REGISTRI A SCORRIMENTO LINEARI

Daniele Argento

26 maggio 2004

1 Introduzione

Oggi oltre all'utilizzo di algoritmi matematici per la generazione di sequenze pseudocasuali (implementati solitamente in software), l'elettronica digitale ci offre dei particolari componenti, chiamati registri a scorrimento, i quali sono in grado di memorizzare dati in forma binaria e farli scorrere. Questa proprietà, se opportunamente trattata, gli permette di generare sequenze pseudocasuali di periodo ben determinato.

2 Registri a scorrimento

I generatori pseudocasuali più importanti sono i **registri a scorrimento**. Essi generano sequenze, quasi sempre binarie, dette pseudocasuali; il nome è dovuto al fatto che appaiono come se fossero casuali, anche se non lo sono. Ovviamente, con l'uso di una chiave pseudocasuale non si può avere una sicurezza perfetta, ma si desidera che i cifrari che le usano siano computazionalmente sicuri, cioè che il computo necessario per forzarli richieda tempi non accettabili (con tutti i mezzi a disposizione).

definizione Si definisce registro a scorrimento di lunghezza n, un dispositivo elettronico formato da n celle, numerate 0, 1, ..., n-1, collegate in serie, e da un clock. Ogni cella è in grado di memorizzare la quantità minima di informazione: il bit, di avere un input ed un output; il clock controlla il movimento del contenuto delle celle. Denotiamo con R_i la singola cella e con c_i i loro contenuti.

Nella figura è rappresentato un registro a scorrimento di lunghezza 4.

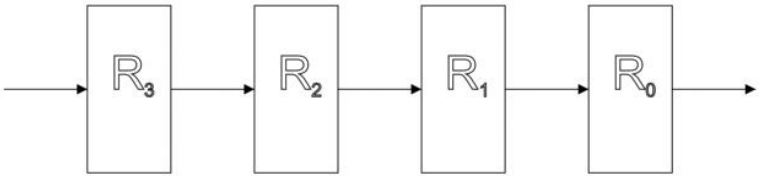


Figura 1: Registro a scorrimento di lunghezza 4.

Un registro a scorrimento di per se non riesce a generare una sequenza di periodo massimo perché ad ogni impulso di clock il contenuto della cella R_0 esce ed è un bit della sequenza degli output, il contenuto della cella R_i passa nella cella R_{i-1} per ogni $1 \leq i \leq n-1$, lasciando libera quindi la cella R_{n-1} del registro, da considerare in uno stato indefinito (nè uno nè zero) e di seguito tutte le altre.
A tal proposito introduciamo una funzione di retroazione, ovvero una particolare funzione di alcuni bit del registro, che combinati tra loro in modo lineare, cioè utilizzando solo la somma binaria (somma modulo 2), ci permette di riempire la locazione del registro rimasta vuota dopo lo spostamento, questo nuovo bit immesso prende il nome di **bit di retroazione o feedback**. Tale somma $\bigoplus$ è chiamata anche **XOR** (exclusive or).

Il tutto si può brevemente schematizzare nel modo seguente (c_i^t denota il contenuto della cella R_i all'istante t):

$$c_{i-1}^{(t-1)} = c_i^{(t)} \tag{1}$$

$$c_{n-1}^{(t+1)} = f(c_0^{(t)}, c_1^{(t)}, ..., c_{n-1}^{(t)}) \tag{2}$$

con f funzione di retroazione (se f lineare $\Rightarrow$ il sistema registro a scorrimento, funzione di retroazione prende il nome di LFSR (Linear Feedback Shift Register)).

Nella rappresentazione di un registro, le celle dalle quali parte una freccia rivolta verso l'alto sono quelle scelte per determinare il contenuto dell'ultima cella; le altre lavorano prendendo semplicemente al tempo i+1 il contenuto della cella precedente al tempo i. Ad esempio nel registro a scorrimento della figura seguente la retroazione è la somma dei contenuti delle prime tre celle.

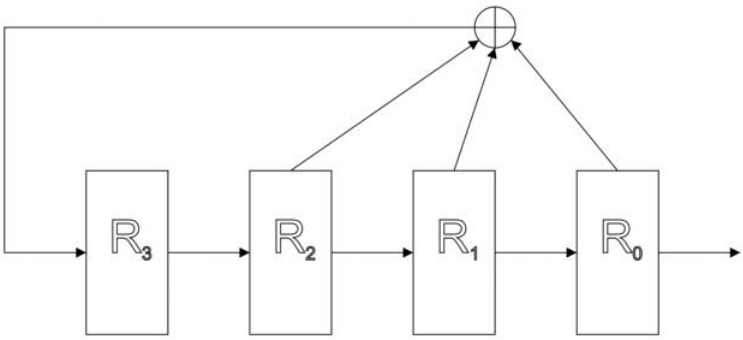


Figura 2: Registro a scorrimento con retroazione.

Come suddetto, il modo di operare per riempire ad ogni segnale di clock la cella R_{n-1} è determinato da una funzione, detta **funzione di retroazione**, che può essere descritta con un polinomio a coefficienti in Z_2. Nel caso in cui il registro sia lineare , l'equazione (1) non viene modificata, mentre l'equazione (2) si modifica nel modo seguente:

$$c_{n-1}^{(t+1)} = a_0 c_0^{(t)} + a_1 c_1^{(t)} + ... + a_{n-1} c_{n-1}^{(t)}, a_i \in \{0, 1\} \tag{3}$$

dove indichiamo con a_i, i = 0, 1,..., n-1, i coefficienti del polinomio di retroazione. Essi valgono 0 oppure 1, precisamente $a_i = 0$ se la cella R_i non è collegata all'addizionatore, in altre parole se non partecipa a riempire l'ultima cella; mentre $a_i = 1$ nel caso contrario. L'elenco dei bit che fanno parte del polinomio di retroazione viene chiamato **tap sequence**.

Ad un registro a scorrimento di lunghezza n si associa come funzione di retroazione il polinomio

$$f(x) = a_0 + a_1 x + ... + a_{n-1} x^{n-1} + x^n \tag{4}$$

cioè un polinomio monico, di grado n (pari alla lunghezza del registro stesso), avente come coefficienti i coefficienti a_i di retroazione (i coefficienti delle celle collegate con l'operazione di retroazione [$\bigoplus$]). Tale polinomio è detto **polinomio del registro**.

La sequenza degli output y_i di un registro a scorrimento lineare ha periodo p se p è il minimo intero positivo tale che sia $y_{i+p} = y_i$, per ogni naturale i (ovvero per periodo si intende il numero di generazioni di bit prima che la sequenza torni a ripetersi).

Chiamiamo **stato del registro** al tempo t l'insieme dei contenuti delle celle al tempo t.

I registri a scorrimento per loro natura hanno una relazione ben precisa tra stati interni e periodo della sequenza generata infatti, quando ritroviamo uno stato del registro uguale a quello iniziale, la sequenza degli output comincia a ripetersi. Allora il periodo di una sequenza pseudocasuale generata da un registro a scorrimento lineare è al più il numero massimo degli stati distinti, non nulli, del registro, cioè è $2^n - 1$. Infatti, assunta la funzione di retroazione lineare, qualora all'interno del registro si verificasse lo stato tutto nullo, il bit retroazionato sarebbe somma di bit nulli. In queste condizioni il registro a scorrimento assumerebbe sempre lo stato tutto nullo e quindi in uscita avremmo una sequenza tutta nulla da un certo bit in poi. Per tale motivo il polinomio di retroazione viene studiato in modo da escludere questa

eventualità, riducendo il periodo massimo a $2^n - 1$.
Attenzione però, realizzare un polinomio di retroazione che escluda lo stato nullo non implica che il registro generi una sequenza di periodo massimo. Per poter dire qualcosa relativamente al periodo di una sequenza pseudocasuale dobbiamo premettere due definizioni.

definizione Un polinomio $f(x)$ di $Z_2[x]$ si dice **irriducibile** se non può essere scritto come prodotto di due polinomi di $Z_2[x]$ aventi grado almeno uno.

definizione Un polinomio $f(x)$ di grado n, irriducibile in $Z_2[x]$, si dice **primitivo** se una sua radice è un generatore del gruppo moltiplicativo di $F_{2^n} = Z_2[x]/f(x)$.

Tornando al periodo di una sequenza pseudocasuale generata da un registro a scorrimento, abbiamo quanto segue.

Se $f(x)$ è un polinomio irriducibile, allora, al variare dello stato iniziale del registro, il periodo della sequenza pseudocasuale è costante e può essere minore del massimo.

Se $f(x)$ è un polinomio primitivo, allora il registro emette una sequenza di output con periodo massimo $2^n - 1$.

Concludendo dunque, affinchè il periodo della sequenza sia massimo si richiede che la funzione di retroazione, sia primitiva ovvero irriducibile in $GF(2)$.

proposizione Se consideriamo un registro R lineare di lunghezza n e periodo massimo $2^n - 1$, anche se si cambia la configurazione iniziale, il periodo resta massimo.

Per ogni lunghezza n, esiste almeno un polinomio primitivo di grado n, ma ci sono degli n per i quali i polinomi primitivi sono più numerosi. Questo accade, ad esempio, se n è un numero primo di *Mersenne*, cioè un primo del tipo $n = 2^m - 1$, con m primo.
Da quanto detto segue che il periodo massimo di una sequenza pseudocasuale cresce esponenzialmente con la lunghezza del registro, di conseguenza le sequenze, per n grande, sono praticamente aperiodiche.

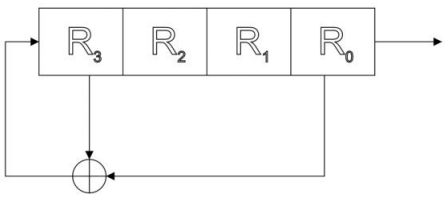
Ogni registro a scorrimento lineare, prima di essere usato come generatore di sequenze pseudocasuali deve essere inizializzato, ovvero bisogna introdurre la sequenza di uno e di zero da cui il generatore deve partire per generare la sequenza pseudocasuale. Tale valore, assieme al polinomio di retroazione $f(x)$, costituisce la chiave segreta che serve al mittente e al destinatario per ricostruire la sequenza pseudocasuale necessaria per le operazioni di cifratura e decifratura del messaggio in chiaro; quindi al variare di uno o di entrambi questi elementi , varia la sequenza degli output.
Dunque, per cambiare la chiave, si può cambiare lo stato iniziale ed usare la stessa funzione di retroazione, oppure cambiare la funzione di retroazione o entrambi.

Di seguito riportiamo un esempio di generazione di una sequenza pseudocasuale per mezzo di un LFSR:

Consideriamo un LFSR di lunghezza $n = 4$ e di polinomio caratteristico pari a: x^4+x^3+1. Il periodo massimo generalizzabile dal LFSR risulta essere $T_{max} = 2^n - 1 = 2^4 - 1 = 15$. Poichè il polinomio caratteristico è primitivo, la sequenza generata avrà periodo $T_{max} = 15$ ed il LFSR assumerà tutti i possibili stati interni distinti.

Nella prima colonna sono rappresentati tutti i possibili **stati interni**, nella seconda tutti gli **output** corrispondenti ai diversi segnali di **clock** rappresentati nella terza colonna.

Figura 3: LFSR con relativa tabella



1	1	1	1		-		1
0	1	1	1		1		2
1	0	1	1		1		3
0	1	0	1		1		4
1	0	1	0		1		5
1	1	0	1		0		6
0	1	1	0		1		7
0	0	1	1		0		8
1	0	0	1		1		9
0	1	0	0		1		10
0	0	1	0		0		11
0	0	0	1		0		12
1	0	0	0		1		13
1	1	0	0		0		14
1	1	1	0		0		15
1	1	1	1		0		16
0	1	1	1		1		17
1	0	1	1		1		18
0	1	0	1		1		19
1	0	1	0		1		20
1	1	0	1		0		21
0	1	1	0		1		22
0	0	1	1		0		23
1	0	0	1		1		24

teorema Per quanto riguarda le sequenze emesse da un registro lineare di lunghezza n e periodo massimo soddisfa i postulati di *Golomb* e quindi è pseudocasuale.

postulati di Golomb

- Il numero di 0 in una sequenza è approssimativamente uguale al numero di 1; $\frac{n}{2}$ se n è pari, $\frac{n+1}{2}$ se n è dispari.
  (Nell'esempio precedente $n = 15 \Rightarrow n$ dispari, $\frac{n+1}{2} = \frac{16}{2} = 8$ o 7)

- Metà dei **run** (sottosequenza della sequenza a costituita da elementi consecutivi uguali) ha lunghezza pari ad 1, un quarto ha lunghezza pari a 2 e così via.

I registri a scorrimento lineari non forniscono sequenze pseudocasuali utili dal punto di vista crittografico. Infatti se un attaccante sa che il testo cifrato è stato ottenuto usando un registro a scorrimento di lunghezza n e riesce a scoprire $2n$ bits di testo in chiaro ed i corrispondenti $2n$ bits di testo cifrato, può ricostruire la chiave facilmente, perchè deve solo risolvere in Z_2 un sistema lineare di n equazioni in n incognite. Illustriamo il problema con un esempio.

---

esempio Supponiamo di sapere che è stato usato un registro a scorrimento lineare di lunghezza 4. Se conosciamo 8 bits di testo in chiaro ed i corrispondenti bits di testo cifrato, sommando bit a bit, otteniamo 8 bit di chiave, come possiamo osservare dal seguente esempio numerico.

testo in chiaro	1 0 1 0 1 1 0 1
testo cifrato	0 1 1 0 0 1 0 1
xor ($\bigoplus$)	———————
chiave	1 1 0 0 1 0 0 0
chiave	$u_0u_1u_2u_3u_4u_5u_6u_7$

Gli otto bits della chiave costituiscono otto output consecutivi del registro (i primi 4 rappresentano proprio lo stato iniziale $u_0u_1u_2u_3u_4$). Applicando l'equazione di ricorrenza (3), trovo:

$$\begin{cases} u_4 = a_0u_0 + a_1u_1 + a_2u_2 + a_3u_3 \\ u_5 = a_0u_1 + a_1u_2 + a_2u_3 + a_3u_4 \\ u_6 = a_0u_2 + a_1u_3 + a_2u_4 + a_3u_5 \\ u_7 = a_0u_3 + a_1u_4 + a_2u_5 + a_3u_6 \end{cases} \tag{5}$$

sostituendo i valori si ottiene:

$$\begin{cases} 1 = a_0 + a_1 \\ 0 = a_0 + a_3 \\ 0 = a_2 \\ 0 = a_1 \end{cases} \Rightarrow \begin{cases} a_0 = 1 \\ a_3 = 1 \\ a_2 = 0 \\ a_1 = 0 \end{cases} \tag{6}$$

Dalle soluzioni ottenute è possibile ricostruire la funzione di retroazione:

$$f(x) = 1 + x^3 + x^4 \tag{7}$$

che risulta essere un polinomio primitivo.

A questo punto siamo tutti convinti che una sequenza pseudocasuale generata da un registro a scorrimento lineare, non dà alcuna sicurezza se usata come chiave di un cifrario.
Tuttavia i registri a scorrimento lineari non sono inutili neanche in crittografia, perchè possono essere usati per almeno i seguenti due scopi:

1. **complessità lineare**
2. **registri a scorrimento non lineari**
3. **immunità da correlazione**

1. Complessità lineare Usiamo i registri a scorrimento lineare per misurare la debolezza di una sequenza binaria qualsiasi. Data una sequenza binaria finita di lunghezza s, esiste sempre un registro a scorrimento lineare che la può generare; ad esempio un registro a scorrimento lineare di lunghezza s può generare certamente la sequenza detta, infatti è sufficiente prendere tale sequenza come stato iniziale del registro. Può esistere un registro a scorrimento lineare di lunghezza minore di s, capace di generare la sequenza detta. Chiamiamo **complessità lineare** di questa sequenza la lunghezza minima di un registro a scorrimento lineare capace di generarla; per calcolarla abbiamo

a disposizione un algoritmo efficiente, l'algoritmo di Berlekamp-Massey.
È chiaro che, se la complessità lineare di una sequenza pseudocasuale è piccola, la sequenza è debole da un punto di vista crittografico; d'altro canto un complessità lineare grande non assicura una buona sequenza pseudocasuale.

2. Registri a scorrimento non lineari Possiamo costruire registri a scorrimento non lineari, cioè con una funzione di retroazione non lineare. Inoltre possiamo combinare registri a scorrimento lineari in modo da ottenere sequenze pseudocasuali buone.

I generatori di sequenze pseudocasuali realizzati tramite LFSR possono essere classificati in due grandi categorie:

- generatori combinatori
- generatori clock controllati

3. immunità da correlazione Si definisce *immunità da correlazione* la probabilità che sussista una relazione privilegiata tra l'uscita del generatore di sequenze pseudocasuali e l'uscita di uno dei suoi LFSR interni. L'immunità da correlazione è importante perchè studiando, tramite un'algebra lineare, il comportamento di un LFSR otteniamo informazioni circa le sequenze interne. Usando queste informazioni ed altre relazioni, circa le varie sequenze interne del generatore, riusciamo a forzare il generatore stesso. Questo genere di attacco è chiamato **attacco correlato**. È importante ricordare che Thomas Siegenthaler ha dimostrato che l'immunità correlativa può essere precisamente definita e c'è una relazione esatta tra complessità lineare e immunità da correlazione.

Esistono diverse implementazioni hardware dei registri a scorrimento, quella finora vista ed anche più utilizzata nell'ambito della crittografia è la tipologia S.I.S.O..
Ecco la classificazione dei registri a scorrimento secondo il tipo di caricamento dei dati: Altre tipologie sono:

1. S.I.S.O. (Serial Input Serial Output): come già visto, l'ingresso e l'uscita dei dati avviene in modo seriale.

2. S.I.P.O (Serial Input Parallel Output): l'ingresso dei dati avviene in modo seriale, mentre l'uscita avviene in modo parallelo (tutti i bits contemporaneamente dopo un numero di shift adeguati).

3. P.I.S.O. (Parallel Input Serial Output): l'ingresso dei dati avviene in modo parallelo, mentre l'uscita avviene in modo seriale.

4. P.I.P.O. (Parallel Input Parallel Output): l'ingresso e l'uscita dei dati avviene in modo parallelo.