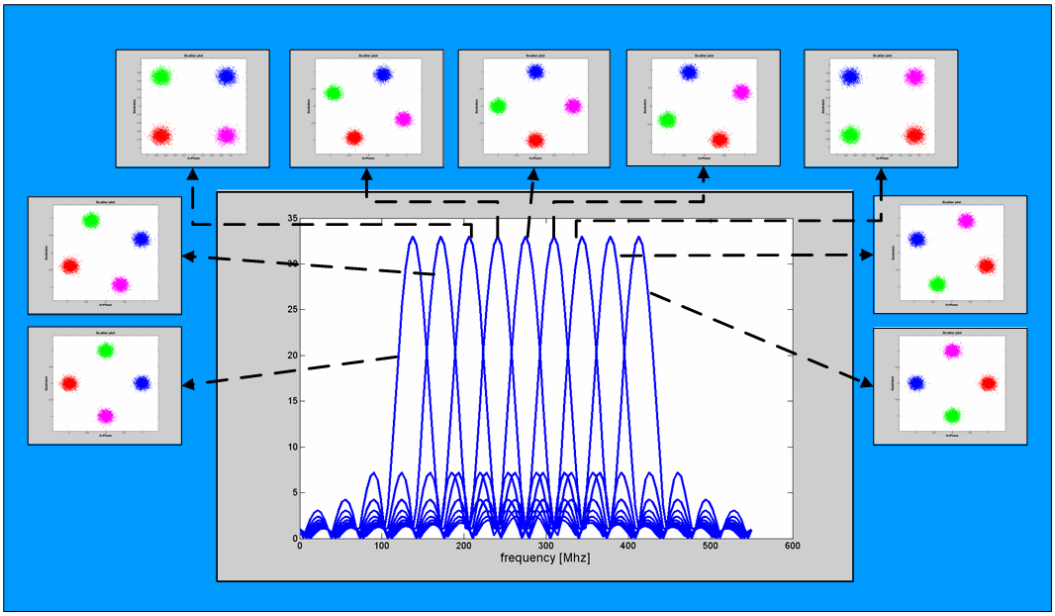

Fig.2 Segnali QPSK-PH-OFDM

Fig.2 illustra con uno schema molto compatto come è costruito un segnale PH-QPSK-OFDM. I punti delle costellazioni sono colorati in modo tale da mettere in rislato le rotazioni di fase relative fra differenti portanti.

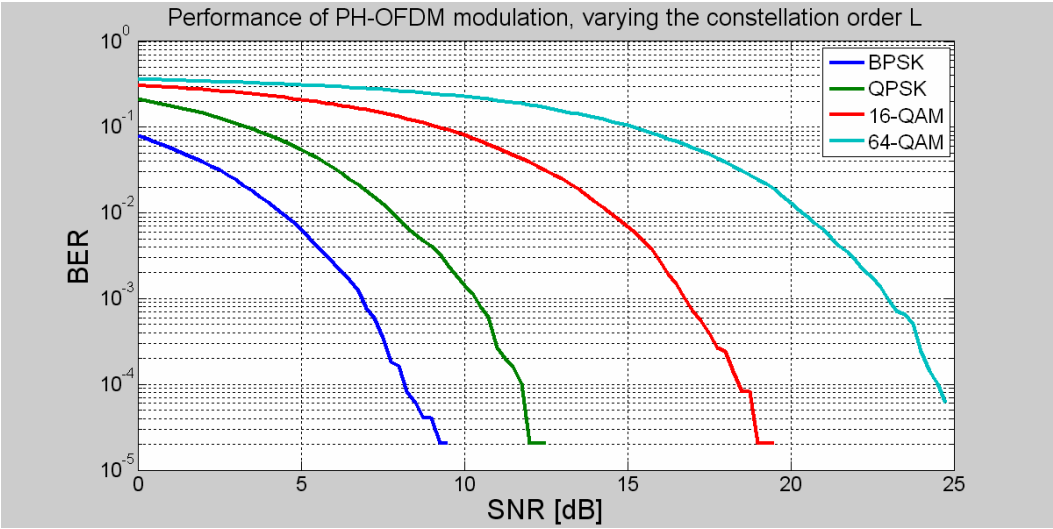

Fig.3 prestazioni dei sistemi PH-OFDM, al variare del numero di punti della costellazione *L*

2.2.1 Insiemi di chiavi private associate all'algoritmo

Facendo riferimento alla notazione introdotta nel capitolo 1, si tratta di definire una lista di insiemi $\boldsymbol{k}_{\mathbf{MOD}}$. Per il sistema in questione ha più senso parlare di lista di chiavi più che di insiemi di chiavi, visto che il grado di libertà del segnale sfruttato è solo uno e cioè la sua fase.

La cardinalità di $\boldsymbol{k}_{\mathbf{MOD}}$ è data dal numero di possibili rotazioni permesse dalle modulazioni adottate sulle sotto-portanti.

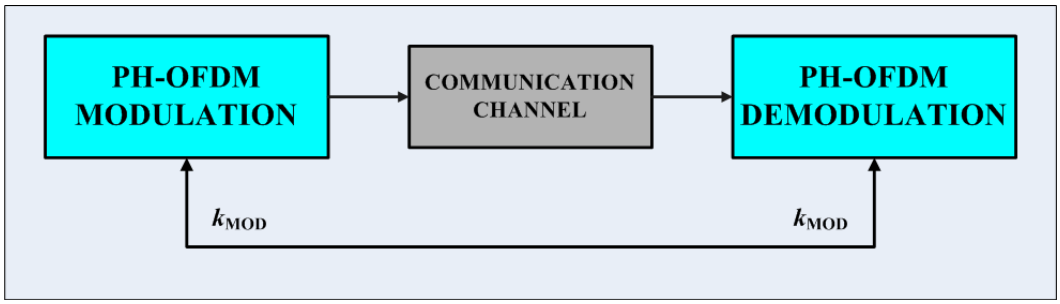

Fig.4 schemi di modulazione e demodulazione PH-OFDM

In accordo alla (1) $\boldsymbol{k}_{\mathbf{MOD}} = \left\{ k_{\mathbf{MOD}}^{(1)} \quad k_{\mathbf{MOD}}^{(2)} \quad \cdots \quad k_{\mathbf{MOD}}^{(n)} \quad \cdots \quad k_{\mathbf{MOD}}^{(N)} \right\}$, dove $k_{\mathbf{MOD}}^{(n)}$ indirizza φ_n, ossia l'offset dell'*n*-esima sotto-portante.

2.2.2 Impatto del phase-hopping sule prestazioni del sistema

Sia $\mathcal{N}_0$ la densità spettrale di potenza del rumore additive, gaussiano e bianco presente sul canale di comunicazione e $\mathbf{n} = \mathbf{n}_I + j\mathbf{n}_Q$ il vettore dei campioni delle componenti in fase e in quadratura di tale rumore la cui matrice di covarianza sia, pertanto:

$$\mathbf{R}_{\mathbf{n}} = E\left\{\mathbf{n}\mathbf{n}^{\dagger}\right\} = 4\mathcal{N}_0 N \Delta f\, \mathbf{I} \qquad (8).$$

Dal momento in cui per l'operatore FFT **W** vale la sequente proprietà:

$$\mathbf{W}\mathbf{W}^{\dagger} = \mathbf{W}^{\dagger}\mathbf{W} = \frac{1}{N}\mathbf{I} \qquad (9),$$

la corrispondente uscita della FFT $\tilde{\mathbf{n}} = \tilde{\mathbf{n}}_{\mathrm{Re}} + j\tilde{\mathbf{n}}_{\mathrm{Im}}$ può essere modellata tramite una variabile gaussiana *n*-dimensionale con matrice di covarianza

$$\mathbf{R}_{\tilde{\mathbf{n}}} = E\left\{\tilde{\mathbf{n}}\tilde{\mathbf{n}}^{\dagger}\right\} = \mathbf{W}\mathbf{R}_{\mathbf{n}}\mathbf{W}^{\dagger} = 4\mathcal{N}_0 \Delta f\, \mathbf{I} \qquad (10).$$

Dunque la matrice di covarianza del rumore dopo la correzione di fase è

$$\mathbf{R}_{\tilde{\mathbf{n}}_{\varphi}} = E\left\{\tilde{\mathbf{n}}_{\varphi}\tilde{\mathbf{n}}_{\varphi}^{\dagger}\right\} = \mathbf{D}(\boldsymbol{\varphi})\mathbf{R}_{\tilde{\mathbf{n}}}\mathbf{D}(\boldsymbol{\varphi})^{\dagger} = 4\mathcal{N}_0 \Delta f\, \mathbf{I} \qquad (11)$$

Dal confronto della (10) con la (11) segue che le componenti di rumore con o senza la correzione di fase sono identiche dal punto di vista probabilistico. Per questo motivo si può ritenere che la correzione di fase non altera le prestazioni di un sistema OFDM.

2.3 Sistemi Hash PH-OFDM

In un sistema PH-OFDM con funzioni di hash, ossia in un HASH-PH-OFDM, *M* delle $\tilde{N}$ frequenze OFDM usate vengono spese per trasmettere simboli di informazione, mentre le rimanenti $\tilde{N} - M$ sotto-bande trasportano l'hash degli stessi dati.

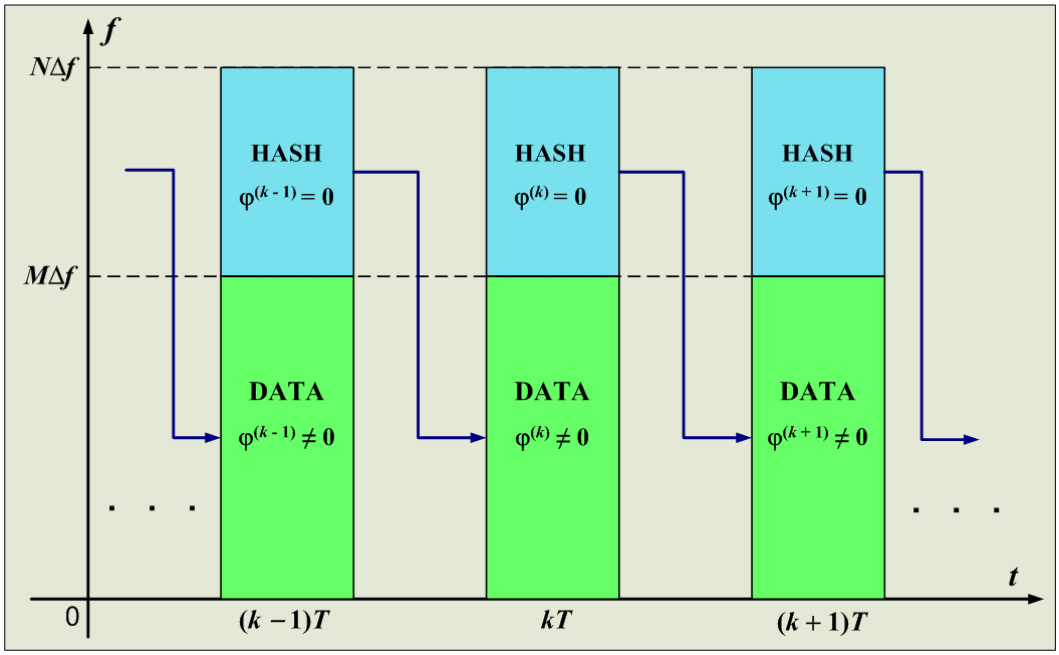

Fig.5 sistemi PH-OFDM

Come illustrato in Fig.5, le prime M costellazioni (relative ai simboli di informazione) sono ruotate, ogni tempo di simbolo T delle quantità φ_m, $1 \le m \le M$, mentre per la parte hash $\varphi_m = 0$, $M+1 \le m \le N$.

Per ridurre la vulnerabilità, la sequenza di phase-hopping $\varphi^{(k)} = [\varphi_0^{(k)}\ \varphi_1^{(k)} \cdots \varphi_{M-1}^{(k)}]$ per il k-esimo periodo di simbolo ($\varphi_i^{(j)}$ sta per la fase dell'i-esima portante del j-esimo frame OFDM), è determinata da un sottoinsieme ***P*** dei bit di informazione trasmessi durante il pacchetto precedente.

2.3.1 Autenticazione dell'utente basata su algoritmi di hash

Dal punto di vista della sicurezza, l'hash può essere utilizzato per riconoscere gli utenti. In altre parole se l'hash ricevuto coincide con quello dei dati ricevuti (ottenuto applicando l'operatore di hash ai dati recuperati) o è uguale ad un hash registrato durante la fase di enrolment l'utente è riconosciuto. Il sistema proposto nel paragrafo 2.3.2 è del primo tipo (l'hash ricevuto coincide con quello dei dati ricevuti) mentre quello presentato nel paragrafo 2.3.3 è una variante basata sullo step di enrolment .

2.3.2 Hash e FEC

La sequenza di phase-hopping $\varphi^{(k)} = [\varphi_0^{(k)}\ \varphi_1^{(k)} \cdots\ \varphi_{M-1}^{(k)}]$ è valida per il *k*-esimo periodo di simbolo ($\varphi_i^{(j)}$ è la fase dell'*i*-esima portante per il *j*-esimo simbolo OFDM), ed è determinata dall'hash di un sottoinsieme ***P*** dei bit di informazione trasmessi durante il pacchetto immediatamente precedente. La cardinalità di ***P*** e la selezione che esso comporta può essere decisa da un'altra chiave di sessione privata, anche questa consegnata all'utente per mezzo di un canale sicuro. Solamente i bit di informazione relativi al primo pacchetto sono trasmessi senza alcuna sequenza di hopping.

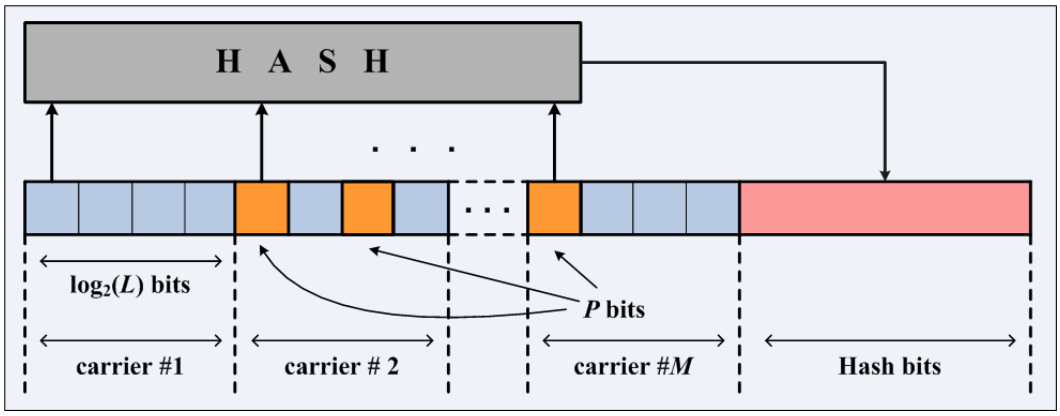

Fig.6 hash a correzione dell'errore

Indicando con $P = |\boldsymbol{P}|$ la cardinalità di ***P*** e nell'ipotesi che i bit di hash siano recuperate senza alcun errore, effettuando, al lato ricevitore, 2^P tentativi, è possible correggere esattamente *P* errori sui bit di informazione che appartengono a ***P***, dal momento in cui gli algoritmi scelti MD (Message Digest) o SHA (Secure Hash Algorithms) sono senza collisione. È evidente come aumentare *P* significhi migliorare la capacità correttiva della funzione di hash utilizzata, ma implica un maggior costo computazionale. Il metodo è sintetizzato in Fig.6.

Si indichi con $\mathbf{b}^{(k)}$ il vettore binario a lunghezza $\log_2(L)\tilde{N}$ relativo al *k*-esimo periodo di simbolo. Il vettore binario di lunghezza $\log_2(L)\tilde{N}$ $\mathbf{b}^{(0)}$, valido per il primo tempo di simbolo, è datao dall'accostamento del vettore di dati binario di lunghezza $[\log_2(L)\tilde{N} - n_h]$ $\mathbf{d}^{(0)}$ con il

vettore binario a n_h componenti dato dall'hash di $\mathbf{d}_P^{(0)}$, indicato con $\mathbf{H}\left(\mathbf{d}_P^{(0)}\right)$ ($\mathbf{d}_P^{(0)}$ sta per "una selezione di *P* bit di $\mathbf{d}^{(0)}$"). Per Φ-GRAY sia l'operazione di mappatura della sequenza binaria $\mathbf{H}\left(\mathbf{d}_P^{(0)}\right)$ (lunga n_h bit) nel vettore delle fasi (di lunghezza n_p) e la ripetizione dello stesso (n_{rep} volte) in modo tale da coprire la sequenza di phase hopping $\varphi^{(0)}$. La lunghezza di questo particolare codice di Gray è $\log_2\left(n_\varphi\right)$. Sebbene $\varphi^{(0)}$ sia calcolata durante la trasmissione del pacchetto numero $k = 0$, sarà applicato alle *M* portanti di dati relativi al periodo di simbolo numero $k = 1$. Il vettore binario $\mathbf{b}^{(0)}$ a lunghezza $\log_2(L)\tilde{N}$ è quindi mappato nel corrispondente vettore di $\tilde{N}$ simboli OFDM $\mathbf{c}^{(0)}$, utilizzando un alfabeto di Gray di misura $\log_2(L)$, (L-GRAY). $\mathbf{c}^{(k)} = \left[c_0^{(k)}\ c_1^{(k)} \cdots c_n^{(k)} \cdots c_{\tilde{N}-1}^{(k)}\right]$ è il vettore di simboli relativo al k-esimo pacchetto. Dunque $x^{(0)}(t)$ è trasmesso in base a $\mathbf{c}^{(0)}$. Fig.7 riepiloga tale meccanismo su un *k* generico.

Il segnale relativo al *k*-esimo pacchetto può essere espresso come:

$$\begin{aligned} x^{(k)}(t) &= x_c^{(k)}(t)\cos(2\pi f_p t) - x_s^{(k)}(t)\sin(2\pi f_p t), \\ x_c^{(k)}(t) &= \mathrm{Re}\left[\underline{x}^{(k)}(t)\right],\ \ x_s^{(k)}(t) = \mathrm{Im}\left[\underline{x}^{(k)}(t)\right] \end{aligned} \qquad (12),$$

dove f_p è la frequenza portante centrale, e dove $\underline{x}^{(k)}(t)$ è l'inviluppo complesso di $x^{(k)}(t)$ dato da:

$$\begin{aligned} \underline{x}^{(k)}(t) &= \underline{x}_M^{(k)}(t) + \underline{x}_{\tilde{N}-M}^{(k)}(t), \\ \underline{x}_M^{(k)}(t) &= rect_T(t-kT)\sum_{n=0}^{M-1} c_n^{(k)} e^{j(2\pi n\Delta f\, t + \varphi_n^{(k)})}, \\ \underline{x}_{\tilde{N}-M}^{(k)}(t) &= rect_T(t-kT)\sum_{n=M}^{\tilde{N}-1} c_n^{(k)} e^{j2\pi n\Delta f\, t} \end{aligned} \qquad (13).$$

Ipotizzando che il canale di comunicazione sia semplicemente un AWGN, la forma d'onda ricevuta al *k*-esimo tempo di simbolo OFDM (o pacchetto) sarà:

$$r^{(k)}(t) = x^{(k)}(t) + n(t) \qquad (14),$$

dove $n(t)$ è un rumore gaussiano, bianco e a valor medio nullo e dove $x^{(k)}(t)$ è descritta dalle (12) e (13).

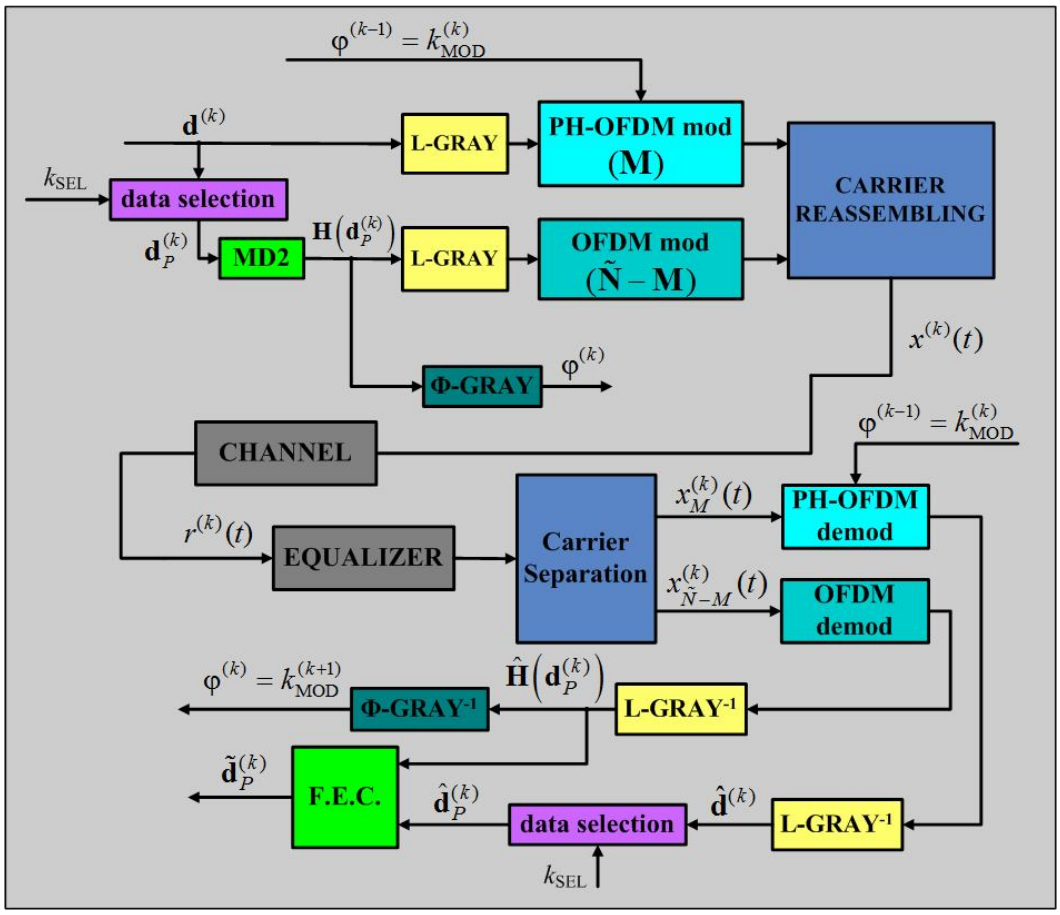

Fig.7 Schema a blocchi di trasmissione/ricezione per un sistema PH-OFDM con HASH

Sia $\hat{\mathbf{x}}$ la stima provvisoria di $\boldsymbol{x}$, e $\tilde{\mathbf{x}}$ quella definitiva. Demodulando $r^{(0)}(t)$ si ottiene il vettore di $\tilde{N}$ simboli $\hat{\mathbf{c}}^{(0)}$. Applicando l'L-GRAY inverso si ottiene $\hat{\mathbf{b}}^{(0)}$. I primi $\log_2(L)\tilde{N} - n_h$ bit, che costituiscono il vettore di dati stimato $\hat{\mathbf{d}}^{(0)}$, vengono pertanto separati dai rimanenti n_h bit, che danno il vettore di stima di hash $\hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right)$. L'operatore di hash $\mathbf{H}(\cdot)$ è dunque applicato a $\hat{\mathbf{d}}_P^{(0)}$. Se il risultato di quest'operazione non coincide con $\hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right)$, tutte le possibili 2^P combinazioni per $\hat{\mathbf{d}}_P^{(0)}$ sono tentate, ottenendo a ogni tentativo un diverso $\tilde{\mathbf{d}}_P^{(0)}$,

finchè $\mathbf{H}\left(\tilde{\mathbf{d}}_P^{(0)}\right) = \hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right)$. Tale condizione risulterà verificata, dopo un numero variablie di tentativi (in media pari a 2^{P-1}), solamente se tutti i bit di $\hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right)$ sono stati recuperati correttamente, in altre parole se $\hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right) = \mathbf{H}\left(\mathbf{d}_P^{(0)}\right)$.

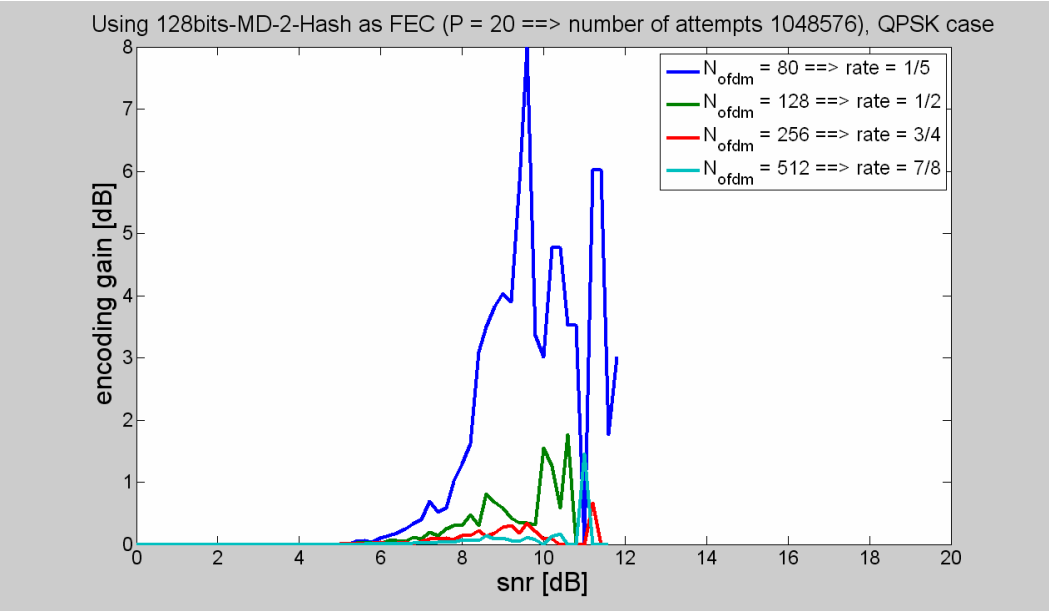

Fig.8 Guadagno di cofdifica ottenuto utilizzando la funzione di hash MD-2 (a 128 bit) come FEC

L'utilizzo di una funzione di hash anche come schema a correzione d'errore non porta sicuramente a guadagni di codifica paragonabili con quelli garantiti da CRC, BCH, Reed-Solomon, convoluzionali, turbo codici. Ad ogni modo il leggero miglioramento delle prestazioni è ottenuto senza introdurre ridondanza aggiuntiva, ossia senza intaccare minimamente l'efficienza spettrale (si veda Fig.8).

2.3.3 Costruzione di hash "robusti"

Una funzione di hash può dirsi "robusta" se una piccola variazione dell'ingresso non comporta un enorme variazione dell'uscita ad essa relativa. Gli hash standard non supportano questa proprietà.

Un modo per ottenere un hash robusto da uno standard consiste nell'introduzione di un algoritmo di FEC (ad esempio un codice lineare a blocco sistematico) ed un processo di clusterizzazione nel dominio di hash.

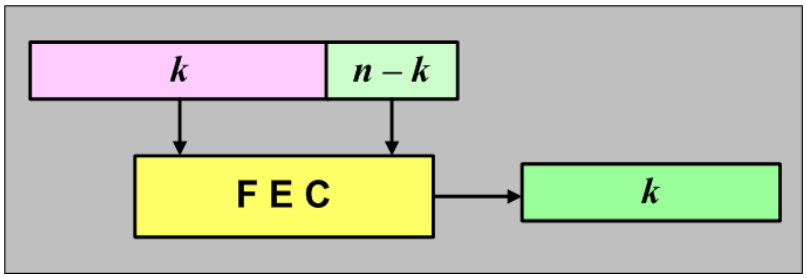

Fig.9 costruzione di un hash robusto

Sia n la lunghezza dell'output dell'hash. Gli ultimi $n - k$ bit possono essere pensati come la componente di parità dovuta a un algoritmo di correzione dell'errore sistematico, come mostrato in Fig.9. In questo modo ogni hash (prodotto o ricostruito) può essere "corretto" ovvero ricondotto ad una delle 2^k parole di codice valide. Questa proprietà, che può essere chiamata di "clusterizzazione" del dominio di hash, è schematizzata in Fig.10.

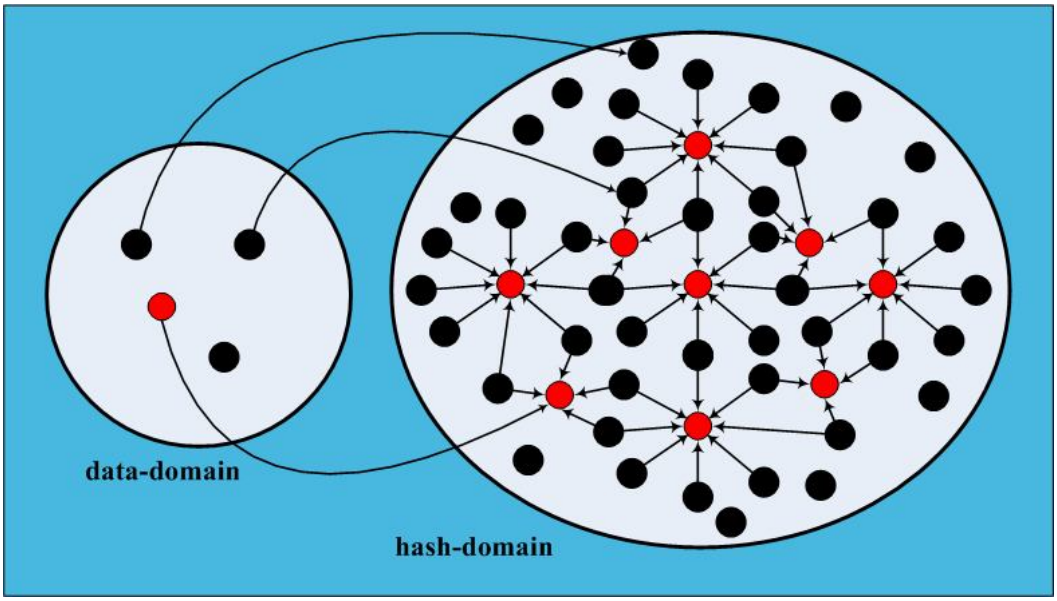

Fig.10 clusterizzazione del dominio di hash

Per capire come può essere applicata questa metodologia alle situazioni reali si consideri uno schema di autenticazione piuttosto semplice.

Lo step di enrollment consiste nella memorizzazione, tramite un canale sicuro, dell'hash di un messaggio $\boldsymbol{x}$ lungo q bit. Sia n invece la lunghezza di $H(\boldsymbol{x})$. Quest'ultimo può assumere qualsiasi valore, a seconda dell'algoritmo di hash utilizzato. Per questo motivo, perfino durante lo step di enrollment, $H(\boldsymbol{x})$ è ricondotto ad una delle 2^k parole di codice valide utilizzando un codice lineare a blocco (n,k), ciclico, sistematico. Dunque il valore registrato non è $H(\boldsymbol{x})$, ma $H_k(\boldsymbol{x})$, dove $H_k(\boldsymbol{x})$ è $H(\boldsymbol{x})$ dopo che l'algoritmo di FEC è stato applicato. In generale $H_k(\boldsymbol{x}) \neq H(\boldsymbol{x})$, dove $H_k(\boldsymbol{x})$ è un punti rosso del dominio di hash illustrato in Fig.10, e $H(\boldsymbol{x})$ un punto nero.

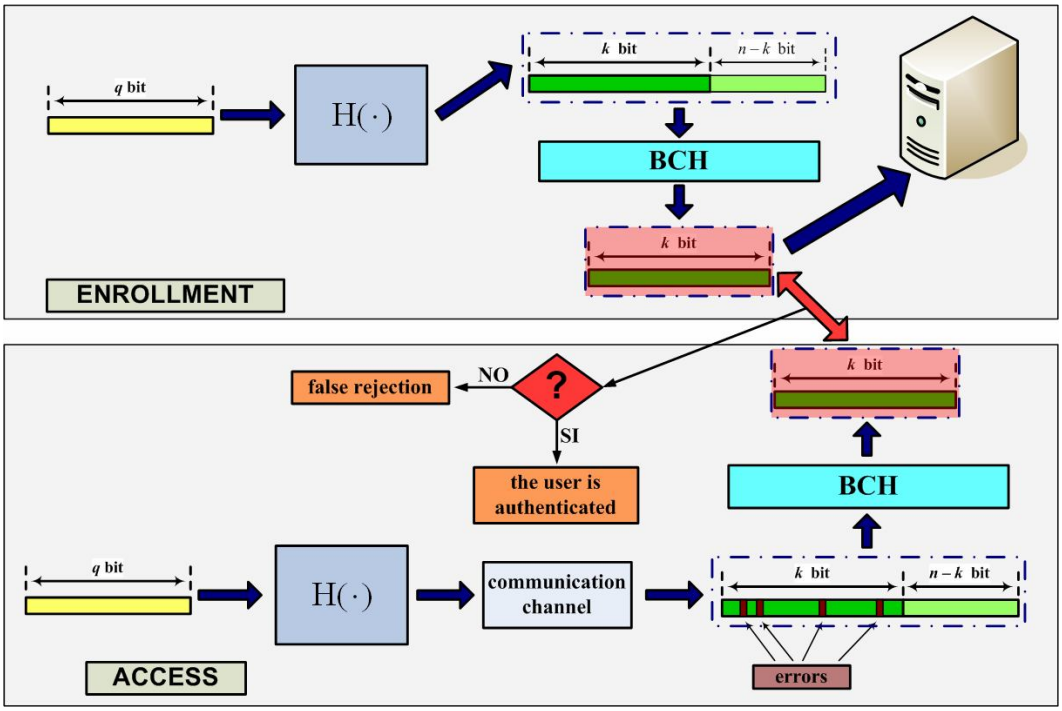

Fig.11 un semplice schema di autenticazione

Durante lo step di autenticazione, se le condizioni del canale sono severe, l'hash ricevuto $\hat{H}(\boldsymbol{x})$ potrebbe essere diverso da $H_k(\boldsymbol{x})$ (in realtà ciò potrebbe anche accadere se $H(\boldsymbol{x})$ non è una parola di codice dell'algoritmo di FEC selezionato). Se il numero di errori (ossia il numero di bit di $\hat{H}(\boldsymbol{x})$ diversi dai corrispondenti bit di $H_k(\boldsymbol{x})$) non supera la capacità correttiva

t dello schema di codifica adottato, $\hat{\mathrm{H}}(\boldsymbol{x})$ può essere ricondotto a $\mathrm{H}_k(\boldsymbol{x})$, e l'utente può essere riconosciuto. Questo scenario è descritto in Fig.11.

2.4 Un sistema PH-OFDM basato su hash robusti

Il sistema consta di tre stadi: 1) registrazione (enrollment); 2) autenticazione; 3) comunicazione criptata.

Durante il primo stadio un *serial number* s, e una chiave privata di cifratura K_c, sono consegnati all'utente attraverso un canale sicuro. Contestualmente l'hash della versione binaria di s, ovvero b, è registrato nell'AuC (Authentication Centre) o AP (Access Point).

Nel corso dello stadio di autenticazione, b_{tc} (versione turbo-codificata di b) è (come primo pacchetto) trasmesso sulle N portanti attive, utilizzando una modulazione OFDM standard, ossia senza PH (Phase Hopping). Sia n la lunghezza di $\mathrm{H}(b)$, dove $\mathrm{H}(\cdot)$ rappresenta l'operatore di hash. Gli ultimi $n - k$ bit sono la parte di controllo di parità di una parola di codice BCH (Bose-Chaudhuri-Hocquenghem). La scelta del codice, così come del suo tasso k/n saranno approfondite in seguito. In questo modo gli ultimi $n - k$ bit sono usati per ricondurre tutti gli n bit ad una delle parole di codice valide. Dunque lo xor logico dei primi k bit con K_c porta alla sequenza binaria K_a. Il logaritmo in base 10 della versione intera decimale di K_a è dunque approssimato all'intero più vicino. Quest'ultimo fornisce lo stato S di un generatore pseudo-randomico di interi che produce N numeri, nel range $[0, r-1]$ e che determina l'offset in fase di ognuna delle N portanti OFDM, per il frame successivo. Si faccia riferimento alla PSK per ogni sotto-banda OFDM.

Lo stadio di autenticazione continua per altri $K - 1$ pacchetti (o simboli OFDM), ognuno caratterizzato da un offset a PH stabilito durante la trasmissione precedente, ma trasportando la stessa sequenza binaria, ovvero il serial number s. Per ogni pacchetto ricevuto, l'AuC controlla se l'hash dell's recuperato, ossia $\mathrm{H}(\hat{b})$, coincide con la versione registrata $\mathrm{H}(b)$.

L'utente è riconosciuto se per almeno Q dei K pacchetti ricevuti risulta $\mathrm{H}(\hat{b}) = \mathrm{H}(b)$. È importante osservare come Q possa essere adattativo, ossia dipendente dalle condizioni del

canale. Per rapporti segnale-rumore eccessivamente bassi possono verificarsi degli eventi di falso respingimento (*False Rejection*).

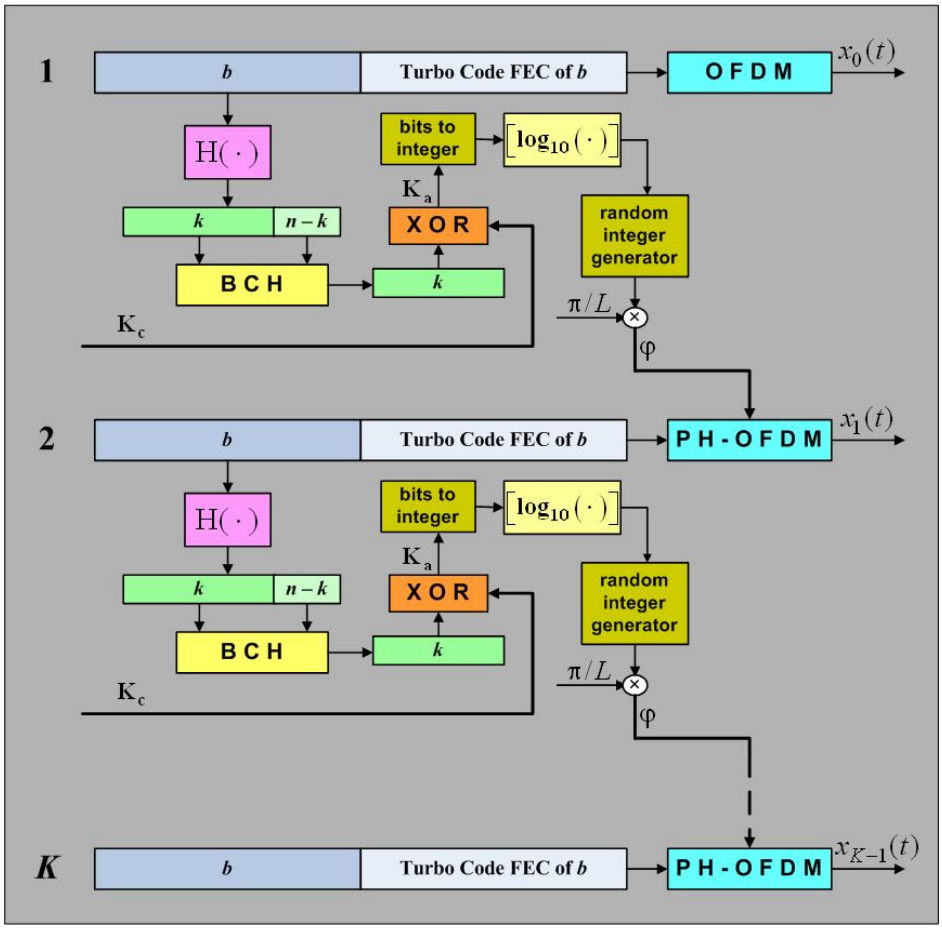

Fig.12 stadio di autenticazione (lato trasmettitore)

Nelle figure12 e 13 sono mostrati rispettivamente gli stadi di autenticazione per il trasmettitore e il ricevitore.

L'AuC non riceve $x_i(t)$, $i = 0, 1, \ldots, K-1$, ma $r_i(t) = x_i(t) + w(t)$, dove $w(t)$ è un rumore gaussiano, bianco, additivo e a valor medio nullo. Dunque, in seguito alla demodulazione OFDM (standard) di $r_0(t)$, $\hat{b}_{tc_0}$ può essere ricostruita. Gli ultimi $(1-1/R)N\log_2(L)$ bit (ossia la parte di FEC) della stessa (R è il rate del turbo-codice scelto), sono usati per determinare la

migliore versione di $\hat{b}_0$, ovvero $\tilde{b}_0$. Una volta ottenuta $\tilde{b}_0$, questa può essere adoperata per ricavare $\tilde{\varphi}_0$.

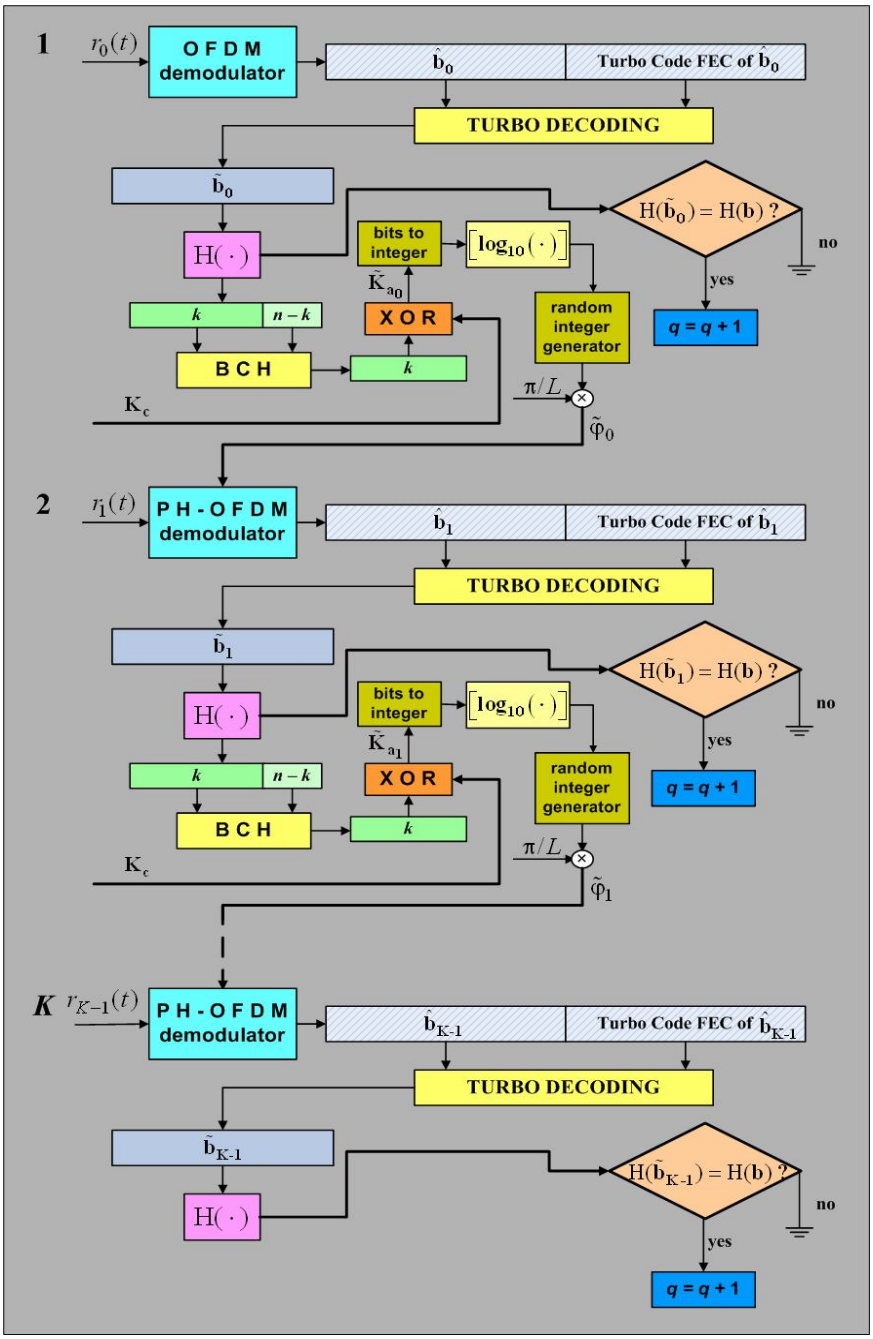

Fig.13 stadio di autenticazione (lato ricevitore)

Sia ora q il contatore dei riscontri positivi adoperato dal ricevitore, inizializzato a $q = 0$. Se $\mathrm{H}(\tilde{b}_0) = \mathrm{H}(b)$ (l'ultimo è il valore dell'hash memorizzato durante l'enrollment) $q = q + 1$, altrimenti q rimane 0. Una nuova chiave di autenticazione $\tilde{K}_{a_0}$ sarà determinata e così un nuovo vettore delle fasi $\tilde{\varphi}_0$. Se $\tilde{\varphi}_0 = \varphi$ il secondo pacchetto ricevuto, ossia $r_1(t)$, sarà elaborato con una demodulazione PH-OFDM, otttenendo un nuovo $\tilde{b}_1$.Se $\mathrm{H}(\tilde{b}_1) = \mathrm{H}(b)$, $q = q + 1$. Una nuova chiave di autenticazione $\tilde{K}_{a_1}$ sarà dunque determinata e così un nuovo vettore delle fasi $\tilde{\varphi}_1$. Se $\tilde{\varphi}_1 = \varphi$ il terzo simbolo OFDM ricevuto $r_2(t)$, potrà essere demodulato correttamente, e così via, finchè $r_{K-1}(t)$ è ricevuto. Per questo, ultimo, pacchetto di autenticazione, viene calcolata $\mathrm{H}(\tilde{b}_{K-1})$, esclusivamente per verificare se è uguale a $\mathrm{H}(b)$, mentre $\tilde{K}_{a_{K-1}}$ e $\tilde{\varphi}_{K-1}$ non sono recuperate dal momento in cui l'AuC non aspetta ulteriori simboli OFDM di autenticazione. Una volta che il K-esimo pacchetto è stato ricevuto, se $q \geq Q$, l'utente è riconosciuto.

Durante la fase di comunicazione si utilizza lo stesso schema, con l'unica differenza che stavolta il serial number è sostituito dai dati. La comunicazione è effettivamente cifrata, in quanto i pacchetti non possono essere ricostruiti se non sono note le sequenza di hopping. Per evitare indesiderate perdite di sincronismo, e per mantenere alto il livello di sicurezza, il serial-number può essere periodicamente ritrasmesso.

Hash algorithm	**n**	**k**	**ECC**
MD2	127	22	23
MD5	127	22	23
SHA-256	255	99	23
SHA-512	511	313	23

Tab.1 compatibilità fra codici BCH e tipici algoritmi di hash

Il numero di portanti non nulle N non ha impatto sul BER di una modulazione OFDM standard. Il BER di una comunicazione PH-OFDM è invece fortemente dipendente dal valore di N (più N è grande è più è alta la probabilità che il vettore delle fasi sia ricostruito in modo

errato). Come conseguenza di ciò, anche l'FRR (False Rejection Rate) varia con *N*. Il numero di portanti prese in considerazione è $N = 512$, 2048 e 8192.

Su tutte le *N* portanti è usata una *L*-PSK. $L = 4$, 8 e 16 sono i valori usati nelle simulazioni.

In seguito $v = 1 / R$ indicherà la quantità di ridondanza dovuta al turbo codice (*R* è il suo rate). I valori usati sono $v = 3$, 5 e 7.

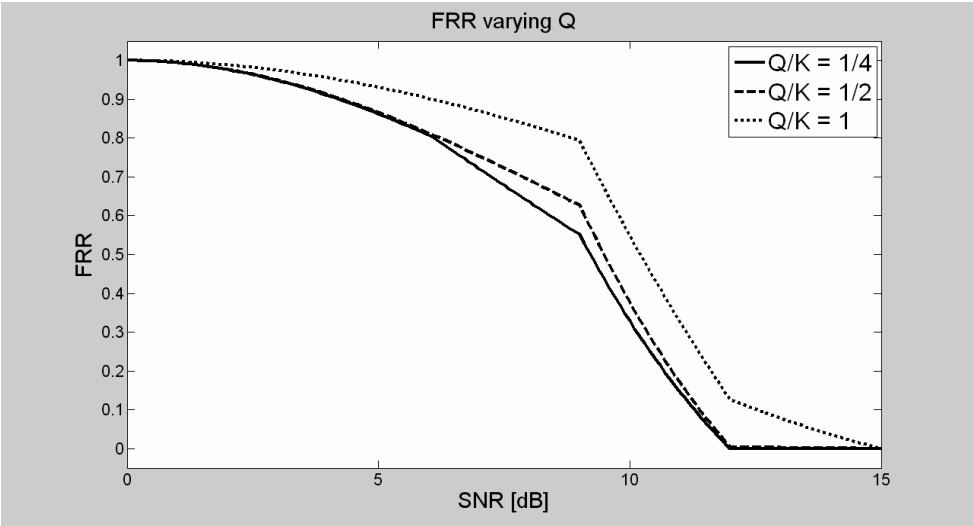

Fig.14 FRR vs SNR al variare del rapporto *Q* / *K*.

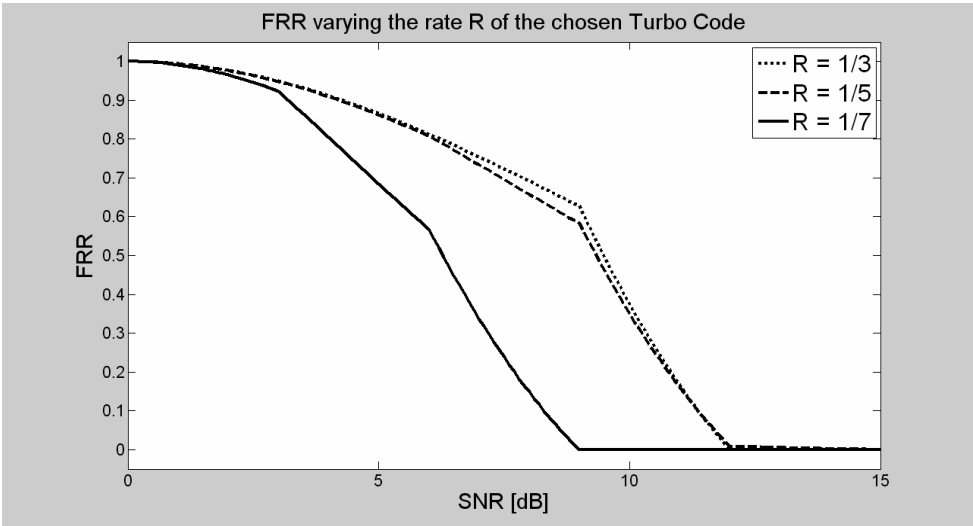

Fig.15 FRR vs SNR, al variare del rate *R* del Turbo Codice.

L'algoritmo di hash scelto è l'SHA-256, e così è stato selezionato il codice BCH (n, k) = (255, 99), dal momento in cui portavano al miglior compromesso tra carico computazionale e lunghezza della chiave di cifratura privata K_c, (che deve essere uguale a k). Il rapporto Q/K è senza dubbio un altro elemento di criticità.

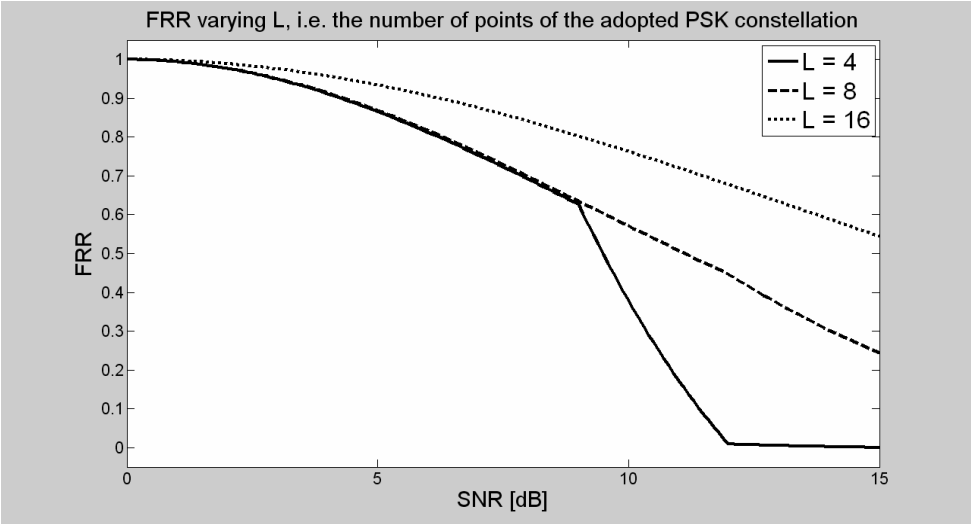

Fig.16 FRR versus SNR, varying the constellation order *L*.

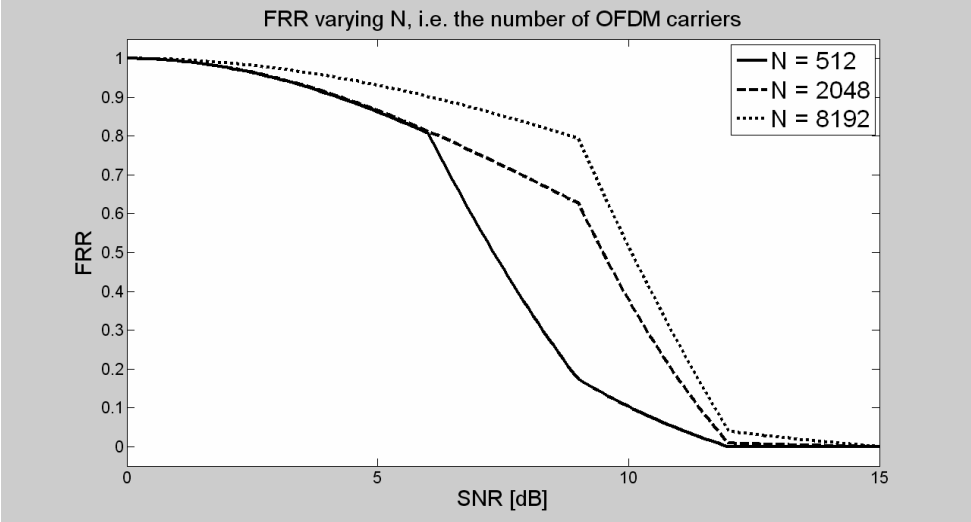

Fig. 17 FRR versus SNR, varying the number of OFDM carriers *N*.

Nelle figure 14, 15, 16 e 17 sono mostrate le curve dell'FRR (False Rejection Rate) rispetto all'SNR (Signal to Noise Ratio) al variare dei parametri Q, v, L e N.

3 Turbo Codici

I Turbo Codici [3], rappresentano una famiglia di codici convoluzionali, costruiti per mezzo di una particolare concatenazione di due codici sistematici ricorsivi, unti da un interleaving in generale non-uniforme. La loro decodifica fa appello ad un'elaborazione iterativa in cui sono presenti due decoder. Ognuno di questi può sfruttare il lavoro dell'altro allo step precedente, con l'aiuto del concetto originale di "informazione estrinseca".

I Turbo Codici sono stati però anche presentati da Benedetto e Montorsi come schemi di codifica PCCC (Parallel Concatenated Convolutional Codes). In particolare, in [4] viene proposto un metodo per valutare il limite superiore della probabilità di errore di un PCCC (mediata su tutti i possibili interleaving di una data lunghezza di vincolo). In [5] sono stati caratterizzati separatamente i contributi che l'interleaving e i codici convoluzionali danno alle prestazioni del PCCC nonchè presentate alcune linee-guida per il progetto dei codici convoluzionali. Quest'ultimo aspetto è inoltre trattato in modo più approfondito in [6].

La complessità del turbo-decodificatore (dal punto di vista implementativo) e la latenza della decisione dovuta al tempo di elaborazione cresce notevolmente all'aumentare del numero di rami (ognuno caratterizzato dal suo interleaver più convoluzionale) usati [7].

In [8] viene fornita una descrizione matematica dei processi di turbo-codifica e decodifica, utilizzando sia lo stimatore di simbolo MAP (Maximum A Posteriori) presentato in [9] e l'APRI-SOVA (A PRIori Soft Output Viterbi Algorithm) illustrato in [10].

Nel prossimo paragrafo l'approccio analitico introdotto in [8] viene esteso al caso in cui i turbo codici sono usati su modulazioni complesse L-arie, come la QAM.

3.1 Turbo Codici su modulazioni L-arie

Come mostrato in Fig.18, che rappresenta lo schema di turbo codifica base, la sequenza di informazione $\boldsymbol{u}$ è multiplata con le sequenze $\boldsymbol{c}_1$ e $\boldsymbol{c}_2$ al fine di ottenere il vettore binario $\boldsymbol{b}$. $\boldsymbol{c}_1$ e

$\boldsymbol{c}_2$ sono rispettivamente le uscite del primo e del secondo codificatore RSC (Recursive Systematic Convolutional). Più precisamente $\boldsymbol{c}_2$ è l'uscita del secondo RSC applicato però a $\boldsymbol{u}_1$, ossia una versione permutata di $\boldsymbol{u}$. Indicando con RSC1 e RSC2 il primo e il secondo codificatore RSC, se 1/2 è il rate sia di RSC1 che di RSC2, il rate globale sarà 1/3.

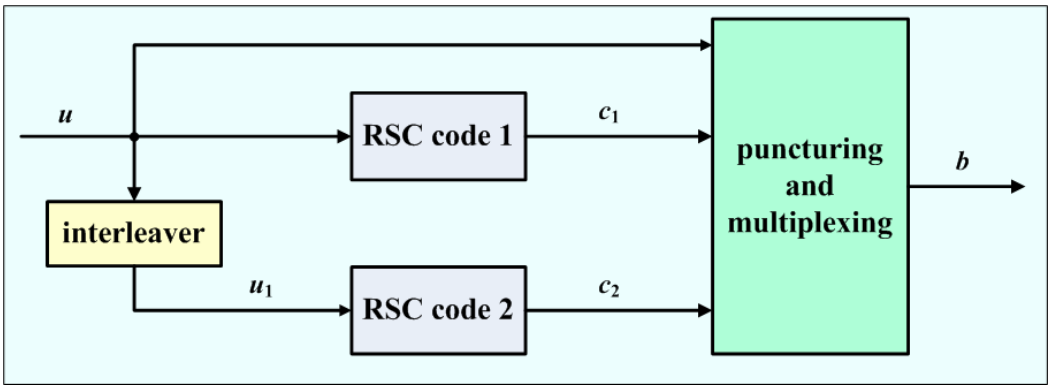

Fig.18 turbo codificatore

Un turbo codificatore prevede anche uno schema di punturazione sia su $\boldsymbol{c}_1$ che su $\boldsymbol{c}_2$. Si può fare riferimento ad un'implementazione semplificata di tale schema a punturazione in cui solo i bit dispari (o pari) di $\boldsymbol{c}_1$ e $\boldsymbol{c}_2$ sono effettivamente inviati. Questo accorgimento alza il rate da 1/3 ad 1/2. Ad ogni modo, la punturazione (che implica un'alterazione nel traliccio di decodifica) non sarà presa in considerazione nella presente derivazione analitica.

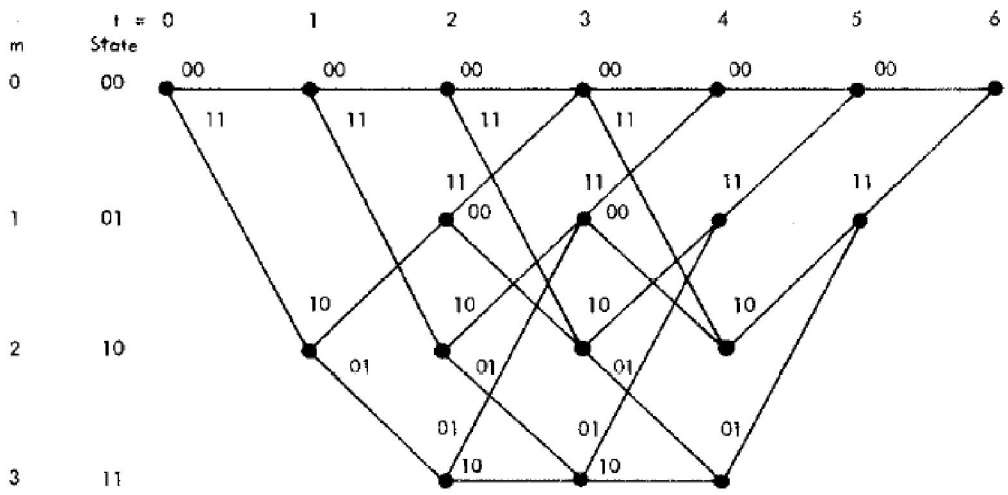

Fig.19 a CRSC decoding trellis example

Si assuma che RSC1 e RSC2 siano la parte di parità di uno stesso codice convoluzionale ricorsivo sistematico il cui rate è $R = k / q$, con memoria v. Un codice di questo tipo può essere trattato come un CRSC (Circular RSC) se, utilizzando una lunghezza di vincolo pari a Nk, tutte le parole di messaggio in ingresso sono date da una concatenazione seriale di τ sequenze a lunghezza k più v sequenze nulle, sempre a lunghezza k. Deve ovviamente risultare $N = \tau + v$. In Fig.19 è mostrato ad esempio un traliccio che fa riferimento a un caso con $\tau = 4$ e $v = 2$.

A questo punto, sia $k = \log_2(L)$, dove L è il numero di punti della costellazione adottata. In questo modo, ogni sequenza di k bit contenuta in $\boldsymbol{u}$, è riferita ad un simbolo.

Il frame di informazione è pertanto:

$$\begin{aligned} u &= \left(u_1, u_2, \cdots, u_n, \cdots, u_N\right), \\ u_n &= \left(u_n^{(1)}, u_n^{(2)}, \cdots, u_n^{(\ell)}, \cdots, u_n^{(k)}\right), \quad u_n^{(\ell)} \in \{0, 1\} \end{aligned} \qquad (15).$$

Sia con M[·] l'operatore di mappatura, che associa ogni k-upla a un punto della costellazione.

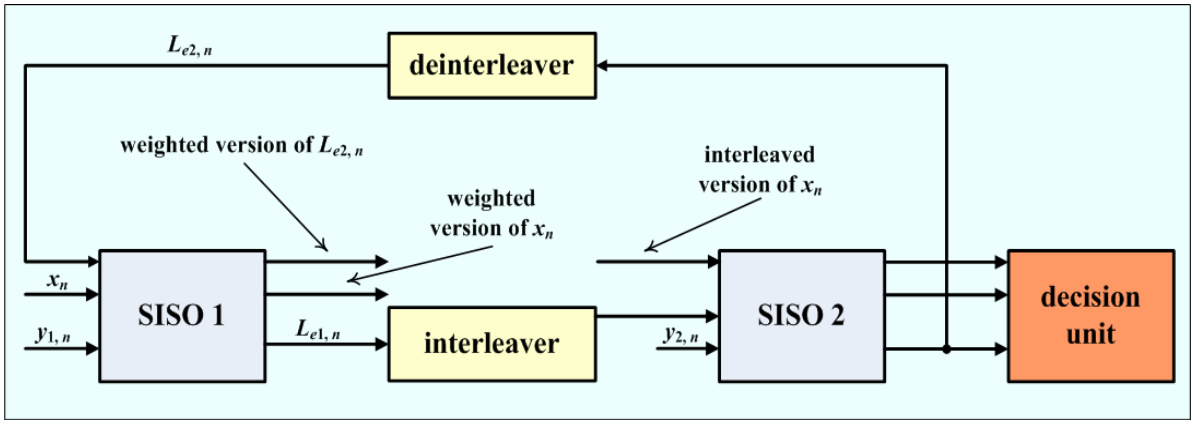

Fig.20 turbo decodificatore

In una versione non-punturata del turbo codice dovrebbe essere:

$$\begin{aligned} \boldsymbol{b} &= \left(u_1, c_{1,1}, c_{2,1}, u_2, c_{1,2}, c_{2,2}, \cdots, u_n, c_{1,n}, c_{2,n}, \cdots, u_N, c_{1,N}, c_{2,N}\right), \\ c_{\eta,n} &= \left(c_{\eta,n}^{(1)}, c_{\eta,n}^{(2)}, \cdots, c_{\eta,n}^{(\ell)}, \cdots, c_{\eta,n}^{(k)}\right), \quad c_{\eta,n}^{(\ell)} \in \{0, 1\}, \quad \eta \in \{1, 2\} \end{aligned} \qquad (16).$$

Tenendo conto anche della punturazione invece:

$$\boldsymbol{b} = \begin{cases} \left(u_1, c_{1,1}, u_2, c_{2,2}, \cdots, u_N, c_{2,N}\right), & \mathrm{mod}(N,2) = 0 \\ \left(u_1, c_{1,1}, u_2, c_{2,2}, \cdots, u_N, c_{1,N}\right), & \mathrm{mod}(N,2) = 1 \end{cases} \quad (17).$$

Durante la ricezione dell'*s*-esimo frame, tre tipi di ingressi soft vanno in ingresso al primo decodificatore convoluzionale, o meglio decodificatore SISO (Soft Input Soft Output) di Fig.20.

Il primo di questi è la versione rumorosa della componente sistematica:

$$\boldsymbol{x} = \left(x_1, x_2, \cdots, x_n, \cdots, x_N\right) \quad (18).$$

Se $\mu_n = \mu_n^c + j\mu_n^s$ è il punto della costellazione ricevuto per la *k*-upla u_n, definita in (15):

$$x_n = a_n \mu_n + w_n \quad (19),$$

dove a_n tiene conto dell'attenuazione dovuta al canale di comunicazione e w_n e un rumore guassiano, bianco, additivo e a valor medio nullo. L'equazione (19) descrive un canale AWGN puro quando $a_n = 1$.

Il secondo ingresso soft è dato dalla versione rumorosa della ridondanza dovuta a RSC1:

$$\boldsymbol{y_1} = \left(y_{1,1}, y_{1,2}, \cdots, y_{1,n}, \cdots, y_{1,N}\right) \quad (20).$$

Indicando con $\varepsilon_{1,n} = \varepsilon_{1,n}^c + j\varepsilon_{1,n}^s$ il punto della costellazione relativo a $c_{1,n}$ definita nella (16),

$$y_{1,n} = a_n \varepsilon_{1,n} + w_n \quad (21).$$

Il terzo input di tipo soft è dato dall'informazione estrinseca dovuta a RSC2:

$$\boldsymbol{L_{e2}} = \left(L_{e2,1}, L_{e2,2}, \cdots, L_{e2,n}, \cdots, L_{e2,N}\right) \quad (22).$$

Il primo decodificatore SISO lavora pertanto ogni frame sul seguente insieme di ingressi soft:

$$\boldsymbol{R_1} = \left(R_{1,1}, R_{1,2}, \cdots, R_{1,n}, \cdots, R_{1,N}\right) = \left(\boldsymbol{x}, \boldsymbol{y_1}, \boldsymbol{L_{e2}}\right) \quad (23),$$

essendo $\boldsymbol{x}$, $\boldsymbol{y_1}$ e $\boldsymbol{L_{e2}}$ rispettivamente dati dalle (18), (20) e (22).

Il secondo decodificatore SISO lavora invece sul seguente insieme di ingressi soft:

$$\boldsymbol{R_2} = \left(R_{2,1}, R_{2,2}, \cdots, R_{2,n}, \cdots, R_{2,N}\right) = \left(\Pi\left(\boldsymbol{x}\right), \boldsymbol{y_2}, \Pi\left(\boldsymbol{L_{e1}}\right)\right) \qquad (24),$$

dove $\Pi(\cdot)$ indica l'operatore di interleaving. In modo simile:

$$\boldsymbol{y_2} = \left(y_{2,1}, y_{2,2}, \cdots, y_{2,n}, \cdots, y_{2,N}\right) \qquad (25),$$

$$y_{2,n} = a_n \varepsilon_{2,n} + w_n \qquad (26).$$

In (26) $\varepsilon_{2,n} = \varepsilon_{2,n}^c + j\varepsilon_{2,n}^s$ è il punto della costellazione relativo alla k-upla $c_{2,n}$. Infine:

$$\boldsymbol{L_{e1}} = \left(L_{e1,1}, L_{e1,2}, \cdots, L_{e1,n}, \cdots, L_{e1,N}\right) \qquad (27).$$

Sia il SISO 1 che il 2 dovrebbero essere in grado di fornire, ogni tempo T, i rapporti logaritmici di verosimiglianza LLR (Log Likelihood Ratio) Λ_n, al fine di stabilire quale degli L possibili simboli è stato trasmesso. In effetti solo una parte di questa informazione, ossia la componente estrinseca, è inoltrata, dopo un'opportuna permutazione, al SISO relativo all'altro codice. Ciò avviene chiaramente durante tutte le oterazioni eccetto l'ultima. Infatti nel corso dell'ultima iterazione, mentre SISO 1 invia $\Pi(\boldsymbol{L_{e1}})$ a SISO 2, quest'ultimo inoltra all'unità di decisione tutta l'informazione contenuta negli LLR. Dal momento in cui $k = \log_2(L)$, il numero di stati del traliccio di decodifica è dato da $M = 2^{kv} = L^v$.

Come è stato detto precedentemente, il simbolo $u_n = u_{n_i}$, $i = 0, 1, \cdots, L-1$, è relativo all'elemento della costellazione $\mu_n = \mu_n^c + j\mu_n^s$. È chiaro come la decisione vada presa separatamente per il canale in fase e in quadratura.

Siano L_c e L_s il numero di possibili valori assunti rispettivamente da μ_n^c e μ_n^s. Estendendo l'equazione (8) di [8] al caso L-ario, se SISO 1 non dovesse comunicare con SISO 2 durante le iterazioni di decodifica, potrebbe tranquillamente decidere per la k-upla binaria $u_n = u_{n_i}$, $i = 0, 1, \cdots, L-1$, relativa a $\mu_n = \mu_n^c + j\mu_n^s$, scegliendo come $\mu_n^c = \mu_{n_i}^c$, $i \in \{0, 1, \cdots, L_c - 1\}$, quello che porta al più alto LLR:

$$\Lambda_{n_c}^i = \ln\left[\Pr\left(\mu_n^c = \mu_{n_i}^c \mid \boldsymbol{R_1}\right)\right], \; i \in \{0, 1, \cdots, L_c - 1\} \qquad (28),$$

mentre come $\mu_n^s = \mu_{n_i}^s$, $i \in \{0, 1, \cdots, L_s - 1\}$, viene scelto quello che porta al più alto:

$$\Lambda_{n_s}^i = \ln\left[\Pr\left(\mu_n^s = \mu_{n_i}^s \mid \boldsymbol{R_1}\right)\right], \; i \in \{0, 1, \cdots, L_s - 1\} \qquad (29).$$

L'intero processo descritto dalle equazioni (18)-(27), accade ora su due canali distinti (in fase e in quadratura):

$$\begin{aligned} \boldsymbol{R_1^c} &= \left(R_{1,1}^c, R_{1,2}^c, \cdots, R_{1,n}^c, \cdots, R_{1,N}^c\right) = \left(\boldsymbol{x^c}, \boldsymbol{y_1^c}, \boldsymbol{L_{e2}^c}\right) \\ \boldsymbol{R_1^s} &= \left(R_{1,1}^s, R_{1,2}^s, \cdots, R_{1,n}^s, \cdots, R_{1,N}^s\right) = \left(\boldsymbol{x^s}, \boldsymbol{y_1^s}, \boldsymbol{L_{e2}^s}\right) \end{aligned} \qquad (30),$$

essendo $\boldsymbol{x^c}$, $\boldsymbol{x^s}$, $\boldsymbol{y_1^c}$ e $\boldsymbol{y_1^c}$ rispettivamente Re($\boldsymbol{x}$), Im($\boldsymbol{x}$), Re($\boldsymbol{y_1}$) e Im($\boldsymbol{y_1}$), e con $\boldsymbol{x}$ e $\boldsymbol{y_1}$ definiti nella (18) e nella (20). Analogamente quindi:

$$\begin{aligned} \boldsymbol{R_2^c} &= \left(R_{2,1}^c, R_{2,2}^c, \cdots, R_{2,n}^c, \cdots, R_{2,N}^c\right) = \left(\Pi\left(\boldsymbol{x^c}\right), \boldsymbol{y_2^c}, \Pi\left(\boldsymbol{L_{e1}^c}\right)\right) \\ \boldsymbol{R_2^s} &= \left(R_{2,1}^s, R_{2,2}^s, \cdots, R_{2,n}^s, \cdots, R_{2,N}^s\right) = \left(\Pi\left(\boldsymbol{x^s}\right), \boldsymbol{y_2^s}, \Pi\left(\boldsymbol{L_{e1}^s}\right)\right) \end{aligned} \qquad (31).$$

Può risultare utile comprendere come gli LLR vengono calcolati e come l'informazione estrinseca può essere estratta da questi. Ricordando che $M = L^v$:

$$\begin{aligned} \Lambda_{n_c}^i &= \ln\left[\Pr\left(\mu_n^c = \mu_{n_i}^c \mid \boldsymbol{R_1^c}\right)\right] = \ln\left[\sum_{m=0}^{M-1} \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m, \boldsymbol{x^c}, \boldsymbol{y_1^c}, \boldsymbol{L_{e2}^c}\right)\right] = \\ &= \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^i\left(R_{1,n}^c, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \; i = 0, 1, \cdots, L_c - 1 \end{aligned} \qquad (32),$$

dove S_n è lo stato del codificatore convoluzionale all'istante n. La ricorsione "in avanti" [9] per un MAPSSE (Maximum A posteriori Probability Single Symbol Estimator) può essere espressa come:

$$\alpha_n^i(m) = \sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^r\left(R_{1,n}^c, m', m\right)\alpha_{n-1}^i\left(m'\right), \; i = 0, 1, \cdots, L_c - 1 \qquad (33).$$

Per quanto riguarda la ricorsione "all'indietro" [9] risulta invece:

$$\beta_n(m) = \sum_{m'=0}^{M-1} \sum_{r=0}^{L_c-1} \gamma_{n+1}^{r}\left(R_{1,n+1}^{c}, m', m\right)\beta_{n+1}(m') \qquad (34),$$

dove le probabilità di transizione tra i rami sono calcolabili tramite il seguente sistema:

$$\begin{aligned}\gamma_n^i\left(R_{1,n}^c, m', m\right) &= p\left(x_n^c \mid \mu_n^c = \mu_{n_i}^c\right) p\left(y_{1,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)\cdot \\ &\cdot p\left(L_{e2,n}^c \mid \mu_n^c = \mu_{n_i}^c\right)\cdot \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right),\ i = 0, 1, \cdots, L_c - 1\end{aligned} \qquad (35),$$

$$p\left(x_n^c \mid \mu_n^c = \mu_{n_i}^c\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(x_n^c - \mu_{n_i}^c\right)^2}{2\sigma^2}} \qquad (36),$$

$$p\left(y_{1,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(y_{1,n}^c - \varepsilon_{1,n_i}^c\right)^2}{2\sigma^2}} \qquad (37),$$

$$p\left(L_{e2,n}^c \mid \mu_n^c = \mu_{n_i}^c\right) = \frac{e^{\mu_{n_i}^c L_{e2,n_i}^c}}{\sum_{r=0}^{L_c-1} e^{\mu_{n_r}^c L_{e2,n_r}^c}} \qquad (38).$$

Alla prima iterazione $\left(L_{e2,n_0}^c, L_{e2,n_1}^c, \cdots, L_{e2,n_i}^c, \cdots, L_{e2,n_{L_c-1}}^c\right) = (0, 0, \cdots, 0, \cdots, 0)$ implica che la probabilità $p\left(L_{e2,n}^c \mid \mu_n^c = \mu_{n_i}^c\right) = 1/L_c$, il che esprime analiticamente il concetto di totale assenza di informazione a priori. Inoltre la quantità $\Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right)$ è l'analogo di $q_t\left(X, m, m'\right)$ definita in [9]. In altre parole $\Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right)$ rappresenta la probabilità che, essendo stata trasmessa la *k*-upla u_n relativa al punto della costellazione la cui parte reale è $\mu_{n_i}^c$, lo stato dopo $S_{n-1} = m'$ sia $S_n = m$. Diversamente da [9], queste grandezze non potranno essere pari solamente a 0 o 1. Infatti la sequenza u_n è quella relativa alla transizione da m a m', se $\mathrm{Re}\left(\mu_n\right) = \mu_n^c = \mu_{n_i}^c$ e, contestualmente se $\mathrm{Im}\left(\mu_n\right) = \mu_n^s = \mu_{n_j}^s$. Perciò, se il numero di sequenze con $\mu_n^c = \mu_{n_i}^c$ è n_ℓ, la probabilità di interesse è $1/n_\ell$.

Inserendo le relazioni (35)-(38) nell'equazione (32), si ottiene:

$$\begin{aligned}\Lambda_{n_c}^{i} &= \mu_{n_i}^{c} L_{e2,n_i}^{c} - \ln\left(\sum_{r=0}^{L_c-1} e^{\mu_{n_r}^{c} L_{e2,n_r}^{c}}\right) - \frac{1}{2\sigma^2}\left[\left(x_n^c - \mu_{n_i}^c\right)^2 + \sigma^2 \ln\left(2\pi\sigma^2\right)\right] + \\ &+ \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^i\left(y_{1,n}^c, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \quad i = 0, 1, \cdots, L_c - 1\end{aligned} \tag{39},$$

$$\begin{aligned}\gamma_n^i\left(y_{1,n}^c, m', m\right) &= p\left(y_{1,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)\cdot \\ &\cdot \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right), \quad i = 0, 1, \cdots, L_c - 1\end{aligned} \tag{40},$$

dove $p\left(y_{1,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)$ e $\Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right)$ sono definite in [8]. L'espressione (39) è basato sullo stimatore MAPSSE2, considerato in [8] profondamente differente dal MAPSSE1, che richiede inoltre la stima di σ_L.

Il terzo termine della (39) è proprio l'*i*-esima informazione estrinseca in fase, la quale, durante le varie iterazioni, è inoltrata all'altro decodificatore (SISO 2).

Per quanto riguarda il canale in quadratura si giunge alla seguente espressione:

$$\begin{aligned}\Lambda_{n_s}^{i} &= \mu_{n_i}^{s} L_{e2,n_i}^{s} - \ln\left(\sum_{r=0}^{L_s-1} e^{\mu_{n_r}^{s} L_{e2,n_r}^{s}}\right) - \frac{1}{2\sigma^2}\left[\left(x_n^s - \mu_{n_i}^s\right)^2 + \sigma^2 \ln\left(2\pi\sigma^2\right)\right] + \\ &+ \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_s-1} \gamma_n^i\left(y_{1,n}^s, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \quad i = 0, 1, \cdots, L_s - 1\end{aligned} \tag{41},$$

$$\begin{aligned}\gamma_n^i\left(y_{1,n}^s, m', m\right) &= p\left(y_{1,n}^s \mid \mu_n^s = \mu_{n_i}^s, S_n = m, S_{n-1} = m'\right)\cdot \\ &\cdot \Pr\left(\mu_n^s = \mu_{n_i}^s, S_n = m \mid S_{n-1} = m'\right), \quad i = 0, 1, \cdots, L_s - 1\end{aligned} \tag{42}.$$

Dal punto di vista di SISO 2 risulta, per il canale in fase:

$$\begin{aligned}\Lambda_{n_c}^{i} &= \ln\left[\Pr\left(\mu_n^c = \mu_{n_i}^c \mid \boldsymbol{R_2^c}\right)\right] = \ln\left[\sum_{m=0}^{M-1} \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m, \Pi\left(\boldsymbol{x^c}\right), \mathbf{y_2^c}, \Pi\left(\boldsymbol{L_{e1}^c}\right)\right)\right] = \\ &= \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^i\left(R_{2,n}^c, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \quad i = 0, 1, \cdots, L_c - 1\end{aligned} \tag{43},$$

dove $\alpha_n^i(m)$, $\beta_n(m)$, $\gamma_n^i\left(R_{2,n}^c, m', m\right)$, $p\left(y_{2,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)$ e $p\left(L_{e1,n}^c \mid \mu_n^c = \mu_{n_i}^c\right)$ sono definiti come nelle equazioni (33), (34), (35), (37) e (38).

Infine si ha:

$$\begin{aligned}\Lambda_{n_c}^i &= \mu_{n_i}^c L_{e1,n_i}^c - \ln\left(\sum_{r=0}^{L_c-1} e^{\mu_{n_r}^c L_{e1,n_r}^c}\right) - \frac{1}{2\sigma^2}\left[\left(x_n^c - \mu_{n_i}^c\right)^2 + \sigma^2 \ln\left(2\pi\sigma^2\right)\right] + \\ &+ \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^i\left(y_{2,n}^c, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \quad i = 0, 1, \cdots, L_c - 1\end{aligned} \tag{44},$$

$$\begin{aligned}\gamma_n^i\left(y_{2,n}^c, m', m\right) &= p\left(y_{2,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)\cdot \\ &\cdot \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right), \quad i = 0, 1, \cdots, L_c - 1\end{aligned} \tag{45},$$

$$\begin{aligned}\Lambda_{n_s}^i &= \mu_{n_i}^s L_{e1,n_i}^s - \ln\left(\sum_{r=0}^{L_s-1} e^{\mu_{n_r}^s L_{e1,n_r}^s}\right) - \frac{1}{2\sigma^2}\left[\left(x_n^s - \mu_{n_i}^s\right)^2 + \sigma^2 \ln\left(2\pi\sigma^2\right)\right] + \\ &+ \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_s-1} \gamma_n^i\left(y_{2,n}^s, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \quad i = 0, 1, \cdots, L_s - 1\end{aligned} \tag{46},$$

$$\begin{aligned}\gamma_n^i\left(y_{2,n}^s, m', m\right) &= p\left(y_{2,n}^s \mid \mu_n^s = \mu_{n_i}^s, S_n = m, S_{n-1} = m'\right)\cdot \\ &\cdot \Pr\left(\mu_n^s = \mu_{n_i}^s, S_n = m \mid S_{n-1} = m'\right), \quad i = 0, 1, \cdots, L_s - 1\end{aligned} \tag{47}.$$

I termini definiti nella (45) e nella (47) rappresentano le informazioni estrinseche che, nel corso delle iterazioni sono inoltrate al SISO 1. In realtà anche le L_c informazioni instrinseche in fase e le corrispettive L_s in quadratura, date dai primi due termini della (44) e della (46), contribuiscono, all'ultima iterazione, a decidere quale simbolo è stato trasmesso. In altre parole, la k-upla binaria u_n è scelta, all'ultima iterazione, come quella relativa al punto della costellazione $\mu_n = \mu_n^c + j\mu_n^s$, dove $\mu_n^c + j\mu_n^s$ è legata al più alto $\Lambda_{n_s}^i$ fra gli L_s della (46).

Un implementazione più snella del processo di turbo decodifica può prevedere il coinvolgimento del SOVA (Soft Output Viterbi Algorithm), il quale è accuratamente

illustrato in [10]. A differenza del MAP che è uno stimatore orientato al singolo simbolo, il SOVA, è orientato alla stima di sequenze di simboli.

3.2 Dai Turbo Codici alla Turbo Equalizzazione

La Turbo Equalizzazione è una tecnica di equalizzazione e decodifica che utilizza i concetti fondamentali dei turbo codici (come ad esempio l'informazione estrinseca de intrinseca) per combattere il problema dell'ISI (Inter Symbol Interference).

È bene inanzitutto ridefinire gli schemi base dei turbo codificatore e decodificatore, includendo la permutazione e l'interleaving su entrambi i rami di codifica e aggiungendo i blocchi relativi alla modulazione e al canale AWGN (si veda Fig.21).

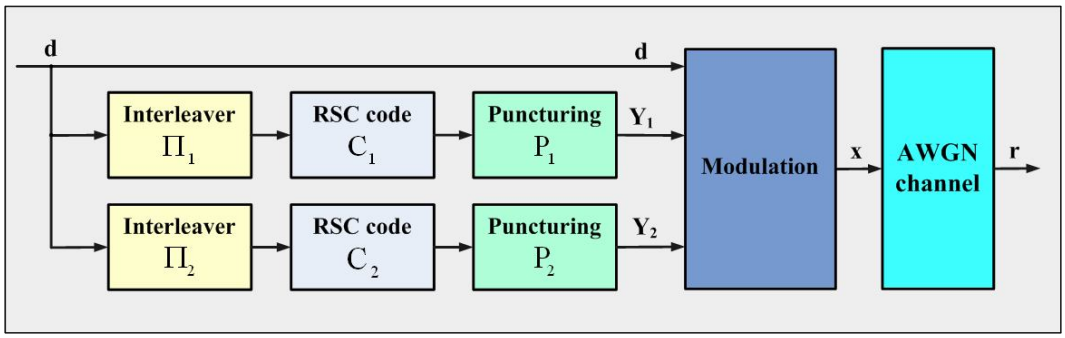

Fig.21 turbo codificatore generalizzato a due rami

In Fig.22 è invece riportato un turbo decodificatore che funziona su dati codificati attraverso lo schema di Fig.21.

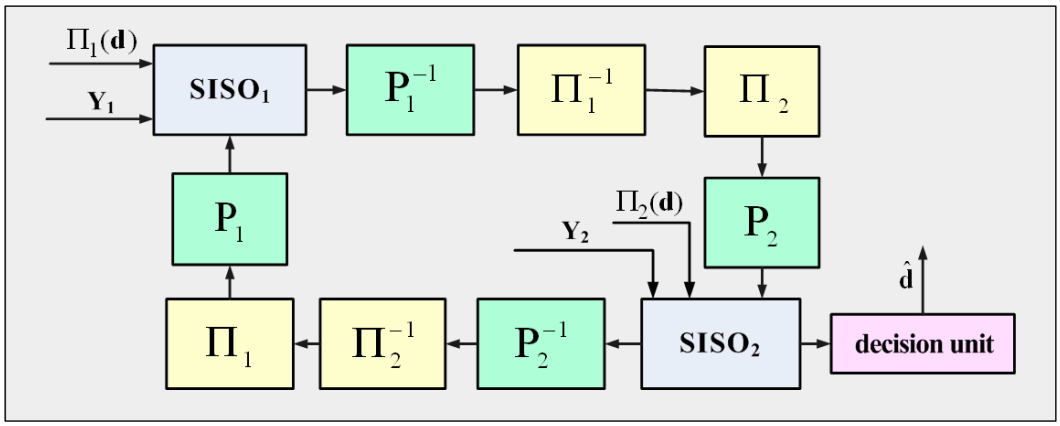

Fig.22 turbo decodificatore generalizzato (a due blocchi SISO)

Se il messaggio viene codificato, le sue proprietà algebriche possono essere sfruttate dal ricevitore anche per equalizzare il segnale (questo è quanto avviene in sostanza nella turbo equalizzazione). In queste condizioni infatti il ricevitore può essere costruito come una concatenazione di un decodificatore SISO ed un equalizzatore alla Viterbi. In [11] viene dimostrato come l'equalizzatore può essere modellato come un ulteriore elemento SISO.
In Fig.23 viene riportato uno schema di trasmissione che consente, in ricezione, di utilizzare la turbo equalizzazione.

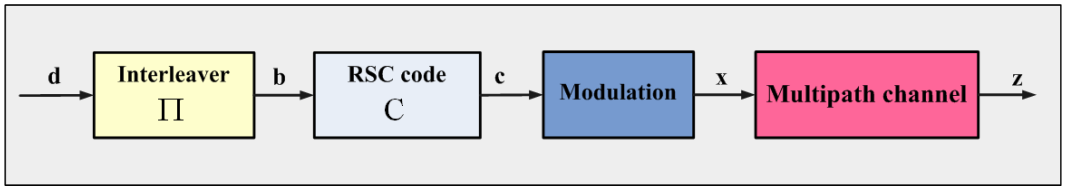

Fig.23 trasmissione di dati che possono essere ricevuti utilizzando la turbo equalizzazione

Facendo riferimento a un codice CRSC (n, k, N) (N è la lunghezza di vincolo) si può ritenere che l'i-esimo bit codificato b_i dipende dai bit del messaggio in accordo alla seguente regola:

$$b_i = \sum_{j=0}^{v} c_j^{(i)} a_{i-j}\,, \quad i = 1, 2, \cdots, n \qquad (48),$$

dove $c_j^{(i)}$ è il peso (che può valere o 0 o 1) relativo al j-esimo tap dell'i-esimo registro a scorrimento, e dove v è la lunghezza del registro a scorrimento (o la memoria del codice). Lo stato del codificatore è dunque rappresentato dai v bit a_{i-1}, a_{i-2}, ···, a_{i-v}. Da ognuno di questi stati, che è un nodo del traliccio relativo al CRSC, si può andare in due distinti stati allo step successivo, in accordo alla (48), e in base al valore del bit corrente a_i. In un simile traliccio, partendo dallo stato S_0, ossia $\begin{bmatrix} a_{i-1} & a_{i-2} & \cdots & a_{i-v} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$, si può assumere che si ritorni ad S_0 dopo N passi. Il numero di possibili stati (nodi del traliccio relativo al CRSC per un fissato istante) è pertanto 2^v.
La ricezione di dati su un canale affetto da multipath, presenterà un'ISI. In questo caso il simbolo ricevuto corrente sarà dato da una combinazione lineare di L simboli precedenti, come riassunto dall'equazione (49).

$$y_i = \sum_{j=0}^{L} \alpha_j x_{i-j} + w_i \qquad (49),$$

dove w_i rappresenta un campione di AWGN. Dal momento in cui y_i è un campione del segnale in banda-base, è chiaro come anche i coefficienti α_i siano in generale complessi, ossia introducano sia una distorsione di ampiezza che di fase (ovvero sia un attenuazione che una rotazione). Una volta che i coefficienti di canale [α_0, α_1, ⋯, α_L] sono stati stimati, può essere costruito un traliccio per il canale, partendo dall'osservazione che il corrente valore di y_i dipende dai coefficienti di canale e dai valori degli L simboli precedenti x_{i-1}, x_{i-2}, ⋯, x_{i-L}. Ogni stato del traliccio rappresenta una combinazione lineare di L simboli dove i pesi sono dati dai coefficienti di canale. Dunque, il numero dei possibili stati sarà M^L, dove M è il numero di punti della costellazione adottata. Da ognuno di questi stati, che rappresenta un nodo del traliccio, è possibile andare in M stati diversi al passo successivo, in accordo alla (49), e in base al valore del simbolo corrente x_i.

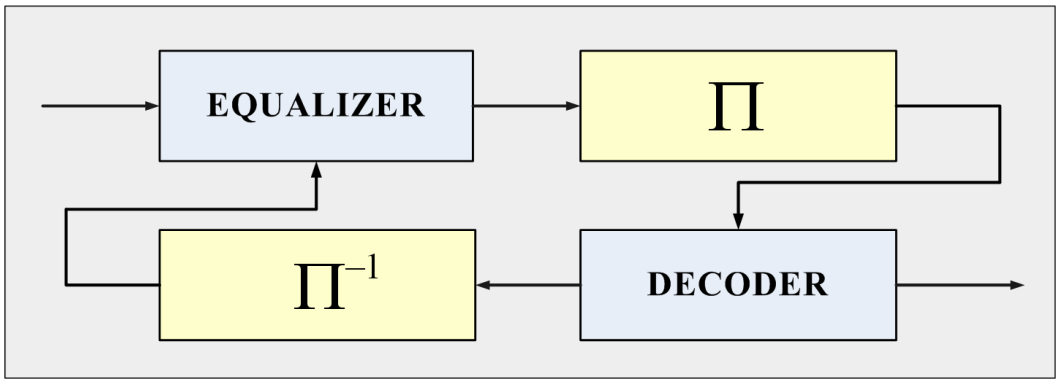

Fig.24 Turbo Equalizzatore

In [11] è dimostrato che l'algoritmo [9] può essere usato per equalizzare e decodificare iterativamente segnali trasmessi, come mostrato in Fig.23, su tratte soggette al fenomeno dei cammini multipli. In sostanza il decodificatore SISO legato al codice convoluzionale, fornisce al'equalizzatore alla Viterbi delle informazioni di natura estrinseca (in forma di LLR) e viceversa. Come ottenere l'informazione estrinseca dall'LLR globale è descritto in [11].

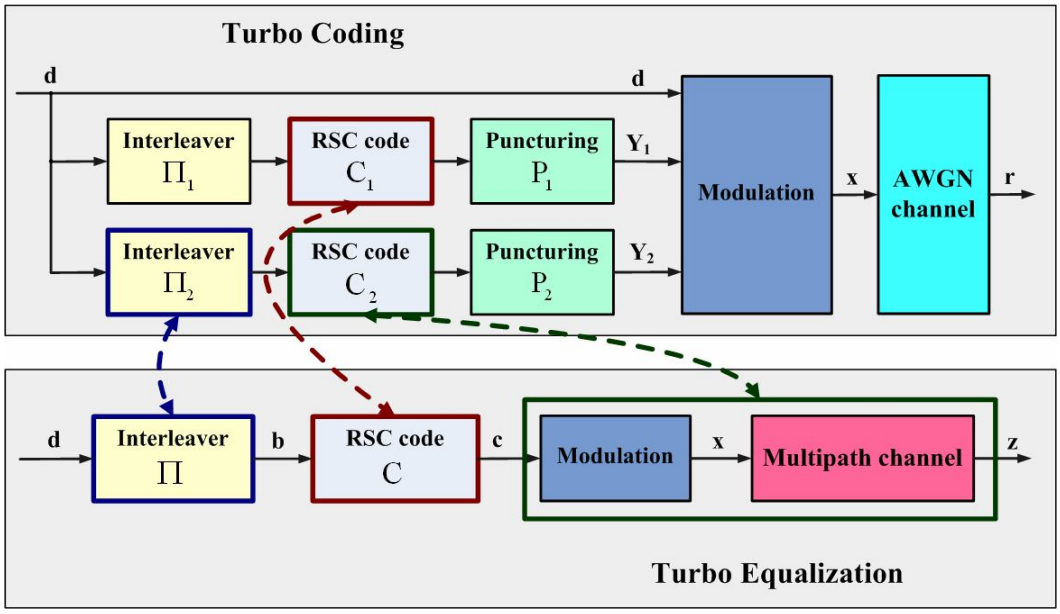

Fig.25 Turbo Codifica e Turbo Equalizzazione (lato trasmettitore)

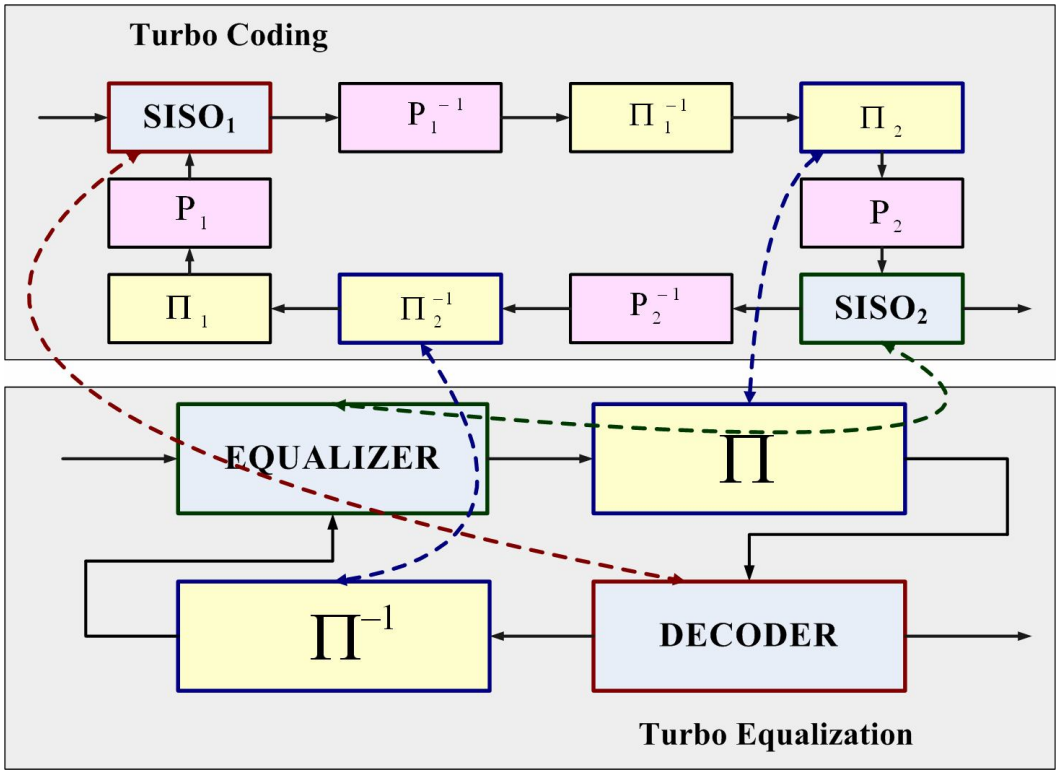

Fig.26 Turbo Codifica e Turbo Equalizzazione (lato ricevitore)

La Turbo Equalizzazione rappresenta veramente un'applicazione degli schemi classici di turbo codifica/decodifica al problema dell'equalizzazione di canale. Ciò può essere senza dubbio compreso mettendo a confronto le figure 21-22 con le 23-24. Più precisamente l'interleaver Π di Fig.23 gioca il ruolo dell'interleaver Π_2 di Fig.21; il codice RSC C di Fig.23 effettua le stesse operazioni del codice RSC C_1 di Fig.21; l' iper-blocco Modulazione + Canale affetto da Multipath di Fig.23 si comporta come il codice RSC C_2 di Fig.24. Queste corrispondenze sono schematizzate in Fig.25.
Dal punto di vista del ricevitore, l'equalizzatore e il decodificatore di Fig.24 giocano rispettivamente il ruolo del SISO 2 e del SISO 1 di Fig.22, come è illustrato in Fig.26.

3.3 Equalizzazione e l'approccio a chiavi di sessione

La Turbo Equalizzazione è stata brevemente introdotta al fine di proporre un nuovo algoritmo di sicurezza che funzioni correttamente anche in condizioni di canale estreme dal punto di vista dei cammini multipli.
La Turbo Equalizzazione richiede però la conoscenza del canale. Le sequenze di addestramento oppure portanti pilota, sono spesso usate per stimare la risposta impulsiva o in frequenza del canale. Mentre l'approccio nel dominio del tempo può essere usato sia nei sistemi a portante singola che in quelli multi-portante, la scelta di lavorare nel dominio della frequenza può essere effettuata solo nel secondo caso (ad esempio nelle comunicazioni OFDM). Il numero di gradi di libertà nella scelta della posizione nel tempo (o in frequenza) dei dati noti sia al trasmettitore è decisamente limitato: la sequenza di addestramento dovrebbe essere un preambolo di un pacchetto di dati; le portanti pilota dovrebbero essere il più possibile equispaziate per ottenere una stima del canale affidabile. La Turbo Equalizzazione su un OFDM avrebbe senso solo se il canale fosse talmente distorcente da introdurre un inteferenza tra frame OFDM visto che l'ISI non è presente grazie alla distribuzione in frequenza dell'informazione contenuta in un frame OFDM (che è dato da N simboli di dati). Per questi motivi dunque, lo schema di Fig.27 fa riferimento a una comunicazione che avviene su una portante singola (ad esempio una QPSK).

Per questo sistema l'insieme di chiavi di sessione private $\boldsymbol{k}_{\mathbf{P}}$ (si veda Fig.1), potrebbe decidere: a) la posizione temporale di una sequenza di addestramento per un sistema in cui I dati sono codificati tramite un convoluzionale e il ricevitore può pertanto usare la tecnica della turbo equalizzazione; b) la locazione in frequenza delle portanti pilota in un sistema di comunicazione basato sull'OFDM.

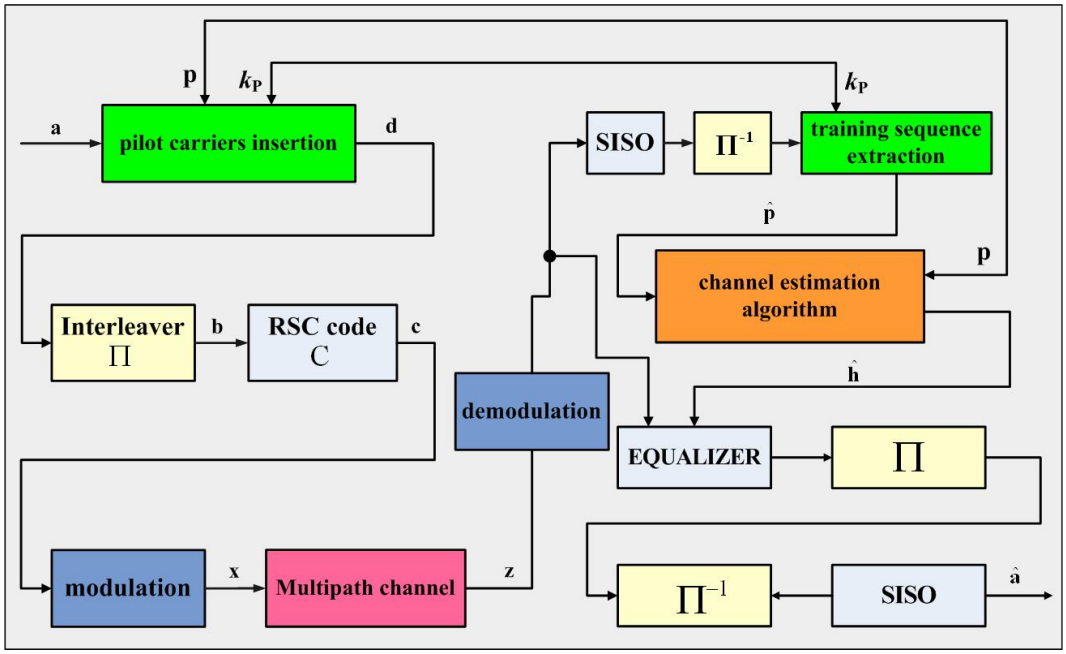

Fig.27 algoritmo di sicurezza basato su sequenze di addestramento e turbo equalizzazione

La soluzione a) è descritta in Fig.27 in cui l'insieme $\boldsymbol{k}_{\mathbf{P}}$ è costituito da due componenti: la chiave k_P, che stabilisce la posizione temporale della sequenza di addestramento e la sequenza di addestramento stessa **p**.

La soluzione b) è invece illustrata in Fig.28 in cui $\boldsymbol{k}_{\mathbf{P}}$ è data da: la chiave k_P, che determina la posizione in frequenza delle portanti pilota e **p**, ossia il vettore che colleziona il valore dei simboli trasmessi sulle pilota.

La proposta b) è stata inoltre testata in [12] ed un confronto, in termini di FRR (False Rejection Rate) tra sistemi OFDM con portanti equi-spaziate e spaziate in maniera pseudo-random è illustrato in Fig.29. Sia P il numero di portanti pilota, $p = [n_1, n_2, \cdots, n_P]$ sia invece il vettore degli indici (relativi alle posizioni in frequenza) di tali portanti, stabilito da k_P.

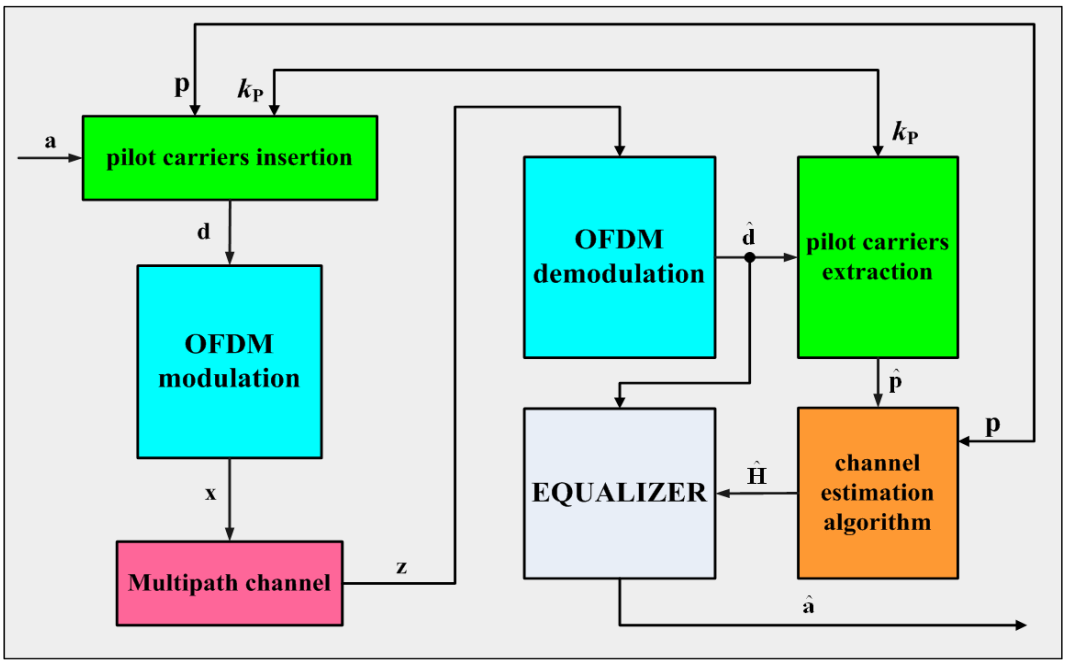

Fig.28 algoritmo di sicurezza basato sul posizionamento delle portanti pilota OFDM

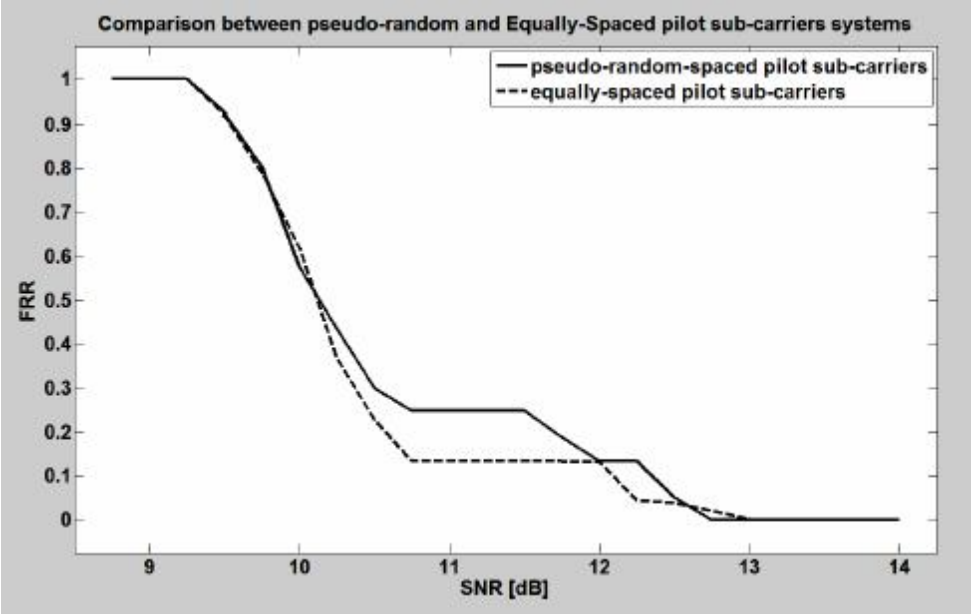

Fig.29 confronto tra sistemi OFDM con pilota equi-spaziate e a posizione pseudo-randomica

Siano invece raccolti nel vettore $\boldsymbol{a} = \left[a_{n_1}, \quad a_{n_2}, \quad \cdots, \quad a_{n_P} \right]$ i simboli che ci si aspetta di trovare sulle pilota stesse. È chiaro che, dal lato del ricevitore OFDM, se l'array dei simboli

recuperati sulle pilota è $\hat{\boldsymbol{a}} = \left[\hat{a}_{n_1} \quad \hat{a}_{n_2} \quad \cdots \quad \hat{a}_{n_P} \right]$ i campioni della risposta frequenza del canale negli indici $p = [n_1, n_2, \cdots, n_P]$ saranno $H_{n_i} = \hat{a}_{n_i} / a_{n_i}$, $i = 1, 2, \cdots, P$, mentre i rimanenti campioni dovranno necessariamente essere ricavati per interpolazione.

I risultati presentati in Fig.29 fanno riferimento ad un sistema QPSK-OFDM con $N_{ofdm} = 8192$ mentre $P = 256$ è il numero delle pilota.

4 Sicurezza e Codifica di Canale

L'approccio comunemente adottato dai più recenti standard IEEE, come l'IEEE 802.16e, per quanto riguarda la sicurezza degli accessi alle reti wireless, è quello di isolare la gestione delle autenticazioni dal livello fisico, implementandola ai livelli più alti della pila ISO OSI. Inoltre la verifica dell'autenticità della sorgente e dell'integrità del messaggio è affidata a soluzioni efficaci in scenari poco rumorosi. Queste tecniche sono basate sull'utilizzo di messaggi *Authentication Code* (AC), spesso indicati anche come messaggi hash, il cui valore dipende da due parametri funzionalmente distinti: il messaggio da autenticare e una chiave segreta, che si suppone sconosciuta all'eventuale "attaccante".

L'autenticità e l'integrità dei dati sono verificate controllando la sovrapposizione tra l'hash registrato e quello calcolato sul messaggio ricevuto dall'entità che provvede all'autenticazione (ad esempio un Acces Point). Il progetto dell'AC dovrebbe garantire che la ricostruzione della chiave segreta da parte dell'attaccante sia irrealizzabile dal punto di vista computazionale nel momento in cui questo possa venire in possesso di una o alcune coppie messaggio-hash. Poiché gli AC sono molto sensibili agli errori introdotti dal canale di comunicazione, devono essere caratterizzati da FEC (Forward Error Correction) molto efficaci e, in condizioni critiche da un sovra-stante protocollo ARQ (Automatic Repeat Request).

In [26] and [27] concatenated turbo coders whose interleaver and puncturing elements are selected on the basis of a secret session key have been proposed for joint FEC and security.

However, although using redundancy introduced by FEC for message authentication has a rather positive impact on the efficient use of the available bandwidth, strong error resilience and detection of deceptive attacks are still antithetic requirements. Therefore some form of trade off has to be applied to meet both constraints on BER and deception probability.

Recenti studi hanno mostrato come il fenomeno dei cammini multipli, temuto per la sua tendenza a introdurre interferenza intersimbolica, possa essere sfruttato non solo per

migliorare l'efficienza spettrale, come si fa nei sistemi MIMO (Multiple Input Multiple Output), ma perfino per aumentare il livello di sicurezza di una comunicazione [13].

Inoltre, l'implementazione della cifratura, mutua autenticazione, e algoritmi di protezione dell'integrità informativa, direttamente sui livelli 1 e 2 della pila ISO OSI, consente di ridurre drasticamente la ridondanza dovuta alla gestione della sicurezza, migliorare l'efficienza spettrale e impiegare delle contromisure più efficaci per malfunzionamenti di sistema o attacchi hijacking.

Per questo motivo, recentemente, diversi autori hanno approfondito le questioni inerenti all'utilizzo di codici a correzione d'errore per scopi di autenticazione, [14]-[27]. In [14] Kabatianskii et Al. hanno analizzato le relazioni teoriche tra gli AC e i normali codici di FEC. In [15] è stato proposto uno schema di autenticazione asimmetrica che fa uso del sistema di crittografia a chiave pubblica di McEliece, basato sui codici di Goppa. In seguito in [16] e [17] sono stati approfonditi alcuni metodi di firma digitale basati sui sistemi di McEliece. Un approccio più efficiente da un punto di vista computazionale è stato proposto da Rao e Nam, e consiste nel considerare privata la matrice generatrice pubblica adoperata nella tecnica McEliece, [22]-[25].

A tal scopo si propone l'adozione di una procedura di Neyman-Pearson al fine di decidere circa l'autenticità e l'integrità dei dati ricevuti nel momento in cui è impiegato un codice A-FEC (Authentication FEC). Conseguentemente, la verifica dell'autenticità e integrità del messaggio è ottenuta confrontando la probabilità a posteriori del messaggio decodificato e della parte hash, condizionate al segnale ricevuto rumoroso, con una soglia il cui valore è determinato dal massimo livello di probabilità di falso allarme definita come l'evento di respingimento di un messaggio autentico a causa del rumore.

Per supportare un progetto efficace, sono state investigate le relazioni fra i limiti delle probabilità di impersonificazione, sostituzione e di inganno, come definito da Simmons, [28], e i parametri strutturali che caratterizzano i codici A-FEC, come la cardinalità dell'insieme dei differenti codificatori che possono effettivamente essere utilizzati. Uno dei più importanti risultati ottenuti tramite l'analisi delle prestazioni e che l'introduzione di una permutazione random a monte di un codificatore di tipo *A*-FEC garantisce una sicurezza incondizionata nei

confronti di attacchi mirati all'autenticazione di uno specifico messaggio nel momento in cui la probabilità a priori del messaggio di ingresso sia uniforme.

4.1 Modello matematico

In accordo a [14], un codice A-FEC è definito in generale da una tripletta ($\mathcal{S}$,$\mathcal{E}$,$\mathcal{M}$) di insiemi finite e da una regola invertibile $f:\mathcal{S}\times\mathcal{E}\rightarrow\mathcal{M}$, dove $\mathcal{S}$ è l'insieme dello stato della sorgente, $\mathcal{E}$ il set delle regole di codifica, e per ogni $\mathbf{s}\in\mathcal{S}$ e $e\in\mathcal{E}$ la sequenza trasmessa **m** è definita ponendo $\mathbf{m}=f(\mathbf{s},\mathbf{e})$. Sia $\mathbf{y}^m\in\mathcal{Y}$ la corrispondente sequenza ricevuta attraverso il canale principale e $\mathbf{y}^w\in\mathcal{Y}$ quella intercettata dall'oppositore.

In generale un decodificatore A-FEC può essere definito attraverso una tripletta ($\mathcal{Y}$,$\mathcal{E}$,$\mathcal{S}$), una regola di decodifica $g:\mathcal{Y}\times\mathcal{E}\rightarrow\mathcal{S}$ e un test di verifica $v:\mathcal{Y}\times\mathcal{E}\rightarrow\{0,1\}$. Un messaggio decodificato $\hat{\mathbf{s}}=g(\mathbf{y}^m,e)$ è ritenuto autentico se $v(\mathbf{y}^m,e)=1$.

Come definito in [14], un codice A-FEC sistematico è una tripletta ($\mathcal{S}$,$\mathcal{E}_S$,$\mathcal{Z}$) di insiemi finiti e una regola di hash-parità $h:\mathcal{S}\times\mathcal{E}_S\rightarrow\mathcal{Z}$, dove per ogni $\mathbf{s}\in\mathcal{S}$ e una regola di codifica $e_S\in\mathcal{E}_S$ la sequenza trasmessa $\mathbf{m}=(\mathbf{s},\mathbf{z})\in\mathcal{S}\times\mathcal{Z}$ è definita ponendo $\mathbf{z}=h(\mathbf{s},e_S)$, [14].

Un decodificatore A-FEC sistematico è pertanto una n-upla ($\mathcal{X}$,$\mathcal{R}$,$\mathcal{E}$,$\mathcal{S}$), una regola $g:(\mathcal{X}\times\mathcal{R})\times\mathcal{E}\rightarrow\mathcal{S}$ e un test di verifica $v:(\mathcal{X}\times\mathcal{R})\times\mathcal{E}\rightarrow\{0,1\}$. Un messaggio decodificato $\hat{\mathbf{s}}=g(\mathbf{X}^m,\mathbf{R}^m;e)$ viene ritenuto autentico se $v(\mathbf{X}^m,\mathbf{R}^m;e)=1$.

Un A-Interleaver è definito da una coppia ($\mathcal{S}$,$\mathcal{E}_\Pi$) di stati finiti e da una permutazione $\Pi:\mathcal{S}\times\mathcal{E}_\pi\rightarrow\mathcal{S}$, dove per ogni $\mathbf{s}\in\mathcal{S}$ e per ogni regola di codifica $e_\Pi\in\mathcal{E}_\Pi$ la sequenza permutata $\mathbf{s}_\Pi\in\mathcal{S}$ è definita ponendo $\mathbf{s}_\Pi=\Pi(\mathbf{s},e_\Pi)$. Un codice A-FEC permutato sistematico è dato dalla cascata di un A-Interleaver e un codice A-FEC sistematico. Ovviamente, per un codice A-FEC permutato sistematico l'insieme delle regole è $\mathcal{E}=\mathcal{E}_\Pi\times\mathcal{E}_S$ e la sequenza trasmessa $\mathbf{m}=(\mathbf{s}_\Pi,\mathbf{z})\in\mathcal{S}\times\mathcal{Z}$ è definita ponendo $\mathbf{s}_\Pi=\Pi(\mathbf{s},e_\Pi)$ e

$\mathbf{z} = h(\mathbf{s}_\Pi, e_S) = h\left[\Pi(\mathbf{s}, e_\Pi), e_S\right]$.

Si indichino con $(\mathbf{x}^m, \mathbf{r}^m) \in \mathcal{X} \times \mathcal{R}$ e $(\mathbf{x}^w, \mathbf{r}^w) \in \mathcal{X} \times \mathcal{R}$ le uscite dal canale principale e quello di intercettazione quando un codificatore A-FEC permutato sistematico trasmette la sequenza $\mathbf{m} = (\mathbf{s}_\Pi, \mathbf{z})$. Il comportamento dei due canali è descritto dal punto di vista statistico tramite le distribuzioni condizionate

$$q^m(\mathbf{x}^m, \mathbf{r}^m, \mathbf{s}_\Pi, \mathbf{z}) = p_{\mathbf{X},\mathbf{R}|\mathbf{S}_\Pi,\mathbf{Z}}(\mathbf{x}^m, \mathbf{r}^m \,|\, \mathbf{s}_\Pi, \mathbf{z}) \qquad (50),$$

$$q^w(\mathbf{x}^w, \mathbf{r}^w, \mathbf{s}_\Pi, \mathbf{z}) = p_{\mathbf{X},\mathbf{R}|\mathbf{S}_\Pi,\mathbf{Z}}(\mathbf{x}^w, \mathbf{r}^w \,|\, \mathbf{s}_\Pi, \mathbf{z}) \qquad (51).$$

Un decodificatore A-FEC permutato sistematico è definibile dall'n-upla ($\mathcal{X}$,$\mathcal{R}$,$\mathcal{E}$,$\mathcal{S}$), dalla regola di decodifica $g:(\mathcal{X} \times \mathcal{R}) \times \mathcal{E} \to \mathcal{S}$ e da un test di verifica $v:(\mathcal{X} \times \mathcal{R}) \times \mathcal{E} \to \{0,1\}$, dove $\mathcal{E} = \mathcal{E}_\Pi \times \mathcal{E}_s$. Un messaggio decodificato $\hat{\mathbf{s}} = g(\mathbf{x}^m, \mathbf{r}^m; \mathbf{e})$ viene ritenuto autentico se $v(\mathbf{x}^m, \mathbf{r}^m; \mathbf{e}) = 1$, dove $\mathbf{e} = (e_\Pi, e_S)$.

Indichiamo con H_0 l'ipotesi che il messaggio sia stato alterato o corrotto e con H_1 l'ipotesi che il segnale ricevuto sia la versione rumorosa del messaggio originale autentico. Se si seguisse un approccio Bayesiano, la decisione riguardo l'autenticità e l'integrità dei dati ricevuti dovrebbe essere presa confrontando con una soglia il rapporto tra le probabilità a posteriore delle due ipotesi:

$$\frac{\Pr\left\{\hat{\mathbf{s}} \,\middle|\, \mathbf{x}^m, \mathbf{r}^m, \mathbf{e}\right\}}{\sum_{\mathbf{s}_i \in F} \Pr\left\{\mathbf{s}_i \,\middle|\, \mathbf{x}^m, \mathbf{r}^m\right\}} \qquad (52),$$

dove F è l'insieme di tutti i messaggi alterati/corrotti. Questa procedura risulta poco fattibile sia dal punto di vista computazionale che concettuale.

Dunque, ispirandoci al contesto radar, proponiamo, come in [29], l'adozione della procedura di Neyman-Pearson. Perciò la verifica dell'integrità e dell'autenticità diventa:

$$v(\mathbf{x}^m, \mathbf{r}^m; e) = \begin{cases} 1 & \log \Pr\left\{\hat{\mathbf{s}} / \mathbf{x}^m, \mathbf{r}^m, \mathbf{e}\right\} > \lambda \\ 0 & otherwise \end{cases} \quad (53).$$

In accordo al criterio di Neyman-Pearson, la soglia λ è determinata in base al massimo livello accettabile di probabilità di falso allarme definito come l'evento di respingere un messaggio autentico a causa del rumore.

Una volta che la soglia è stata fissata, è possibile scegliere i rimanenti parametri del codificatore A-FEC al fine di massimizzare la probabilità di rilevamento di un attacco.

4.1.1 Turbo Codici Permutati

Si consideri il caso, illustrato in Fig.30, in cui il codificatore sistematico sia un turbo codice parallelo dato da 2 interleaver Π_1, Π_2, in cascata con 2 codificatori convoluzionali sistematici ricorsivi (RSC) C_1, C_2 e due blocchi di punturazione P_1, P_2. La regola di codifica $e_S \in \mathcal{E}_S$ seleziona Π_1, Π_2, C_1, C_2, P_1, e P_2 in modo pseudo-random utilizzando un dizionario predefinito, basato sull'uscita di un generatore pilotato da una chiave segreta. Si consideri che, al fine di ridurre l'occupazione di memoria, è possibile mantenere bassa la probabilità di utilizzo di codificatori a basse prestazioni, utilizzando una gray-list.

L'ingresso del trasmettitore è dunque costituito dal testo cifrato $\mathbf{s}_\Pi$ che rappresenta la parte sistematica del turbo codificatore, e la sequenza di parità/hash punturata $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$.

Con riferimento alla Fig.31, il ricevitore è costituito da un turbo decodificatore, [30], in cascata con un blocco che calcola la verosimiglianza della coppia (testo cifrato, hash) fornendo perciò un indicatore di autenticazione soft. La decisione hard è ottenuta confrontando il funzionale di verosimiglianza con la soglia.

Per canali AWGN, indicando con μ(.) la mappatura dall'ingresso binario al punto della costellazione della specifica modulazione digitale impiegata a livello fisico, si ha:

$$\boldsymbol{x} = \mu\left(\boldsymbol{s}_{\boldsymbol{\Pi}}\right) + \boldsymbol{n} \quad (54),$$

$$\boldsymbol{r}_1 = \mu\left(\boldsymbol{z}_1\right) + \boldsymbol{n}_1 \quad (55),$$

$$\boldsymbol{r}_2 = \mu\left(\boldsymbol{z}_2\right) + \boldsymbol{n}_2 \qquad (56).$$

dove $\mathbf{n}=\mathbf{n}_I+j\mathbf{n}_Q$, $\mathbf{n}_1 = \mathbf{n}_{1_I} + j\mathbf{n}_{1_Q}$, e $\mathbf{n}_2 = \mathbf{n}_{2_I} + j\mathbf{n}_{2_Q}$ sono campioni di rumore bianco gaussiano circolarmente complesso che modellano l'inviluppo complesso del rumore ricevuto, le cui componenti in fase e in quadratura sono rispettivamente indicate tramite i pedici *I* e *Q*. Dunque il rapporto logaritmico di verosimiglianza $\log\Lambda(\hat{\mathbf{s}};\mathbf{x},\mathbf{r}_1,\mathbf{r}_2)$ del messaggio decodificato, condizionato al segnale ricevuto, può essere scritto come segue:

$$\begin{aligned}\log\Lambda\left(\hat{\boldsymbol{s}};\boldsymbol{x},\boldsymbol{r}_1,\boldsymbol{r}_2\right) &= \log\Pr\left\{\boldsymbol{x},\boldsymbol{r}_1,\boldsymbol{r}_2/\hat{\boldsymbol{s}}\right\} = -\frac{M}{2}\log 2\pi\sigma_N^2 - \\ &-\frac{1}{2\sigma_N^2}\left[\boldsymbol{x}-\mu(\hat{\boldsymbol{s}}_\Pi)\right]^\dagger\left[\boldsymbol{x}-\mu(\hat{\boldsymbol{s}}_\Pi)\right]- \qquad (57), \\ -\frac{1}{2\sigma_N^2}&\left[\boldsymbol{r}_1-\mu(\hat{\boldsymbol{z}}_1)\right]^\dagger\left[\boldsymbol{r}_1-\mu(\hat{\boldsymbol{z}}_1)\right]-\frac{1}{2\sigma_N^2}\left[\boldsymbol{r}_2-\mu(\hat{\boldsymbol{z}}_2)\right]^\dagger\left[\boldsymbol{r}_2-\mu(\hat{\boldsymbol{z}}_2)\right]\end{aligned}$$

dove † indica l'operatore Hermitiano, σ_N^2 la varianza del rumore ricevuto, $\hat{\mathbf{z}}_1$ e $\hat{\mathbf{z}}_2$ le parità/hash, eventualmente punturate, sequenze estimate corrispondenti $\hat{\mathbf{s}}$, e *M* è la somma delle dimensioni dei vettori complessi $\mathbf{x}$, $\mathbf{r}_1$, e $\mathbf{r}_2$.

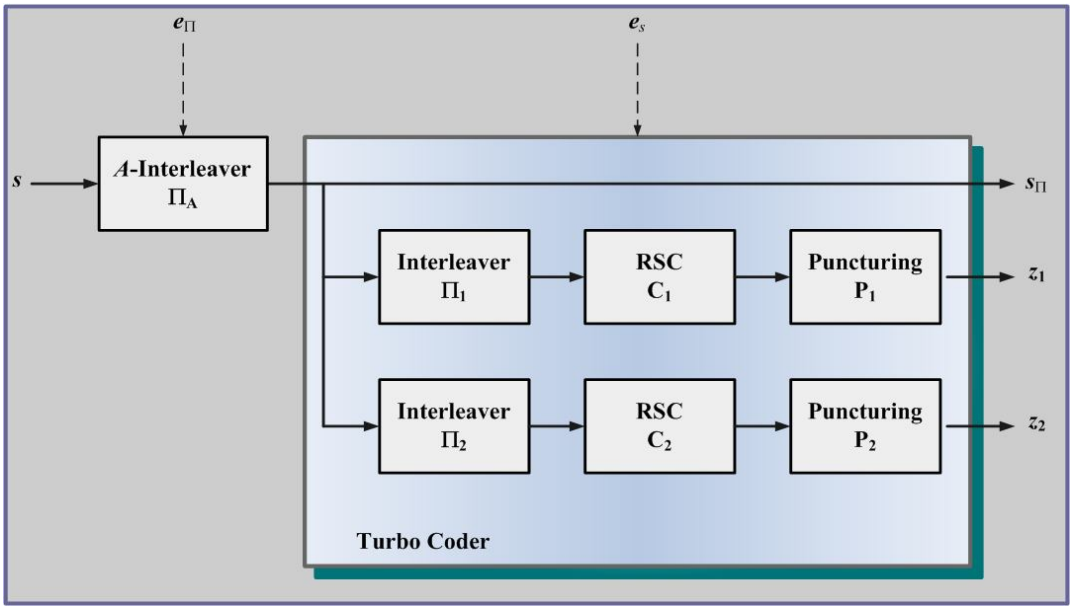

Fig.30 Turbo Codificatore Permutato *A*-FEC

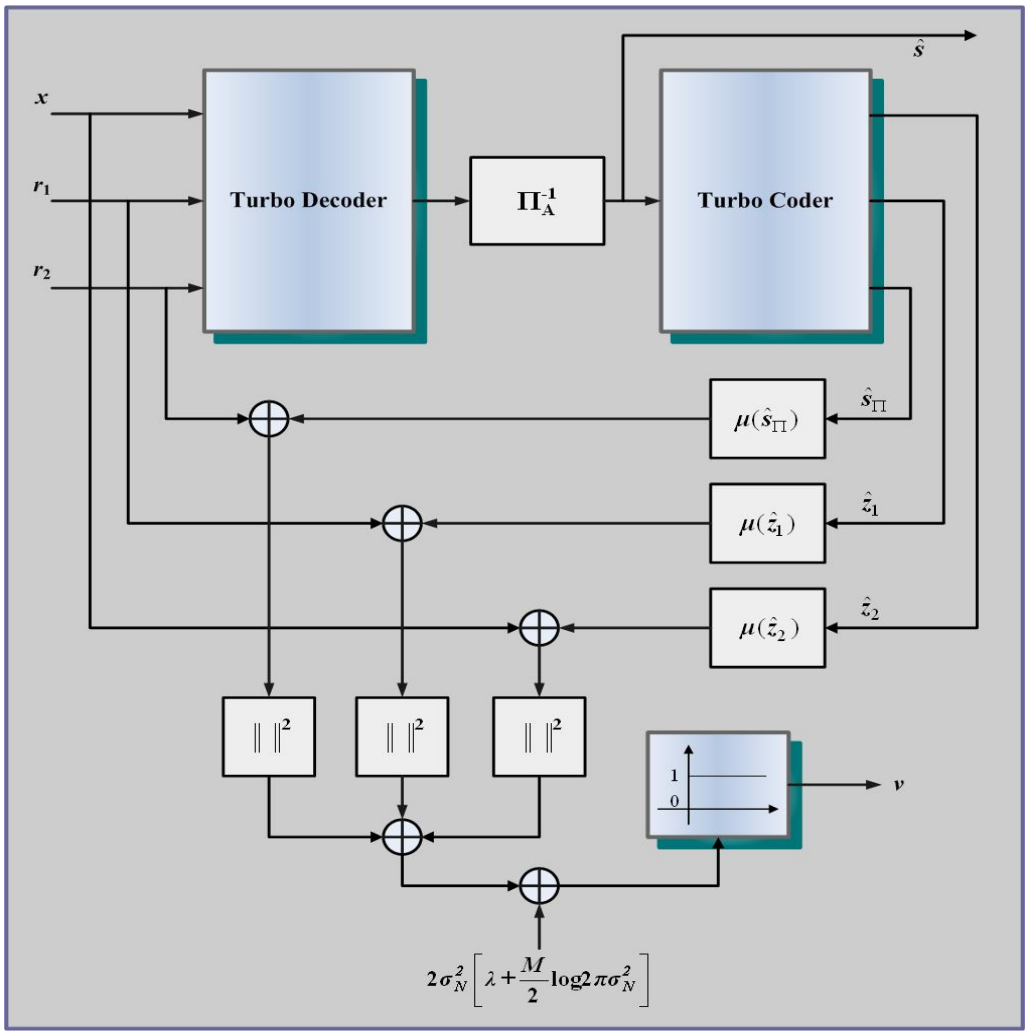

Fig.31 Turbo Decodificatore permutato *A*-FEC

Per valutare la probabilità di falso allarme P_{fa}, si osservi che, se il BER all'uscita del decodificatore è sufficientemente basso, $\hat{\mathbf{s}} \cong \mathbf{s}$, dunque, sotto l'ipotesi H_1, il funzionale di verosimiglianza logaritmico (57), può essere opportunamente approssimato come segue:

$$\log \Lambda\left(\hat{\mathbf{s}}; \boldsymbol{x}, \boldsymbol{r}_{1,} \boldsymbol{r}_2 / H_1\right) \cong -\frac{M}{2} \log 2\pi\sigma_N^2 - \frac{\nu}{2} \qquad (58),$$

dove

$$\nu = \frac{1}{\sigma_N^2} \sum_{i=1}^{M} \left(n_{I_k}^2 + n_{Q_k}^2\right) \qquad (59)$$

è una variabile aleatoria a distribuzione chi-quadratica con $2M$ gradi di libertà. Pertanto:

$$P_{fa} = \int_{-2\lambda - M\log 2\pi\sigma_N^2}^{\infty} p_{\chi_{2M}^2}(x)dx = \frac{\gamma\left(M, -\lambda - M\log 2\pi\sigma_N^2\right)}{(M-1)!} \qquad (60),$$

dove $\gamma(k,z)$ è la Gamma incompleta superiore:

$$\gamma(a,x) = \int_x^{\infty} t^{a-1}e^{-t}dt \qquad (61).$$

Dunque il test di autenticità e integrità può essere effettuato mediante un'inversione numerica della (60). Affinché la soglia sia adattativa è necessario pertanto la stima della densità spettrale del rumore.

4.2 Analisi delle prestazioni

La massima probabilità di successo di un attacco per impersonificazione, indicate come P_I, è formalmente definita, in base a quanto detto prima, come:

$$\begin{aligned} P_I &= \underset{(\boldsymbol{x}^m,\boldsymbol{r}^m)\in\mathcal{X}\times\mathcal{R}}{Max} \Pr\left\{v(\boldsymbol{x}^m,\boldsymbol{r}^m,\boldsymbol{e}) = 1\right\} = \\ &= \underset{(\boldsymbol{x}^m,\boldsymbol{r}^m)\in\mathcal{X}\times\mathcal{R}}{Max} \log\Pr\left\{\hat{\boldsymbol{s}} / \boldsymbol{x}^m,\boldsymbol{r}^m,\boldsymbol{e}\right\} > \lambda \end{aligned} \qquad (62).$$

Per una data sequenza $(\mathbf{x}^m,\mathbf{r}^m)$, si indichi con $\mathcal{E}^m(\mathbf{x}^m,\mathbf{r}^m)$ l'insieme delle regole di codifica A-FEC che passano il test di autenticità e di integrità, più precisamente:

$$\mathcal{E}^m(\boldsymbol{x}^m,\boldsymbol{r}^m) = \left\{\boldsymbol{e}\in\mathcal{E}\,\middle|\,\log\Pr\left\{\hat{\boldsymbol{s}} / \boldsymbol{x}^m,\boldsymbol{r}^m,\boldsymbol{e}\right\} > \lambda\right\} \qquad (63).$$

e con $\mathcal{E}_S^m(\mathbf{x}^m,\mathbf{r}^m)$ l'insieme delle regole di codifica A-FEC sistematiche che passano il test di autenticità e di integrità, ossia:

$$\mathcal{E}_S^m(\boldsymbol{x}^m,\boldsymbol{r}^m) = \left\{e_S\in\mathcal{E}_S\,\middle|\,\underset{e_\Pi\in\mathcal{E}_\Pi}{Max}\log\Pr\left\{\hat{\boldsymbol{s}} / \boldsymbol{x}^m,\boldsymbol{r}^m,(e_\Pi,e_S)\right\} > \lambda\right\} \qquad (64).$$

In modo simile, sia $\mathcal{E}^{NF}(\mathbf{s},\mathbf{z})$ l'insieme delle regole di codifica che passano il test di autenticità e di integrità, nel caso di assenza di rumore (Noise Free), ovvero:

$$\mathcal{E}^{NF}(\mathbf{s}_\Pi, \mathbf{z}) = \left\{ \boldsymbol{e} = (e_\Pi, e_S) \in \mathcal{E} \,\middle|\, \mathbf{s}_\Pi = \Pi(\boldsymbol{s}, e_\Pi),\ \mathbf{z} = h\left[\mathbf{s}_\Pi, e_S\right] \right\} \tag{65}$$

e con $\mathcal{E}_S^{NF}(\mathbf{s}_\Pi, \mathbf{z})$ il corrispondente insieme di regole di codifica A-FEC sistematica, ossia:

$$\mathcal{E}_S^{NF}(\mathbf{s}_\Pi, \mathbf{z}) = \left\{ e_S \in \mathcal{E}_S \,\middle|\, \mathbf{z} = h(\mathbf{s}_\Pi, e_S) \right\} \tag{66}.$$

Dal momento in cui Π è una permutazione, $\mathcal{E}^{NF}(\mathbf{s}_\Pi, \mathbf{z}) = \mathcal{E}_\Pi \times \mathcal{E}_S^{NF}(\mathbf{s}_\Pi, \mathbf{z})$.

Dunque per canali AWGN, $\left\{ \mathbf{x} \,\middle|\, \mathbf{x} = \mu(\mathbf{s}), \mathbf{s} \in \mathcal{S} \right\} \subseteq \mathcal{X}$, $\left\{ \mathbf{r} \,\middle|\, \mathbf{r} = \mu(\mathbf{z}), \mathbf{z} \in \mathcal{Z} \right\} \subseteq \mathcal{R}$, e $\left|\mathcal{E}_S^{NF}(\mathbf{s}_\Pi, \mathbf{z})\right| \le \left|\mathcal{E}_S^{m}(\mathbf{s}_\Pi, \mathbf{z})\right|$, dove $|\cdot|$ indica la cardinalità di un insieme. Dunque, assumendo come in [14] che l'avversario selezionerà la regola di codifica in modo completamente random:

$$P_I = \underset{(\boldsymbol{x}^m, \boldsymbol{r}^m) \in X \times R}{Max} \frac{\left|\mathcal{E}^m(\boldsymbol{x}^m, \boldsymbol{r}^m)\right|}{\left|\mathcal{E}\right|} \ge \underset{(\mathbf{s}_\Pi, \mathbf{z}) \in \mathcal{S} \times \mathcal{Z}}{Max} \frac{\left|\mathcal{E}^m\left(\mu(\mathbf{s}_\Pi), \mu(\mathbf{z})\right)\right|}{\left|\mathcal{E}_\Pi\right|\left|\mathcal{E}_S\right|} = \underset{(\mathbf{s}_\Pi, \mathbf{z}) \in \mathcal{S} \times \mathcal{Z}}{Max} \frac{\left|\mathcal{E}_S^{NF}(\mathbf{s}_\Pi, \mathbf{z})\right|\left|\mathcal{E}_\Pi\right|}{\left|\mathcal{E}_\Pi\right|\left|\mathcal{E}_S\right|} \tag{67}.$$

Il termine più a destra della (67) è la probabilità di successo di un attacco a impersonificazione per canali privi di rumore, [13],

$$P_I^{NF} = \underset{(\boldsymbol{s}_\Pi, \boldsymbol{z}) \in \mathcal{S} \times \mathcal{Z}}{Max} \frac{\left|\mathcal{E}_S^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z})\right|}{\left|\mathcal{E}_S\right|} \tag{68}.$$

Perciò

$$P_I \ge P_I^{NF} \ge \frac{1}{\left|\mathcal{E}_S\right|} \tag{69}.$$

Com'era prevedibile la probabilità di successo di un attacco a impersonificazione cresce all'aumentare del rumore ed ha come estremo inferiore l'inverso del numero di differenti codificatori.

Nell'attacco a sostituzione, l'avversario osserva l'uscita del canale d'intercettazione $(\mathbf{x}^w, \mathbf{r}^w)$ e rimpiazza il messaggio originale $\mathbf{s}$ con $\mathbf{s}'$. Per rendere la più alta possibile la probabilità di successo, l'intruso dovrà utilizzare una sequenza di hash-parità $\mathbf{z}'$ che rende massima la cardinalità dell'intersezione tra $\mathcal{E}^w(\mathbf{x}^w, \mathbf{r}^w)$ e $\mathcal{E}^{NF}(\mathbf{s}', \mathbf{z}')$, in altre parole,

$$\boldsymbol{z}' = Arg \underset{\mathbf{z}\in\mathcal{Z}}{Max}\left|\mathcal{E}^{NF}(\boldsymbol{s}',\boldsymbol{z})\cap\mathcal{E}^{w}(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\right| \qquad (70).$$

Dunque, in accordo a [13], la massima probabilità di successo di un attacco a sostituzione, indicate con P_S, è:

$$\begin{aligned} P_S &= \underset{\substack{(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\in\mathcal{X}\times\mathcal{R}\\(\boldsymbol{x}',\boldsymbol{r}')\in\mathcal{X}\times\mathcal{R}\\(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\neq(\boldsymbol{x}',\boldsymbol{r}')}}{Max}\Pr\left\{v(\boldsymbol{x}',\boldsymbol{r}',\boldsymbol{e})=1\middle|\boldsymbol{x}^{w},\boldsymbol{r}^{w}\right\} \geq \underset{\substack{(\boldsymbol{s}',\boldsymbol{z}')\in\mathcal{S}\times\mathcal{Z}\\(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\in\mathcal{X}\times\mathcal{R}\\\boldsymbol{s}'\neq\hat{\boldsymbol{s}}^{w}}}{Max}\frac{\Pr\left\{\mathcal{E}^{NF}(\boldsymbol{s}',\boldsymbol{z}')\cap\mathcal{E}^{w}(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\right\}}{\Pr\left\{\mathcal{E}^{w}(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\right\}} = \\ &= \underset{\substack{(\boldsymbol{s}_{\Pi},\boldsymbol{z}')\in\mathcal{S}\times\mathcal{Z}\\(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\in\mathcal{X}\times\mathcal{R}\\\boldsymbol{s}'\neq\hat{\boldsymbol{s}}^{w}}}{Max}\frac{\left|\mathcal{E}_S^{NF}(\boldsymbol{s}_{\Pi},\boldsymbol{z}')\cap\mathcal{E}_S^{w}(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\right|}{\left|\mathcal{E}_S^{w}(\boldsymbol{x}^{w},\boldsymbol{r}^{w})\right|} \end{aligned} \qquad (71).$$

Nell'attacco a impersonificazione, come è definito in [14], lo scopo dell'avversario è quello di passare il test di autenticità e integrità per qualsiasi messaggio,senza prendersi cura della semantica associata al messaggio che massimizza sia P_I che P_{Su}. Ad ogni modo, in molte situazioni pratiche l'intruso ha la necessità di autenticare un messaggio ben preciso, diciamo $\tilde{s}$. In questo caso, tenendo a mente che Π è una permutazione, la probabilità di successo per un attacco ad impersonificazione mirato al particolare messaggio $\tilde{s}$, ossia $\tilde{P}_I(\tilde{s})$, può essere valutata come segue:

$$\tilde{P}_I(\tilde{\boldsymbol{s}}) = \underset{\substack{\mathbf{z}\in\mathcal{Z}\\e_{\Pi}\in\mathcal{E}_{\Pi}}}{Max}\frac{\left|\mathcal{E}_S^{m}\left(\mu\left[\Pi(\tilde{\boldsymbol{s}},e_{\Pi})\right],\mu(\boldsymbol{z})\right)\right|}{\left|\mathcal{E}_{\Pi}\right|\left|\mathcal{E}_S\right|} \qquad (72).$$

Pertanto

$$\underset{\tilde{\mathbf{s}}\in\mathcal{S}}{Max}\,\tilde{P}_I(\tilde{\boldsymbol{s}}) = \frac{P_I}{\left|\mathcal{E}_{\Pi}\right|} \qquad (73).$$

In modo analogo, sia $\tilde{P}_S(\tilde{s})$ la probabilità di autenticare con success oil messaggio $\tilde{s}$, avendo osservato $(\mathbf{x}^{w},\mathbf{r}^{w})$ all'uscita del canale di intercettazione. Dunque, può essere facilmente verificato che:

$$\underset{\tilde{\mathbf{s}}\in\mathcal{S}}{Max}\,\tilde{P}_S(\tilde{\boldsymbol{s}}) = \frac{P_S}{\left|\mathcal{E}_{\Pi}\right|} \qquad (74).$$

In sostanza la permutazione iniziale consente di ridurre di un fattore $|\mathcal{E}_\Pi|$ la probabilità di impersonificazione e sostituzione per attacchi mirati.

Inoltre, l'attaccante può solamente osservare l'uscita rumorosa a valle dell'A-interleaver insieme alla sequenza di hash-parità associata. Perciò se gli stati della sorgente sono equiprobabili, l'informazione mutual di $(\mathbf{X}^w,\mathbf{R}^w)$ e E_Π è nulla e l'entropia di E_Π condizionata a $(\mathbf{X}^w,\mathbf{R}^w)$ equivale all'entropia a priori di E_Π. Infatti,

$$\begin{aligned} & p_{\mathbf{X}^w,\mathbf{R}^w|E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|e_\Pi) = \\ & = \sum_{s_\Pi\in\mathcal{S}}\sum_{e_S\in\mathcal{E}_S} p_{\mathbf{X}^w,\mathbf{R}^w|\mathbf{S}_\Pi,E_S,E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|\boldsymbol{s}_\Pi,e_S,e_\Pi)P_{\mathbf{S}_\Pi|E_\Pi}(\boldsymbol{s}_\Pi|e_\Pi)P_{E_S|\mathbf{S}_\Pi,E_\Pi}(e_S|\boldsymbol{s}_\Pi,e_\Pi) \end{aligned} \quad (75).$$

Osservando che $E_{\underline{S}}$ è statisticamente indipendente da E_Π e $\mathbf{S}_\Pi$, e in virtù della (51), per i canali AWGN risulta:

$$p_{\mathbf{X}^w,\mathbf{R}^w|\mathbf{S}_\Pi,E_S,E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|\boldsymbol{s}_\Pi,e_S,e_\Pi) = p_{\mathbf{X}^w,\mathbf{R}^w|\mathbf{S}_\Pi,E_S}(\boldsymbol{x}^w,\boldsymbol{r}^w|\boldsymbol{s}_\Pi,e_S) = q^w\left[\boldsymbol{x}^w,\boldsymbol{r}^w,\boldsymbol{s}_\Pi,h(\boldsymbol{s}_\Pi,e_S)\right] \quad (76)$$

$$p_{\mathbf{X}^w,\mathbf{R}^w|E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|e_\Pi) = \sum_{s_\Pi\in\mathcal{S}}\left\{\sum_{e_S\in\mathcal{E}_S} q^w\left[\boldsymbol{x}^w,\boldsymbol{r}^w,\boldsymbol{s}_\Pi,h(\boldsymbol{s}_\Pi,e_S)\right]P_{E_S}(e_S)\right\}P_{\mathbf{S}_\Pi|E_\Pi}(\boldsymbol{s}_\Pi|e_\Pi) \quad (77).$$

Di converso, indicando con Π^{-1} la permutazione inversa di Π, si ha:

$$P_{\mathbf{S}_\Pi|E_H}(\boldsymbol{s}_\Pi|e_\Pi) = P_{\mathbf{S}}\left[\Pi^{-1}(\boldsymbol{s}_\Pi,e_\Pi)\right] \quad (78).$$

Pertanto, se $\mathbf{S}$ è uniformemente distribuita, allora anche $\mathbf{S}_\Pi$ lo è ed è anche statisticamente indipendente da E_Π, inoltre l'osservazione $(\mathbf{X}^w,\mathbf{R}^w)$ e E_Π sono mutuamente statisticamente indipendenti, il che implica che la mutual informazione è nulla. Infatti $P_{\mathbf{S}_\Pi|E_H}(\mathbf{s}_\Pi|e_\Pi) = P_{\mathbf{S}_\Pi}(\mathbf{s}_\Pi) = |\mathcal{E}|^{-1}$, e pertanto

$$p_{\mathbf{X}^w,\mathbf{R}^w|E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|e_\Pi) = p_{\mathbf{X}^w,\mathbf{R}^w}(\boldsymbol{x}^w,\boldsymbol{r}^w) = \frac{1}{|\mathcal{S}|}\sum_{s_\Pi\in\mathcal{S}}\left\{\sum_{e_S\in\mathcal{E}_S} q^w\left[\boldsymbol{x}^w,\boldsymbol{r}^w,\boldsymbol{s}_\Pi,h(\boldsymbol{s}_\Pi,e_S)\right]P_{E_S}(e_S)\right\} \quad (79).$$

Infine osserviamo che quando la selezione della permutazione è puramente randomica:

$$H(E_\Pi|\mathbf{X}^w,\mathbf{R}^w) = H(E_\Pi) = \log_2|\mathcal{E}_\Pi| \quad (80).$$

Dunque un attaccante che voglia portare a termine un intrusione mirata per sostituzione, non è in grado di acquisire dal canale di osservazione alcuna informazione circa la permutazione da utilizzare. Ecco perchè un codificatore sistematico A-FEC è incondizionatamente sicuro nei confronti di attacchi a sostituzione mirati.

Per comunicazioni che avvengono su canali rumorosi il limite di Simmon specifica, [14], [31], [32],

$$P_I \geq 2^{-H(E)+H(E|\mathbf{X}^m,\mathbf{R}^m)} \quad (81),$$

$$P_S \geq 2^{-H(E|\mathbf{X}^w,\mathbf{R}^w)} \quad (82).$$

Dunque, dalla (81) e dalla (82) segue che:

$$P_I P_S \geq 2^{-H(E)-\left[H(E|\mathbf{X}^w,\mathbf{R}^w)-H(E)+H(E|\mathbf{X}^m,\mathbf{R}^m)\right]} \quad (83).$$

Per questo motivo quando il canale di osservazione è più rumoroso di quello principale, $H(E|\mathbf{X}^m,\mathbf{R}^m) \leq H(E|\mathbf{X}^w,\mathbf{R}^w)$ e per la probabilità di inganno (deception) $P_D = \mathrm{Max}(P_I, P_S)$ si ha $P_D \geq |\mathcal{E}|^{-1/2}$.

4.2.1 Analisi delle prestazioni dei turbo codici permutati

Nell'analisi delle prestazioni dei turbo codici permutati, l'attenzione sarà concentrata sul calcolo di $|\mathcal{E}|$ il cui valore risulta fondamentale per il calcolo delle probabilità di sostituzione, impersonificazione e inganno, discusse nel paragrafo precedente.

A tal scopo si consideri come gli interleaver puramente randomici possono portare a scarse prestazioni dei turbo codici, specialmente in tutte le applicazioni in cui è richiesta una lunghezza di vincolo ridotta. Per ridurre il tasso d'errore sul bit sono stati proposti diversi algoritmi di progettazione delle permutazioni su cui si basano gli interleaving. Purtroppo, un'analisi esaustiva della cardinalità di ogni classe d'interleaver esistente in letteratura, è eccessivamente onerosa, e richiede una notevole quantità di tempo.

Per questo motivo, solamente la classe degli interleaving semi-randomici, sarà indagata. Tali interleaver sono anche chiamati in letteratura con il nome di *S*-random [33]. In effetti questi

interleaving consentono di migliorare notevolmente le prestazioni dei turbo codici soprattutto per lunghezze di vincolo molto contenute. Essi inoltre sono caratterizzati da una complessità computazionale accettabile.

In prima approssimazione, in un interleaver S-random, la cui distanza minima è S, l'n-esimo indice di permutazione non è altro che un intero i_n generato in modo casuale. Se $|i_n - i_{n-k}| > S$, per $k = 1, 2, \cdots, S$, allora i_n viene selezionato come n-esimo indice, altrimenti viene generato un nuovo intero. Gli interleaver puramente randomici rappresentano dunque un caso particolare degli S-random per $S = 1$.

Sia $\mathcal{E}_\pi^{(N,S)}$ l'insieme degli interleaver S-random con lunghezza di vincolo pari a N; si indichi invece con $\mathcal{E}_c^{(N,v)}$ l'insieme dei codici RSC con rate $1/v$ e lunghezza di vincolo N e infine sia $\mathcal{E}_p^{(N,\rho)}$ l'insieme degli schemi di punturazione con lunghezza di vincolo N e numero di sopravvissuti pari a ρ. Per un turbo codice permutato che utilizzi gli interleaver S-random si ha che $e_\pi \in \mathcal{E}_\pi = \mathcal{E}_\pi^{(N,1)}$, $\Pi_1 \in \mathcal{E}_\pi^{(N,S_1)}$, $\Pi_2 \in \mathcal{E}_\pi^{(N,S_2)}$, $C_1 \in \mathcal{E}_c^{(N,v_1)}$, $C_2 \in \mathcal{E}_c^{(N,v_2)}$, $P_1 \in \mathcal{E}_p^{(N,\rho_1)}$, $P_2 \in \mathcal{E}_p^{(N,\rho_2)}$. Dunque la cardinalità dell'insieme $\mathcal{E} = \mathcal{E}_\pi \times \mathcal{E}_s$ può essere espressa come segue:

$$|\mathcal{E}| = |\mathcal{E}_\pi||\mathcal{E}_s| = |\mathcal{E}_\pi|\left|\mathcal{E}_\pi^{(N,S_1)}\right|\left|\mathcal{E}_\pi^{(N,S_2)}\right|\left|\mathcal{E}_c^{(N,v_1)}\right|\left|\mathcal{E}_c^{(N,v_2)}\right|\left|\mathcal{E}_p^{(N,\rho_1)}\right|\left|\mathcal{E}_p^{(N,\rho_2)}\right| \qquad (84).$$

In [34] è fornita una definizione di interleavier S-random più generale: un interleaver S-random garantisce che, se due bit di ingresso si trovano ad una distanza S_1 non possono essere mappati ad una distanza inferiore a S_2 dall'operazione di interleaving. Quindi, considerando due indici i e j, definiti in modo tale che $0 \le |i - j| \le S_1$, la regola di progetto impone che $|\Pi(i) - \Pi(j)| > S_2$. In genere $S_1 = S_2 = S$.

Da questo momento in poi gli interleaver S-random con $S_1 = S_2 = S$ saranno chiamati "full-S-random" mentre quelli con $S_1 = 1$ e $S_2 = S$ "half-S-random". Mentre gli interleaver half-S-random rappresentano formalmente un caso particolare dei full-S-random, è l'insieme di questi ultimi a essere in realtà un sottoinsieme degli half-S-random. Il numero di chiavi necessarie a indicizzare gli half-S-random è infatti maggiore a quello necessario per i full-S-

random. Ciò implica pertanto che gli interleaving di autenticazione half-*S*-random sono più sicuri degli *A*-interleaver full-*S*-random.

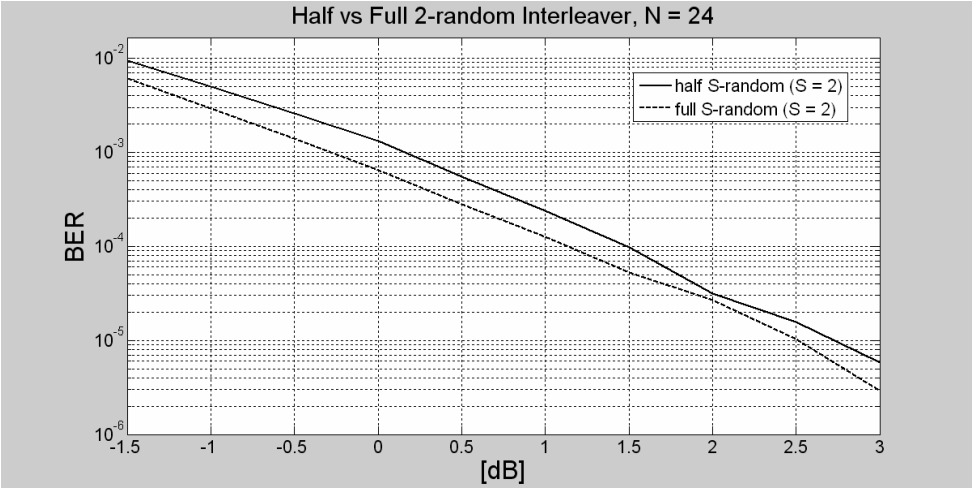

Fig.32 prestazioni degli half-*S*-random e dei full *S*-random (*N* = 24, *S* = 2, *v* = 5).

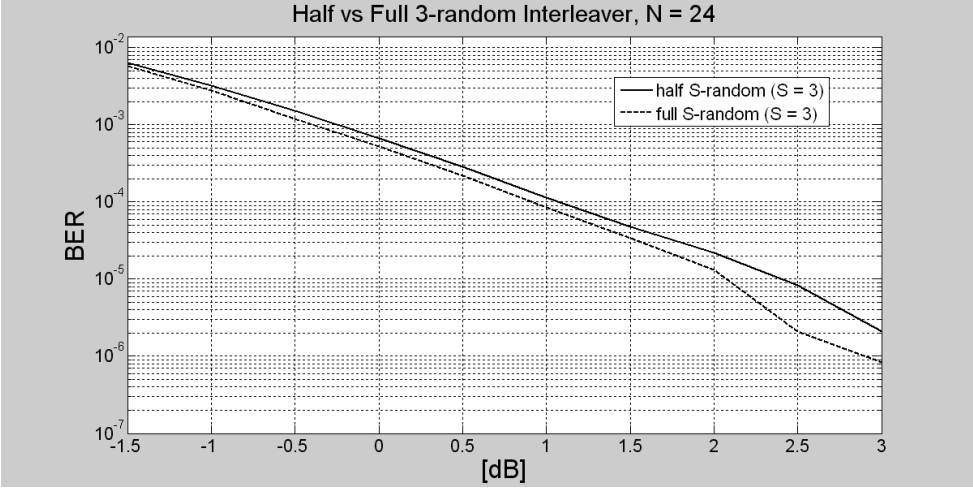

Fig.33 prestazioni degli half-*S*-random e dei full *S*-random (*N* = 24, *S* = 3, *v* = 5).

Come mostrano però le curve del BER riportate nelle figure 32 e 33, le prestazioni dei full-*S*-random sono, seppur di poco, superiori a quelle degli half-*S*-random.

Per calcolare i termini $\left|\mathcal{E}_{\pi}^{(N,S_1)}\right|$ e $\left|\mathcal{E}_{\pi}^{(N,S_2)}\right|$ della (84), può essere utilizzata un'approssimazione, la cui bontà sarà inoltre adeguatamente provata.

In seguito, $\left|\mathcal{E}_{\pi}^{(N,S)}\right|$ indicherà la cardinalità dell'insieme degli interleaver full *S*-random con lunghezza di vincolo *N*, e $\left|\mathcal{E}_{\pi_{HALF}}^{(N,S_1)}\right|$ quella dell'insieme degli half *S*-random con la stessa lunghezza di vincolo.

Una breve descrizione di come viene progettato un interleaver full-*S*-random (l'half *S*-random non è che un suo caso particolare), consentirà di comprendere meglio il metodo approssimato proposto. Sia $A = \{1, 2, \cdots, N\}$ l'alfabeto degli indici. Indicando con $A^{(j)}$ l'insieme dei superstiti disponibili prima che venga operata la *j*-esima scelta, si può senza ombra di dubbio ritenere che $A^{(1)} = A$. Sia $n_s^{(j)}$ il numreo di possibili selezioni che possono essere effettuate al *j*-esimo passo, mentre i_j è il superstite scelto al passo numero *j*. Ovviamente al primo passo $n_s^{(1)} = |A^{(1)}| = N$. Al secondo invece:

$$n_s^{(2)} = \left|\left\{i_2 \in A^{(2)} / \left|i_2 - i_1\right| > S\right\}\right|, \quad A^{(2)} = A - \{i_1\} \qquad (85).$$

Al terzo:

$$n_s^{(3)} = \left|\left\{i_3 \in A^{(3)} / \left|i_3 - i_2\right| > S \cap \left|i_3 - i_1\right| > S\right\}\right|, \quad A^{(3)} = A - \{i_1, i_2\} \qquad (86),$$

e così via, fino al passo numero $S + 1$, quando:

$$\begin{aligned} n_s^{(S+1)} &= \left|\left\{i_{S+1} \in A^{(S+1)} / \left|i_{S+1} - i_S\right| > S \cap \left|i_{S+1} - i_{S-1}\right| > S \cap \cdots \left|i_{S+1} - i_1\right| > S\right\}\right|, \\ A^{(S+1)} &= A - \{i_1, i_2, \cdots, i_S\} \end{aligned} \qquad (87).$$

Dall'*S*+2-esimo passo il numero delle selezioni precedenti (che indica anche la riduzione dell'alfabeto dei superstiti) è maggiore di *S* ossia del numero di condizioni che devono essere soddisfatte da i_j, ossia:

$$\begin{aligned} n_s^{(S+2)} &= \left|\left\{i_{S+2} \in A^{(S+2)} / \left|i_{S+2} - i_{S+1}\right| > S \cap \left|i_{S+2} - i_S\right| > S \cap \cdots \left|i_{S+1} - i_2\right| > S\right\}\right|, \\ A^{(S+2)} &= A - \{i_1, i_2, \cdots, i_{S+1}\} \end{aligned} \qquad (88).$$

Dunque al generico *j*-esimo passo, con $j > S$, risulta:

$$n_s^{(j)} = \left|\left\{ i_j \in A^{(j)} / \left|i_j - i_{j-1}\right| > S \cap \left|i_j - i_{j-2}\right| > S \cap \cdots \left|i_j - i_{j-S}\right| > S \right\}\right|, \quad A^{(j)} = A - \left\{ i_1, i_2, \cdots, i_{j-1} \right\} \qquad (89).$$

Per valutare $\left|\mathcal{E}_\pi^{(N,S)}\right|$ può essere usata la

$$\left|\mathcal{E}_\pi^{(N,S)}\right| = \prod_{j=1}^{N} \overline{n}_s^{(j)} \qquad (90)$$

in cui $\overline{n}_s^{(j)}$ è il valore atteso di $n_s^{(j)}$ rispetto a tutte le possibili *S*-uple $\left\{ i_{j-1}, i_{j-2}, \cdots, i_{j-S} \right\}$, e tutti i possibili insiemi di superstiti $A^{(j)} = A - \left\{ i_1, i_2, \cdots, i_{j-1} \right\}$. Questo metodo, sebbene esatto, è impraticabile, poiché richiede un tempo molto lungo. Per questa ragione, anziché calcolare

$$\overline{n}_s^{(j)} = E_{A^{(j)}, \left\{ i_{j-1}, i_{j-2}, \cdots, i_{j-S} \right\}} \left\{ \left|\left\{ i_j \in A^{(j)} / \left|i_j - i_{j-1}\right| > S \cap \left|i_j - i_{j-2}\right| > S \cap \cdots \left|i_j - i_{j-S}\right| > S \right\}\right| \right\} \qquad (91),$$

è preferibile valutare la

$$\overline{n}_s^{(j)} \approx E_{\left\{ i_{j-1}, i_{j-2}, \cdots, i_{j-S} \right\}} \left\{ \left|\left\{ i_j \in A / \left|i_j - i_{j-1}\right| > S \cap \left|i_j - i_{j-2}\right| > S \cap \cdots \left|i_j - i_{j-S}\right| > S \right\}\right| \right\} - \left(j - 1 \right) \qquad (92).$$

In alter parole anzichè stimare il valore atteso di $n_s^{(j)}$ rispetto a tutti i possibili insiemi di superstiti $A^{(j)}$ e tutte le possibili *S*-uple, $n_s^{(j)}$ può essere mediate rispetto a tutte le possibili *S*-uple, considerando come insieme dei superstiti sempre l'intero alfabeto A. La riduzione del numero di possibili selezioni è tenuta in conto dal termine $-\left(j - 1 \right)$, (anziché usare $A^{(j)}$). Il calcolo del termine $E_{\left\{ i_{j-1}, i_{j-2}, \cdots, i_{j-S} \right\}} \left\{ n_s^{(j)} \right\}$ può essere facilmente implementato su un PC, mediante un semplice programma C o Matlab. Quando si ha a che fare con interleaver half *S*-random questi termini possono essere espressi in forma chiusa.

Al primo passo di selezione di un interleaver half *S*-random con lunghezza di vincolo *N* si ha $n_{s_{HALF}}^{(1)} = \left|A^{(1)}\right| = N$. Per il *j*-esimo passo, con *j* >1, risulta invece:

$$n_{s_{HALF}}^{(j)} = \left|\left\{ i_j \in A^{(j)} / \left|i_j - i_{j-1}\right| > S \right\}\right|, \quad A^{(j)} = A - \left\{ i_1, i_2, \cdots, i_{j-1} \right\} \qquad (93).$$

Come nel caso dei full *S*-random:

$$\left|\mathcal{E}_{\pi_{HALF}}^{(N,S)}\right| = \prod_{j=1}^{N} \overline{n}_{s_{HALF}}^{(j)} \qquad (94),$$

essendo

$$\overline{n}_{s_{HALF}}^{(j)} \approx E_{i_{j-1}}\left\{\left|\left\{i_j \in A / \left|i_j - i_{j-1}\right| > S\right\}\right|\right\} - \left(j-1\right) \qquad (95).$$

Per questo tipo di interleaver, che rappresenta un caso particolare dei full *S*-random, è possibile esprimere i termini $\left|\left\{i_j \in A / \left|i_j - i_{j-1}\right| > S\right\}\right|$ nella seguente forma chiusa :

$$\left|\left\{i_j \in \mathrm{A} / \left|i_j - i_{j-1}\right| > S\right\}\right| = \begin{cases} N - S - \left(i_{j-1} - 1\right) + 1, & 1 \le i_{j-1} \le S \\ N - 2\left(S-1\right), & S+1 \le i_{j-1} \le N-S \\ i_{j-1} - S + 1, & N-S-1 \le i_{j-1} \le N \end{cases} \qquad (96).$$

Perciò

$$E_{i_{j-1}}\left\{\left|\left\{i_j \in A / \left|i_j - i_{j-1}\right| > S\right\}\right|\right\} = \frac{1}{N}\left\{\sum_{i_{j-1}=1}^{S}\left[N - S - (i_{j-1} - 1) + 1\right] + \right.$$
$$\left. + \sum_{i_{j-1}=S+1}^{N-S}\left(N - 2S + 2\right) + \sum_{i_j - 1 = N-S+1}^{N}\left(i_{j-1} - S + 1\right)\right\} \qquad (97).$$

Sostituendo l'equazione (97) nella (95) si ottiene l'espressione analitica di $\overline{n}_{s_{HALF}}^{(j)}$, il quale può essere sostituito nella (94). La cardinalità delle permutazioni completamente randomiche di lunghezza *N* è pari a $\left|\mathcal{E}_{\pi_{HALF}}^{(N,1)}\right| = \overline{n}_{s_{HALF}}^{(1)} \prod_{j=2}^{N} \overline{n}_{s_{HALF}}^{(j)} = N \prod_{j=2}^{N} \overline{n}_{s_{HALF}}^{(j)} = N \prod_{j=2}^{N}\left(N - j + 1\right) = N!$.

Per verificare la robustezza delle approssimazioni descritte dalle (92) e (95) (rispettivamente per il caso degli interleaver full e half *S*-random) sono stati progettati, con attenzione, quattro interleaver: un full 2-random, un half 2-random, un full 3-random e un half 3-random, tutti caratterizzati da una lunghezza di vincolo $N = 24$. Per valutare, dopo ogni scelta, il numero di

possibili selezioni che possono essere effettuate al passo successivi, è stato sviluppato un semplice programma Matlab. Per esempio il full 2-random usato è:

{2, 20, 16, 4, 21, 17, 10, 5, 23, 11, 19, 7, 13, 3, 8, 14, 18, 9, 22, 1, 15, 12, 24, 6} (98).

Per calcolare $n_s^{(2)}$ bisogna considerare che $A^{(2)} = A - \{2\}$, dove $A = \{1, 2, \cdots, 24\}$. Allo stesso tempo $n_s^{(2)}$ è dato dal numero degli elementi i_2 di $A^{(2)}$ che soddisfano la condizione $|i_2 - 2| > 2$. L'insieme degli elementi di $A^{(2)}$ che soddisfano la condizione è {5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24} e la sua cardinalità è $n_s^{(2)} = 20$. Per quanto riguarda $n_s^{(3)}$ si ha $A^{(3)} = A - \{2, 20\}$. Contestualmente $n_s^{(3)}$ è dato dal numero di tutti gli elementi i_3 di $A^{(3)}$ che soddisfano la condizione $|i_3 - 20| > 2 \cap |i_3 - 2| > 2$. L'insieme degli elementi di $A^{(3)}$ che soddisfano la condizione è {5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 23, 24}, è la sua cardinalità è $n_s^{(3)} = 15$, e così via.

Al fine di confermare la validità delle approssimazioni descritte nella (92) e nella (95), sono stati effettuati alcuni confronti tra i termini $n_s^{(j)}$ trovati nel modo appena descritto (ossia per lo specifico interleaving), e il valore approssimato di $\bar{n}_s^{(j)}$ ottenuto tramite la (92) e la (95).

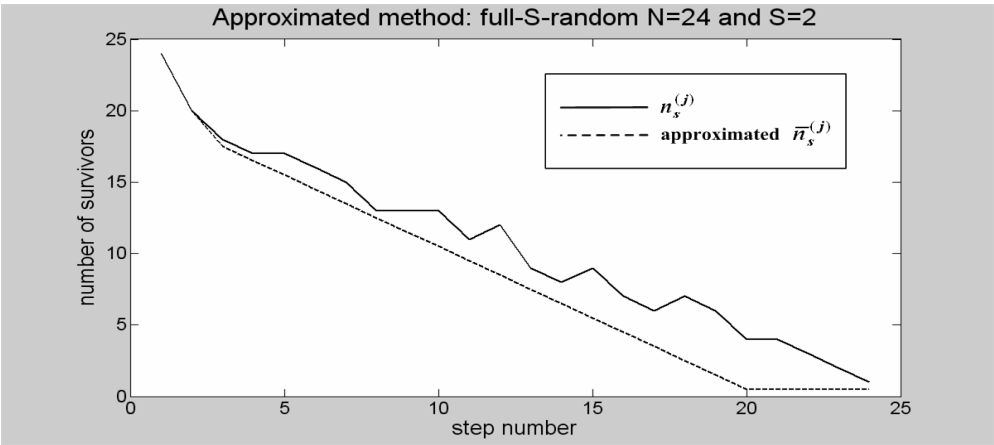

Fig. 34 confronto tra i termini $n_s^{(j)}$ e la versione approssimata $\approx \bar{n}_s^{(j)}$ (caso del full 2-random)

L'interleaver half 2-random adottato è:

{1, 4, 7, 3, 6, 9, 5, 10, 13, 16, 11, 2, 12, 8, 14, 17, 22, 18, 21, 24, 19, 23, 15, 20} (99).

Il full 3-random invece:

{12, 7, 2, 21, 17, 8, 13, 22, 3, 9, 15, 24, 5, 19, 11, 23, 4, 16, 10, 20, 1, 14, 6, 18} (100).

E infine l'half 3-random:

{13, 5, 9, 20, 12, 3, 7, 22, 11, 6, 10, 14, 18, 1, 17, 21, 16, 4, 15, 24, 19, 23, 8, 2} (101).

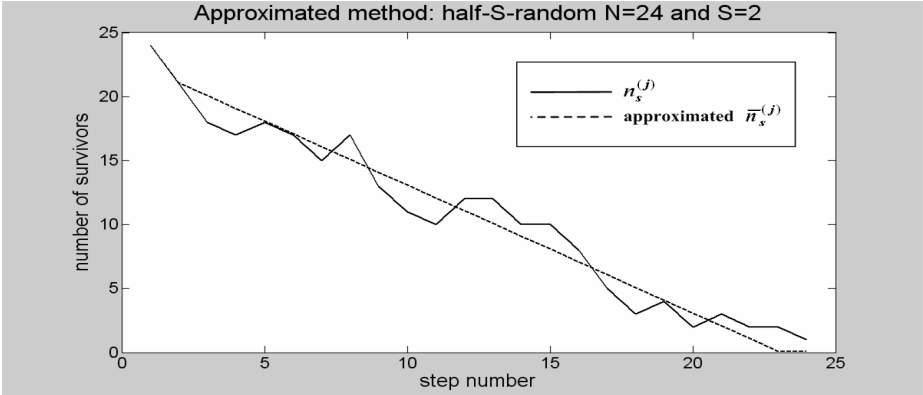

Fig. 35 confronto tra i termini $n_s^{(j)}$ e la versione approssimata $\approx \bar{n}_s^{(j)}$ (caso del half 2-random)

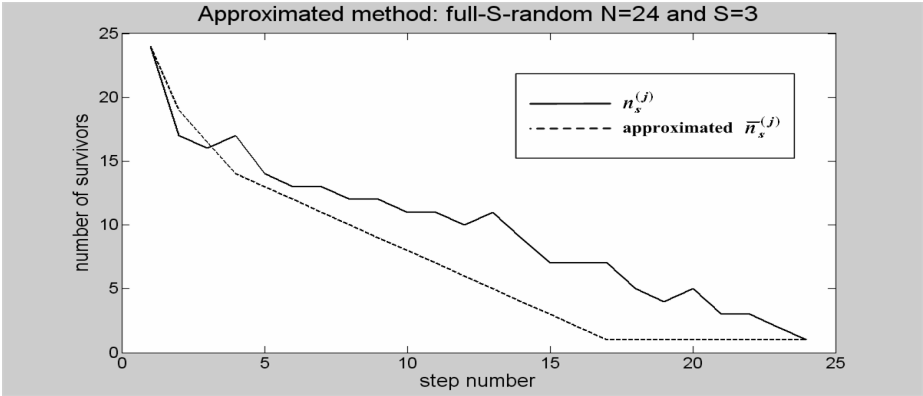

Fig. 36 confronto tra i termini $n_s^{(j)}$ e la versione approssimata $\approx \bar{n}_s^{(j)}$ (caso del full 3-random)

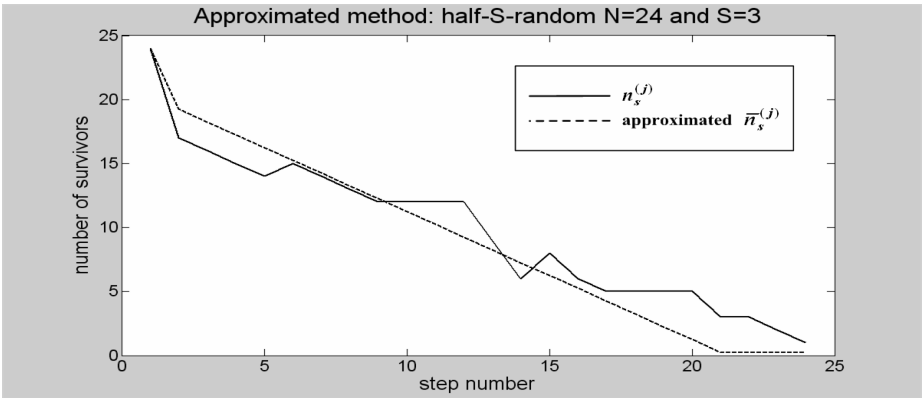

Fig. 37 confronto tra i termini $n_s^{(j)}$ e la versione approssimata $\approx \bar{n}_s^{(j)}$ (caso del half 3-random)

L'approssimazione funziona molto bene con gli half *S*-random, che sono, fra le due sotto-classi in esame, i migliori dal punto di vista della sicurezza dell'autenticazione.

Ad ogni modo, i termini $\left|\mathcal{E}_{\pi}^{(N,S_1)}\right|$ e $\left|\mathcal{E}_{\pi}^{(N,S_2)}\right|$ della (84) possono essere espressi tramite le (90) e (92) nel caso degli interleaving full *S*-random, e per mezzo delle (94), (95) e (96) quando l'interleaver è un half *S*-random.

A questo punto devono essere calcolati i termini $\left|\mathcal{E}_{c}^{(N,v_1)}\right|$ e $\left|\mathcal{E}_{c}^{(N,v_2)}\right|$ della (84).

Il numero di registri a scorrimento relativi al codificatore convoluzionale è $v + 1$. La cardinalità dell'insieme dei registri a scorrimento è invece $J = \sum_{j=0}^{N/\log_2(l_g)-1} (l_g - 1) l_g^j$, ($l_g$ = 8 utilizzando una notazione ottale). Per questo motivo il numero di possibili codificatori RSC è:

$$\left|\mathcal{E}_{c}^{(N,v)}\right| = \left|\mathcal{E}_{c_{RSC}}^{(N,v)}\right| = \binom{J}{v+1} = \binom{\sum_{j=0}^{N/\log_2(l_g)-1} (l_g - 1) l_g^j}{v+1} \qquad (102).$$

Infine, il numero di possibili schemi di punturazione è dato da:

$$\left|\mathcal{E}_{p}^{(N,\rho)}\right| = \binom{N}{N-\rho} \qquad (103).$$

Ad esempio, nella tabella 2 sono riportati i valori di $\log_2\left|\mathcal{E}_{\pi}^{(N,S)}\right|$ rispetto ass *S* per $N = 24$.

	Half-*S*-random	Full-*S*-random
$S = 1$	80	80
$S = 2$	64	55
$S = 3$	55	50

Tab.2 $\log_2\left|\mathcal{E}_{\pi}^{(24,S)}\right|$ vs *S*

Per quanto riguarda l'analisi delle prestazioni dei turbo codici, si consideri che in [3] e [35] sono state trovate delle formule analitiche sia per l'estremo superiore dei coefficienti di errore D_m, che per la distanza minima $d_{\min}$. Purtroppo il calcolo esatto di queste grandezze seguendo tali procedure è piuttosto difficile dal punto di vista computazionale, e la complessità cresce all'aumentare della lunghezza di vincolo N. In [35] è proposto un calcolo approssimato dei D_m estremamente accurato, ma valido solo nell'ipotesi di interleaver uniforme (S = 1). Un metodo per calcolare la $d_{\min}$ dei turbo codici è presentato in [36]. In [37] vengono indagate le prestazioni dei turbo codici usando il metodo della funzione di trasferimento, ma tale procedura non può fare a meno di un'operazione di media rispetto a tutti i possibili interleaving e pertanto non è utile per i nostri scopi. In [38] viene stimata la distanza libera (free-distance) dei turbo codici, ma questo risultato non è utile quando si ha a che fare con comunicazioni a pacchetto come quella presa in considerazione nel presente studio.
È per questo che le prestazioni (in termini di probabilità di errore sul bit) dei turbo codici S-random sono state indagate utilizzando un approccio Monte Carlo.

4.3 Simulazione del sistema

Per verificare l'efficacia dello schema di autenticazione e codifica proposto sia dal punto di vista della certezza dell'autenticazione che da quello dell'integrità dei dati trasmessi, sono state effettuate un gran numero di simulazioni di un sistema di comunicazione OFDM che fa uso dello schema descritto nelle Fig.30 e 31.
Il sistema OFDM testato è caratterizzato da N_c = 1024 sotto-portanti, una lunghezza di vincolo del turbo codice pari a N = 24 bit e una probabilità di falso allarme $P_{fa} = 10^{-3}$.
In Fig.38 è mostrato l'impatto del numero di punti della costellazione L adottata su ogni sotto-banda OFDM sulla probabilità di rifiutare un messaggio autentico a cuas del rumore presente sul canale, noto in letteratura come False Rejection Rate (FRR). Le modulazioni che sono state prese in considerazione nelle simulazioni sono una QPSK (L = 4), una 16-QAM (L = 16) ed una 64-QAM (L = 64). Come era prevedibile, l'FRR aumenta con L.

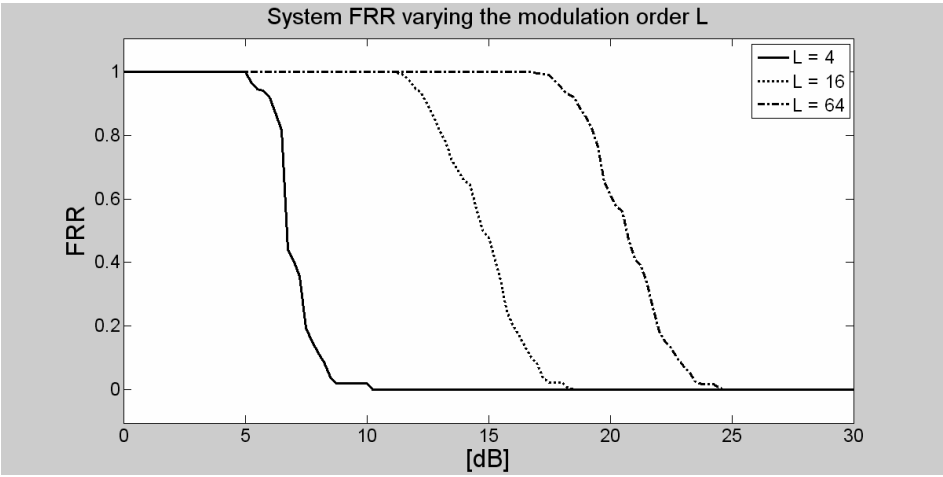

Fig.38 FRR di sistema al variare del formato di modulazione

Come è stato detto precedentemente, ad ogni parola di codice/hash ricevuta, l'utente è riconosciuto solo se l'LLR definito nella (57) è più alto della soglia adattativa λ, che può essere determinata risolvendo l'equazione (60).

In altre parole l'utente è autenticato se:

$$\log \Lambda\left(\hat{\boldsymbol{s}}; \boldsymbol{x}, \boldsymbol{r}_{1,} \boldsymbol{r}_2\right) > \lambda \qquad (104).$$

In Fig.39 viene mostrato un confronto tra il $\log \Lambda\left(\hat{\boldsymbol{s}}; \boldsymbol{x}, \boldsymbol{r}_{1,} \boldsymbol{r}_2\right)$ e λ. Le curve sono state ottenute in questo modo. I primi n_{frame} punti si riferiscono agli LLR calcolati su n_{frame} distinte parole di codice allo stesso SNR = 3dB. I secondi n_{frame} punti sono relativi agli LLR valutati rispetto a n_{frame} differenti parole di codice allo stesso SNR = 3.25dB. I k-esimi n_{frame} punti si riferiscono agli LLR stimati su n_{frame} diverse parole di codice allo stesso SNR = [(k – 1)*0.25 + 3]dB, e così via fino all'SNR = 15dB.

Come si può notare osservando la figura, mentre, all'interno di ogni blocco ad SNR costante (costituito da n_{frame} punti) gli LLR possono differire da un punto all'altro la soglia λ rimane fissata. Ciò è dovuto al fatto che λ, definita nella (60), dipende, per un fissato valore di M (quindi di v e N), soltanto dalle condizioni di canale (dunque da σ_N^2 e quindi dall'SNR) e dalla probabilità di falso allarme P_{fa}. La variazione degli LLR all'interno di un blocco ad SNR costante è dovuta alla randomicità sia dei dati che del rumore additivo

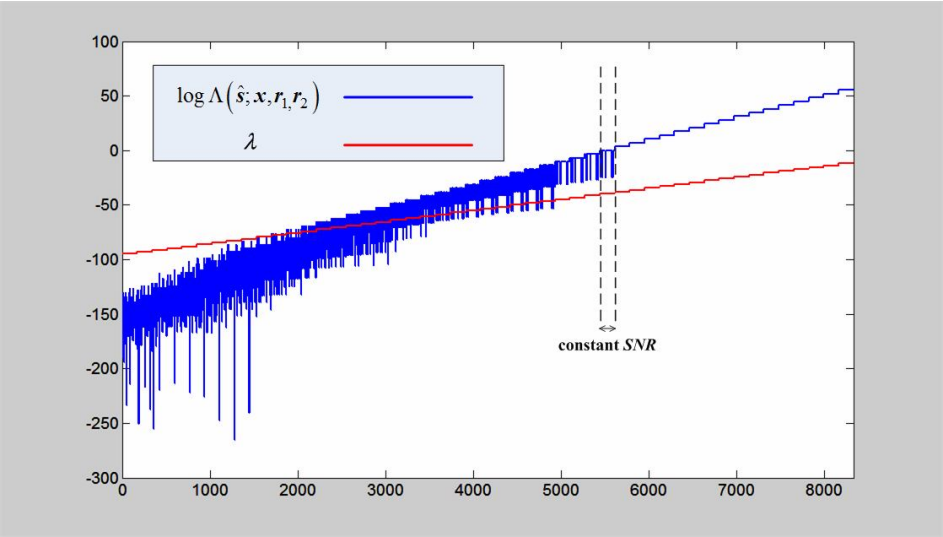

Fig.39 confronto fra $\log\Lambda(\hat{s};x,r_1,r_2)$ e λ

È chiaro che quando l'SNR diventa sufficientemente alto (come nella zona più a destra del grafico di Fig.39), accadde che le differenze $x-\mu(\hat{s}_\Pi)$, $r_1-\mu(\hat{z}_1)$ e $r_2-\mu(\hat{z}_2)$ diventano molto piccole e pertanto risulta difficile apprezzarle sul grafico stesso.

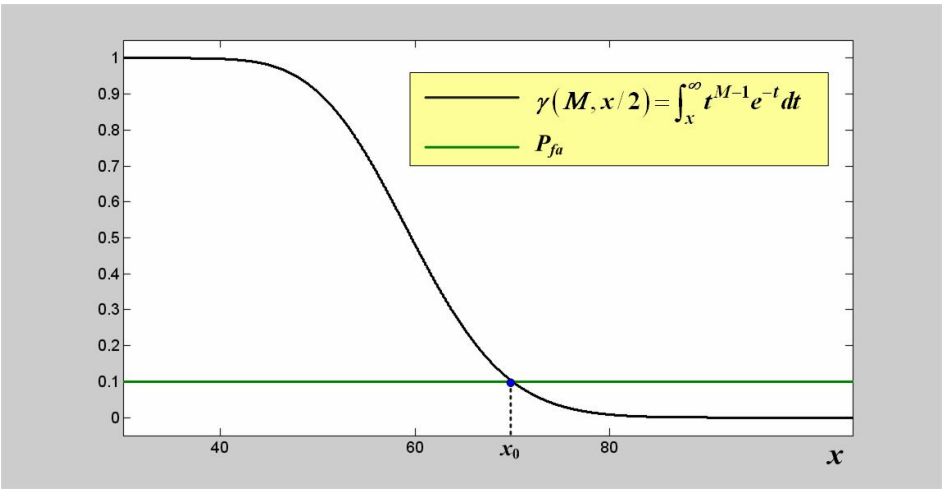

Fig.40 confronto tra $\gamma(M, x/2)$ e P_{fa}.

In Fig.40 è mostrato un confronto fra la funzione gamma incompleta superiore $\gamma(M,x/2)=\int_x^\infty t^{M-1}e^{-t}dt$ e la probabilità di falso allarme P_{fa}.

L'equazione (60) può essere risolta in un modo molto semplice: si trova quell'$x = x_0$ che soddisfi:

$$P_{fa} = \frac{\gamma\left(M, x/2\right)}{(M-1)!} \qquad (105).$$

La soglia λ può essere dunque trovata invertendo l'espressione $x_0 = -2\lambda - 2M\log\left(2\pi\sigma_N^2\right)$, dove $M = vN\ / \log_2(L)$ è il numero di simboli contenuti in una parola di turbo codice, mentre $\sigma_N^2 = 10^{-SNR_{\mathrm{dB}}/10} P_s$ (essendo SNR_{dB} l'SNR in dB e P_s la potenza del segnale utile in scala lineare [W/Ω]). La soluzione x_0 dipende solo da P_{fa} e M, mentre λ varia con M e σ_N^2. In Fig.40 $P_{fa} = 0.1$, $N = 24$, $v = 5$ e $L = 4$ (QPSK) $\Rightarrow M = 60$. In tali condizioni $x_0 = 70.1170$ (si veda Fig. 40).

Alcune delle curve $\lambda\left(\sigma_N^2, M\right) = \frac{x_0\left(M, P_{fa}\right)}{2} - M\log\left(2\pi\sigma_N^2\right)$ parametriche rispetto a M, sono mostrate in Fig.41 per $P_{fa} = 0.1$. In particolare vengono mostrati i tre casi $v = 3$, 5 e 7 che con $N = 24$ e $L = 4$ implicano rispettivamente $M = 36$, 60, 84.

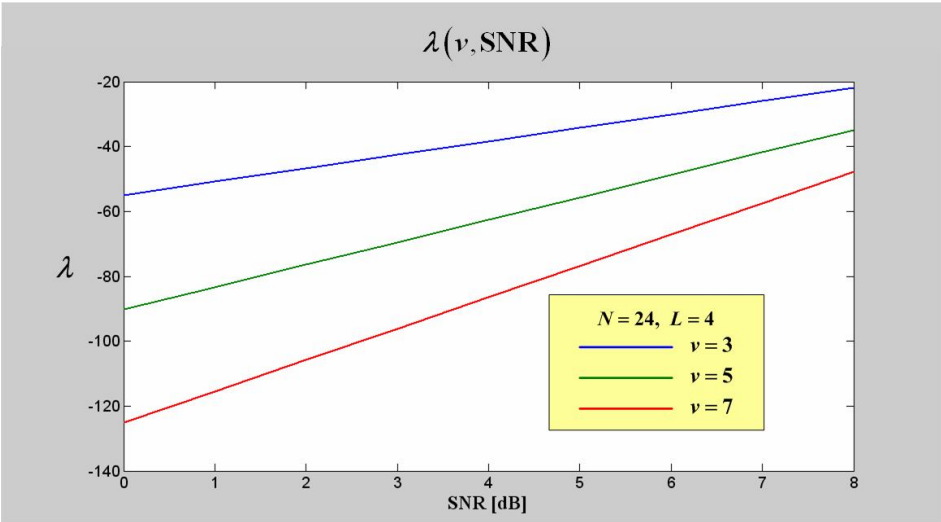

Fig.41 $\lambda(v$, SNR)

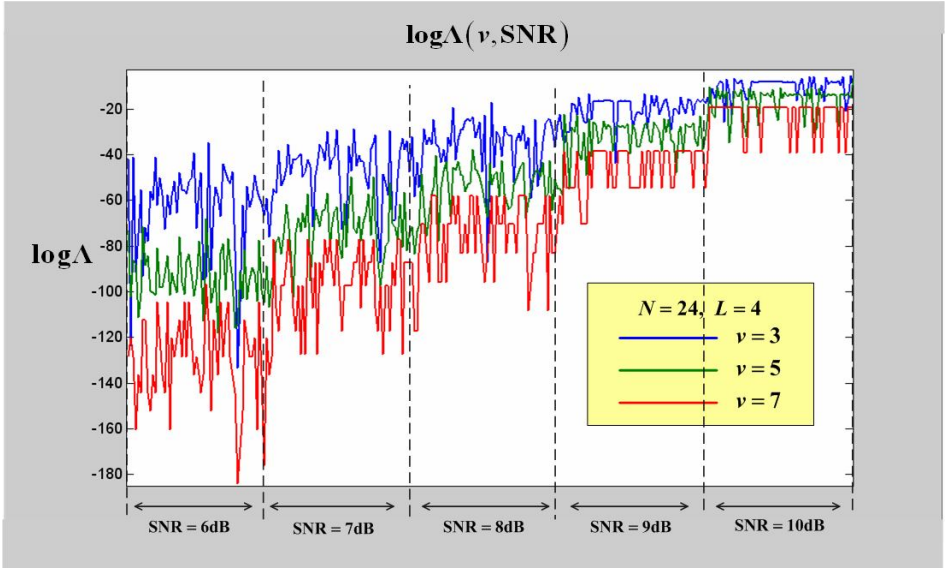

Fig.42 logΛ(*v*, SNR)

Come si può osservare in Fig.42, nonostante la capacità correttiva del turbo codice aumenti con v, gli LLR diminuiscono con v più o meno nello stesso modo in cui λ decresce con v. Perciò non bisogna aspettarsi varaizione dell'FRR rispetto a v. (Le curve degli LLR di Fig.41 sono ottenute nello stesso modo di Fig.39).

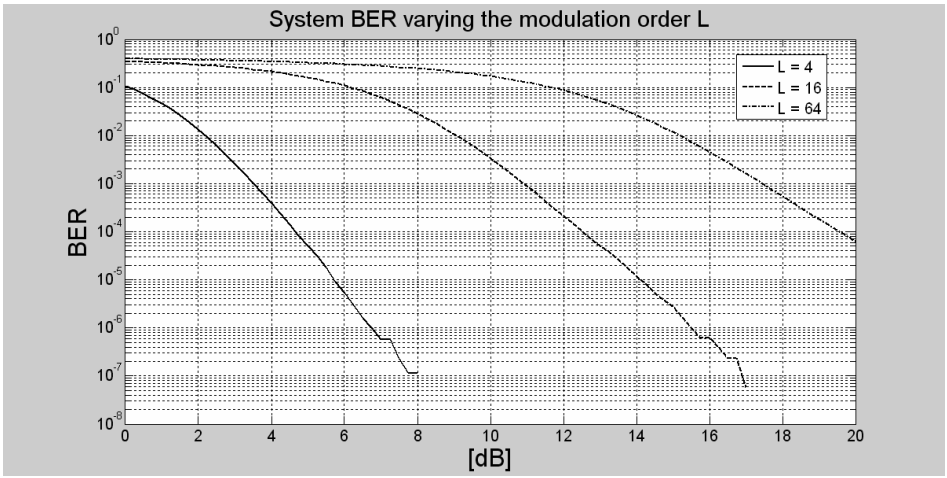

Fig.43 BER di sistema al variare del formato di modulazione

La Fig.44 mostra l'impatto della ridondanza, controllata dal rate del turbo codice $R_{tc} = 1/v$, sull'FRR, per la costellazione QPSK. In queste simulazioni S vale ancora 1.

L'incremento della ridondanza migliora, come è mostrato in Fig.45, le prestazioni dal punto di vista del BER, ma mantiene l'FRR praticamente costante come era previsto in base ai risultati presentati nelle figure 41 e 42.

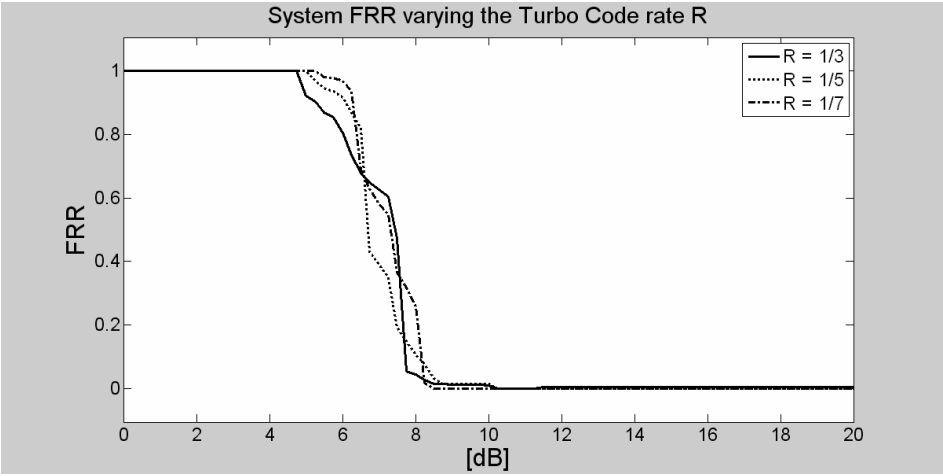

Fig.44 FRR di sistema al variare della ridondanza del turbo codice (caso QPSK)

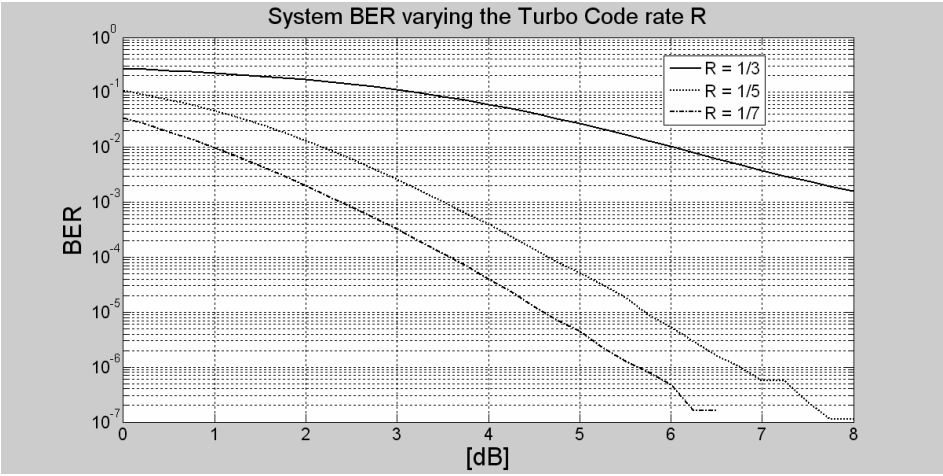

Fig.45 BER di sistema al variare della ridondanza del turbo codice (caso QPSK)

La figura 46 mostra l'impatto della distanza minima *S* dell'interleaver *S*-random sull'FRR. In questo insieme di simulazioni il numero di sotto-portanti OFDM N_c è stato fissato a 1024; è stato impiegato un turbo codice con rate R_{tc} pari a 1/5 ed è stata adottata la QPSK su tutte le sotto-bande.

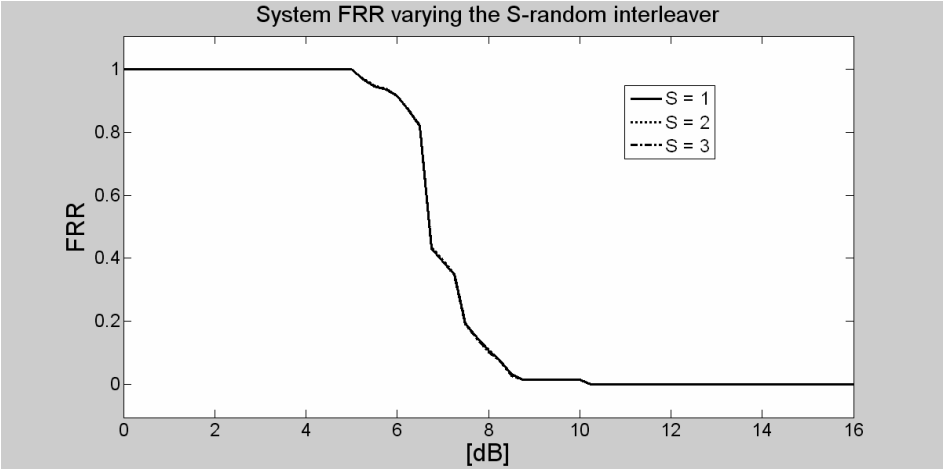

Fig.46 FRR di sistema al variare dell' interleaver *S*-random

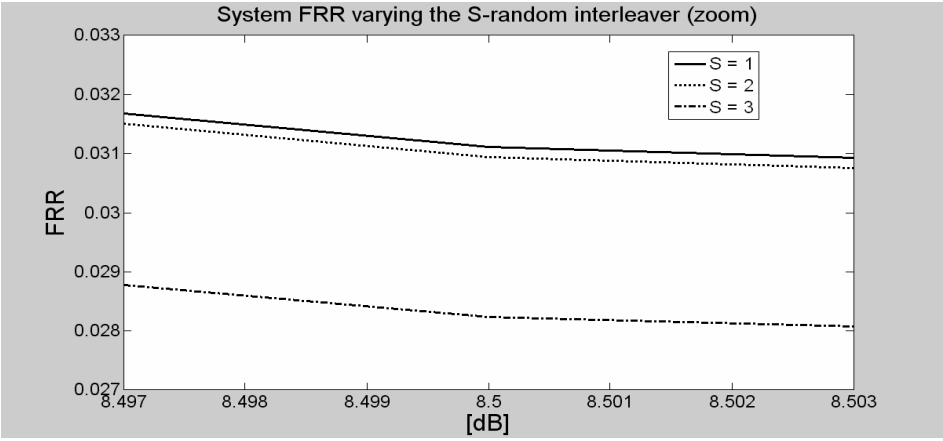

Fig.47 FRR di sistema al variare dell' interleaver *S*-random (zoom)

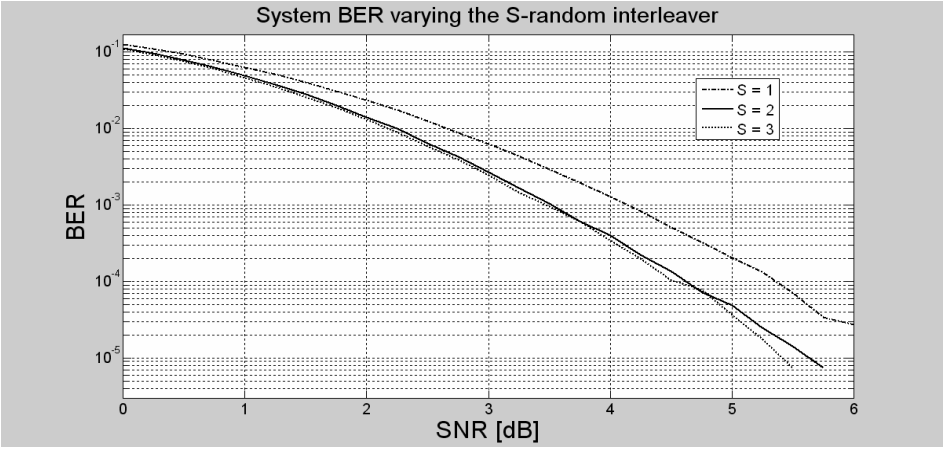

Fig.48 BER di sistema al variare dell' interleaver *S*-random

Il comportamento dell'FRR nei confronti della distanza *S* non riflette quello del BER illustrato invece in Fig.48. Probabilmente il guadagno di codifica raggiunto utilizzando gli interleaver half *S*-random non è alto al punto da far si che un hash parzialmente scorretto possa essere ricondotto ad uno corretto.

5 Sistemi TH CSMA Persistenti

L'algoritmo di sicurezza proposto in questo capitolo fa riferimento ad una tecnica di accesso al canale CSMA (Carrier Sense Multiple Access) in cui il protocollo *p*-persistent è sovrapposto ad una TDMA (Time Division Multiple Access) basata sul TH (Time Hopping).
Lo scenario di riferimento è quello di una rete wireless di computer (o di qualsiasi dispositivo elettronico in rete) equipaggiata con un AP (Access Point). Per rivelare accessi non autorizzati ottenuti grazie alla conoscenza dei requisiti di sicurezza definiti agli strati superiori della pila ISO-OSI (pass-word, serial-number e così via), ad ogni utente autorizzato è assegnata una diversa sequenza di hopping con la quale può accedere al canale. L'AP ha il compito di verificare periodicamente le sequenze di hopping utilizzate dalle comunicazioni sopraggiungenti. Inoltre, l'uso di differenti sequenze di hopping permette diverse trasmissioni simultanee. Se l'AP scopre che la *j*-esima connessione utilizza una sequenza di hopping non corretta, tronca immediatamente il relativo collegamento. Ciò significa che un attacco di questo tipo non può durare più di un periodo di verifica delle sequenze.
La formula per la portata normalizzata (detta anche *throughput*) di un sistema *p*-persistent fornita in [39] è dunque estesa al caso in cui siano usate contemporaneamente N_c sequenze di hopping. Dunque si tiene conto dell'impatto che i periodi di verifica delle sequenze di hopping hanno sul throughput e infine si evidenzia come proprio questi periodi di elaborazione, che comportano una piccola riduzione della portata in condizioni di normalità, impediscano una "caduta libera" del throughput stesso quando uno o più attacchi siano inoltrati alla rete.

5.1 Modello matematico di un TH-CSMA-Slotted-*p*-Persistente

La maggior parte dei lavori disponibili in letteratura riguardanti l'analisi del throughput dei protocolli ad accesso randomico, partono dal presupposto che il numero di utenti che

potenzialmente può accedere al canale sia infinito e pertanto possa essere modellato mediante un processo di Poisson ([40]-[41]). In [42] è fornita un'analisi del throughput per sistemi CSMA persistenti slotted e unslotted con rilevamento della collisione (CD = Collision Detection).

In questo studio si assume che il numero di utenti sia finito. In questo modo è possibile estendere i risultati presenti in [39].

Si assuma che ogni utente trascorra dei periodi, che sono indipendenti e geometricamente distribuiti, durante i quali non ha nemmeno un pacchetto da trasmettere. Sia I la variabile aleatoria che caratterizza la durata di uno di questi periodi. Il teorema di Palm-Khinchine [43] garantisce che tali periodi siano sempre indipendenti ed esponenzialmente o geometricamente distribuiti.

Due to the above assumption, it results that each epoch in the system idle period is a regenerative point, in the sense that the system state after any such epoch is a probabilistic replica of the system state beginning at the previous such epoch. Il sistema alterna il suo stato tra la condizione "idle" periodi la cui durata è caratterizzata dalla v.a. I e lo scenario "busy" di durata caratterizzata da B in cui almeno un utente ha un pacchetto da inviare. Una coppia di B e I consecutivi è chiamato un *ciclo rigenerativo*. Dunque il throughput S è [39]:

$$S = \frac{\overline{U}}{\overline{B} + \overline{I}} \qquad (106),$$

dove U è il tempo speso, durante un ciclo rigenerativo, per trasmettere effettivamente dati, e dove $\overline{\chi}$ rappresenta il valore atteso di χ.

Sia M il numero di utenti che potenzialmente può accedere alla rete, e N il numero di quelli autorizzati. Ad ogni utente autorizzato è assegnata una diversa sequenza di hopping la cui lunghezza è definita dal parametro N_s. Il messaggio consta di N_t slot di durata a (per il significato dei slot e di a si veda [39]). Questo significa che la sequenza di hopping è integralmente ripetuta $\lfloor N_t / N_s \rfloor$ volte durante la trasmissione del messaggio. In Fig.49 sono mostrati N_s blocchi da N_c slot ciascuno.

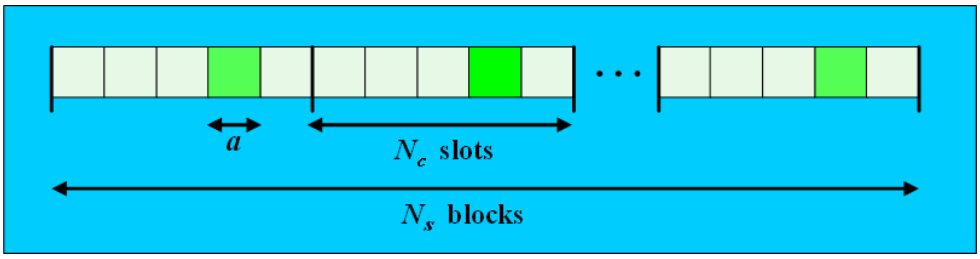

Fig.49 accesso a time-hopping

In [39] il periodo *busy* continua finche c'è almeno un arrivo durante l'ultimo periodo trasmissivo. Nel sistema a time-hopping, un periodo busy continua finchè c'è almeno un arrivo durante gli ultimi N_c periodi trasmissivi, ossia durante gli ultimi $N_c(1+1/a)$ slot.

L'equazione (5) di [39] dunque diventa:

$$\Pr[J=j]=\left[1-(1-g)^{(1+1/a)M}\right]^{j-1}(1-g)^{(1+1/a)M} \tag{107}$$

e pertanto $\bar{J}=(1-g)^{-N_c(1+1/a)M}$.

Di conseguenza le equazioni (6) di [39] diventano:

$$\begin{aligned}\bar{B}&=\sum_{j=1}^{N_c}E\left[B^{(j)}\right]+\left(\bar{J}-N_c\right)E\left[B^{(N_c+1)}\right]\\ \bar{U}&=\sum_{j=1}^{N_c}E\left[U^{(j)}\right]+\left(\bar{J}-N_c\right)E\left[U^{(N_c+1)}\right]\end{aligned} \tag{108}$$

La (7) e la (8) di [39] restano invariate mentre la (9) di [39] ora è:

$$\Pr\left[N_0^{(j)}=n\right]=\begin{cases}\Pi_n(1), & j=1\\ \Pi_n(2), & j=2\\ \vdots & \\ \Pi_n(N_c), & j=N_c\\ \Pi_n\left(N_c(1+1/a)\right), & j=N_c+1,\ N_c+2,\ \cdots\end{cases} \tag{109}$$

La (12) di [39] garantisce che

$$\mathrm{E}[R^{(j)} \mid N_0^{(j)} = n] = a\sum_{k=1}^{\infty}(1-p)^{kn}\left[\frac{p(1-g)^k - g(1-p)^k}{p-g}\right]^{\left\lceil \frac{M}{N_C} \right\rceil - n} \qquad (110).$$

Si noti come M è rimpiazzato da $\lfloor M / N_c \rfloor$, per via dell'N_c-TH. Decondizionando la (110) utilizzando la (109):

$$\mathrm{E}[R^{(j)}] = \begin{cases} \sum_{n=1}^{\lfloor M/N_C \rfloor} \mathrm{E}[R^{(j)} \mid N_0^{(j)} = n]\Pi_n(j), & j = 1, \cdots, N_c \\ \sum_{n=1}^{\lfloor M/N_C \rfloor} \mathrm{E}[R^{(j)} \mid N_0^{(j)} = n]\Pi_n\left(N_c(1+1/a)\right), & j = N_c + 1, \cdots \end{cases} \qquad (111).$$

Sfruttando la (2) di [39], e dopo alcune manipolazioni algebriche:

$$\begin{aligned} \bar{B} = \bar{J}(1+a) + \sum_{j=1}^{N_C}\sum_{n=1}^{\lfloor M/N_C \rfloor} \mathrm{E}[R^{(j)} \mid N_0^{(j)} = n]\Pi_n(j) + \\ + (\bar{J} - N_c)\sum_{n=1}^{\lfloor M/N_C \rfloor} \mathrm{E}[R^{(j)} \mid N_0^{(j)} = n]\Pi_n\left(N_c(1+1/a)\right) \end{aligned} \qquad (112),$$

dove $\bar{J}$, $\Pi_n(X)$ e $E\left[R^{(j)} \mid N_0^{(j)} = n\right]$ sono rispettivamente date dalla (1), (8) di [39] e dalla (110).

Per quanto riguarda $\bar{U}$, la (16) e la (17) di [39], estese al caso N_c-TH diventano:

$$E\left[U^{(j)} \mid R^{(j)} \geq ka, N_k^{(j)} = n+m, N_0^{(j)} = n\right] = \sum_{q=1}^{\min(n+m,N_C)} q\binom{n+m}{q} p^q (1-p)^{n+m-q} \qquad (113),$$

$$\begin{aligned} \mathrm{E}[U^{(j)} \mid N_0^{(j)} = n] = \sum_{k=0}^{\infty}\sum_{m=0}^{\lfloor M/N_C \rfloor - n}\sum_{q=1}^{\min(n+m,N_C)} (1-p)^{kn}(1-g)^{k(M-n)} \cdot \\ \cdot \binom{\lfloor M/N_c \rfloor - n}{m}\left(\frac{g}{p-g}\right)^m \left[1 - \left(\frac{1-p}{1-g}\right)^k\right]^m q\binom{n+m}{q} p^q (1-p)^{n+m-q} \end{aligned} \qquad (114).$$

Decondizionando la (114) usando la (11) di [39], e dopo alcuni passaggi,

$$\bar{U} = \sum_{j=1}^{N_C} \sum_{n=1}^{\lfloor M/N_C \rfloor} \mathrm{E}[U^{(j)} \mid N_0^{(j)} = n] \Pi_n(j) + (\bar{J} - N_c) \cdot \\ \cdot \sum_{n=1}^{\lfloor M/N_C \rfloor} \mathrm{E}[U^{(j)} \mid N_0^{(j)} = n] \Pi_n \left(N_c (1 + 1/a) \right) \quad (115),$$

dove $\bar{J}$, $\Pi_n(X)$ and $E\left[U^{(j)} \mid N_0^{(j)} = n\right]$ sono date dalla (1), la (8) di [39] e dalla (114).

Dunque il throughput del sistema analizzato è:

$$S = \frac{\sum_{r=1}^{N_c} \frac{r}{N_c} \sum_{n=1}^{\lfloor M/N_c \rfloor} \left\{ \mathrm{E}\left[U^{(j)} \mid N_0^{(j)} = n\right] \left[\sum_{j=1}^{N_c} \Pi_n(j) + \left(\bar{J} - N_c\right) \Pi_n\left(N_c(1+1/a)\right) \right] \right\}}{\bar{I} + \bar{J}(1+a) + \sum_{n=1}^{\lfloor M/N_c \rfloor} \left\{ \mathrm{E}\left[R^{(j)} \mid N_0^{(j)} = n\right] \left[\sum_{j=1}^{N_c} \Pi_n(j) + \left(\bar{J} - N_c\right) \Pi_n\left(N_c(1+1/a)\right) \right] \right\}} \quad (116),$$

dove $\bar{J}$, $\bar{I}$, $\Pi_n(X)$, $E\left[R^{(j)} \mid N_0^{(j)} = n\right]$ e $E\left[U^{(j)} \mid N_0^{(j)} = n\right]$ sono rispettivamente forniti dalla (1), la (7) di [39], la (8) di [39], la (110) e la (114).

In Fig.50 S è plottato per differenti valori di N_c.

Nel sistema descritto il periodo di trasmissione è costituito da $X_T = N_c(1+1/a)$ slot. L'introduzione dell'algoritmo di sicurezza di livello MAC implica che ogni tempo pari a R ripetizioni delle sequenze di hopping, ossia ogni RN_sX_T slot X_{elab} slot saranno usati dall'AP per elaborare I segnali ricevuti da tutti gli utenti.

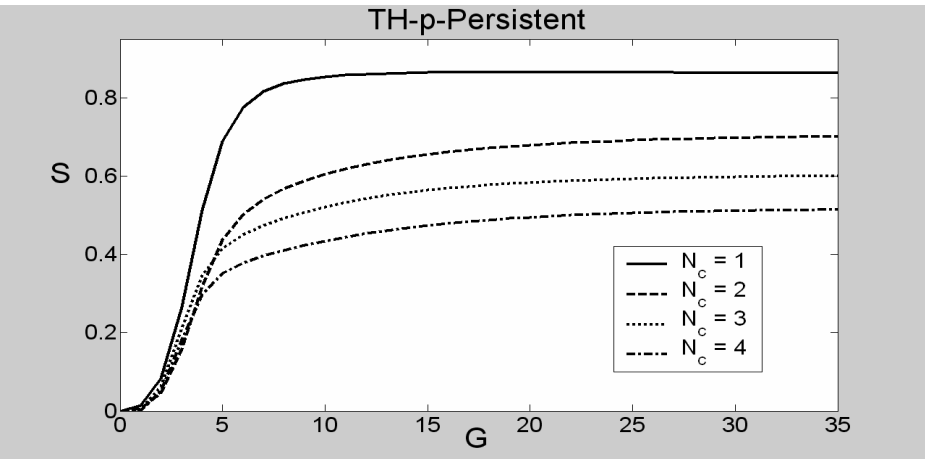

Fig.50 throughput di sistema per differenti valori di N_c (caso senza attacchi)

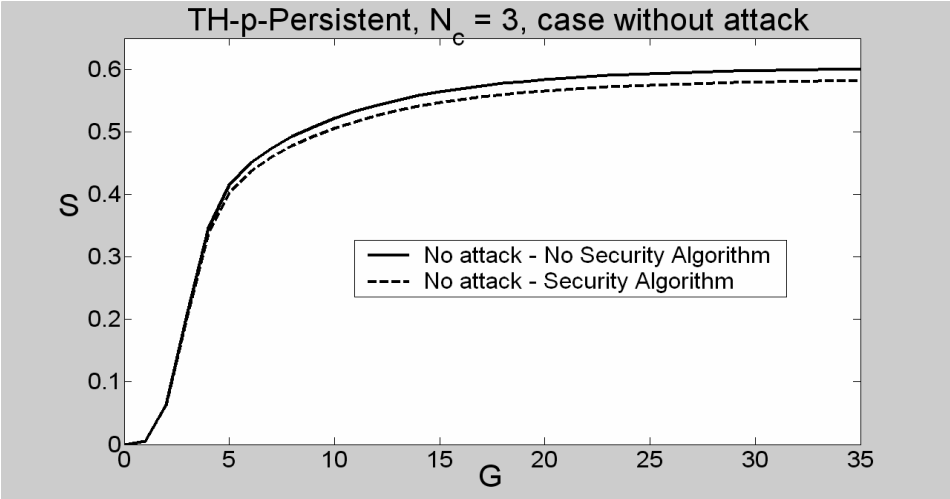

Fig.51 perdita di throughput dovuta alla presenza dell'algoritmo (caso senza attacchi)

Se l'AP scopre che una comunicazione utilizza una sequenza non corretta, la relativa connessione è immediatamente troncata.

In questo modo la definizione generale di throughput, fornita dalla (106), diventa:

$$S = \frac{\overline{U}}{\overline{I} + \overline{B} + \overline{T}_{elab}} \qquad (117),$$

dove $\overline{T}_{elab} = T_{elab} = aX_{elab}$ è il periodo di tempo speso dall'AP per elaborare i segnali ricevuti dai vari utenti. Se $\overline{J}$ è il numero medio di cicli osservati, è evidente come X_{elab} slot di elaborazione seguano RN_s periodi di trasmissione. Tutto va dunque come se $X_{elab}\overline{J} / RN_s$ slot di elaborazione seguano $\overline{J}$ cicli. Perciò, ponendo $X_{elab} / RN_s = q_{elab}$:

$$S = \frac{\overline{U}}{\overline{I} + \overline{B} + q_{elab}\overline{J}} \qquad (118).$$

In Fig.51 i throughput di sistema con e senza lalgoritmo di sicurezza di livello MAC sono messi a confronto. In entrambi i casi $M = 10$, $a = 0.01$, $p = 0.02$, $N_c = 3$ e $q_{elab} = 0.1$.

5.2 Comportamento del sistema in presenza di attacchi

Il numero di time-slot spesi per fermare un attacco dipende fortemente da "quando" l'attacco è inoltrato.

Se l'attacco comincia immediatamente prima di un periodo di elaborazione, sarà fermato in $X_{block} = X_{block}^{(min)} = 0$ slot, mentre se inizia immediatamente dopo un periodo di elaborazione (ossia di verifica da parte dell'AP delle sequenze di hopping), $X_{block} = X_{block}^{(max)} = RN_s X_T$ slot. Se l'istante d'inizio di un attacco può essere modellato mediante una variabile aleatoria a distribuzione uniforme, si può ritenere che il valore atteso di X_{block} sia proprio $\bar{X}_{block} = RN_s X_T / 2$. Ora, si faccia riferimento a una finestra di osservazione temporale data da K periodi di comunicazione, equivalenti a KRN_sX_T slot. È chiaro come, durante tale finestra, saranno usati in media $\bar{X}_{block}$ slot per fermare un attacco, mentre durante i rimanenti $KRN_sX_T - \bar{X}_{block} = RN_sX_T(K-1/2)$ slot, tutti gli N_c slot a TH saranno usati per le comunicazioni autorizzate.

Se N_a è il numero medio di slot a TH usati da utenti non autorizzati, il termine $\bar{U}$ dell'equazione (118) può essere pensato come la somma di due contributi: 1) $\bar{U}_{N_c-N_a}$ (quando solo N_c – N_a degli N_c slot sono disponibili per le trasmissioni autorizzate e quindi contribuiscono al throughput di sistema), globalmente lungo X_{block} slot; 2) $\bar{U}_{N_c}$ (quando c'è la totale disponibilità di slot), globalmente equivalente a $KRN_sX_T - X_{block}$ slot. È chiaro che $\bar{U} = \bar{U}_{N_c-N_a} + \bar{U}_{N_c} \Leftrightarrow KRN_sX_T$ slot.

Perciò, quando $X_{block} = X_{block}^{(max)}$:

$$\bar{U}_{N_C-N_a} \Leftrightarrow \left(\frac{N_c - N_a}{N_c}\right) RN_sX_T = \frac{N_c - N_a}{KN_c}\bar{U} \qquad \bar{U}_{N_C} \Leftrightarrow RN_sX_T(K-1) = \frac{K-1}{K}\bar{U} \quad (119),$$

ciò porta a

$$\bar{U}_{N_C-N_a} + \bar{U}_{N_C} \Leftrightarrow \left(1 - \frac{N_a}{N_c} + K - 1\right)\frac{\bar{U}}{K} = \left(1 - \frac{N_a}{KN_c}\right)\bar{U} \qquad (120).$$

Se $X_{block} = \bar{X}_{block}$:

$$\bar{U}_{N_C-N_a} \Leftrightarrow \left(\frac{N_c-N_a}{N_c}\right)\frac{RN_sX_T}{2} = \frac{N_c-N_a}{2KN_c}\bar{U} \quad \bar{U}_{N_C} \Leftrightarrow RN_sX_T\left(K-\frac{1}{2}\right) = \frac{2K-1}{2K}\bar{U} \text{ (121)},$$

$$\bar{U}_{N_C-N_a} + \bar{U}_{N_C} \Leftrightarrow \left(1-\frac{N_a}{N_c}+2K-1\right)\frac{\bar{U}}{2K} = \left(1-\frac{N_a}{2KN_c}\right)\bar{U} \qquad (122).$$

Infine, quando $X_{block} = X_{block}^{(min)}$:

$$\bar{U}_{N_C-N_a} \Leftrightarrow 0 \quad \bar{U}_{N_C} \Leftrightarrow \bar{U} \qquad (123),$$

$$\bar{U}_{N_C-N_a} + \bar{U}_{N_C} \Leftrightarrow \bar{U} \qquad (124).$$

Se su un'osservazione temporale pari a K periodi di comunicazione sono inoltrati N_a attacchi questi potrebbero essere arrestati in $\bar{X}_{block} = RN_sX_T$ slot (caso peggiore relativo alla portata $S^{(min)}$), $\bar{X}_{block} = RN_sX_T/2$ (caso medio relativo a $S^{(avg)}$) e $\bar{X}_{block} = 0$ (caso migliore relativo a $S^{(max)}$). In altre parole:

$$\begin{aligned} S^{(min)} &= \left(1-\frac{N_a}{KN_c}\right)\frac{\bar{U}}{\bar{I}+\bar{B}+\bar{J}q_{elab}} \\ S^{(avg)} &= \left(1-\frac{N_a}{2KN_c}\right)\frac{\bar{U}}{\bar{I}+\bar{B}+\bar{J}q_{elab}} \\ S^{(max)} &= \frac{\bar{U}}{\bar{I}+\bar{B}+\bar{J}q_{elab}} \end{aligned} \qquad (125).$$

Si dimostra facilmente come in assenza dell'algoritmo di sicurezza proposto N_a attacchi ridurrebbero la portata normalizzata a:

$$S = \left(1-\frac{N_a}{N_c}\right)\frac{\bar{U}}{\bar{I}+\bar{B}} \qquad (126).$$

In Fig.52 sono confrontati i throughput dei due sistemi con e senza l'algoritmo di sicurezza ma entrambi sottoposti ad N_a attacchi su una finestra di osservazione data da $K = 30$ periodi di comunicazione. Gli altri parametri sono invece fissati a $p = 0.02$, $M = 10$, $N_c = 3$. Si fa riferimento a $S^{(avg)}$ per le curve riguardanti il sistema dotato dell'algoritmo di sicurezza.

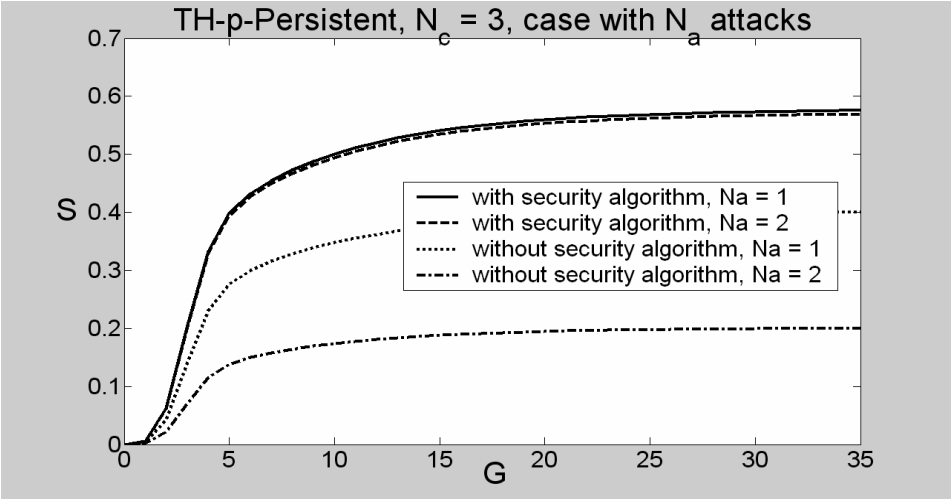

Fig.52 vantaggio derivante dall'uso dell'algoritmo in presenza di uno o più attacchi

5.3 Conclusioni

In [39], lo stato del sistema resta *Busy* se c'è almeno un arrivo durante l'ultimo periodo di trasmissione. In un sistema N_c-TH-Slotted-CSMA-*p*-Persistent tale periodo è N_c volte più lungo, per questo la probabilità che lo stato del sistema ritorni *Idle* è basso. Questo giustifica come mai, perfino in assenza di attacchi, la portata normalizzata decresce linearmente con N_c. In compenso però il sistema è molto più resistente agli attacchi, ovvero gli attacchi intaccano molto meno la capacità del sistema stesso.

Riferimenti

[1] *Digital Modulation Techniques*, Fuqing Xiong, Artech House, 2000.

[2] *Secure OFDM-UWB Communications based on Phase-Hopping*, SPIE Proceedings Vol. 6579, Mobile Multimedia/Image Processing for Military and Security Applications 2007, Sos S.Agaian; Sabah A. Jassim, Editors, 2May 2007.

[3] *Near Optimum Error Correcting Coding and Decoding: Turbo-Code*s, IEEE Transaction on Communications, vol. 44, no. 10, pp. 1261–71, C. Berrou and A. Glavieux , Oct. 1996.

[4] *Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes*, IEEE Transaction on Information Theory, Vol.42, pp.409-429, Sergio Benedetto, Guido Montorsi, March 1996.

[5] *Design of Parallel Concatenated Convolutional Codes*, IEEE Transaction on Communications, vol. 44, no. 5, pp. 591-600, Sergio Benedetto, Guido Montorsi, May 1996.

[6] *A search for good convolutional codes to be used in the construction of Turbo Codes*, IEEE Transaction on Communications, Vol. 46, pp. 1101-1105, S.Benedetto, R.Garello, G.Montorsi, September 1998.

[7] *On the design of Turbo Codes*, TDA Progress Report 42-123, pp. 99-121, D.Divsalar, F.Pollara, November 1995.

[8] *Comprehensive comparison of Turbo-Code decoders*, 45th IEEE Vehicular Technology Conference, vol. 2, pp. 624-628, Digital Object Identifier 10.1109/VETEC.1995.504943, P. Jung, M.M. Nasshan, 25-28 July 1995

[9] *Optimal Decoding of Linear Codes form Minimizing Symbol Error Rate* IEEE Transaction on Information Theory, pp.284-287, L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv, March 1974.

[10] *A Viterbi Algorithm with Soft Decision Outputs and its Applications*, Global Telecommunications Conference, 1989, and Exhibition. Communications Technology for the 1990s and Beyond. GLOBECOM apos; 89., IEEEVolume , Issue, pp. 1680 – 1686, vol.3. Digital Object Identifier 10.1109/GLOCOM.1989.64230 J.Hagenauer e Peter Hoeher, 27-30 Nov 1989.

[11] *Turbo Equalization*, IEEE Signal Processing Magazines, Vol. 21, no. 1, pp. 67-80, R. Koetter, A.C. Singer and M. Tuchler, January 2004.

[12] *Joint Security and Channel Coding for OFDM communications*,16th European Signal Processing Conference (EUSIPCO) 2008, A. Neri, P. Campisi, D. Blasi, L. Gizzi, Lausanne, Switzerland, 25-29 August 2008.

[13] *Secure Communication over fading channel*, IEEE Transaction on Information Theory, vol. 54, no. 6, pp. 2470–2492, L. Yingbin, H.V. Poor, S. Shamai, June 2008.

[14] *On the cardinality of systematic authentication codes via error-correcting codes*, IEEE Transaction on Information Theory, vol. 42, pp. 566-578, G. A. Kabatianskii, B. Smeets, and T. Johansson, March 1996.

[15] *Error-correcting codes for authentication and subliminal channels*, IEEE Trans. on Information Theory, vol. 37, pp. 13-17, R. S. Safavi-Naini, J. R. Seberry, January 1991.

[16] *Digital signature scheme based on error-correcting codes*, Electronics Letters, vol. 26, pp. 898-899, W. Xinmei, June 1990.

[17] *Cryptanalysis and modification of digital signature scheme based on error-correcting code*, Electronics Letters, vol. 28, pp. 157-159, L. Harn and D.-C. Wang, January 1992.

[18] *A proposal of a cryptography algorithm with techniques of error correction*, Computer Communications, vol. 20, no. 15, pp. 1374-1380, W. Godoy Junior and D. Pereira Junior, 1997.

[19] *Cryptanalysis of the Hwang-Rao secret error-correcting code schemes*, in Information and Communications Security, Third International Conference, ICICS 2001, Xian, China (S. Qing, T. Okamoto, and J. Zhou, eds.), vol. 2229 of Lecture Notes in Computer Science, pp. 419–428, Springer-Verlag, (obtained online at http://crypto.nknu.edu.tw/publications/icics2001.pdf). K. Zeng, C.-H. Yang, and T. R. N. Rao, 2001.

[20] *On the design of error detection and correction cryptography schemes*, in EUROCOMM 2000: Information Systems for Enhanced Public Safety and Security, IEEE, AFCEA, IEEE Communications Society, IEEE, 2000. Munich, Germany, N. V. Patsei and P. P. Urbanovich, 2000.

[21] *Authentication, enhanced security and error correcting codes*, in Advances in Cryptology - Crypto ’98 (H. Krawczyk, ed.), vol. 1462 of Lecture Notes in Computer Science, Springer-Verlag, 1998, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, Y. Aumann and M. O. Rabin, August 1998.

[22] *Joint encryption and Error Correction Schemes*, ACM SIGARCH Computer Architecture News, Vol.12, Issue 3, pp. 240-241, T.R.N. Rao, June 1984.

[23] *Cryptosystems using Algebraic codes*, Intl. Conf. On Computer Systems and Signal Proc. Bangalore, India, T.R.N. Rao, December 1984.

[24] *Private-Key Algebraic-Coded Cryptosystems*, Advances in Cryptology - CRYPTO '86: Proceedings, Volume 263/1987, pp. 35-48, Springer Berlin / Heidelberg, T. R. N. Rao, Kil-Hyun Nam, 1987.

[25] *Secret error-correcting codes (SECC)*, in Advances in Cryptology - CRYPTO ’88 (S. Goldwasser, ed.), vol. 403 of Lecture Notes in Computer Science, pp. 540–563, Springer-Verlag, T. Hwang and T. R. N. Rao, 1988.

[26] *Adaptive secure channel coding based on punctured turbo codes*, IEE Proc.-Commun., Vol. 153, No. 2, pp. 313-316, A. Payandeh, M. Ahmadian and M. Reza Aref, April 2006.

[27] *Joint Source, Channel Coding, and Secrecy*, EURASIP Journal on Information Security, Vol. 2007 (2007), Article ID 79048, 7 pages, doi:10.1155/2007/79048, E. Magli, M. Grangetto, and G. Olmo, 2007.

[28] *A survey of Information Authentication*, Proc. Of the IEEE, Vol. 76. N0.5, pp. 603-620, G. J. Simmons, May 1988.

[29] *Authentication theory and hypothesis testing*, IEEE Transactions on Information Theory, Vol. 46, No. 4, pp.1350 – 1356, , D.O.I. 10.1109/18.850674, U.M. Maurer, July 2000.

[30] *Near Optimum Error Correcting Coding and Decoding: Turbo-codes*, IEEE Transaction on Communications, Vol. 44, No. 10, pp. 1261-1271, C. Berrou and A. Glavieux, October 1996.

[31] *Information-theoretic bounds in authentication theory*, Proceedings of IEEE International Symposium on Information Theory, 1995, Page 12, U. M. Maurer, 17-22 Sept. 1995.

[32] *Authentication over Noisy Channels*, CoRR abs/0802.2701,2008, available at http://arxiv.org/abs/0802.2701, L. Lai, H. El Gamal, H. Vincent Poor.

[33] *The performances of interleavers used in turbo codes*, Int. Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005. Volume 1, Issue 14-15, pp. 363 – 366, M. Kovaci, H.G. Balta, M.M. Nafornita , July 2005.

[34] *Variable size Interleaver Design for parallel Turbo Decoder Architecture*, IEEE Transaction on Communications, Vol. 53, No. 11, pp. 1833-1840, S. Benedetto, L. Dinoi, November 2005.

[35] *Combined Turbo Codes and Interleaver Design*, IEEE Transaction on Communications, Vol. 47, pp. 484-487, J. Yuan, B. Vucetic, W. Feng, April 1999.

[36] *Computing the Minimum Distance of Turbo-Codes Using Iterative Decoding Techniques*, Proceedings of the 22th Biennial Symposium on Communications, Kingston, Ontario, Canada, pp. 306-308, S. Crozier, P. Guinand, A. Hunt, May/June 2004.

[37] *An Algorithm to compute the free-distance of Turbo Codes*, IEEE Proceedings on International Symposium on Information Theory 2000, page 287, R.Garello, P.Pierleoni, S.Benedetto, G.Montorsi, June 2000.

[38] *Transfer Function Bounds on the performance of Turbo Codes*, TDA Progress Report 42-122, JPL, Cal Tech, D.Divsalar, S.Dolinar, F.Pollara, 1995.

[39] *Throughput analysis for Persistent CSMA Systems*, IEEE Transaction on Communications, vol. 33, no. 7, pp. 627-638, H. Takagi, L. Kleinrock, July 1985.

[40] *Packet switching in radio channels: Part I-Carrier sense multiple-access modes and their throughput-delay characteristics*, IEEE Transaction on Communications, vol. 23, pp. 1400-14016, F. A. Tobagi, L. Kleinrock, December 1975.

[41] *Performance analysis of carrier sense multiple access with collision detection*, Computer Networks, vol. 4, pp. 245-259, F. B. Tobagi, V. B. Hunt, Oct-Nov 1980.

[42] *A theoretical performance analysis of polling and carrier sense collision detection communication systems*, Local Computer Networks, E. Arthurs, B. W. Stuck, 1982.

[43] *Stochastic Models in Operations Research*, Stochastic Processes and Operating Characteristics, Vol. I, McGraw-Hill, D. P. Heyman, M. J. Sobel, 1982.

ROMA TRE
UNIVERSITÀ DEGLI STUDI

Dipartimento di Elettronica Applicata
Scuola Dottorale di Ingegneria
Sezione di Ingegneria dell'Elettronica Biomedica, dell'Elettromagnetismo e delle Telecomunicazioni

Security management in last generation radio communications

Ph.D Student: Daniele Blasi

Tutor: Alessandro Neri

Index

1 Introduction ---------- 89

1.1 Purpose of the thesis ---------- 90

1.2 Sets of secret keys ---------- 91

1.2.1 Sets of secret keys ---------- 92

1.3 Structure of the thesis ---------- 93

2 Phase-Hopping Authentication ---------- 94

2.1 Phase-Offset modulation ---------- 94

2.2 Phase-Hopping OFDM systems---------- 95

2.2.1 Private key sets of the algorithm ---------- 99

2.2.2 Impact of phase hopping on system performances---------- 99

2.3 Encrypted Hash PH-OFDM systems ---------- 100

2.3.1 User Authentication based on Hash Algorithms ---------- 102

2.3.2 Encrypted-hash and Forward Error Correction ---------- 103

2.3.3 Robust hash function development ---------- 107

2.4 A PH-OFDM system based on robust hashes ---------- 110

3 Turbo Codes ---------- 118

3.1 Turbo Codes on *L*-ary modulations ---------- 119

3.2 From Turbo Codes to Turbo Equalization ---------- 127

3.3 Equalization and the session keys' approach ---------- 131

4 Security and Channel Coding---------- 135

4.1 Mathematical description ---------- 139

4.1.1 Permuted Turbo Codes ---------- 142

4.2 Performance analysis ---------- 145

4.2.1 Security performances analysis of permuted turbo codes ---------- 149
4.3 System simulation ---------- 160
4.4 Conclusions ---------- 166
5 TH CSMA Persistent Systems ---------- 168
5.1 TH-Slotted-CSMA-*p*-Persistent model ---------- 168
5.2 System behaviour in presence of attacks ---------- 174
5.3 Conclusions ---------- 177
References ---------- 178

1 Introduction

With the spread of personal wireless communications, the demand for secured digital communications, assuring privacy, as well as data integrity and authenticity, is constantly increasing. This means that the transmitted data should not be detectable by unauthorized users, still maintaining a strong robustness against the errors introduced by a noisy communication channel, even in severe conditions like indoor and outdoor scenarios affected by multipath. In the meanwhile the receiver should be capable of authenticating the transmitter to avoid *man in the middle* attacks. Often, to isolate the security mechanisms from the characteristics of the technologies adopted at the physical layer, security services are implemented at higher levels of the ISO OSI stack. The adoption of security mechanisms at network or application level allows using public infrastructures, attaining a required level of security, without the need of taking care of the characteristics of the technologies adopted by each operator.

On the other hand, implementing security at physical and link levels in cognitive radio networks allows deploying all countermeasures that can be devised to face denial of service or hijacking attacks. Moreover, exploitation of degrees of freedom not employed for digital data modulation and/or channel coding can be used to face the overhead produced by encryption, mutual authentication, and data integrity, and, then, increasing the overall spectral efficiency.

In the traditional approach source authenticity and message integrity are usually verified by means of solutions designed for noiseless situations. These techniques are based on the use of a message Authentication Code (AC), often denoted as message hash, whose value depends on two functionally distinct parameters, the message to be authenticated and a secret key, supposed unknown to any adversary. Data authenticity and integrity are verified by controlling the coincidence between the received hash and the hash computed on the received message by the authenticator. In this work, a solution to this problem is thus provided,

developing the idea to authenticate the user in different way depending on communication channel condition.

1.1 Purpose of the thesis

The main target of this thesis is to provide a set of solutions in order to manage important security issues of wireless systems, such as user authentication, communications privacy and, at the same time, data integrity.

Referring to the ISO-OSI stack, most of these algorithms are built on the physical level. The use of these techniques does not impair spectral efficiency. Moreover the intruder must be in possess of an advanced knowledge in signal processing, communication and information theory, (in some cases real issues of crypto-analysis and/or enormous computing resources). To detect and by-pass high-level security mechanisms (for example built on network or application layer) many software are nowadays available on the web, while signal analysis programs, software demodulator, spectrum analyzer or other devices are less commercial, and in many cases more expensive since they make use of high-performance hardware like DSP (Discrete Signal Processor) and FPGA (Field Programmable Gate Array).

Also a MAC (Medium Access Control) layer authentication algorithm has been developed. This algorithm is actually based both on MAC and physical-level. The unavoidable impairment on spectral efficiency has been further investigated, (in terms of theoretical throughput analysis).

All the algorithms that will be proposed can be assembled into a unique, secure, communication system providing multiple steps of MAC/physical layers authentication, privacy guardianship and data integrity guarantee.

A sub-system of this ideal solution has been properly deepened, from both theoretical and simulations point of views. The proposal moves into a cognitive radio conception of wireless communication interfaces, where the critical components of the system take decision depending on the perception of the external world. Moreover the users' authentication is not submitted to hard decision rules. This implies that the system components learn, at every

access attempts, something about the user, for examples understand not only if an user is recognized or not, but can see *how much* is, or does not, authenticated.

1.2 Sets of secret keys

A general wireless communication system is characterized by some physical level parameters. Most of them regard, from the transmitter side, the modulation, the encoding rules and the channel estimation, while, from the receiver point of view, the demodulation, the decoding rules and the equalization. At data-link layer, different MAC protocols can be chosen. Every one of this can be described by a dedicated set of private keys.

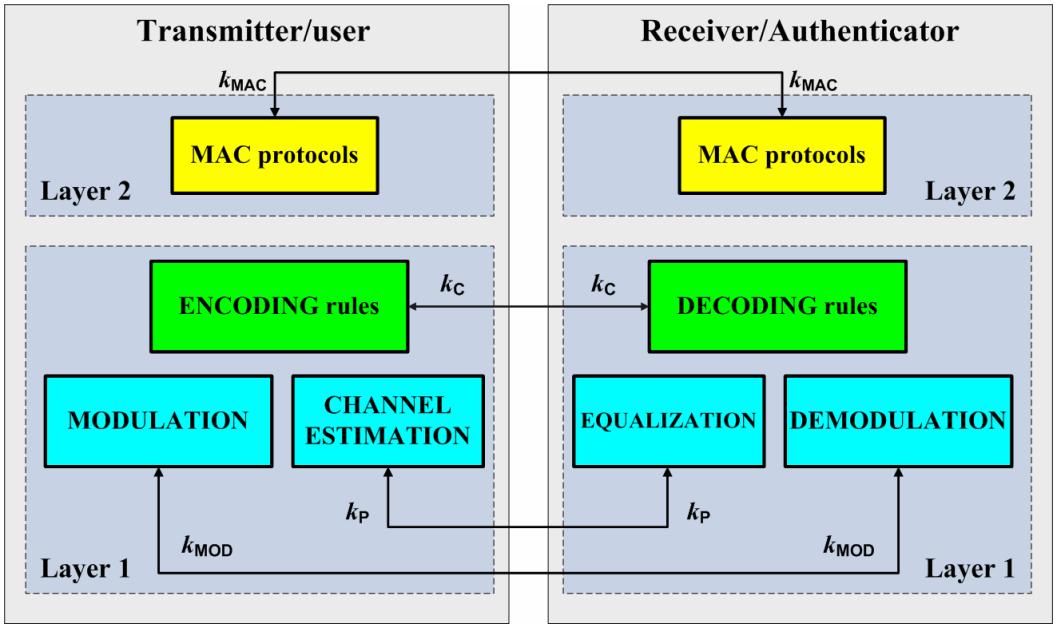

Fig.1 secret keys block-scheme

In Fig.1 the role played from the sets of secret keys, which are supposed to be given to the authorized user on a secure channel. In the picture the following keys' set appear:

5) $\boldsymbol{k}_{\mathbf{MOD}}$, used to lock some freedom degrees of the adopted modulation;
6) $\boldsymbol{k}_{\mathbf{C}}$, that establishes the characteristics of the encoder;
7) $\boldsymbol{k}_{\mathbf{P}}$, which explains the way the channel is estimated;
8) $\boldsymbol{k}_{\mathbf{MAC}}$, that decides the configuration of the link-layer protocol.

It is important to use the terms *set of keys* instead of *keys* since not every issue can be addressed from a single private key.

As an instance, $\boldsymbol{k}_{\mathbf{MOD}}$ could include $k_{\mathrm{MOD}}^{(1)}$, that chooses one of the possible transmission filters $h_{\mathrm{T}}(t)$, and another key, $k_{\mathrm{MOD}}^{(2)}$ that decides the value of a particular freedom degree such as the constellation phase-offset.

Thus, in general, a private session keys' set is defined as:

$$\boldsymbol{k} = \left\{ k^{(1)}, \quad k^{(2)}, \quad \cdots, \quad k^{(n)} \right\} \tag{127}$$

being n the number of element of $\boldsymbol{k}$.

1.2.1 Keys' cardinality

Every key of the set described in (1) actually belongs from a different key's set. In other words $k^{(j)} \in \boldsymbol{K}^{(j)}$, $j = 1, 2, \cdots, n$. In general, denoting by $|\boldsymbol{K}^{(j)}|$ the cardinality if the set $\boldsymbol{K}^{(j)}$, the more $|\boldsymbol{K}^{(j)}|$ is big, the more $k^{(j)}$ is secure, i.e., is harder to be found trough a brute-force attack. When dealing with physical level keys, such as $k_{\mathrm{MOD}}^{(j)} \in \boldsymbol{K}_{\mathbf{MOD}}^{(j)}$, $k_{\mathrm{C}}^{(j)} \in \boldsymbol{K}_{\mathbf{C}}^{(j)}$ and $k_{\mathrm{P}}^{(j)} \in \boldsymbol{K}_{\mathbf{P}}^{(j)}$, increasing the cardinality of the key's set improves system security but not always the performance from the data integrity point of view. Let assume that $k_{\mathrm{MOD}}^{(i)}$ chooses the transmission filter and $k_{\mathrm{MOD}}^{(m)}$ the constellation phase-offset: not every filter leads to good spectral properties, and too many different phase-offsets involve poor performances at low signal to noise ratio levels. At the same way, if $k_{\mathrm{C}}^{(i)}$ decides the puncturing pattern, while $k_{\mathrm{C}}^{(m)}$ the interleaver structure, it is clear that not every choice leads to the lowest BER (Bit Error Rate), although increasing $\left|k_{\mathrm{C}}^{(i)}\right|$ and $\left|k_{\mathrm{C}}^{(m)}\right|$ means decreasing the probability of secret key detection. A further example could be the choice of the pilot carriers in a multi-carrier signal, like an OFDM (Orthogonal Frequency Division Multiplexing). Theoretically, the best channel estimation, could be performed when the pilot carriers are equally-spaced. For this reason, increasing the number of possible value of $k_{\mathrm{P}}^{(j)}$ means increasing the possible frequency

distributions of the pilots. A too far from equally-spaced distribution could lead to a worse evaluation of noise (or multipath channel) spectral features.

So it can be said that the cardinality of a secret-key set must be chosen in order to reach the best compromise between security and performance.

1.3 Structure of the thesis

The work is organized as follows:

In Chapter 2 authentication and encryption systems based on phase-hopping are described. In one of these solutions hash function are also involved being used both for authentication and error correcting purposes. The coding gain is relevant only in limited SNR regions and implies a processing delay.

In Chapter 3 the Turbo Codes are formally described. Moreover the modern joint decoding and equalization iterative technique note as Turbo Equalization is depicted since this can be useful when dealing with physical level security algorithms.

In Chapter 4, an exhaustive mathematical description of *A*-FEC codes is provided. The performances of the authentication codes are thus investigated in terms of impersonation, substitution and deceiving probabilities. The general model is then particularized to the permuted-A-Turbo-Codes. An analytical technique allows the computation of a threshold employed in the code verification test, based on a Neyman-Pearson procedure. Such a threshold is also adaptive, i.e. dependent on channel conditions. The system has been simulated using OFDM signals (with QAM constellations on the sub-carriers).

Finally, the security algorithm proposed in Chapter 5 is based both on MAC and physical layer authentication mechanisms. It refers to a CSMA (Carrier Sense Multiple Access) system where the *p*-persistent protocol is supposed to be overlapped to a particular TDMA (Time Division Multiple Access) technique based on TH (Time Hopping) procedures.

2 Phase-Hopping Authentication

Angular modulation, such as PSK (Phase Shift Keying), QAM (Quadrature Amplitude Modulation) or TCM (Trellis Code Modulation) are often adopted in digital microwave radio relays, or in multimedia short-range communications, either in the singular or in the multi-carrier version (like OFDM).

The demodulation of this kind of signals requires a perfect knowledge of the central frequency f_c, the symbol frequency f_L, and the phase-offset φ_0. In literature are available many algorithms for the estimation and tracking of f_c and f_L. A lot of these solutions exploit the stability of transmitters' oscillators over long time observation periods.

Actually, using DPSK (Differential PSK) does not implies a perfect evaluation of f_c and definitively does not require knowledge of φ_0. In many applications where performances matter, absolute encoded phase modulation are still more used than differential, due to their lower BER [1].

2.1 Phase-offset modulation

A phase-offset modulated base-band signal, at the output of the receiver matched-filter (i.e. at one sample per symbol), can be written, for the k-th symbol-time as:

$$\underline{x}(k) = \sum_k c_k e^{j\varphi_k} = \sum_k c_k e^{j(\theta_k + \phi_0)} \qquad (128),$$

Where ϕ_0 is the constellation phase-offset. If ϕ_0 is unknown, it is impossible to correctly demodulate the signal. Note that a phase-offset PSK signal is characterized from equation (2) with c_k = cost. Also ϕ_0-DMPSK signals deal with a phase parameter ϕ_0, but this is actually a constellation-rotation parameter. There are a few methods to detect the phase-rotation parameter in a ϕ_0-DMPSK modulation, especially when $\phi_0 = \pi/M$. As an instance, referring to

a π/*M*-D*M*PSK modulation, the complex envelope discrete phases are built using the following rule in the time-domain:

$$\begin{aligned} &\varphi_0 = \theta_0 + \frac{\pi}{M} \\ &\varphi_1 = \theta_0 + \theta_1 + \frac{\pi}{M} \\ &\cdots \quad \cdots \quad \cdots \\ &\varphi_k = \theta_{k-1} + \theta_k + \frac{\pi}{M} \\ &\cdots \quad \cdots \quad \cdots \end{aligned} \qquad (129).$$

The terms π/*M* switches alternatively the constellation between two sub-*M*-constellation. Thus, if the estimated central-frequency f_0 is exactly the real carrier f_c, the I-Q signal's diagram will show, on a sufficiently long observation period, 2*M* points (i.e. 4 phases for a π/2-DBPSK, 8 for a π/4-DQPSK and so on). If the received signal is base-band down-converted using an $f_0 = f_c \pm \frac{f_L}{2M}$ (f_c and f_L are respectively the carrier frequency and the symbol rate) a $\Delta\varphi_k = 2\pi \frac{f_0 - f_c}{f_L} = \pm\frac{\pi}{M}$ rotation is forced at every *k*-th symbol time. In these conditions only *M* points will be displayed although the modulation is a π/*M*-D*M*PSK. (i.e. 2 points for a π/2-DBPSK, 4 for a π/4-DQPSK and so on). This techniques and the analysis of spectral properties of interpolated versions of the signal raised to different powers, absolutely permit to recognize a π/*M*-D*M*PSK from a *M*PSK and vice-versa.

For these reasons (and for others before introduced) differential PSK will no longer taken into account, in this work. In other words, phase-rotation in DPSK constellation is not chosen as a freedom degree to be used for security purposes.

2.2 Phase-Hopping OFDM systems

When dealing with OFDM signal, on every sub-carrier can be adopted a different phase-offset constellation whose complex envelope can be expressed by equation (2).

Thus, the complex envelope of a Phase-Hopping OFDM (PH-OFDM) signal transmitted during the symbol interval [0, T) is:

$$\underline{x}(t) = rect_T(t)\sum_{n=0}^{N-1} c_n e^{j(2\pi n\Delta f t+\varphi_n)}\ , \quad c_n = a_n + jb_n \qquad (130),$$

where N is the number of OFDM carrier, $\Delta f = 1/T$, c_n the complex information symbols, and φ_n are arbitrary phase shifts that must be known at the receiver for a proper demodulation. An OFDM signal is a particular case of (4) when $\varphi_n = 0$.

To simplify implementations and reduced mutual interference between systems employing adjacent bands, usually only a subset $\tilde{N}$ of sub-carries is effectively employed.

The OFDM demodulator converts into the digital form the analog in-phase and quadrature components of the received signal, with a sampling frequency $f_s = N/T$. Introducing a reference time-vector $\mathbf{t} = [t_0\ t_1 \cdots\ t_{N-1}]^{\mathbf{T}}$, in order to limit the signal duration like the *rect* window does, the standard OFDM signal can be represented as a $N \times \tilde{N}$ matrix $\mathbf{S}(\mathbf{t})$, being N the number of samples and $\tilde{N}$ the number of carrier.

$$\mathbf{S}(\mathbf{t}) = \left[c_0 \mid c_1 e^{j2\pi\Delta f\,\mathbf{t}} \cdots \mid c_{\tilde{N}-1} e^{j2\pi(\tilde{N}-1)\Delta f\,\mathbf{t}} \right] =$$

$$= \begin{bmatrix} c_0 & c_0 & \cdots & c_0 \\ c_1 e^{j2\pi\Delta f\, t_0} & c_1 e^{j2\pi\Delta f\, t_1} & \cdots & c_1 e^{j2\pi\Delta f\, t_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ c_{\tilde{N}-1} e^{j2\pi(\tilde{N}-1)\Delta f\, t_0} & c_{\tilde{N}-1} e^{j2\pi(\tilde{N}-1)\Delta f\, t_1} & \cdots & c_{\tilde{N}-1} e^{j2\pi(\tilde{N}-1)\Delta f\, t_{N-1}} \end{bmatrix} \qquad (131).$$

The last correspond, summing the element in each column, to the digital version of the waveform presented in (4), when $\varphi_n = 0$, for every n. In this way the discrete-time waveform is fully depicted in both the time and frequency domain. Let

$$\mathbf{M}(\varphi) = diag\left(e^{j\varphi_0}\ e^{j\varphi_1} \cdots e^{j\varphi_{\tilde{N}-1}}\right) = \begin{bmatrix} e^{j\varphi_0} & 0 & \cdots & 0 \\ 0 & e^{j\varphi_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & e^{j\varphi_{\tilde{N}-1}} \end{bmatrix} \qquad (132)$$

be the phase correction matrix at the transmitter/modulator side, where $\varphi = [\varphi_0 \ \varphi_1 \cdots \varphi_{\tilde{N}-1}]$ is the PH vector. The digital version of (4), when $\varphi_n \neq 0$ is thus obtained by the sum of every column of $\mathbf{M}(\varphi)\mathbf{S}(\mathbf{t})$, i.e. of

$$\begin{bmatrix} c_0 e^{j\varphi_0} & c_0 e^{j\varphi_0} & \cdots & c_0 e^{j\varphi_0} \\ c_1 e^{j(2\pi\Delta f\, t_0+\varphi_1)} & c_1 e^{j(2\pi\Delta f\, t_1+\varphi_1)} & \cdots & c_1 e^{j(2\pi\Delta f\, t_{N-1}+\varphi_1)} \\ \vdots & \vdots & \ddots & \vdots \\ c_{\tilde{N}-1} e^{j[2\pi(\tilde{N}-1)\Delta f\, t_0+\varphi_{\tilde{N}-1}]} & c_{\tilde{N}-1} e^{j[2\pi(\tilde{N}-1)\Delta f\, t_1+\varphi_{\tilde{N}-1}]} & \cdots & c_{\tilde{N}-1} e^{j[2\pi(\tilde{N}-1)\Delta f\, t_{N-1}+\varphi_{\tilde{N}-1}]} \end{bmatrix} \tag{133}$$

In other words the sum of the element of the *n*-th columns of the last matrix gives the *n*-th sample of (4). Introduction of the arbitrary sub-carrier phase shift at the modulator side, requires that the complex output of the FFT (Fast Fourier Transform) of the received complex samples is multiplied by the phase correction operator $\mathbf{D}(\boldsymbol{\varphi}) = diag(e^{-j\varphi_1} \quad e^{-j\varphi_2} \quad \ldots \quad e^{-j\varphi_N})$ before (QAM or PSK) digital demodulation is applied. Indeed it can be easily shown that $\mathbf{M}(\varphi)\mathbf{S}(\mathbf{t})\mathbf{D}(\varphi) = \mathbf{S}(\mathbf{t})$.

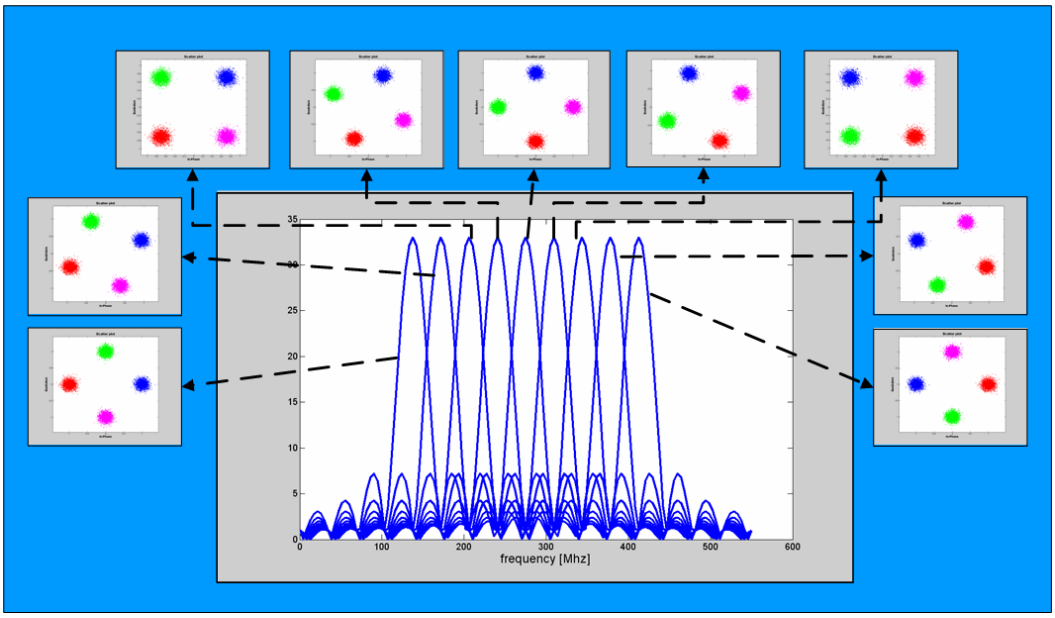

Fig.2 Phase-Hopping QPSK OFDM signals

In Fig.2 a compact scheme depicts how a PH-QPSK-OFDM signal is built. Constellation point are coloured in order to enhance phase-rotations between different subcarriers.

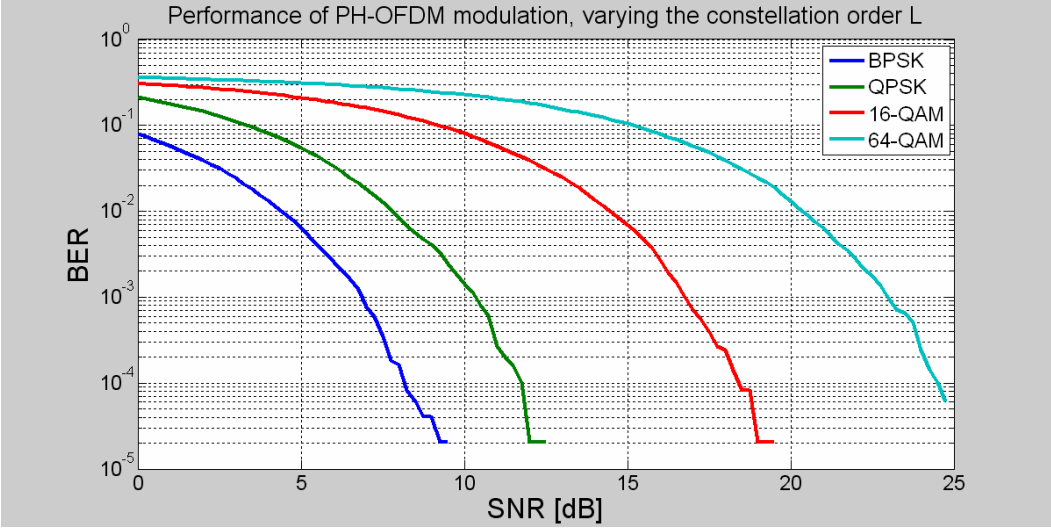

Fig.3 PH-OFDM system performance, varying the number of constellation point *L*

Theoretical results fully match the results of simulations carried out to assess the performance of encryption in the OFDM domain. The BER curves depicted in Fig.3 were obtained simulating a PH-OFDM signal on an AWGN channel, being $N = 1024$ the number of sub-carriers.

Phase-Hopping OFDM signal can be used to implement a stream cipher at symbol level. Still preserving simplicity and ease of implementation of the more diffused binary additive stream ciphers (e.g. Vernam ciphers) that XOR a plaintext binary sequence with a binary key sequence of the same length, the Phase Hopping (PH) cipher, the proposed method is equivalent to a *soft* stream cipher defined on an input alphabet of q-symbols, being q the cardinality of the digital modulation constellation.

The cipher-text alphabet varies with time and, with the exception of a few phase hops, differs from the plaintext alphabet. On the other hand, cipher-text alphabet can not be directly observed for the inherent presence of noise in any communication system.

Higher cardinality of plaintext alphabet, analogue dependence of cipher-text alphabet on phase shift, cipher-text alphabet blurring induced by noise increase the strength of the PH-OFDM soft stream cipher when compared to hard binary additive stream ciphers.

In the PH-OFDM soft stream cipher the phase hops can be generated independently of the plaintext message and of the cipher-text. However, this mode is prone to active attacks against synchronous stream ciphers.

2.2.1 Private key sets of the algorithm

Referring to the notation introduced in chapter 1, a list of sets $\boldsymbol{k}_{\mathbf{MOD}}$ has to be defined. Actually for the proposed technique it does not make sense talking about a list of key sets, since only one freedom-degree of the modulation scheme is involved, i.e. the phase offset. However, it is important to remember that this algorithm is thought to be just a single component of a more complete system whose characteristic parameters could be addressed by other key sets and/or lists of key sets.

The cardinality of $\boldsymbol{k}_{\mathbf{MOD}}$ is given by the number of possible rotation permitted by the adopted sub-carriers' modulations.

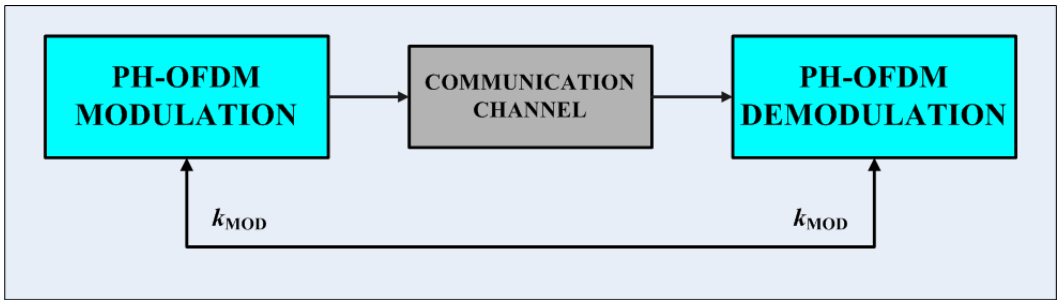

Fig.4 PH-OFDM modulation and demodulation schemes

The block-scheme in Fig.4 describes how this physical-level built-on security algorithm can be introduce in a more complex system since, encoded bits, will be phase-hopping OFDM modulated, using the session key sets (1) $\boldsymbol{k}_{\mathbf{MOD}} = \left\{ k_{\mathbf{MOD}}^{(1)} \quad k_{\mathbf{MOD}}^{(2)} \quad \cdots \quad k_{\mathbf{MOD}}^{(n)} \quad \cdots \quad k_{\mathbf{MOD}}^{(N)} \right\}$, where $k_{\mathbf{MOD}}^{(n)}$ addresses φ_n, i.e. the n-th sub-carrier's phase offset.

2.2.2 Impact of phase hopping on system performances

Concerning communication reliability, we observe that the introduction of an arbitrary phase shift at the transmitter side does not affect performance of the synchronous PF soft stream ciphers in presence of Additive White Gaussian Noise (AWGN). Although it appears to be

rather obvious that $\mathbf{D}(\varphi)$ completely restores the information symbols, the effects on the eventual noise have to be analyzed. Fortunately, linearity of the Discrete Fourier Transform allows us to focus our attention on the noise alone.

Let us denote with $\mathcal{N}_0$ the noise power spectral density and with $\mathbf{n} = \mathbf{n}_I + j\mathbf{n}_Q$ the array of in-phase and quadrature AWGN samples whose covariance matrix is, therefore,

$$\mathbf{R_n} = E\left\{\mathbf{n}\mathbf{n}^\dagger\right\} = 4\mathcal{N}_0 N\Delta f\ \mathbf{I} \tag{134}$$

Since for the FFT operator $\mathbf{W}$ the following property holds

$$\mathbf{W}\mathbf{W}^\dagger = \mathbf{W}^\dagger\mathbf{W} = \frac{1}{N}\mathbf{I} \tag{135}$$

the corresponding FFT output $\tilde{\mathbf{n}} = \tilde{\mathbf{n}}_{\mathrm{Re}} + j\tilde{\mathbf{n}}_{\mathrm{Im}}$ has a jointly Gaussian n-variate distribution with covariance matrix

$$\mathbf{R}_{\tilde{\mathbf{n}}} = E\left\{\tilde{\mathbf{n}}\tilde{\mathbf{n}}^\dagger\right\} = \mathbf{W}\mathbf{R_n}\mathbf{W}^\dagger = 4\mathcal{N}_0\Delta f\ \mathbf{I} \tag{136}$$

Thus the covariance matrix of noise after phase correction is

$$\mathbf{R}_{\tilde{\mathbf{n}}_\varphi} = E\left\{\tilde{\mathbf{n}}_\varphi\tilde{\mathbf{n}}_\varphi^\dagger\right\} = \mathbf{D}(\boldsymbol{\varphi})\mathbf{R}_{\tilde{\mathbf{n}}}\mathbf{D}(\boldsymbol{\varphi})^\dagger = 4\mathcal{N}_0\Delta f\ \mathbf{I} \tag{137}$$

From the comparison of (10) with (11) it follows that noise components with and without arbitrary phase shift have the identical statistics and, therefore, any known phase shift does not alter performance.

2.3 Encrypted Hash PH-OFDM systems

The limits of the system described in paragraph 2.2, could be found essentially in three aspects: 1) a third transmitter antenna could generate a signal that could waste the requested, perfect phase-alignment between the two authorized user; 2) a third user could detect, after a long observation, the right key-stream constellation phase sequences, vanishing the security algorithm; 3) multipath affected channels, typical of indoor scenarios, could make difficult the correction of signal demodulation even if blind channel estimation techniques are adopted,

due to the complexity of a multi-carrier signal equalization and because of the sensibility of similar waveforms to ISI (Inter Symbol Interference).

In order to simplify implementations and reduced mutual interference between systems employing adjacent bands, usually only a subset $\tilde{N}$ of sub-carries is effectively employed. Let assume $\tilde{N} = N/2$. In this case, M out of the $\tilde{N}$ used OFDM frequencies are spent to transmit information symbols, while the remaining $\tilde{N} - M$ sub-bands carry the encrypted-hash version of a part of the same data.

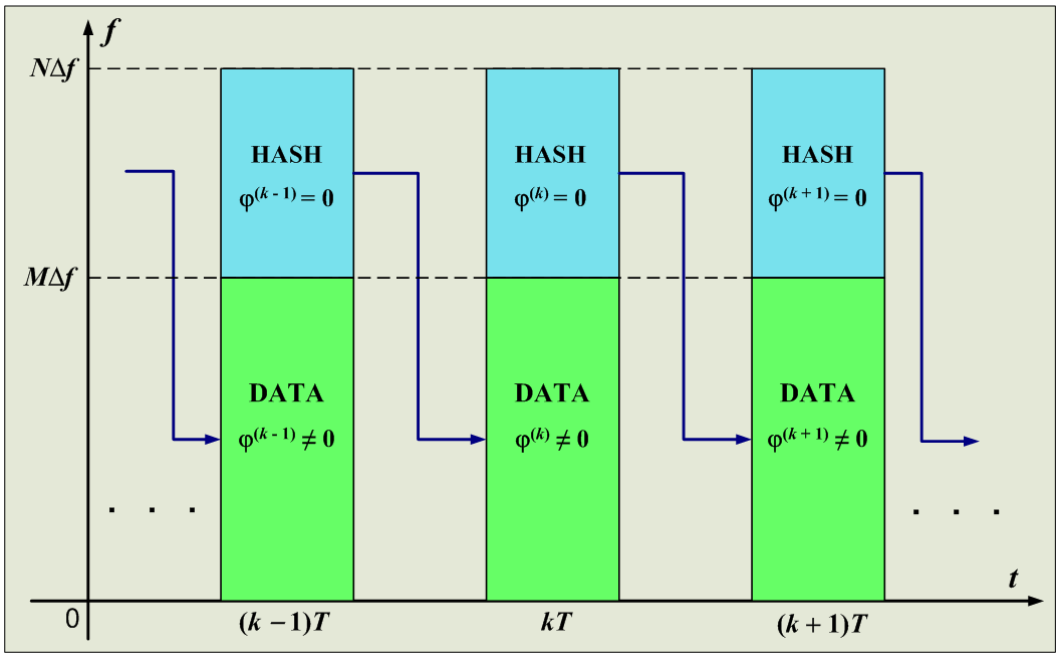

Fig.5 Encrypted Hash PH-OFDM systems

As it is depicted in Fig.5, the first M constellations (related to of the information symbols) are rotated, every symbol time T by the quantities φ_m, $1 \le m \le M$, while for the remaining encrypted-hash constellations it can be set $\varphi_m = 0$, $M+1 \le m \le N$.

To reduce vulnerability to active attacks, the phase-hopping sequence $\varphi^{(k)} = [\varphi_0^{(k)}\ \varphi_1^{(k)} \cdots \varphi_{M-1}^{(k)}]$ for the k-th symbol period ($\varphi_i^{(j)}$ stand for the phase of the i-th carrier of the j-th symbol), is determined by the encrypted-hash of a subset $\boldsymbol{P}$ of the information bits transmitted during the previous packet.

A critical parameter is thus represented by n_φ, i.e. the number of possible value of $\varphi_i^{(j)}$. Referring to the hypothesis in which the values of $\varphi_i^{(j)}$ must be inferior to the minimum phase separation between the different L possible symbol, it is clear that, when L increases, n_φ must necessary decrease, in order to keep an acceptable level of the system performance.

If n_h is the number of bit required for the encrypted hash of the information bit (n_h is independent by the data length where the hash is applied), every symbol period the first $\log_2(L)M = \log_2(L)\tilde{N} - n_h$ bit are used for data encoding while the remaining n_h represent the hash of the previous $\log_2(L)M$ bits. Thus the number of elements of the principal diagonal of $\mathbf{D}(\boldsymbol{\varphi})$, (i.e. of phase corrections) that can be addressed by the n_h available bits is given by $n_P = \lfloor n_h / \log_2(n_\varphi) \rfloor$. In all the simulated cases $n_P < M$, since $M = \tilde{N} - N_H = N/2 - \lceil n_h / \log_2(L) \rceil$. For this reason the M-length PH-sequence was obtained repeating exactly $n_{rep} = \lceil M / n_P \rceil$ time the partial n_P-length sequence on the data-carriers of the subsequent symbol.

L	n_φ	n_P	N_H	M	n_{rep}
4	16	32	64	448	14
8	8	42	43	469	12
16	4	64	32	480	8

Tab.1 System Parameter used during PC simulation

PC simulation were carried out using $n_h = 128$, i.e. adopting, as encrypted-hash algorithm, the 128 bit Message Digest (MD-2), the values 4, 8 and 16 were considered for L. In table 1, referring to the chosen number of OFDM carrier $N = 1024 \Rightarrow \tilde{N} = 512$, are depicted all these parameters.

2.3.1 User Authentication based on Hash Algorithms

From the security point of view, the hash can be used to recognize the user. In other words if the received hash is the hash of the received data (based on the selected algorithm) or equals a hash logged during an enrolment step, the user is authenticated. The system proposed in next

paragraph is based on the fist type, while in paragraph 2.3.3 a variant of this algorithm is explained introducing the enrolment step.

2.3.2 Encrypted-hash and Forward Error Correction

The phase-hopping sequence $\varphi^{(k)} = [\varphi_0^{(k)}\ \varphi_1^{(k)} \cdots\ \varphi_{M-1}^{(k)}]$ is valid for the *k*-th symbol period ($\varphi_i^{(j)}$ stand for *i*-th carrier's phase of the *j*-th symbol), and is determined by the encrypted-hash of a subset ***P*** of the information bits transmitted during the previous packet. The cardinality of ***P*** and the selection it involves can be decided through another private key, given to the user on a secure channel too. Only the information data of the first packet are transmitted without any hopping-sequence.

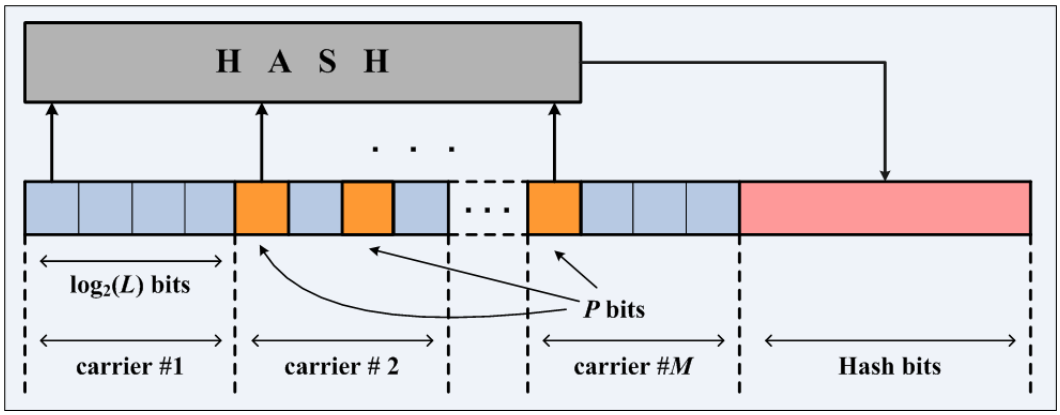

Fig.6 hash error correction

Denoting by $P = |\boldsymbol{P}|$ the cardinality of ***P*** and in the hypothesis that the hash-bits are recovered without any error, performing, on receiver side, 2^P attempts, means correcting exactly *P* errors on information-bits that belong from ***P***, since chosen MD (Message Digest) or SHA (Secure Hash Algorithms) are collision-free. It is clear that increasing *P* means improving the correction property of the employed hash, but involves a higher computational cost. The method is summarized in Fig.6.

We denote with $\mathbf{b}^{(k)}$ the $\log_2(L)\tilde{N}$-length binary vector relative to the *k*-th symbol period. The $\log_2(L)\tilde{N}$-length binary vector $\mathbf{b}^{(0)}$, valid for the first symbol time, is given by the union

of the $[\log_2(L)\tilde{N} - n_h]$-length binary data vector $\mathbf{d}^{(0)}$ with the n_h-length binary vector represented by the encrypted hash of $\mathbf{d}_P^{(0)}$, denoted by $\mathbf{H}\left(\mathbf{d}_P^{(0)}\right)$ ($\mathbf{d}_P^{(0)}$ stands for "a selection of P bits from $\mathbf{d}^{(0)}$"). A Φ-GRAY includes both the mapping operation of the n_h-length binary sequence $\mathbf{H}\left(\mathbf{d}_P^{(0)}\right)$ into the n_P-length phase vector and the repetition of the same (n_{rep} time) to cover the desired M-length PH sequence $\varphi^{(0)}$. The length of this gray code is given by $\log_2\left(n_\varphi\right)$. Although $\varphi^{(0)}$ is evaluated during the transmission of the k-th symbol period with $k = 0$, it will be applied to the M data-carrier relative to the k-th symbol period with $k = 1$. The $\log_2(L)\tilde{N}$-length binary vector $\mathbf{b}^{(0)}$ is then mapped into the correspondent $\tilde{N}$-length OFDM symbol vector $\mathbf{c}^{(0)}$, using a $\log_2(L)$-sized Gray alphabet, (L-GRAY). $\mathbf{c}^{(k)} = \left[c_0^{(k)}\ c_1^{(k)} \cdots c_n^{(k)} \cdots c_{\tilde{N}-1}^{(k)}\right]$ is the symbol vector relative to the k-th symbol period, and that $c_n^{(k)}$ is a more general form of c_n used in equation (4), where only the frequency dependence (i.e. n) is put in evidence, neglecting the time-dependence (represented by k). Finally, using $\mathbf{c}^{(0)}$, a complex envelope of the form (4) , with $\varphi_n = 0$, is then generated and transmitted as $x^{(0)}(t)$.

Like it is shown in Fig.7, the PH-sequence generated during the transmission of the (k – 1)-th packet, i.e. $\varphi^{(k-1)}$, is used to modulate the data OFDM symbol obtained mapping with the L-GRAY code the $[\log_2(L)\tilde{N} - n_h]$-length binary data vector $\mathbf{d}^{(k)}$. The encrypted hash of the last is then mapped, using the Φ-GRAY code, and then repeated n_{rep} time, in order to create the PH sequence that will be applied to the data symbol generated during the transmission of the (k + 1)-th packet. The current k-th packet is thus given by the union of two contributes. The first of this is obtained applying the PH-OFDM modulator for the first M data-carrier frequencies on the mapped version of $\mathbf{d}^{(k)}$, while the second is gained using a standard OFDM modulator on the mapped version of $\mathbf{H}\left(\mathbf{d}_P^{(k)}\right)$, in the subsequent $\tilde{N} - M$ sub-bands. This means that the signal at the k-th symbol-time can be expressed as:

$$x^{(k)}(t) = x_c^{(k)}(t)\cos(2\pi f_p t) - x_s^{(k)}(t)\sin(2\pi f_p t),$$
$$x_c^{(k)}(t) = \mathrm{Re}\left[\underline{x}^{(k)}(t)\right],\ \ x_s^{(k)}(t) = \mathrm{Im}\left[\underline{x}^{(k)}(t)\right] \qquad (138),$$

being f_p the central carrier-frequency, and where $\underline{x}^{(k)}(t)$ is the complex envelope of $x^{(k)}(t)$ given by:

$$\underline{x}^{(k)}(t) = \underline{x}_M^{(k)}(t) + \underline{x}_{\tilde{N}-M}^{(k)}(t),$$
$$\underline{x}_M^{(k)}(t) = rect_T(t-kT)\sum_{n=0}^{M-1} c_n^{(k)} e^{j(2\pi n\Delta f\, t + \varphi_n^{(k)})}, \qquad (139).$$
$$\underline{x}_{\tilde{N}-M}^{(k)}(t) = rect_T(t-kT)\sum_{n=M}^{\tilde{N}-1} c_n^{(k)} e^{j2\pi n\Delta f\, t}$$

In the hypothesis of AWGN channel, the received waveform at the k-th symbol time, will be:

$$r^{(k)}(t) = x^{(k)}(t) + n(t) \qquad (140),$$

where n(t) is a zero-mean white Gaussian noise and where $x^{(k)}(t)$ is described in (12) and (13). Let denote by $\hat{\mathbf{x}}$ the provisional estimation of $\boldsymbol{x}$, while $\tilde{\mathbf{x}}$ represents the definitive estimation of $\boldsymbol{x}$. After $r^{(0)}(t)$, i.e. the signal received during the first symbol period, has been equalized, it can be demodulated obtaining the $\tilde{N}$-length symbol vector $\hat{\mathbf{c}}^{(0)}$. The applications of the inverse L-GRAY code provides the $\log_2(L)\tilde{N}$-length vector $\hat{\mathbf{b}}^{(0)}$, that collects the provisional estimations of the bits that form the first packet. The first $\log_2(L)\tilde{N} - n_h$ bits, forming the estimated data vector $\hat{\mathbf{d}}^{(0)}$, are separated by the last n_h bits, that constitutes the estimated hash vector $\hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right)$. The chosen encrypted hash operator $\mathbf{H}(\cdot)$ is then applied to $\hat{\mathbf{d}}_P^{(0)}$. If the result of this operation does not match with $\hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right)$, all the possible 2^P combination are tried for $\hat{\mathbf{d}}_P^{(0)}$, obtaining every time a different $\tilde{\mathbf{d}}_P^{(0)}$, until $\mathbf{H}\left(\tilde{\mathbf{d}}_P^{(0)}\right) = \hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right)$. This condition will be verified, after a variable number of comparison, only if all the bits of $\hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right)$ are properly received, in other words if $\hat{\mathbf{H}}\left(\mathbf{d}_P^{(0)}\right) = \mathbf{H}\left(\mathbf{d}_P^{(0)}\right)$. These attempts are summarized in the scheme

of Fig.7 by the block 'FEC'. In the same picture the method is extended for the *k*-th period, where the only difference is that the phase-addressing depends on the previous frame.

Using the hash also as a correcting code does not lead to coding gain comparable with the ones obtained by CRC, linear block, convolutional, turbo code, and so on. However the slight performance improvement is obtained without impairing spectral efficiency, since the hash is employed in order to increase system security.

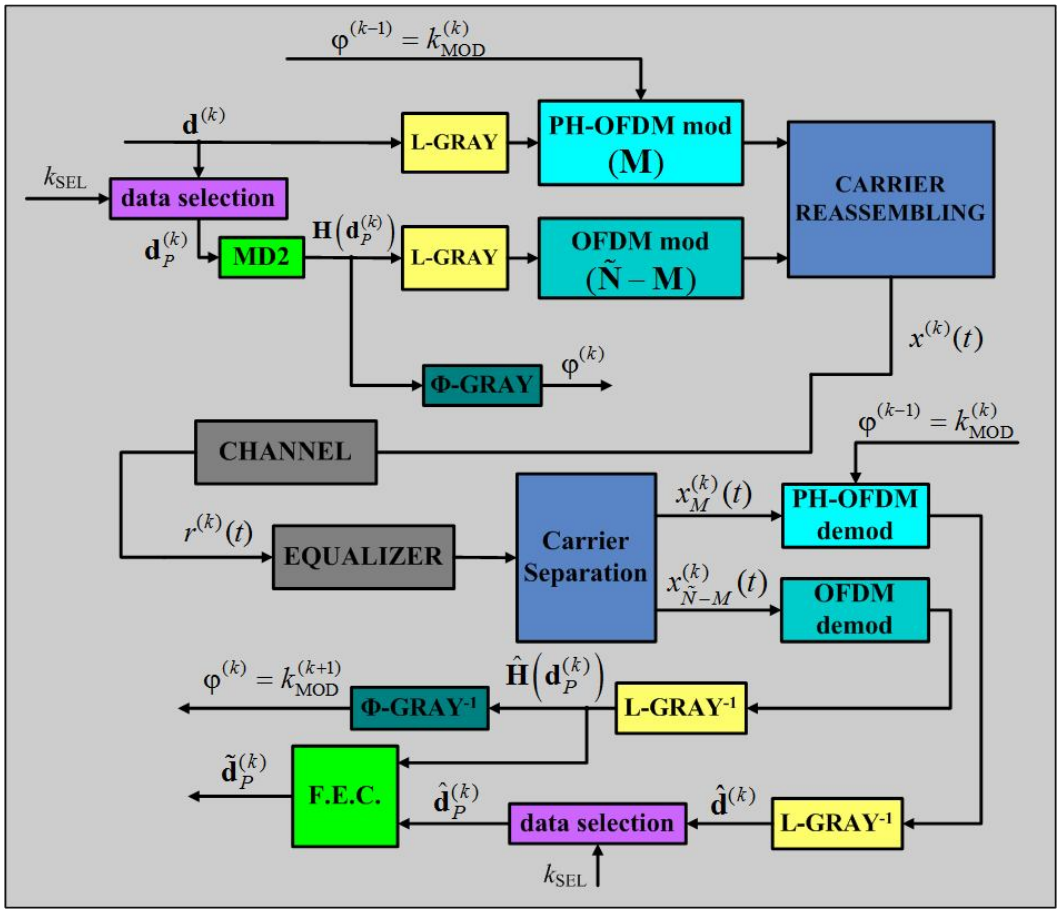

Fig.7 Encrypted-hash PH-OFDM transmitter/receiver block-schemes

The graph in Fig.8 shows the coding gain due to the use of the hash as FEC, evaluated as $(\mathrm{BER}/\mathrm{BER}_{hash})_{dB}$, for $P = 20$, varying the number of OFDM carrier and using the MD-2 hash algorithm (thus $n_h = 128$), when using QPSK constellation on every sub-carrier. The selected P bits, whose hash is transmitted by the last n_h bits of the frame, belong from the

$\tilde{N}\log_2(L)-n_h$ data bits. Simulations show how there is an snr-zone where, the SNR is not too low to make impossible the total absence of error in the hash bits, and is not too high for finding an error among the *P* selected data-bits. Moreover, the coding gain increases when the number of OFDM carriers $\tilde{N}$ decreases. Indeed, the number of data bits for every OFDM frame is actually $\tilde{N}\log_2(L)-n_h$, while the hash bits are n_h, for any value of *L* and $\tilde{N}$. It is clear that a lower number of $\tilde{N}$ leads to an higher probability to find an error into the $\tilde{N}\log_2(L)-n_h$ data bits, and in the hypothesis that the hash is perfectly recovered, the ratio between corrected and total bits in the frame will be superior.

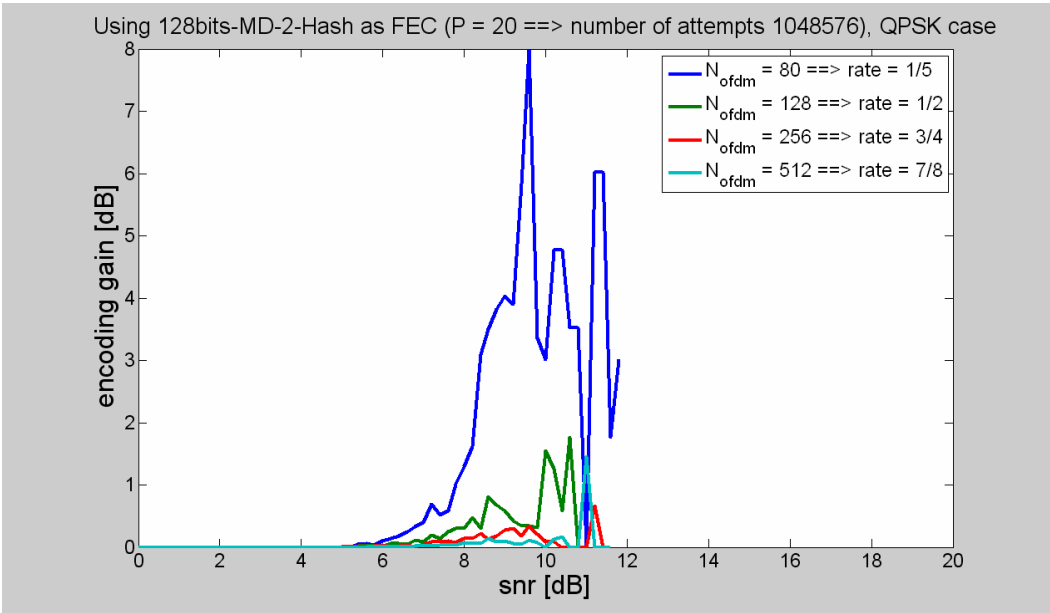

Fig.8 Coding gain using hash function MD-2 as FEC

In the legends of the Fig.8 curves the hash/code rate *R*, given by $1-n_h/(\tilde{N}\log_2 L)$ is displayed too. The depicted algorithm is also explained in [2].

2.3.3 Robust hash function development

An hash function can be said "robust" when a slight variation on the input does not produce an enormous difference in the output. Standard hash does not support this property, since it could represent an help to the attackers' actions.

A way to get a robust hash from a standard one, is the introduction of a FEC algorithm (as an instance a cyclic systematic linear block code) and a hash-domain clustering process.

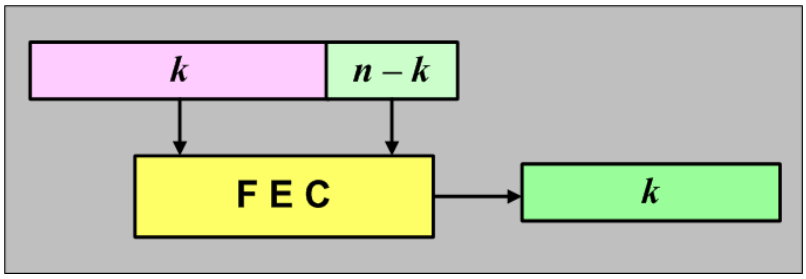

Fig.9 robust hash development

Let denote by n the length of the hash output. The last $n - k$ bits can be thought as a parity component due to a systematic error correction algorithm, as it is depicted in Fig.9. In this way every n bits length hash that is produced can be "corrected", i.e. brought back to one of the available 2^k codewords. This property, that can be meant as a "clustering" of the hash-domain, is summarize in Fig.10.

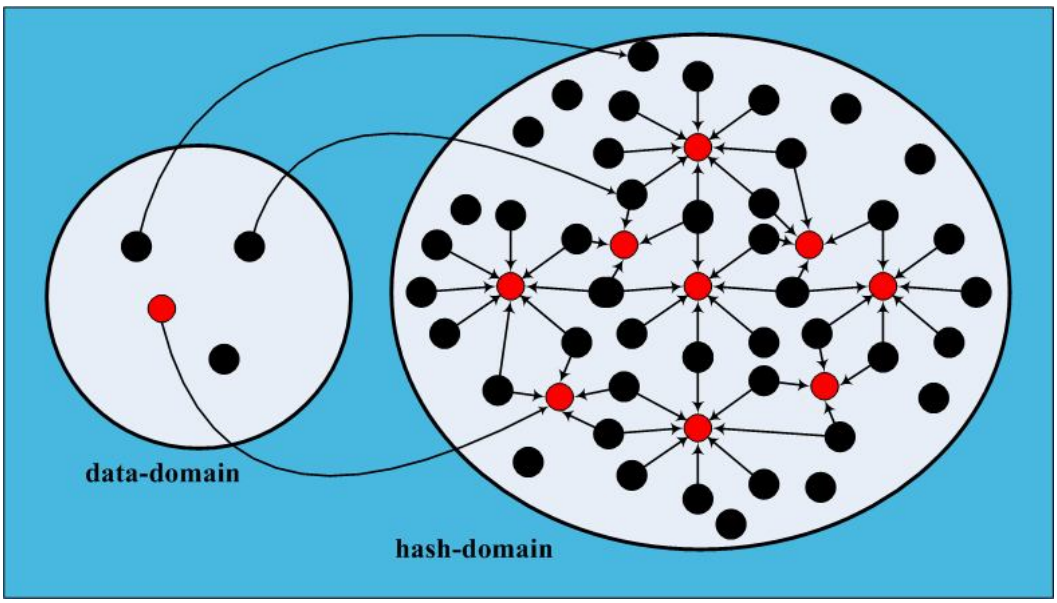

Fig.10 hash-domain clustering

A simple authentication scheme can be now considered, in order to understand how the proposed method can be apply in a real context. (In next paragraph a more complex

technique, representing a variant of the encrypted-hash-PH-OFDM method, analyzed in 2.3.2, will be discussed).

The enrollment step consist of logging, through a secure channel, the hash of a predefined q bit length message word, denoted as $\boldsymbol{x}$. Let n be instead the length of $H(\boldsymbol{x})$, i.e. the hash of $\boldsymbol{x}$. It is clear that $H(\boldsymbol{x})$ can assume whatever value, depending on the hash algorithm. For this reason, even in the recording step, $H(\boldsymbol{x})$ is brought back to one of the 2^k valid codeword using an (n,k) cyclic, systematic, linear block code. Thus, the stored value is not $H(\boldsymbol{x})$, but $H_k(\boldsymbol{x})$, where $H_k(\boldsymbol{x})$ is $H(\boldsymbol{x})$ after the FEC algorithm is applied. In general $H_k(\boldsymbol{x}) \neq H(\boldsymbol{x})$, where $H_k(\boldsymbol{x})$ can be thought as one of the red point of the hash-domain depicted in Fig.10, and $H(\boldsymbol{x})$ as one of the black element of the same set.

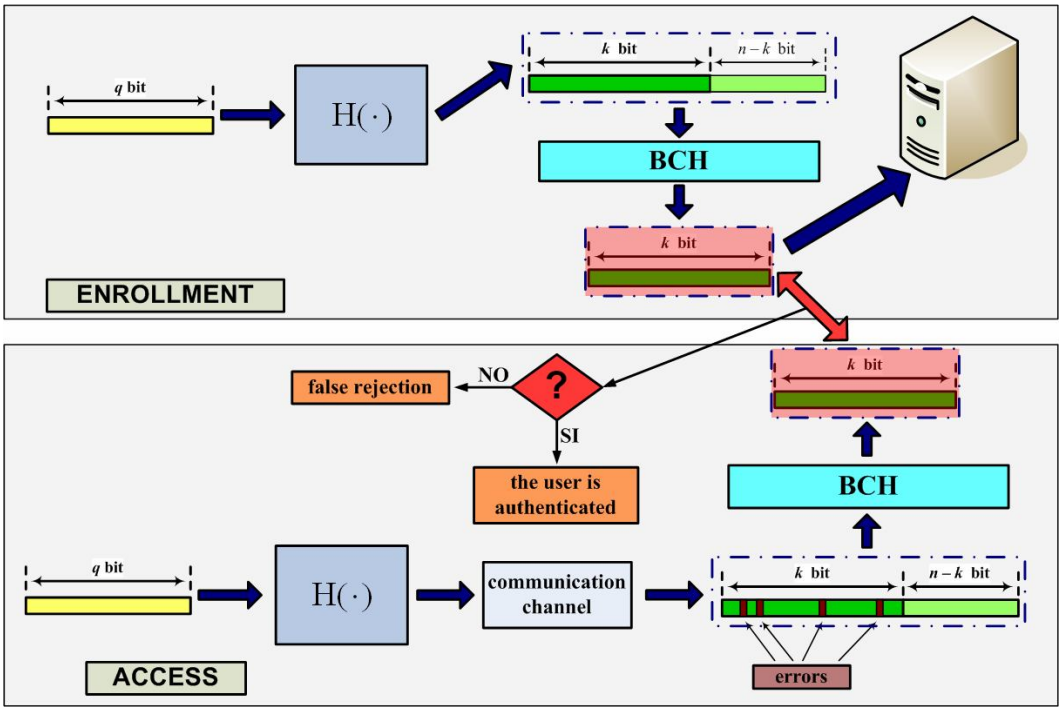

Fig.11 a simple authentication scheme

During the authentication step, if channel conditions are severe, the received hash $\hat{H}(\boldsymbol{x})$ can be different from $H_k(\boldsymbol{x})$ (actually this can happen also if $H(\boldsymbol{x})$ is not a codeword of the chosen FEC algorithm). If the number of errors (i.e. bits of $\hat{H}(\boldsymbol{x})$ not equal to the correspondent bits

of $H_k(\boldsymbol{x})$) does not exceed the correction capability t of the adopted coding scheme, $\hat{H}(\boldsymbol{x})$ can be brought back to $H_k(\boldsymbol{x})$, and the user can be recognized. This scenario is described in Fig.11.

2.4 A PH-OFDM system based on robust hashes

The system is based on three steps: 1) enrollment; 2) authentication; 3) encrypted communication.

During the first stage a serial number s, and a private ciphering key K_c, is given to the user through a secure channel. Moreover the hash of the binary version of s, i.e. b, is stored in the AuC (Authentication Centre) or AP (Access Point).

In the authentication stage, b_{tc} (turbo-encoded version of b) is first transmitted on the N active carriers, using a standard OFDM modulation, i.e. without PH (Phase Hopping). We denote by n the length of $H(b)$, where $H(\cdot)$ is the hash operator. The last $n - k$ bits are thought as parity check of BCH codeword (the choice of the code and of its rate k/n will be discussed later). In this way the last $n - k$ bits are used to bring all the n bits to one of the available codeword. Thus the logical xor of the first k bits with K_c leads to a binary sequence K_a. The 10-base logarithm of the integer version of K_a is then approximated to the nearest integer. This one represents the state S of a random integer generator that produces N numbers, in the range $[0, r-1]$, that determinate the phase-offset of every one of the N OFDM carriers, for the subsequent frame. On every OFDM sub-band a PSK (Phase Shift Keying) modulation is adopted.

The authentication stage still last for $K - 1$ frames, every one characterized by a PH offset established during the previous transmission, but carrying the same binary sequence, i.e. the serial number s. For every received frame, the AuC checks if the hash of the recovered s, i.e. $H(\hat{b})$, is equal to the stored version $H(b)$.

Note that the receiver can every time exploit the turbo-code redundancy to build the best version of $\hat{b}$. The user is recognized if at least for Q of the K received frame results $H(\hat{b}) = H(b)$. It is worth pointing out that Q can be adaptive, i.e. dependent from channel

conditions. Since for low SNR levels the transmitted symbols can be modified by the channel, False Rejection may occur.

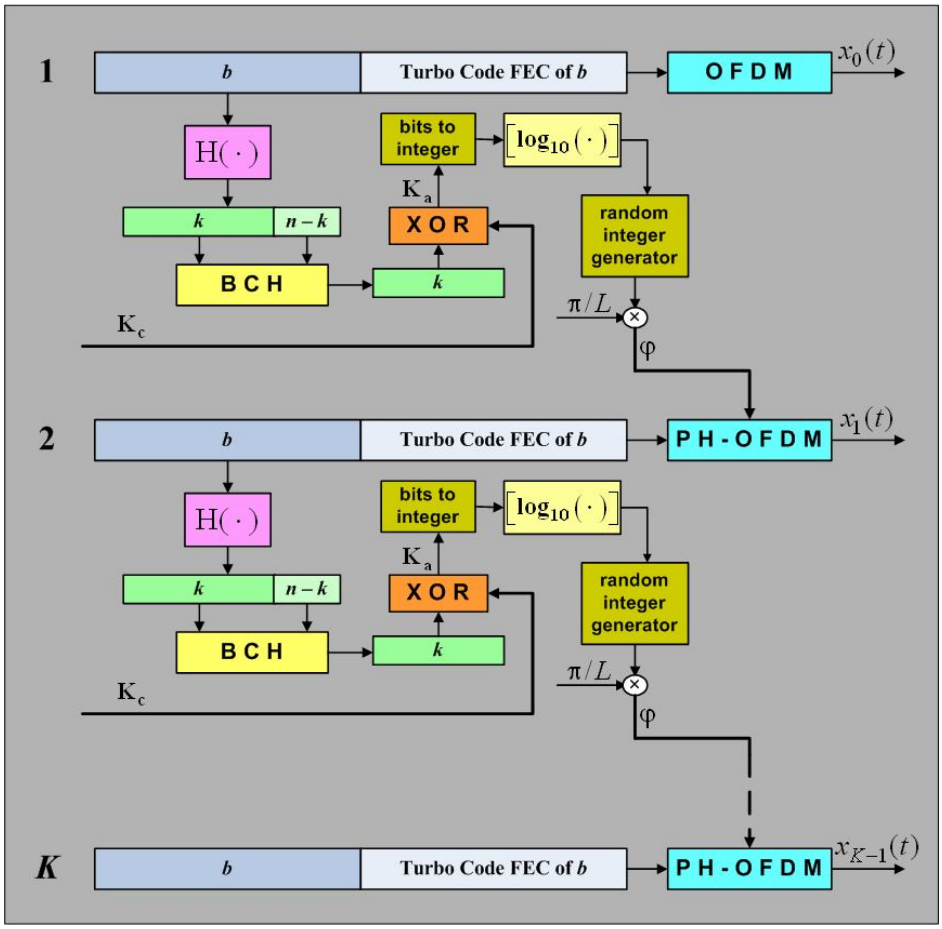

Fig.12 authentication stage (at transmitter side)

In Figs.12 and 13 the authentication step is respectively shown for the transmitter and for the receiver.

The AuC does not receive $x_i(t)$, $i = 0, 1, \ldots, K-1$, but $r_i(t) = x_i(t) + w(t)$, where $w(t)$ is an additive, white, zero-mean, gaussian noise. So, after a standard OFDM demodulation of $r_0(t)$, it can recover $\hat{b}_{tc_0}$. The last $(1-1/R)N\log_2(L)$ bits, (i.e. the FEC bits), being R the rate of the chosen Turbo Code, are used to determinate the best version for $\hat{b}_0$, i.e. $\tilde{b}_0$. Since $\tilde{b}_0$ is obtained, it can be used to get $\tilde{\varphi}_0$.

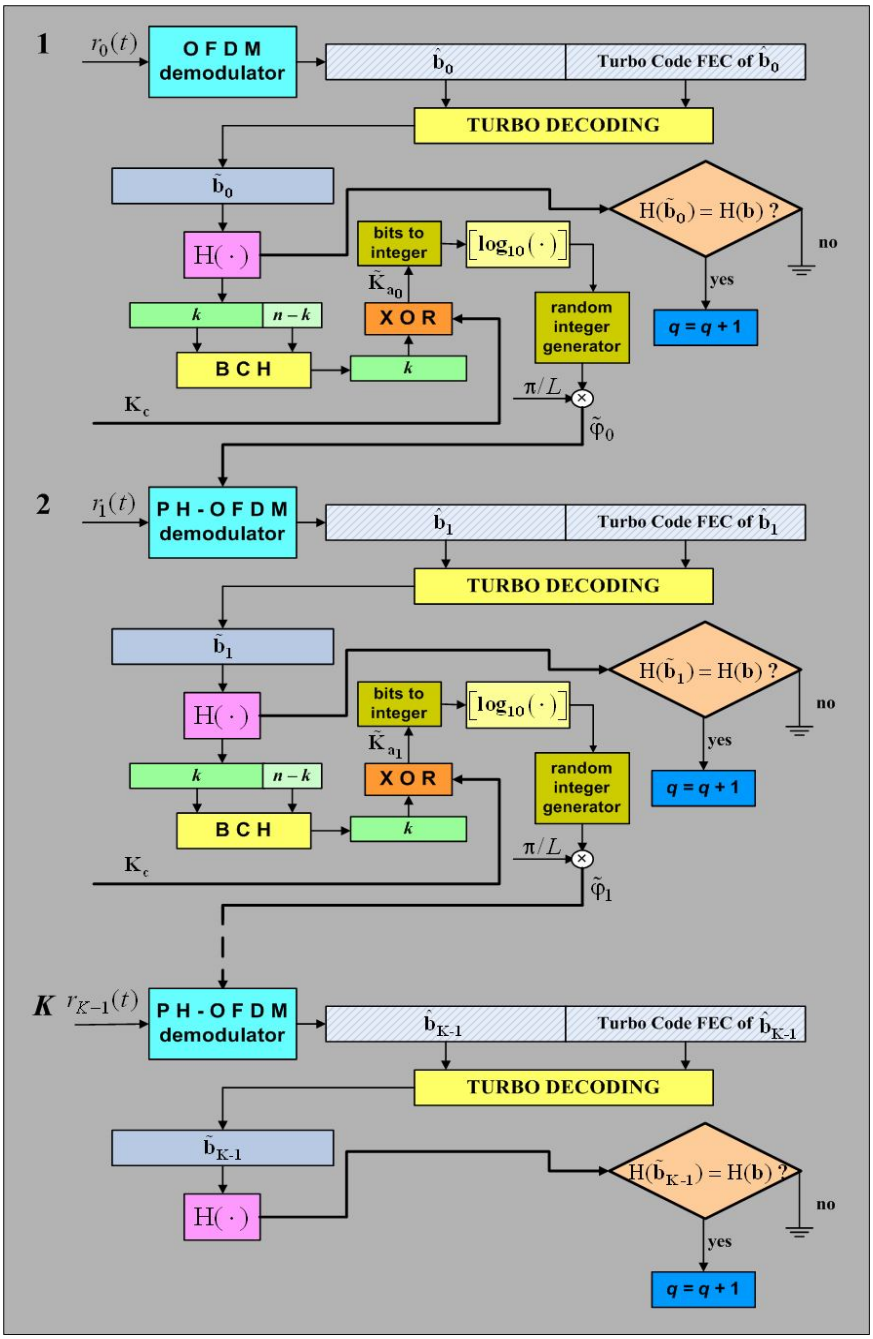

Fig.13 authentication stage (at receiver side)

We denote by q the counter of the positive checks used by the receiver, initialized at $q = 0$. If $H(\tilde{b}_0) = H(b)$ (the last is the stored hash value) $q = q + 1$, otherwise q remains 0. A new

authentication key $\tilde{K}_{a_0}$ will be determined and so a new phase vector $\tilde{\varphi}_0$. If $\tilde{\varphi}_0 = \varphi$ the second received frame, i.e. $r_1(t)$, will be PH-OFDM demodulated, obtaining a new $\tilde{b}_1$. If $\mathrm{H}(\tilde{b}_1) = \mathrm{H}(b)$, $q = q + 1$. A new authentication key $\tilde{K}_{a_1}$ will be determined and so a new phase vector $\tilde{\varphi}_1$. If $\tilde{\varphi}_1 = \varphi$ the second received frame, i.e. $r_2(t)$, will be properly demodulated, and so on, until $r_{K-1}(t)$ is received. For this, last, authentication frame, only $\mathrm{H}(\tilde{b}_{K-1})$ is evaluated, in order to check if it is equal to H(b), while nor $\tilde{K}_{a_{K-1}}$ or $\tilde{\varphi}_{K-1}$ is recovered since the AuC does not wait for more identification packets. After the K-th packet is received, if $q \geq Q$, the user is successfully recognized.

The true encrypted communication starts since the user has been recognized during the last stage. In such an event that user can transmit information bits (instead of the serial number) using the depicted concatenated-hash technique. To avoid loss of synchronization from the receiver, and to enforce security, the serial number can be periodically transmitted. It is clear that in severe channel condition binary errors may occur.

It is very important to observe that either the error correction capacity of the Turbo Code and the robustness of the segmented hash really enforce the proposed authentication and data-transmission scheme.

There are many degrees of freedom in the system, since it is built on different issues, from modulation techniques to FEC algorithms. (See Tab.2).

Hash algorithm	**n**	**k**	**ECC**
MD2	127	22	23
MD5	127	22	23
SHA-256	255	99	23
SHA-512	511	313	23

Tab.2 BCH codes and different involved hash algorithms

The number of non-null carriers N has not impact on the BER (Bit Error Rate) of a standard OFDM modulation. The BER of a PH-OFDM system is instead strongly dependent on the value of N (the more N is high the more is high the probability for the phase-array to be

wrongly recovered). As a consequences, also the FRR (False Rejection Rate) varies with N. The number of non-null carriers taken into account was $N = 512$, 2048 and 8192.

In each OFDM sub-band, as it is known in literature, a different constellation can be adopted, even with different numbers of levels. In PC simulations the same constellation type was assumed (i.e. an L-PSK) in all the carriers. As we expected, L has a remarkable impact on performances, both in terms of FRR and of BER. The values used in computer simulations were $L = 4$, 8 and 16. As for the number of possible phase offsets of the PH-OFDM modulation, that is equal to the range r of the integer number random generator, (introduced in section 2), we found that the system performance are not affected by varying it. For this reason, it was set to $r = 4$, that give a good compromise between BER and FRR. Indeed, choosing a greater value for r means an increment in the BER, but a reduction of the FRR, and vice-versa.

The choice of the Turbo Code is another important element. As it was predictable, incrementing the redundancy of the codes means reducing the BER and so also the FRR. In the following we will denote by $v = 1 / R$ the amount of turbo code redundancy. The values used in computer simulations were $v = 3$, 5 and 7.

As we said before, the BCH codes were used to segment the hash domain. As it is known, the BCH codes have a particular algebraic structure such that if k is the number of bits of the message the length of the codeword is $n = 2^k - 1$. Moreover, the ECC (Error Correction Capacity) of the code, strongly depends on k and n . Since the hash-word produced by the algorithm MD2, MD5, SHA-256 and SHA-512 (MD stand for Message Digest and SHA for Secure Hash Algorithm) were respectively made of 128, 128, 256 and 512 bits, i.e., 2^7, 2^7, 2^8 and 2^9, the last bit of the hash-word should not be considered in the schemes of Figs 12-13. To make a fare comparison among the encrypted hash algorithms, we used BCH codes having all the same ECC, like it is shown in Tab.2. Preliminary simulations showed that the choice of the hash algorithm had no relevant impact on system performances. The SHA-256 algorithm, and so the $(n,k) = (255, 99)$ BCH code were chosen, since they represented the best compromise between computational load and length of the private ciphering key K_c, (remembering that it should to be equal to k).

The parameter Q, or, more precisely the ratio Q / K has to be considered only for the analysis of the FRR. It is clear that increasing this ratio means increasing the FRR, and, at the same time, reducing the FAR (False Acceptance Rate). In all simulations $K = 8$ was chosen.

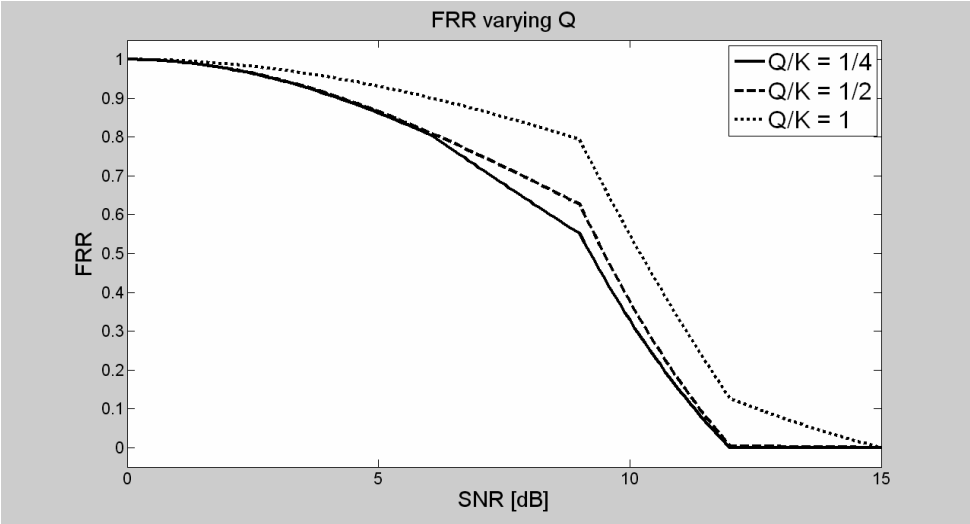

Fig.14 FRR versus SNR varying the ratio *Q* / *K*.

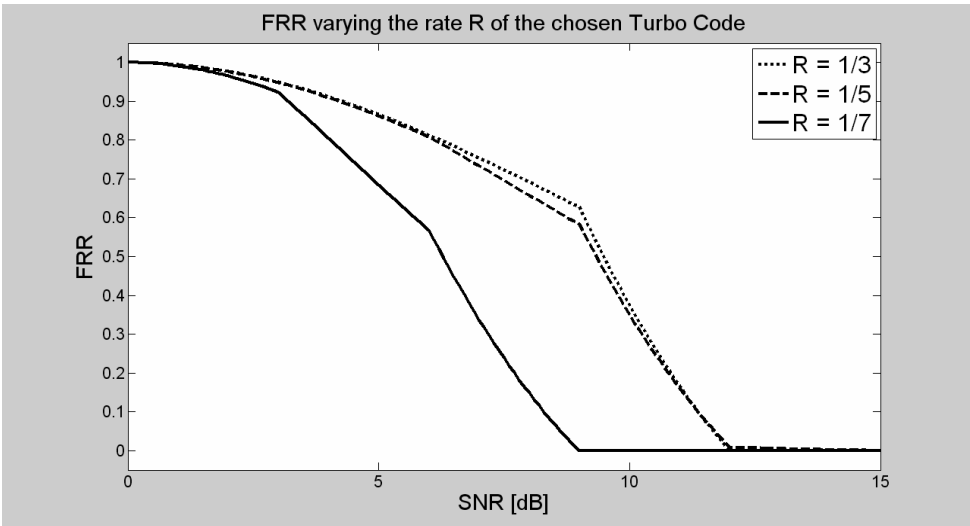

Fig.15 FRR versus SNR, varying the rate *R* of the used Turbo Code.

To verify the possibility of event of False Acceptance it must be considered that three elements must be known to be authenticated: 1) the transmission scheme based on concatenated hash; 2) the serial number; 3) the private encryption key. We tried all the cases

in which the attacker knows two of the three issues. In each attempt, 1000 authentication stage were simulated, the SNR varying between 0 and 15dB, but it was never revealed a false acceptance event. We can thus conclude that, in the SNR range of interest, the FAR is under 10^{-3}. In Figs.14, 15, 16 and 17 curves of FRR (False Rejection Rate) versus SNR (Signal to Noise Ratio) are shown, varying respectively the parameters *Q*, *v*, *L* and *N*.

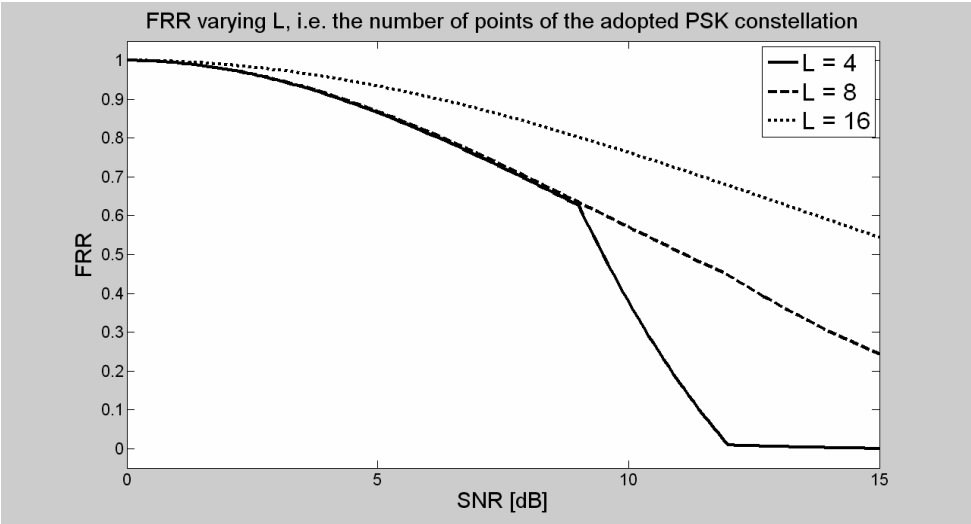

Fig.16 FRR versus SNR, varying the constellation order *L*.

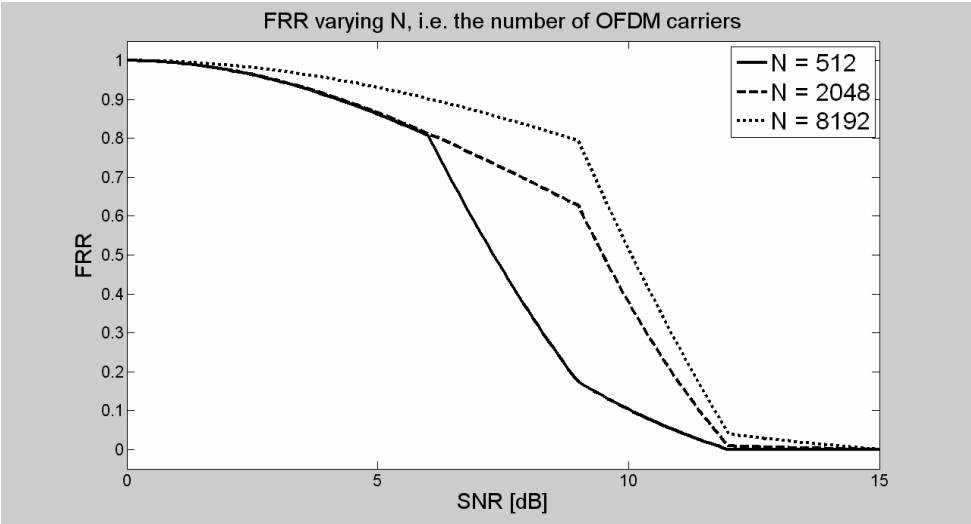

Fig. 17 FRR versus SNR, varying the number of OFDM carriers *N*.

The curves in Fig.17 were obtained assuming $K = 8$, $Q = 4$, $v = 1 / R = 3$ and $L = 4$. The system has been simulated assuming that for each of the K authentication frames the N sub-bands carry only a serial number binary version plus the redundancy due to the Turbo Code. This means that the length of the serial number depends on the number of carriers N and the Turbo Code rate R. More precisely, the more N and R are high, the more the serial number is long. If R increases, the number of bits dedicated to the FEC redundancy decreases. It is clear that a longer serial number increases the probability of false rejection, like it was confirmed by the empirical results.

3 Turbo Codes

Turbo Codes, as introduced in [3], represents a family of convolutional codes, built from a particular concatenation of two recursive systematic codes, linked together by nonuniform interleaving. Decoding calls on iterative processing in which each component decoder takes advantage of the work of the other at the previous step, with the aid of the original concept of "extrinsic information". For sufficiently large interleaving size, the correcting performance of turbo-codes are close to the theoretical limit predicted by Shannon.

Turbo Codes have been also presented by Benedetto and Montorsi as a PCCC (Parallel Concatenated Convolutional Codes) scheme. In particular, in [4] a method to evaluate an upper-bound to the bit error probability of a PCCC (averaged over all interleavers of a given length) is proposed. In [5] they characterize the separate contributions that the interleaver length and the constituent codes give to the overall performance of the PCCC and present some guide-lines for the optimal design of the constituent convolutional codes. More issue on the design of the convolutional turbo code components can be found in [6].

In nowadays applications turbo codes are often employed on satellite links using complex encoding/decoding scheme that involves more block of interleaver/deinterleaver, systematic convolutional encoders/SISO (Soft Input Soft Output) decoders, as it is described in [7]. It is clear that the decoder complexity and the decision latency grow with the number of used branches.

In [8] an accurate mathematical description of the encoding/decoding process of turbo codes as introduced in [3] is shown, using both the MAP (Maximum A Posteriori) symbol estimator algorithm presented in [9] and the APRI-SOVA (A PRIori Soft Output Viterbi Algorithm) depicted in [10].

In next paragraph the mathematical approach introduced in [8] is applied when turbo codes are used on *L*-ary, complex modulations, like QAM.

3.1 Turbo Codes on *L*-ary modulations

As shown in Fig.18, that represents a basic turbo encoding scheme, the information sequence $\boldsymbol{u}$ is multiplexed with the sequences $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ in order to obtain the sequence $\boldsymbol{b}$. $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are respectively the output of the first and the second RSC (Recursive Systematic Convolutional) encoder. As it can be noted $\boldsymbol{c}_2$ is actually the output of the second RSC when applied to $\boldsymbol{u}_1$, i.e. a permutated version of $\boldsymbol{u}$. Let RSC1 and RSC2 denote respectively the first and the second RSC encoder. If 1/2 is the rate of both RSC1 and RSC2, the global turbo code rate will be 1/3

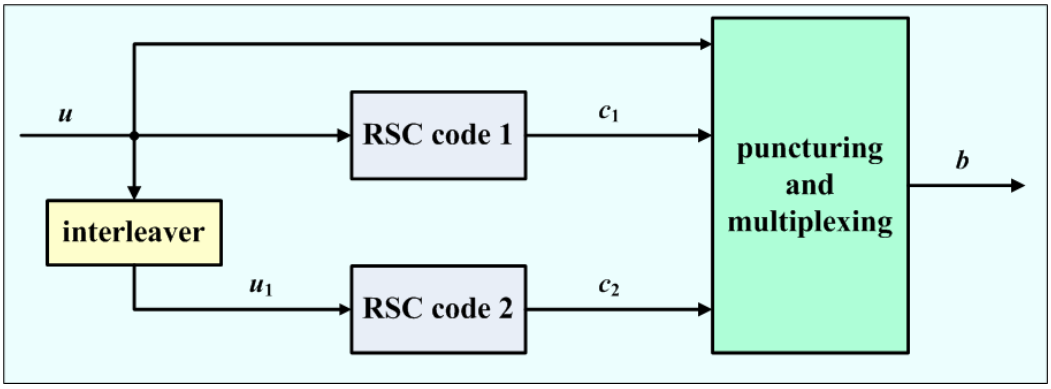

Fig.18 Turbo Code encoder

A typical turbo encoder consist also of a puncturing block. In a simple implementation of it only odd (or even) bits of $\boldsymbol{c}_1$ (and $\boldsymbol{c}_2$) could be actually sent. In this way the final rate will be raised to 1/2. Puncturing (that implies alteration in the decoding trellis) will be no longer considered in this derivation.

Let RSC1 and RSC2 belong to the same recursive systematic convolutional code whose rate is $R = k / q$, memory v. Such a code can be threathed as a CRSC (Circular RSC) if, using a constraint-length equal to Nk, all the input message words are given by a serial concatenation of τ k-bits-length non-null sequences plus v k-bits-length null sequences. Obviously it must results $N = \tau + v$.

In Fig.19 an example trellis is shown, relative to a case with $\tau = 4$ and $v = 2$.

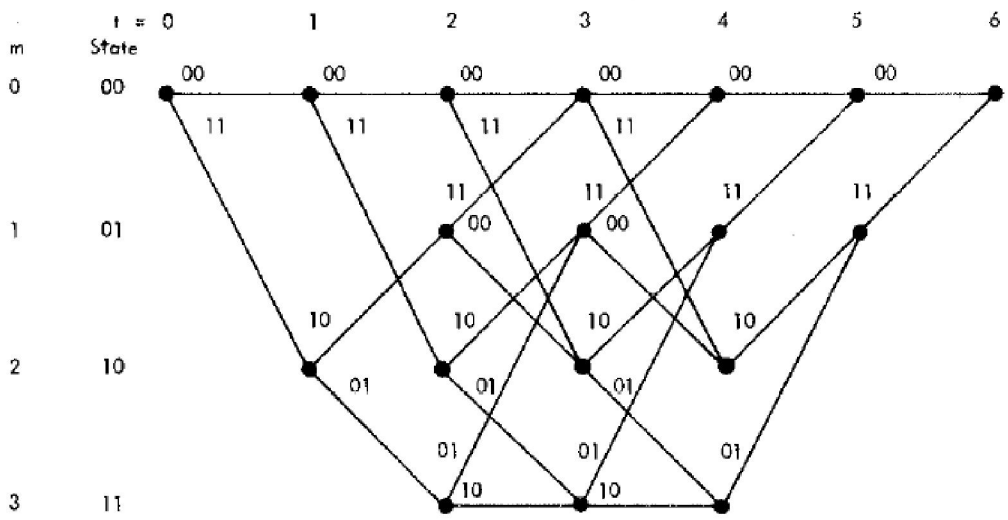

Fig.19 a CRSC decoding trellis example

Let k be equal to $\log_2(L)$ where L is the number of adopted constellation points. In this way, every k-bits length sequence in $\boldsymbol{u}$, is due to one symbol.

The information frame is thus:

$$\begin{aligned} u &= \left(u_1, u_2, \cdots, u_n, \cdots, u_N\right), \\ u_n &= \left(u_n^{(1)}, u_n^{(2)}, \cdots, u_n^{(\ell)}, \cdots, u_n^{(k)}\right), \quad u_n^{(\ell)} \in \{0,1\} \end{aligned} \qquad (141).$$

Let M[·] be the mapping operator, that links every k-tuple, to a constellation point.

In a non-punctured version of turbo code it should be:

$$\begin{aligned} \boldsymbol{b} &= \left(u_1, c_{1,1}, c_{2,1}, u_2, c_{1,2}, c_{2,2}, \cdots, u_n, c_{1,n}, c_{2,n}, \cdots, u_N, c_{1,N}, c_{2,N}\right), \\ c_{\eta,n} &= \left(c_{\eta,n}^{(1)}, c_{\eta,n}^{(2)}, \cdots, c_{\eta,n}^{(\ell)}, \cdots, c_{\eta,n}^{(k)}\right), \quad c_{\eta,n}^{(\ell)} \in \{0,1\}, \quad \eta \in \{1,2\} \end{aligned} \qquad (142).$$

Instead in a punctured version:

$$\boldsymbol{b} = \begin{cases} \left(u_1, c_{1,1}, u_2, c_{2,2}, \cdots, u_N, c_{2,N}\right), & \mathrm{mod}(N,2) = 0 \\ \left(u_1, c_{1,1}, u_2, c_{2,2}, \cdots, u_N, c_{1,N}\right), & \mathrm{mod}(N,2) = 1 \end{cases} \qquad (143).$$

During the receiving of the s-th frame, three type of soft input go into the first convolutional decoder or SISO (Soft Input Soft Output) decoder of Fig.20.

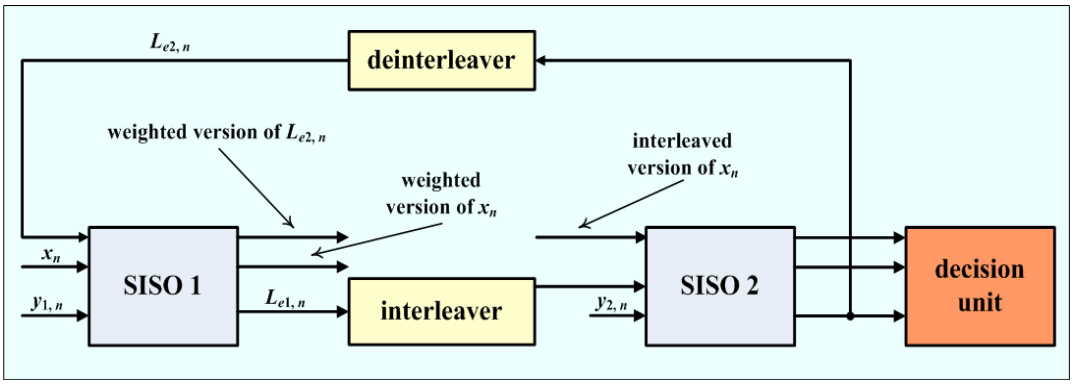

Fig.20 Turbo Code decoder

The first if this input is the noisy systematic component:

$$\boldsymbol{x} = \left(x_1, x_2, \cdots, x_n, \cdots, x_N\right) \qquad (144).$$

Denoting by $\mu_n = \mu_n^c + j\mu_n^s$ the constellation point received for the k-tuple u_n, defined in (15):

$$x_n = a_n \mu_n + w_n \qquad (145),$$

where a_n takes into account the attenuation (fading) due to the communication channel and w_n is an Additive White Gaussian Noise (AWGN). Equation (19) describes a pure AWGN channel when $a_n = 1$.

The second soft input is given by the noisy redundancy due to the first RSC, i.e.

$$\boldsymbol{y_1} = \left(y_{1,1}, y_{1,2}, \cdots, y_{1,n}, \cdots, y_{1,N}\right) \qquad (146).$$

If $\varepsilon_{1,n} = \varepsilon_{1,n}^c + j\varepsilon_{1,n}^s$ is the constellation point relative to the k-tuple $c_{1,n}$ defined in (16),

$$y_{1,n} = a_n \varepsilon_{1,n} + w_n \qquad (147).$$

The third soft input is given by the extrinsic information due to the second RSC:

$$\boldsymbol{L_{e2}} = \left(L_{e2,1}, L_{e2,2}, \cdots, L_{e2,n}, \cdots, L_{e2,N}\right) \qquad (148).$$

The SISO decoder 1 works every frame on the following soft inputs' set:

$$\boldsymbol{R_1} = \left(R_{1,1}, R_{1,2}, \cdots, R_{1,n}, \cdots, R_{1,N}\right) = \left(\boldsymbol{x}, \boldsymbol{y_1}, \boldsymbol{L_{e2}}\right) \qquad (149),$$

being $\boldsymbol{x}$, $\boldsymbol{y_1}$ and $\boldsymbol{L_{e2}}$ respectively expressed by (18), (20) and (22).

The SISO decoder 2 works instead on the following inputs' set:

$$\boldsymbol{R_2} = \left(R_{2,1}, R_{2,2}, \cdots, R_{2,n}, \cdots, R_{2,N}\right) = \left(\Pi\left(\boldsymbol{x}\right), \boldsymbol{y_2}, \Pi\left(\boldsymbol{L_{e1}}\right)\right) \qquad (150),$$

where $\Pi(\cdot)$ denotes the interleaving operator. Similarly:

$$\boldsymbol{y_2} = \left(y_{2,1}, y_{2,2}, \cdots, y_{2,n}, \cdots, y_{2,N}\right) \qquad (151),$$

$$y_{2,n} = a_n \varepsilon_{2,n} + w_n \qquad (152).$$

In (26) $\varepsilon_{2,n} = \varepsilon_{2,n}^c + j\varepsilon_{2,n}^s$ is the constellation point relative to the k-tuple $c_{2,n}$. At last:

$$\boldsymbol{L_{e1}} = \left(L_{e1,1}, L_{e1,2}, \cdots, L_{e1,n}, \cdots, L_{e1,N}\right) \qquad (153).$$

Both SISO 1 and 2 would be able to provide, every time T, LLRs (Log Likelihood Ratio) Λ_n, in order to establish which one of the L possible symbols was been transmitted. Actually only a part of this information, i.e. the extrinsic component, is forwarded, after a proper permutation, to the SISO relative to the other code, during all the iterations except the last. Indeed during the last iteration, while SISO 1 sends to SISO 2 $\Pi(\boldsymbol{L_{e1}})$, SISO 2 forwards to the decision unit all the LLR's information. Since $k = \log_2(L)$, the number of states of the decoding trellis is $M = 2^{kv} = L^v$.

As it has been said before, the symbol $u_n = u_{n_i}$, $i = 0, 1, \cdots, L-1$, is relative to the constellation element $\mu_n = \mu_n^c + j\mu_n^s$. It is clear that the decision must be separately taken on the inphase and quadrature channel.

Let L_c and L_s respectively be the number of possible values for μ_n^c and μ_n^s. Extending equation (8) of [8] to the L-ary case, if SISO 1 should not communicate with SISO 2 during the turbo decoding iterations, it could decide for binary k-tuple $u_n = u_{n_i}$, $i = 0, 1, \cdots, L-1$, relative to $\mu_n = \mu_n^c + j\mu_n^s$, choosing as $\mu_n^c = \mu_{n_i}^c$, $i \in \{0, 1, \cdots, L_c - 1\}$, the one that leads to the highest LLR:

$$\Lambda_{n_c}^i = \ln\left[\Pr\left(\mu_n^c = \mu_{n_i}^c \mid \boldsymbol{R_1}\right)\right], \; i \in \{0, 1, \cdots, L_c - 1\} \qquad (154),$$

while as $\mu_n^s = \mu_{n_i}^s$, $i \in \{0, 1, \cdots, L_s - 1\}$, is chosen the one that leads to the highest:

$$\Lambda_{n_s}^i = \ln\left[\Pr\left(\mu_n^s = \mu_{n_i}^s \mid \boldsymbol{R_1}\right)\right], \; i \in \{0, 1, \cdots, L_s - 1\} \qquad (155).$$

The entire process described from equations (18)-(27), happens on two distinct channel (inphase and quadrature):

$$\begin{aligned} \boldsymbol{R_1^c} &= \left(R_{1,1}^c, R_{1,2}^c, \cdots, R_{1,n}^c, \cdots, R_{1,N}^c\right) = \left(\boldsymbol{x^c}, \boldsymbol{y_1^c}, \boldsymbol{L_{e2}^c}\right) \\ \boldsymbol{R_1^s} &= \left(R_{1,1}^s, R_{1,2}^s, \cdots, R_{1,n}^s, \cdots, R_{1,N}^s\right) = \left(\boldsymbol{x^s}, \boldsymbol{y_1^s}, \boldsymbol{L_{e2}^s}\right) \end{aligned} \qquad (156),$$

being $\boldsymbol{x^c}$, $\boldsymbol{x^s}$, $\boldsymbol{y_1^c}$ and $\boldsymbol{y_1^c}$ respectively Re($\boldsymbol{x}$), Im($\boldsymbol{x}$), Re($\boldsymbol{y_1}$) and Im($\boldsymbol{y_1}$), where $\boldsymbol{x}$ and $\boldsymbol{y_1}$ are defined in (18) and (20). Similarly:

$$\begin{aligned} \boldsymbol{R_2^c} &= \left(R_{2,1}^c, R_{2,2}^c, \cdots, R_{2,n}^c, \cdots, R_{2,N}^c\right) = \left(\Pi\left(\boldsymbol{x^c}\right), \boldsymbol{y_2^c}, \Pi\left(\boldsymbol{L_{e1}^c}\right)\right) \\ \boldsymbol{R_2^s} &= \left(R_{2,1}^s, R_{2,2}^s, \cdots, R_{2,n}^s, \cdots, R_{2,N}^s\right) = \left(\Pi\left(\boldsymbol{x^s}\right), \boldsymbol{y_2^s}, \Pi\left(\boldsymbol{L_{e1}^s}\right)\right) \end{aligned} \qquad (157).$$

It is important to understand the way to evaluate the LLR and how extrinsic information can be extracted from them Remembering that $M = L^v$:

$$\begin{aligned} \Lambda_{n_c}^i &= \ln\left[\Pr\left(\mu_n^c = \mu_{n_i}^c \mid \boldsymbol{R_1^c}\right)\right] = \ln\left[\sum_{m=0}^{M-1} \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m, \boldsymbol{x^c}, \boldsymbol{y_1^c}, \boldsymbol{L_{e2}^c}\right)\right] = \\ &= \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^i\left(R_{1,n}^c, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \; i = 0, 1, \cdots, L_c - 1 \end{aligned} \qquad (158),$$

where S_n is convolutional encoder state at time n. The forward recursion [9] for a MAPSSE (Maximum A posteriori Probability Single Symbol Estimator) can be expressed as:

$$\alpha_n^i(m) = \sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^r\left(R_{1,n}^c, m', m\right)\alpha_{n-1}^i\left(m'\right), \; i = 0, 1, \cdots, L_c - 1 \qquad (159).$$

For the backward recursion [9] it instead results:

$$\beta_n(m) = \sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_{n+1}^r\left(R_{1,n+1}^c, m', m\right)\beta_{n+1}\left(m'\right) \qquad (160),$$

where the branches' transitions probabilities are given by the following set of equations:

$$\begin{aligned}\gamma_n^i\left(R_{1,n}^c, m', m\right) &= p\left(x_n^c \mid \mu_n^c = \mu_{n_i}^c\right) p\left(y_{1,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)\cdot \\ &\cdot p\left(L_{e2,n}^c \mid \mu_n^c = \mu_{n_i}^c\right)\cdot \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right),\ i = 0, 1, \cdots, L_c - 1\end{aligned} \quad (161),$$

$$p\left(x_n^c \mid \mu_n^c = \mu_{n_i}^c\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(x_n^c - \mu_{n_i}^c\right)^2}{2\sigma^2}} \quad (162),$$

$$p\left(y_{1,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(y_{1,n}^c - \varepsilon_{1,n_i}^c\right)^2}{2\sigma^2}} \quad (163),$$

$$p\left(L_{e2,n}^c \mid \mu_n^c = \mu_{n_i}^c\right) = \frac{e^{\mu_{n_i}^c L_{e2,n_i}^c}}{\sum_{r=0}^{L_c-1} e^{\mu_{n_r}^c L_{e2,n_r}^c}} \quad (164).$$

At the first iteration $\left(L_{e2,n_0}^c, L_{e2,n_1}^c, \cdots, L_{e2,n_i}^c, \cdots, L_{e2,n_{L_c-1}}^c\right) = (0, 0, \cdots, 0, \cdots, 0)$ implies that the probability $p\left(L_{e2,n}^c \mid \mu_n^c = \mu_{n_i}^c\right) = 1/L_c$ (total absence of a priori information). Moreover, the quantity $\Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right)$ is similar to $q_t\left(X, m, m'\right)$ defined in [9]. In other words $\Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right)$ represents the probability that, once transmitted a *k*-tuple u_n relative to a constellation point whose real part is $\mu_{n_i}^c$, the state after $S_{n-1} = m'$ is $S_n = m$. Differently from [9], these quantities will not be only 0 or 1. Indeed a sequence u_n is the one relative to the transition from m to m', if $\mathrm{Re}\left(\mu_n\right) = \mu_n^c = \mu_{n_i}^c$ and, at the same time $\mathrm{Im}\left(\mu_n\right) = \mu_n^s = \mu_{n_j}^s$. Thus, if the number of sequences with $\mu_n^c = \mu_{n_i}^c$ is n_ℓ, the probability of interest is $1/n_\ell$.

Inserting relations (35)-(38) into equation (32), it results:

$$\Lambda_{n_c}^{i} = \mu_{n_i}^{c} L_{e2,n_i}^{c} - \ln\left(\sum_{r=0}^{L_c-1} e^{\mu_{n_r}^{c} L_{e2,n_r}^{c}}\right) - \frac{1}{2\sigma^2}\left[\left(x_n^c - \mu_{n_i}^c\right)^2 + \sigma^2 \ln\left(2\pi\sigma^2\right)\right] + \\ + \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^i\left(y_{1,n}^c, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \quad i = 0, 1, \cdots, L_c - 1 \tag{165}$$

$$\gamma_n^i\left(y_{1,n}^c, m', m\right) = p\left(y_{1,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right) \cdot \\ \cdot \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right), \quad i = 0, 1, \cdots, L_c - 1 \tag{166}$$

where $p\left(y_{1,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)$ and $\Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right)$ are defined in [8]. Equation (39) is based on the concept of MAPSSE2, that in [8] is considered as deeply different from MAPSSE1, that requires also the estimation of σ_L.

It is important to observe that the third term of (39) quite represents the *i*-th in-phase extrinsic information, that is, during the iterations, forwarded to the other decoder.

In a similar way, it can be obtained, for the quadrature channel:

$$\Lambda_{n_s}^{i} = \mu_{n_i}^{s} L_{e2,n_i}^{s} - \ln\left(\sum_{r=0}^{L_s-1} e^{\mu_{n_r}^{s} L_{e2,n_r}^{s}}\right) - \frac{1}{2\sigma^2}\left[\left(x_n^s - \mu_{n_i}^s\right)^2 + \sigma^2 \ln\left(2\pi\sigma^2\right)\right] + \\ + \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_s-1} \gamma_n^i\left(y_{1,n}^s, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right], \quad i = 0, 1, \cdots, L_s - 1 \tag{167}$$

$$\gamma_n^i\left(y_{1,n}^s, m', m\right) = p\left(y_{1,n}^s \mid \mu_n^s = \mu_{n_i}^s, S_n = m, S_{n-1} = m'\right) \cdot \\ \cdot \Pr\left(\mu_n^s = \mu_{n_i}^s, S_n = m \mid S_{n-1} = m'\right), \quad i = 0, 1, \cdots, L_s - 1 \tag{168}$$

The third term of (41) quite represents the *i*-th in-quadrature extrinsic information, that is, during the iterations, forwarded to the other decoder (SISO 2).

From the SISO 2 point of view, it results, for the in-phase channel

$$\Lambda_{n_c}^{i} = \ln\left[\Pr\left(\mu_n^c = \mu_{n_i}^c \mid \boldsymbol{R_2^c}\right)\right] = \ln\left[\sum_{m=0}^{M-1} \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m, \Pi\left(\boldsymbol{x^c}\right), \mathbf{y_2^c}, \Pi\left(\boldsymbol{L_{e1}^c}\right)\right)\right] =$$
$$= \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^i\left(R_{2,n}^c, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right],\ i = 0, 1, \cdots, L_c - 1$$

(169),

where $\alpha_n^i(m)$, $\beta_n(m)$, $\gamma_n^i\left(R_{2,n}^c, m', m\right)$, $p\left(y_{2,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)$ and $p\left(L_{e1,n}^c \mid \mu_n^c = \mu_{n_i}^c\right)$ defined like in equations (33), (34), (35), (37) and (38).

It finally results:

$$\Lambda_{n_c}^{i} = \mu_{n_i}^c L_{e1,n_i}^c - \ln\left(\sum_{r=0}^{L_c-1} e^{\mu_{n_r}^c L_{e1,n_r}^c}\right) - \frac{1}{2\sigma^2}\left[\left(x_n^c - \mu_{n_i}^c\right)^2 + \sigma^2 \ln\left(2\pi\sigma^2\right)\right] +$$
$$+ \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_c-1} \gamma_n^i\left(y_{2,n}^c, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right],\quad i = 0, 1, \cdots, L_c - 1 \qquad (170),$$

$$\gamma_n^i\left(y_{2,n}^c, m', m\right) = p\left(y_{2,n}^c \mid \mu_n^c = \mu_{n_i}^c, S_n = m, S_{n-1} = m'\right)\cdot$$
$$\cdot \Pr\left(\mu_n^c = \mu_{n_i}^c, S_n = m \mid S_{n-1} = m'\right),\ i = 0, 1, \cdots, L_c - 1 \qquad (171),$$

$$\Lambda_{n_s}^{i} = \mu_{n_i}^s L_{e1,n_i}^s - \ln\left(\sum_{r=0}^{L_s-1} e^{\mu_{n_r}^s L_{e1,n_r}^s}\right) - \frac{1}{2\sigma^2}\left[\left(x_n^s - \mu_{n_i}^s\right)^2 + \sigma^2 \ln\left(2\pi\sigma^2\right)\right] +$$
$$+ \ln\left[\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\sum_{r=0}^{L_s-1} \gamma_n^i\left(y_{2,n}^s, m', m\right)\alpha_{n-1}^r\left(m'\right)\beta_n(m)\right],\quad i = 0, 1, \cdots, L_s - 1 \qquad (172),$$

$$\gamma_n^i\left(y_{2,n}^s, m', m\right) = p\left(y_{2,n}^s \mid \mu_n^s = \mu_{n_i}^s, S_n = m, S_{n-1} = m'\right)\cdot$$
$$\cdot \Pr\left(\mu_n^s = \mu_{n_i}^s, S_n = m \mid S_{n-1} = m'\right),\ i = 0, 1, \cdots, L_s - 1 \qquad (173).$$

The terms defined in (45) and (47) represent the extrinsic information that, during the iterations, are forwarded to the SISO decoder 1. Actually also the L_c in-phase intrinsic information and the L_s in-quadrature intrinsic information, given by the first two terms of (44) and (46), contribute, at the last iteration, to decide about the transmitted symbol. In other

words, the binary k-tuple u_n is chosen, at the last iteration, as the one relative to the constellation point $\mu_n = \mu_n^c + j\mu_n^s$, where $\mu_n^c + j\mu_n^s$ is linked to the highest $\Lambda_{n_s}^i$ among the Ls of the (46).

A faster implementation of the turbo decoding process can be based on the SOVA, that is described in [10].

3.2 From Turbo Codes to Turbo Equalization

In this paragraph the joint encoding and equalization algorithm nicknamed as *Turbo Equalization* is presented. It will be shown as this technique can be obtained by an application of the more important turbo codes' concepts (likelihood ratio, extrinsic information, trellis, and so on) to a typical problem of radio communications: the channel equalization. This issue is critical when dealing with channel affected by severe multipath fading that implies ISI (Inter Symbol Interference).

First of all, it is useful to redefine the basic turbo encoding and decoding schemes, in a more general way, i.e. including a permutation not only the second redundancy component (third encoding branch), but also on the first one (second encoding branch), like depicted in Fig.21. Modulation and AWGN channel blocks are instead represented for purposes that regard the derivation of the turbo equalization from turbo coding and that will be clear in the following.

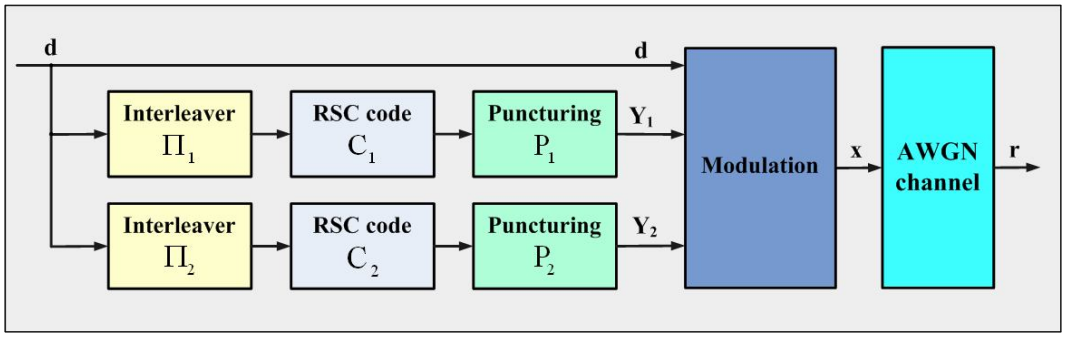

Fig.21 generalized basic turbo encoder

It is important to show that the turbo encoder shown in Fig.21 is not the most general, that should include until n branches, every one with an interleaver, a RSC and a puncturing block. There are fundamentally two reason why only $n = 2$ branches are taken into account in this

paragraph: a) when $n > 2$ the complexity of the decoding scheme grows as it is shown in [7]; b) the case with $n = 2$ is the more simple to be extended to the turbo equalization.

In Fig.22 is instead described the turbo decoder that works for data encoded through the scheme of Fig.21.

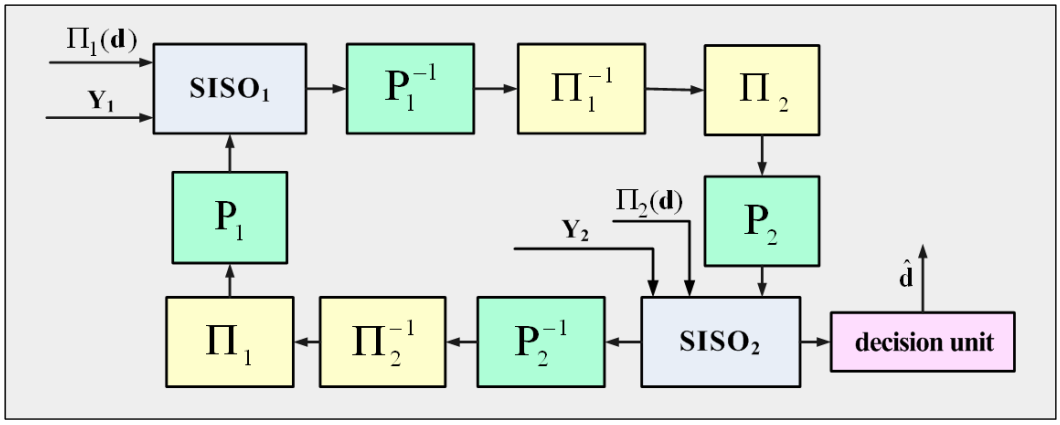

Fig.22 generalized basic turbo decoder

If the message is transmitted with a particular algebraic structure, like, for example, trough an interleaver and a CRSC code, this data's features can be used to better equalize the signal, i.e. to compensate the effect of multipath propagation. In these conditions, the receiver can be built as a concatenation of a SISO decoder and an equalizer. In [11] it is proofed that the equalizer can be modelled as another SISO element. In this paragraph the fundamental concepts will be described without entering into all the formal details.

In Fig.23 a typical data-transmission for which turbo equalization can be used for the receiver, is thus depicted.

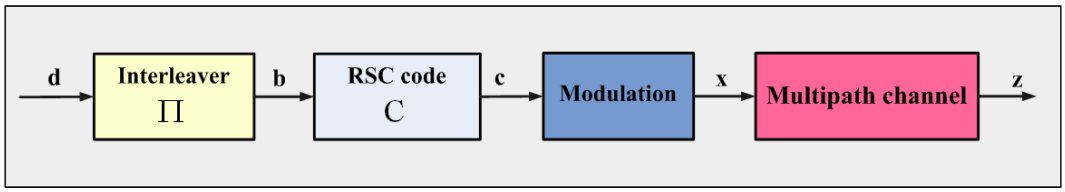

Fig.23 Transmission of data to be received using turbo equalization

Referring to a (n, k, N) CRSC code (N is the constraint-length) it can be said that the i-th encoded bits b_i depends on the message bits in accord with the following rule:

$$b_i = \sum_{j=0}^{v} c_j^{(i)} a_{i-j}\ , \quad i = 1, 2, \cdots, n \qquad (174),$$

where $c_j^{(i)}$ is the weight (that can be 0 or 1) relative to the j-th tap of the i-th shift register, and where v is the shift-register length or code memory. The state of the encoder is then represented by v bits that represent the values of the v bits a_{i-1}, a_{i-2}, ···, a_{i-v}. From every one of this state, that is a node of the CRSC trellis, one can go to two different states at the subsequent step, in accord to equation (48), and depending on the value of the bit current bit a_i. In a CRSC trellis, starting from the state S_0, i.e. $\begin{bmatrix} a_{i-1} & a_{i-2} & \cdots & a_{i-v} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$, after N steps the it comes back to S_0. The number of possible state is then 2^v.

In a ISI channel, the current received symbol is given by a linear combination of L previous symbols, as summarize in equation (49)

$$y_i = \sum_{j=0}^{L} \alpha_j x_{i-j} + w_i \qquad (175),$$

where w_i is an AWGN sample. If y_i refers to a base-band signal, it is clear that the coefficients α_i will be complex, i.e. introducing both an amplitude and phase distortion. It can be observed as the problem can be also treated using separate in-phase and in-quadrature derivations, i.e. considering $y_i^{(c)} = \sum_{j=0}^{L} \alpha_j^{(c)} x_{i-j}^{(c)} + w_i^{(c)}$ and $y_i^{(s)} = \sum_{j=0}^{L} \alpha_j^{(s)} x_{i-j}^{(s)} + w_i^{(s)}$, where $y_i = y_i^{(c)} + j y_i^{(s)}$ and so on for the other quantities. Anyway the derivation will here use the base-band analysis, referring to complex channel coefficients (every path consist of both an attenuated and phase-rotated version of the complex envelope). Once the channel coefficient [α_0, α_1, ···, α_L] have been estimated (through a blind or a data-aided algorithm), a channel-trellis can be built, starting from the observation that the current y_i value depends on the channel coefficient and on the values of the L previous symbols x_{i-1}, x_{i-2}, ···, x_{i-L}. Every trellis state represents a linear combination of L symbols where the weights are given by the channel coefficients. Thus, the number of possible state will be M^L, where M is the number of points of the adopted constellation. From every one of this state, that is a node of the channel trellis, one can go to

M different states at the subsequent step, in accord to equation (49), and depending on the value of the current symbol x_i.

In [11] it is proofed that the algorithm [9] can be used to iteratively equalize and decode signal transmitted, like shown in Fig.23, on multipath channel. In other words, the SISO decoder due to the convolutional code, gives the equalizer some information, or better only the *extrinsic* information (in the form of LLRs), and viceversa. (How to get extrinsic information from the global LLRs is described in [11]). More simply, the channel coding takes advantage of channel equalization and viceversa (Fig.24).

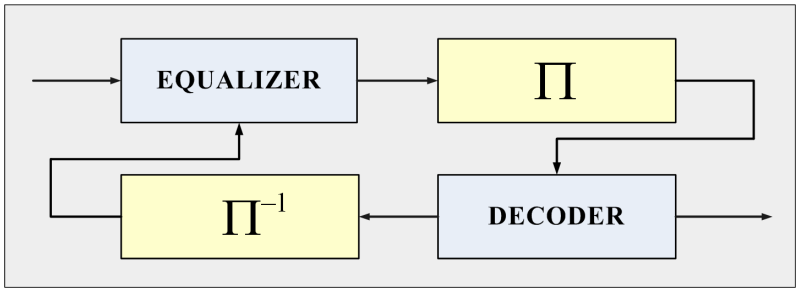

Fig.24 Turbo Equalizer

Turbo Equalization is really an application of Turbo Coding schemes to the problem of the channel equalization as it can be noted making comparison between Fig.22-23 and 23-24.

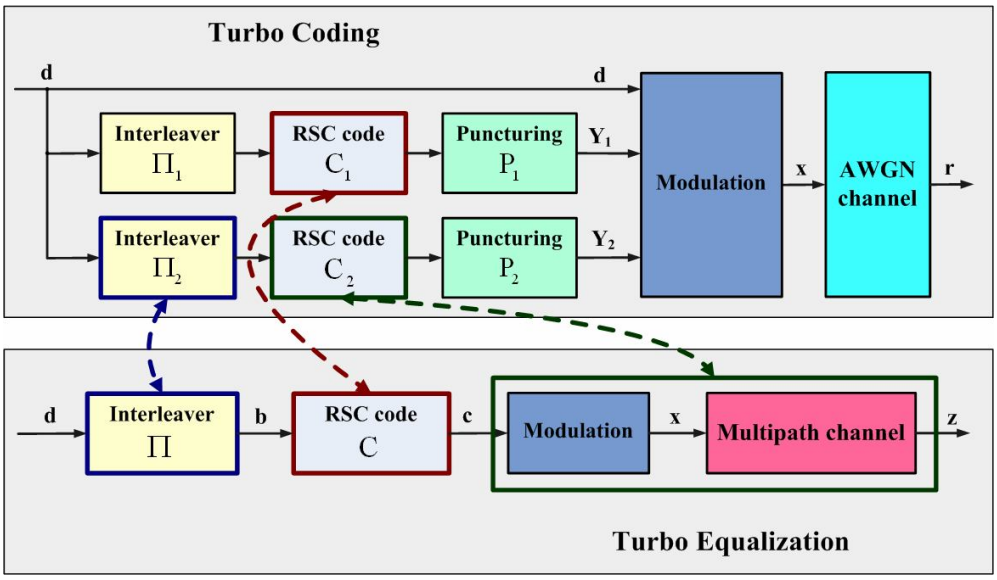

Fig.25 Turbo Coding and Turbo Equalization (transmitter side)

More precisely the interleaver Π of Fig.23 acts the role of the interleaver Π_2 of Fig.21; the RSC code *C* of Fig.23 does the same operations of the RSC code C_1 of Fig.21; the hyper-block Modulation + Multipath Channel of Fig.23 behaves as the RSC code C_2 of Fig.24. These relationship are summarized in Fig.25.

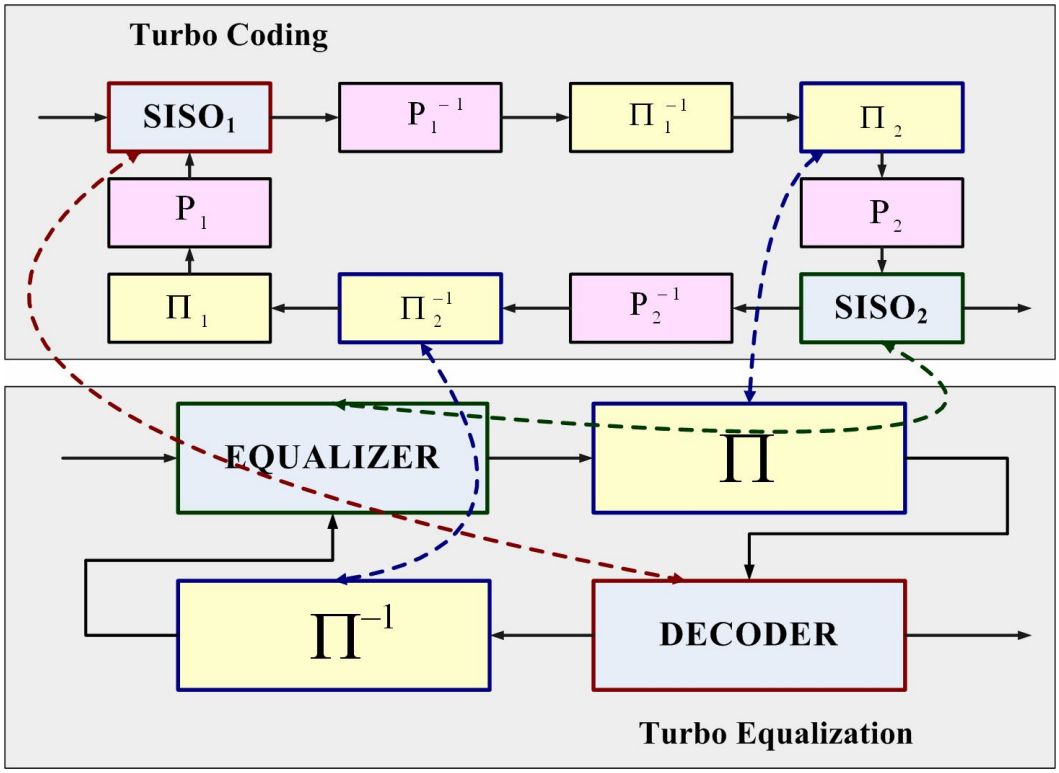

Fig.26 Turbo Coding and Turbo Equalization (receiver side)

From the receiver side, the equalizer and the decoder of Fig.24 respectively act the role of the SISO 2 and the SISO 1 of Fig.22, like it is depicted in Fig.26.

3.3 Equalization and the session keys' approach

Turbo Equalization has been briefly introduced to propose a new security algorithm to be used in critical (from the communication channel point of view) scenarios.

Turbo Equalization needs the channel to be, in some way, known, or estimated. Time-domain training sequences, or pilot-carriers are often used to evaluate the channel impulse response. While the time-domain approach can be used either in single or multi-carrier systems, the

frequency-domain choice can be obviously used only in multi-carrier (for example OFDM) communications. The number of freedom-degree in time (or frequency) positioning of the known (to both transmitter and receiver) data is absolutely limited: a training sequences should be a preamble of an information frame; pilot carriers should be equally-spaced in order to achieve the best channel frequency-transfer-function evaluation. Moreover, an estimation of the channel coefficients performed to adopt a turbo equalization on the receiver side, will make no sense in OFDM systems that could suffer, from ISI point of view, only from inter-frame-interference. The last phenomena is rare since it is extremely difficult that the multipath propagation delay T_m could overlap the OFDM frame duration. In such a system, it is more useful to understand, for every data-carrier, the impact of the communication channel on the sub-constellation points, in terms of attenuation and phase-rotation.

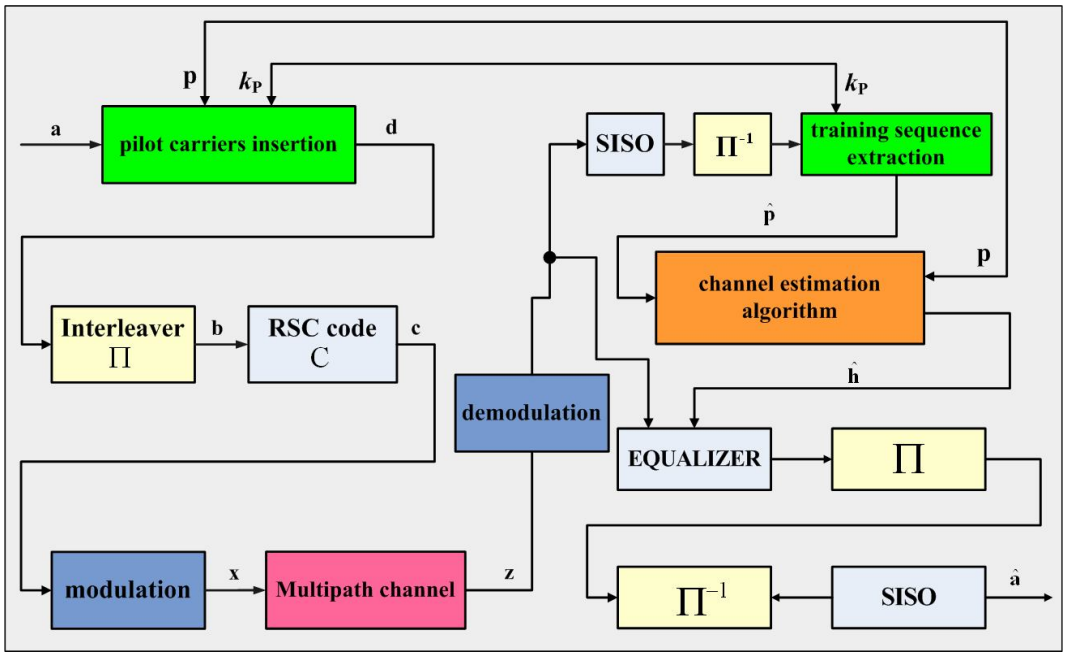

Fig.27 security algorithm based on training sequences and turbo equalization

Referring to the notation introduced in Chapter 1, a private session keys' set $\boldsymbol{k_P} = k_P$ (see Fig.1), could decide: a) the time-position of a training sequence of a system where data are convolutionally encoded, and the receiver makes use of the turbo equalization technique; b) the frequency location of the pilot-carriers in a OFDM system.

Solution a) is described in Fig.27 where the private session keys' set $\boldsymbol{k_P}$ is made of two components: the key k_P, that establishes the time position of the training sequence and **p**, i.e. the vector that collects the values of the training bits.

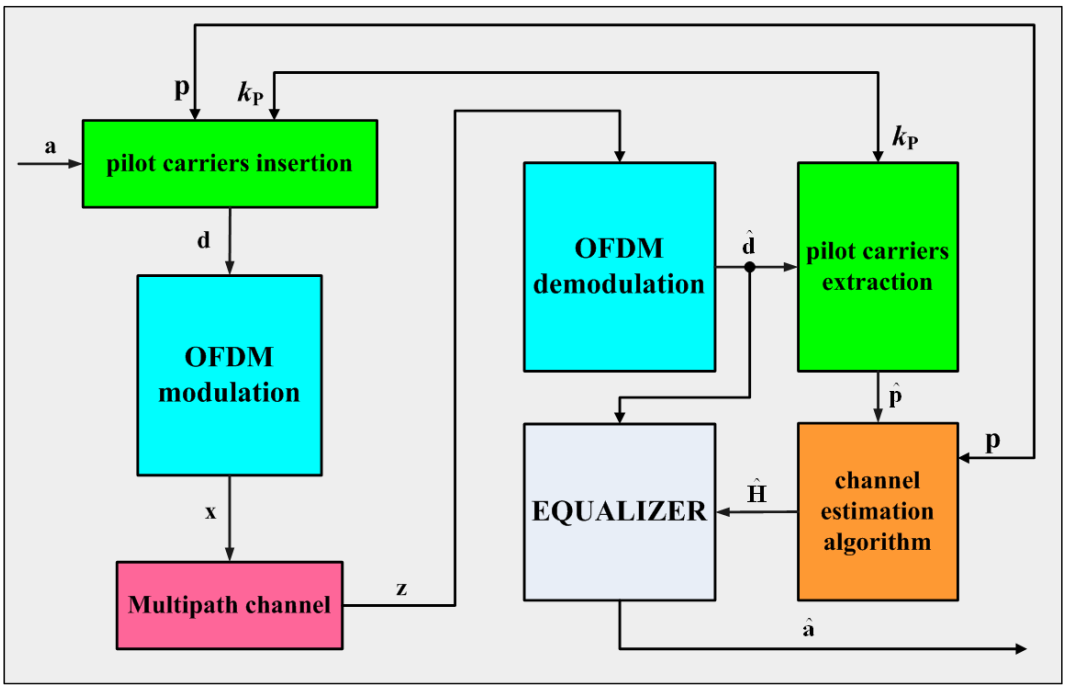

Fig.28 security algorithm based on OFDM pilot carriers positioning

Solution b) is instead depicted in Fig.28 where the private session keys' set $\boldsymbol{k_P}$ is made of two components: the key k_P, that establishes the frequency location of the pilot carriers and **p**, i.e. the vector that collects the values of the pilot symbols.

Among the two proposal, b), depicted in Fig.28, has been also tested in [12] and a comparison, in terms of FRR (False Rejection Rate) monte-carlo simulations between equally and pseudo-random spaced pilot-carriers OFDM systems is depicted in Fig.29. Let P be the number of pilot-carriers, $p = [n_1, n_2, \cdots, n_P]$ be the pilot-carriers array of indexes decided by k_P. Let instead $\boldsymbol{a} = \left[a_{n_1}, \quad a_{n_2}, \quad \cdots, \quad a_{n_P}\right]$ be the expected symbol in that carriers. (Normally, they all have the same values to make the pilots more detectable and to be an help for carrier synchronization, but a more sophisticated version of this security algorithm would use the same key k_P to establish also which values must be sent on pilot carriers). It is clear that, at

the OFDM receiver side, if $\hat{\boldsymbol{a}} = \begin{bmatrix} \hat{a}_{n_1} & \hat{a}_{n_2} & \cdots & \hat{a}_{n_P} \end{bmatrix}$ is the pilots recovered array, the channel frequency transfer function evaluated samples would be exactly $H_{n_i} = \hat{a}_{n_i} / a_{n_i}$, $i = 1, 2, \cdots, P$, while the remaining samples can be interpolated.

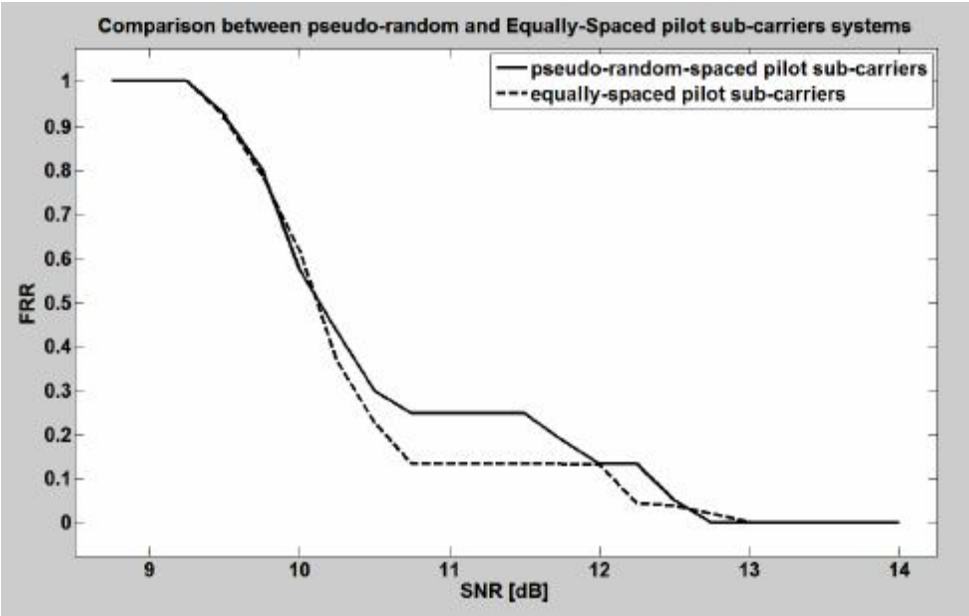

Fig.29 comparison between pseudo-random and equally spaced pilots OFDM systems

Fig.29 refers to an OFDM system with N_{ofdm} = 8192 while P = 256 is the number of pilot carriers. A QPSK constellation was adopted on every sub-band.

4 Security and Channel Coding

As enlightened by the design of the most recent IEEE standards, like IEEE 802.16e, modern wireless access networks have to provide effective security mechanisms even in severely adverse conditions, like those involved by indoor and outdoor scenarios affected by multipath and weather.

Source authenticity and message integrity are usually verified by means of solutions designed for noiseless situations, since in many works the authentication theory is separated from the coding theory. Referring to a communications scenario in which a transmitter attempts to inform a remote receiver of the state of a source by sending messages through an imperfect communications channel. There are two fundamentally different ways in which the receiver can end up being misinformed. The channel may be noisy so that symbols in the transmitted message can be received in error, or the channel may be under control of an opponent who can either deliberately modify legitimate messages or else introduce fraudulent ones to deceive the receiver, i.e. an “active wiretapper”. The device by which the receiver improves his chances of detecting error (deception) is the same in either case: the deliberate introduction of redundant information in to the transmitted message. For a statistically described noisy channel, coding theory is concerned with schemes (codes) that introduce redundancy in such a way that the most likely alternations to the encoded messages are in some sense close to the code they derive from. The receiver can then use a maximum likelihood detector to decide which (acceptable) message he should infer as having been transmitted form the (possibly altered) cod that was received. In other words, the object of coding theory is to cluster the most likely alterations of an acceptable codes as closely as possible (in a appropriate metric) to the code itself, an d disjoint from the corresponding clusters about other acceptable codes. To provide some degree of immunity to deception (of the receiver) the transmitter also introduces redundancy in this case, but does so in such a way

that, for any message the transmitter may send, the altered messages that the opponent would introduce using his optimal strategy are spread randomly, i.e., as uniformly as possible (again with respect to an appropriate metric) over the set of possible messages $\mathcal{M}$. Authentication theory is concerned with devising and analyzing schemes (A-codes) to achieve this spreading. It is in this sense that coding theory and authentication theory are dual theories: one is concerned with clustering the most likely alterations as closely about the original code as possible and the other with spreading the optimal (to the opponent) as uniformly as possible over $\mathcal{M}$.

In authentication, there are three participants: a transmitter who observes an information source $\mathcal{S}$ an d wishes to communicate these observations to a remotely located receiver over a publicly exposed, noiseless, communications channel and a receiver who wishes to not only learn what the transmitter hash observed but also to assure himself that the messages that he receives actually came from the transmitter and that no alterations have been made in transit to messages sent by the transmitter. The third participant, the opponent, wishes to deceive the receiver into accepting a message that will misinform him as to the state of the source. He can achieve this in either of two ways: by impersonating the transmitter an d sending a fraudulent message to the receiver when in fact none has been sent by the transmitter, or else by waiting and intercepting a message sent by the transmitter and substituting some other message.

Moreover, in this chapter, we will show how to formally described the authentication in a noiseless channel, i.e. a source state $s \in \mathcal{S}$ is encoded into the message $m \in \mathcal{M}$, the Simmons' theory is enough, but if we want to analytically characterize also the problem of a communication over a noisy channel, i.e. $s \in \mathcal{S}$ is encoded into $m \in \mathcal{M}$ that becomes $x \in \mathcal{X}$, we also introduce a Neyman-Pearson procedure that makes use of threshold that is adaptive, i.e. dependent on channel conditions.

The traditional approach isolates the security mechanisms from the physical layer, implementing them at higher levels of the ISO-OSI stack. These techniques are based on the use of a message Authentication Code (AC), often denoted as message hash, whose value

depends on two functionally distinct parameters, the message to be authenticated and a secret key, supposed unknown to any adversary. Data authenticity and integrity are verified by controlling the coincidence between the received hash and the hash computed on the received message by the authenticator. The AC design should guarantee that for the opponent it should be at least computationally infeasible to recover the unknown key and to compute the authentication code for any other message, even when he can observe one or more (message, hash) pairs. Since ACs are intrinsically sensitive to errors, strong forward error correction (FEC) codes and ARQ protocols have to be adopted in severe environments.

However, recent studies on wiretap channels demonstrated that phenomena like fading, considered in the past as potentials sources of impairments, can be exploited not only to increase the spectral efficiency, for instance, through the use of MIMO, but even to achieve the secrecy capacity limit, [13]. This modern approach requires a tighter cooperation between channel coding and security mechanisms. Moreover, implementation of encryption, mutual authentication, and data integrity mechanisms at physical and link levels allows even to reduce the security overhead, thus additionally increasing the overall spectral efficiency, and to deploy more effective countermeasures facing denial of service or hijacking attacks.

Thus, recently, several authors have investigated the use of FEC codes for authentication purposes, [14]-[27]. In [14], Kabatianskii et Al. analyzed the theoretical relationships between authentication codes and FECs. In [15] an asymmetric authentication scheme that makes use of the McEliece public-key cryptosystem, based on Goppa codes has been proposed. Digital signature schemes based on the McEliece systems have been further investigated in [16] and [17]. A more efficient approach, from a computational complexity point of view, has been proposed by Rao and Nam, that keep the public generator matrix used in the McEliece technique as private [22]-[25].

A security framework for OFDM systems combining encryption, authentication and channel coding at physical layer, has been proposed in [2]. In essence, joint authentication and error correction rely on a 128 bit Message Digest (MD-2) encrypted-hash algorithm. Despite its effectiveness, this method presented some lack of flexibility in managing the amount of redundancy introduced by channel coding.

In [26] and [27] concatenated turbo coders whose interleaver and puncturing elements are selected on the basis of a secret session key have been proposed for joint FEC and security.
However, although using redundancy introduced by FEC for message authentication has a rather positive impact on the efficient use of the available bandwidth, strong error resilience and detection of deceptive attacks are still antithetic requirements. Therefore some form of trade off has to be applied to meet both constraints on BER and deception probability.
At this aim, in this paper we propose the adoption of the Neyman-Pearson procedure in order to decide about authenticity and integrity of the received data when an Authentication and Forward Error Correction code, denoted in the following as A-FEC code, is employed. Consequently, verification of message authenticity and integrity is performed by comparing the posterior probability of the decoded message and hash pair, given the received noisy signal, with a threshold whose value is determined by the maximum acceptable level of probability of *false alarm* defined as the event of rejecting an authentic message because of the noise.
To meet additional constraints on bit error rate and on the probability of detecting any security attack we can properly tune several parameters characterizing an A-FEC code. To support an effective design, here, we investigate the relationship existing between bounds on impersonation, substitution and deception probability, as defined by Simmons, [28], and structural parameters characterizing the A-FEC code, like the cardinality of the set of different encoders that the authenticating party can effectively employ. One of the major outcomes of the performance analysis is that the introduction of a random permutation behind a systematic A-FEC coder guarantees unconditional security against attacks aimed at the authentication of a specific message (targeted deception) whenever the a priori probability of the input messages is uniform.
A special attention is devoted, here, to the analysis of the performance of A-FEC coders based on turbo-codes, because they present performance near the Shannon-limit, evaluation of the likelihood of the decoded sequence can be easily incorporated into the decoder, and, last but not least, their structure can be easily randomized acting, for instance, on the interleaver. In particular, since most random interleavers with large block-size exhibit BER. near the

theoretical limit and their number is so high to push deception probability bounds to extremely small values, we focus our analysis on small block-size interleavers, whose choice can strongly impact on achievable performance.

In literature several solutions have been proposed in order to attain a lower bit error rate. Among them semirandom interleavers appear to be rather attractive for their performance, computational complexity and simplicity of the associated mathematical model. Here, the trade-off between their block length and minimum distance with respect to both BER and deception probability is thus discussed.

4.1 Mathematical description

As illustrated in the introduction, the goal is to send an authenticated message over a noisy channel in presence of a malicious opponent that has access to a noisy wiretap channel, jointly providing protection against both the errors introduced by the noise, the impersonation attack where the opponent tries to insert a forged message on the noisy channel and the substitution attack where the opponent tries to substitute a part of an observed message with a false one. Here, the adoption of a two-stage system is proposed. The first stage provides encryption services, while the second stage jointly provides channel coding and mutual authentication and data integrity security services.

According to [14], a general *A*-FEC-code is a triple ($\mathcal{S}$,$\mathcal{E}$,$\mathcal{M}$) of finite sets and an invertible map $f:\mathcal{S}\times\mathcal{E}\rightarrow\mathcal{M}$, where $\mathcal{S}$ is the set of *source states*, $\mathcal{E}$ is the set of encoding rules, and for any $\mathbf{s}\in\mathcal{S}$ and $e\in\mathcal{E}$ the transmitted sequence $\mathbf{m}$ is defined by posing $\mathbf{m}=f(\mathbf{s},\mathbf{e})$. Let $\mathbf{y}^m\in\mathcal{Y}$ be the corresponding sequence received through the main channel and $\mathbf{y}^w\in\mathcal{Y}$ be the sequence received by the opponent through the wire-tap channel.

A general *A*-FEC-decoder is a triple ($\mathcal{Y}$,$\mathcal{E}$,$\mathcal{S}$), a decoding map $g:\mathcal{Y}\times\mathcal{E}\rightarrow\mathcal{S}$ and a verification test $v:\mathcal{Y}\times\mathcal{E}\rightarrow\{0,1\}$. A decoded message $\hat{\mathbf{s}}=g(\mathbf{y}^m,e)$ is said to be authentic if $v(\mathbf{y}^m,e)=1$.

In the remainder the attention will be focused on a particular class of *A*-FEC-coders, named *permuted systematic A*-FEC-codes. As defined in [14], a *systematic A*-FEC-code is a triple ($\mathcal{S}$,$\mathcal{E}_S$,$\mathcal{Z}$) of finite sets and an hash-parity map $h:\mathcal{S}\times\mathcal{E}_S \to \mathcal{Z}$, where for any $\mathbf{s}\in\mathcal{S}$ and encoding rule $e_S \in \mathcal{E}_S$ the transmitted sequence $\mathbf{m}=(\mathbf{s},\mathbf{z})\in\mathcal{S}\times\mathcal{Z}$ is defined by posing $\mathbf{z}=h(\mathbf{s},e_S)$, [14].

A systematic *A*-FEC-decoder is a quadruple ($\mathcal{X}$,$\mathcal{R}$,$\mathcal{E}$,$\mathcal{S}$), a decoding map $g:(\mathcal{X}\times\mathcal{R})\times\mathcal{E}\to\mathcal{S}$ and a verification test $v:(\mathcal{X}\times\mathcal{R})\times\mathcal{E}\to\{0,1\}$. A decoded message $\hat{\mathbf{s}}=g(\mathbf{X}^m,\mathbf{R}^m;e)$ is said to be authentic if $v(\mathbf{X}^m,\mathbf{R}^m;e)=1$.

An *A*-interleaver is a pair ($\mathcal{S}$,$\mathcal{E}_\Pi$) of finite sets and a permutation $\Pi:\mathcal{S}\times\mathcal{E}_\pi\to\mathcal{S}$, where for any $\mathbf{s}\in\mathcal{S}$ and encoding rule $e_\Pi\in\mathcal{E}_\Pi$ the permuted sequence $\mathbf{s}_\Pi\in\mathcal{S}$ is defined by posing $\mathbf{s}_\Pi=\Pi(\mathbf{s},e_\Pi)$. A *permuted systematic A*-FEC-code is defined as the cascade of an *A*-interleaver and a *systematic A*-FEC-code. Obviously, for a *permuted systematic A*-FEC-code the set of encoding rules is $\mathcal{E}=\mathcal{E}_\Pi\times\mathcal{E}_S$ and the transmitted sequence $\mathbf{m}=(\mathbf{s}_\Pi,\mathbf{z})\in\mathcal{S}\times\mathcal{Z}$ is defined by posing $\mathbf{s}_\Pi=\Pi(\mathbf{s},e_\Pi)$ and $\mathbf{z}=h(\mathbf{s}_\Pi,e_S)=h\left[\Pi(\mathbf{s},e_\Pi),e_S\right]$. The use of an *A*-inteleaver allows to drastically reduce the impersonation and substitution probability when the opponent want to authenticate a specific message and, when the source states are i.i.d., guarantees that the authentication scheme is unconditionally secure against targeted substitution attacks.

Let us respectively denote with $(\mathbf{x}^m,\mathbf{r}^m)\in\mathcal{X}\times\mathcal{R}$ and $(\mathbf{x}^w,\mathbf{r}^w)\in\mathcal{X}\times\mathcal{R}$ the main channel and wire-tap channel outputs when a *permuted systematic* A-FEC-coder transmits the sequence $\mathbf{m}=(\mathbf{s}_\Pi,\mathbf{z})$. The behaviour of the two channels is statistically described by means of the conditional distributions

$$q^m(\mathbf{x}^m,\mathbf{r}^m,\mathbf{s}_\Pi,\mathbf{z})=p_{\mathbf{X},\mathbf{R}|\mathbf{S}_\Pi,\mathbf{Z}}(\mathbf{x}^m,\mathbf{r}^m\,|\mathbf{s}_\Pi,\mathbf{z}) \qquad (176),$$

$$q^w(\mathbf{x}^w,\mathbf{r}^w,\mathbf{s}_\Pi,\mathbf{z})=p_{\mathbf{X},\mathbf{R}|\mathbf{S}_\Pi,\mathbf{Z}}(\mathbf{x}^w,\mathbf{r}^w\,|\mathbf{s}_\Pi,\mathbf{z}) \qquad (177).$$

A *permuted systematic* A-FEC-decoder is a quadruple ($\mathcal{X}$,$\mathcal{R}$,$\mathcal{E}$,$\mathcal{S}$), a decoding map $g:(\mathcal{X}\times\mathcal{R})\times\mathcal{E}\rightarrow\mathcal{S}$ and a verification test $v:(\mathcal{X}\times\mathcal{R})\times\mathcal{E}\rightarrow\{0,1\}$, where $\mathcal{E}=\mathcal{E}_{\Pi}\times\mathcal{E}_{s}$. A decoded message $\hat{\mathbf{s}}=g(\mathbf{x}^{m},\mathbf{r}^{m};\mathbf{e})$ is said to be authentic if $v(\mathbf{x}^{m},\mathbf{r}^{m};\mathbf{e})=1$, where $\mathbf{e}=(e_{\Pi},e_{S})$. Let H_0 denotes the hypothesis that the message has been altered or forged and H_1 the hypothesis that the received signal is the noisy version of the authentic original message. Following a Bayesian approach, decision about integrity and authenticity of the received data could be performed by thresholding the ratio between the posterior probabilities of the two hypotheses, namely,

$$\frac{\Pr\left\{\hat{\mathbf{s}}\middle|\mathbf{x}^{m},\mathbf{r}^{m},\mathbf{e}\right\}}{\sum_{\mathbf{s}_i\in F}\Pr\left\{\mathbf{s}_i\middle|\mathbf{x}^{m},\mathbf{r}^{m}\right\}} \qquad (178),$$

where F is the set of all forged/altered messages.

However this procedure appears to be unfeasible for both theoretical and practical concerns. From a theoretical point of view, its application would require the knowledge of the attack a priori probability distribution which is quite often unavailable. From a practical point of view, the computational burden prevents its application even in the rare cases for which one succeeds in attributing a value to the a priori probability of being attacked by an opponent.
Therefore, inspired to the radar context where similar situations arise, it can be proposed, as in [29], the adoption of the Neyman-Pearson procedure in order to decide about authenticity and integrity of the received data. Then, the authenticity and integrity verification test becomes:

$$v(\mathbf{x}^{m},\mathbf{r}^{m};e)=\begin{cases}1 & \log\Pr\left\{\hat{\mathbf{s}}/\mathbf{x}^{m},\mathbf{r}^{m},\mathbf{e}\right\}>\lambda\\ 0 & \textit{otherwise}\end{cases} \qquad (179).$$

According to the Neyman-Pearson criterion, the threshold λ is determined by the maximum acceptable level of probability of *false alarm* defined as the event of rejecting an authentic message because of the noise.

Once the threshold has been set, the remaining parameters of the *A*-FEC-coder can be chosen in order to maximize the probability of detecting any security attack. Obviously, a trade-off to meet additional constraints on bit error rate, maximum transmitting power, bit rate, hardware and software complexity, etc., may be required.

4.1.1 Permuted Turbo Codes

In Fig.30 is depicted a systematic coder given by a parallel turbo code, consisting of 2 interleavers Π_1, Π_2, cascaded with 2 recursive convolutional coders C_1, C_2, and two puncturing blocks P_1, P_2. The encoding rule randomly selects Π_1, Π_2, C_1, C_2, P_1 and P_2 from a predefined dictionary, based on the output of a pseudo-random generator driven by a secret key. Incidentally it can be observed that, to reduce the amount of memory required to store the dictionary, while keeping low the probability of use of coders with poor performance, a gray-list of poor elements can be employed.

The input to the transmitter is therefore constituted by the ciphertext $\mathbf{s}_\Pi$ that represents the systematic part of the turbo encoder, and the punctured parity/hash sequence $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$.

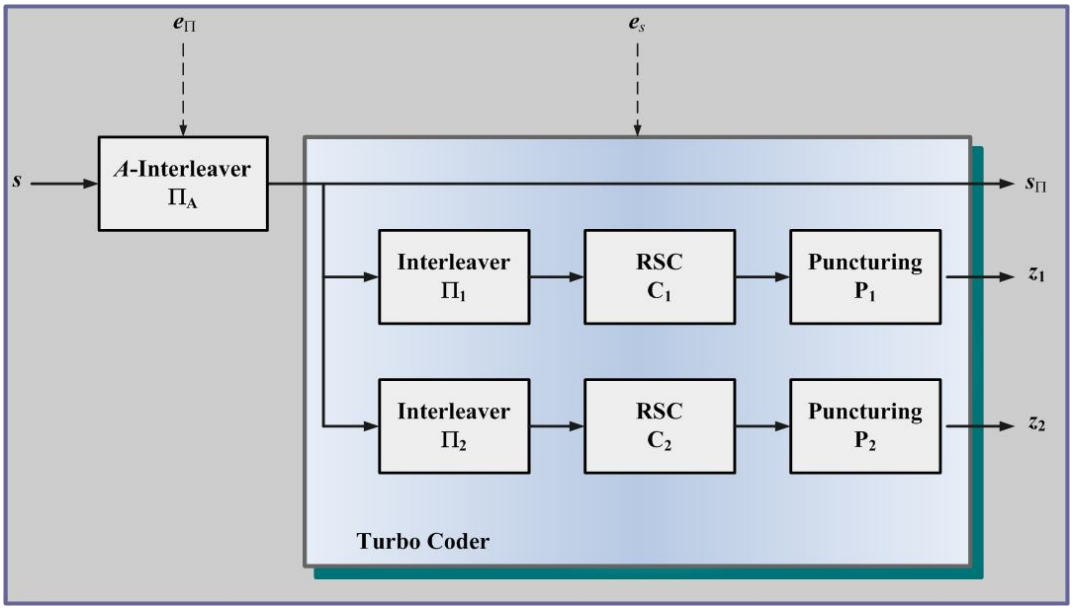

Fig.30 *A*-FEC permuted turbo coder

With reference to Fig.31, the receiver is constituted by a turbo decoder, [30], cascaded with a block that computes the likelihood of the (ciphertext, hash) pair, thus providing a soft authentication verification indicator. Hard decision is performed by thresholding the likelihood functional.

For AWGN channels, denoting with $\mu(\cdot)$ the mapping from the binary input to the adopted constellation's points, it results:

$$\boldsymbol{x} = \mu\left(\boldsymbol{s}_{\Pi}\right) + \boldsymbol{n} \qquad (180),$$

$$\boldsymbol{r}_1 = \mu\left(\boldsymbol{z}_1\right) + \boldsymbol{n}_1 \qquad (181),$$

$$\boldsymbol{r}_2 = \mu\left(\boldsymbol{z}_2\right) + \boldsymbol{n}_2 \qquad (182).$$

where $\boldsymbol{n} = \boldsymbol{n}_I + j\boldsymbol{n}_Q$, $\boldsymbol{n}_1 = \boldsymbol{n}_{1_I} + j\boldsymbol{n}_{1_Q}$ and $\boldsymbol{n}_2 = \boldsymbol{n}_{2_I} + j\boldsymbol{n}_{2_Q}$ are Circularly Complex White Gaussian Noise samples modelling the complex envelop of the receiver noise, whose in-phase and in quadrature components are respectively denoted by subscripts *I* and *Q*. Then, the log-likelihood $\log\Lambda(\hat{\boldsymbol{s}}; \boldsymbol{x}, \boldsymbol{r}_{1,}\boldsymbol{r}_2)$ of the decoded message, given the received signal can be written as follows:

$$\begin{aligned}\log\Lambda\left(\hat{\boldsymbol{s}}; \boldsymbol{x}, \boldsymbol{r}_{1,}\boldsymbol{r}_2\right) = \log\Pr\left\{\boldsymbol{x}, \boldsymbol{r}_{1,}\boldsymbol{r}_2 / \hat{\boldsymbol{s}}\right\} = -\frac{M}{2}\log 2\pi\sigma_N^2 - \\ -\frac{1}{2\sigma_N^2}\left[\boldsymbol{x} - \mu(\hat{\boldsymbol{s}}_{\Pi})\right]^{\dagger}\left[\boldsymbol{x} - \mu(\hat{\boldsymbol{s}}_{\Pi})\right] - \\ -\frac{1}{2\sigma_N^2}\left[\boldsymbol{r}_1 - \mu(\hat{\boldsymbol{z}}_1)\right]^{\dagger}\left[\boldsymbol{r}_1 - \mu(\hat{\boldsymbol{z}}_1)\right] - \frac{1}{2\sigma_N^2}\left[\boldsymbol{r}_2 - \mu(\hat{\boldsymbol{z}}_2)\right]^{\dagger}\left[\boldsymbol{r}_2 - \mu(\hat{\boldsymbol{z}}_2)\right]\end{aligned} \qquad (183),$$

where † denotes the Hermitian operator, σ_N^2 is the receiver noise variance, $\hat{\boldsymbol{z}}_1$ and $\hat{\boldsymbol{z}}_2$ are hash/parity, eventually punctured, estimated sequences corresponding to $\hat{\boldsymbol{s}}$, and *M* is the sum of the sizes of the complex vectors $\boldsymbol{x}$, $\boldsymbol{r}_1$, and $\boldsymbol{r}_2$.

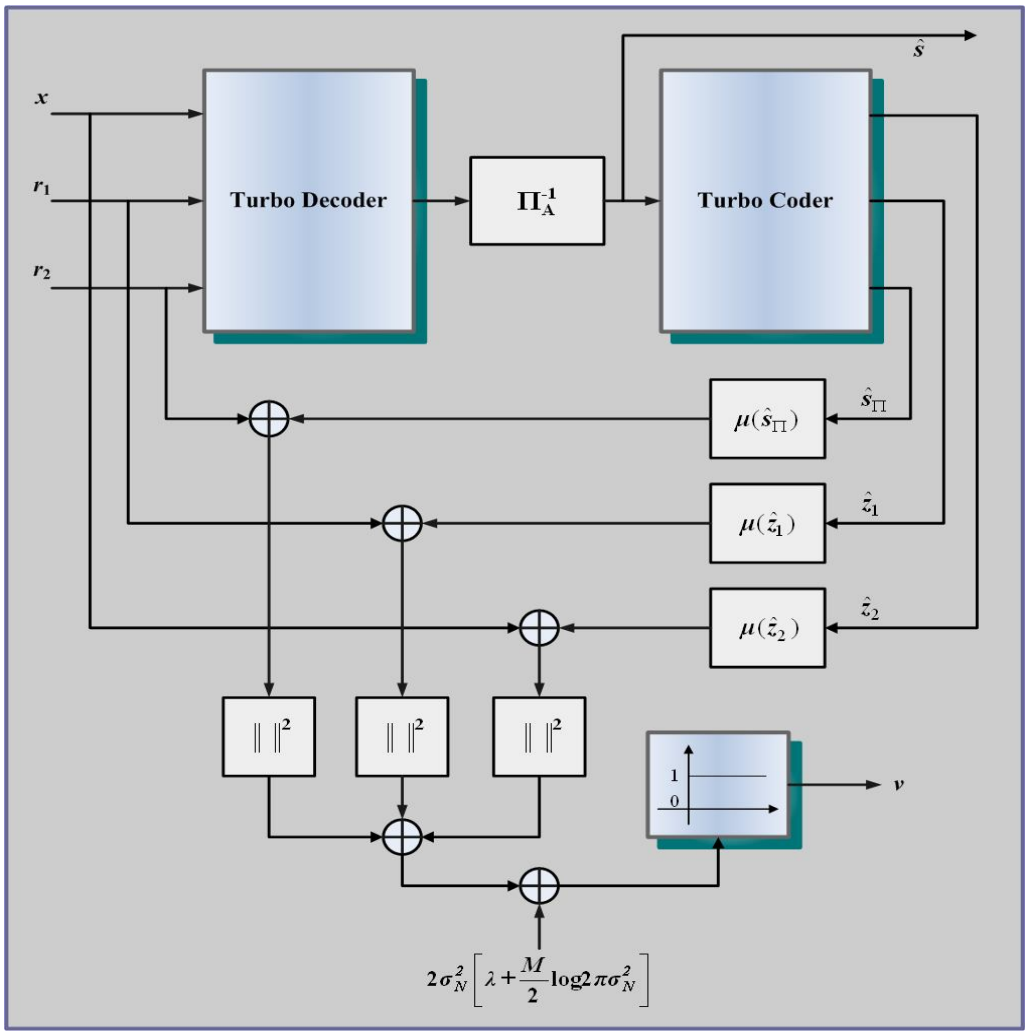

Fig.31 *A*-FEC permuted turbo decoder

To evaluate the false alarm probability P_{fa}, it can be observed that, when the bit error rate at the output of the decoder is small, $\hat{\boldsymbol{s}} \cong \boldsymbol{s}$, and thus, under the hypothesis H_1, the log-likelihood functional (57), can be well approximated as follows:

$$\log \Lambda\left(\hat{\boldsymbol{s}}; \boldsymbol{x}, \boldsymbol{r}_1, \boldsymbol{r}_2 / H_1\right) \cong -\frac{M}{2} \log 2\pi\sigma_N^2 - \frac{\nu}{2} \qquad (184),$$

where

$$\nu = \frac{1}{\sigma_N^2} \sum_{i=1}^{M} \left(n_{I_k}^2 + n_{Q_k}^2\right) \qquad (185)$$

is a random variate with a chi-square distribution with $2M$ degrees of freedom. Therefore, for the false alarm probability it results:

$$P_{fa} = \int_{-2\lambda - M\log 2\pi\sigma_N^2}^{\infty} p_{\chi_{2M}^2}(x)dx = \frac{\gamma\left(M, -\lambda - M\log 2\pi\sigma_N^2\right)}{(M-1)!} \qquad (186),$$

where $\gamma(k,z)$ is the upper incomplete Gamma function:

$$\gamma(a,x) = \int_x^{\infty} t^{a-1}e^{-t}dt \qquad (187).$$

Thus, the authenticity and integrity test threshold can be computed by numerical inversion of equation (60). Threshold adaptivity requires the on line estimation of the noise power spectrum density.

4.2 Performance analysis

The maximum probability of success of impersonation attack, denoted as P_I, is formally defined as:

$$\begin{aligned} P_I &= \underset{(\boldsymbol{x}^m,\boldsymbol{r}^m)\in\mathcal{X}\times\mathcal{R}}{Max} \Pr\left\{v(\boldsymbol{x}^m,\boldsymbol{r}^m,\boldsymbol{e}) = 1\right\} = \\ &= \underset{(\boldsymbol{x}^m,\boldsymbol{r}^m)\in\mathcal{X}\times\mathcal{R}}{Max} \log\Pr\left\{\hat{\boldsymbol{s}} / \boldsymbol{x}^m,\boldsymbol{r}^m,\boldsymbol{e}\right\} > \lambda \end{aligned} \qquad (188).$$

For a given sequence $(\boldsymbol{x}^m,\boldsymbol{r}^m)$, let $\mathcal{E}^m(\boldsymbol{x}^m,\boldsymbol{r}^m)$ be the set of the encoding rules that pass the authenticity and integrity test, namely

$$\mathcal{E}^m(\boldsymbol{x}^m,\boldsymbol{r}^m) = \left\{\boldsymbol{e}\in\mathcal{E}\,\middle|\,\log\Pr\left\{\hat{\boldsymbol{s}} / \boldsymbol{x}^m,\boldsymbol{r}^m,\boldsymbol{e}\right\} > \lambda\right\} \qquad (189).$$

And with $\mathcal{E}_S^m(\boldsymbol{x}^m,\boldsymbol{r}^m)$ the set of encoding rules of the systematic A-FEC code that pass the authenticity and integrity test, namely

$$\mathcal{E}_S^m(\boldsymbol{x}^m,\boldsymbol{r}^m) = \left\{e_S\in\mathcal{E}_S\,\middle|\,\underset{e_\Pi\in\mathcal{E}_\Pi}{Max}\log\Pr\left\{\hat{\boldsymbol{s}} / \boldsymbol{x}^m,\boldsymbol{r}^m,(e_\Pi,e_S)\right\} > \lambda\right\} \qquad (190).$$

Similarly, let us denote with $\mathcal{E}^{NF}(\boldsymbol{s},\boldsymbol{z})$ the set of the encoding rules that pass the authenticity and integrity test in the noise free case, i.e.:

$$\mathcal{E}^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z}) = \left\{ \boldsymbol{e} = (e_\Pi, e_S) \in \mathcal{E} \,\middle|\, \boldsymbol{s}_\Pi = \Pi(\boldsymbol{s}, e_\Pi),\ \boldsymbol{z} = h\left[\boldsymbol{s}_\Pi, e_S\right] \right\} \tag{191}$$

and with $\mathcal{E}_S^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z})$ the corresponding set of encoding rules of the systematic A-FEC code:

$$\mathcal{E}_S^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z}) = \left\{ e_S \in \mathcal{E}_S \,\middle|\, \boldsymbol{z} = h(\boldsymbol{s}_\Pi, e_S) \right\} \tag{192}.$$

Incidentally, it can be observed that, being Π a permutation, $\mathcal{E}^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z}) = \mathcal{E}_\Pi \times \mathcal{E}_S^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z})$.

Since, for AWGN channels, $\left\{ \boldsymbol{x} \,\middle|\, \boldsymbol{x} = \mu(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S} \right\} \subseteq \mathcal{X}$, $\left\{ \boldsymbol{r} \,\middle|\, \boldsymbol{r} = \mu(\boldsymbol{z}), \boldsymbol{z} \in \mathcal{Z} \right\} \subseteq \mathcal{R}$ and $\left|\mathcal{E}_S^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z})\right| \le \left|\mathcal{E}_S^{m}(\boldsymbol{s}_\Pi, \boldsymbol{z})\right|$, where $|\cdot|$ denotes the cardinality of a set. Then, assuming as in [14] that the opponent will randomly select the encoding rule, for the probability of success of impersonation attack the following bound holds:

$$P_I = \underset{(\boldsymbol{x}^m, \boldsymbol{r}^m) \in X \times R}{Max} \frac{\left|\mathcal{E}^m(\boldsymbol{x}^m, \boldsymbol{r}^m)\right|}{\left|\mathcal{E}\right|} \ge \underset{(\boldsymbol{s}_\Pi, \boldsymbol{z}) \in \mathcal{S} \times \mathcal{Z}}{Max} \frac{\left|\mathcal{E}^m\left(\mu(\boldsymbol{s}_\Pi), \mu(\boldsymbol{z})\right)\right|}{\left|\mathcal{E}_\Pi\right|\left|\mathcal{E}_S\right|} = \underset{(\boldsymbol{s}_\Pi, \boldsymbol{z}) \in \mathcal{S} \times \mathcal{Z}}{Max} \frac{\left|\mathcal{E}_S^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z})\right|\left|\mathcal{E}_\Pi\right|}{\left|\mathcal{E}_\Pi\right|\left|\mathcal{E}_S\right|} \tag{193}.$$

The rightmost term of (67) is the probability of success of impersonation attack for noise free channels, [13],

$$P_I^{NF} = \underset{(\boldsymbol{s}_\Pi, \boldsymbol{z}) \in \mathcal{S} \times \mathcal{Z}}{Max} \frac{\left|\mathcal{E}_S^{NF}(\boldsymbol{s}_\Pi, \boldsymbol{z})\right|}{\left|\mathcal{E}_S\right|} \tag{194}.$$

Therefore

$$P_I \ge P_I^{NF} \ge \frac{1}{\left|\mathcal{E}_S\right|} \tag{195}.$$

As expected the probability of impersonation attack increases and in presence of noise with a lower bound given by the inverse of the number of different systematic A-FEC codes.

In the substitution attack, the opponent observes the output $(\boldsymbol{x}^w, \boldsymbol{r}^w)$ of the wiretap channel and replaces the original message $\boldsymbol{s}$ with $\boldsymbol{s}'$. To maximize his probability of success, the opponent will use a parity hash sequence $\boldsymbol{z}'$ maximizing the cardinality of the intersection between $\mathcal{E}^w(\boldsymbol{x}^w, \boldsymbol{r}^w)$ and $\mathcal{E}^{NF}(\boldsymbol{s}', \boldsymbol{z}')$, i.e.,

$$\boldsymbol{z}' = Arg \underset{\boldsymbol{z} \in \mathcal{Z}}{Max} \left|\mathcal{E}^{NF}(\boldsymbol{s}', \boldsymbol{z}) \cap \mathcal{E}^w(\boldsymbol{x}^w, \boldsymbol{r}^w)\right| \tag{196}.$$

Thus, according to [13], the maximum probability of success of substitution attack P_S is:

$$P_S = \underset{\substack{(\boldsymbol{x}^w,\boldsymbol{r}^w)\in\mathcal{X}\times\mathcal{R}\\(\boldsymbol{x}',\boldsymbol{r}')\in\mathcal{X}\times\mathcal{R}\\(\boldsymbol{x}^w,\boldsymbol{r}^w)\neq(\boldsymbol{x}',\boldsymbol{r}')}}{Max} \Pr\left\{v(\boldsymbol{x}',\boldsymbol{r}',\boldsymbol{e})=1\middle|\boldsymbol{x}^w,\boldsymbol{r}^w\right\} \geq \underset{\substack{(\boldsymbol{s}',\boldsymbol{z}')\in\mathcal{S}\times\mathcal{Z}\\(\boldsymbol{x}^w,\boldsymbol{r}^w)\in\mathcal{X}\times\mathcal{R}\\\boldsymbol{s}'\neq\hat{\boldsymbol{s}}^w}}{Max} \frac{\Pr\left\{\mathcal{E}^{NF}(\boldsymbol{s}',\boldsymbol{z}')\cap\mathcal{E}^w(\boldsymbol{x}^w,\boldsymbol{r}^w)\right\}}{\Pr\left\{\mathcal{E}^w(\boldsymbol{x}^w,\boldsymbol{r}^w)\right\}} =$$

$$= \underset{\substack{(\boldsymbol{s}_\Pi,\boldsymbol{z}')\in\mathcal{S}\times\mathcal{Z}\\(\boldsymbol{x}^w,\boldsymbol{r}^w)\in\mathcal{X}\times\mathcal{R}\\\boldsymbol{s}'\neq\hat{\boldsymbol{s}}^w}}{Max} \frac{\left|\mathcal{E}_S^{NF}(\boldsymbol{s}_\Pi,\boldsymbol{z}')\cap\mathcal{E}_S^w(\boldsymbol{x}^w,\boldsymbol{r}^w)\right|}{\left|\mathcal{E}_S^w(\boldsymbol{x}^w,\boldsymbol{r}^w)\right|} \tag{197}$$

.

In the impersonation and substitution attacks as defined in [14], the scope of the opponent is to pass the authenticity and integrity test for any message, without taking care of the semantic meaning associated to the message maximizing either P_I or P_S. However, in many practical situations the opponent want to authenticate a target message, let say $\tilde{\boldsymbol{s}}$. In this case, considering that Π a permutation, the probability $\tilde{P}_I(\tilde{\boldsymbol{s}})$ of a successful targeted impersonation attack evaluates as follows:

$$\tilde{P}_I(\tilde{\boldsymbol{s}}) = \underset{\substack{\boldsymbol{z}\in\mathcal{Z}\\e_\Pi\in\mathcal{E}_\Pi}}{Max} \frac{\left|\mathcal{E}_S^m\left(\mu\left[\Pi(\tilde{\boldsymbol{s}},e_\Pi)\right],\mu(\boldsymbol{z})\right)\right|}{\left|\mathcal{E}_\Pi\right|\left|\mathcal{E}_S\right|} \tag{198}$$

Therefore

$$\underset{\tilde{\boldsymbol{s}}\in\mathcal{S}}{Max}\,\tilde{P}_I(\tilde{\boldsymbol{s}}) = \frac{P_I}{\left|\mathcal{E}_\Pi\right|} \tag{199}$$

Similarly, let denote by $\tilde{P}_S(\tilde{\boldsymbol{s}})$ the probability of successfully authenticating a target message, let say $\tilde{\boldsymbol{s}}$, having observed $(\boldsymbol{x}^w,\boldsymbol{r}^w)$ at the output of the wiretap channel. Then, proceeding as for the probability of targeted impersonation it can be easily verified that:

$$\underset{\tilde{\boldsymbol{s}}\in\mathcal{S}}{Max}\,\tilde{P}_S(\tilde{\boldsymbol{s}}) = \frac{P_S}{\left|\mathcal{E}_\Pi\right|} \tag{200}$$

Thus the initial permutation allows to reduce by a factor $\left|\mathcal{E}_\Pi\right|$ the impersonation and the substitution probability for targeted attacks.

Moreover, the opponent can only observe the noisy output of the A-interleaver together with the associated has-parity sequence. Therefore, when the source states are equally probable, the mutual information of $(\mathbf{X}^w,\mathbf{R}^w)$ and E_Π is null and the conditional entropy of E_Π given $(\mathbf{X}^w,\mathbf{R}^w)$ equals the a priory entropy of E_Π. Indeed,

$$p_{\mathbf{X}^w,\mathbf{R}^w|E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|e_\Pi) = \\ = \sum_{s_\Pi\in\mathcal{S}}\sum_{e_S\in\mathcal{E}_S} p_{\mathbf{X}^w,\mathbf{R}^w|\mathbf{S}_\Pi,E_S,E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|\boldsymbol{s}_\Pi,e_S,e_\Pi)P_{\mathbf{S}_\Pi|E_\Pi}(\boldsymbol{s}_\Pi|e_\Pi)P_{E_S|\mathbf{S}_\Pi,E_\Pi}(e_S|\boldsymbol{s}_\Pi,e_\Pi) \quad (201).$$

Observing that E_S is statistically independent from E_Π and S_Π, and from (51) for the AWGN channels it results

$$p_{\mathbf{X}^w,\mathbf{R}^w|\mathbf{S}_\Pi,E_S,E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|\boldsymbol{s}_\Pi,e_S,e_\Pi) = p_{\mathbf{X}^w,\mathbf{R}^w|\mathbf{S}_\Pi,E_S}(\boldsymbol{x}^w,\boldsymbol{r}^w|\boldsymbol{s}_\Pi,e_S) = q^w\left[\boldsymbol{x}^w,\boldsymbol{r}^w,\boldsymbol{s}_\Pi,h(\boldsymbol{s}_\Pi,e_S)\right]$$

(202)

$$p_{\mathbf{X}^w,\mathbf{R}^w|E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|e_\Pi) = \sum_{s_\Pi\in\mathcal{S}}\left\{\sum_{e_S\in\mathcal{E}_S} q^w\left[\boldsymbol{x}^w,\boldsymbol{r}^w,\boldsymbol{s}_\Pi,h(\boldsymbol{s}_\Pi,e_S)\right]P_{E_S}(e_S)\right\}P_{\mathbf{S}_\Pi|E_\Pi}(\boldsymbol{s}_\Pi|e_\Pi) \quad (203).$$

On the other hand, denoting by Π^{-1} the inverse permutation of Π, it results:

$$P_{\mathbf{S}_\Pi|E_H}(\boldsymbol{s}_\Pi|e_\Pi) = P_{\mathbf{S}}\left[\Pi^{-1}(\boldsymbol{s}_\Pi,e_\Pi)\right] \quad (204).$$

Therefore, if $\mathbf{S}$ is uniformly distributed, then $\mathbf{S}_\Pi$ is uniformly distributed too and statistically independent from E_Π, and the observation $(\mathbf{X}^w,\mathbf{R}^w)$ and the encoding rule random variate E_Π are mutually statistically independent, which in turn implies that the mutual information is null. In fact $P_{\mathbf{S}_\Pi|E_H}(\boldsymbol{s}_\Pi|e_\Pi) = P_{\mathbf{S}_\Pi}(\boldsymbol{s}_\Pi) = |\mathcal{E}|^{-1}$, thus:

$$p_{\mathbf{X}^w,\mathbf{R}^w|E_\Pi}(\boldsymbol{x}^w,\boldsymbol{r}^w|e_\Pi) = p_{\mathbf{X}^w,\mathbf{R}^w}(\boldsymbol{x}^w,\boldsymbol{r}^w) = \frac{1}{|\mathcal{S}|}\sum_{s_\Pi\in\mathcal{S}}\left\{\sum_{e_S\in\mathcal{E}_S} q^w\left[\boldsymbol{x}^w,\boldsymbol{r}^w,\boldsymbol{s}_\Pi,h(\boldsymbol{s}_\Pi,e_S)\right]P_{E_S}(e_S)\right\}$$

(205).

Finally it can be noted that when the selection of the permutation is purely random:

$$H(E_\Pi|\mathbf{X}^w,\mathbf{R}^w) = H(E_\Pi) = \log_2|\mathcal{E}_\Pi| \quad (206).$$

Therefore, when the opponent want to authenticate a target message, he can do no better than randomly select a permutation irrespective of the observations he made and the computing power and time he can spend. Therefore the permuted systematic A-FEC coder is unconditionally secure with respect to targeted substitution attacks.

For noisy channels the Simmon's bounds specifies as follows, [14], [31], [32],

$$P_I \geq 2^{-H(E)+H(E|\mathbf{X}^m,\mathbf{R}^m)} \quad (207),$$

$$P_S \geq 2^{-H(E|\mathbf{X}^w,\mathbf{R}^w)} \quad (208).$$

Thus, from (81) and (82) it follows that:

$$P_I P_S \geq 2^{-H(E)-\left[H(E|\mathbf{X}^w,\mathbf{R}^w)-H(E)+H(E|\mathbf{X}^m,\mathbf{R}^m)\right]} \quad (209).$$

Therefore, if the wiretap channel is more noisy than the main channel, $H(E|\mathbf{X}^m,\mathbf{R}^m) \leq H(E|\mathbf{X}^w,\mathbf{R}^w)$ and for P_D = Max(P_I, P_S) it results $P_D \geq |\mathcal{E}|^{-1/2}$. P_D is the probability of deception.

4.2.1 Security performances analysis of permuted turbo codes

In the analysis of the permuted turbo codes security performances, the attention will be focused on the computation of $|\mathcal{E}|$ whose value determines the bounds on substitution, deception and impersonation probabilities already discussed in the previous paragraph.

At this aim it can be observed that purely random interleavers may perform rather poorly when the block-size is small. To lower the BER many different interleaver schemes have been proposed in literature. However, an extensive analysis of the cardinality of each subclass would be a to an heavy task. Here, the subset of semi-random interleavers, often referred in literature as *S*-random interlevears, [33], will be analyzed, from this popint of view. *S*-random interleaver exhibit rather effective performance even at small block-size and present an acceptable computational complexity.

In essence, in a semi-random interleaver, characterized by a minimum distance equal to *S*, the *n*-th permutation index is computed by randomly generating an integer i_n. If $|i_n - i_{n-k}| > S$, for

$k = 1, 2, \cdots, S$, then i_n is retained, otherwise it is rejected and a new integer generated. Fully random interleavers are a subset of the semi-random interleavers corresponding to $S = 1$.

Let us denote with $\mathcal{E}_\pi^{(N,S)}$ the set of S-random interleavers with constraint-length N, with $\mathcal{E}_c^{(N,v)}$ the set of RSC encoder with rate $1/v$ and constraint-length N, and with $\mathcal{E}_p^{(N,\rho)}$ the set of puncturing schemes with constraint-length N and number of survivor equal to ρ. For a permuted turbocode that employs the S-random interleavers we have that $e_\pi \in \mathcal{E}_\pi = \mathcal{E}_\pi^{(N,1)}$, $\Pi_1 \in \mathcal{E}_\pi^{(N,S_1)}$, $\Pi_2 \in \mathcal{E}_\pi^{(N,S_2)}$, $C_1 \in \mathcal{E}_c^{(N,v_1)}$, $C_2 \in \mathcal{E}_c^{(N,v_2)}$, $P_1 \in \mathcal{E}_p^{(N,\rho_1)}$, $P_2 \in \mathcal{E}_p^{(N,\rho_2)}$. Therefore, the cardinality of the set $\mathcal{E} = \mathcal{E}_\pi \times \mathcal{E}_s$ can be expressed as follows:

$$|\mathcal{E}| = |\mathcal{E}_\pi||\mathcal{E}_s| = |\mathcal{E}_\pi|\left|\mathcal{E}_\pi^{(N,S_1)}\right|\left|\mathcal{E}_\pi^{(N,S_2)}\right|\left|\mathcal{E}_c^{(N,v_1)}\right|\left|\mathcal{E}_c^{(N,v_2)}\right|\left|\mathcal{E}_p^{(N,\rho_1)}\right|\left|\mathcal{E}_p^{(N,\rho_2)}\right| \qquad (210).$$

Incidentally it can be observed that the overall turbo-code rate R_{tc} is equal to $R_{tc} = 1 \Big/ \left[1 + v_1 + v_2 - (\rho_1 + \rho_2)/N\right]$.

An S-random interleaver is characterized by a minimum distance equal to S, since the n-th permutation index is computed by randomly generating an integer i_n. If $|i_n - i_{n-k}| > S$, for $k = 1, 2, \cdots, S$, then i_n is retained, otherwise it is rejected and a new integer generated. Fully random interleavers are a subset of S-random interleavers corresponding to $S = 1$.

In [34] a more general definition of S-random interleaver is provided: an S-random interleaver guarantees that, if two inputs bit are within a distance S_1 they cannot be mapped to distance less than S_2 apart at the output. Usually $S_1 = S_2 = S$ is chosen. Thus, considering two indexes i and j, such that $0 \le |i - j| \le S_1$, the design imposes that $|\Pi(i) - \Pi(j)| > S_2$. Usually $S_1 = S_2 = S$ is chosen.

S-random interleavers with $S_1 = S_2 = S$ will be named as “full-S-random interleavers” and S-random interleavers with $S_1 = 1$ and $S_2 = S$ will be denoted by “half-S-random interleavers”..

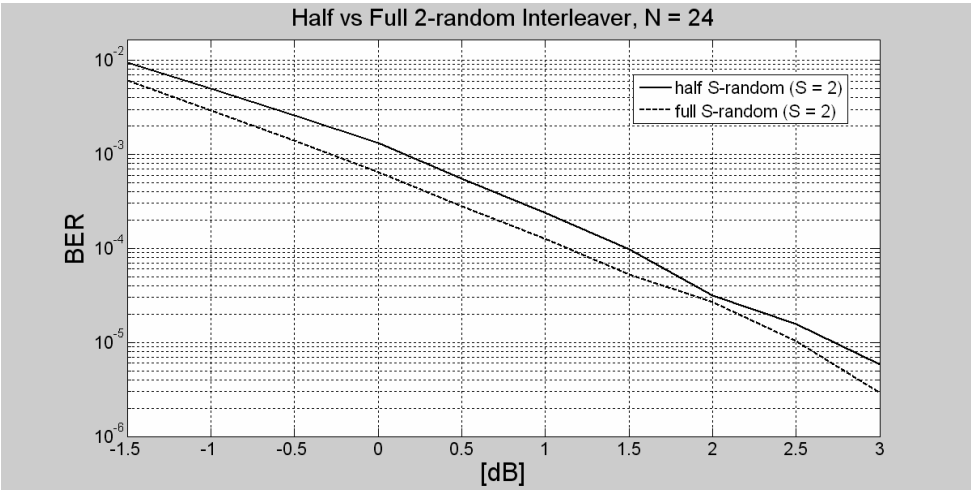

Fig.32 half-*S*-random vs full *S*-random performances (*N* = 24, *S* = 2, *v* = 5).

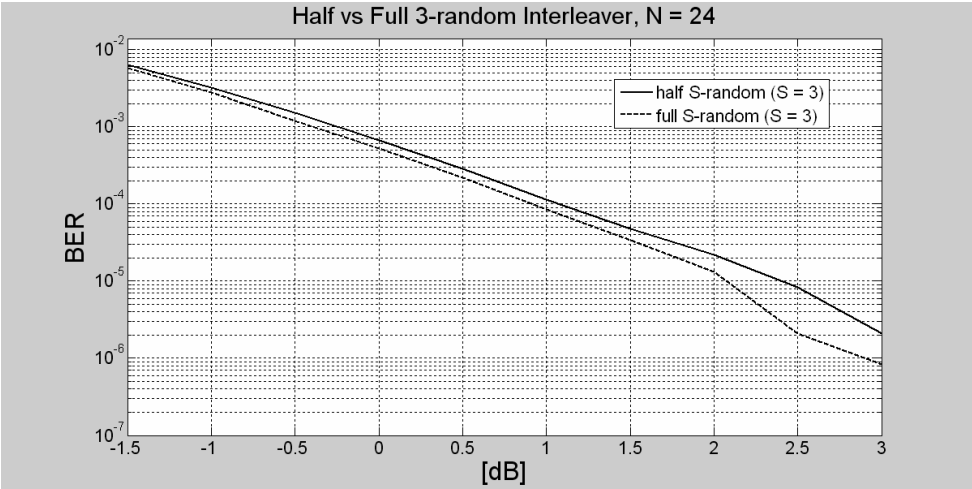

Fig.33 half-*S*-random vs full *S*-random performances (*N* = 24, *S* = 3, *v* = 5).

While half-*S*-random interleavers formally represents a particular case of full-*S*-randoms, the set of full-*S*-randoms is actually a subset of half-*S*-randoms. This means that the number of keys to address a half-*S*-random will be superior to the number of keys needed for a full-*S*-random (with the same constraint-length, obviously). Thus half-*S*-randoms authentication interleavers are more securing than full-*S*-randoms. The drawback is that full-*S*-randoms should lead to better performance, in terms of B.E.R.. In Fig.32 the performances of a half-2-random interleaver are compared with the performances of a full-2-random, for a constraint-length N = 24. In Fig.33 the performances of a half-3-random interleaver are instead

compared with the performances of a full-3-random, being N = 24. Both pictures refer to a 1/5-rated turbo codes. Fig.32 and 33 show that the coding gain from half to full S-random is not so impressive.

In order to evaluate the terms $\left|\mathcal{E}_{\pi}^{(N,S_1)}\right|$ and $\left|\mathcal{E}_{\pi}^{(N,S_2)}\right|$ of equation (84), an approximated method to compute the cardinality of both half and full S-random interleavers is thus proposed. It will also proved that the intrinsic approximation is absolutely acceptable for the proposed system configuration.

In the following, $\left|\mathcal{E}_{\pi}^{(N,S)}\right|$ will denote the cardinality of the set of full S-random interleaver with constraint length N, and $\left|\mathcal{E}_{\pi_{HALF}}^{(N,S_1)}\right|$ the cardinality of the set of half S-random interleaver with the same constraint length.

As introduced before, an half S-random interleaver implies that being two inputs within a distance 1 they must be mapped to distance more than S apart at the output. Moreover a full S-random interleaver involves that being two inputs within a distance S they have to be mapped to distance more than S apart the output.

The general generating process for a full-S-random (the half S-random is a special case of this), will now briefly recalled, in order to better understand the proposed cardinality estimation methods. Let A = {1, 2, ···, N} be the indexes alphabet. Denoting by $A^{(j)}$ the survivor set available before the j-th choice is performed we can say that $A^{(1)} = A$ since no one elements was selected before the first step. Let $n_s^{(j)}$ be the number of possible selection that can be performed at the j-th step, while i_j is the survivor chosen at step number j. At the first step it obviously results $n_s^{(1)} = |A^{(1)}| = N$. At the second step, it results:

$$n_s^{(2)} = \left|\left\{i_2 \in A^{(2)} / \left|i_2 - i_1\right| > S\right\}\right|, \quad A^{(2)} = A - \left\{i_1\right\} \qquad (211).$$

At the third step:

$$n_s^{(3)} = \left|\left\{i_3 \in A^{(3)} / \left|i_3 - i_2\right| > S \cap \left|i_3 - i_1\right| > S\right\}\right|, \quad A^{(3)} = A - \left\{i_1, i_2\right\} \qquad (212),$$

and so on, until the step number S + 1, when:

$$n_s^{(S+1)} = \left|\left\{ i_{S+1} \in A^{(S+1)} / \left|i_{S+1} - i_S\right| > S \cap \left|i_{S+1} - i_{S-1}\right| > S \cap \cdots \left|i_{S+1} - i_1\right| > S \right\}\right|, \quad A^{(S+1)} = A - \left\{i_1, i_2, \cdots, i_S\right\} \qquad (213).$$

From the S+2-th step the number of previous choices (equal to how much the survivor alphabet has been reduced) is bigger than the number of conditions (S) that must be satisfied by i_j, i.e.:

$$n_s^{(S+2)} = \left|\left\{ i_{S+2} \in A^{(S+2)} / \left|i_{S+2} - i_{S+1}\right| > S \cap \left|i_{S+2} - i_S\right| > S \cap \cdots \left|i_{S+1} - i_2\right| > S \right\}\right|, \quad A^{(S+2)} = A - \left\{i_1, i_2, \cdots, i_{S+1}\right\} \qquad (214).$$

Thus, for every j-th step, with $j > S$, it results:

$$n_s^{(j)} = \left|\left\{ i_j \in A^{(j)} / \left|i_j - i_{j-1}\right| > S \cap \left|i_j - i_{j-2}\right| > S \cap \cdots \left|i_j - i_{j-S}\right| > S \right\}\right|, \quad A^{(j)} = A - \left\{i_1, i_2, \cdots, i_{j-1}\right\} \qquad (215).$$

To evaluate $\left|\mathcal{E}_\pi^{(N,S)}\right|$ the approach

$$\left|\mathcal{E}_\pi^{(N,S)}\right| = \prod_{j=1}^{N} \overline{n}_s^{(j)} \qquad (216)$$

can be used, where $\overline{n}_s^{(j)}$ is the expected value of $n_s^{(j)}$ respect to all the possible S-tuples $\left\{i_{j-1}, i_{j-2}, \cdots, i_{j-S}\right\}$, and all the possible survivors sets $A^{(j)} = A - \left\{i_1, i_2, \cdots, i_{j-1}\right\}$. This method is quite unfeasible, since it represents an extreme time-consuming task. For this reason, instead of evaluating

$$\overline{n}_s^{(j)} = E_{A^{(j)}, \left\{i_{j-1}, i_{j-2}, \cdots, i_{j-S}\right\}} \left\{\left|\left\{ i_j \in A^{(j)} / \left|i_j - i_{j-1}\right| > S \cap \left|i_j - i_{j-2}\right| > S \cap \cdots \left|i_j - i_{j-S}\right| > S \right\}\right|\right\} \qquad (217),$$

it is preferable to compute

$$\overline{n}_s^{(j)} \approx E_{\left\{i_{j-1}, i_{j-2}, \cdots, i_{j-S}\right\}} \left\{\left|\left\{ i_j \in A / \left|i_j - i_{j-1}\right| > S \cap \left|i_j - i_{j-2}\right| > S \cap \cdots \left|i_j - i_{j-S}\right| > S \right\}\right|\right\} - (j-1) \qquad (218).$$

In other words instead of averaging $n_s^{(j)}$ respect to all the possible survivors sets $A^{(j)}$ and to all the possible *S*-tuples, $n_s^{(j)}$ can be averaged respect to all the possible *S*-tuples, referring to a survivor set equals to the whole alphabet A. The reduction of the number of possible selection is taken into account by the terms $-(j-1)$, (instead of using $A^{(j)}$). The evaluation of the terms $E_{\{i_{j-1},i_{j-2},\cdots,i_{j-S}\}}\left\{n_s^{(j)}\right\}$ can be easily implemented in a fast (and low-resources consuming) PC program. This terms can be expressed in a closed analytic form, when dealing with half *S*-random interleavers.

At the first selection step of an half-*S*-random interleaver with constraint length *N*, we have $n_{s_{HALF}}^{(1)} = \left|A^{(1)}\right| = N$. For the *j*-th step, with *j* >1, it instead results:

$$n_{s_{HALF}}^{(j)} = \left|\left\{i_j \in A^{(j)} / \left|i_j - i_{j-1}\right| > S\right\}\right|, \quad A^{(j)} = A - \left\{i_1, i_2, \cdots, i_{j-1}\right\} \qquad (219).$$

It can be still used the approach

$$\left|\mathcal{E}_{\pi_{HALF}}^{(N,S)}\right| = \prod_{j=1}^{N} \overline{n}_{s_{HALF}}^{(j)} \qquad (220),$$

where

$$\overline{n}_{s_{HALF}}^{(j)} \approx E_{i_{j-1}}\left\{\left|\left\{i_j \in A / \left|i_j - i_{j-1}\right| > S\right\}\right|\right\} - (j-1) \qquad (221).$$

For this type of interleaver, that represents a special case of full *S*-randoms, it is possible to express the terms $\left|\left\{i_j \in A / \left|i_j - i_{j-1}\right| > S\right\}\right|$ in a closed form:

$$\left|\left\{i_j \in \mathrm{A} / \left|i_j - i_{j-1}\right| > S\right\}\right| = \begin{cases} N - S - \left(i_{j-1} - 1\right) + 1, & 1 \le i_{j-1} \le S \\ N - 2(S-1), & S+1 \le i_{j-1} \le N - S \\ i_{j-1} - S + 1, & N - S - 1 \le i_{j-1} \le N \end{cases} \qquad (222).$$

Thus

$$E_{i_{j-1}}\left\{\left|\left\{i_j \in A / \left|i_j - i_{j-1}\right| > S\right\}\right|\right\} = \frac{1}{N}\left\{\sum_{i_{j-1}=1}^{S}\left[N - S - (i_{j-1} - 1) + 1\right] + \right.$$
$$\left. + \sum_{i_{j-1}=S+1}^{N-S}\left(N - 2S + 2\right) + \sum_{i_{j-1}=N-S+1}^{N}\left(i_{j-1} - S + 1\right)\right\} \quad (223).$$

Using equation (97) in (95) the analytic expression of $\overline{n}_{s_{HALF}}^{(j)}$ is obtained. The last that can be substituted into equation (94). It is easy to prove that the cardinality of pure-random permutations is equivalent to $\left|\mathcal{E}_{\pi_{HALF}}^{(N,1)}\right| = \overline{n}_{s_{HALF}}^{(1)} \prod_{j=2}^{N} \overline{n}_{s_{HALF}}^{(j)} = N \prod_{j=2}^{N} \overline{n}_{s_{HALF}}^{(j)} = N \prod_{j=2}^{N} \left(N - j + 1\right) = N!$.

When selecting a S-random interleaver for performances evaluation by Monte Carlo simulations one must be very careful in the choice of the permutation. Indeed, as an instance referring to full S-random interleaver, it must be considered that, for example, a S-random interleaver with constraint length N (remembering that $\sqrt{N/2}$ must be superior to S), could be, in some cases also, a (S+1)-random interleaver even a (S+2)-random, and so on. This especially for large N, could happen since the condition $\left|i_j - i_{j-1}\right| > S \cap \left|i_j - i_{j-2}\right| > S \cap \cdots \left|i_j - i_{j-S}\right| > S$ does not exclude that the condition $\left|i_j - i_{j-1}\right| > S+1 \cap \left|i_j - i_{j-2}\right| > S+1 \cap \cdots \left|i_j - i_{j-S-1}\right| > S+1$ or in some rare case the condition $\left|i_j - i_{j-1}\right| > S+2 \cap \left|i_j - i_{j-2}\right| > S+2 \cap \cdots \left|i_j - i_{j-S-2}\right| > S+2$ could be satisfied too. For this reason the generated interleaver must be an S-random, but no more. Without considering this aspect, one can generate a 2-random interleaver (enough "good" to be also a 4-random interleaver) and pure 3-random interleaver. What it would happen is that the one referred as a 2-random performs better than the 3-random, since the 2-random satisfied the conditions towards $S = 4$, and so is actually a 4-random.

To verify the robustness of the approximations described by equations (92) and (95) (respectively for the case of full and half S-randoms) four interleavers have been carefully designed, (taking into account all the considerations just made): a full 2-random, an half 2-random, a full 3-random and an half 3-random interleaver, all characterized by a constraint

length N = 24, since it represents the number of data bits of the turbo codeword used in system simulations. A PC programs has been written to evaluate after every choice, the number of possible selection that can be made, i.e. $n_s^{(j)}$, for $j > 2$. As an instance, the adopted full 2-random interleaver, with constraint length $N = 24$, is:

{2, 20, 16, 4, 21, 17, 10, 5, 23, 11, 19, 7, 13, 3, 8, 14, 18, 9, 22, 1, 15, 12, 24, 6} (224).

To evaluate $n_s^{(2)}$ it must be considered that $A^{(2)} = A - \{2\}$, where $A = \{1, 2, \cdots, 24\}$. At the same time $n_s^{(2)}$ is given by the number of all the elements i_2 of $A^{(2)}$ that satisfies the condition $|i_2 - 2| > 2$. The set of elements of $A^{(2)}$ that satisfies the condition is {5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24} and its cardinality is $n_s^{(2)} = 20$. About $n_s^{(3)}$ we have $A^{(3)} = A - \{2, 20\}$. At the same time $n_s^{(3)}$ is given by the number of all the elements i_3 of $A^{(3)}$ that satisfies the condition $|i_3 - 20| > 2 \cap |i_3 - 2| > 2$. The set of elements of $A^{(3)}$ that satisfies the condition is {5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 23, 24}, its cardinality is $n_s^{(3)} = 15$, and so on. To confirm the validity of the approximations described in (92) and (95), some comparison were made between the terms $n_s^{(j)}$, evaluated for a specific interleaving like the one just discussed, and the approximated value of $\bar{n}_s^{(j)}$ obtained through the (92) and (95). The program implemented for the full S-random, for a fixed j, works in this way: 1) generate all the possible S-tuple $\{i_{j-1}, i_{j-2}, \cdots, i_{j-S}\}$; 2) for every one of this, it counts all the possible choices (in a way similar to the one before described), using the complete alphabet A as survivors set; 3) add $1 - j$ to the result of stage 2).

The adopted half 2-random interleaver is:

{1, 4, 7, 3, 6, 9, 5, 10, 13, 16, 11, 2, 12, 8, 14, 17, 22, 18, 21, 24, 19, 23, 15, 20} (225).

The chosen full 3-random interleaver is instead:

{12, 7, 2, 21, 17, 8, 13, 22, 3, 9, 15, 24, 5, 19, 11, 23, 4, 16, 10, 20, 1, 14, 6, 18} (226).

Finally, the selected half 3-random interleaver is:

{13, 5, 9, 20, 12, 3, 7, 22, 11, 6, 10, 14, 18, 1, 17, 21, 16, 4, 15, 24, 19, 23, 8, 2} (227).

The comparison between the exact values of $n_s^{(j)}$ and the approximation of $\bar{n}_s^{(j)}$ is displayed in Figs. 34, 35, 36 and 37, that respectively refer to the full 2-random, half 2-random, full 3-

random and half 3-random interleaving. As it can be noted by observing Fig. 34 and 36, in the case of full *S*-random interleavers the use of the approximated formula averagely leads to a slightly lower number of possible selection at every single step, and so to a slightly lower cardinality of the interleaving set.

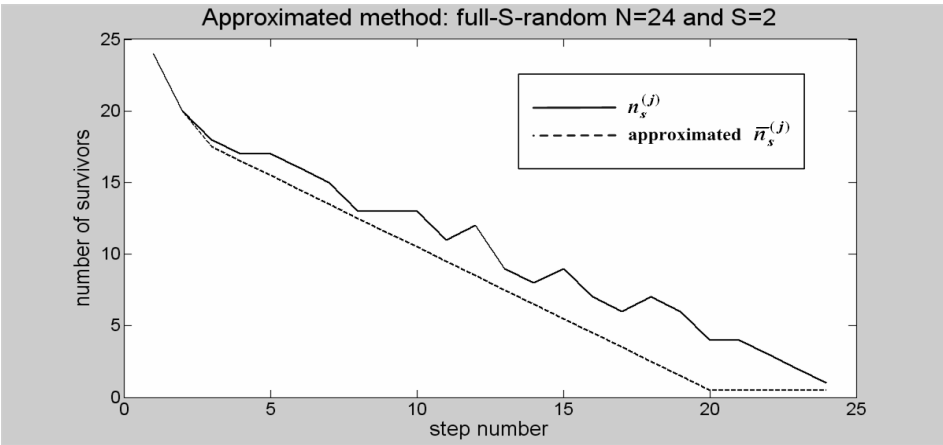

Fig. 34 comparison between the terms $n_s^{(j)}$ and $\approx \overline{n}_s^{(j)}$ (case full 2-random)

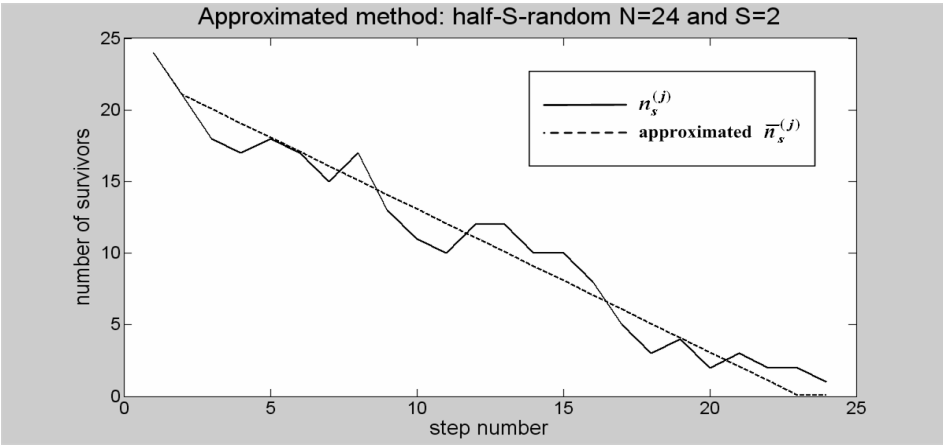

Fig. 35 comparison between the terms $n_s^{(j)}$ and $\approx \overline{n}_s^{(j)}$ (case half 2-random)

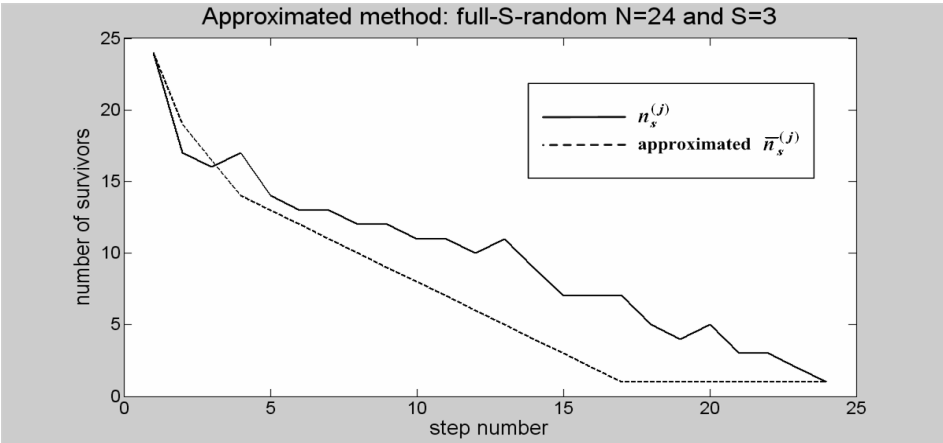

Fig. 36 comparison between the terms $n_s^{(j)}$ and $\approx \bar{n}_s^{(j)}$ (case full 3-random)

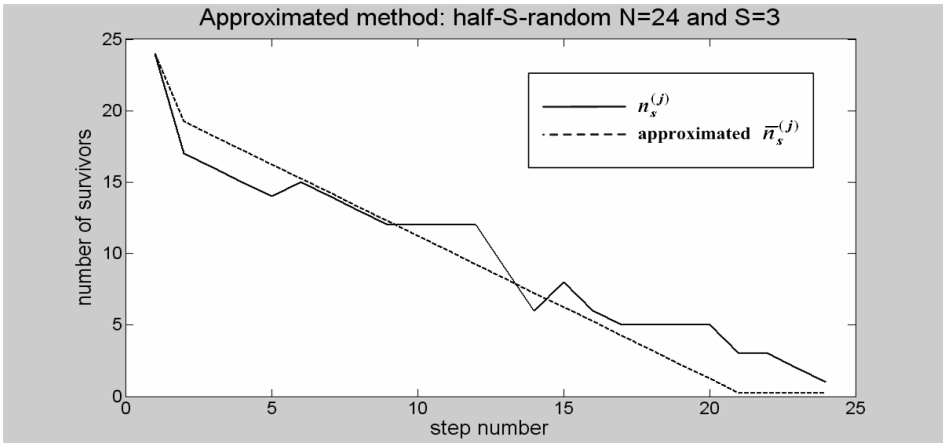

Fig. 37 comparison between the terms $n_s^{(j)}$ and $\approx \bar{n}_s^{(j)}$ (case half 3-random)

This does not represents a problem since the method seems to work better when dealing with half *S*-randoms, that represents the more useful type of *S*-random interleaving for authentication purposes. However, the terms $\left|\mathcal{E}_\pi^{(N,S_1)}\right|$ and $\left|\mathcal{E}_\pi^{(N,S_2)}\right|$ of equation (84) can be expressed trhough equations (90), (92) when dealing with full S-random interleaving, and by (94), (95), (96) if the interleaver is an half *S*-random.

At this point the terms $\left|\mathcal{E}_c^{(N,v_1)}\right|$ and $\left|\mathcal{E}_c^{(N,v_2)}\right|$ of equation (84) must be evaluated.

The number of shift-register related to the convolutional encoder is $v+1$. The cardinality of the set of shift-registers is instead $J = \sum_{j=0}^{N/\log_2(l_g)-1} (l_g - 1) l_g^j$, ($l_g = 8$ for the octal representation). For this reason, the number of possible RSC encoders on the single branch is:

$$\left|\mathcal{E}_c^{(N,v)}\right| = \left|\mathcal{E}_{c_{RSC}}^{(N,v)}\right| = \binom{J}{v+1} = \binom{\sum_{j=0}^{N/\log_2(l_g)-1} (l_g - 1) l_g^j}{v+1} \qquad (228).$$

It must be observed that the number of shift-register in the second and third branch is $v+1$, because also the feed-back register is considered. It is clear that the number of possible NSC (Non Systematic Convolutional) encoders would simply be $\left|\mathcal{E}_{c_{NSC}}^{(N,v)}\right| = \binom{J}{v}$.

Finally, the number of possible puncturing schemes is given by:

$$\left|\mathcal{E}_p^{(N,\rho)}\right| = \binom{N}{N-\rho} \qquad (229).$$

As an example, in Table 1 the values of $\log_2\left|\mathcal{E}_\pi^{(N,S)}\right|$ versus S for block-size N equal to 24 are reported, referring to both half and full generating algorithms. The maximum value of S is determined by the fact that the S-random algorithm tends to diverge when S is close to $\sqrt{N/2}$, [33].

	Half-S-random	Full-S-random
$S=1$	80	80
$S=2$	64	55
$S=3$	55	50

Tab.1 $\log_2\left|\mathcal{E}_\pi^{(24,S)}\right|$ versus S

Concerning the BER it can be observed that, in [3] and [35] an upper bound for this quantity in terms of error coefficients D_m, and the minimum hamming code distance $d_{\min}$ has been formulated. However, the exact evaluation of both these quantities is computationally hard, especially when the constraint-length N is high. In [35] an accurate approximation for the D_m has been proposed, but it is valid only in the hypothesis of uniform interleaver ($S = 1$), while a method to evaluate turbo-code $d_{\min}$ is presented in [36]. In [37] the performance of turbo code have been evaluated using the transfer function approach, but this derivation includes averaging with respect to every possible interleaver and so it is not worthy for our purposes. In [38] the free-distance is estimated, but this result is not useful when dealing with radio-packets network. For these reasons to estimate the performance of S-random Turbo Code we resorted to a Monte Carlo approach.

4.3 System simulation

To verify the effectiveness of the proposed scheme in practical applications, an extensive simulation of an OFDM digital communication system over AWGN channels making use of the proposed joint authentication and Forward Error Correction based on turbo codes has been carried out.

In particular we considered OFDM signals with $N_c = 1024$ carriers, a turbo code constraint length $N = 24$ bits and a false alarm probability $P_{fa} = 10^{-3}$ have been employed in the simulations reported hereafter.

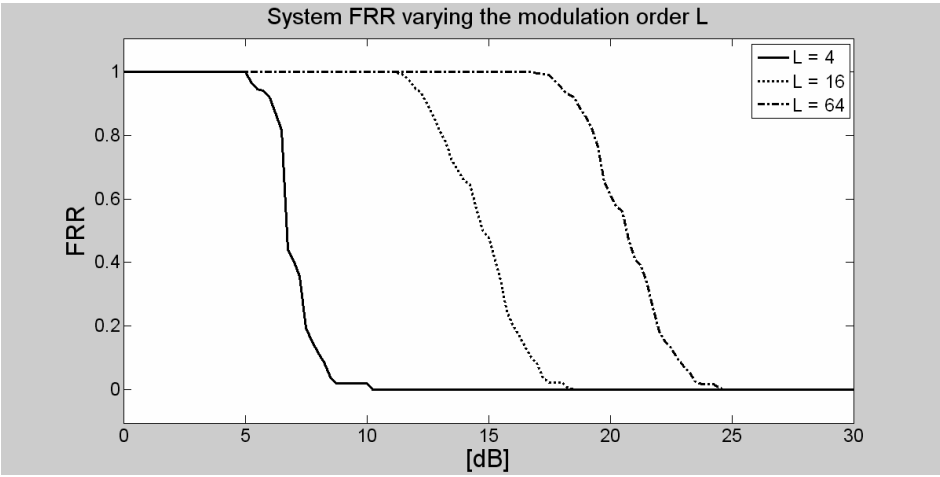

Fig.38 system FRR versus the modulation order

Fig.38 shows the impact of the number of constellation points L of the modulation format of each OFDM sub-carrier on the probability of rejecting an authentic message also referred in literature as False Rejection Rate (FRR). More specifically QPSK (L = 4), 16-QAM (L = 16) and 64-QAM (L = 64) modulations have been investigated. As expected, the FRR increases with L. As it has been said before, at every received hash/codeword the user is successfully recognized only if the LLR defined in (57) is higher than the adaptive threshold λ, that can be derived by numerical inversion of equation (60). In other words the user is authenticated if:

$$\log \Lambda\left(\hat{\boldsymbol{s}}; \boldsymbol{x}, \boldsymbol{r}_{1,} \boldsymbol{r}_2\right) > \lambda \qquad (230).$$

In Fig.39 a comparison between $\log \Lambda\left(\hat{\boldsymbol{s}}; \boldsymbol{x}, \boldsymbol{r}_{1,} \boldsymbol{r}_2\right)$ and λ is displayed. The curves have been obtained in this way. The first n_{frame} points refer to the LLRs evaluated on n_{frame} different codewords at the same SNR = 3dB. The second n_{frame} points refer to the LLRs computed on n_{frame} different codewords at the same SNR = 3.25dB. The k-th n_{frame} points refer to the LLRs evaluated on n_{frame} different codewords at the same SNR = [(k – 1)*0.25 + 3]dB, and so on until SNR = 15dB.

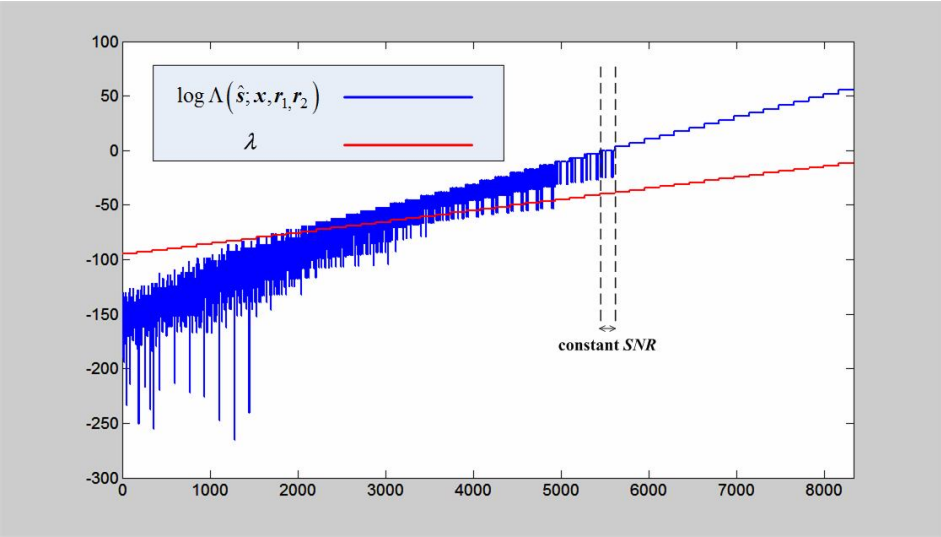

Fig.39 comparison between $\log\Lambda(\hat{s};x,r_1,r_2)$ and λ

As it can be noted observing the picture, while, on every n_{frame}-length *SNR*-constant block the LLR can vary while the threshold λ remains fixed. This is due to the fact that λ depends (60) only on the channel conditions (i.e. on σ_N^2 and so on the *SNR*) and on the false-alarm probability P_{fa}. The LLR varies because at every frame the noise profile is changed by the simulation program. It is clear that when the SNR is very high (right-part of the plot), it happens that being the differences $\boldsymbol{x}-\mu(\hat{\boldsymbol{s}}_\Pi)$, $\boldsymbol{r}_1-\mu(\hat{\boldsymbol{z}}_1)$ and $\boldsymbol{r}_2-\mu(\hat{\boldsymbol{z}}_2)$ very little it is hard to find their variations on the graph.

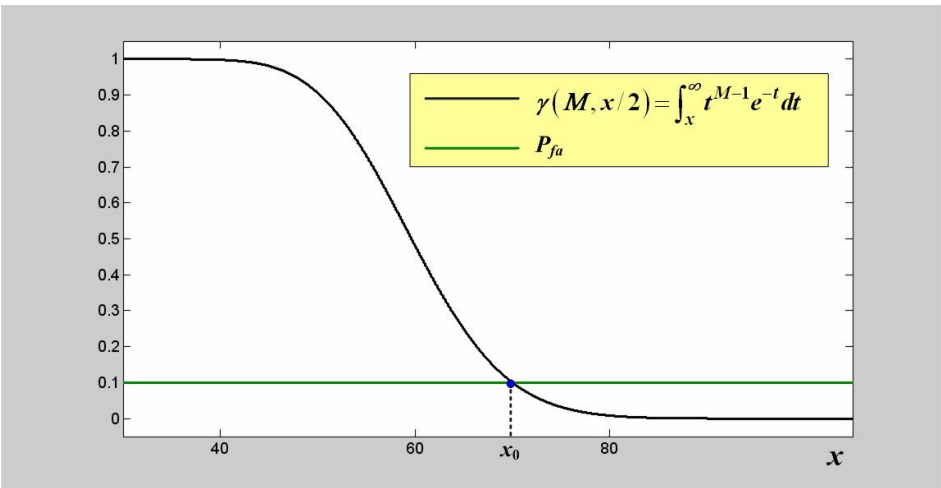

Fig.40 comparison between $\gamma(M, x/2)$ and P_{fa}.

In Fig.40 a comparison between the upper incomplete gamma function $\gamma(M, x/2) = \int_x^{\infty} t^{M-1} e^{-t} dt$ and the desired false alarm probability P_{fa} is instead depicted. Equation (60) can be solved in a more simple way: after have found $x = x_0$ that satisfies:

$$P_{fa} = \frac{\gamma(M, x/2)}{(M-1)!} \qquad (231).$$

Threshold λ can be found using equation $x_0 = -2\lambda - 2M \log\left(2\pi\sigma_N^2\right)$, where $M = vN \ / \log_2(L)$ is the number of symbol carried on a turbo codeword, and where σ_N^2 is the additive white gaussian noise variance equal to $10^{-SNR_{\mathrm{dB}}/10} P_s$ (where SNR_{dB} is the SNR in dB and P_s is the useful signal power). The variable x_0 depends only on the false-alarm probability P_{fa} and M, while λ varies with the values of M and σ_N^2. In Fig.40 P_{fa} = 0.1, N = 24, v = 5 and L = 4 (QPSK) $\Rightarrow$ M = 60. In these conditions x_0 = 70.1170 (see Fig. 40).

Some of the parametric (respect to M) curves $\lambda\left(\sigma_N^2, M\right) = \frac{x_0\left(M, P_{fa}\right)}{2} - M \log\left(2\pi\sigma_N^2\right)$ are shown in Fig.41 for P_{fa} = 0.1. In particular the three cases v = 3, 5 and 7 that with N = 24 and L = 4 imply M = 36, 60, 84 are displayed, since are of interest for the following FRR (False Rejection Rate) Monte Carlo evaluations.

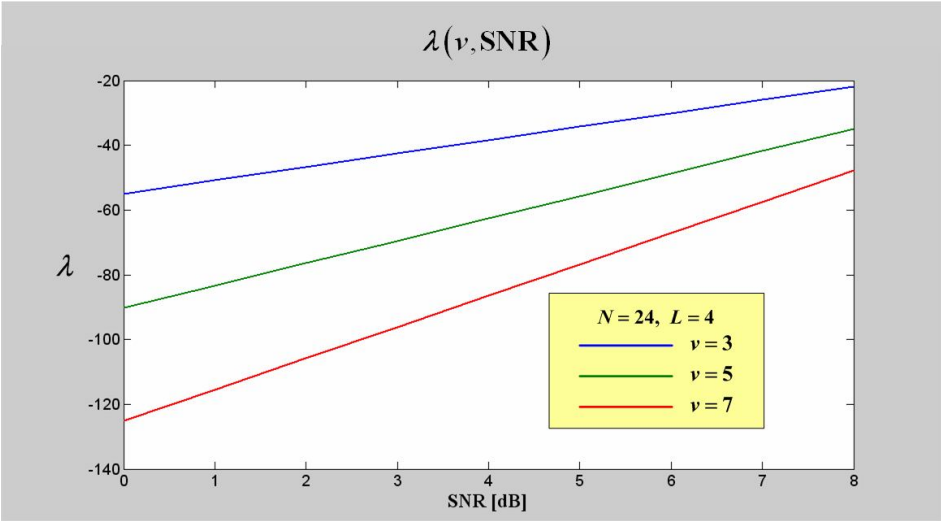

Fig.41 $\lambda(v$, SNR)

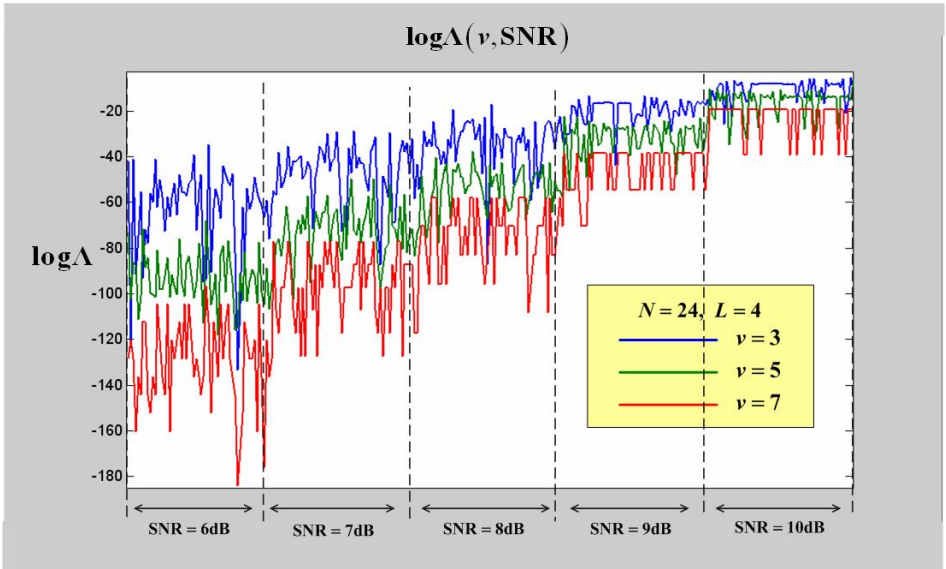

Fig.42 logΛ(v, SNR)

As it can be observed in Fig.42, although the correcting capability of the turbo code increases with v, the LLRs decreases with v at the same way in which λ decreases with v. Thus one must expect to see no FRR variations depending on the value of v. (The LLR's curves of Fig.41 are obtained in the same way of Fig.39).

More specific for Fig.43, referring to a turbo-code with v = 5 and interleaver minimum distance S = 1, increasing L by a factor 4 produces an equivalent *SNR* loss of about 8 dB. This

behavior can be explained observing that an increase in L produces an increment in the BER at the output of the OFDM demodulator without FEC equivalent to an SNR loss equal to $\log_2(L) / (L - 1)$. In the meanwhile, when the signal bandwidth and subcarrier spacing is kept constant, the size of the packet to be authenticated increases by a factor $\log_2(L)$.

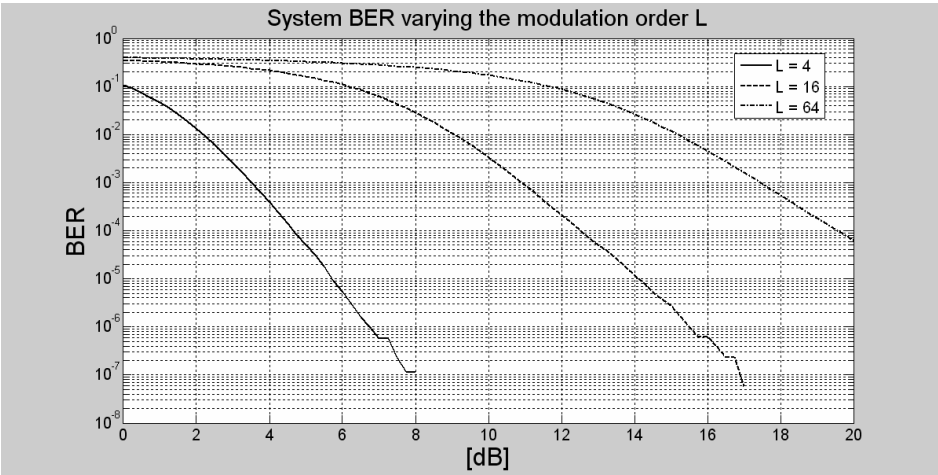

Fig.43 system BER versus the QAM modulation order

Fig.44 depicts the impact of the redundancy, controlled by the turbo-code rate $R_{tc} = 1/v$, on the FRR, for the QPSK constellation. In these simulations S is still 1.

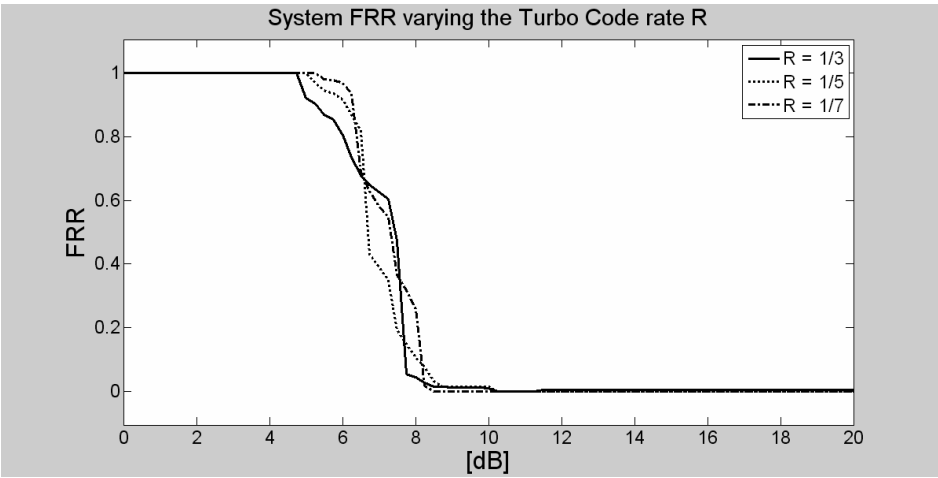

Fig.44 system FRR versus turbo-code rate for QPSK modulation

Increasing the redundancy improves, as it is shown in Fig.45, the BER. performances, but keeps the FRR almost constant since increasing the redundancy means increasing the length

of the hash to be verified, and so the probability that its LLR is not big enough to pass the threshold test (see equation (8)). More in detail, as it is explained by Figs. 41 and 42, PC simulation show how increasing v involves decreasing (for an assigned SNR) both the threshold λ and the LLRs $\log\Lambda$.

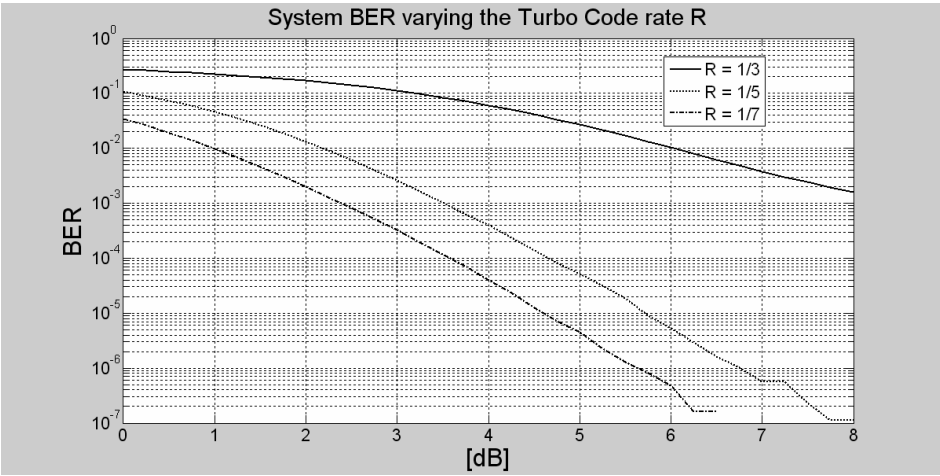

Fig.45 system BER versus turbo-code rate for QPSK modulation

Fig.46 illustrates the impact on the FRR of the minimum distance S of the S-random interleaver. In this set of simulations the total number of sub-carriers N_c has been set to 1024; a turbo-code with a rate R_{tc} equal to 1/5 has been employed. QPSK constellation are adopted on every sub-carrier.

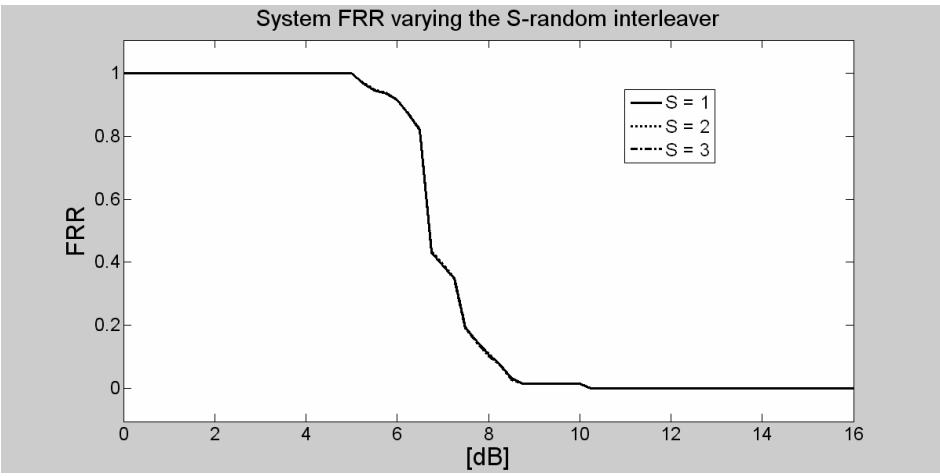

Fig.46 system FRR varying the *S*-random interleaver

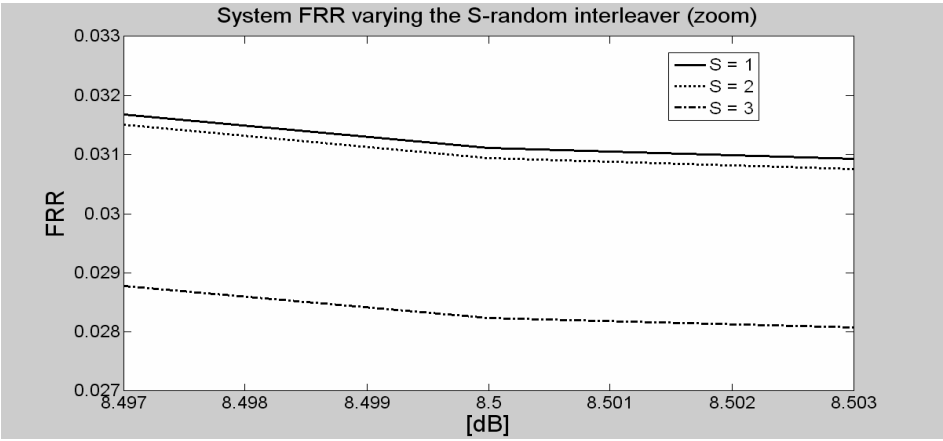

Fig.47 system FRR varying the *S*-random interleaver (zoom)

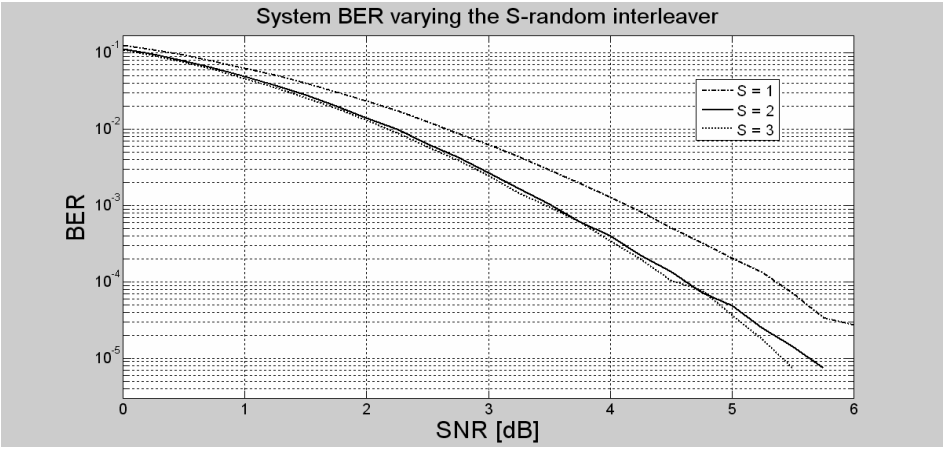

Fig.48 system BER varying the *S*-random interleaver

The behavior of the FRR versus *S* does not reflects the behavior of the BER versus the same quantity illustrated by Fig.45. Probably the coding gain using half-*S*-random is not sufficiently high in order to make a partially uncorrect hash completely correct. Actually a slight performance improvement, from the FRR point of view, is provided, like it is shown by the “zoom” displayed in Fig.47, but it is quite neglectable.

4.4 Conclusions

Simulation results have demonstrated the feasibility of joint authentication, integrity verification and channel coding, and forward error correction based on the Neyman-Pearson procedure.

As in classical turbo code applications, the posterior probability of the decoded message and hash pair, given the received noisy signal, gives a rich information about the reliability of the authentication test result and of the restored message. Moreover, it can be at the basis of new Hybrid ARQ schemes integrating authentication and integrity verification with error correction.

The great number of different A-FEC turbo codes adopting an *S*-random interleaver that can be constructed even for small block sizes guarantees a low deception probability. Moreover, an unconditionally secure architecture with respect to targeted substitution attacks can be obtained by inserting an additional random interleaver before the turbo encoder.

5 TH CSMA Persistent Systems

The security algorithm proposed in this chapter is based both on MAC and physical layer authentication mechanisms. It refers to a CSMA (Carrier Sense Multiple Access) system where the *p*-persistent protocol is supposed to be overlapped to a particular TDMA (Time Division Multiple Access) technique based on TH (Time Hopping) procedures.

The possibility of improving security by slightly modifying the MAC protocol without degrading throughput is here investigated.

The scenario is a wireless computer or electronics devices network equipped with an AP (Access Point) with decentralised medium access control. To detect unauthorized users trying to access the network thanks to the knowledge of some higher-layer security data, to every authenticated user a different time hopping sequence, defining which slot subset he/she can use to randomly access the medium, is assigned. The AP should periodically check the hopping sequences utilized by the incoming communications. Moreover, the usage of different hopping sequences allows different simultaneous transmissions. Once the sequence of the *j*-th connection is found to be incorrect, the *j*-th link is immediately dropped. This means that an attack to the network will last, in the worst case, not more than the hopping sequences check period. The throughput formula provided in [39] is properly extended to the general case in which N_c hopping sequences are used at the same time. Then, the presence of TH sequences check periods (supposed to last X_{elab} time-slots) are taken into account. Finally the effectiveness of the TH monitoring in presence of denial of service attacks is evaluated.

5.1 TH-Slotted-CSMA-*p*-Persistent model

A number of studies on throughput analysis for random access protocols have already appeared in the literature. However, most of them have been based in the assumption that there are infinitely many users, so that the collective channel traffic can be modelled by a

Poisson process with a finite rate ([40] and [41]). In [42] an analysis of throughput for slotted and unslotted persistent CSMA with collision detection is provided.

The number of users is here assumed finite. In this way this study can give useful information for ground packet radio systems as well as wireless local area computer network, which consist of relatively small number of users. At this aim the results provided in [39] will be extended.

Let assume that each user has periods, which are independent and geometrically distributed, in which he has no packets to transmit.. By superimposing this idle periods over all users, the system idle periods in which no user has a packet (denoted by I) is easily seen also to be geometrically distributed. The case of infinite population does not need a specific assumption on the distribution of each user's idle period because the Palm-Khinchine theorem (see [43]) guarantees that the collective idle period is always independent and exponentially or geometrically distributed.

Due to the above assumption, it results that each epoch in the system idle period is a regenerative point, in the sense that the system state after any such epoch is a probabilistic replica of the system state beginning at the previous such epoch. Thus the system state alternates between idle periods I and busy periods B in which at least one user has a packet. A consecutive pair B and I can be named a *regeneration cycle*. Then, the channel throughput S is generally expressed by [39]:

$$S = \frac{\overline{U}}{\overline{B} + \overline{I}} \qquad (232),$$

where U is the time spent, during a regeneration cycle, in effective data transmission, and where $\overline{\chi}$ stand for the expected value of χ. Following [39], in slotted CSMA systems the time is slotted with slot size $T = a$, being a the maximum network propagation delay. All users are synchronized to start transmission only at slot boundaries (for convenience, let $1/a$ be an integer). An attempted transmission is successful if none of the other users start transmission at the same time. For the next 1/a slot boundaries, all other users suppress the start of their transmission due to carrier sensing. Thus, the duration of a transmission period (whether successful or not) is $1 + 1/a$ slots. Let each user have an arrival with probability g in any slot.

In addition, it can be assumed that each nonempty user (who has a buffered packet) start transmission with probability p at the slot boundaries following any idle slot.

Let M be the number of user that potentially can access to the network, and N the number of authorized users. To each authorized user, an N_s-length hopping sequence is assigned. The message is supposed to consist of N_t slots of size a. This means that the hopping sequence is integrally repeated $\lfloor N_t / N_s \rfloor$ times during the transmission of the message. In Fig.49, N_s N_c-length blocks of the message are shown.

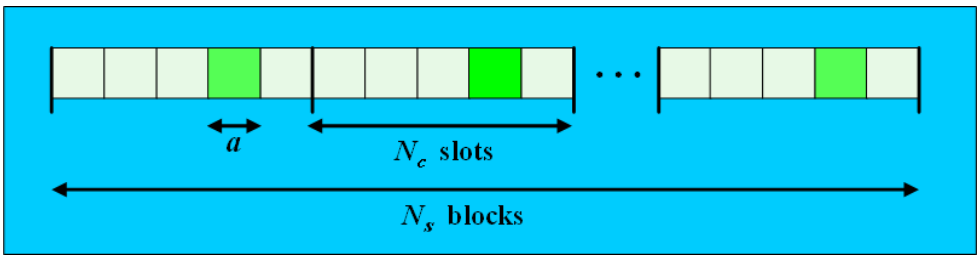

Fig.49 time hopping acces

In [39] the busy periods continue until there is at least one arrival during the last transmission time. In this more general case, a busy period continues until there is at least one arrival during the last N_c transmission times, i.e. during the last $N_c(1+1/a)$ slots.

Thus equation (5) of [39] becomes:

$$\Pr[J=j]=\left[1-(1-g)^{(1+1/a)M}\right]^{j-1}(1-g)^{(1+1/a)M} \tag{233}$$

and therefore $\bar{J}=(1-g)^{-N_c(1+1/a)M}$.

Furthermore equations (6) of [39] become:

$$\begin{aligned}\bar{B}&=\sum_{j=1}^{N_c}E\left[B^{(j)}\right]+\left(\bar{J}-N_c\right)E\left[B^{(N_c+1)}\right]\\ \bar{U}&=\sum_{j=1}^{N_c}E\left[U^{(j)}\right]+\left(\bar{J}-N_c\right)E\left[U^{(N_c+1)}\right]\end{aligned} \tag{234}$$

Equations (7) and (8) of [39] are unchanged while (9) of [39] is now:

$$\Pr\left[N_0^{(j)}=n\right]=\begin{cases}\Pi_n(1), & j=1\\ \Pi_n(2), & j=2\\ \vdots & \\ \Pi_n(N_c), & j=N_c\\ \Pi_n\left(N_c(1+1/a)\right), & j=N_c+1,\ N_c+2,\ \cdots\end{cases} \quad (235).$$

From equation (12) in [39] it is assured that

$$\mathrm{E}[R^{(j)}\mid N_0^{(j)}=n]=a\sum_{k=1}^{\infty}(1-p)^{kn}\left[\frac{p(1-g)^k-g(1-p)^k}{p-g}\right]^{\left\lceil\frac{M}{N_c}\right\rceil-n} \quad (236).$$

It must be noted as M has been replaced by $\lfloor M/N_c\rfloor$, due to the N_c-TH. Unconditioning (110) using (109):

$$\mathrm{E}[R^{(j)}]=\begin{cases}\sum_{n=1}^{\lfloor M/N_c\rfloor}\mathrm{E}[R^{(j)}\mid N_0^{(j)}=n]\Pi_n(j), & j=1,\ \cdots,\ N_c\\ \sum_{n=1}^{\lfloor M/N_c\rfloor}\mathrm{E}[R^{(j)}\mid N_0^{(j)}=n]\Pi_n\left(N_c(1+1/a)\right), & j=N_c+1,\ \cdots\end{cases} \quad (237).$$

Exploiting (2) of [39], and after some algebraic manipulations:

$$\begin{aligned}\bar{B}=\bar{J}(1+a)&+\sum_{j=1}^{N_c}\sum_{n=1}^{\lfloor M/N_c\rfloor}\mathrm{E}[R^{(j)}\mid N_0^{(j)}=n]\Pi_n(j)+\\ &+(\bar{J}-N_c)\sum_{n=1}^{\lfloor M/N_c\rfloor}\mathrm{E}[R^{(j)}\mid N_0^{(j)}=n]\Pi_n\left(N_c(1+1/a)\right)\end{aligned} \quad (238),$$

where $\bar{J}$, $\Pi_n(X)$ and $E\left[R^{(j)}\mid N_0^{(j)}=n\right]$ are respectively given by (1), (8) of [39] and (110).

The last quantity that must be evaluated in order to obtain the complete expression of S for the TH-Slotted-CSMA-p-Persistent proposed system, is quite $\bar{U}$. Equations (16) and (17) of [39], extending to the N_c-TH case, become:

$$E\left[U^{(j)}\mid R^{(j)}\geq ka, N_k^{(j)}=n+m, N_0^{(j)}=n\right]=\sum_{q=1}^{\min(n+m,N_c)}q\binom{n+m}{q}p^q(1-p)^{n+m-q} \quad (239),$$

$$\mathrm{E}[U^{(j)} \mid N_0^{(j)} = n] = \sum_{k=0}^{\infty} \sum_{m=0}^{\lfloor M/N_C \rfloor - n} \sum_{q=1}^{\min(n+m, N_C)} (1-p)^{kn} (1-g)^{k(M-n)} \cdot$$
$$\cdot \binom{\lfloor M/N_c \rfloor - n}{m} \left(\frac{g}{p-g} \right)^m \left[1 - \left(\frac{1-p}{1-g} \right)^k \right]^m q \binom{n+m}{q} p^q (1-p)^{n+m-q} \quad (240).$$

Unconditioning (114) using (11) in [39], after some algebraic manipulation,

$$\bar{U} = \sum_{j=1}^{N_C} \sum_{n=1}^{\lfloor M/N_C \rfloor} \mathrm{E}[U^{(j)} \mid N_0^{(j)} = n] \Pi_n(j) + (\bar{J} - N_c) \cdot$$
$$\cdot \sum_{n=1}^{\lfloor M/N_C \rfloor} \mathrm{E}[U^{(j)} \mid N_0^{(j)} = n] \Pi_n \left(N_c (1 + 1/a) \right) \quad (241),$$

where $\bar{J}$, $\Pi_n(X)$ and $E\left[U^{(j)} \mid N_0^{(j)} = n \right]$ are respectively given by (1), (8) of [39] and (114).

The useful part of every global transmission period T varies between 1 and N_c slots. This means that when inserting $\bar{U}$ into equation this one must be replaced by $\sum_{r=1}^{N_c} (r / N_c) \bar{U}$.

Thus the throughput of the analyzed system is:

$$S = \frac{\sum_{r=1}^{N_c} \frac{r}{N_c} \sum_{n=1}^{\lfloor M/N_c \rfloor} \left\{ \mathrm{E}\left[U^{(j)} \mid N_0^{(j)} = n \right] \left[\sum_{j=1}^{N_c} \Pi_n(j) + \left(\bar{J} - N_c \right) \Pi_n \left(N_c (1 + 1/a) \right) \right] \right\}}{\bar{I} + \bar{J}(1+a) + \sum_{n=1}^{\lfloor M/N_c \rfloor} \left\{ \mathrm{E}\left[R^{(j)} \mid N_0^{(j)} = n \right] \left[\sum_{j=1}^{N_c} \Pi_n(j) + \left(\bar{J} - N_c \right) \Pi_n \left(N_c (1 + 1/a) \right) \right] \right\}} \quad (242),$$

where $\bar{J}$, $\bar{I}$, $\Pi_n(X)$, $E\left[R^{(j)} \mid N_0^{(j)} = n \right]$ and $E\left[U^{(j)} \mid N_0^{(j)} = n \right]$ are respectively provided by (1), (7) of [39], (8) of [39], (110) and (114).

To emphasize the effect of the introduction of the TH, in Fig.50 S is plotted for different values of N_c.

In the described N_c-TH-Slotted-CSMA-p-Persistent system, the transmission period is made of $X_T = N_c(1 + 1/a)$ slots. Introducing the MAC security algorithm means that every period equal to R hopping-sequence repetition, i.e. RN_sX_T slots X_{elab} slots will be used by the AP, acting the role of coordinator of the network, to process the signal received from all the users, in order to check if the hopping sequences that they are using are admissible.

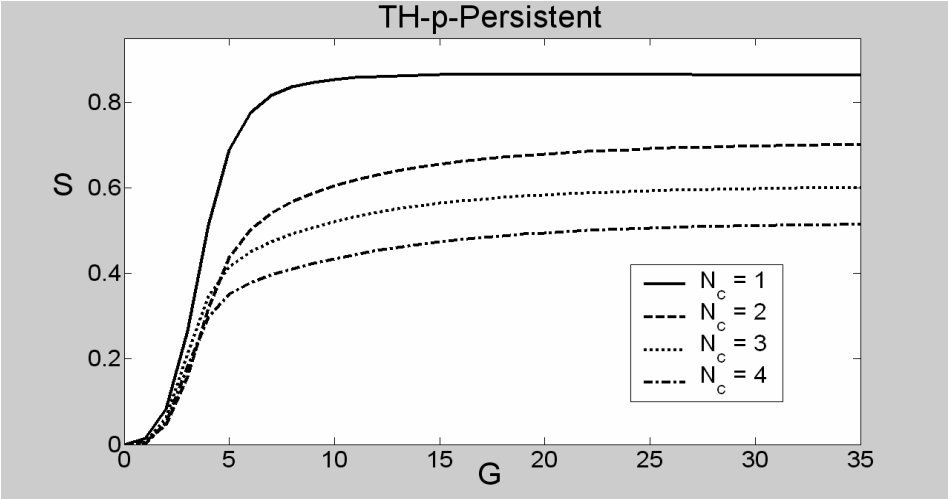

Fig.50 system throughput for different values of N_c (without attacks)

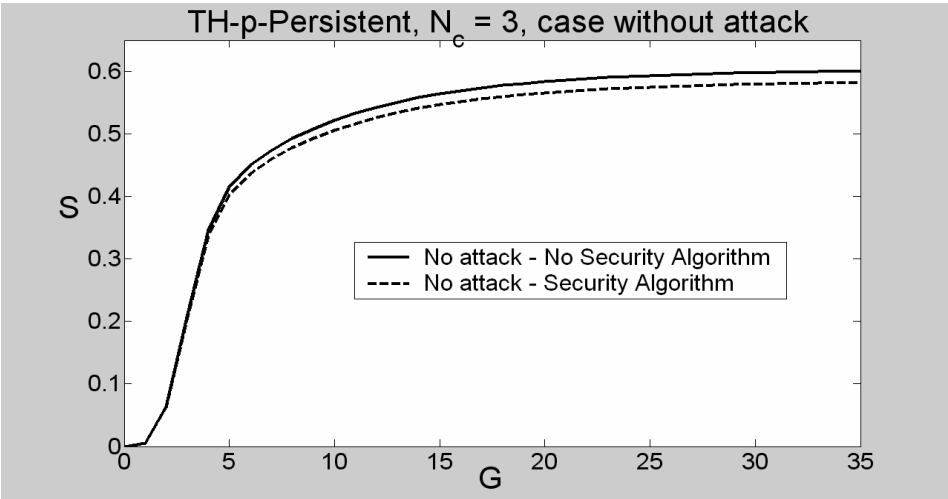

Fig.51 throughput loss due to the security algorithm (absence of attacks)

If any user is found to use an incorrect sequence, its connection is immediately dropped. In this way the general throughput definition, provided by (106), must change into:

$$S = \frac{\bar{U}}{\bar{I} + \bar{B} + \bar{T}_{elab}} \qquad (243),$$

where $\bar{T}_{elab} = T_{elab} = aX_{elab}$ is the signals processing period in temporal units. Being $\bar{J}$ the average number of observed sub-periods, it is clear that since X_{elab} processing slots follow RN_s transmission periods, everything goes like $X_{elab}\bar{J}/RN_s$ processing slots follow $\bar{J}$ sub-

periods. Thus, setting X_{elab} / RN_s = q_{elab}, the throughput expression for the considered TH-Slotted-CSMA-p-Persistent System with the described physical level security algorithm becomes:

$$S = \frac{\overline{U}}{\overline{I} + \overline{B} + q_{elab}\overline{J}} \qquad (244).$$

In Fig.51 the throughput of the system with and without the security algorithm are compared. Both curves refer to a scenario with $M = 10$, $a = 0.01$, $p = 0.02$, $N_c = 3$ and $q_{elab} = 0.1$.

5.2 System behaviour in presence of attacks

The advantage in using the physical level security algorithm described in the last paragraph is given by the possibility of detecting an attack after X_{elab} + X_{block} slots. If the attack start immediately before a processing period, it results $X_{block} = X_{block}^{(min)} = 0$ slots, while if it starts immediately after a processing period, $X_{block} = X_{block}^{(max)} = RN_s X_T$ slots. If the starting time of the attack is uniformly distributed, the average number of slots spent in blocking the attack can be seen as $X_{block} = \overline{X}_{block} = RN_s X_T / 2$. Next, let refer to a time observation window lasting K communication periods, i.e. KR hopping-sequence repetition periods, equivalent to KRN_sX_T slots. It is clear that, during such a temporal window, averagely $\overline{X}_{block}$ slots will be used to stop the attack, while during the remaining $KRN_sX_T - \overline{X}_{block} = RN_sX_T(K - 1/2)$ slots, all the N_c TH slots will be used for the admitted communications.

If N_a is the average number of TH slots used by unauthorized users, the term $\overline{U}$ of equation (118) can be thought as the sum of two different terms: 1) $\overline{U}_{N_c - N_a}$ (when only $N_c - N_a$ of N_c slots are available for authorized transmissions and so contribute to the system throughput), that is globally long as X_{block} slots; 2) $\overline{U}_{N_c}$ (when dealing with full slots availability), globally equivalent to $KRN_sX_T - X_{block}$ slots. It is easy to understand that, using the considered

temporal observation window, $\bar{U} = \bar{U}_{N_c - N_a} + \bar{U}_{N_c} \Leftrightarrow KRN_s X_T$ slots (where $\Leftrightarrow$ stand for "is part of").

Thus, when $X_{block} = X_{block}^{(max)}$:

$$\bar{U}_{N_C - N_a} \Leftrightarrow \left(\frac{N_c - N_a}{N_c}\right) RN_s X_T = \frac{N_c - N_a}{KN_c}\bar{U} \qquad \bar{U}_{N_C} \Leftrightarrow RN_s X_T (K-1) = \frac{K-1}{K}\bar{U} \quad (245),$$

this leads to

$$\bar{U}_{N_C - N_a} + \bar{U}_{N_C} \Leftrightarrow \left(1 - \frac{N_a}{N_c} + K - 1\right)\frac{\bar{U}}{K} = \left(1 - \frac{N_a}{KN_c}\right)\bar{U} \qquad (246).$$

If $X_{block} = \bar{X}_{block}$:

$$\bar{U}_{N_C - N_a} \Leftrightarrow \left(\frac{N_c - N_a}{N_c}\right)\frac{RN_s X_T}{2} = \frac{N_c - N_a}{2KN_c}\bar{U} \qquad \bar{U}_{N_C} \Leftrightarrow RN_s X_T \left(K - \frac{1}{2}\right) = \frac{2K-1}{2K}\bar{U} \quad (247),$$

$$\bar{U}_{N_C - N_a} + \bar{U}_{N_C} \Leftrightarrow \left(1 - \frac{N_a}{N_c} + 2K - 1\right)\frac{\bar{U}}{2K} = \left(1 - \frac{N_a}{2KN_c}\right)\bar{U} \qquad (248).$$

Finally, when $X_{block} = X_{block}^{(min)}$:

$$\bar{U}_{N_C - N_a} \Leftrightarrow 0 \qquad \bar{U}_{N_C} \Leftrightarrow \bar{U} \qquad (249),$$

$$\bar{U}_{N_C - N_a} + \bar{U}_{N_C} \Leftrightarrow \bar{U} \qquad (250).$$

For these reasons, the throughput of the system enforced by the security algorithm when, over a K communication periods long observation window, N_a attacks are forwarded and, then, stopped in $\bar{X}_{block} = RN_s X_T$ slots (worst case referred as $S^{(min)}$), $\bar{X}_{block} = RN_s X_T / 2$ (mean case denoted by $S^{(avg)}$) and $\bar{X}_{block} = 0$ (best case $S^{(max)}$), is given by:

$$S^{(min)} = \left(1 - \frac{N_a}{K N_c}\right) \frac{\bar{U}}{\bar{I} + \bar{B} + \bar{J}q_{elab}}$$
$$S^{(avg)} = \left(1 - \frac{N_a}{2K N_c}\right) \frac{\bar{U}}{\bar{I} + \bar{B} + \bar{J}q_{elab}} \qquad (251).$$
$$S^{(max)} = \frac{\bar{U}}{\bar{I} + \bar{B} + \bar{J}q_{elab}}$$

When the network is not protected by the discussed authentication mechanism, an attack can continue until it ends without any interruption. In other words, an unauthorized user that, in some way, knows the higher ISO-OSI stack tricks to access to the network, can definitively utilize the service although he/she is not in possess of the correct physical level hopping sequence.

Thus, still referring to a temporal observation window of KR bits, if N_a of the N_c available slots are used for non admitted transmissions, during all the observation window, the global throughput will be reduced from S to $\left(1 - N_a / N_c\right) S$, where S is defined in (106). Therefore:

$$S = \left(1 - \frac{N_a}{N_c}\right) \frac{\bar{U}}{\bar{I} + \bar{B}} \qquad (252).$$

The throughput of a TH-Slotted-CSMA-*p*-Persistent System with MAC/physical level security algorithm is compared with the throughput of a TH-Slotted-CSMA-*p*-Persistent System without the authentication algorithm, in presence of external attacks, managed by the parameter N_a. The other quantities are instead $p = 0.02$, $M = 10$, $N_c = 3$ and $K = 30$.

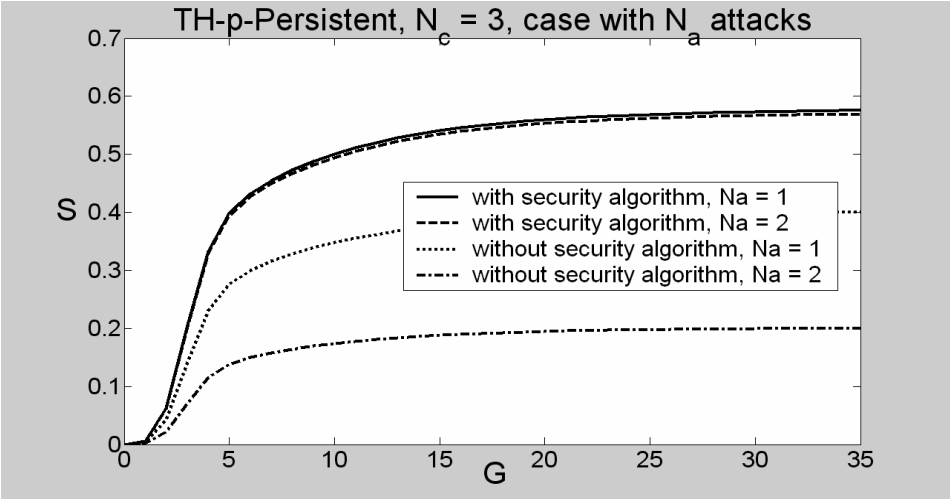

Fig.52 advantage of using TH security algorithm when one or more attacks are forwarded

5.3 Conclusions

With respect to [39], the system stays in the *Busy* condition if there is at least one arrival during the last transmission period. In a N_c-TH-Slotted-CSMA-*p*-Persistent system that period is N_c times longer, so the probability that the state of the system becomes *Idle* is lower. This justifies why, in absence of attacks, the throughput decreases linearly with Nc. When one or more attacks are forwarded, the throughput remains almost the same.

References

[1] *Digital Modulation Techniques*, Fuqing Xiong, Artech House, 2000.

[2] *Secure OFDM-UWB Communications based on Phase-Hopping*, SPIE Proceedings Vol. 6579, Mobile Multimedia/Image Processing for Military and Security Applications 2007, Sos S.Agaian; Sabah A. Jassim, Editors, 2May 2007.

[3] *Near Optimum Error Correcting Coding and Decoding: Turbo-Code*s, IEEE Transaction on Communications, vol. 44, no. 10, pp. 1261–71, C. Berrou and A. Glavieux , Oct. 1996.

[4] *Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes*, IEEE Transaction on Information Theory, Vol.42, pp.409-429, Sergio Benedetto, Guido Montorsi, March 1996.

[5] *Design of Parallel Concatenated Convolutional Codes*, IEEE Transaction on Communications, vol. 44, no. 5, pp. 591-600, Sergio Benedetto, Guido Montorsi, May 1996.

[6] *A search for good convolutional codes to be used in the construction of Turbo Codes*, IEEE Transaction on Communications, Vol. 46, pp. 1101-1105, S.Benedetto, R.Garello, G.Montorsi, September 1998.

[7] *On the design of Turbo Codes*, TDA Progress Report 42-123, pp. 99-121, D.Divsalar, F.Pollara, November 1995.

[8] *Comprehensive comparison of Turbo-Code decoders*, 45th IEEE Vehicular Technology Conference, vol. 2, pp. 624-628, Digital Object Identifier 10.1109/VETEC.1995.504943, P. Jung, M.M. Nasshan, 25-28 July 1995

[9] *Optimal Decoding of Linear Codes form Minimizing Symbol Error Rate* IEEE Transaction on Information Theory, pp.284-287, L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv, March 1974.

[10] *A Viterbi Algorithm with Soft Decision Outputs and its Applications*, Global Telecommunications Conference, 1989, and Exhibition. Communications Technology for the 1990s and Beyond. GLOBECOM apos; 89., IEEEVolume , Issue, pp. 1680 – 1686, vol.3. Digital Object Identifier 10.1109/GLOCOM.1989.64230 J.Hagenauer e Peter Hoeher, 27-30 Nov 1989.

[11] *Turbo Equalization*, IEEE Signal Processing Magazines, Vol. 21, no. 1, pp. 67-80, R. Koetter, A.C. Singer and M. Tuchler, January 2004.

[12] *Joint Security and Channel Coding for OFDM communications*,16th European Signal Processing Conference (EUSIPCO) 2008, A. Neri, P. Campisi, D. Blasi, L. Gizzi, Lausanne, Switzerland, 25-29 August 2008.

[13] *Secure Communication over fading channel*, IEEE Transaction on Information Theory, vol. 54, no. 6, pp. 2470–2492, L. Yingbin, H.V. Poor, S. Shamai, June 2008.

[14] *On the cardinality of systematic authentication codes via error-correcting codes*, IEEE Transaction on Information Theory, vol. 42, pp. 566-578, G. A. Kabatianskii, B. Smeets, and T. Johansson, March 1996.

[15] *Error-correcting codes for authentication and subliminal channels*, IEEE Trans. on Information Theory, vol. 37, pp. 13-17, R. S. Safavi-Naini, J. R. Seberry, January 1991.

[16] *Digital signature scheme based on error-correcting codes*, Electronics Letters, vol. 26, pp. 898-899, W. Xinmei, June 1990.

[17] *Cryptanalysis and modification of digital signature scheme based on error-correcting code*, Electronics Letters, vol. 28, pp. 157-159, L. Harn and D.-C. Wang, January 1992.

[18] *A proposal of a cryptography algorithm with techniques of error correction*, Computer Communications, vol. 20, no. 15, pp. 1374-1380, W. Godoy Junior and D. Pereira Junior, 1997.

[19] *Cryptanalysis of the Hwang-Rao secret error-correcting code schemes*, in Information and Communications Security, Third International Conference, ICICS 2001, Xian, China (S. Qing, T. Okamoto, and J. Zhou, eds.), vol. 2229 of Lecture Notes in Computer Science, pp. 419–428, Springer-Verlag, (obtained online at http://crypto.nknu.edu.tw/publications/icics2001.pdf). K. Zeng, C.-H. Yang, and T. R. N. Rao, 2001.

[20] *On the design of error detection and correction cryptography schemes*, in EUROCOMM 2000: Information Systems for Enhanced Public Safety and Security, IEEE, AFCEA, IEEE Communications Society, IEEE, 2000. Munich, Germany, N. V. Patsei and P. P. Urbanovich, 2000.

[21] *Authentication, enhanced security and error correcting codes*, in Advances in Cryptology - Crypto ’98 (H. Krawczyk, ed.), vol. 1462 of Lecture Notes in Computer Science, Springer-Verlag, 1998, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, Y. Aumann and M. O. Rabin, August 1998.

[22] *Joint encryption and Error Correction Schemes*, ACM SIGARCH Computer Architecture News, Vol.12, Issue 3, pp. 240-241, T.R.N. Rao, June 1984.

[23] *Cryptosystems using Algebraic codes*, Intl. Conf. On Computer Systems and Signal Proc. Bangalore, India, T.R.N. Rao, December 1984.

[24] *Private-Key Algebraic-Coded Cryptosystems*, Advances in Cryptology - CRYPTO '86: Proceedings, Volume 263/1987, pp. 35-48, Springer Berlin / Heidelberg, T. R. N. Rao, Kil-Hyun Nam, 1987.

[25] *Secret error-correcting codes (SECC)*, in Advances in Cryptology - CRYPTO ’88 (S. Goldwasser, ed.), vol. 403 of Lecture Notes in Computer Science, pp. 540–563, Springer-Verlag, T. Hwang and T. R. N. Rao, 1988.

[26] *Adaptive secure channel coding based on punctured turbo codes*, IEE Proc.-Commun., Vol. 153, No. 2, pp. 313-316, A. Payandeh, M. Ahmadian and M. Reza Aref, April 2006.

[27] *Joint Source, Channel Coding, and Secrecy*, EURASIP Journal on Information Security, Vol. 2007 (2007), Article ID 79048, 7 pages, doi:10.1155/2007/79048, E. Magli, M. Grangetto, and G. Olmo, 2007.

[28] *A survey of Information Authentication*, Proc. Of the IEEE, Vol. 76. N0.5, pp. 603-620, G. J. Simmons, May 1988.

[29] *Authentication theory and hypothesis testing*, IEEE Transactions on Information Theory, Vol. 46, No. 4, pp.1350 – 1356, , D.O.I. 10.1109/18.850674, U.M. Maurer, July 2000.

[30] *Near Optimum Error Correcting Coding and Decoding: Turbo-codes*, IEEE Transaction on Communications, Vol. 44, No. 10, pp. 1261-1271, C. Berrou and A. Glavieux, October 1996.

[31] *Information-theoretic bounds in authentication theory*, Proceedings of IEEE International Symposium on Information Theory, 1995, Page 12, U. M. Maurer, 17-22 Sept. 1995.

[32] *Authentication over Noisy Channels*, CoRR abs/0802.2701,2008, available at http://arxiv.org/abs/0802.2701, L. Lai, H. El Gamal, H. Vincent Poor.

[33] *The performances of interleavers used in turbo codes*, Int. Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005. Volume 1, Issue 14-15, pp. 363 – 366, M. Kovaci, H.G. Balta, M.M. Nafornita , July 2005.

[34] *Variable size Interleaver Design for parallel Turbo Decoder Architecture*, IEEE Transaction on Communications, Vol. 53, No. 11, pp. 1833-1840, S. Benedetto, L. Dinoi, November 2005.

[35] *Combined Turbo Codes and Interleaver Design*, IEEE Transaction on Communications, Vol. 47, pp. 484-487, J. Yuan, B. Vucetic, W. Feng, April 1999.

[36] *Computing the Minimum Distance of Turbo-Codes Using Iterative Decoding Techniques*, Proceedings of the 22th Biennial Symposium on Communications, Kingston, Ontario, Canada, pp. 306-308, S. Crozier, P. Guinand, A. Hunt, May/June 2004.

[37] *An Algorithm to compute the free-distance of Turbo Codes*, IEEE Proceedings on International Symposium on Information Theory 2000, page 287, R.Garello, P.Pierleoni, S.Benedetto, G.Montorsi, June 2000.

[38] *Transfer Function Bounds on the performance of Turbo Codes*, TDA Progress Report 42-122, JPL, Cal Tech, D.Divsalar, S.Dolinar, F.Pollara, 1995.

[39] *Throughput analysis for Persistent CSMA Systems*, IEEE Transaction on Communications, vol. 33, no. 7, pp. 627-638, H. Takagi, L. Kleinrock, July 1985.

[40] *Packet switching in radio channels: Part I-Carrier sense multiple-access modes and their throughput-delay characteristics*, IEEE Transaction on Communications, vol. 23, pp. 1400-14016, F. A. Tobagi, L. Kleinrock, December 1975.

[41] *Performance analysis of carrier sense multiple access with collision detection*, Computer Networks, vol. 4, pp. 245-259, F. B. Tobagi, V. B. Hunt, Oct-Nov 1980.

[42] *A theoretical performance analysis of polling and carrier sense collision detection communication systems*, Local Computer Networks, E. Arthurs, B. W. Stuck, 1982.

[43] *Stochastic Models in Operations Research*, Stochastic Processes and Operating Characteristics, Vol. I, McGraw-Hill, D. P. Heyman, M. J. Sobel, 1982.

A Francesca e Federico.