Presentazione di BunnyTN e prima valutazione di sicurezza

Stefano Martin

Università degli studi di Trento

3 marzo 2011

Block Cipher

Per costruire un Block Cipher abbiamo bisogno che esso riesca a resistere ai vari attacchi noti a cui esso può essere sottoposto.
Gli attacchi possono essere creati con lo scopo di voler ottenere:

- una key recovery;
- una global deduction;
- una key recovery parziale;
- una local deduction o
- un distinguisher.

Block Cipher

Le tipologie di attacco cambiano parecchio in relazione al tipo di attacco a cui si vuole sottoporre il Block Cipher ed, inoltre, si ha il problema effettivo di giudicare se l'attacco proposto è effettivamente efficace o meno.
Con sistemi di sicurezza con un insieme delle chiavi grande è difficile valutare se l'attacco porti dei vantaggi rispetto alla forza bruta in quanto farlo girare rispetto a tutti le chiavi avrebbe un costo troppo elevato.

BunnyTN

Per tale scopo si è voluto creare BunnyTN.

BunnyTN

BunnyTN è un BlockCipher che dovrebbe essere sicuro a tutti i possibili attacchi ma che ha la particolarità di possedere un insieme delle chiavi con solo 2^{24} chiavi.
Ciò che cercheremo di ottenere è che nessun attacco sia migliore della forza bruta.

BunnyTN

Usando una metafora è come se il nostro coniglio facesse un gran numero di buche per nascondersi e l'unico metodo per il cacciatore di trovare la buca dove si nasconde il coniglio fosse quello di guardare tutte le buche.
Cercare un metodo per trovare il coniglio, come la forma delle buche (analisi delle frequenze), l'erba schiacciata dal coniglio (attacco differenziale) o la disposizione delle buche (attacco distinguisher), non dovrebbe portare ad alcun efficace risultato se non ad un vano tentativo equivalente alla forza bruta.

Nell'analisi del Block Cipher passaremo spesso dallo spazio vettoriale $(\mathbb{F}_2)^6$ al campo $\mathbb{F}_{2^6}$; per comodità di lettura useremo le seguenti notazioni:

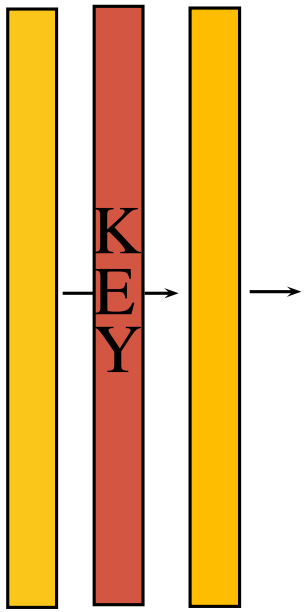
$$\mathbb{F} := \mathbb{F}_2$$

e

$$\mathbb{E} := \mathbb{F}_{2^6}.$$

Il polinomio primitivo usato per costruire $\mathbb{E}$ è stato:
$x^6 + x^4 + x^3 + x + 1$.
Chiameremo e l'elemento primitivo di $\mathbb{E}$.

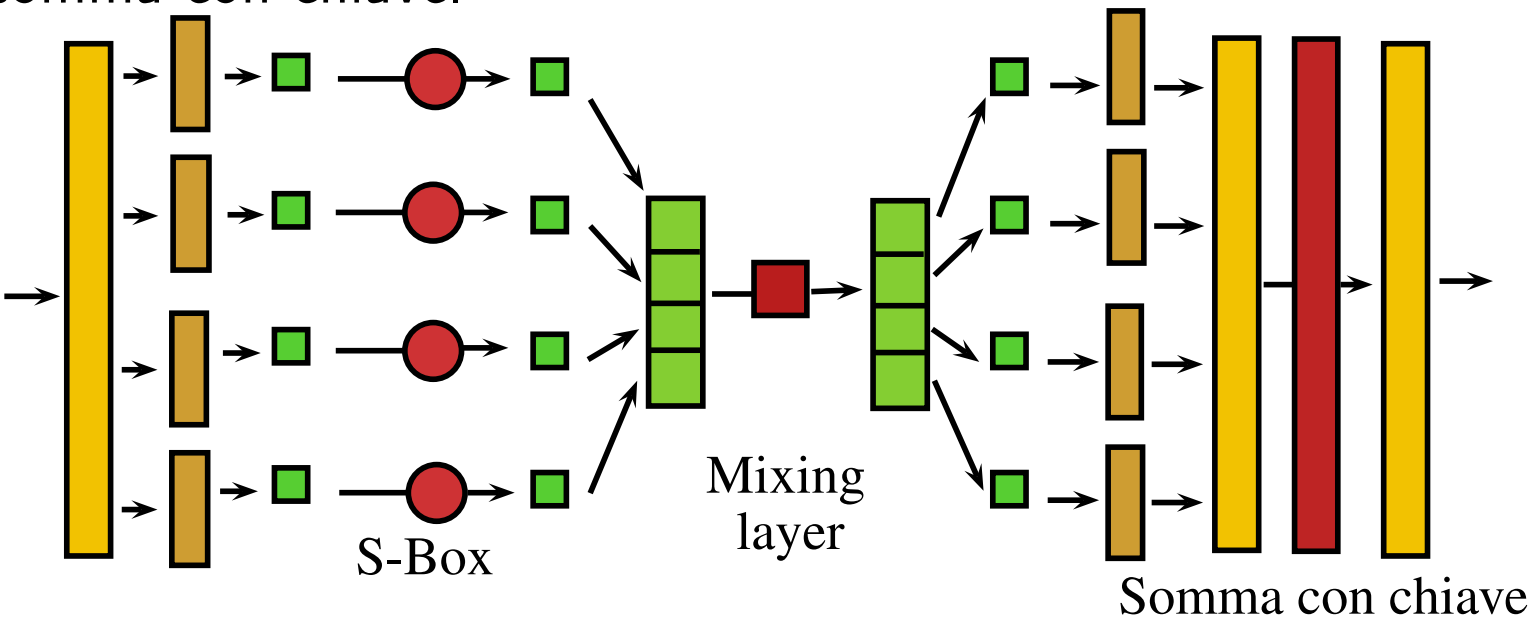
Whitening



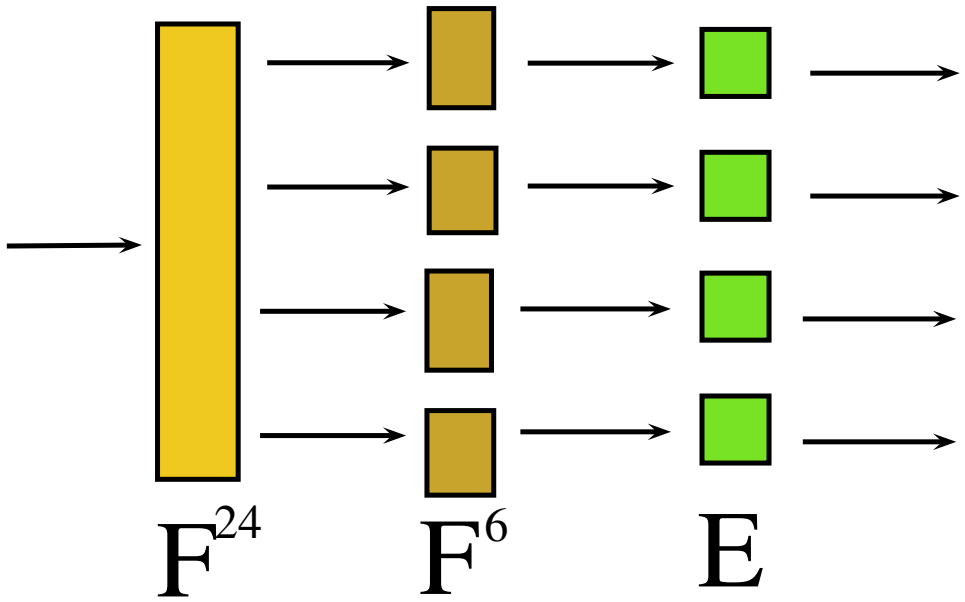
La prima operazione del BunnyTN è la somma con la chiave iniziale.

BunnyTN

I round successivi (tipici) sono costituiti da una funzione gamma non lineare (S-Box), una funzione lineare (Mixing layer) e la somma con chiave.

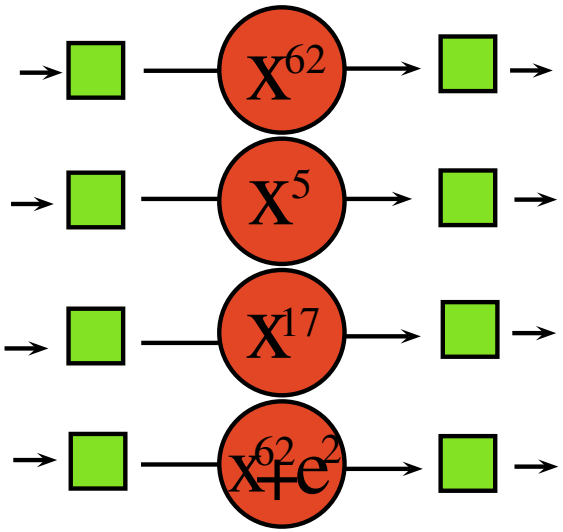


Cambio di campo



All'inizio del round passiamo dal vettore $\mathbb{F}^{24}$ a quattro valori nel campo $\mathbb{E}$.

S-Box



I quattro valori ottenuti passano nelle rispettive S-Box.
La prima e la quarta S-Box sono l'inversione x^{62} e l'inversione traslata $x^{62} + e^2$.
La seconda S-Box è x^5, mentre la terza è x^{17}.

S-Box

Abbiamo progettato queste S-Box per ottenere:

- invertibilità;
- alta non linearità N(S-Box) = 24 (Nmax = 28);
- 4 - differenziabilità uniforme.

Con un controllo a posteriori abbiamo notato che x^{62} è anche weakly-APN, mentre le altre due S-Box non lo sono.

S-Box

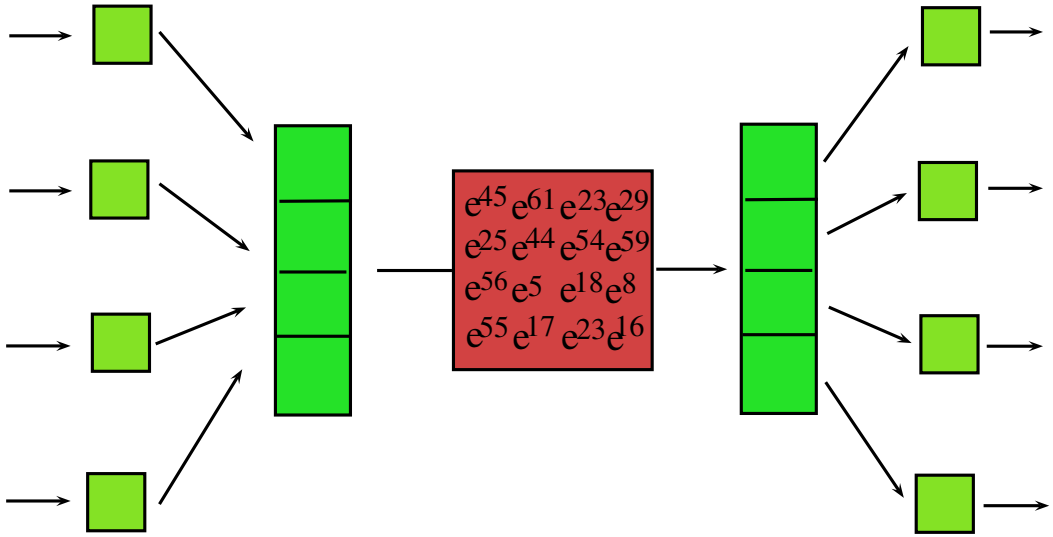
Definizione

Due funzioni $f, g : \mathbb{F}^n \to \mathbb{F}^n$ *si dicono affinamente equivalenti se esistono due matrici invertibili* A, B $n \times n$ *e due vettori* c *e* d *lunghi* n *tale che:*

$$A(f(Bx + c)) + d = g(x) \quad \forall x \in \mathbb{F}^n.$$

Abbiamo verificato che x^{62} non è affinamente equivalente con x^5 ed x^{17}.

Mixing Layer

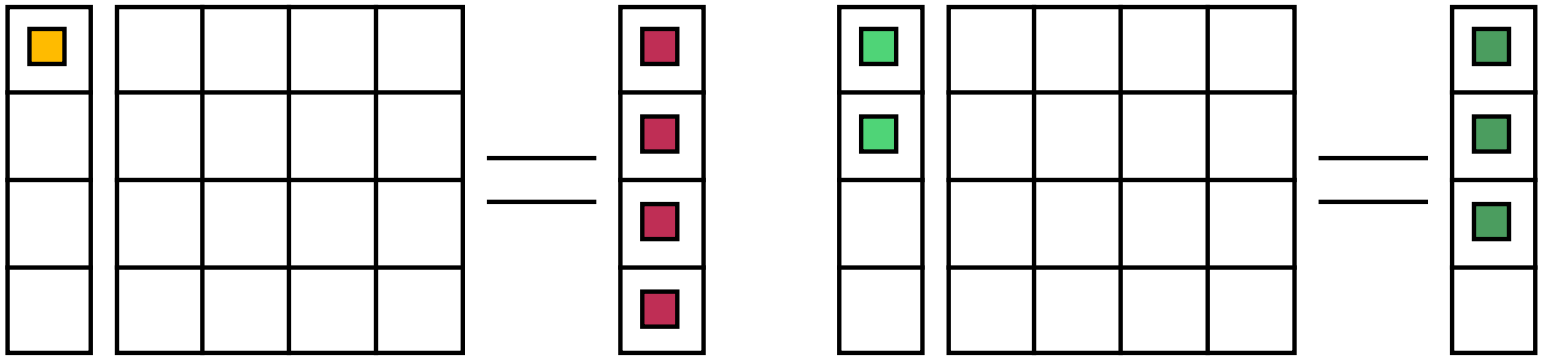


Moltiplichiamo il vettore in $\mathbb{E}^4$, ottenuto dalle S-Box precedenti, per una matrice 4×4 a coefficienti in $\mathbb{E}$; ciò che otterremo sarà un altro vettore in $\mathbb{E}^4$.

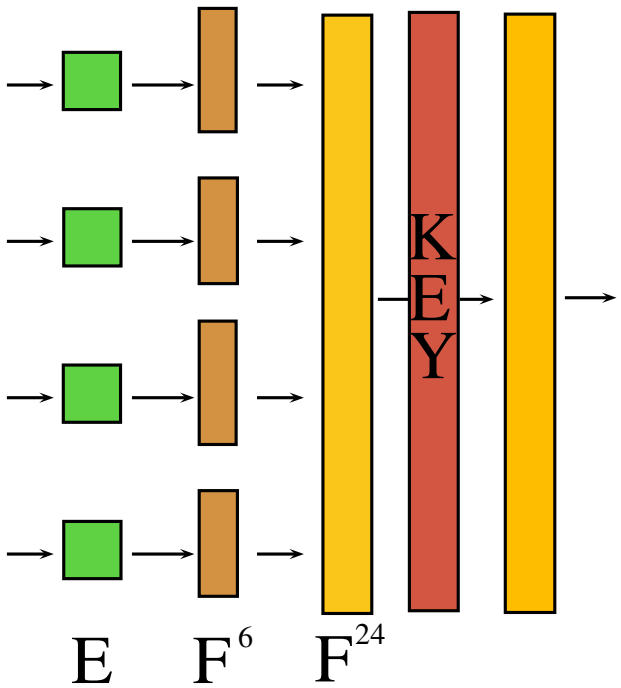
Mixing Layer

Per la costruzione della funzione lambda abbiamo considerato una matrice generatrice di un codice Reed-Solomon $[63, 4, 60]$.
Ne abbiamo estratto una matrice 4×4 e, affinchè fosse MDS, abbiamo verificato che avesse tutti i minori diversi da 0.

Questo ci garantisce una buona diffusione.



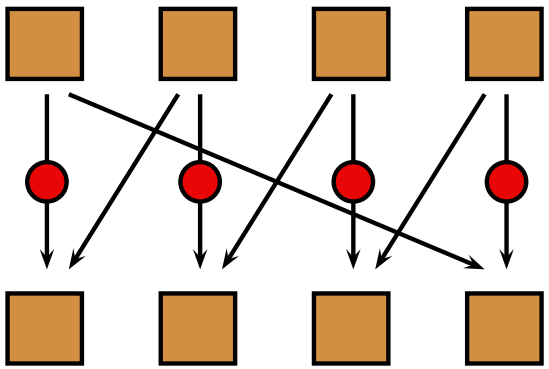
Somma con chiave



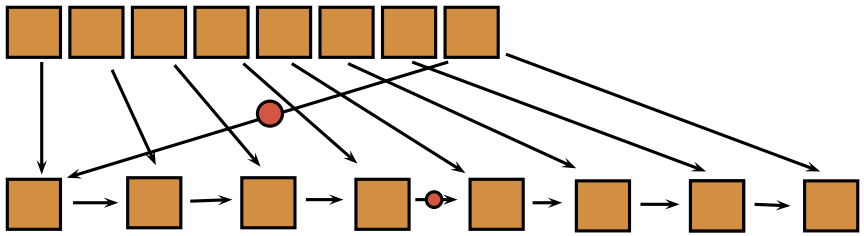
A questo punto facciamo tornare ogni valore nello spazio vettoriale $\mathbb{F}^6$ e le riunifichiamo in una nuova stringa $\mathbb{F}^{24}$.
Sommiamo il risultato ottenuto con la chiave di round.

Key-Schedule

Per costruire il Key Schedule abbiamo agito in tre passaggi.
Data una chiave in $\mathbb{F}^{24}$ abbiamo generato 4 stringhe $W_1, \ldots, W_4$ semplicemente spezzando la chiave.
Abbiamo generato $W_5, \ldots, W_8$ prendendo:
$W_i = SB(W_{i-4}) + W_{i-3} \quad i = 5, 6, 7$
$W_8 = SB(W_4) + W_1$.

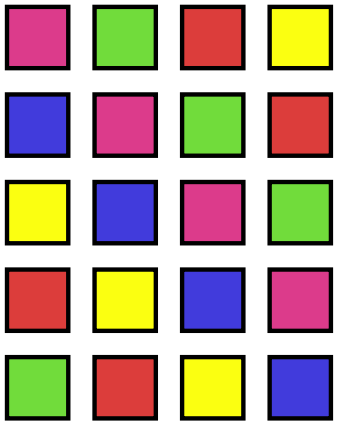


Key-schedule

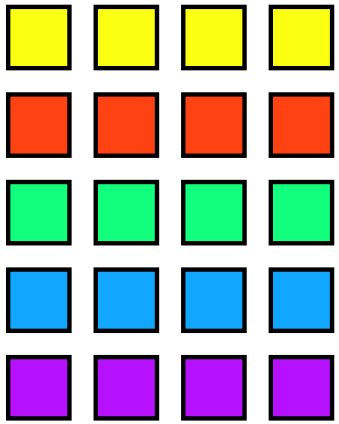


Dalle 8 stringhe $\mathbb{F}^6$ $W_1, \ldots, W_8$ abbiamo inizializzato gli altri vettori W_i come segue:
$W_i = W_{i-8} + W_{i-1}$ se $i \bmod 4 \neq 1$
$W_i = W_{i-8} + SB2(RB(W_{i-1})) + (1,0,1,0,1,0)$ se $i \bmod 8 = 1$
$W_i = W_{i-8} + SB3(W_{i-1})$ se $i \bmod 8 = 5$.
Dove SB2 è la seconda S-Box usata nel BunnyTN (cioè x^5) e SB3 è la terza S-Box usata nel BunnyTN (cioè x^{17}).
RB è lo spostamento di un bit verso sinistra.

Key-schedule



BunnyTN



AES

Abbiamo generato le chiavi di ciclo unendo ogni quartetto di W in vettori lunghi $\mathbb{F}^{24}$.
Per costruire tale chiavi abbiamo preso i valori W_i e li abbiamo suddivisi in gruppi da 20; abbiamo costruito le chiavi di round prendendo i valori in "diagonale".

Test statistici e scelta del numero di round

Per scegliere un numero adeguato di round abbiamo fatto uso dei principali test statistici che il NIST (National Institute of Standards and Technology) propone per la verifica della sicurezza dei block cipher.
I principali test statistici proposti dal NIST sono 15, verificano diverse caratteristiche dell'output del codice e le paragonano con le statistiche che dovrebbero risultare affinchè BunnyTN assomigli ad un insieme di valori casuali.

Test statistici e scelta del numero di round

Siamo partiti con i test statistici con minor costo di calcolo.

- Con un round: non sussiste alcuna sicurezza. Il Block Cipher fallisce tutti i test.
- Con due round: BunnyTN risulta ancora debole sull'"Approximate entropy test".
- Con tre round: sussistono problemi sul "Cumulative Sums" e sul "Longest Run Test".
- Con quattro round: non si hanno problemi.

Test statistici e scelta del numero di round

Avevamo scelto 12 round, in maniera conservativa, come tre volte il minimo dell'indistinguibilità, ma c'è risultata qualche debolezza nel "Random Excursion test" quindi abbiamo preferito scegliere 15 round.
I risultati sono stati incoraggianti tranne una lieve discrepanza nel NonPeriodic Template Matchings.

Test statistici e scelta del numero di round

Senza il primo passo del Key-schedule, per raggiungere una sicurezza adeguata, dovevamo usare 21 round anzichè 15.

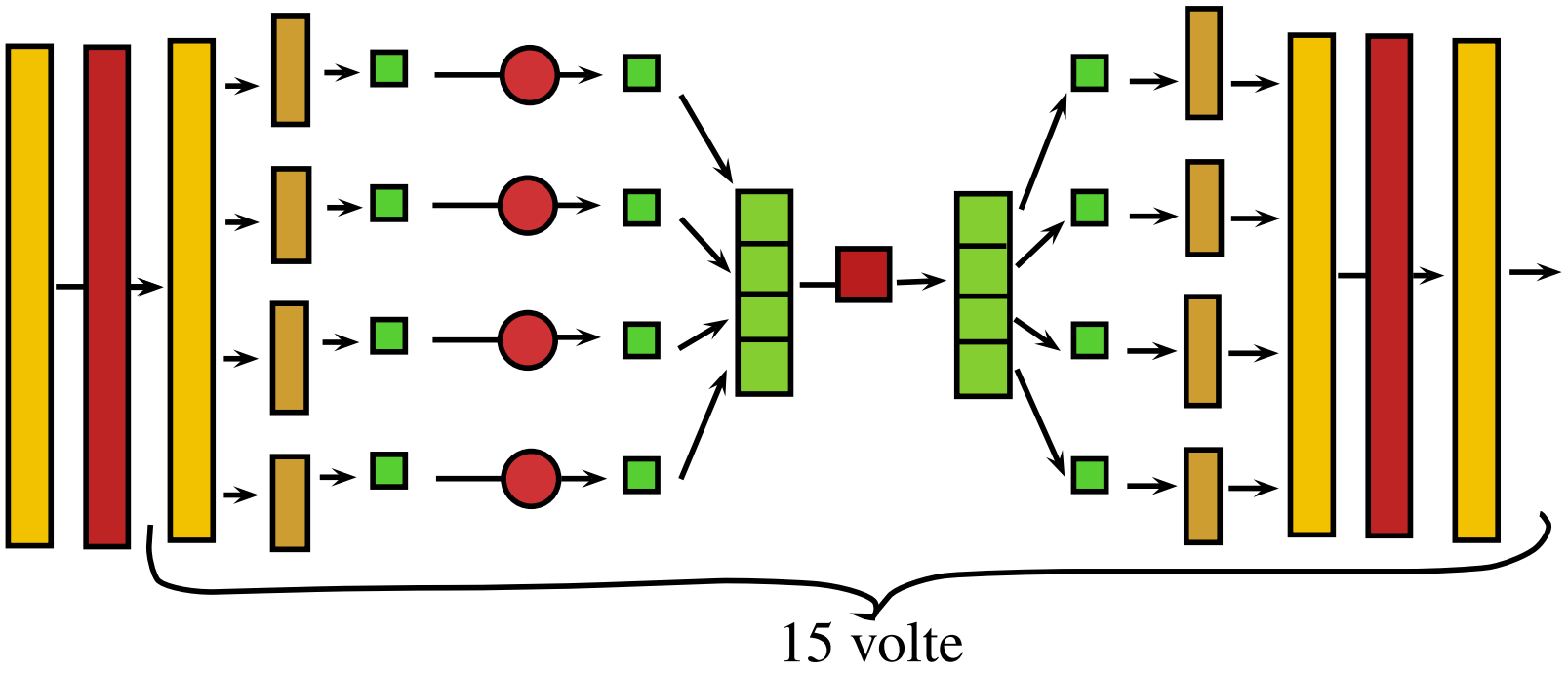
Test statistici e scelta del numero di round

Test con 2500000 di output, il test del NIST esamina 60 stringhe lunghe 1000000 bit.

Nome del test	P-value
Frequency	0.090936
Block Frequency	0.019188
Cumulative Sums (1 di 2)	0.366918
Runs	0.090936
Longest Run	0.162606
Rank	0.595549
FFT	0.304126
Non Over Lapping Template (1 di 148)	0.224821
Over Lapping Template	0.366918
Universal	0.595549
Approximate Entropy	0.224821
Random Excursion (1 di 8)	0.888137
Random Excursion Variant (1 di 18)	0.060239
Serial (1 di 2)	0.002559
Linear Complexity	0.181557

BunnyTN

In conclusione abbiamo il seguente Block Cipher BunnyTN costituito da un round d'inizializzazione e da 15 round tipici.



Obiettivi del valutatore

Finora il valutatore ha esaminato in parte l'S-Box, comunque il lavoro del valutatore è appena iniziato.

- L'esame sulla S-Box va approfondito ulteriormente provando ad usare S-Box alternative.
- Bisogna verificare se il Mixing Layer è proprio, fortemente proprio e valutare i suoi parametri di diffusione.
- Sul key-schedule si cercheranno metodi per invertirlo e per trovare dei possibili altri miglioramenti.

Obiettivi del valutatore

- Su tutto il Block Cipher si cercheranno differential trail per attacchi differenziali, compresi i related key.
- I test del NIST verranno compiuti con più cambiamenti di chiave.
- Si vorrà stimare il differenziale lineare e quello potenziale.
- Si cercheranno valutazioni statistiche globali come l'effective linearity.