Domande:
1) Che migliorie si possano fare a WEP considerando il suo utilizzo di RC4,
2) che cosa viene effettuato da Boon a fine analisi con esempio,
3) possibile difesa a SQL injection
4)Buffer Overflow possa semplicemente bloccare il programma vittima o meno

1) Le classifiche CWE e OWASP cambiano nel tempo? Perché?
2) Per l'attacco buffer overflow le istruzioni no-op sono utili? Perché?
3) Differenze fra KSA e RNG di RC4, dicendo scopo, struttura e ricorrenza di entrambi
4) Perché in Spin e Promela è possibile verificare delle proprietà in un futuro illimitato?

Boon

char b[200] $\{200\} \leq alloc(b)$

strcpy (s, a) $len\{a\} \leq len\{b\}$

strncpy (b, a, n) $min(len\{a\}, n) \leq len\{b\}$

s = "Hello!" $7 \leq len(s)$, $7 \leq alloc(s)$

s[n] = '\0' $min(len(s), n+1) \leq len(s)$

KSA → usato una volta sola inizializza l'array S + aggiunge rumore

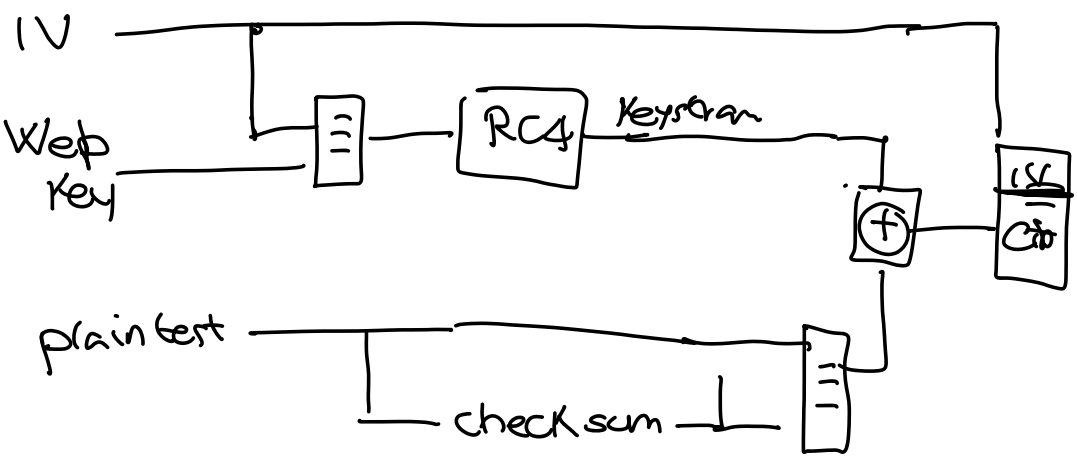
```
for i=0 to 255
    S[i] = i
for i=0 to 255
    J = J + S[i] + key[i]  mod 256
    swap S[i] & S[J]
```

RNG → numero indefinito di volte, creazione di bit pseudocasuali

```
i=0
j=0

while ...
  i = (i+1) mod 256
  j = (j + S[i]) mod 256
  swap  S[i], S[j]
  K = S[ S[i] + S[j] ] mod 256
  output K
```

IV → 4byte



$$C_1 = P_1 \oplus K$$

$$C_2 = P_2 \oplus K$$

$$C_1 \oplus C_2 = (P_1 \oplus K) \oplus (P_2 \oplus K) =$$
$$= P_1 \oplus P_2$$

Model Checking → permette di modellare una certa risorsa

↓

sulla base di come modelliamo il problema vengono effettuate delle verifiche (check)

↘ Spin

↓ definizione del modello + check

↓ LTL

↳ Il fulcro del model checking!

↗ tempo infinito